

«С.Сейфуллин атындағы Қазақ агротехникалық зерттеу университеті» КеАҚ

ӘОЖ 004.032.28:004.94 (043.3)

Қолжазба құқығында

МУТАЛОВА ЖАЗИРА САТКАНОВНА

**Нейрондық желілер негізінде бейне ағындардағы кескінді танудың
математикалық моделін әзірлеу**

8D06101 – Үлкен деректер аналитикасы

Философия докторы (PhD)
дәрежесін алу үшін дайындалған диссертация

Ғылыми кеңесшісі
техника ғылымдарының кандидаты,
қауымдастырылған профессор
Шаушенова А. Г.

Шетелдік ғылыми кеңесшісі
техника ғылымдарының докторы,
профессор Гриф М.Г.
(Новосібір, Ресей)

Қазақстан Республикасы
Астана, 2026

МАЗМҰНЫ

НОРМАТИВТІК СІЛТЕМЕЛЕР.....	3
АНЫҚТАМАЛАР.....	4
БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР.....	6
КІРІСПЕ.....	7
1 БЕЙНЕ АҒЫНДАРДАҒЫ КЕСКІНДЕРДІ ТАНУДЫҢ	14
ЗАМАНАУИ ӘДІСТЕРІНЕ ТАЛДАУ.....	14
1.1 Бейне ағындардағы кескіндерді танудың жалпы тәсілдері.....	14
1.2 Бейне ағындардағы кескіндерді өңдеуге арналған нейрондық желілер архитектуралары.....	21
1.3 Нақты уақыт режимінде бетті тану модельдері мен әдістеріне шолу....	27
Бірінші бөлім бойынша қорытынды.....	31
2 НЕЙРОНДЫҚ ЖЕЛІЛЕР НЕГІЗІНДЕ БЕЙНЕ АҒЫНДАРДАҒЫ	33
КЕСКІНДІ ТАНУДЫҢ МАТЕМАТИКАЛЫҚ МОДЕЛІН	33
ӘЗІРЛЕУ.....	33
2.1 Бейне ағындарындағы кескінді тану әдістері және математикалық модельдері	33
2.2 Шу кезінде бет кескіндерін тану дәлдігін жақсартудың адаптивті әдістері.....	40
2.3 Терең оқытуға негізделген бет-әлпетті танудың көпкезеңді моделі.....	50
Екінші бөлім бойынша қорытынды.....	73
3 NEXIFACE БИОМЕТРИЯЛЫҚ СӘЙКЕСТЕНДІРУ	75
ЖҮЙЕСІНІҢ АРХИТЕКТУРАЛЫҚ ШЕШІМДЕРІН ӘЗІРЛЕУ	75
ЖӘНЕ БАҒДАРЛАМАЛЫҚ ЖҮЗЕГЕ АСЫРУ.....	75
3.1 Бейне ағындарда тану дәлдігін арттырып, өңдеу уақытын қысқартатын көпкезеңді модельді іске асыру.....	75
3.2 Бетті тану тиімділігін арттыруға арналған нейрондық желі архитектурасын әзірлеу.....	79
3.3 Nexiface веб-платформасында бетті тану жүйесінің функциональдық жүзеге асырылуы.....	84
Үшінші бөлім бойынша қорытынды.....	118
ҚОРЫТЫНДЫ.....	119
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ.....	122
ҚОСЫМША А- Анықтама.....	133
ҚОСЫМША Ә - Оқу процесіне енгізу актісі.....	134
ҚОСЫМША Б - Өндіріске енгізу актісі.....	136
ҚОСЫМША В – Патент.....	138
ҚОСЫМША Г – Nexiface бетті тану жүйесінің бағдарламалық коды.....	139

НОРМАТИВТІК СІЛТЕМЕЛЕР

Диссертацияда келесі мемлекеттік стандарттар мен нормативтік құжаттарға сілтемелер пайдаланылады:

Диссертация мен авторефератты рәсімдеу жөніндегі нұсқаулық, ҚР БҒМ ЖАК, № 377-3ж.

МЕСТ 7.32-2017. Ақпарат, кітапхана және баспа ісі жөніндегі стандарттар жүйесі. «Ғылыми-зерттеу жұмысы туралы есеп. Құрылымы және рәсімдеу ережелері».

ҚР СТ 34.005-2002. Ақпараттық технологиялар. Негізгі терминдер және анықтамалар.

ҚР СТ 34.006-2002. Ақпараттық технологиялар. Деректер базалары. Негізгі терминдер және анықтамалар.

ҚР СТ 34.014-2002. Ақпараттық технология. Автоматтандырылған жүйелерге арналған стандарттар кешені. Автоматтандырылған жүйелер. Терминдер мен анықтамалар.

ҚР СТ 34.019-2005. (ISO/IEC 12207:1995, MOD) Ақпараттық технологиялар. Бағдарламалық құралдардың өмірлік циклінің процестері.

Қазақстан Республикасының Заңы. 2015 жылғы 24 қарашадағы № 418-V «Ақпараттандыру туралы» Заңы.

Қазақстан Республикасының Заңы. 2011 жылғы 18 ақпандағы N407-IV3 «Ғылым туралы» Заңы.

МЕСТ 7.1-2003. Библиографиялық жазба. Библиографиялық сипаттама. Құрастырудың жалпы талаптары мен ережелері.

Қазақстан Республикасының кодексі. 2020 жылғы 7 шілдеде, NO360-VI қабылданған «Халық денсаулығы және денсаулық сақтау жүйесі туралы» Кодексі.

АНЫҚТАМАЛАР

Диссертациялық жұмыста төмендегідей терминдер мен анықтамалар қолданылды:

Дерекқор (Database) — құрылымдық деректер жиынтығын ұзақ мерзімді сақтауға, жаңартуға және басқаруға арналған құрылымдық ақпараттық жүйе.

Үлкен деректер (ағылш. Big Data) — дәстүрлі деректерді өңдеу құралдары тиімді өңдей алмайтын көлемі үлкен және құрылымы күрделі деректер жиынтығы.

Жасанды интеллект — машиналардың ойлау, үйрену және шешім қабылдау қабілетін зерттейтін және жүзеге асыратын ғылыми сала.

Ақпаратты шығарып алу (Information Extraction) — құрылымдалмаған мәтіндерден автоматты түрде құрылымдалған деректерді алу процесі.

Бейнені тану (Image Recognition) — кескін немесе бейне негізінде объектілерді автоматты анықтау және жіктеу процесі.

Бейне ағын (Video Stream) — бейнематериалды толық жүктемей-ақ үздіксіз тарату және нақты уақыт режимінде көру технологиясы.

Объектіні анықтау (Object Detection) — кескін немесе бейнеде объектінің орналасуын табу әдісі.

Кескінді сегментациялау (Image Segmentation) — кескінді мағыналық аймақтарға (пиксель топтарына) бөлу процесі.

Объектілерді трекингтеу (Object Tracking) — уақыт бойынша қозғалып жатқан объектінің орнын анықтау процесі.

Машиналық көру (Computer Vision) — камера және сенсорлар арқылы алынған визуалды ақпаратты талдау жүйесі.

Конволюциялық нейрондық желі (CNN) — кескіндерді өңдеуге арналған, локалды аймақтарды талдайтын нейрондық желі түрі.

Рекуррентті нейрондық желі (RNN) — тізбекті деректерді (уақыттық қатарларды) өңдеуге арналған нейрондық желі.

Трансформер — тізбекті деректерді контекстпен бірге өңдейтін нейрондық желі архитектурасы.

Градиенттік түсуді оңтайландыру — функцияның минимумын немесе максимумын табуға арналған итерациялық сандық әдіс.

Активтендіру функциясы — нейронның шығыс сигналын анықтайтын және желіге бейсызықтық енгізетін функция.

Кескінді фильтрациялау — пиксель мәндерін өзгерту арқылы кескінді тегістеу, айқындау немесе шекараларын анықтау процесі.

Фурье түрлендіруі (Fourier Transform) — сигналды уақыт/кеңістік аймағынан жиілік аймағына түрлендіретін математикалық әдіс.

Бағытталған градиенттер гистограммасы (HOG) — кескіннің жергілікті аймақтарындағы градиент бағыттарын талдау негізінде объектілерді сипаттауға арналған әдіс.

Басты компоненттер талдауы (PCA) — деректер өлшемін азайту үшін оларды ортогоналды компоненттерге проекциялайтын статистикалық әдіс.

Дәлдікті бағалау (Accuracy, Precision, Recall, F1-score) — жіктеу моделінің сапасын бағалауға арналған негізгі метрикалар жиынтығы.

PyTorch — динамикалық есептеу графтарына негізделген терең оқыту фреймворкі, ғылыми зерттеулер мен прототиптеуде кең қолданылады.

TensorFlow — ауқымды және өндірістік деңгейде қолдануға арналған терең оқыту платформасы.

ResNet (Қалдықты нейрондық желі) — оқытуды жақсарту үшін қалдық байланыстарды пайдаланатын терең нейрондық желі архитектурасы.

OpenCV — нақты уақыт режимінде кескіндер мен бейнелерді өңдеуге арналған компьютерлік көру кітапханасы. Бағытталған градиенттер гистограммасы (HOG) — кескіннің жергілікті аймақтарындағы градиент бағыттарын талдау негізінде объектілерді сипаттауға арналған әдіс.

Басты компоненттер талдауы (PCA) — деректер өлшемін азайту үшін оларды ортогоналды компоненттерге проекциялайтын статистикалық әдіс.

Дәлдікті бағалау (Accuracy, Precision, Recall, F1-score) — жіктеу моделінің сапасын бағалауға арналған негізгі метрикалар жиынтығы.

PyTorch — динамикалық есептеу графтарына негізделген терең оқыту фреймворкі, ғылыми зерттеулер мен прототиптеуде кең қолданылады.

TensorFlow — ауқымды және өндірістік деңгейде қолдануға арналған терең оқыту платформасы.

OpenCV — нақты уақыт режимінде кескіндер мен бейнелерді өңдеуге арналған компьютерлік көру кітапханасы.

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР

AI (Artificial Intelligence)	Жасанды интеллект
NN (Neural Network)	Нейрондық желі
DNN (Deep Neural Network)	Терең нейрондық желі
CNN (Convolutional Neural Network)	Конволюциялық нейрондық желі
RNN (Recurrent Neural Network)	Рекуррентті нейрондық желі
LSTM (Long Short-Term Memory)	Ұзақ қысқа мерзімді жад нейрондық желісі
GRU (Gated Recurrent Unit)	Рекурренттік желі модификациясы
ViT (Vision Transformer)	Кескінді өңдеуге арналған трансформерлер
GAN (Generative Adversarial Network)	Генеративті қарсылас желі
ML (Machine Learning)	Машиналық оқыту
DL (Deep Learning)	Терең оқыту
CV (Computer Vision)	Компьютерлік көру
YOLO (You Only Look Once)	Нақты уақыт режимінде объектілерді анықтау моделі
SSD (Single Shot MultiBox Detector)	Объекті біркезеңде анықтау әдісі
Faster R-CNN (Faster Region-based CNN)	Анықтауға арналған R-CNN жақсартылған версиясы
RCNN (Region-based CNN)	Объектілерді анықтауға арналған модельдер жиынтығы
HOG (Histogram of Oriented Gradients)	Бағытталған градиенттер гистограммасы
SIFT (Scale-Invariant Feature Transform)	Негізгі нүктелерді анықтау әдісі
mAP (Mean Average Precision)	Анықтаудың орташа дәлдігі
TF (TensorFlow)	Терең оқытуға арналған фреймворк
PT (PyTorch)	Терең оқыту кітапханасы
MLOps (Machine Learning Operations)	ML-модельдерді енгізу және сүйемелдеу
RT (Real-Time)	Нақты уақытта деректерді өңдеу
ДҚ	Деректер қоры
АЖ	Ақпараттық жүйелер
ДҚБЖ	Деректер қорын басқару жүйесі
БҚЕ	Бағдарламалық қамтамасыз ету

КІРІСПЕ

Зерттеу тақырыбының өзектілігі. Диссертациялық жұмыстың тақырыбы Қазақстан Республикасы Үкіметінің 2024 жылғы 24 шілдедегі №592 қаулысымен бекітілген «Жасанды интеллектті дамытудың 2024–2029 жылдарға арналған тұжырымдамасымен» байланысты екендігін аңғаруға болады. Аталған тұжырымдама жасанды интеллект технологияларын дамыту мен оларды экономика мен мемлекеттік басқарудың түрлі салаларына енгізуге арналған. Цифрлық трансформацияны қамтамасыз етуге, озық цифрлық технологияларды пайдалану есебінен халықтың өмір сүру деңгейін арттыруға бағытталған стратегиялық құжат ретінде қолдануға болатындығын көрсетті [1]. Бейнелерді тану технологиялары Цифрлық трансформация және жасанды интеллектті дамыту дәуірінде Қазақстандағы стратегиялық міндеттерді шешудің маңызды құралына айналды. Тану технологияларын экономика мен басқарудың стратегиялық маңызды салаларында пайдалану Қазақстан үшін басымдық болып отыр. Олардың дамуы қауіпсіздікті нығайтады. Тану технологиялары Мемлекеттік, жеке қызметтердің сапасын арттырады және елдің цифрлық экономикадағы бәсекеге қабілеттілігін арттырады.

Бет бейнелерін талдау негізінде тұлғаны тану мәселесін шешу компьютерлік көру дамуының алғашқы кезеңдерінен басталады. Бетті тану - объектілерді тану теориясының дамуына түрткі болған алғашқы практикалық міндеттердің бірі. Бұл зерттеу бағытының дамуы 1990-жылдары жеке тұлғаны анықтауға арналған идентификациялық және ақпараттық-іздеу жүйелерін құрудан бастау алады [2]. Әртүрлі қызмет салаларында соңғы жылдары бейне ағындар негізінде адамды нақты уақыт режимінде жылдам, дәл және автоматты түрде сәйкестендіруге деген қажеттілік анағұрлым артып келеді [3-6]. Бейне ағындардағы бет кескіндерін сәйкестендіру процесі бірнеше кезеңнен тұрады: 1) кескіндерден бет аймағын анықтау. 2) анықталған бет кескінін бейнелер тізбегінде орналасуын қадағалау. 3) түсірілім жағдайларына байланысты кескінді алдын ала өңдеу. 4) алынған бет бейнесін деректер базасындағы бет кескіндерімен салыстыру. Қазіргі уақытта бет кескіндерін анықтау мәселесі кеңінен зерттелуде. Ол көптеген техникалық жүйелерде қолданылуда [7].

Бейнебақылау жүйелерінде қолдануға арналған жасанды интеллект технологиялары әлемнің көптеген елдерінде қарқынды дамып отыр. Бұл бағытта Huawei, Hikvision, Dahua және ZTE сияқты қытайлық компаниялар алдыңғы орында. Олар жасанды интеллект технологияларын әлемнің көптеген елдеріне жеткізуде. NEC (Жапония), IBM, Palantir және Cisco (АҚШ) сияқты алпауыт компаниялар да бейнебақылау жүйелеріне арналған жасанды интеллект технологияларын белсенді түрде зерттеуде және халықаралық нарықта ұсынуда.

Еуропалық Одақтың бірқатар елдерінде қоғамдық орындарда бет тану технологияларын қолдануға мораторий енгізілген [8].

Бет тану технологияларын қолдануға байланысты құпиялылық мәселелеріне және ықтимал тәуекелдерді азайтуға Еуропалық Одақ елдері баса

назар аударуда. Азаматтардың жеке құқықтарын қорғау және жеке өміріне қол сұғылмауы мұқият қорғалған.

"Әділетті Қазақстанның экономикалық бағдары" атты Қазақстан Республикасының Президенті Қ.К. Тоқаевтың Жолдауында жасанды интеллект, блокчейн, заттар интернеті және үлкен деректер сияқты технологиялар экономиканың барлық салаларына елеулі өзгерістер әкелетіндігін аңғаруға болады. Жолдау интеллектуалды ақпараттық жүйелерді дамыту қажеттілігін айқындап отырғандығы анық екендігі мәлім.

Кескіндерді автоматты түрде тану саласын жасанды интеллект пен компьютерлік көрудің қарқынды дамуы айтарлықтай жетістіктерге жеткендігін көрсетіп отыр. Медициналық диагностика, бейнебақылау, автономды басқару жүйелері, өнеркәсіптік автоматтандыру және цифрлық криминалистика сияқты көптеген салаларда бейне ағындардағы кескіндерді тану маңызды мәселе екендігі анықталып отыр.

Бейне ағындарды нақты уақыт режимінде өңдеудің жоғары есептеу жүктемесі, түсірілім жағдайларының өзгеруі, модельдерді динамикалық жағдайларға бейімдеу қажеттілігі күрделі ғылыми-техникалық мәселе екендігі күмәнсіз. Көптеген қолданылып жүрген алгоритмдер негізінен тек белгілі бір шарттарда ғана тиімділік көрсететіндігі анықталды. Бейне ағындарды өңдеу кезінде жоғары есептеу ресурстарын қажет ететіндігін байқауға болады.

Yolo (You Only Look Once), SSD (Single Shot MultiBox Detector) және Faster R-CNN сияқты заманауи әдістері объектілерді анықтау мен жіктеуде жоғары дәлдік көрсеткенін пайымдауға болады. Дегенмен, оларды бейне ағындарды өңдеуде қолдану алгоритмдер мен математикалық модельдерді оңтайландыруды, жоғары өнімді есептеу ресурстарын қажет етіп отыр.

Жоғарыда айтылған мәселелерден бейне ағындардағы кескіндерді өңдеу ерекшеліктерін ескеретін, жоғары дәлдік пен тиімділікті қамтамасыз ететін нейрондық желілерге негізделген бейне ағындардағы кескіндерді танудың математикалық моделін әзірлеу өзекті ғылыми міндет екендігін атап өту өте маңызды.

Бейне ағындардағы бет бейнелерін тану дәлдігін арттыру, әрбір кадрды өңдеу уақытын қысқарту, нейрондық желі архитектурасын жетілдіру арқылы есептеу шығындарын азайту, бет бейнелерін түсіру жағдайларының өзгерістеріне бейімделу мәселелері зерттеудің мақсатына жету үшін бүгінгі күннің өзекті мәселесі болып қала бермек.

Бұл мәселені шешу компьютерлік көрудің интеллектуалды жүйелерін жетілдіруге болатындығын көрсетті. Олардың жылдамдығы мен тиімділігін арттыруын қамтамасыз етеді. Кең ауқымды практикалық қолдануға мүмкіндік беретіндігін көруге болады.

Ғылыми мәселе. Нақты уақыт режимінде бейне ағындардағы кескіндерді дәл және тұрақты тануды қамтамасыз ете алатын, шу болған кезде тиімді жұмыс істейтін нейрондық желілерді қолдану арқылы алынған математикалық модель әзірленіп, негізделді.

Зерттеу тақырыбының зерттелу деңгейі. Кескіндердегі бет бейнелерін тануды зерттеу үшін көптеген әдістер ұсынылған. Олардың ең танымалдарының бірі 2001 жылы Пол Виола мен Майкл Джонс ұсынған әдісті айта аламыз. Виола-Джонс әдісі [9, 10] нақты уақыт режимінде жоғары ажыратымдылықтағы бейне ағындарда объектілерді айқын анықтауға мүмкіндік беретіндігі анықталды.

Бұл әдіс кейіннен кеңінен таралып, оның әртүрлі модификациялары қазіргі таңда да көптеген бейнеталдау жүйелерінде кеңінен қолданылатындығын байқауға болады. Үлгілерді тану, графикалық модельдеу, сигналдарды өңдеу [11], компьютерлік көру [12,13], сөйлеуді тану [14,15], дауысты тану және бетті тану [16] салаларындағы зерттеулерде терең оқыту әдістері кеңінен қолданысқа ие болып жүр. Терең оқыту көптеген аутентификация және тану жүйелерінің тиімділігін арттыруға мүмкіндік береді. Биометриялық жүйелерде терең оқыту бірегей биометриялық ақпаратты ұсыну үшін қолданылады. Қазақстандық ғалымдар Е. Әмірғалиев, Р. Мұсабаев, А. Дүйсембаев және т. б. [17-19] машиналық оқыту әдістерін пайдалана отырып, кескіндерді талдау бағытында зерттеулер жүргізуде.

Көптеген шетелдік ғалымдар бұл зерттеу саласына айтарлықтай үлес қосты және оны белсенді түрде дамытуда, олардың қатарында U. Nepal [20], N. Tippins [21], W. Boonsuk [22], D. Huang [23], Q. Zhu [24], J. H. Hong [25], A. V. Nefian [26], J. Deng [27] және басқа зерттеушілер бар.

Диссертациялық зерттеудің объектісі – бейне ағындардағы кескіндерді тану процестері.

Диссертациялық зерттеудің пәні - нейрондық желілер негізінде бейне ағындардағы кескіндерді танудың математикалық моделі.

Диссертациялық зерттеудің мақсаты - нейрондық желілер негізінде бейне ағындардағы кескіндерді танудың математикалық моделін әзірлеу және ұсынылған модельдер мен алгоритмдер негізінде бетті тану жүйесін құру.

Қойылған мақсатқа жету үшін диссертациялық зерттеудің келесі міндеттерін шешу қажет:

1. нейрондық желілер негізінде бейне ағындардағы кескінді тану саласындағы отандық және шетелдік ғылыми зерттеулерді шолу жасап, салыстырмалы талдау жүргізу;

2. конволюциялық нейрондық желілер мен Retinex алгоритмдері негізінде бейне ағындардағы шу әсері бар кескіндерді алдын ала өңдеуге және бетті тануға арналған математикалық модель әзірлеу;

3. бейне ағындардағы кескіндерді тану дәлдігін арттыру және кадрларды өңдеу уақытын оңтайландыру мақсатында конволюциялық нейрондық желіге негізделген көпкезеңді модель әзірлеу;

4. деректерді жіктеу дәлдігін арттыруға және өңдеу тиімділігін жоғарлатуға бағытталған конволюциялық нейрондық желі архитектурасын тұрғызу;

5. ұсынылған математикалық модельдер мен алгоритмдер негізінде бейне ағындардағы кескіндерде бетті тану жүйесін әзірлеу.

Зерттеу әдістері. Диссертациялық жұмыста қойылған міндеттерді орындау мақсатында теориялық және қолданбалы зерттеу әдістері жиынтығы қолданылып отыр. Бейне ағындардағы кескіндерді тану процесінің математикалық моделін құру үшін математикалық модельдеу әдістерін қолдану тиімді болды.

Кескіндерді автоматты түрде анықтау және классификациялау үшін конволюциялық нейрондық желілер (CNN) әдісін пайдалану ұсынылды. Зерттеудің әдіснамалық негізін жүйелер теориясы, жүйелік талдау және деректерді статистикалық өңдеу, сондай-ақ есептеу эксперименттерін жүргізу мен алынған нәтижелерді талдау әдістері құрайтындығын көрсетті.

Қорғауға шығарылатын негізгі тұжырымдар:

1. Конволюциялық нейрондық желілер мен Retinex алгоритмдері негізінде бейне ағындардағы шу әсері бар кескіндерді алдын ала өңдеуге және бетті тануға арналған математикалық модель әзірленді.

2. Бейне ағындардағы кескіндерді тану дәлдігін арттыру және кадрларды өңдеу уақытын оңтайландыру мақсатында конволюциялық нейрондық желіге негізделген көпкезеңді модель құрылды.

3. Деректерді жіктеу дәлдігін арттыруға және өңдеу тиімділігін жоғарылатуға бағытталған конволюциялық нейрондық желі архитектурасы жасақталды.

Зерттеудің ғылыми жаңалығы:

1. Терең оқытуға негізделген конволюциялық нейрондық желілер (CNN) мен Retinex алгоритмінің интеграциясы негізінде, үлкен деректер (Big Data) жағдайында жоғары көлемді, жоғары жылдамдықпен келіп түсетін бейне ақпараттарды өңдеу және бетті тануға арналған математикалық моделі құрылды.

Аталған математикалық модель әртекті шу әсеріне ұшыраған бейне ақпараттарды өңдеу және бетті тану үшін пайдалануға болатындығы анықталды.

2. Бейне ағындардағы кескіндерді күрделі жарықтандыру, шу және масштаб өзгерістері жағдайында тану дәлдігін арттыру және кадрларды өңдеу уақытын оңтайландыру мақсатында конволюциялық нейрондық желілерге негізделген көпкезеңді моделі құрылды.

Зерттеу нәтижелерінің теориялық және практикалық маңыздылығы:

Зерттеудің теориялық маңыздылығы бейне ағындардағы кескіндерді танудың математикалық моделін және нейрондық желілер негізінде кескіндерді өңдеу әдістерін дамытуға бағытталғандығын аңғаруға болады.

Практикалық маңыздылығы ұсынылған алгоритмдерді бейнебақылау және интеллектуалды бейне аналитика жүйелерінде қолдану арқылы кескіндерді нақты уақыт режимінде танудың тиімділігін арттыруға болатындығын байқауға болады.

Диссертациялық жұмыстың негізгі нәтижелері Қазақстан Республикасы Ғылым және жоғары білім министрлігінің 2024–2026 жылдарға арналған гранттық қаржыландыруы аясында жүзеге асырылып жатырған АР23486538 «Жасанды интеллект негізінде бейне ағындардағы бейнелерді тану жүйесін

зерттеу және әзірлеу» ғылыми жобасы аясында орындалып жатырғандығын атап өткен маңызды.

Зерттеу нәтижелерін іске асыру бейне ағындардағы бейнелерді автоматты түрде анықтау және тану үшін жасанды интеллект пен терең оқыту әдістеріне негізделген интеллектуалды жүйені әзірлеуге бағытталып отыр. Ұсынылған әдістер мен модельдер бейнеақпаратты өңдеу және талдау процесінің тиімділігін арттыруға, объектілерді дәл анықтауға және ақпараттық жүйелердің жұмыс сапасын жақсартуға мүмкіндік беретіндігін көруге болады.

Әзірленген математикалық модельдер мен алгоритмдер бейне ағындардағы бейнелерді тану міндеттерін шешуде жоғары дәлдік пен тиімділікті қамтамасыз етіп отыр. Оларды түрлі ақпараттық-аналитикалық жүйелерде қолдануға мүмкіндік беретіндігімен тиімділігі айқындалады.

Зерттеу нәтижелерін енгізу ұсынылған тәсілдің практикалық маңыздылығы және оны бейнебақылау, қауіпсіздік жүйелері және интеллектуалды талдау жүйелері сияқты салаларда қолданудың тиімділігі анықталып отыр. Диссертациялық жұмыстың аталған гранттық жоба аясында орындалғаны туралы анықтаманы (Қосымша А)-да көрсетеміз. Бет тану жүйесі «QUARES» ЖШС технологиялық стегіне сәтті енгізіліп қолданылуда, өндіріске ендіру туралы акті (Қосымша Б)-да көрсетіліп отыр.

Диссертациялық зерттеу нәтижелерінің апробациясы. Диссертациялық жұмыстың негізгі нәтижелері С. Сейфуллин атындағы Қазақ агротехникалық зерттеу университетінің «Ақпараттық жүйелер» білім беру бағдарламаларының ғылыми семинарларында баяндалды.

Диссертациялық зерттеудің негізгі нәтижелері келесі халықаралық конференцияларда ұсынылды:

1. Муталова Ж.С., Шаушенова А.Г., Нурпейсова А.А. Алгоритм сверточной нейронной сети, основанный на улучшении функций глубокого обучения. Қазақстан Республикасы Ұлттық инженерлік академиясының академигі, техника ғылымдарының докторы, профессор Ахметов Бақытжан Сражатдинұлының 70 жылдығына арналған «Жаһандық цифрландыру дәуірінде жасанды интеллекттің білім беру үдерісіне интеграциясы: стратегиялар, инновациялар және киберқауіпсіздік мәселелері» атты Халықаралық ғылыми-практикалық конференция материалдары, Алматы қ. 5 сәуір, 2024 жыл, 1148-1153 б [28].

Зерттеу нәтижелерін талқылау және енгізу:

2 мақала Scopus және Web of Science мәліметтер базасына енгізілген журналдарда жарияланды:

1. Mutalova, Z.; Shaushenova, A.; Nurpeisova, A.; et.al. Development of a mathematical model for detecting moving objects in video streams in real-time. IEEE Access. 2024, 10, 136. <https://doi.org/10.1109/ACCESS.2024.3487783> [29].

2. Nurpeisova, A., Shaushenova A., Kuznetsov O., Ispussinov A., Mutalova Z., Kassymova A. Deep Residual Learning for Face Anti-Spoofing: A Mathematical Framework for Optimized Skip Connections. Technologies, 2025, Volume 13, Issue 9, 413; <https://doi.org/10.3390/technologies13090413> [30].

Қазақстан Республикасы Ғылым және жоғары білім министрлігінің Ғылым және жоғары білім саласындағы сапаны қамтамасыз ету комитеті ұсынатын ғылыми басылымдар тізбесіндегі журналдарға 2 мақала:

3. Муталова Ж.С., Шаушенова А.Г., Исакова Г.О., Нурпейсова А.А., Онгарбаева М.Б., Абдыкаликова Г.А. Метод распознавания человека по изображению лица на основе перемещения точки по направляющим. Известия Национальной Академии Наук Республики Казахстан. Физико-математическая серия, ISSN 1991-346X Том 3. Номер 351 (2024). 118-137 <https://doi.org/10.32014/2024.2518-1726.296> [31].

4. Муталова Ж. С., Шаушенова А.Г., Нурпейсова А.А., Айнагулова А. С., Онгарбаева М. Б. Нейрондық желілер негізінде окклюзиялы бет-әлпетті тану әдістерін салыстырмалы талдау. Университет еңбектері. – 2026. №1 (102) 374-380 https://doi.org/10.52209/1609-1825_2026_1_374 [32].

Басқа журналдарда 2 мақала:

5. Муталова Ж.С. Шаушенова А.Г. Нурпейсова А.А. Онгарбаева М.Б. Исследование алгоритма распознавания лиц на основе YOLO. Актуальные вопросы современной экономики. №6, 2025. <https://avse-journal.ru/Editions> [33].

6. Муталова Ж.С. Шаушенова А.Г. Нурпейсова А.А. Онгарбаева М.Б. Сверточная нейронная сеть, основанная на функции распознавания лиц. Актуальные вопросы современной экономики. №6, 2025. <https://avse-journal.ru/Editions> [34].

Патент.

1. Муталова Ж.С., Нурпейсова А.А., Шаушенова А.Г., Испусинов А.М., Жумалиева Л.Д. Нейрондық желі негізінде тұлғаны тану жүйесі, Қазақстан Республикасының патенті. №9870, 2024 (Қазақстан). <https://gosreestr.kazpatent.kz/Utilitymodel/Details?docNumber=409046> [35].

12.01.2026 ж. бекітілген ғылыми-зерттеу жұмыстарының нәтижелерін оқу процесіне енгізу туралы акті (Қосымша Ә) -да көрсетілген.

02.04.2026 ж. бекітілген ғылыми-зерттеу жұмыстарының нәтижелерін өндіріске ендіру туралы акті (Қосымша Б) -да көрсетілген.

Диссертациялық зерттеу барысында бейне деректерді өңдеу жылдамдығын арттыруға және тану дәлдігін жоғарылатуға бағытталған бейне ағындардағы кескіндерді танудың нейрондық желілерге негізделген математикалық моделі әзірленіп, оған патент алынды. Муталова Ж.С., Нурпейсова А.А., Шаушенова А.Г., Испусинов А.М., Жумалиева Л.Д. Нейрондық желі негізінде тұлғаны тану жүйесі, Қазақстан Республикасының патенті. №9870, 2024 (Қазақстан). <https://gosreestr.kazpatent.kz/Utilitymodel/Details?docNumber=409046> (Қосымша В).

Автордың жеке үлесі. Диссертацияда келтірілген барлық ғылыми және практикалық нәтижелерді автор өзі орындады. Тікелей автор нейрондық желілерге негізделген бейне ағындарындағы кескіндерді танудың математикалық моделін жасады.

Диссертациялық зерттеу тақырыбы бойынша жарияланымдар. Диссертациялық жұмыс тақырыбы бойынша 8 ғылыми жұмыс жарияланды,

оның ішінде Scopus дерекқорына кіретін журналда 2 мақала, ҚР ҰӘҚ уәкілетті органы ұсынған басылымдарда 2 мақала, басқа басылымдарда 2 мақала, халықаралық конференциялар еңбектерінде 1 мақала. «Нейрондық желі негізінде тұлғаны тану жүйесі» атты пайдалы модельге 1 патент алынды.

Диссертациялық жұмыстың құрылымы мен көлемі. Диссертациялық жұмыстың құрылымы кіріспеден, 3 бөлімнен тұратын негізгі тараудан, қорытындыдан, 146 атаудан тұратын пайдаланылған әдебиеттер тізімінен тұрады. Жұмыстың жалпы көлемі 148 бет, оның ішінде 56 сурет пен 7 кесте бар.

Бірінші бөлімде бейне ағындардағы кескіндерді танудың негізгі әдістері, соның ішінде дәстүрлі және терең оқыту тәсілдері қарастырылды. Қолданылып жүрген дәстүрлі әдістер есептеуде тиімді болғанымен, күрделі есептеулер жүргізуге бейімделу мүмкіндігінің шектеулі екендігі айқындалып отыр. Нақты уақыт режимінде нейрондық желілер, оның ішінде әсіресе CNN архитектуралары, тану дәлдігін арттырып, сапалы жұмыс істеуге мүмкіндік бере алатындығы анықталып отыр. Қазіргі трансформерлер сияқты жаңа тәсілдердің әлеуеті мен олардың жоғары есептеу ресурстарын талап ететіндігін аңғаруға болады. Жүргізілген талдаулар нәтижесінде әртүрлі әдістерді біріктіретін көпкезеңді модельдерді қолдану перспективалы бағыт екендігін атап айтуға болады.

Екінші бөлімде жан-жақты қарастырылған бейне ағындардағы кескіндерді танудың математикалық моделі маңызды мәселе екендігін мәлімдей аламыз. Зерттеу нәтижесінде конволюциялық нейрондық желілердің жоғары дәлдікті көрсететіндігі анықталды. Кадрларды кезең-кезеңімен өңдеуге негізделген тиімді модельді ұсынып отырмыз. Жарықтандырудың өзгермелі жағдайларында жұмыс істеуді жақсарту үшін Retinex теориясына негізделген әдісті қолдану қажеттілігі ұсынылды. Оның тану сапасын арттыру мүмкіндігі жоғарылайтындығын байқауға болады. Әртүрлі желілер арқылы Faster R-CNN архитектурасын қолдану арқылы модельдердің дәлдік, жылдамдық және ресурстық тиімділік тұрғысынан артықшылықтары мен шектеулері айқындалғандығы анықталып отыр.

Үшінші бөлімде бейне ағындардағы кескіндерді тану дәлдігін арттыру және өңдеу уақытын қысқарту мақсатында конволюциялық нейрондық желілер негізделген көпкезеңді модель әзірленді. Ұсынылған тәсілде алдын-ала өңдеу, бет аймағын анықтау, белгілерді терең анықтап алу және сәйкестендіру кезеңдерін біріктіретін кешенді өңдеу конвейері қарастырылды. Retinex алгоритмі Faster R-CNN, EfficientNetB0 және Self-Attention механизмдерінің үйлесуі кескін сапасын жақсартып, тану дәлдігін арттыруға мүмкіндік берді, FaceNet моделі арқылы эмбеддингтер қалыптастырылды. Жүргізілген зерттеу нәтижесінде Nexiface жүйесінің қауіпсіз және жоғары өнімді архитектурасы ұсынылып, оның практикалық қолдану тиімділігі дәлелденді.

1 Бейне ағындардағы кескіндерді танудың заманауи әдістеріне талдау

1.1 Бейне ағындардағы кескіндерді танудың жалпы тәсілдері

Бетті тану технологиялары теориялық және практикалық потенциалға ие маңызды сала. Бетті тану саласындағы зерттеулер кескінді тануды, бейнені талдауды, компьютерлік көруді, визуализацияны және басқа салаларды дамытуға үлес қосады. Бетті тану технологиялары биометриялық аутентификация, бейнебақылау және қауіпсіздік жүйелері салаларында қолданылуымен байланысты болып келеді.

Кескінді тану жасанды интеллект саласындағы зерттеудің негізгі бағыттарының бірі ретінде өзекті мәселелер болып қала бермек. Кескінді танудың негізгі мақсаты кескіндегі объектілерді, құбылыстарды немесе жағдайларды автоматты түрде тану деп есептеледі. Ол кескіннің ерекше сипаттамаларын талдау арқылы жүзеге асатынын атап өту маңызды.

Күрделі кескіндерді, кескін деректерінің үлкен көлемін өңдеуде әртүрлі қиындықтарға тап болатындығын атап өту қажет. Мысалы, дәстүрлі әдістер кескіннің масштабының өзгеруіне, шу, жарық жағдайларына және басқа факторларға өте сезімтал болатындығы анықталды. Мұндай жағдайлар тану дәлдігінің күрт төмендеуіне әкеліп отыр.

Дәстүрлі әдістер келесі факторларға байланысты тану дәлдігі төмендейтінін байқауға болады: кескіннің шуы, масштабтың өзгеруі, жарық жағдайлары және т.б. Үлкен кескін деректер жиыны үшін дәстүрлі жазбаларды тану әдістерін пайдалану есептеу күрделілігі жоғары және өңдеу талаптарын орындау қиынға соғады. Бетті тану технологиясының ғылыми зерттеулерде үлкен жетістіктерге жетіп келеді.

Цифрлық бейне суреттер қазіргі қоғам өміріндегі сандық бейнелеуге арналған аппараттық және бағдарламалық технологиялардың таралуымен маңызды ақпарат көзіне айналып отыр [36].

Зерттеудегі кескіндегі адам бетінің төмен ажыратымдылығы, жарықтың жылдам өзгеруі және кеңістіктегі орналасуы мәселелеріне байланысты бүгінгі күннің өзекті мәселелер болып қала бермек. Массачусетс технологиялық институты [2], Карнеги Меллон университеті [3,4], Оңтүстік Калифорния университеті [5,6], Мерилэнт университеті [7-9], Кембридж университеті [10-12], Тошиба университеті [13-15], Қытай ғылым академиясының автоматизация институты [16-18] бейне ағындарға негізделген кескінді тану бойынша кең зерттеулер жүргізіп келеді. Бетті тану бойынша отандық және шетелдік әдебиеттер негізінен қозғалмайтын кескіндер негізінде кескінді тану мүмкіндіктеріне арналған. Бейне ағындарға негізделген кескінді тану зерттеулерін терең талдау және дамыту қажет.

Құрылымдалмаған деректердің үлкен көлемімен жұмыс жасауда кескіндерді тез және тиімді тануды ұйымдастыру керек. Адам бетінің жеке ерекшеліктерін ескеріп, табиғи тілде жеке кескін элементтерін автоматты түрде

жасай алатын құрылымдық және жартылай құрылымдалған үлкен деректерді ұсыну қажеттілігін атап өтеміз.

Үлгіде бейнеленген адамның бейнесімен салыстыру, адамның жеке басын тексеру, жеке тұлғаны анықтауды қамтитындығы анықталып отыр. Аутентификация үлгіге ұқсас адамның бейнесін өзінің кескінімен салыстыратын және оны сұрауда көрсететін сәйкестендіруді қамтамасыз етеді. Жеке сәйкестендіру 1: 1 қатынасындағы сұраудағы беттің кескіні барлық сақталған деректер үлгілерімен салыстыра алатындығымен түсіндіріледі.

Бейне ағындары ең құнды ақпарат көздерінің бірі ретінде кеңінен танылатындығы анықталды. Адаммен байланысты қосымшалар (бейне ағындық жүйелер, виртуалды орта, адамның машинамен өзара әрекеттесуі, бейнеконференциялар) кескіндер туралы ақпаратты талдаумен байланысты болады.

Нақты уақыттағы өңдеу талаптары мен өткізу қабілеттілігінің шектеулері кескін сапасы төмен болатындығымен анықталып отыр. Кейбір жағдайларда нақты уақыт режимінде өңдеу үшін 160×20 немесе 352×240 сияқты трег кескіндерді қолданылатындығын айтуға болады.

Бейнеге негізделген бетті тану бірнеше онжылдықтан бастап машиналық көру, компьютерлік көру және биометриялық аутентификация саласындағы ең өзекті ғылыми мәселелердің бірі болып отыр. Қозғалмайтын кескіндермен салыстырғанда бейне кеңістіктік-уақыттық сипаттамаларға ие болуымен түсіндіріледі.

Бірнеше кадрлардан алынған ақпарат пен уақытша деректер үйлесімді болуы қажет екені атап өтуге болады. Ол жүйенің сенімділігі мен тұрақтылығын қамтитындығы және бейнелердегі бетті тану дәлдігін арттыратындығын көрсетіп отыр.

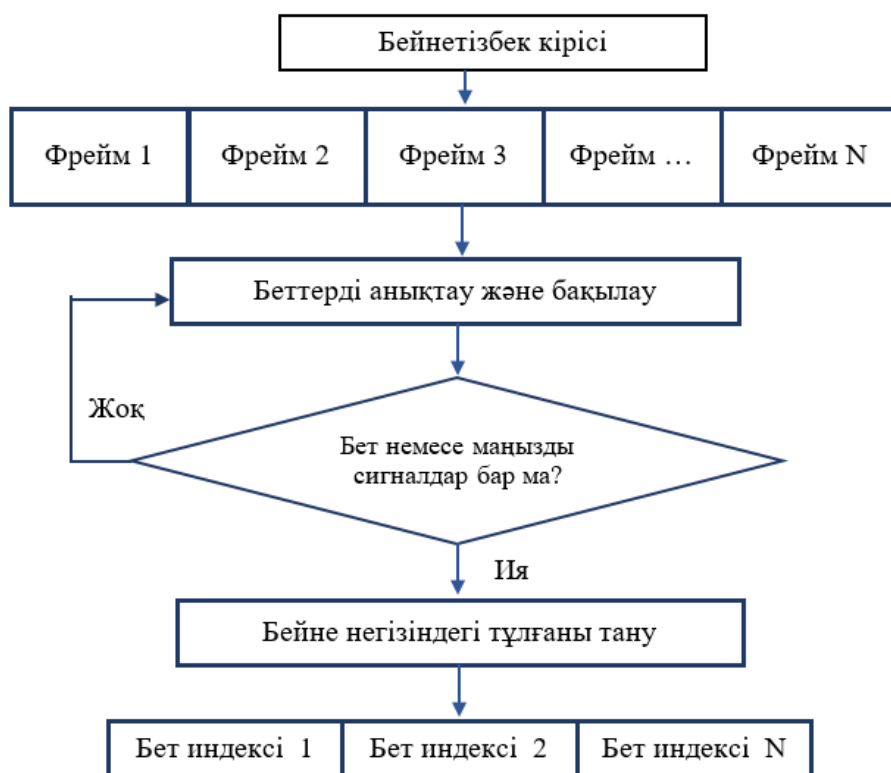
Біріншіден, уақытты пайдалану тану міндетін жеңілдететінін атап өту қажет. Екіншіден, тану нәтижелерін жақсартатын жоғары ажыратымдылықтағы бет кескіндері және 3D бет үлгілері сияқты бейнелерден жоғары сапалы кескіндерді алуға болатындығын ескеру қажет. Үшіншіден, бейнеге негізделген тану әдісі уақыт өте келе объект моделін, содан кейін кадрларды жаңарту қажеттілігін атап өту керек. Оған кескін сапасының төмендігі, бұлыңғыр қозғалыс, масштабтың өзгеруі, бет күйінің өзгеруі, жарықтандырудың өзгеруі, бет бөліктерін жабу сияқты факторлар әсер ететін көрсетеді [54].

Бейне әдістерімен салыстырғанда, осы санаттағы жүйелік деректер мен мәліметтер базасы бейне форматында ұсынылған. Мұндай мәселелерді шешудің қазіргі техникалық деңгейге сәйкес 3 әдісін айтуға болады:

1. кіріс бейне сигналдарынан алынған ерекшеліктер векторға негізделген әдіс;
2. бейнелердегі беттердің сипаттайтын тарату функция және ықтималдықтың тығыздығына негізделген әдіс;
3. кескіндегі беттердің динамикалық таралуын сипаттайтын модельдерді құруға негізделген әдіс.

Дәстүрлі тану алгоритмдері статистикалық кескіндерге негізделеді. Бейнеге негізделген бетті тану-соңғы онжылдықтарда белсенді зерттелген сала екендігін айтуға болады. Бұл әдістерді екі негізгі тәсілге бөлуге болады: 1-ші-деректер жиынтығына негізделген тәсіл бойынша; 2-ші - бейне кадрлар тізбегіне негізделген тәсіл бойынша. Деректер жиынтығына негізделген тәсіл бейнені ретсіз кескіндер жиынтығы ретінде сипаттауға болады. Көптеген бақылауларды қолдануға болатындығын көрсетті. Тану тиімділігін арттыру және күрделі көру жағдайында тұрақтылықты қамтамасыз ету үшін тізбекті тәсілдер уақыттық ақпаратты тікелей қолдануға болатындығын атап өту маңызды.

Бетті тану жүйесінің архитектурасы бетті анықтау, белгілерді шығару, бетті тану модульдерінен тұрады (1.1-сурет).



Сурет 1.1. Бейне тізбектеріндегі бетті тану процесі

Бетті тану жүйесінің бірінші кезеңіне бетті анықтауды жатқызады. Жүйелік модуль кадрларды бейнематериалдардан шығарады. Бет аймағын табу үшін кескінді өңдеу технологиясын қолданады. Жүйе қозғалмайтын кескіндермен жұмыс істеген жағдайда әдіс бет орналасуы деп аталады. Ал бейнемен жұмыс істеу кезінде бетті бақылау әдісі қолданылады. Оның басты мақсаты-бейнеден адамның бетін табу және аймақты бөлектеу.

Әртүрлі параметрлерге бетті анықтау негізделеді. Мысалы, бет құрылымы, бейне кадрдағы беттің қозғалыс, шу, жарықтану жағдайлары, көптеген параметрлердің комбинациясы жағдайларын айта аламыз. Кіріс кескін әртүрлі позициялар мен масштабтарда жылжымалы терезе әдісі арқылы қадағаланатыны белгілі. Бұл тәсілде жылжымалы терезедегі үлгінің “бет” немесе “бет емес” класына жататынын автоматты түрде жіктей алатындығында.

Беттердің нақты орналасуын анықтау үшін визуалды тану әдістеріне ұқсас тәсілдер қолданылатындығын көрсетті. Сол арқылы бет пен оның негізгі белгілерін қадағаланады. Бетті тану технологияларына бас қозғалысын бақылау әдісі де кіріп отыр. Онда бас қозғалыс жасайтын объект ретінде қарастыруға болатындығын көрсетті. Қадағалау модулі алдыңғы кадрларда анықталған бет позициясына немесе бет белгілеріне сүйенетіндігі анықталып отыр.

Бейне негізінде бетті тану процесі бірнеше кезеңнен тұратындығын айта аламыз.

Біріншіден, кадрларға негізделген дәстүрлі әдістер екенін көруге болады [55]. Кадрларға негізделген әдістерді атап шығатын болсақ: статикалық модельдеу әдістері, нейрондық желіге негізделген әдістер, векторлық байланыстыру әдістері (SVM), жасырын Марков модельдері, оңтайландыру әдістері және бетті түсі бойынша анықтау әдістері.

Екіншіден, бетті бірінші кадрда анықтап алғаннан кейін, бүкіл бейнетізбек бойында қадағаланатынын байқаймыз. Алайда анықтау және қадағалау процесстері бір-біріне тәуелсіз орындалатындығын білеміз.

Бетті тану процесінің маңызды кезеңдерінің бірі-белгілерді алу болатындығын көрсетті. Беттің кескіні алынғаннан кейін келесі қадам - бет белгілерін анықтау болып қала бермек. Белгілер екі түрге бөлінетінін айтып кетсек: геометриялық белгілер; сыртқы көрініске негізделген белгілер. Кейбір жағдайларда кескін сапасының нашарлығы, жарықтандырудың өзгеруі немесе басқа факторлар бет белгілерінің дәл сақталуына кедергі келтіретіндігі анықталды. Бет құрылымындағы өзгерістерді сыртқы көрініске негізделген белгілер сипаттауды қамтамасыз етеді.

Бейнені пайдаланудың басты артықшылықтарына бейнедегі статикалық кескінге негізделген тану тиімділігін арттыру жатады. Бейне деректерінде тану нәтижелерін жақсартатын динамикалық ақпаратты құрайды. Ал динамикалық ақпарат жіктеу нәтижелерін жақсартады. Бейне тізбектен жоғары ажыратымдылықтағы кескінді немесе 3D бет үлгісін шығаруға болады. Бейнеге негізделген әдіспен кескін моделін жаңартуға және кейінгі кескіндерді тану сапасын жақсартуға болады.

Бейне негізіндегі бетті тану бірқатар қиындықтарға да ие. Қиындықтарға бейненің төмен сапасы, кескіннің төмен ажыратылымдығы, жарықтандырудың өзгеруі, бет позасының өзгеруі, қозғалыс кезінде пайда болатын бұлыңғырлық, объектілердің бір-бірін жабуы, бет мимикасының өзгеруі немесе камерадан қашықтықтың үлкен болуы сияқты факторлар жатады.

Кадрларға негізделген тану әдістері статикалық кескіндерге сүйенеді. Ал бейнетізбектерге негізделген әдістер динамикалық бейнекадрларды пайдаланады. Бейнетізбекке негізделген тану әдістері бірнеше кадрдағы бейнелерді тануда уақыттық ақпаратты пайдаланады. Мұндай әдістерге жасырын Марков модельдері, рекуррентті нейрондық желілер және ережелерге негізделген классификаторлар жатады.

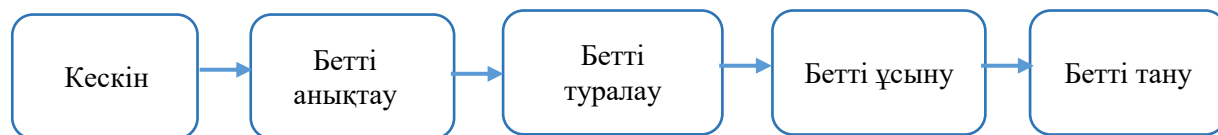
Бетті тану жүйелері компоненттерден тұрады (1.2-сурет):

1. Бетті анықтау. Бетті тану процессіндегі ең маңызды кезеңі - ерекшелік белгілерін шығару. Бет кескіні алынғаннан кейін келесі қадам бет белгілерін анықтау. Белгілер екі негізгі түрге бөлінеді: геометриялық белгілер; сыртқы көрініске негізделген белгілер.

2. Бетті туралау. Бетті туралаудың мақсаты - бет кескіндерін бірдей масштабта және бірдей пішінде келтіру болып анықталады. Кескіннің белгілі бір бекітілген нүктелерінде орналасқан тірек нүктелері қолданылып анықталады.

3. Бетті ұсыну. Бет кескінінің пиксель мәні түрлендіруді қажет ететін объектілер векторының айырмашылығы болады. Бұл векторды шаблон деп атайды. Адамның әртүрлі бет кескіндері ұқсас белгілер векторлары ретінде бейнелену қажет.

4. Бетті тану. Бетті тану кезеңінде екі шаблон салыстырылады. Ұқсастық деңгейі анықталады. Ұқсастық көрсеткіші бір адамға тиесілі болу ықтималдығын айқындайды.



Сурет 1.2. Бетті тану жүйесінің негізгі блоктары

Кескіндерді танудың классикалық әдістері. Кескіндерді танудың классикалық әдістері деп кескіндерді талдау және жіктеу үшін қолданылатын алгоритмдер мен тәсілдердің жиынтығымен түсіндіріледі.

SIFT (Scale-Invariant Feature Transform) танымал классикалық әдістердің бірі ретінде 1999 жылынан бастап сенімді тәсіл болып пайдалануға ұсынылды. SIFT әдісі кескіндегі негізгі нүктелерді әртүрлі масштабтарда анықтайды. Масштабқа тәуелсіз жоғары нәтижелер алатындығы анықталды.

Кескіннің масштабтық кеңістігі $L(x, y, \sigma)$ гаусстық функция $G(x, y, \sigma)$ мен кіріс кескін $I(x, y)$ арасындағы конволюция операциясы арқылы есептелетіндігін көрсетеміз (1.1).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1.1)$$

мұндағы, $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$. Гаусс функциясының орнына Гаусс айырмашылығы (1.2) есептеуді жылдамдату үшін пайдаланылатынғы анықталып отыр:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)h = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1.2)$$

SIFT алгоритмінің негізгі кезеңдері. SIFT әдісі төрт негізгі кезеңдерге бөлінетіндігі анықталды:

Масштабтық кеңістіктегі экстремум нүктелерін анықтау. Гаусс айырмашылығы арқылы кескінді локалды минимумдар мен максимумдарды анықтауға болатындығын көрсетті.

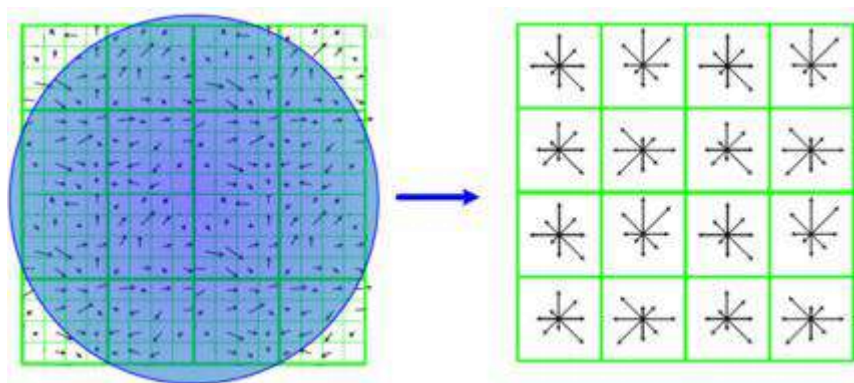
Негізгі нүктелерді локализациялау. Нүктелердің нақты орналасуы Тейлор қатары арқылы аппроксимацияланады және сенімсіз нүктелер алынып тасталатындығы көрсетіледі.

Бағытты анықтау. Градиент бағыты негізінде әрбір негізгі нүктеге ориентация айқындалатындығы көрсетіледі. Әдістің бұрылуға тәуелсіздігін пайдалану іске асырылады.

Дескриптор құру. 16×16 аймақ 4×4 ұяшықтарға бөлініп, әрбір ұяшық үшін градиент бағытының гистограммасы құрастырылады. Нәтижесінде 128 өлшемді дескриптор құруға болатындығымен түсіндіріледі.

Гаусс айырмашылығы көмегімен $3 \times 3 \times 3$ өлшемді көрші аймақтағы локалды экстремумдар ізделіп, негізгі нүктелер ретінде белгілеу ұсынылды. Әрбір негізгі нүктенің бағыты болады. Негізгі нүкте маңындағы аймақ талданады. Градиент бағыттарының ең үлкен мәні бағдар ретінде алынатындығы анықталды. Басқа бағыттардың өзгеру амплитудасы 80 %-дан жоғары болса, онда олар да ескерілетіндігін көрсетті.

Дескрипторды құру үшін негізгі нүкте айналасындағы 16×16 аймақ алынады және 4×4 өлшемдегі ұяшықтарға бөледі. Әр ұяшық үшін бағыт гистограммасы есептеледі. Барлық гистограммалар біріктіріледі. 128 өлшемді дескриптор жасайды (1.3-сурет).



Сурет 1.3. SIFT дескрипторы

SURF (Speeded Up Robust Feature) функциясы-нүктеге жақын масштабқа тәуелді пиксель қарқындылығының таралуын көрсету әдісі. Бұл нүктелер Гессеннің экспресс әдісімен анықталады. SURF әдісі SIFT әдісіне ұқсас болып келеді, бірақ есептеу уақытын қысқарту және тұрақтылықты арттыру үшін кіріктірілген шаштар мен толқындарды қолданады. Кескін бұрылуына инварианттылық орындалуы үшін әрбір нақты нүктелерге белгілі бір бағыт беріледі.

SURF алгоритмінің негізгі кезеңдеріне жатады:

- кескіндер Хаар вейвлеті арқылы сүзіледі;
- негізгі позицияға жақын модельдеу нүктесіндегі сүзгінің жауабы екі өлшемді кеңістіктегі вектор түрінде көрсетіледі;
- барлық векторларды жинақтау үшін 60° –қа өзгертілетін терезе алынады, осылайша негізгі бағыт ең ұзын вектормен анықталады.

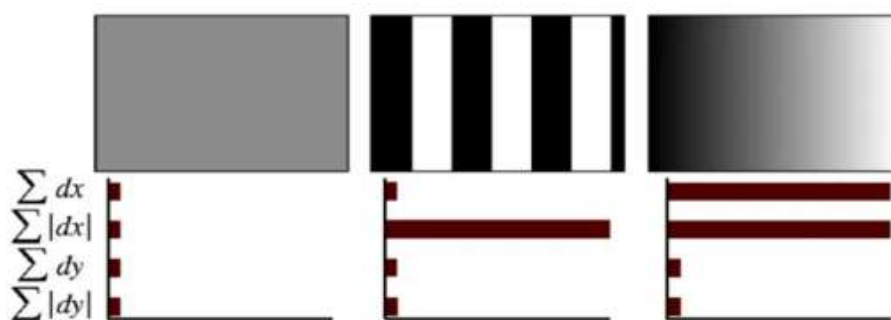
SURF алгоритмі әдетте 2 негізгі кезеңнен тұрады: негізгі нүктелерді анықтау және негізгі нүктелерді түрлендіру. Негізгі нүктелерді анықтау кезінде SURF әдісі Гаусс вариациясының орнына интегралдық кескіндерді пайдаланады [56]. Интегралдық кескін келесі теңдеумен анықталады (1.3):

$$I(X) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (1.3)$$

$I(x)$ - кіріс кескін I -дің тікбұрышты аймағындағы барлық пиксель мәндерінің қосындысына интегралдық кескін сәйкес келеді. Интегралдық кескінді пайдалану арқылы фильтр өлшеміне тәуелсіз есептеу уақытқа қол жеткізуге болады. Гессиан матрицасының анықтауышы арқылы негізгі нүктелер анықталады. Гессиан матрицасы келесі X нүктесі үшін (1.4) орындалады:

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \delta) & L_{xy}(X, \delta) \\ L_{xy}(X, \delta) & L_{yy}(X, \delta) \end{bmatrix} \quad (1.4)$$

SURF дескрипторы таңдалған бағытқа сәйкес квадрат ауданын құру арқылы есептеледі (1.4-сурет). Бағытты анықтау үшін Хаар вейвлеттері x және y осьтері бойынша есептелетіндігі анықталып отыр. Бұл тәсіл ерекшеліктерді шығаруға және анықтауға кететін уақытты қысқартатынын анықтаймыз.



Сурет 1.4. SURF дескрипторын құру

HOG дескрипторлары кескінді ұяшықтарға бөліп, әрбір ұяшық градиент бағытын анықтауға мүмкіндік береді. Ұяшықтағы барлық пикселдер үшін градиент мәндерін қосатындығы анықталды [57]. Градиент бір вектор түріндегі дескрипторларды алу үшін біріктірілетінін атап өтейік. Градиент келесі сүзу шкаласымен есептелетінін көруге болады (1.5):

$$D_x = [-1 \ 0 \ 1] \text{ и } D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (1.5)$$

I кескінің x және y бағытындағы туындылары болып келеді:

$$I_x = I * D_x, \quad I_y = I * D_y.$$

Градиент модулі мен бағыты (1.6):

$$|G| = \sqrt{I_x^2 + I_y^2}, \quad \theta = \arctan \frac{I_x}{I_y} \quad (1.6)$$

SIFT әдісі шу әсеріне тұрақты болып келеді. Көптеген негізгі нүктелерді анықтай алатындығы анықталды. Бірақ оның есептеу жылдамдығы төмен болатындығы белгілі. SURF алгоритмі SIFT-ке қарағанда жылдамырақ жұмыс

жасайтыны анық. HOG әдісі кескіндегі текстуралар туралы ақпаратты сипаттайтынын аңғаруға болады (1.1-кесте).

Кесте 1.1. HOG, SIFT және SURF әдістерін негізгі сипаттамаларын салыстыру

Негізгі сипаттамалар	HOG	SIFT	SURF
Негізгі	Градиенттер және гистограммалар	Негізгі нүктелер + дескрипторлар	Жеделдетілген негізгі нүктелер
Жарыққа инварианттылығы	Орташа	Жоғары	Жоғары
Шу әсеріне тұрақтылық	Төмен	Жоғары	Орташа
Масштабқа инварианттылығы	Жоқ	Иә	Иә
Бұрылуға инварианттылығы	Ішінара	Иә	Иә
Есептеу күрделілігі	Төмен	Жоғары	Орташа
Дескриптор өлшемі	Орташа (36-81)	Үлкен (128)	Орташа (64)
Жылдамдығы	Жоғары	Төмен	Орташа

Сыртқы көрініске негізделген басқа классикалық әдістер де жақсы нәтижелер көрсететіндігі анықталып отыр. Оларға PCA (Principal Component Analysis), ICA (Independent Component Analysis), LDA (Linear Discriminant Analysis), 2DPCA, SDG -сызықты регрессия классификаторлары жататынын атап өту керек. Алайда жарықтандыру жағдайының өзгеруі, бет мимикасы және басқа факторлардың әсерінен бұл әдістер әрқашан жеткілікті нәтиже бере бермейді. Мұның себебі - бет кескіндерінің көпөлшемді кеңістікте күрделі сызықтық емес көптүрлілік (nonlinear manifold) бойында орналасуы.

LBPН (Local Binary Patterns Histogram) бет кескініндегі маңызды белгілерді айқындайтын аралық кескін құруға бағытталған алгоритм [58]. Бұл әдісте кескін әдетте 3x3 өлшемді матрицаға бөлінеді. Мұнда матрица матрица элементтері пиксель тығыздығын білдіретіндігі анықталған (0-255). LBPН алгоритмінің басты артықшылықтары OpenCV ашық кітапханасы және Matlab, C++ және т.б. сияқты көптеген бағдарламалау тілдерінде қолдануға болатындығын көрсетті. Кез-келген алгоритм барлық қосымшалар үшін тиімді емес екені анықталып отыр.

1.2 Бейне ағындардағы кескіндерді өңдеуге арналған нейрондық желілер архитектуралары

Нейрондық желілер белгілі және белгісіз деректерді болжау мен жіктеудің ең тиімді және сенімді әдістерінің бірі болып қала бермек. Нейрондық желілер

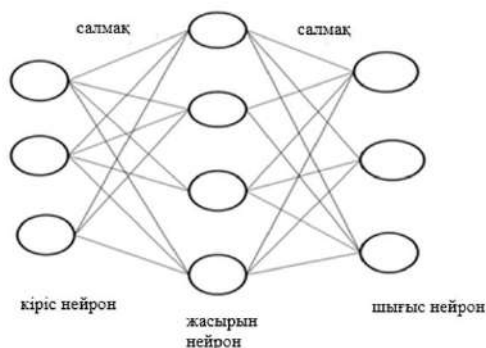
сызықтық және сызықтық емес деректер жиынтығымен жұмыс істегенде жоғары тиімділікті көрсетіп отыр.

Бүгінгі күні нейрондық желілер көптеген салаларда қолданатынын атап өтсек: кескіндерді визуалды түсіндіру, сөйлеуді тану, бетті тану, саусақ ізін тану және көздің қабығын тану және т.б.

Нейрондық желі жасанды нейрондардан немесе түйіндерден тұратын құрылым ретінде түсіну ұсынылған. Түйіндер бір-бірімен байланысқан және олардың арасындағы байланыс салмақпен сипатталатындығын көрсетті.

Эталондық салмақ мәні неғұрлым жоғары болатын болса, түйіндер арасындағы байланыс соғұрлым күшті болатындығын аңғаруға болады. Әрбір түйінде активтендіру немесе тасымалдау функциясы ретінде ұйымдастырылатындығы белгілі болды.

Жасанды нейрондық желі нейрондардың үш негізгі түрінен тұратындығы мәлім: кіріс деңгейі, жасырын деңгей, шығыс деңгейі (1.5-сурет).



Сурет 1.5. Жасанды нейрондық желі құрылымы

Кіріс деңгейі деректерді сандар түрінде қабылдайды. Ақпарат активтендіру мәні арқылы беріледі және әр түйінге белгілі бір сандық мән беріледі. Ақпарат қосылыстың маңыздылығына, басу немесе қозу әсеріне және белсендіру функциясына байланысты желі арқылы беріледі.

Әрбір нейрон өзіне келіп түскен активация мәндерін қосып, оларды активация функциясы арқылы түрлендіреді. Осылайша ақпарат жасырын қабаттар арқылы өтіп, соңында шығыс қабатына жетеді. Шығыс қабаты желінің соңғы нәтижесін көрсетеді.

Конволюциялық нейрондық желілер (Convolutional Neural Networks, CNN) - кескін мазмұнын талдау үшін қолданылатын ең тиімді модельдердің бірі. Олар кескінді тану, сегментация, іздеу және классификациялау міндеттерінде жоғары нәтижелер көрсетеді [59-61].

CNN желілерінің жоғары тиімділігі келесі факторлармен түсіндіріледі:

- желі параметрлерінің ең үлкен саны (ондаған миллион параметрлер);
- оқыту үшін қолданылатын үлкен көлемдегі деректер.

CNN көбіне трансляцияға инвариантты жасанды нейрондық желі деп аталады, себебі ол кескіндегі объектінің орналасу өзгерістеріне тұрақты келді.

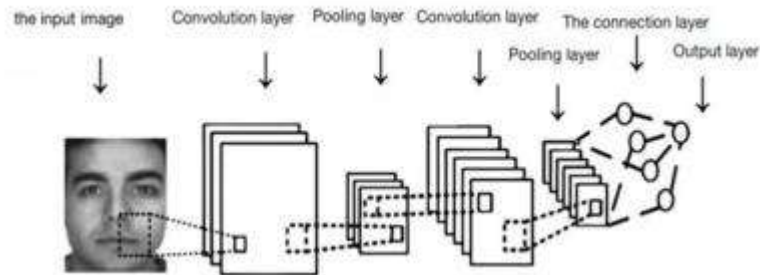
CNN негізін үш негізгі принцип құрайды:

1. Салмақтарды бөлу (weight sharing)

2. Кеңістікті біріктіру (pooling)
3. Локалды қабылдау өрісі (local receptive field)

Локалды қабылдау өрісі -кіріс кескіннің белгілі бір аймағына сәйкес келетін нейрондардың бөлігі. Кескіндерде көрші пиксельдер арасында кеңістік корреляция болатындықтан, нейрондар тек жақын аймақтарды өңдейді.

Конволюциялық нейрондық желінің негізгі архитектурасына кіреді: кіріс қабаты, конволюциялық қабат, біріктіру қабаты, толық байланысқан қабат, шығыс қабаты. 1.6-суретте CNN архитектурасының схемасы бейнеленген.

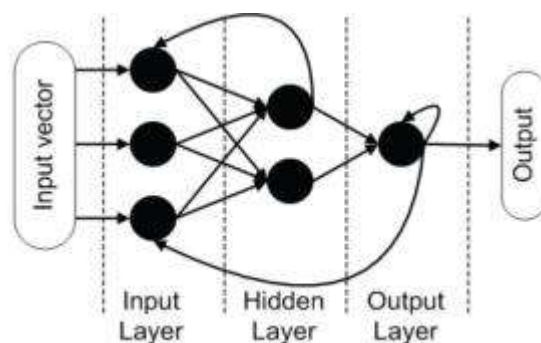


Сурет 1.6. Конволюциялық нейрондық желі архитектурасы

1. Конволюциялық қабат. Конволюциялық қабат нейрондық желінің негізгі элементтерінің бірі болып келеді. Фронтальды қабаттың белгілі бір аймағынан деректерді әрбір нейрон алады. Бұралу ядросының мөлшері әдетте 3x3, 5x5 құрайтынын атап өту керек. Бұл қабат кіріс кескінін кішігірім аймақтарға бөлетін сипаттамаларын анықтайтынымен байланысты. Конволюциялық қабатынан кейін алынған белгілер картаның тереңдігін арттыратынын көруге болады.

2. Біріктіру қабаты. Біріктіру қабаты кескіннің өлшемін азайтатыны атап өту керек. Есептеу күрделілігін төмендететінімен байланысты. Жоғары ажыратымдылықтағы кескінді төмен ажыратымдылықтағы кескінге түрлендіруімен түсіндіріледі. Орнату қабаты параметрлер санын азайтады. Есептеу уақытын қысқартады. Қажет емес ақпаратты жояды.

3. Толық байланысқан қабат. Конволюциялық қабаты мен біріктіру қабатынан кейін толық біріктірілген қабат орналасады. Олар желіні бағалаудың соңғы нәтижесін береді. Конволюциялық қабаты мен біріктіру үлгінің айрықша ерекшеліктерін автоматты түрде шығарады. Толық байланысқан қабаттар оларды түпкілікті шешім қабылдау үшін пайдаланады. 1.7-суретте рекурентті нейрондық желінің қарапайым топологиясы көрсетілген.



Сурет 1.7. Рекуррентті нейрондық желі құрылымы

Құрылым ақпаратты шығысқа кірістен ғана емес, кері бағытта да жібереді. Желілерде ішкі күй сақталады. Уақыт бойынша өзгертін деректерді өңдейді. Рекуррентті нейрондық желінің ең үлкен артықшылығы уақыттық тәуелділіктер өңдейді[62].

GAN (Generative Adversarial Network) екі нейрондық желіден тұрады: генератор және дискриминатор. Генератор жаңа деректер моделін жасайды. Ал дискриминатор оның дәлдігін бағалайды. Екі желі бір-бірімен бәсекелеседі және бірігеді. GAN технологиясында қолданылады: кескіндерді біріктіру, кескін сапасын жақсарту, бейне жасау.

Жасанды нейрондық желілердің бірнеше түрі бар. Оған тікелей байланысты бір қабатты нейрондық желілер, көп қабатты нейрондық желілер және қайталанатын нейрондық желілер жатады. Бетті тану саласында көптеген зерттеушілер PCA негізіндегі нейрондық желілерді [63], терең конволюциялық нейрондық желілерді [64], негізгі функциялары бар нейрондық желілерді [65], каскадты конволюциялық нейрондық желілерді [66], екі сызықты нейрондық желілерді, фондық нейрондық желілерді және көп қабатты перцептрон (MLB) т.б. зерттеп келеді.

Кескінді автоматты түрде тану жүйесі үш негізгі процестен тұрады. Бірінші процесс-бұл деректерді түрлендіру. Яғни бастапқы кірістерді векторлық кескінге түрлендіру. Екінші процесс- үлгіні талдау. Деректер белгілерді таңдау, сипаттамалық белгілерді бөлектеу, мөлшерін азайту, анықтау арқылы өңделетіндігін көрсетті. Үшінші процеске кескіндерді жіктеу, яғни кескіндерді талдау кезінде алынған деректерді пайдалану қажеттілігін ескерген жөн.

Қазіргі уақытта терең оқыту көптеген салаларда жоғары көрсеткіштер көрсетіп келеді. Дәстүрлі әдістермен салыстырғанда терең оқыту дәлдік пен тиімділік деңгейін көрсететіндігін айтуға болады.

Конволюциялық нейрондық желілер (CNN) және кері генеративті желілер (GAN) қуатты нейрондық желілерді құруға мүмкіндік беретіндігімен байланысты. Бұл модельдер объектілерді жіктеу, объектілерді тану, бетті тану және кескін жасау сияқты әртүрлі міндеттерде жоғары нәтижелер беретіндігі анықталды.

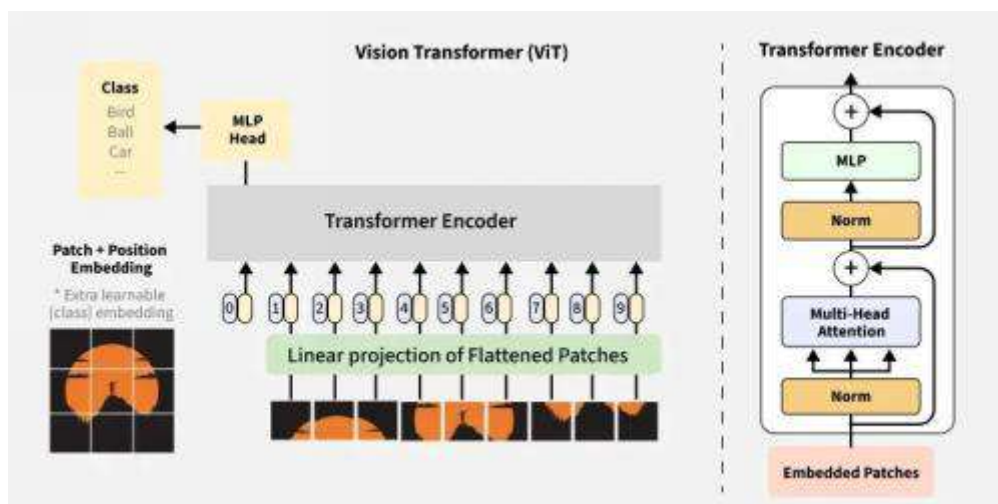
Машиналық оқыту соңғы жылдары өзгерістерге ұшырап отыр. Олардың біріне терең оқыту әдісі жататындығын атап өткен дұрыс. Оның мақсаты - сызықтық және сызықтық емес түрлендірулерден тұратын деректерді жоғары деңгейде модельдеу.

Конволюциялық нейрондық желілер соңғы он жылдықта компьютерлік көру саласындағы басым нейрондық желі архитектурасы айтуға болады. AlexNet [67] моделінің ImageNet кескіндерді жіктеу міндетіндегі серпінді нәтижелер көрсеткенімен белгілі.

CNN архитектурасы бірнеше бағытта дамыды. ResNet [68], RegNet [69] және MobilNet [70] сияқты ортаға бейімделген EfficientNet [71] жеңіл және тиімді желілер пайда болатындығын көрсетті. Мұндай желілер әдетте ImageNet деректер жиынындағы шағын кескіндерде оқытылғандығы көрсетіледі.

Нысандарды тану, толық көріністі талдау, семантикалық сегментация және басқа да арнайы міндеттерді шешуге пайдаланыла алатындығы анықталды. Белгілі бір қолданба үшін толыққанды нейрондық желінің негізі ретінде пайдаланыла алатындығы анықталды. Сол міндеттерге сәйкес деректер жиынына бейімделеді деп қарастырылады.

Компьютерлік көрудегі трансформерлер (Vision Transformers). Vision Transformers (ViT) қазіргі кезде компьютерлік көру саласындағы озық әдістердің бірі деп есептеледі. Конволюциялық нейрондық желілерге балама ретінде пайда болған архитектура ретінде пайдалану ұсынылды. Трансформер - кіріс деректерінің әртүрлі бөліктері арасындағы тәуелділікті ішкі назар механизмі арқылы модельдеуге болатындығында. Vision Transformer бұл архитектураны кескіндерге қолданады, яғни кескінді пиксельдер торы ретінде емес, бірнеше фрагменттердің тізбегі ретінде қарастыру ұсынылды (1.8-сурет).



Сурет 1.8. Vision Transformer (ViT) архитектурасы

Екі кіріс таңбасы арасындағы өзара байланысты бақылау үшін трансформерлер қолданылатынын көреміз. Олар үшін келесі қадамдар орындалады:

1. Толық кескінді шағын фрагменттер торына бөлу бойынша;
2. Әрбір фрагментті сызықтық проекция арқылы ендіру бойынша;
3. Ендірілген әрбір фрагмент токенге айналып, алынған токендер тізбегі трансформерлердің кодтаушы блогына (TE) берілуі бойынша;

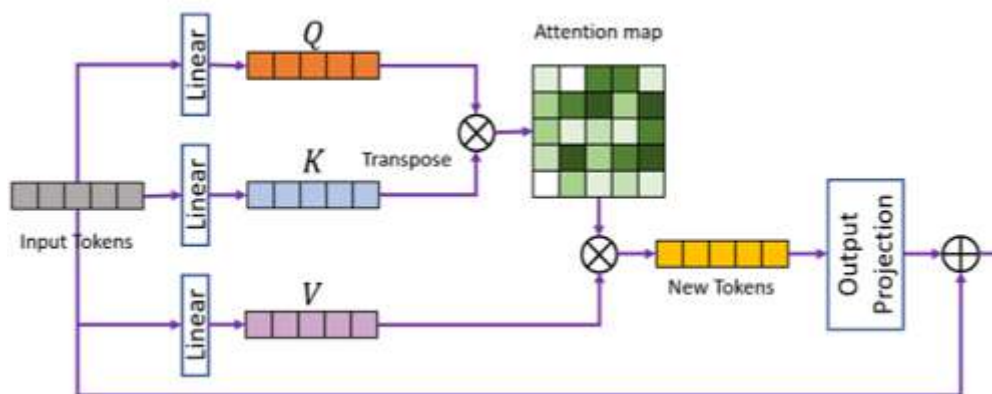
4. Трансформер кіріс токендерді кодтайды, содан кейін нәтиже көпқабатты перцептрон басына (MLP head) беріледі, ал оның шығысы кескіннің класын анықтайды.

Досовецкий ұсынған Vision Transformer [72] желісі 16x16 өлшемді бөліктерге бөлу ретінде пайдалануды ұсынды. Стандартты трансформерге енгізуді, қалай жүзеге асырылатыны көрсетті. Үлгілер иілу әдісін қолданғанда нейрондық желілердің тиімдірек өңделетіндігі анықталып отыр.

Әрбір сегмент сызықтық проекция арқылы векторлық кеңістікке айналады. Осы кірістірілген бөліктерден таңбалауыштар жасалады. Түрлендіргішке кіріс ретінде берілетіндігі анықталып отыр.

Фрагменттердегі ақпаратты кодтау көпбасымды назар және нормализация функциялары қолданылады. Нәтижесінде алынған шығыс MLP head блогына жіберіледі, ал оның нәтижесі кескіннің тиісті класы болып табылады. Кескіндерді жіктеуде ең кең таралған архитектуралардың бірі - бірнеше кіріс токендерін өңдейтін Vision Transformer.

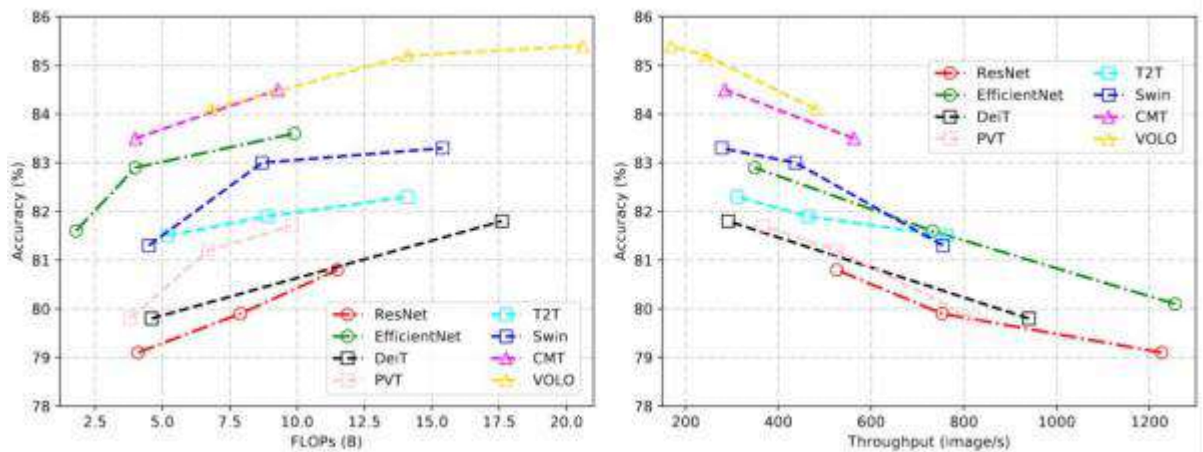
Мұнда K , Q және V - бірдей кіріс белгілер картасының сызықтық проекциялары. Attention maps - QK^T матрицалық көбейтіндісіне softmax функциясын қолдану арқылы алынады [73]. ViT табиғи тілді өңдеуге арналған трансформерден алынған көпбасымды ішкі назар механизмін қолданады. Self-attention механизмі кіріс деректердің салмақталған қосындысын есептейді, мұндағы салмақтар кіріс белгілерінің ұқсастығына қарай анықталады (1.9-сурет).



Сурет 1.9 – Self-attention модулі

Бұл модельге маңызды белгілерге көбірек назар аударуға мүмкіндік береді және деректердің ақпараттық ұсынуын жақсартады. Ішкі назар қабатының екі негізгі ерекшелігі бар: контекстке байланысты белгілердің маңызын динамикалық түрде қайта бағалай отырып, назарды икемді бағыттап алады; ғаламдық ішкі назар қолданылған жағдайда толық қабылдау өрісіне ие болады. Ғаламдық ішкі назар барлық кіріс токендеріне қолданылғанда жүзеге асады. Бұл жағдайда кескіннің әртүрлі кеңістік фрагменттеріне сәйкес келетін барлық токендер бір-бірімен байланысып, толық қабылдау өрісін құрайды.

Vision Transformer кескінді тұрақты өлшемдегі фрагменттерге бөліп, әрқайсын ендіріп, позициялық ендіруді трансформер кодтаушы кіріс ретінде береді. Кейбір жағдайларда ViT модельдері есептеу тиімділігі мен дәлдік бойынша CNN архитектураларынан жоғары нәтиже көрсете алады (1.10-сурет).



Сурет 1.10. CNN мен Vision Transformer модельдерінің FLOP пен өткізу қабілеті бойынша салыстырылуы

ViT-тің ішкі назар қабаты бүкіл кескін бойынша ақпаратты жаһандық деңгейде біріктіруге мүмкіндік береді. Сонымен қатар модель оқу барысында кескін фрагменттерінің салыстырмалы орналасуын да кодтауды үйрететіндігімен байланысты. Бұл кескін құрылымын қалпына келтіруге көмектесетіндігін айтуға болады.

Қазіргі уақытта Vision Transformers моделі кеңінен таралып жатқанын атап өту маңызды. Үлкен деректер жиынтығында әлсіз индуктивті ығысуы, кең қабылдау өрісі арқылы модельдеу қабілеті CNN-ге қарағанда жоғары дәлдікке жете алатындығын көрсетті. PiT және SWIN сияқты заманауи шағын ViT модельдері белгілі бір дәрежеде CNN қағидаттарына жақындап, қабылдау өрісін шектейтіндігін атап өту маңызды. Иерархиялық белгілер пирамидаларын пайдаланады. Қосымша деректерсіз немесе білімді көшірусіз оқытылған жағдайда модель ViT-ден кем түспейтінін көрсетті.

1.3 Нақты уақыт режимінде бетті тану модельдері мен әдістеріне шолу

Кескіндердегі беттерді табу және тану - тану процесінің алғашқы қадамы болып анықталады. Тиімді және дәл анықтау әдістері бетті тану жүйесінің жұмысын жақсарта алатындығымен белгілі. Тиімсіз алгоритмдер жалпы өнімділікке теріс әсер ететіндігін көрсетті.

Ғылым мен техниканың қарқынды дамыған дәуірінде кескінді алу ыңғайлы болып келеді. Дегенмен, мұндай құрылғылар есептеу ресурстарына жоғары талаптар қоятындығын көрсетті. Жоғары жылдамдықты және тиімді есептеу алгоритмдерін әзірлеу маңызды міндеттердің бірі болып келеді.

Бетті тану саласындағы алғашқы зерттеулер кескіндерден қолмен анықталатын белгілерді шығару әдістеріне негізделетіндігін атап өту керек. Мұндай әдістерге бағытталған градиенттер гистограммасы жатқыза аламыз. Уақыт өте келе терең оқыту негізделген объектілерді анықтау әдістері бетті анықтау және тану міндеттерінде кеңінен қолданылатындығы белгілі. Бақылау аймағында орналасқан камералар кескіндерді тіркеп, бейнекадрлар арасындағы өзгерістерді талдау арқылы объектілерді анықтауға септігін тигізіп келеді.

Конволюциялық нейрондық желіге негізделген Yolo алгоритмі кескінді тек желі құрылымдарын бөлуге болатындығын көрсетті. Әрбір тор ұяшығы болжау үшін шектеу қорабы болатындығын да көрсетті.

Әрбір кадр және кеңістіктік объектілердің белгілі бір класына тиесілі екенін атап өту маңызды. Yolo нысандарды танудың бір деңгейлі архитектурасы процесті жылдам жасайтындығы анықталды. Бұл нақты уақыттағы жылдам және дәл шешімдер қабылдауды талап ететіндігін көрсетіп отыр.

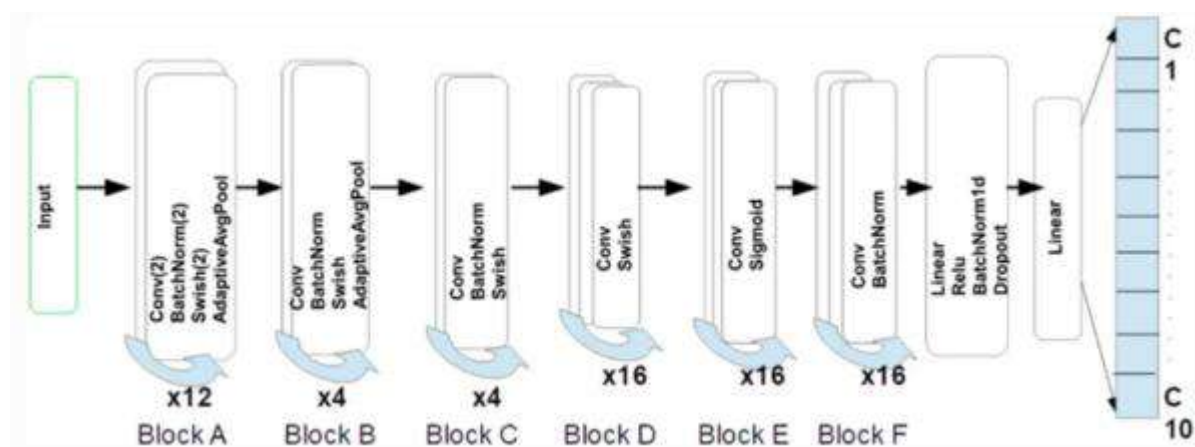
Нысанды тану дәлдігін тиімді, дәл табуда маңызды рөл атқаратындығын R-CNN (аймақтарға негізделген конволюциялық нейрондық желілер) [74] және оның жетілдірілген нұсқалары Fast R-CNN [75] және Faster R-CNN [76] әдістерінен көруге болатындығы анықталды.

Бұл әдістер екі кезең негізінде жұмыс істейтіндігін көреміз: алдымен потенциалды аймақтарды таңдалады; екінші кезеңде конволюциялық нейрондық желі осы аймақтарды жіктейтіндігіне, нақтылайтындығына басты назар аударылатындығын көрсетті.

Басты бағдарды жылдамдық пен тиімділікке бағыттауының әдіс ретінде SSD (Single-Shot MultiBox Detector) әдісін пайдалану ұсынылды.

Нысандарды кескін нүктесі деңгейінде сегменттеуге болатындығын Mask R-CNN [78] әдісінде де аңғарамыз. Mask R-CNN әдісінде объектілердің нақты орналасқан жерін анықтап, жоғары нәтиже қол жеткізе алады. Объектілердің контурларын аздап сегменттеу арқылы жүзеге асыруға мүмкіндік береді.

RetinaNet [79] және EfficientNet [80] сияқты заманауи модельдер де нысанды тану алгоритмдерінің жоғары нәтиже көрсететіні анықталып отыр. Жылдамдық, дәлдік және есептеу күрделілігі арасындағы белгілі бір тепе-теңдікті сақтайтынын ескеру қажет. Модельдер әртүрлі қосымшалармен және компьютерлік ресурстардың шектеулерімен байланысты тиімді жұмыс жасайтындығы анық. Тордың үлкен биіктіктері арнайы біріктірілген модульдері арқылы жойылатындығымен байланысты. EfficientNet желісінің архитектурасы алты негізгі блоктан тұрады (сурет.1.11).



Сурет 1.11. EfficientNet желісінің архитектурасы

EfficientNet желілік архитектурасы компоненттік масштабтау әдісін қолданады. Конволюциялық нейрондық желілерді тиімді масштабтауды

жасайды. Бұл әдіс тордың үш негізгі өлшемін бірден өлшейді: тереңдігі (қабаттар саны), ені (жазбалар саны) және кіріс кескінінің ажыратымдылығы (кескін өлшемі). ConvNets құрылымын кеңейту тепе-теңдікті сақтайды. Тиімді желілік архитектурадағы әрбір блок қалыпқа келтіру қабатынан басталады. Swish белсендіру функциясы іске қосылады. Топтау қабаты және конволюциялық қабаттары жұмыс жасайды.

Терең нейрондық желілер арқылы бет кескіндерінен белгілерді алу және жіктеу процесі әлдеқайда тиімді.

Болашақта терең оқыту технологияларының қарқынды дамуына байланысты мұндай алгоритмдер әртүрлі қолданбалы салаларда кеңінен қолданып, әрі қарай жетілдіріле түседі (1.2-кесте).

Кесте 1.2. Әртүрлі әдістердің артықшылықтары мен шектеулерін салыстыру

Авторлар	Әдіс	Артықшылық- тары	Кемшіліктері	Дәлдігі
Almabdy et al, 2019 [81]	AlexNet+ SVM, RestNet-50+SVM, Transfer Learning AlexNet	Жоғары дәлдік, трансферлік оқытуды тиімді қолдану	Қайта оқыту қаупі, жоғары есептеу ресурстары қажет	94-100%
Sungandi et al, 2019 [82]	Haar, PCA, BPNN	Сенімді, тиімді және нақты уақыт режимінде істейді	Жарықтың өзгеруі мен жылдам қозғалысқа сезімтал	Орташа тану жылдамдығы - 87 %
Tao et al, 2019 [83]	CNN, Haar Cascade, кескінді оңтайландыру	Нақты уақыт режимінде істейді, тиімді, жадыны үнемдейді	Қозғалыс пен жарыққа сезімтал	97,63 %
Zhenzhou &Ding, 2019 [84]	Enhanced RasNet, ArcFace Loss	Жоғары ажыратымдылық және тереңдік мәселелерін шешеді	Бір кезеңде күрделі оқыту процедурасы	LFW -97.7%
Yadav et al, 2019 [85]	CNN, ReLU, Max Pooling	Жоғары дәлдік, нақты уақыт режимінде жылдам өңдеу, жадыны тиімді пайдалану	Масштабтау мен икемділікке талап жоғары, жарық пен қозғалысқа сезімтал	97,63 %

Rajput & Arya, 2017 [86]	CNN классификациясы	Шектеулі дәлдік жағдайында тиімді	Есептеу күрделілігі жоғары болуы мүмкін	AUC-97.56% жоғары PSNR және SSIM
Lee et al, 2020 [87]	LBP-AdaBoost, Гаусс филтрлері	Жарық жағдайында тұрақты, қолдануы қарапайым	Өлшеу кезінде қиындықтар болуы мүмкін	96,27%
Borra et al, 2020 [88]	CNN (AlexNet)	Жоғары дәлдік, күрделілікті тиімді басқару	MATLAB ортасында оқыту жылдамдығы шектеулі	100%
Xu et al, 2020 [89]	11 конволюциялық қабат, 4 pooling кабаты бар CNN	Жоғары дәлдік және тиімділік	Оқыту күрделі және ресурсты көп талап етеді	$\geq 97\%$
Ning et al, 2021 [90]	Improved CNN, Raspberry Pi	Қолжетімді және портативті	Деректерді өңдеу мүмкіндігі шектеулі	>85 %
Amjoud & Amrouch, 2023 [91]	GoogLeNet	Параметрлер аз, есептеу күрделілігі төмен	Оқыту ұзақ, архитектура күрделі	Accuracy - 98 %
Peiyuan Jiang et al, 2022[92]	YOLO, YOLOv1, R-CNN	Жоғары дәлдік, ViT және Swin Transformer-ден тиімді	Белгілі қолданбаларға тәуелді	Accuracy - 98,67%
Zhou Z. et al, 2018 [93]	ResNet, DenseNet, EfficientNetNet, VGG	Жоғары жылдамдық, Inception негізіндегі архитектура	Белгілі қолданбаларға тәуелді	Accuracy- 98,67 %

Дәстүрлі классификация және тану әдістері негізінде машиналық оқыту алгоритмдеріне, мысалы SVM (Support Vector Machine) және KNN (K-Nearest Neighbors) әдістеріне негізделген. Ал терең терең оқытуға негізделген классификация және тану әдістері алынған белгілер векторларын терең нейрондық желі моделін құру арқылы жіктеуді жүзеге асырады. Мұндай модельдерге толық байланысқан нейрондық желі (DNN) немесе softmax классификаторлары жатады. Терең нейрондық желілерді пайдалану арқылы классификация нәтижелерінің дәлдігін айтарлықтай арттыруға болады.

Inception архитектурасына негізделген желі 2014 жылы Google компаниясымен ұсынылды. Бұл желі арнайы Inception модульдері арқылы кіріс

деректеріне сәйкес оңтайлы фильтрлерді таңдайтынын көрсетті. GoogLeNet архитектурасы тығыз Inception блогынан тұратындығы мәлім. Бұл блоктар үш топқа біріктірілген және олардың арасында max-pooling қабаттары орналасқандығы анықталды. Жалпы алғанда желі 22 қабаттан тұратындығы белгілі.

Аталған модульдер әртүрлі өлшемдегі белгілерді бөліп алып, біріктіреді және кейінгі қабаттарға жіберетіндігі белгілі. global average pooling қабатына ерекшелік белгілер жеткізілетіні анықталды. Inception-Res-Net архитектурасы ResNet-тегі байланыстарды пайдаланатын конволюциялық нейрондық желі екендігі мәлім. Мұндай байланыстар градиенттің таралуын жақсартып, терең желілерді тиімді оқытуға ықпал ететіндігімен түсіндіріледі.

Inception-ResNet модульдері әртүрлі масштабта алынған белгілерді біріктіріп, кейінгі қабаттарға жеткізуге жол ашатындығын ескереміз. Бұл модель ImageNet деректер жиынтығындағы бір миллионнан астам кескін арқылы оқытылған және шамамен 164 қабаттан тұратындығын ескерген жөн.

Терең оқыту әдістері, конволюциялық нейрондық желілер бетті тану міндеттерінде жақсы нәтиже беретіндігі анықталып отыр. Терең оқыту әдістері әзірлемелер бетті танудың дәлдігі мен тиімділігін арттыратындығы анықталды.

Қауіпсіздік, биометриялық аутентификация және басқа да көптеген қолданбаларда бетті тану технологияларын кеңінен қолданылып келетіндігін көрсетті.

Бірінші бөлім бойынша қорытынды

Бұл бөлімде бейне ағындарындағы кескіндерді танудың негізгі тәсілдері қарастырылды. Дәстүрлі әдістердің жарық, шу және масштаб өзгерістері жағдайында тану дәлдігін төмендететіні анықталды. Бейне ағындарының кеңістіктік-уақыттық ақпаратты қамтуы бірнеше кадр арасындағы байланысты ескеруді талап ететіні көрсетілді. Сонымен қатар бейне негізіндегі әдістердің уақыттық ақпаратты пайдалану арқылы тану сапасын арттыратыны анықталды. Сондықтан бейне ағындарындағы тұлғаларды сенімді тану үшін жоғары дәлдік пен тұрақтылықты қамтамасыз ететін интеллектуалды әдістерді қолдану қажеттілігі негізделді.

Кескіндерді өңдеуге арналған нейрондық желілер архитектураларының негізгі түрлері мен олардың жұмыс принциптері талданды. Конволюциялық нейрондық желілердің (CNN) кескіндерді тану, классификациялау және объектілерді анықтау міндеттерінде ең тиімді модельдердің бірі екені анықталды. CNN архитектуралары салмақтарды бөлу, pooling және локалды қабылдау өрісі сияқты механизмдер арқылы маңызды белгілерді автоматты түрде бөліп алуға мүмкіндік береді. Салыстырмалы талдау нәтижелері бойынша CNN негізіндегі модельдер жоғары дәлдік көрсеткен: AlexNet + Transfer Learning — 94–100%, CNN + Haar Cascade — 97.63%, Enhanced ResNet + ArcFace Loss — 97.7%, GoogLeNet — 98%, YOLO және R-CNN модельдері — 98.67%. Сонымен қатар EfficientNet архитектурасы дәлдік пен есептеу күрделілігі арасындағы

тиімді тепе-теңдікті қамтамасыз ететіні анықталды. Vision Transformer және Self-Attention механизмдері де жоғары нәтижелер көрсеткенімен, олардың есептеу ресурстарына қойылатын талаптары жоғары екені байқалды. Сондықтан бейне ағындарындағы кескіндерді тану міндеттерінде CNN архитектуралары негізгі және тиімді шешім ретінде таңдалды.

2 Нейрондық желілер негізінде бейне ағындардағы кескінді танудың математикалық моделін әзірлеу

2.1 Бейне ағындардағы кескінді тану әдістері мен математикалық модельдері

Бетті тану технологиясы бет ерекшеліктерін алуға және оларды белгілі бір үлгілермен салыстыру үшін қолданылады. Бетті тану жүйесі беттердің кескіндерінен алынған ерекшеліктер деректерін дерекқорда сақталған ерекшеліктер үлгілерінен іздейді және салыстырады. Осы процесс екі категорияға бөлінеді: тұлғаны растау-верификация (1:1) және тұлғаны анықтау-идентификация (1:N) [94].

Бетті тану технологиясының әдістері үш топқа бөлінеді:

1. Классикалық әдістер бет кескіндерінің статистикалық, геометриялық қасиеттері бойынша бет белгілерін қолмен бөліп алып, салыстыру арқылы жұмыс істейді. Оның түрлері: негізгі компоненттерді талдау, сызықтық дискриминантты талдау, кескін белгілерін талдау [95].

2. Нейрондық әдістер деректерді өңдеу, үйрену және болжау үшін қолданылатын модельдер мен алгоритмдер жиынтығынан тұрады. Ол келесі әдістерді қолданады: машиналық оқыту, жасанды нейрондық желілер, тірек векторлар [96].

3. Бетті танудың гибридті әдістері — бұл тану дәлдігін күшейту үшін бірнеше технологияны интеграциялайтын жүйелер [97].

Сызықтық түрлендіру әдістері деректерді өңдеу кезінде өлшемді азайтуға қолданылады. Кең таралған әдістерге негізгі компоненттерді талдау және сызықтық дискриминанттық талдау жатады. Негізгі компоненттерді талдау ақпаратқа мән бермейтін, объекті класы туралы өлшемді азайтудың бақыланбайтын әдісі. Деректерді төмен өлшемді кеңістікке көшіреді. Деректердің максималды дисперсиясы сақталады, қайта құру қатесі азаяды [98]. Сызықтық дискриминанттық талдау әдісінің идеясы: Фишер мақсатты функциясы деректерді жаңа кеңістікке орналастыру арқылы әртүрлі кластарды бір-бірінен алшақтату; кластың ішіндегі нүктелерді бір-біріне жақындату; деректерді бір-бірінен жақсы ажырату [99]. Матрицалық деректерге екіөлшемді негізгі компоненттерді талдау, екіөлшемді сызықтық дискриминанттық талдау әдістері қолданылады. Мұнда деректер векторларға түрлендірусіз тікелей өңделеді. Есептеулерді жеңілдетеді, әрі өлшем мәселесінің әсерін азайтады [100].

Жасырын Марков моделі (НММ) ішкі күйі тікелей бақыланбайтын жүйелерді сипаттайтын; жүйенің нақты ішінде не болып жатқанын көрмей, тек оның нәтижесін ғана (бақыланатын деректерді) көруге мүмкіндік беретін статистикалық модель. Біз тек сыртқы белгілерді байқап, ал сол белгілерді тудырған жасырын процесті модель арқылы болжаймыз [101].

Терең оқыту әдістері кеңінен қолданылса да, жасырын Марков моделі әлі де ықтималдық теориясы мен жасанды интеллекттің негізгі модельдерінің бірі ретінде саналады. Марков моделі деректердің уақыт бойынша ретін ескереді. Оларды талдауға мүмкіндік береді. Модельдің негізгі қағидасы -болашақ күй тек

ағымдағы күйге ғана тәуелді, ал бұрынғы күйлерге тәуелді емес, мұны Марков қасиеті деп те атайды. НММ екі деңгейден тұрады: жасырын күйлер біз көре алмайтын ішкі жағдайлар, бақыланатын деректер біз тікелей көретін нәтижелер. Модель жасырын күйлердің ауысуы мен бақылауларды генерациялау ықтималдықтары арқылы құрылады.

Жасырын Марков моделі — ықтималдық модель. Жасырын күйлердің бастапқы ықтималдығын, олардың бір-біріне ауысуын және бақылауларды тудыруын сипаттайды. А күйінің ауысу ықтималдығы матрицасы және В бақылауының ықтималдық матрицасы анықталады. Жасырын Марков моделін таңбалар жиынтығымен сипаттауға болады: $\lambda=(A,B,\pi)$.

Жасырын Марков моделі негізінен үш мәселені шешеді:

1. Ықтималдық есептеу мәселесі. Берілген модель $\lambda=(A,B,\pi)$ және $O=o_1, o_2, \dots, o_T$ бақылаулар тізбегі үшін осы тізбектің пайда болу ықтималдығын есептеу керек. Яғни $P(O|\lambda)$ мәнін анықтау қажет. λ моделінің бақыланған O тізбегін тудыру ықтималдығын анықтайды.

2. Оқыту мәселесі. $O=o_1, o_2, \dots, o_T$ бақылаулар тізбегі берілген кезде, $\lambda=(A,B,\pi)$ модель параметрлерін анықтау қажет. Модель параметрлері бақылауларға ең жақсы сәйкес келетіндей таңдалады: $\max_{\lambda} P(O|\lambda)$.

3. Болжау мәселесі. $\lambda=(A,B,\pi)$ моделі және $O=o_1, o_2, \dots, o_T$ бақылаулар тізбегі берілген. Осы бақылауларға сәйкес $I = (i_1, i_2, \dots, i_n)$ ең ықтимал жасырын күйлер тізбегін табу қажет. Осы үш мәселенің шешімі жасырын Марков моделінің үш негізгі алгоритмімен шешіледі [102].

Нейрондық желілер қабаттарда орналасқан, өзара байланысты түйіндерден тұрады. Түйіндер нейрондар деп аталады. Әрбір нейрон басқа нейрондармен ақпарат алмасып отырады. Оларды өңдейді, нәтижелерін басқа нейрондарға жібереді. Нейрондар арасындағы байланыстар белгілі бір салмаққа ие болады. Ол осы байланыстардың күшін көрсетеді. Нейрондық желі оқыту процесінде осы мәндерді реттейді, белгілі бір тапсырмалар бойынша жұмысының тиімділігін арттыру үшін қолданылады. Мұндай оқыту механизмі арқылы нейрондық желілер үлгілерді болжайды және таниды.

Конволюциялық нейрондық желілер - жасанды интеллектіні зерттеудің озық технологиясы. Ол сөйлеуді тануда, компьютерлік көруде және т.б. салаларда қолданылады. Конволюциялық нейрондық желілер күрделі сызықтық емес құрылымдарды зерттеуде, күрделі функциялардың өрнектерін талдауда пайдаланылады. Терең оқыту әдістері төмен деңгейлі функцияларды абстракциялай отырып біріктіреді. Жоғары деңгейлі ақпаратқа айналдырып шешім қабылдайды.

Зерттеуде беттерді тану мәселесін шешу үшін конволюциялық нейрондық желі (CNN) қарастырылды. Конволюциялық нейрондық желілер кескіндерден ерекшеліктерді бөліп шығаруда өз тиімділігін көрсетті. Конволюциялық нейрондық желі әдісі зерттеудің ыңғайлы құралы ретінде келесі бөлімдерде бет тану үшін қолданылады.

2.1-кестеде әдеби деректерді талдай отырып, бет тану әдістерінің тану дәлдіктері салыстырылды.

Кесте 2.1. Бетті тану әдістерін салыстырмалы талдау

Бетті тану әдістері	Авторлар	Кескіндер саны	Тану дәлдігі (%)
Негізгі компоненттерді талдау	Tomasi C., Kanade T., 1991 [7]	400	79.65%
Толқынды түрлендіру	Viola, P.; Jones, M.J. 2001 [10]	100	80–90%
Белсенді форма моделі	Tim Cootes) пен Кристофер Тейлор (Christopher Taylor) [103]	100	78.12-91%
Жасырын Марков моделі	Xue, J.; Han, J.; Zheng, T.; Gao, X.; Guo, J. A , 2012 [12]	200	84%
Меншікті беттер әдісі (Eigenfaces)	Мэттью Тёрк (Matthew Turk) пен Алекс Пентланд (Alex Pentland) [104]	2500	92–95%
Тірек векторларын анықтау әдістері (Support Vector Machines - SVM)	Владимир Вапник (Vladimir Vapnik) пен Коринна Кортес (Corinna Cortes) [105]	400	85–92%
Конволюциялық нейрондық желілер CNN	Rajput & Arya, 2017 [86]	200	97.56%
GoogleNet	Amjoud & Amrouch, 2023 [91]	4500	98 %
YOLO, YOLOv1, R-CNN	Peiyuan Jiang et al, 2022 [92]	10000	98,67%
ResNet, DenseNet, EfficientNet, VGG	Zhou Z. et al, 2018 [93]	23779	98,67 %

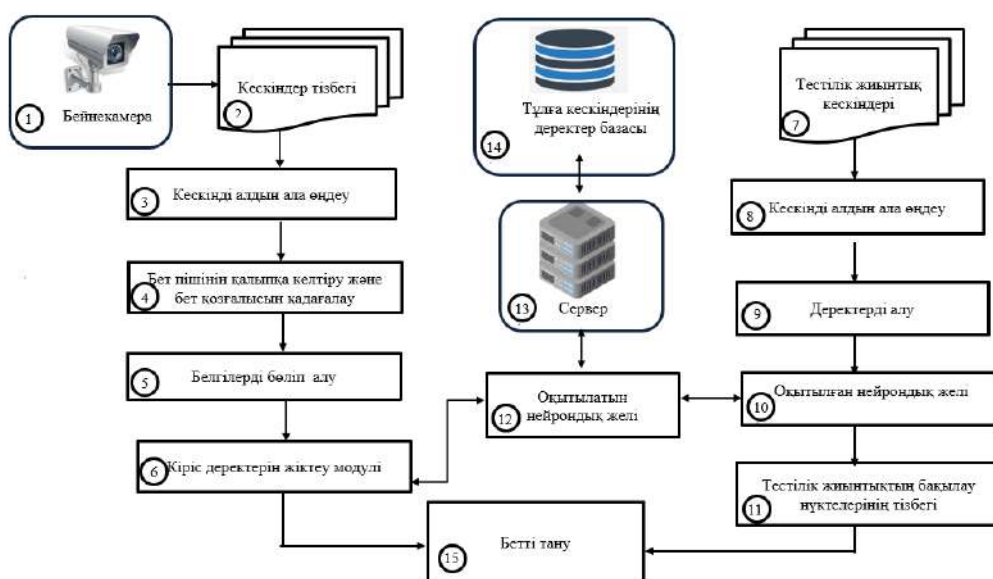
Дәстүрлі статистикалық әдістердің, соның ішінде негізгі компоненттерді талдау әдісі мен толқындық түрлендірудің орташа тану дәлдігі 79-90% аралығында құрайды. Машиналық оқытуға негізделген әдістер (тірек векторлар әдісі, арнайы беттік әдіс) дәстүрлі статистикалық әдістермен салыстырғанда жоғары нәтиже көрсетті, тану дәлдігі 85–95% дейін жетті. Ал нейрондық желілерге негізделген әдістердің тану дәлдігі 97-98% (1-кесте), бұл бетті тану әдістерінің ішінде ең жоғары көрсеткіш. Сондықтан осы әдістер көбінесе бетті тану бағытындағы зерттеулерде қолданылуда.

2.1-суретте бетті тану әдісінің нейрондық желіге негізделген схемасы көрсетілген. Суретте келесі элементтер көрсетілген: 1-камера; 2-кескін диаграммасының модулі; 3-кескінді алдын ала өңдеу модулі; 4-бет формасын қалыпқа келтіру және бет қозғалысын қадағалау модулі; 5-белгілерді бөліп алу модулі; 6-кіріс деректерін жіктеу модулі; 7 – тестілік жиынтық кескіндері; 8-кескіндерді алдын ала өңдеу модулі; 9-деректерді алу модулі; 10-оқытылған нейрондық желі модулі; 11-тестілік жиынтықтың бақылау нүктелерінің тізбегі модулі; 12-оқытылатын нейрондық желі модулі; 13-сервер; 14-бет кескінінің дерекқоры; 15-бетті тану модулі.

Бетті тану жүйесі келесі компоненттерден тұрады:

Камера (1), деректерді өңдеу және сақтау сервері (13), дерекқор (14) және бетті тану, деректерді жинау және тану жүйелерін оқыту модульдері көрсетілген. Бейнекамера (1) сервермен (13) Интернет байланысы арқылы байланысады, ал

сервер (13) бет кескінінің дерекқорына сақтайды (14). Сонымен қатар, сервер (13) бейне камерадан деректерді ала алады (1) және оны келесі дәйекті модульдер арқылы өңдей алады: кескіндер тізбегі (2), кескінді алдын ала өңдеу модулі (3), бет пішінін қалыпқа келтіру және бет қозғалысын қадағалау модулі (4), өңдеу нәтижесін алу үшін серверден (13) бетті тану модуліне (15) жіберіледі. Сервер (13) сонымен қатар деректерді келесі модульдер арқылы жібере алады: сынақ кескінін жинау блогы (7), кескінді алдын ала өңдеу блогы (8) және деректерді алу модулі (9). Бұл деректер сервердегі (13) оқытылған нейрондық желіге (10) жіберіледі. Оқытылған нейрондық желі (10) келесі модульдермен өзара әрекеттеседі: оқытылатын нейрондық желі (12), сервер (13), бет-әлпетті тану модулі (15), тестілік жиынтықтың бақылау нүктелерінің тізбегі модулі (11). Сервер (13) оқыту нейрондық желісі (12) арқылы кірісті жіктеу модулімен (6) мәлімет алмасады.



Сурет 2.1. Бетті тану процесінің схемасы

Тану процесі нақты уақыт режимінде орындалады. Жүйе бейнені жинау құрылғысының құрамдас бөліктерінен тұрады: бүкіл процесс барысында бейне сигналдарды үздіксіз жинау бөлігі; кодтауды қамтамасыз ету бөлігі. Құрылғы алынған бейне сигналын тасымалдауға, сақтауға қолайлы форматқа түрлендіреді. Желілік беру жүйесі жүйенің барлық модульдерінің арасында бейне деректер алмасуын реттеп отырады. Бейне сигналдарды сақтау және резервтік көшірмесін алу үшін деректерді сақтаудың арнайы ішкі жүйесі қолданылады.

Конволюциялық нейрондық желіге негізделген бетті танудың математикалық моделі

Бейне ағыны кадрлар тізбегі ретінде беріледі. Конволюциялық нейрондық желі әр кадрда бет аймағын жеке векторға айналдырады. Бет векторлар арасындағы қашықтықты салыстыру арқылы анықталады. Бейне ағынының кадр белгісі келесідей жазылады (2.1):

$$\mathcal{I} = \{I_t\}_{t=1}^T, I_t \in R^{H \times W \times C} \quad (2.1)$$

мұндағы H, W — кадр өлшемі, C -арналар саны. $\mathcal{I}$ — бұл кескіндер тізбегі. I_t — бейне ағынның t -кадрының түпнұсқасы, T —қарастырылып отырған бейне тізбегіндегі кадрлар саны.

1- кезең. Кадрды алу және бақылау моделі.

Деректерді жинау және бақылау моделі шу мен кедергілерді ескере отырып қарастырылады. Әр кадр көрінісі пайдалы ақпараттық сигнал мен қосымша шудан тұратын кездейсоқ процесс ретінде сипатталады. Шу кездейсоқ процесс, бақылау нәтижесіне әсер етеді. Әсер сыртқы және ішкі бұрмаланулардан туындайды. Шу сыртқы және ішкі әсерлерді имитациялайды, осы факторлар бақылау нәтижесіне ықпал етеді (2.2).

$$I_t = S_t + \varepsilon_t \quad (2.2)$$

мұндағы S_t — шынайы кескін, ε_t — шу немесе кедергі ($\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$).

2- кезең. Кескінді өңдеу, форматты түрлендіру.

Бастапқы деректер кадрды алдын ала өңдеу кезеңінде кіріс деректерге түрлендіріледі. Ол нейрондық желі талаптарына сай жүргізіледі. Процесс кескін өлшемін, пиксельдер диапазонын, статистикалық деректерін біріктіреді. Қалыпқа келтіру келесідей орындалады:

$$\tilde{I}_t = \phi(I_t) = \frac{I_t - \mu}{\sigma} \quad (2.3)$$

мұндағы μ -пиксель жарықтығының орта мәні, σ -орта квадраттық ауытқу. $\tilde{I}_t$ -өңдеуден кейінгі кадр, $\phi(I_t)$ - кескінді алдын ала өңдеу және қалыпқа келтіру функциясы.

Осы түрлендіргеннен кейін деректердің таралуы қалыпқа келеді. Дисперсия бірлікке жақындаса, онда нейрондық желінің тұрақтылығы артады, есептеу процесі тиімді болады.

Кескіннің кеңістіктік өлшемдерін өзгерту үшін кескіннің өлшемін өзгерту операциясы қолданылады (2.4). Түрлендіру келесі түрде көрсетіледі:

$$\tilde{I}_t = \text{Resize}(I_t; H', W') \quad (2.4)$$

I_t —бастапқы t -кадр, $\tilde{I}_t$ —өлшемдері өзгертілген кадр, H' және W' —сәйкесінше қайта масштабтаудан кейінгі кадрдың биіктігі мен ені. Операция кескінді нейрондық желінің кіріс қабатының талаптарына сәйкес біріктіреді. Модельдің есептеу күрделілігін арттырады.

3 кезең. Бетті анықтау. Кадрдағы бетті анықтау үшін детекция кезеңі орындалады. Әрбір өңделген кадр үшін бетке сәйкес келетін тікбұрышты аймақ анықталады.

Бет детекторы модель ретінде қарастырылады. $D(\cdot; \theta_D)$ параметрлері θ_D арқылы сипатталады. Кадрдан бет болуы мүмкін бірнеше аймақтарды табады (2.5):

$$\mathcal{B}_t = D(\tilde{I}_t; \theta_D) = \{(b_{t,k}, s_{t,k})\}_{k=1}^K \quad (2.5)$$

мұндағы $b_{t,k} = (x, y, w, h)$ координаттары k -аймақ координаталары. $s_{t,k} \in [0,1]$ -дәлдік коэффициенті, берілген болжам аймағының бетке қатысты болу ықтималдығын анықтайды. K - берілген болжам аймақтарының жалпы саны.

Беттің ықтимал аймағын анықтау үшін сенімділік көрсеткішін максимизациялау қажет. Шекті мән τ_D ескерілген болса, онда ең сенімді бет аймағы келесі өрнек арқылы таңдалады (2.6).

Ең жақсы кадрды таңдау:

$$b_t^* = \arg \max_{b_{t,k}} s_{t,k}, \text{ шарт } s_{t,k} \geq \tau_D \quad (2.6)$$

b_t^* - кадрдағы ең дұрыс деп таңдалған бет аймағы. Сенімділік мәні шектен жоғары, ең үлкен болған аймақ. Осы тәсіл бетті табу қателіктерін азайтады, бетті дәл табуға көмектеседі.

4 кезең. Бетті негізгі нүктелер бойынша туралау. Бетті негізгі нүктелер арқылы тегістеу процесі кескінді стандартты қалыпқа келтіру арқылы орындалады. Тәсіл көз, мұрын сияқты маңызды нүктелерді арнайы эталон нүктелермен салыстырады да бет кескінін стандартты координат жүйесіне келтіреді. Алдымен алгоритм әр кескіндегі бет аймағының негізгі нүктелерін анықтау үшін қолданылады (2.7). Нәтижесінде негізгі бағыттар тізбегі келесідей сипатталады:

$$P_t = L(\tilde{I}_t, b_t^*; \theta_L) = \{p_{t,i}\}_{i=1}^m, p_{t,i} \in R^2 \quad (2.7)$$

Мұндағы $L(\tilde{I}_t, b_t^*; \theta_L)$ -негізгі нүктелерді анықтайтын модель, $\tilde{I}_t$ -алдын ала өңделген бейне кадр, b_t^* таңдалған бет аймағы, ал θ_L -модель параметрлері.

Содан кейін геометриялық түрлендіру параметрлері эталондық негізгі нүктелерді бақылау нүктелерімен салыстырылып бағаланады. Осыған аффиндік немесе проективтік түрлендіру қолданылады, ол оңтайландыру есебін шешу арқылы келесідей жүзеге асырылады (2.8):

$$T_t = \arg \min_T \sum_{i=1}^m \|T(p_{t,i}) - p_i^{ref}\|^2 \quad (2.8)$$

мұндағы T_t -геометриялық түрлендіру матрицасы, $p_{t,i}$ - ағымдағы кадрдың негізгі нүктесі, ал p_i^{ref} - беттің бақылау нүктесі.

Соңғы түрлендіру бойынша бет қалыпқа келтіріледі, белгілі бір мөлшерге дейін қиылып алынады (2.9). Соңында тегіс, тураланған бет кескіні алынады:

$$F_t = \text{WarpCrop}(\tilde{I}_t, b_t^*, T_t) \in R^{h \times w \times c} \quad (2.9)$$

мұндағы F_t -тураланған бет кескін аймағы, $h \times w$ -кескін өлшемі, ал c - арналар саны.

5 кезең. Нейрондық желі арқылы эмбеддинг белгілерді алу. Бетті тану жүйесінде келесі кезең бет кескінінен белгілерді алу орындалады. Мұнда тураланған бет кескіні нейрондық желіге негізделген энкодер арқылы өңделі. Бетке тән ықшам сипаттамалық эмбеддинг векторлары түзеледі. Осы вектор кейінгі салыстыру мен сәйкестендіру процестерінде қолданылады. Бет кескінінің негізгі айқындаушы ерекшеліктерін жинақтайды.

Энкодер моделі бет кескінін көпөлшемді белгілер кеңістігіне проекциялайды. Соңында келесі түрде сипатталатын белгілер векторы алынады (2.10):

$$z_t = E(F_t; \theta_E) \in R^d \quad (2.10)$$

мұндағы $E(F_t; \theta_E)$ -нейрондық желіге негізделген энкодер, F_t -алдын ала конфигурацияланған бет кескіні, θ_E - модель параметрі, ал z_t -д өлшемді бет эмбеддингi.

Алынған эмбеддинг векторы әртүрлі бет кескіндерін салыстыруды тұрақтандыру үшін қалыпқа келтіріледі. Ол үшін L_2 нормасы қолданылады. Қалыпқа келтірілген эмбеддинг келесідей анықталады (2.11):

$$\hat{z}_t = \frac{z_t}{\|z_t\|_2} \quad (2.11)$$

мұндағы $\|z_t\|_2$ -вектордың Евклидтік нормасы.

6- кезең. Бет эмбеддингтерін деректер базасымен салыстыру (тексеру, сәйкестендіру). Шешім қабылдау кезеңінде бетті тану жүйесінде алынған эмбеддингтер деректер базасында сақталған үлгілермен салыстырылады. Процесс екі режимде орындалады: тексеру; сәйкестендіру.

Тексеруде екі бет бір адамға тиесілі ме, соны анықтау үшін олардың эмбеддинг векторларының ұқсастығын есептеледі (2.12). $\hat{z}_a, \hat{z}_b$ қалыпқа келтірілген эмбеддингтері үшін ұқсастықты өлшеуде косинустық ұқсастық көрсеткіші қолданылады.

$$\text{sim}(\hat{z}_a, \hat{z}_b) = \hat{z}_a^T \hat{z}_b \quad (2.12)$$

мұндағы $\hat{z}_a, \hat{z}_b$ -қалыпқа келтірілген эмбеддинг векторлары. Алынған ұқсастық мәні алдын-ала анықталған шекті мәнмен салыстырылады (2.13).

$$\text{same} = \begin{cases} 1, & \text{sim} \geq \tau_R \\ 0, & \text{sim} < \tau_R \end{cases} \quad (2.13)$$

мұндағы 1- екі кескіннің бір адам екендігін, ал 0- олардың басқа адамдар екенін білдіреді.

Жүйе сәйкестендіру режимінде бір бет кескінін бірнеше адамнан тұратын деректер қорымен салыстырады. Мұнда шаблондар жиыны қолданылады (2.14):

$$\mathcal{G} = \{(\mu_j, y_j)\}_{j=1}^N \quad (2.14)$$

мұндағы μ_j - j-адамның эмбеддингiнің үлгісі, ал y_j - сол адамға сәйкес келетін класс белгісі. Берілген бет кескіні үшін алынған қалыпқа келтірілген эмбеддинг векторы деректер қорындағы барлық эталондармен салыстырылады. Яғни $\hat{z}_t$ векторы әрбір эталонмен салыстырылады, нәтижесінде ұқсастығы ең жоғары класс таңдалады (2.15):

$$j^* = \arg \max_j \hat{z}_t^T \mu_j \quad (2.15)$$

Максималды ұқсастық мәні τ шекті мәнінен аз болса, жүйе бұл бетті дерекқорда жоқ "анықталмаған тұлға" ретінде табады (2.16). Шешім:

$$y_t = \begin{cases} y_{j^*}, & \max_j \hat{z}_t^T \mu_j \geq \tau_U \\ \text{белгісіз}, & \text{әйтпесе} \end{cases} \quad (2.16)$$

7 кезең. Эталондар жиынтығын жаңарту.

Әрбір адам үшін деректер қорында сәйкес эталондық вектор сақталады. Бұл вектор бірнеше бақылаулардан кейін қалыптасады. Егер жүйе жаңа кадрдағы бетті деректер қорындағы y_j —белгілі бір адам деп таныса, онда сол адамның эмбеддинг эталоны жаңа дерек арқылы жаңартылады. Жаңарту экспоненциалды жылжымалы орта әдісі арқылы жаңартылады (2.17):

$$\mu_j^{(new)} = (1 - \alpha)\mu_j^{(old)} + \alpha\hat{z}_t \quad (2.17)$$

Мұндағы $\mu_j^{(old)}$ - эмбеддинг эталоны. $\hat{z}_t$ -ағымдағы кадрдан қалыпқа келтірілген эмбеддинг, α -жаңарту коэффициенті ($0 < \alpha < 1$).

Ұқастықты есептеу үшін эталон векторын L2-нормализация арқылы қалыпқа келтіреміз. Ол масштабты бірдей етіп, нәтижелерді тұрақты жасау үшін қажет (2.18):

$$\mu_j^{(new)} \leftarrow \frac{\mu_j^{(new)}}{\|\mu_j^{(new)}\|_2} \quad (2.18)$$

Деректер қорындағы эталондар жаңа мәліметтермен біртіндеп жаңарып отыратынын аңғаруға болады. Бұл әдістің бет ерекшеліктерін дәл анықтайтыны, жүйе жұмысының ұзақ уақытта тиімді болатыны байқалды.

8 кезең. Бейне ағынын уақытша біріктіру. Уақытша тұрақтандыру әдісі кадрлар арасындағы кездейсоқ ауытқуларды азайтуды қамтамасыз етеді. Ол бейне ағынды өңдеу кезінде қолданылды. Бет кескінінің ерекшеліктерін үздіксіз кескінге біріктіру арқылы тану нәтижелерін тұрақтандыратынын байқауға болады. Тиімділік үшін терезелерді уақытша біріктіру әдісі ұсынылды (2.19). Уақытша тураланған эмбеддинг келесідей анықталды:

$$\bar{z}_t = \frac{1}{M} \sum_{k=0}^{M-1} \hat{z}_{t-k} \quad (2.19)$$

мұндағы $\hat{z}_{t-k}$ - алдыңғы кадрға нормаланған эмбеддингтер, ал M - уақытша терезенің өлшемі. Салыстыру тұрақтылығы үшін зерттеу барысында алынған орташа мән қайтадан қалыпқа келтірілді (2.20):

$$\hat{\hat{z}}_t = \frac{\bar{z}_t}{\|\bar{z}_t\|_2} \quad (2.20)$$

$\hat{\hat{z}}_t$ нәтижелік векторы бейне ағынындағы беттің уақытша және тұрақты қасиеттерін білдіреді. Әрбір сәйкестендіру осы тураланған эмбеддинг арқылы орындалды.

2.2 Шу кезінде бет кескіндерін тану дәлдігін жақсартудың адаптивті әдістері

Бетті тану кезінде негізгі мәселелердің бірі — шу. Шу кескін сапасын төмендетеді. Бет белгілерін дұрыс анықтауға кедергі келтіреді. Ол көбінесе жарықтың жеткіліксіздігінен немесе түсіру құрылғыларының ерекшеліктерінен пайда болады. Жарықтандырудың өзгеруі де бет тану жүйесінің тану дәлдігін түсіреді. Шамадан тыс немесе біркелкі емес жарық кескін сапасын төмендетеді. Бет белгілері толық анықталмай, тану кезінде қателіктер пайда болады. Әсіресе бейне ағындарда жарықтың уақыт бойынша өзгеруі жүйенің тұрақты жұмысын қиындата түседі. Сондықтан алдын ала өңдеу кезеңінде шуды алып тастау өте маңызды. Ол үшін сүзгілеу және кескінді жақсарту әдістері қолдану қажет болады. Мұндай әдістер жарық әсерін азайтып, бет белгілерін айқынырақ көрсетуге мүмкіндік береді. Жалпы тану жүйесінің тиімділігін арттыра түседі.

Қойылған мәселені шешу үшін бейне ағындардағы кескінді жақсарту әдістерін зерттеп, тиімдісін математикалық моделінің алдын ала өңдеу кезеңіне интеграциялайтын боламыз.

Көптеген жылдар бойы бетті тану дәстүрлі әдістер негізінде зерттеліп келді [106], ал қазіргі уақытта ол терең оқытуға негізделген жаңа алгоритмдер көмегімен зерттелуде [107-112]. Бетті танудың тиімділігін арттыру мақсатында модельдерді оқытуда нормализация әдістері және бет белгілеріндегі айырмашылықтарды анықтауға арналған эвристикалық модельдер енгізілді. Z. Li және т.б. (2011) локалдық белгілерді сипаттауға негізделген көптапсырмалы дискриминанттық талдау әдісін ұсынған [113]. Ал D. Gong және т.б. (2013) тұлғаға тән белгілерді модельдеу және жас ерекшеліктеріне байланысты айырмашылықтарды азайту үшін жасырын факторлар талдауын (HFA) қолдануды ұсынған [114].

Жарықтандыру мәселесі қазіргі заманғы бетті тану жүйелерінің негізгі мәселелерінің бірі болып қала бермек. Көптеген қолданбалар үшін шектеуші факторға айналған. Зерттеулер жарық деңгейінің айтарлықтай өзгеруі тану алгоритмдерінің өнімділігіне тікелей әсер ететінін көрсетеді. Осыған байланысты біркелкі емес жарықтандырудың кескін сапасына әсерін азайту, кескін айқындылығын және жарықтықтың біркелкілігін арттыру тәсілдері зерттелуде [115].

Қойылған мәселені зерттей келе шу әсеріне ұшыраған кескінді жақсарту үшін Retinex алгоритмі таңдалды. Бұл бөлімде бет кескінін тануда кедергілерді жою мақсатында интеграцияланатын көпмасштабты Retinex алгоритмі зерттелді. Зерттеу барысында Retinex алгоритміне негізделген сызықтық емес жарықтықты түзетуге арналған адаптивті кескінді оңтайландыру алгоритмі әзірленіп, оның эксперименттік тұрғыдан тексеру нәтижелері ұсынылды. Ұсынылған алгоритм жоғары есептеу тиімділігімен ерекшеленеді. Мақсаты-күрделі жарықтандыру жағдайларының кескіндерді өңдеу сапасына кері әсерін жою. Алгоритм кескін сапасын жақсартып түседі. Әдіс алдын ала өңдеу кезеңіне енгізгеннен кейін модель кескінді әртүрлі жағдайларда (жарық, бұрыш) да айқын тани береді.

Retinex теориясына сәйкес, кескіннің жарықтығы сыртқы жарық пен объектінің шағылысуына байланысты. Біркелкі емес жарықта кескіннің көршілес аймақтарының жарықтығы әртүрлі болады. Бұл бұрмалануға әкеледі және кескін сапасын нашарлатады [116-117].

Көптеген ғалымдар өз зерттеулерінде бетті тану кезінде модельдеуді жақсарту үшін Retinex теориясын қолданған. Retinex әдісі арқылы кескін сапасын жақсартады. Алдымен жарықтандыру компоненті ерекшеленеді. Бұл кескіннің баяу өзгертін бөлігі, оны бағытталған сүзу арқылы алуға болады. Мұндай сүзгі жарықтандыруды тегістейді және шекараларды сақтайды. Сондықтан ол басқа әдістерге қарағанда тиімді жұмыс істейді [118-122].

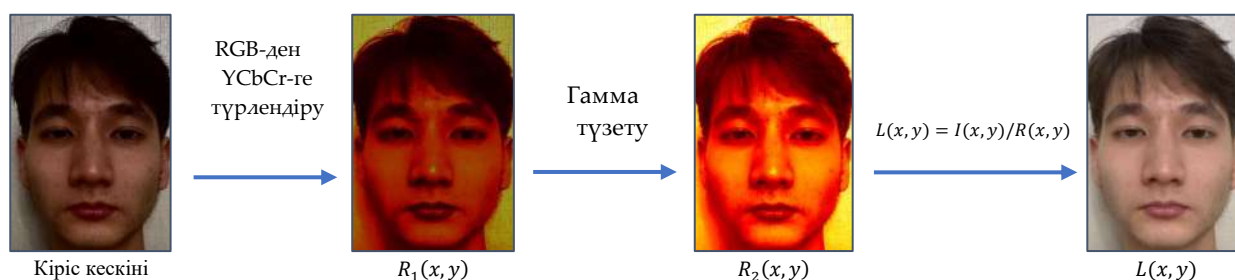
Біркелкі емес жарық әсерін азайту үшін Retinex алгоритмі ұсынылып отыр. $I(x,y)$ кескіннің берілуі:

$$I(x,y) = R(x,y) \times L(x,y) \quad (2.21)$$

Мұнда, $I(x,y)$ -кіріс кескін, $L(x,y)$ -жарықтандыру компоненті, $R(x,y)$ -шағылу компоненті. Бұл әдісте $I(x,y)$ кескіні өңдеу үшін негіз ретінде қолданылады. Кескінді туралап, шекараларын сақтап қалу үшін бағытталған көп масштабты сүзгі қолданыла алады. Ұсынылған әдіс сапасыз кескін бөлшектерін әжептәуір жақсартады. Кескіндерді біріктіру арқылы оларды қалпына келтіруге болады.

Retinex алгоритмінде кескіннің жарығын дәл анықтау қиын. Сондықтан алдымен кескін RGB форматынан жарықты бөлуге ыңғайлы түстік кеңістікке ауыстырылады. Содан кейін жарықтық бөлігі (Y) бөлініп алынады. Осы жарық мәліметі негізінде кескін түзетіледі. Кейін гамма-түзету қолданылады. Осылай кескіннің сапасы жақсарады.

Алгоритмнің жұмысы бірнеше қарапайым қадамнан тұрады: 1) қараңғыланған беттік кескінін RGB түстік кеңістігінен YCbCr кеңістігіне түрлендіру және жарықтық компонентін бөліп алып, бастапқы жарықтандырылған $R_1(x,y)$ кескінін қалыптастыру; 2) жарықтықты түзету арқылы жақсартылған $R_2(x,y)$ кескінін алу; 3) Retinex алгоритміне сәйкес жарықтандыру компонентін өңдеу нәтижесінде жақсартылған $L(x,y)$ кескінін қалыптастыру (2.2-сурет).



Сурет 2.2. Алгоритм жұмысының негізгі механизмі

Бет кескіні алдымен алдын ала өңдеуден өтеді. Содан кейін бет белгілері бөлініп алынады. Алдымен бет аймағы бастапқы фотосуреттен координаттар бойынша кесіп алынады. Кейін көру бұрышы түзетіледі. Келесі қадамда кескін өлшемі 640×640 пиксель өлшеміне келтіріліп түзетіледі. Осыдан кейін жарықтандыру бағаланады және қажет болған жағдайда қосымша жарық енгізіледі. Мұндай түзету бет позициясы пен жарықтың әсерін азайтады және тану дәлдігін жақсартады (2.2-сурет).

Бетті тану кезеңіне дейін Faster R-CNN моделі жүктеледі [120]. Бұл модельдің LFW деректер жиынында тану дәлдігі 99,05%, кадр өңдеу жылдамдығы секундына 25 кадр. Ал бетті тану үшін өлшемі 512 болатын эмбеддинг-векторлар қолданады. Face Net моделі кескіндерін евклид кеңістігінде тікелей салыстыру арқылы олардың ұқсастық деңгейін анықтайды. Алдын ала өңдеу аяқталғаннан кейін бет кескіні стандартталып, сәйкес белгілер векторы арқылы алынады.

Кескін өлшемі $M \times N$, ал сұр реңк деңгейі L болсын. Бастапқы шектік мән k таңдалып, кескін пикселдері осы шек бойынша екі класқа бөлінеді: C_1 – қараңғы аймақтар, C_2 -жарық аймақтар.

Бірінші кластың ықтималдығы (2.22):

$$P_1(k) = \sum_{i=0}^k p_i \quad (2.22)$$

мұндағы, p_i сұр реңк деңгейі i болатын пиксельдердің ықтималдығы. Екінші кластың ықтималдығы (2.23):

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \quad (2.23)$$

$t_1(k), t_2(k)$ кластарының орташа сұр реңк мәндері (2.24-2.25):

$$t_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^k i p_i, \quad (2.24)$$

$$t_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^L i p_i, \quad (2.25)$$

$t_{mean}(k)$ жалпы орташа сұр реңк мәні (2.26):

$$t_{mean}(k) = P_1(k)t_1(k) + P_2(k)t_2(k). \quad (2.26)$$

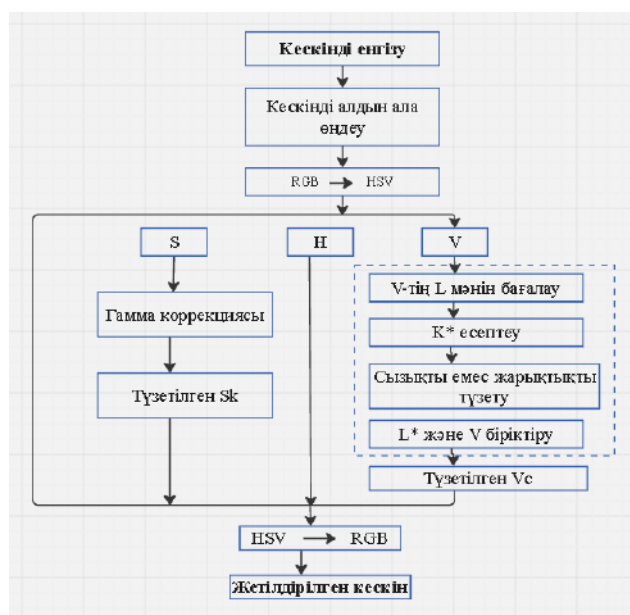
(2.27) формуласы арқылы класс аралық дисперсияны келесі түрде есептеуге болады [123]:

$$\delta_B^2(k) = P_1(k)[t_1(k) - t_{mean}(k)]^2 + P_2(k)[t_2(k) - t_{mean}(k)]^2 \quad (2.27)$$

$\delta_B^2(k)$ класаралық дисперсиясы үшін k^* максималды мәнінің оңтайлы шегі болып табылады (2.28):

$$\delta_B^2(k^*) = \max_{0 \leq k \leq N-1} \delta_B^2(k) \quad (2.28)$$

Егер максимум мән бірегей болмаса, онда бірнеше мәндердің орташа арифметикалық шамасы қабылданады. Осы шектік мән негізінде кескіннің жарық және қараңғы аймақтары бөлініп, сызықтық емес жарықтықты түзету арқылы адаптивті жақсарту алгоритмі жүзеге асырылады (2.3-сурет).



Сурет 2.3. Ұсынылған алгоритмнің блок-схемасы

1-қадам. Бастапқы кескін гаусстық сүзгілеу және адаптивті гистограмманы теңестіру арқылы алдын ала өңделеді. Нәтижесінде өлшемі $M \times N$ болатын кескіндегі айқын шудың әсері азайып, жарықтықтың таралуы біркелкі болады және кескін кейінгі өңдеу үшін алдын-ала оңтайландырылады.

2-қадам. 1-қадамда алынған алдын ала өңделген кескін RGB түстік кеңістігінен HSV түс кеңістігіне түрлендіріледі. R, G және B арналарындағы жоғары корреляцияға байланысты олардың бір мезгілде өзгеруі түстердің бұрмалануына әкелуі мүмкін. Ал HSV түс кеңістігі адамның түсті қабылдау ерекшеліктеріне көбірек сәйкес келеді. HSV үш компоненттен тұрады: реңк (H), қанықтылық (S) және жарықтық (V) [124]. Кескіннің жарықтық ақпараты жарықтық V компонентінде сақталатындықтан, ол арналарды бөлу арқылы жеке алынады.

3-қадам. V жарықтықты компоненті адаптивті жарықтықты реттеу жүйесіне енгізіледі, нәтижесінде жақсартылған жарықтық компоненті V_c алынады.

4-қадам. Қанықтылық компонентіне гамма-коррекция қолданылып, түзетілген S_k мәні алынады. Кескіннің табиғи реңкін сақтау мақсатында H компоненті өзгеріссіз қалдырылады. Кейін түзетілген компоненттер біріктіріліп, жаңартылған HSV кескіні қалыптастырылады.

5-қадам. Алынған жаңа HSV кескіні бастапқы HSV кескінімен сызықтық интерполяция әдісі арқылы біріктіріліп, жақсартылған соңғы HSV кескіні алынады. Кейін бұл кескін RGB түстік кеңестігіне қайта түрлендіріліп, жақсартылған соңғы кескін қалыптастырылады.

RSA алгоритмін қолдана отырып, кескіндегі жарық және қараңғы аймаққа бөлуге арналған шектік мән анықталады. Осы шектік мән негізінде кескін екі бөлікке бөлінеді. Бастапқы кескін, орташа мән әдісі бойынша алынған сегменттеу картасы және OTSU әдісі арқылы алынған сегменттеу картасы салыстырылған кезде, Otsu әдісінің сегменттеу әсері жақсырақ екені байқалады. Бұл әдіс жарық және қараңғы аймақтарды анық әрі дәл бөлуге мүмкіндік береді. Фрагменттелген аймақтардың шекаралары айқын сақталады.

Шу кезінде кескін сапасын жақсарту үшін ұсынып отырған көпмасштабты Retinex алгоритмінің тиімділігін бағалау үшін эксперимент жасалды. Жарық деңгейінің өзгеруі кескін сапасына және бетті тану нәтижелеріне қалай әсер болатындығы зерттелді.

Эксперименттік зерттеулер келесі құралдар мен деректер негізінде жүргізілді:

1. Әртүрлі жарықтандыру жағдайларын имитациялау мақсатында жарық деңгейі мен бағытын өзгертуге мүмкіндік беретін реттелетін шам қолданылды. Жарықтандыру деңгейі әлсіз жарықтан (50 люкс) жоғары жарыққа дейін (600 люкс) өзгертіліп отырды. Жарық көзі түсірілім объектісіне қатысты 45° бұрышта және 1,6 метр тұрақты қашықтықта орнатылды. Кескіндер нақты уақыт режимінде Hikvision DS-2CD1347G0-I IP камерасы арқылы түсірілді. Камера 1/3 дюймдік Progressive Scan CMOS жарыққа сезімтал матрицасымен жабдықталған, ажыратымдылығы 4 Мп, бейне беру форматы 2560x1440 пиксель

және кадр жиілігі секундына 20-30 кадрды құрайды. Эксперименттің қайталанымдылығын қамтамасыз ету мақсатында камера түсіру объектісіне қатысты тұрақты қашықтықта орнатылды.

2. Эксперименттерді жүргізу үшін адамдардың 3045 фотосуреттерін қамтитын жеке мәліметтер жиынтығы қолданылды. Зерттеу барысында бет-әлпетті тану моделін оқыту және бағалау үшін VGGFace2 деректер жиыны пайдаланылды. Кескіндер интернеттен (Google Image Search) ашық деректер қорларынан (<https://huggingface.co/datasets/RichardErkhov/VGGFace2/tree/main/data>) жиналған, сондықтан әр адамның: әртүрлі жастағы, әртүрлі ракурстағы, әртүрлі жарықтағы фотолары бар. Деректер ішкі жарықтандырудың өзгеруімен нақты жағдайларда жиналды. Әр қатысушы әр түрлі жарық жағдайларында түсірілді: а) әлсіз жарық (ымырт жағдайын имитациялау); б) қалыпты күндізгі жарық; с) бағытталған күшті жарық. Қатысушылардың құқықтарын қорғау мақсатында деректер жиынтығы жасырын болды, тек ғылыми-зерттеу мақсатында қолданылды.

3. Адаптивті кескінді жақсарту әдісі Retinex тәсіліне негізделіп, жарықтықты сызықтық емес реттеу арқылы жүзеге асырылды. Өңделген кескіндер YOLOv5 моделі арқылы талданып, әртүрлі жарық жағдайларында объектілерді анықтау дәлдігі бағаланды.

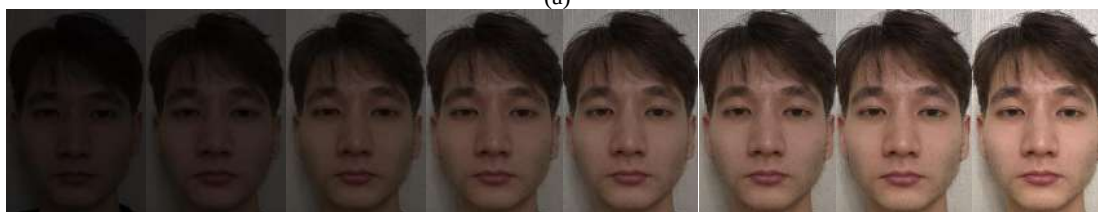
4. Қатысушылар бір бөлмеде, камераның бекітілген орналасу жағдайында фотоға түсірілді. Жарық деңгейі біртіндеп өзгертіліп отырды, ал жарықтық мәндері фотометрмен (люксметр) арқылы тіркелді.

5. Алгоритм тиімділігін бағалау үшін келесі көрсеткіштер пайдаланылды: кескіннің орташа сұр деңгейі, кескінді өңдеу жылдамдығы, орташа абсолюттік қателер.

2.4-суретте 1 м, 2 м және 3 м қашықтықта түсірілген, әртүрлі қараңғылану деңгейіндегі бет кескіндері көрсетілген.



(a)



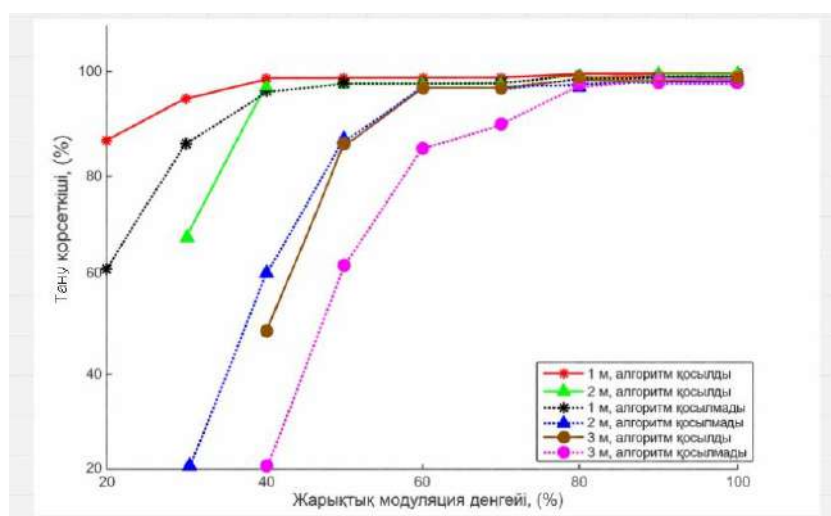
(б)



(6)

Сурет 2.4. 1, 2 және 3 м қашықтықта әр түрлі қараңғылану деңгейлерінде алынған бет кескіндері, (а) 1 м қашықтықта түсірілген бет кескіндері, солдан оңға қарай қараңғылану деңгейі 20%-дан 100%-ға дейін, (ә) 2 м қашықтықта түсірілген бет кескіндері, солдан оңға қарай қараңғылану деңгейі 30%-дан 100%-ға дейін, (б) 3 м қашықтықта түсірілген бет кескіндері, солдан оңға қарай қараңғылану деңгейі 40%-дан 100%-ға дейін.

2.5-суретте 1, 2 және 3 метр қашықтықта әртүрлі жарықтық деңгейлерінде бетті тану нәтижелері көрсетілген. Жарықтық деңгейі артқан сайын тану тиімділігі да жоғарылайтындығы байқалады. 1 м қашықтықта жүргізілген эксперимент көрсеткендей, кескіннің жарық деңгейі 70% болғанда бет тану дәлдігі 98%-ға жетті. Бұл нәтиже кескінді оңтайландыру алгоритмін қолданған кездегі көрсеткіштермен шамалас.



Сурет 2.5. Жарық деңгейіне байланысты бет тану дәлдігі (алгоритмге дейін/кейін)

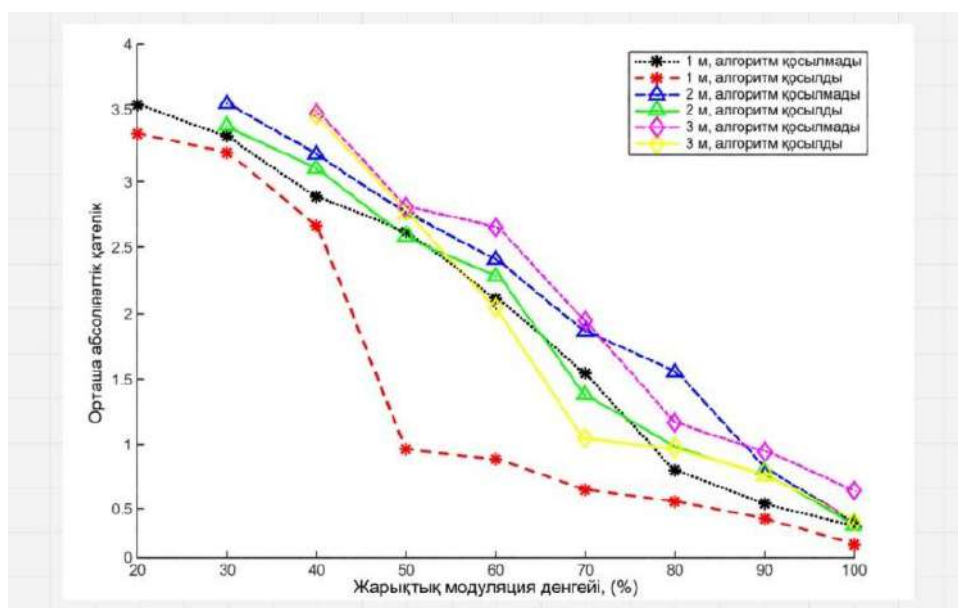
1 м қашықтықта және жарықтық деңгейі 20% болған кезде бетті танудың орташа жылдамдығы 60.98% болды. Кескінді жақсарту алгоритмін қолданғаннан кейін бұл көрсеткіш 86%-ға дейін өсті. Ал 1 м қашықтықта жарықтық деңгейі 60% болған кезде бетті танудың орташа дәлдігі 97% болды, ал ұсынылған кескінді жақсарту алгоритмін қолдану нәтижесінде бұл көрсеткіш 98%-ге дейін жоғарылады. 2 метр қашықтықта және жарықтық деңгейі 20% болған кезде бетті тану жүйесі кескінді анықтай алмады. Талдау жүргізу үшін ақпарат жеткіліксіз болғандықтан есептеу тоқтатылды.

2 метр қашықтықта және 50% қараңғылану деңгейінде бетті тану дәлдігі 86% құрады. Кескінді жақсарту алгоритмін қолданғаннан кейін бетті тану дәлдігі 11% -ға артты. 3 метр қашықтықта және 50% қараңғылану деңгейінде бетті тану дәлдігі 61,09% болғанын көруімізге болады. Ал кескінді жақсарту алгоритмін енгізгеннен кейін бетті тану дәлдігі 23.91%-ге көтерілгенін көруімізге болады.

Эксперимент барысында алынған нәтижелер ұсынылған кескінді жақсарту алгоритмінің жарықтандыру деңгейі төмен жағдайларда да бетті тану тиімділігін

арттыратынын көрсетті. Ол әртүрлі қашықтық пен жарық жағдайында жүйенің тұрақты жұмысын қамтамасыз етеді.

2.6-суретте 1, 2 және 3 м қашықтықта әртүрлі қараңғылану деңгейлерінде алгоритмге дейін және кейінгі орташа абсолюттік қателік мәндері көрсетілген.



Сурет 2.6. Әртүрлі жарықта алгоритмге дейін/кейінгі орташа абсолюттік қате

Жарықтық деңгейі артқан сайын және кескінді оңтайландыру алгоритмі қолданғаннан кейін орташа абсолюттік қателік мәндері азаятыны байқалады. Бұл жарықтылықтың артуына байланысты бет кескіні айқынырақ көрініп, кескіндегі бөлшектердің анықтығы жоғарылануымен түсіндіріледі. 1 м қашықтықта және жарықтық деңгейі 50% болған жағдайда орташа абсолюттік қателік 2,63 құрады. Кескінді жақсарту алгоритмін қолданғаннан кейін бұл көрсеткіш 0,96 төмендеді. 2 метр қашықтықта және 50% қараңғылану деңгейінде орташа абсолюттік қателік 2,77 болды. Кескінді оңтайландыру алгоритмін қолданғаннан кейін орташа абсолюттік қателік мәні 2,60-қа азайды. 3 метр қашықтықта және 50% қараңғылану деңгейінде орташа абсолюттік қателік 2,82 құрады. Ал алгоритмді қолданғаннан кейін бұл көрсеткіш 2,77-ге төмендеді.

2.2-кесте, 2.3-кесте, 2.4-кестеде 1, 2 және 3 м қашықтықтарда кескінді жақсарту алгоритмін қолданғанға дейінгі және қолданғаннан кейінгі сұр реңктің орташа мәні, бетті тану жылдамдығы және орташа абсолюттік қателік көрсеткіштерінің салыстырмалы нәтижелері берілген.

Кесте 2.2. 1 м қашықтықта кескінді жақсарту алгоритмінің әсері

Индикатор	Алгоритмді қолданғанға дейін			Алгоритмді қолданғаннан кейін		
	Сұр орташа мәні	Тану көрсеткіші, %	Орташа абсолютті қателік	Сұр орташа мәні	Тану көрсеткіші, %	Орташа абсолютті қателік
Жарықтық модуляциясының деңгейі, 20%	7.54	60.98	3.56	24. 99	86	3.34
Жарықтық модуляциясының деңгейі, 30%	10.71	85.56	3.32	30. 07	95	3.21
Жарықтық модуляциясының деңгейі, 40%	13.29	95.44	2.89	37. 41	98	2.67
Жарықтық модуляциясының деңгейі, 50%	25.30	97.00	2.63	56. 61	98	0.96
Жарықтық модуляциясының деңгейі, 60%	35.89	97.00	2.15	68. 77	98.98	0.88
Жарықтық модуляциясының деңгейі, 70%	61.46	97.38	1.59	84. 97	98	0.64
Жарықтық модуляциясының деңгейі, 80%	89.60	97.40	0.81	103. 28	98.60	0.55
Жарықтық модуляциясының деңгейі, 90%	93.34	97.60	0.54	115. 09	98.60	0.42
Жарықтық модуляциясының деңгейі, 100%	93.21	97.60	0.38	123. 38	98.60	0.22

Алгоритм арқылы оңтайландырылған кескіндердегі сұр реңктің орташа мәні бастапқы кескіндермен салыстырғанда әртүрлі қашықтықта артқанын байқауға болады. Сонымен қатар жарықтық деңгейі артқан сайын және кескінді оңтайландыру алгоритмі қолданылғаннан кейін орташа абсолюттік қателік азайып, бетті тану дәлдігінің артқаны байқалады. Зерттеу нәтижелері көрсеткендей, бетті тану жүйесі 2 метр қашықтықта жарықтық деңгейі 20% болған жағдайда бетті анықтай алмады.

Кесте 2.3. 2 м қашықтықта кескінді жақсарту алгоритмінің әсері.

Индикатор	Алгоритмді қолданғанға дейін			Алгоритмді қолданғаннан кейін		
	Сұр орташа мәні	Тану көрсеткіші, %	Орташа абсолютті қателік	Сұр орташа мәні	Тану көрсеткіші, %	Орташа абсолютті қателік
Жарықтық модуляциясының деңгейі, 20%	N/A	N/A	N/A	N/A	N/A	N/A
Жарықтық модуляциясының деңгейі, 30%	10. 87	20.00	3.56	18. 90	67.00	3.40
Жарықтық модуляциясының деңгейі, 40%	17. 42	60.00	3.19	30. 11	96.89	3.10

деңгейі, 40%							
Жарықтық деңгейі, 50%	модуляциясының	26. 68	86.00	2.77	43. 88	97.00	2.60
Жарықтық деңгейі, 60%	модуляциясының	37. 61	96.00	2.43	57. 39	97.00	2.31
Жарықтық деңгейі, 70%	модуляциясының	45. 34	96.00	1.89	66. 61	97.00	1.45
Жарықтық деңгейі, 80%	модуляциясының	55. 51	96.30	1.60	79. 90	98.00	1.01
Жарықтық деңгейі, 90%	модуляциясының	68. 05	97.60	0.84	80. 81	98.50	0.79
Жарықтық деңгейі, 100%	модуляциясының	79. 33	97.60	0.42	90. 74	98.50	0.38

Сондай-ақ 3 метр қашықтықта жарықтандыру деңгейі 20% және 30% болған кезде жүйе бетті анықтай алмады.

Кесте 2.4. 3 м қашықтықта кескінді жақсарту алгоритмін қолданғанға дейінгі және қолданғаннан кейінгі нәтижелер

Индикатор	Алгоритмді қолданғанға дейінгі нәтижелер			Алгоритмді қолданғаннан кейінгі нәтижелер		
	Сұр орташа мәні	Тану көрсеткіші, %	Орташа абсолютті қателік	Сұр орташа мәні	Тану көрсеткіші, %	Орташа абсолютті қателік
Жарықтық деңгейі, 20%	N/A	N/A	N/A	N/A	N/A	N/A
Жарықтық деңгейі, 30%	N/A	N/A	N/A	N/A	N/A	N/A
Жарықтық деңгейі, 40%	15.53	21.02	3.50	35.00	49.00	3.45
Жарықтық деңгейі, 50%	17.00	61.09	2.82	40.07	85.00	2.77
Жарықтық деңгейі, 60%	22.12	84.00	2.67	40.23	95.90	2.08
Жарықтық деңгейі, 70%	30.16	88.67	1.97	51.21	96.0	1.08
Жарықтық деңгейі, 80%	41.33	96.56	1.19	60.56	98.00	1.00
Жарықтық деңгейі, 90%	51.22	97.00	0.98	68.00	98.00	0.79
Жарықтық деңгейі, 100%	61.90	97.00	0.66	71.98	98.00	0.42

2.5-кестеде ұсынылған алгоритм және базалық CNN моделі үшін есептелген өлшемдер (Accuracy, Recall, Precision, Specificity және F1-Score) көрсетілген. Алынған нәтижелерге сәйкес, ұсынылған тәсіл жіктеу дәлдігін (Accuracy) 6.52% -ға, толықтығын (Recall) 3.53% -ға, болжам дәлдігін (Precision) 9.78% -ға, ерекшелік көрсеткішін (Specificity) 9.22% -ға, сондай-ақ F1-Score мәнін 6.82% -ға артқанын байқауымызға болады. Зерттеу нәтижелері күрделі

жарықтандыру жағдайларында объектілерді анықтау сапасын арттыру үшін Retinex әдісіне негізделген адаптивті кескінді жақсарту алгоритмін интеграциялау тиімді екенін дәлелдейді.

Кесте 2.5. Ұсынылған алгоритм мен базалық CNN моделі үшін есептелген метрикалар (Accuracy, Recall, Precision, Specificity және F1-Score).

Бағалау көрсеткіші	Retinex+CNN (ұсынылған алгоритм)	CNN (базалық)	Жақсаруы (%)
Accuracy	0.98	0.92	+6.52
Recall	0.88	0.85	+3.53
Precision	0.99	0.91	+9.78
Specificity	0.99	0.90	+9.22
F1-Score	0.94	0.88	+6.82

Технологиялардың дамуы бет тану жүйелерін қарапайым әрі жылдам етті. Соның арқасында түрлі қолдану салаларында кеңінен тарала бастады. Қазіргі кезде цифрлық технологияларды пайдалану қоғамда қалыпты үрдіске айналды.

Ескере кететіні, бет тану технологиясын қолдану заңды және қауіпсіз болуы тиіс. Адам құқықтарын сақтау отырып қолданылуы жөн. Қауіпсіздік әрине маңызды, бірақ ол технология дамуына кедергі келтірмеуі қажет. Сондықтан бұл жүйелердің қалай жұмыс істейтінін және қайда қолданылатынын түсіну өте орынды.

Зерттеу нәтижелері; эксперимент нақты көрсеткіштері Retinex әдісіне негізделген алгоритмнің тиімді екенін дәлелдеді. Ол жарықтың біркелкі емес әсерін азайтып, кескіннің контрастын тым жақсы тазалайтынын көрсетті. Нәтижесінде конволюциялық нейрондық желі мен Retinex алгоритмі арқылы объектілерді тану сапасы артатынына көз жеткіздік. Айта кететіні, әдістің тиімділігі параметрлерді дұрыс баптауға байланысты. Сонымен бірге өңдеу уақытының да артқаны байқалды, бұл алгоритмді әлде де жетілдіру қажет екенін көрсетеді. Болашақта жүйе жұмысының нақты уақыттағы жылдамдығын одан әрі жоғарлату мақсатында бұл әдісті конволюциялық нейрондық желі моделіне тікелей енгізу жоспарға қойылды. Ұсынылған тәсілді бейнебақылау, қауіпсіздік салаларында жарық күрделі жағдайларда қолдануға болады.

2.3 Терең оқытуға негізделген бет-әлпетті танудың көпкезеңді моделі

Нақты қолданбалы есептерде терең оқыту әдістерін пайдалану кезінде кездесетін негізгі қиындықтардың бірі – деректердің шектеулілігі. Нейрондық желілерді тиімді оқыту үшін үлкен көлемдегі деректер қажет екені белгілі. Тәжірибеде мұндай үлкен көлемді деректерді жинау мүмкін бола бермейтіні кездеседі.

Мультимодальды деректермен жұмыс істейтін есептерде деректердің жеткіліксіздігін байқауға болады. Мұндай жағдайда әртүрлі дереккөздерден алынған ақпаратты (мысалы, кескіндер, мәтіндер, уақыттық қатарлар) біріктіру қажет екен. Шектеулерді болдырмас үшін деректерді көбейту, трансферлік

оқыту және генеративті-қарсылас желілерді пайдалану арқылы синтетикалық деректерді генерациялау әдістері жиі қолданылады. Мысалға алсақ, TriChronoNet моделі мультимодальды деректерді біріктіру арқылы электр энергиясының бағасын болжауға арналған тәсілді ұсынып отыр. Бастапқы деректер көлемі шектеулі болғанымен де жоғары тиімділікті көрсетеді.

Бетті тану жүйелерін әзірлеу [125]–[127] компьютерлік көру мен жасанды интеллект салаларындағы кеңінен зерттеліп жүрген бағыттардың бірі екені белгілі. Қауіпсіздікке және тұлғаны сәйкестендіруге қойылатын талаптар артып отыр. Бетті тану жүйелері қолжетімділікті бақылау, қоғамдық орындардағы бейнебақылау және дербестендірілген сервистер мен қосымшаларда қолданылады [128]–[132]. Осыған байланысты бетті тану әдістерін дамыту қажеттілігі туындайтыны өзекті. Соңғы жылдары терең оқыту технологияларын [133]–[135], бет бейнесін векторлық сипатта ұсынатын (face embeddings) тәсілдері [136], [137] пайдаланылып келеді.

Face embeddings – адамның бет кескініндегі бірегей ерекшеліктерін сандық түрде сипаттайтын векторлық көрініс. Тәжірибеде мұндай векторлық көріністер бет бейнесін ықшам әрі ақпараттық тұрғыдан тиімді сандық мәндер жиынтығы түрінде көрсетуге мүмкіндік беретіні кездеседі. Осы тәсіл машиналық оқыту мен терең нейрондық желілер әдістерін қолдану арқылы тану мен сәйкестендіру процесін жеңілдетіні анық.

Векторлық ұсыныстар бет кескіндерін талдау мен салыстыруға ыңғайлы сандық форматқа түрлендіруде маңызды рөл атақарады. Бұл тәсіл әсіресе бір кадрда бірнеше адамды қатар тану қажет болатын көпбағынды жүйелерде тиімді екені белгілі болды.

Faster Region – Convolutional Neural Network (Faster R-CNN) [138, 139] сияқты бетті танудың заманауи әдістері конволюциялық нейрондық желілердің (CNN) архитектураларын қолданады. Архитектуралар кескіндерді өңдеу дәлдігі мен жылдамдығын едәуір арттырып отыр.

Бұл жұмыста нақты уақыт режимінде бетті тану жүйесінің тиімділігін арттыруға арналған Self-Attention механизмімен біріктірілген әртүрлі бэкбондарды (ResNet50, MobileNetV3, EfficientNetB0) пайдалануы қарастырылған.

Архитектуралар танудың жоғары дәлдігін қамтамасыз етіп қана қоймай, жүйені әртүрлі жарық жағдайларына, түсіру бұрыштарына және бейне тізбектеріндегі өзгеретін сценарийлерге бейімдеуге мүмкіндік беріп отыр.

Мақалада [140] CDVGG-16 терең нейрондық желісін қолдана отырып, ым тілін танудың жаңа әдісі қарастырылған. Әдіс негізінде бейнекадрларды өңдеу арқылы ым қимылдарды нақты уақыт режимінде жіктеу мәселесі шешілетіні анықталды. Процесс үш кезеңнен тұрады:

- деректерді жинау;
- кескінді алдын ала өңдеу;
- жіктеу.

VGG-16 архитектурасы конволюциялық және максималды біріктіру қабаттарынан құралып отырғаны көрінеді. Бұл қимылдың белгілерін тиімді

түрде бөліп алуға мүмдік береді. VGG-19 архитектурасынан айырмашылығы бар. Модель қайта оқыта алу мүмкіндігін азайтады екен.

Әдіс 95,6% дәлдікпен, 99,9% толық болуымен және жылдам жұмыс істеу уақытымен (0,125с) жоғары нәтижелерді көрсетті. Бұл оның қосымшаларына тілін тану жарамды екенін көрсетеді.

Мақалада [141] спорттық жаттығуларды жақсарту үшін бейне кескіндерді өңдеу технологиялары қарастырылып отыр. Және бөлшектер ройын оңтайландыру алгоритміне қолдану жасау қарастырылып отыр.

Жаттықтырушылардың визуалды түрде бағалауға негізделген дәстүрлі әдістердің тиімділігі төмен болды. Бұл спортшылардың дайындық деңгейінің өсуін шектеп отыр. Бейнені өңдеуді қолдана отырып ұсынылған қозғалысты тану жүйесі спортшылардың қозғалыс траекторияларын талдауға және эталондармен салыстыруға мүмкіндік беріп отыр. Мүмкіндік жаттығу процесін жақсартуға көмектеседі.

Бөлшектер ройын оңтайландыру алгоритмі қозғалыстарды тануды оңтайландыруда қолданылады. Ол спорттық талдау технологияларының дамуына ықпал етеді.

Мақалада [142] бейнеде адамның әрекеттерін тануға арналған динамикалық PSO-ConvNet жаңа моделі ұсынылған. Модель конволюциялық нейрондық желілерді (ConvNets) және бөлшектер ройы алгоритмін (PSO) біріктіреді. Бейнедегі әрекеттерді классификациялау дәлдігін арттыру үшін Transformer және рекуррентті нейрондық желілер (RNN) сияқты уақытша деректерді өңдеу әдістерімен интеграциялау пайдаланылады. UCF-101, Kinetics-400 және HMDB-51 деректер жиынтықтарында жүргізілген эксперименттер тану дәлдігі 9%-ға дейін артқанын көрсетті. PSO-ConvNet моделі тану сапасын жақсарта отырып, адам әрекеттерінің кеңістіктік-уақыттық динамикасын тиімді түсіреді.

Мақалада [143] бейне деректер негізінде бет эмоцияларын тануға (VFER) арналған үшөлшемді конволюциялық желіні (3D-CNN) және LSTM (ConvLSTM) моделін біріктіретін гибриді нейрожелілік архитектурасы ұсынылды. Модель эмоциялар тізбегінің бейнелерінен кеңістіктік-уақыттық ақпаратын тиімді түсіреді және SAVEE, CK+, AFEW деректер жиынтықтарында жоғары дәлдікті көрсетеді. Дәстүрлі модельдерден айырмашылығы ұсынылған архитектура аз параметрлерді пайдаланып, орындау жылдамдығын арттырады. Оны ресурстары шектеулі құрылғыларда нақты уақыт режимінде қолдануға мүмкіндік береді.

Мақалада [144] түрлі белгілерді біріктіруге негізделген бейнеде адам әрекеттерін тануға арналған (HAR) жаңа әдіс ұсынылды. Камера қозғалысы, адамның қозғалысындағы тұрақсыздық және күрделі фон мәселелерін шешу үшін көру бұрышына тәуелсіз тұрақты дескриптор әзірленіп отыр.

Қозғалысты талдау үшін әдіс оптикалық ағынды (Optical Flow, OF) және бағыттала алатын градиенттер гистограммасын (Histogram of Oriented Gradients, HOG) қолданады. Пішінді сипаттау үшін жергілікті үштік паттерндерді (Local Ternary Patterns, LTP) және Зернике моменттерін (Zernike Moments, ZM) пайдаланады екен. Соңғы векторды Бұл белгілердің бірігуі құрап отыр.

IXMAS, CASIA және TV-HI деректер жиынтығында өтетін тәжірибелер 98,66%-ға дейін жоғары дәлдікті көрсеткен. Нәтижелер ұсынылған тәсілдің тиімділігін дәлелдей келе, қолданыстағы әдістерден асып түсетінін көрсеткен.

Компьютерлік көру саласында Vision Transformers (ViTs) [145] кескіндерді өңдеуде назар механизмін қолдануға негізделген заманауи тәсілдердің бірі ретінде қарастырылып отыр.

Әдістер кескіндердегі жаһандық тәуелділіктерді ескереді. Кейбір есептерде дәстүрлі конволюциялық нейрондық желілермен (CNN) салыстырғанда жоғары нәтижелер береді. Мысал ретінде Modality Fusion Vision Transformer [146] зерттеуінде гиперспектралды және лидар деректерін біріктіріп классификациялау үшін ViT қолданылғанын келтіруге болады.

Зерттеу нәтижелері модельдің мультимодальды деректер арасындағы байланыстарды тиімді өңдей алатынын және жоғары классификация дәлдігіне қол жеткізетінін көрсетіп отыр.

Evaluation of Vision Transformers зерттеуінде жол белгілерін классификациялау тапсырмасында ViT модельдері нақты уақыт режиміндегі күрделі жағдайларда да жоғары нәтиже көрсететіні жан жақты сипатталған.

Ұсынылған тәсілдің бір ерекшелігі терең оқыту модельдерін оқыту кезеңінде бет кескінінің векторлық ұсыныстарын (face embeddings) біріктіруде.

Жүйе көпағынды тану міндеттерін орындауға мүмкіндік беріп отыр. Модельдердің әр түрлі оқыту кезеңдеріндегі дәлдігі мен тұрақтылығы талданды. Нәтижелердің ұсынылған әдістердің нақты пайдалану жағдайларында қаншалықты тиімді екені бағаланды. Тәжірибелік жарамдылығы туралы қорытынды жасауға мүмкіндік беруге болады.

Жұмыстың негізгі мақсаты – бейне ағындардағы кескіндерді нақты уақыт режимінде тануға арналған теңгерімді тәсілді әзірлеу мен тиімділігін талдау.

Алынған нәтижелерді бет-әлпетті тану жүйелерін жетілдіруге қолдануға болады екен. Бұл тәсіл әртүрлі салаларда қауіпсіздік пен дербестендірілген қызметтерді дамытуда құнды болып табылады

Бетті және қозғалыстағы объектілерді тану жүйелерінде жетістіктерге қол жеткізілгенімен, қазіргі әдістердің жарық жағдайының өзгеруі, күрделі фон және нақты уақыт режимінде өңдеу сияқты факторларға байланысты шектеулері бар. Бұл қауіпсіздік пен бейнебақылау жүйелерінде өзекті мәселенің бірі болып отыр. Себебі мұндай жүйелер есептеу ресурстары шектеулі болатын құрылғыларда орындалады. Модельдерді мобильді және ендірілген құрылғыларға бейімдеу есептеу кезіндегі күрделі болуына байланысты қиындықтар туғызады. Бұл олардың тиімді болуын төмендетіп отыр.

Аталған мәселелер терең оқытудағы әдістерді қолдану арқылы қарастырылды. Faster R-CNN архитектурасы мына берілген ResNet-50, MobileNetV3 Large, EfficientNetB0 базалық модельдерімен Self-Attention механизмі арқылы біріктіріле отырып қолданылады. Тәсілдің күрделі жағдайларда объектілерді анықтау мен тану жүйелерінің дәлдігі мен тұрақтылығын арттыруға мүмкіндік беретіні анықталып отыр.

Face embeddings – бет кескінін сандық вектор ретінде ұсыну тәсілі. Әрбір вектор адамның бетіне тән бірегей белгілерді сипаттайтыны төмендегі 2.7-суретте көрсетілген.

$$f(\text{face image}) = \begin{pmatrix} 0.112 \\ 0.067 \\ 0.091 \\ 0.129 \\ 0.002 \\ 0.012 \\ 0.175 \\ \vdots \\ 0.023 \end{pmatrix}$$

Сурет 2.7. Бет кескінін вектор ретінде ұсыну

Бетті сәйкестендіру міндеттерінде эмбеддингтер бет кескінін талдауға және салыстыруға ыңғайлы пішінге айналдыруда шешуші рөл атқаратыны көрінеді. Бет эмбеддинг деп көп өлшемді кеңістіктегі векторды айтады (мысалы, 128 немесе 512 өлшем). Мұнда вектордың әрбір элементі беттің белгілі бір ерекшеліктерін сандық түрде сипаттайтыны көруге болады. Ерекшеліктерге көздің арақашықтығы, мұрынның пішіні, еріннің өлшемі және әр бетке тән басқа геометриялық параметрлер сияқты бет құрылымы жатады. Терінің текстуралық сипаттамалары, соның ішінде, әжімдер, мендер мен тері тесігі де ескеріліп отыр. Эмбеддинг ішінара мимика туралы ақпаратты қамтуы мүмкін. Дегенмен, басты міндет мимкаға тәуелді емес сипаттамаларды сақтау. Жүйе жарық пен түсіру бұрышының өзгеруіне төзімді болуы тиіс. Бұл жағдайда face embeddings бет кескінінің тұрақты белгілерін бөліп көрсету арқылы мұндай өзгерістердің әсерін азайтады екен.

Бет эмбеддингтері бет кескіндерінің үлкен деректер жиынтығында оқытылған нейрондық желілерді пайдалану арқылы жасалатыны көрінді. FaceNet сияқты желілер беттің кіріс кескінін эмбеддингке түрлендіретін конволюциялық нейрондық желілердің (CNN) құрылымын пайдаланып отыр.

Жұмыста бет эмбеддингтерін алу үшін Inception-ResNet V1 архитектурасына негізделген, VGGFace2 деректер жиынтығында алдын ала оқытылған FaceNet моделі қолданылған.

Деректерді алдын ала өңдеу барысында нормализациялау, кескін өлшемін 160×160 пиксельге дейін өзгерту және MTCNN алгоритмі арқылы бет аймағын анықтау қадамдары орындалып отыр. Модельді оқыту кезінде негізгі гиперпараметрлер ретінде оқыту жылдамдығы 0,001, Adam оптимизаторы және 32- 128 аралығындағы батч өлшемдері қолданылғанын көрсетеді.

Процесс келесі түрде сипатталады екен: әуелі бет бейнесінің кіріс кескіні X нейрондық желінің бірнеше конволюциялық қабаттары арқылы өңделеді. Осы кезеңде бет құрылымы, текстурасы және басқа да сипаттамаларды көрсететін белгілер карталары (feature maps) қалыптасады. Ол төмендегі (2.29) формуламен берілген:

$$F = CNN(X) \quad (2.29)$$

мұнда CNN – конволюциялық нейрондық желі; F – желі арқылы алынған белгілер жиынтығы.

Келесі кезеңде алынған белгілер толық байланысқан қабат арқылы өңделіп, нормализациядан өтеді. Сосын тұрақты ұзындықтағы вектор түрінде ұсынылады. Алынған вектор бет эмбеддингі ретінде қарастырылатыны келесі формулада (2.30) көрсетілген:

$$E = \text{Normalize}(FC(F)) \quad (2.30)$$

мұнда FC – толық байланысқан қабат;

Normalize – векторды нормализациялау операциясы.

Нормализация модельдің жарықтық, контраст және басқа да кескін параметрлерінің өзгерістеріне төзімділігін арттыруға көмектеседі екен.

Эмбеддингтер әртүрлі беттерді тиімді салыстыра алады. Жаңа кескін алынғанда жүйе сол бетке сәйкес эмбеддингті құрады. Ол дерекқорда сақталған эмбеддингтермен салыстырылады. Егер екі эмбеддинг арасындағы қашықтық белгіленген шектен төмен болса, сәйкес беттер бір адамға тиесілі деп танылады екен.

Тәжірибеде мұндай салыстыру көбінесе Евклид қашықтығы арқылы жүзеге асырылатынын төмендегі формуладан (2.31) көрінеді:

$$D(E_1, E_2) = \sqrt{\sum_{i=1}^n (E_1[i] - E_2[i])^2} \quad (2.31)$$

мұндағы n – эмбеддинг векторының өлшемі.

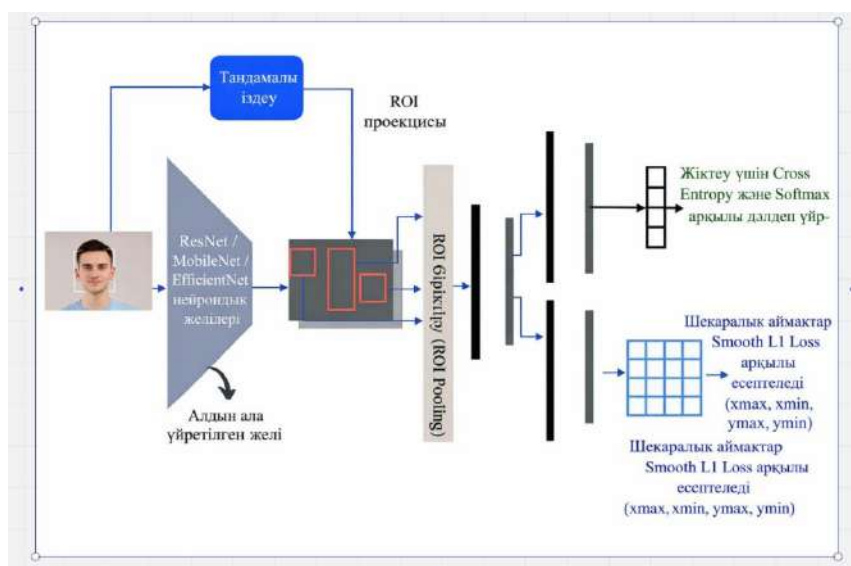
Эмбеддинг сәйкестендіру жүйелерінің дәлдігі мен сенімділігін едәуір арттыруға мүмкіндік береді. Әсіресе бірнеше бетті бір уақытта тану қажет болатын көп ағынды режимде. Эмбеддинг жарықтандыру және бастың орналасуы сияқты түрлі факторлардың әсерін азайтады. Деректерді өңдеудің жоғары жылдамдығын қамтамасыз етеді. Бұл эмбеддингтерді Fast-RCNN терең оқыту модельдерінде пайдалану бетті тану дәлдігі мен жылдамдығын арттыруға мүмкіндік береді. Бірнеше бет бір кадрда қатар кездесетін күрделі жағдай кезінде олардың тиімділігі жоғары.

Жүйені оқыту кезіне арналған белгілер ретінде мобильдік телефон нөмірлері пайдаланылып отыр. Ақпараттық жүйелерде пайдаланушыны тіркеу, аутентификациялау және байланыс орна алу үшін қолданылатыны белгілі. Пайдаланушының жеке телефон нөмірлері жеке деректердің бір бөлігі болып деректер қорында сақталады. Оны сәйкестендіру процесінде қолданады. Бет кескінін бірімәнді түрде нақты адаммен байланыстыруға мүмкіндік береді. Автоматтандырылған сәйкестендіре алатын жүйелерінің тиімділігін арттырады.

Бұл жұмыста бетті тану модельдерін құру мен оқытудың әртүрлі тәсілдері қарастырылады. ResNet және MobileNet көмегі арқылы Faster-RCNN сияқты архитектураларды қолдану. Әр тәсілдің артықшылықтары, кемшіліктері мен нақты жағдайларда қолданыла алуы бағаланған. Оқыту процесінде модельдердің дәл болуы мен жоғалып кетуіне талдау жасалып отыр. Олардың жалпы өнімді болуы мен нақты берілген мәселелерді шешуге жарамды болуы туралы қорытындылау мүмкін болады.

Жұмыста Faster R-CNN архитектурасы ұсынылды. Архитектура кескіндердегі объектілерді тиімді түрде анықтау мен жіктей алу үшін конволюциялық нейрондық желілердің (CNN) әдістері мен қызығушылық

аймақтарын бөлу алгоритмдерін (ROI – Region of Interest) біріктіргені келтірілді. Біріктіру 2.8-суретте көрсетілген.



Сурет 2.8. Faster R-CNN архитектурасы

Faster R-CNN жұмысының Region Proposal Network (RPN) немесе бастапқы нұсқаларда Selective Search алгоритмі арқылы қызығушылық аймақтарын генерациялаудан басталатыны байқалады. Бұл алгоритм кескіннің ішінде объектілер болуы мүмкін аймақтарды анықтауға жауап береді екен. Анықталған ықтимал аймақтар әрі қарай таңдалған конволюциялық нейрондық желіге беріліп, олардан белгілер шығарылып отыр.

Бұл архитектурада осы мақсат үшін ResNet-50 FPN, MobileNet V3 Large немесе EfficientNetB0 модельдерінің бірін пайдалану ұсынылады. Модельдердің әрқайсысының белгілі бір сценарийлерде қолданылуын анықтайтын өзіндік күшті жақтары бар.

ResNet-50 FPN – 50 қабаттан тұратын, қалдық блоктарды қолданатын терең нейрондық желі.

Архитектура үлкен деректер жиынтығында тиімді оқытылуымен ерекшеленіп отыр. Feature Pyramid Network (FPN) көмегімен ResNet-50 әр түрлі масштабтағы белгілерді өңдей алады. Бұл әртүрлі өлшемдегі объектілерді анықтау үшін маңызды болып табылады.

MobileNet V3 Large – мобильді құрылғылар мен ендірілген жүйелер сияқты шектеулі есептеу ресурстары бар ортада қолдануға арналған.

Модель тереңдік конволюцияларын пайдаланады. Параметрлер санын және есептеу күрделілігін азайтып, дәлдікті сақтау мүмкіндігін беріп отыр.

Self-Attention механизмі қосылған EfficientNetB0 моделі – белгілерді алу тәсілі.

Әртүрлі backbone-модельдерді таңдау мүмкіндігі Faster R-CNN архитектурасын жоғары дәлдікті талап ететін медициналық жүйелерден жылдам мобильді шешімдерге дейін бейімдеуге мүмкіндік беретінін көрсетіп отыр. Faster R-CNN сияқты әртүрлі backbone-модельдерге (ResNet, MobileNet) негізделген

терең оқыту жүйелерінің математикалық модельдерін құру желілердің жұмыс принциптерін түсіну үшін керек болады. Процесс математикалық аппаратты меңгерумен тікелей байланысты.

Faster R-CNN компоненттеріне белгілерді бөліп алу, аймақтарды генерациялау желісі, ROI Pooling, объектілерді жіктеу және олардың координаттарын регрессиялау кіреді.

Белгілерді бөліп алуда ResNet немесе MobileNet тәрізді конволюциялық нейрондық желілер қолданылды.

Кіріс кескіні X – өлшемі $H \times W \times C$ болатын тензор түрінде беріледі. Мұнда:

H – кескіннің биіктігі;

W – ені;

C – түстік арналар саны.

Бірнеше конволюциялық қабаттардан өтіп, кескін өлшемі $H' \times W' \times D'$ болатын белгілер картасына түрленеді (2.32):

$$F = CNN(X) \quad (2.32)$$

Faster R-CNN құрылымындағы Regional Proposal Network (RPN) алынған белгілер картасын F сканерлейді. Объектілер орналасуы мүмкін аймақтарды ұсынады. Ұсынылған аймақтар жиыны (2.33):

$$\{P_i\} = RPN(F) \quad (2.33)$$

RPN белгілер картасының әрбір позициясы үшін бірнеше аймақ-кандидаттарды (region proposals) генерациялайды. Әрбір аймақ (x_i, y_i, w_i, h_i) параметрлерімен сипатталады. Мұндағы:

- (x_i, y_i) – аймақ орталығының координаттары,
- w_i – аймақтың ені,
- h_i – аймақтың биіктігі.

Ұсынылған аймақтар кейіннен жіктеу мен регрессия кезеңдеріне сәйкес келетін тұрақты өлшемге келтіріледі.

Аймақтар анықталғаннан кейін, әрқайсысы RoI Pooling (Region of Interest Pooling) кезеңінен өтеді екен. Процесс барлық аймақтарды кейінгі жіктеу мен регрессия кезеңдеріне сәйкес келетін бірдей өлшемге келтіреді.

Алынған аймақтар негізінде объектінің класын анықтау үшін жіктеу (classification) орындалады. Ал шектейтін рамканың (bounding box) орналасуы мен өлшемдерін нақтылай алу үшін регрессия орындала отырады (2.34, 2.35):

$$p_i = \text{softmax}(W_{cls} * f_{ROI}(F, P_i)) \quad (2.34)$$

$$(t_x, t_y, t_w, t_h) = W_{reg} * f_R(F, P_i) \quad (2.35)$$

мұндағы p_i – аймақтың белгілі болған бір класқа тиесілі болу ықтималдығын, (t_x, t_y, t_w, t_h) , bounding box өлшемдері мен орналасуын нақтылау үшін арналған регрессияның параметрлері. Faster R-CNN моделі конволюциялық нейрондық желілердің мүмкіндіктері мен объект орналасуы мүмкін болған аймақтарды ұсына алу механизмін (region proposals) біріктіре отырып, кескіндердегі объектілерді тиімді анықтай отырып, оларды дәл жіктеуге мүмкіндік бере алады.

ResNet терең конволюциялық нейрондық желі қатарына жатады. Ол терең модельдерді оқыту кезінде пайда болатын градиенттің жоғалып кетуін

болдырмау мәселесін шешуге арналған резидуалды блоктарды қолданады алады. ResNet архитектурасының негізгі идеясының бірі – резидуалды байланыстарды (residual connections) пайдалана білу, яғни кіріс деректерін қабатты пайдалану арқылы еш өзгеріссіз өткізу мүмкіндігі екен.

Желіде X_l – желінің l -қабатындағы кіріс сигнал, ал W_l – осы қабаттың параметрлері болсын дейік. Онда $(l + 1)$ -қабаттың шығысы келесі формуламен анықталады (2.36):

$$X_{l+1} = X_l + f(W_l * X_l) \quad (2.36)$$

мұнда $f(\cdot)$ – сызықтық емес активация функциясы (мысалы, ReLU), ал $*$ белгісі конволюция операциясын білдіріп отыр.

Резидуалды блоктар бір-бірімен тізбектеле байланыса отырып, терең нейрондық желіні құрайды. Желінің соңғы қабатындағы шығыс сигналдың жалпыланғандағы түрі төмендегі түрде беріледі (2.37):

$$X_L = X + \sum_{l=1}^L f(W_l * X_l) \quad (2.37)$$

Резидуалды байланыстардың арқасында градиент желі бойында жоғалмай таралады. Бұл ResNet-50 сияқты өте терең болатын нейрондық желілерді тиімді түрде оқытуға мүмкіндік береді.

MobileNet – мобильді құрылғылар мен ендірілетін жүйелер сияқты есептеу ресурстары шектеулі ортада қолдануға оңтайландырылатын конволюциялық жеңіл нейрондық желі. MobileNet архитектурасының негізгі ерекшелігінің бірі бөлінген тереңдетілген конволюцияларын (depthwise separable convolutions) қолдану. Бұл тәсіл модельдегі параметрлер саны мен есептеу күрделілігін әжептәуір азайтады.

Дәстүрлі конволюция операциясында кіріс кескіннің әрбір арнасы көп сүзгілер арқылы өңделе алады. Бұл есептеу кездегі шығындардың артуына әкеледі. Ал MobileNet архитектурасында бұл операция екі кезеңге бөлінеді. Алғашқы кезеңде кіріс кескіннің әр арнасы жекелеген сүзгі арқылы өңделіп отырады. Бұл процесс тереңдетілген конволюция (depthwise convolution) деп аталады және келесі түрде өрнектеледі (2.38):

$$F_i' = W_{depthwise} * F_i \quad (2.38)$$

мұнда $W_{depthwise}$ – тереңдетілген конволюция сүзгілерінің параметрлері, ал F_i' – алынған аралық белгілердің карталары.

Екінші кезеңде алынған аралық белгілер карталары 1×1 өлшемді конволюция (pointwise convolution) арқылы біріктіріле алады (2.39):

$$F_{output} = W_{pointwise} * F_i' \quad (2.39)$$

мұндағы $W_{pointwise}$ – 1×1 конволюциясының параметрлері.

Тереңдік және нүктелік конволюцияларды біріктіріп қолдану есептеу кезіндегі шығындарын айтарлықтай азайтады. MobileNet моделінің тиімді болуы мен өнімді болуын арттырады. Faster R-CNN, ResNet және MobileNet – объектілер мен бет бейнелерін анықтауда және тану есептерін шешуде жағдайында қолданылатын тиімді модельдердің бірі. Faster R-CNN объектілерді анықтау, жіктеу үшін аймақ-кандидаттарды генерациялау әдісімен (region proposals) қатар конволюциялық нейрондық желілерді қолдануға болады. Ал

ResNet және MobileNet архитектуралары белгілерді бөліп алу барысында қолданатын екі түрлі backbone-модель болады. Олардың әрқайсысының өзіндік артықшылықтары бар.

ResNet – резидуалды блоктарға негізделіп отырған терең нейрондық желі. Ол өте терең архитектураларды тиімді деңгейде оқытуға мүмкіндік береді. Және күрделі есептеулерді жоғары дәлдікпен есептеуге мүмкіндік береді.

MobileNet – мобильді құрылғылар мен ендірілген жүйелер үшін оңтайландырылатын ықшам нейрондық желі. Бұл модель есептеу шығындары мен дәл нәтиже ала алу арасындағы тиімді тепе-теңдікті жасайды.

Face embeddings әдісімен бірге қолданылған жағдайда, бұл модельдер бет бейнелерін көпәғынды режимде тану міндеттері кезінде жоғары тиімділік пен дәлдікті көрсетеді екен.

Ұсынылған архитектурада белгілерді бөліп алудың заманауи тәсілі Self-Attention механизмімен толықтырылатын EfficientNetB0 моделі интеграцияланған. EfficientNetB0 моделі желі параметрлерін – тереңдігін, ені мен кіріс кескінінің ажыратыла алатынын теңгерімді түрде масштабтау арқылы жоғары тиімділік беретіндігімен ерекшеленеді. Бұл тәсіл дәстүрлі модельдермен салыстырғанға қарағанда есептеу кезіндегі шығындарын азайта отырып, жоғары дәлдікке қол жеткізуде көп мүмкіндік береді.

Осы жұмыста Self-Attention қабатымен толықтырылатын модификацияланған EfficientNetB0 архитектурасы қолданыла алды. Бұл механизм модельге кескіннің маңызды деген бөліктеріне айтарлықтай көбірек назар аударады. Маңыздылығы жағынан төмен аймақтарды елемей жұмыс істеуге мүмкіндік бере алады. Нәтижесінде күрделі деген объектілер мен детальдары көптеген көріністерді тану кезінде жүйенің дәл болуы айтарлықтай артатын болады.

Self-Attention механизмі кескіннің неше түрлі бөліктері арасында жаһандық өзара байланыстыруларды тиімді ескеруге мүмкіндік бере алады. Бұл ең бастысы контекстік ақпарат маңызды болған жағдайда, бет бейнелерін тануда немесе үлкен көріністер фонында орналасқан шағын объектілерді анықтауда аса маңызды.

EfficientNet архитектурасы нейрондық желіні үш негізгі параметр бойынша масштабтай алу әдісін қолданады. Олар желінің ені, желінің тереңдігі мен кіріс кескінінің ажыратымды бола алуы. Модельдің негізгі идеясы бұл шағын конволюциялық блоктарды пайдалана отыру барысында, дәл болу мен есептеу кезіндегі шығындар арасындағы оңтайлы болатын тепе-теңдікті қамтамасыз ете отырып кеңейте алатын архитектура құру. EfficientNet моделінің негізінде күрделі деген есептерге бейімделе алуына мүмкіндік береді. Және есептеу жағдайындағы тиімділігін сақтай алатын теңгерімді масштабтау механизмі жатыр екен.

EfficientNet архитектурасының басты құрылымдалған элементіні ол MBConv блогы (Mobile Inverted Bottleneck Convolution). Бұл блок стандарттала алатын конволюцияларды, тереңдете алатын конволюцияларын, активацияның

функциялары мен нормализацияның механизмдерін біріктіре алатын операциялар жиынтығынан тұрады.

Егер X – кіріс кескін, онда MBConv блогының шығыс сигналы төмендегі өрнекпен өрнектеледі (2.40):

$$Y = FMBCConv(X) \quad (2.40)$$

мұнда $FMBCConv(\cdot)$ – MBConv блогының ішіндегі барлық операциялардың жиынтығын сипаттай алады. Яғни стандартты және тереңдік конволюцияларының комбинациясын, активациялау функциялары мен нормализация операцияларын қамтиды. MBConv блогы EfficientNet архитектурасының негізгі деген элементі желінің барлық деңгейлерінде қолданыла алды. EfficientNet моделінде желіні масштабтай алуда үш басты параметр арқылы жүзеге асырыла алады:

- тереңдік (α) – желі қабаттарының санын анықтай алады;
- ені (β) – әр қабаттағы арналар санын анықтай алады;
- ажыратымдылық (γ) – кіріс кескінінің өлшемін сипаттай алады.

Бұл параметрлер модельдің күрделілігі жағынан нақты есептің талаптарына сәйкес икемделуді реттеуге мүмкіндік береді. Модельді масштабтау кезеңінде пайдаланушы масштабтау коэффициентін k деп белгілейді. Ол желінің жалпы күрделілі болуын басқару жағдайында көп мүмкіндік береді.

Егер N MBConv блоктары қолданылған жағдайда, онда олардың нәтижесінде алынған белгілер картасының шығысы мына төмендегі өрнекпен сипатталады (2.41):

$$Output\ features = F_{MBConv}^{(N)}(\alpha, \beta, \gamma, X) \quad (2.41)$$

мұнда $F_{MBConv}^{(N)}(\cdot)$ – масштабтау коэффициенттерінің көмегінің арқасында параметрленген конволюция, активация мен нормализация операцияларының жиынтығын білдіре алады. Модельдің өнімді болуын арттыру мен кескіннің әртүрлі бөліктерінің арасындағы жаһандық ішінара байланыстарды ескеру мақсатында EfficientNet архитектурасына Self-Attention механизмі енгізілуі мүмкін болады. Осы механизм модельге кескіннің маңызды деген аймақтарына көбірек назар аударуға, белгілер картасының әр бөлігінің арасындағы жаһандық тәуелділіктерді ескеруге мүмкіндік бере алады. Self-Attention механизмі математикалық түрде төмендегі өрнекпен сипатталады (2.42):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (2.42)$$

мұндағы $Q = XW_q$ – сұраныстар (query) матрицасы, $K = XW_k$ – кілттер (key) матрицасы, $V = XW_v$ – мәндер (value) матрицасы, W_q , W_k , W_v – үйренілетін параметрлер матрицалары, d_k – кілт векторының өлшемі.

Self-Attention механизмі белгілер картасының әртүрлі бөліктері үшін салмақ коэффициенттерін есептейді. Модельдің маңызды элементтерге назар аудара алуына мүмкіндік береді. Self-Attention механизмімен толықтырылған EfficientNet моделі конволюциялық блоктар арқылы алынған локалды белгілерді біріктіреді. Және өзіндік назар механизмі арқылы алынған жаһандық

тәуелділіктерді біріктіре алады. Модельдің ең соңғы шығысы келесі түрде анықталады (2.43):

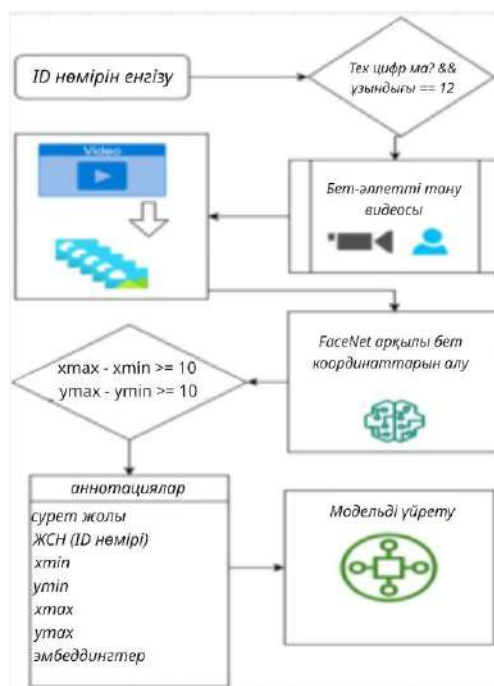
$$Z = \gamma * Attention(Q, K, V) + (1 - \gamma) * Y_{output} \quad (2.43)$$

мұндағы γ – Self-Attention механизмінің әсер ету деңгейін анықтайтын үйрене алатын параметр. Ал Y_{output} – конволюциялық блоктардың арқасында алынған локалды белгілер. EfficientNet архитектурасына Self-Attention механизмін енгізу кескіннен алатын белгілердің сапасын арттырады. Модельдің күрделі объектілер мен көріністерді тану дәлдігін жоғары қыла алады. Бұл тәсіл объектілерді анықтау мен тануды есептеу кезінде маңызды болып табуға болатын локалды және жаһандық ақпаратты бір мезгілде ескеруге мүмкіндік бере алады. Барлық үш модельді – Faster R-CNN (ResNet50 FPN backbone), MobileNetV3 Large және Self-Attention механизмімен толықтыруға болатын EfficientNetB0 – оқыту кезінде келесі гиперпараметрлер қолданылды:

- оқыту кезеңіндегі жылдамдығы (learning rate) – 0.005;
- оптимизатор – SGD (momentum = 0.9);
- bath өлшемі – 4;
- weight decay – 0.0005.

Параметрлер модельдердің тұрақты түрде жинақтала алуын (convergence) және объектілерді анықтау барысындағы есептерін шешу кезінде жоғары өнімділікті қамтамасыз ете алады.

Ұсынылған схема жеке тұлғаны тіркеудегі бет-әлпетті тану жүйесін әзірлеу кезеңдегі процестің негізгі кезеңдерінің рет ретін көрсетеді. Процесс жеке сәйкестендіру нөмірін (ID) енгізу кезеңінен басталады. Жиналған деректердің негізінде модельді оқыту кезеңімен аяқтала алады (2.9-сурет). Жүйенің әр кезеңіне бұл бөлімде толықтай сипаттама беріледі. Іске асыру әдістері, қолданылған әдістер мен алгоритмдерді таңдаудағы негіздемесі қарастырылады.



Сурет 2.9. Бетті тану процесінің кезеңдік реті

Бетті тіркеу процесі пайдаланушының жеке тұлғаның жеке сәйкестендіру нөмірін (ID) енгізіп бастайды. Жеке нөмір 12 таңбадан тұратын бірегей сандық код болады. Ол пайдаланушының жеке деректеріне қол сұғудың негізгі кілті ретінде қызмет етеді. Және екі негізгі критерий бойынша тексеріледі. Олар сандық форматқа сәйкес екендігі және ұзындығының дұрыс болуы. Тексеру сәтті өткеннен соң жүйе келесі кезеңге өтеді.

Келесі кезеңде жеке пайдаланушының бет бейнесі бар бейнежазба жазылады. Бейнежазбаның ұзақтығы шамамен 10 секундты құрайды. Осы уақытта жүйе 250-300 кадр көлеміндегі бет бейнелерін тіркей отырады. Мұндай деректер көлемі кескіндердің әртүрлі болуын қамтамасыз ету үшін қажет. Бұл кейінгі өңдеу және модельді оқыту кезі үшін маңызды болады. Бет бейнелері түрлі бұрыштардан және түрлі жарықтандыру жағдайларында алынған сайын, жүйенің тану дәлдігі мен сәйкестендіру дәлдігі артып отырады.

Бейнежазба аяқталған кезден кейін әрбір кадрдан бет аймағын анықтау үшін алдын ала оқытылып дайындалған FaceNet моделі қолданылады. FaceNet - бет бейнелерін тану міндеттеріне арналған арнайы модель. Ол кескінде бейнеленген бет аймағының шекарасын төрт координата арқылы анықтай алады: *x_{min}*, *y_{min}*, *x_{max}* және *y_{max}*. Бет бейнесінің өлшемдері жеткілікті болу мақсатында жүйе қосымша тексеру жүргізеді: *x_{max}* – *x_{min}* және *y_{max}* – *y_{min}* айырмалары кем дегенде 10 пиксель болуы керек. Бұл талап бет бейнесі тым кішірек болатын кадрларды автоматты түрде алып тастауға мүмкіндік бере алады.

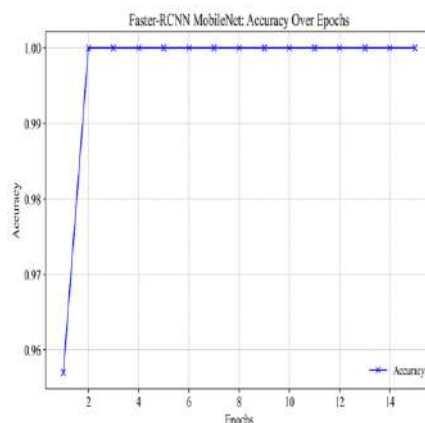
Жүйенің маңызды кезеңдерінің бірі – face embeddings генерациялау. Бұл процесс те FaceNet моделінің арқасында жүзеге асырылады. Эмбеддинг – бет бейнесінің жаппай ерекшеліктерін сандық түрде сипаттайтын көпөлшемді вектор болады. Мұндай векторлық ұсыныстар жүйеге бет бейнелерін жоғары дәлдікпен салыстыра алуға және жіктей алуға мүмкіндік береді. Эмбеддингтер түсіру жағдайларының өзгерістеріне, мысалға алғанда жарықтандырудың айырмашылығына, бас бұрышының өзгеруі немесе мимикаға төзімділікті қамтамасыз ете алады.

Бет аймағының координаттары мен эмбеддингтері алынғаннан соң, деректер аннотациялар түрінде сақтала алады. Олар кейін модельді оқыту процесі кезінде пайдаланылады. Аннотациялар құрамына кескінге апара алатын жол, бірегей сәйкестендірудің нөмірі (ID), бет аймағының шекара координаттары мен бет бейнесінің векторлық ұсынысы (эмбеддинг) кіреді екен. Деректер модельді оқыту кезінде әрбір эмбеддинг-векторды нақты сәйкестендіре алу нөмірімен байланыстыруға мүмкіндік бере алады. Нәтижесінде жүйе жаңадан қосылған кескіндердегі адамдарды дәл анықтай алады.

Модельді оқыту бастамас бұрын деректерді тазалау жүргізіледі. Бейнежазба барысында төмен сападағы кескіндер пайда болуы мүмкін. Мысалға алғанда бұлыңғыр сападағы суреттер немесе бет бейнесі толық көрінбейтін кадрлар. Мұндай кескіндер оқыту процесіне кері әсер етуі мүмкін. Себебі модель дұрыс емес белгілерді үйренуі ықтималдығы бар. Сондықтан деректер

жиынтығынан сапасы нашар кадрлар алынып тасталып, тек сапалы және оқытуға жарамды болатын кескіндер ғана қалдырыла алады.

MobileNetV3 Large негізіндегі Faster R-CNN моделі есептеудің ресурстары шектеу бар орталарда қолдануға арнайы бейімделген болады. Сондықтан бұл архитектура мобильді және ендірілген жүйелерде қолдануға қолайлы шешім болады. 2.10-суретте модельді оқыту процесіндегі бет бейнелерін жіктеудегі дәлдігінің өзгерісі көрсетілген. Графиктен модельдің аз уақытта оқу дәуірлерінің (epochs) өзінде жоғары жіктеу дәлдігіне тез қол жеткізуге болатынын байқауға болады. Бұл модельдің параметрлерінің саны аз және архитектурасы жеңілдетілген түрде болса да, ResNet50 сияқты күрделенген модельдермен салыстырғанға қарағанда бет бейнелерін тиімді жағдайда тануға қабілетті болатынын көрсетеді.

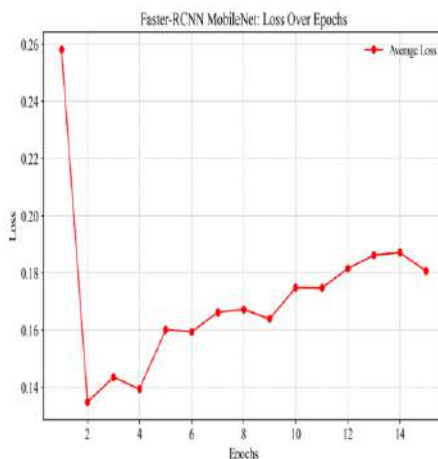


Сурет 2.10. MobileNetV3 Large негізінде Faster R-CNN моделінің дәлдігі

MobileNetV3 Large backbone архитектурасын қолдану барысында оқытылған Faster R-CNN моделінің оқу процесі де жоғары дәлдік көрсетеді. Алайда функция шығынының (loss function) динамикасы мен оқу барысының ResNet50 FPN негізіндегі модельден көбірек ерекшеленеді. Оқытудың бірінші кезеңінде (epochs) орташа шығын мәні 0.2581, ал жіктеу дәлдігі 95-70% болды. Бұл бастапқы шығынның мәні ResNet негізіндегі модельмен салыстырғанға қарағанда әлдеқайда жоғары. Себебі MobileNetV3 архитектурасы мобильді және алдын ала ендірілген жүйелерге бейімделген жеңіл модель болып табылады екен. Екінші кезеңде шығын функциясының мәні айтарлықтай төмендейді. 0.1348-ге дейін азайды. Ал модельдің дәл екені 100% деңгейіне жетті. Бұл модельдің тез жинақтала алатынын (convergence) және есептеу ресурстары шектеулі кезде де тиімді оқи білетінін көрсетеді. Келесі кезеңде шығын функциясының бір деңгейде ауытқу белгілілері байқалады. Үшінші кезеңде шығынның мәні 0.1435, ал бесінші кезеңде 0.1601 болды. Соған қарамастан 100% модельдің дәлдігі деңгейінде сақталды. Оқыту кезінде шығын функциясының мәні алтыншы кезеңдегі 0.1594 көрсеткішінен он төртінші кезеңдегі 0.1871 мәніне дейін өзгеріп отыр. Мұндай ауытқуларға қарамай модель жоғары дәлдікті сақтап қала алады. Бұл оның деректерді жақсы деңгейде жалпылай алатынын және деректер жиынтығындағы өзгерістерге салыстырмалы түрде тұрақты болатынын көрсетеді. Шығын функциясының өзгере алатындығы модельдің күрделі

деректерге сезімтал болатынын немесе белгілі бір деңгейде қайта оқытыла алуы (overfitting) ықтималдығын білдіруі әбден мүмкін. Сондықтан модельді ары қарай оңтайландыру мен қосымша талдау жүргізу қажет болады.

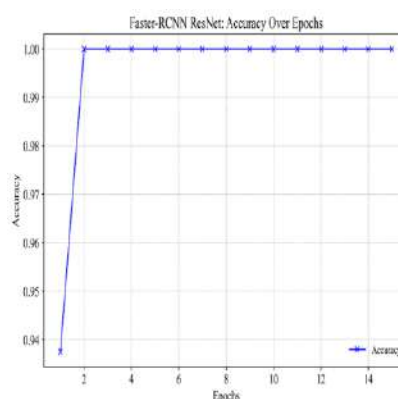
2.11-суретте MobileNetV3 Large backbone қолданған Faster R-CNN моделінің дәл қисығы көрсетіліп отыр. График модельдің дәлдігінің оқыту дәуірлерінің санына тәуелді болатын өзгерісін сипаттайды. Тік осьте дәлдіктің мәні, ал көлденең осьте оқытудың дәуірлері саны көрсетіледі. Графиктен модельдің алғашқы кезеңдерде айтарлықтай жылдам жақсарғанын және алғашқы екі дәуірден кейін 100% дәлдікке жеткендігін байқауға болады. Кейінгі дәуірлерде дәлдік тұрақты түрде осы деңгейде сақталады. Бұл модельдің ерте кезеңде-ақ тұрақты болатын күйге жететінін көрсетеді. Бұл жағдай модельдің жылдам жинақтай алу қабілетін көрсетеді. Ал бұл қасиет мобильді және ендірілген жүйелер сияқты есептеу кезіндегі ресурстары шектеулі ортада модельдерді оқыта алу үшін өте маңызды болады. Графикті оқыту дәуірлері артқан сайын шығын функциясының орташа мәнінің өзгеруін бейнелей отырады. Оқытудың бастапқы кезеңінде шығын шамамен 0.26 деңгейінде болады. Бірінші дәуірден кейін ол 0.14 мәніне дейін күрт төмендейтін болады. Кейінгі дәуірлерде шығын функциясы шамалы өсім көрсетеді. Бұл өзгерістер аз мөлшердегі ауытқулар шегінде қалатын болады. Мұндай динамика модельдің тез оқытыла алатынын және салыстырмалы түрде тұрақты күйге жете алатынын көрсетеді. Шығын функциясының аз мөлшерде өсуі модельдің белгілі бір деңгейде қайта оқытыла алуы мен күрделі деректерге сезімталдығын білдіруі мүмкін болады.



Сурет 2.11. MobileNet backbone қолданылған Faster R-CNN моделінің шығын жағдайындағы функциясының өзгеру графигі

ResNet50 FPN базалық архитектурасын қолданатын Faster R-CNN моделі объектілерді тану кезінде жоғары дәлдікті, тұрақтылықты көрсетті. Бұл нәтиже модельдің терең, күрделі архитектурасымен түсіндіріле алады. ResNet50 – 50 қабаттан тұратын конволюциялық терең нейрондық желі. Онда резидуалды блоктар қолданыла алды. Бұл блоктар терең нейрондық желілерді оқытуда жиі кездесетін градиенттің жоғалу мәселесін азайтуға мүмкіндік бере алады. Соның нәтижесінде модель үлкен деректер жиынтығында тиімді оқытады. Күрделі объектілерді тану кезінде де жоғары дәлдікті сақтай алады. Feature Pyramid

Network (FPN) архитектурасының ResNet моделімен біріктіріле алуы жүйенің мүмкіндіктерін одан әрі кеңейтеді екен. FPN әртүрлі масштабтағы белгілерді бөліп алуға болады. Бұл кескіндегі объектілер түрлі өлшемде, әртүрлі ракурста орналасқанда оларды дәлдікпен анықтауға көмектеседі. Мұндай күрделі архитектура есептеу ресурстарын айтарлықтай талап етеді. ResNet50 FPN негізіндегі Faster R-CNN моделін оқыта алу 183 минутты құрады. Бұл уақыт нейрондық желінің әр қабатында көп параметрлерді есептей алу қажеттілігімен және көпдеңгейлі белгілерді өңдей алу процесімен түсіндіріле алады (2.12-сурет). Оқыту уақытының ұзақ болуына қарамастан, модель жоғары дәлдік пен тұрақты болуды және деректерді жалпылай алу қабілетін көрсетті. Сондықтан архитектура жоғары сенімділік пен максималды дәл болуын талап ететін міндеттер үшін тиімді шешім болып табыла алады. Бұл ерекше қауіпсіздік жүйелерінде немесе медициналық кескіндерді талдау сияқты күрделі қолданбаларда маңызды болады. Мұндай жағдайларда объектілердің стандартты емес ракурстар болуында жиі кездеседі. Осындай сценарийлерде ResNet50 FPN архитектурасы қамтамасыз ете алатын белгілерді терең талдай алу мүмкіндігі ерекше рөл атқаратын болады.



Сурет 2.12. ResNet50 FPN негізіндегі Faster R-CNN модельдің дәлдік көрсеткіші

ResNet50 FPN базалық архитектурасын қолданатын Faster R-CNN моделін оқыту процесі барлық 15 оқу дәуірі (epoch) бойы тұрақты жоғары өнімділік көрсете алады. Бірінші дәуірде шығын функциясының орташа мәні 0.0922 болған болса, ал модельдің дәлдігі 93.75% деңгейіне жеткенін көрсетті.

Бұл көрсеткіш модельдің объектілерді тану қабілетінің бастапқы кезеңде-ақ жоғары болатындығын көрсетті. Келесі дәуірлерде шығын функциясының айтарлықтай төмендеу болатындығы көрінді. Екінші дәуірде шығын мәні 0.0261-ге дейін азайып, модель дәлдігі 100% деңгейіне жететіндігі анықталды. Мұндай нәтижелер модельдің тез жинақталатынын және салмақ коэффициенттерінің тиімді түрде реттелетінін көрсетеді.

Оқытудың келесі кезеңдеріде де шығын функциясының біртіндеп төмендеуі байқалатындығы анықталған. Модель дәлдігі барлық дәуірлерде 100% деңгейінде тұрақты сақталып отырды. Мысалы, үшінші дәуірде шығын мәні 0.0192, ал бесінші дәуірде 0.0152 болатындығын көрсетті.

Шығын функциясының біртіндеп азайып, он бесінші дәуірде 0.0105 минималды мәніне жетуі модельдің деректерді тиімді жалпылай алатынын және тану қателіктерін азайтатынын көрсетеді.

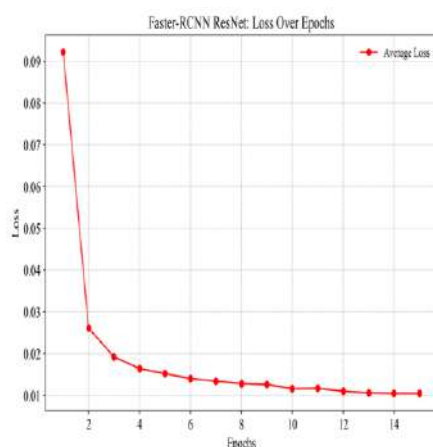
Модель максималды дәлдікке ерте жеткеніне қарамастан, оқыту процесі жалғасып, модель деректерге бейімделуін жалғастыратындығы көрінеді. Бұл кейінгі кезеңдерде шығын функциясының аздап жақсаруынан байқауға болады. Динамика ResNet50 FPN архитектурасының объектілерді анықтау міндеттерінде болатын жоғары тұрақтылығы мен сенімділікті көрсетіп отыр.

MobileNet жағдайындағыдай модель оқытудың бастапқы кезеңдерінде дәлдіктің тез өсуі болатынын көрсетеді. Бірінші дәуірден кейін дәлдік деңгейі 99% жетеді. Келесі дәуірлерде 100% шамасында тұрақты сақтала алатыны анықталды.

Нәтижелер ResNet архитектурасының объектілерді анықтау міндеттеріндегі жоғары тиімділігі мен модельдің жіктеу тапсырмасына тез бейімделе алатын қабілетін көрсетіп отыр. Сонымен қатар дәлдік қисығының тұрақты сипаты модельдің кіріс деректеріндегі өзгерістерге төзімді екенін көрсетеді.

Faster R-CNN моделінің шығын функциясының графигі 2.13-суретте ResNet backbone моделін қолданылғанын көрсеткен. Оқытудың бастапқы кезеңінде шығын функциясының мәні шамамен 0.09 болғандығын байқаймыз. Бұл MobileNet негізіндегі модельдің бастапқы шығын мәнінен төмен екендігі көрінеді. Мұны ResNet архитектурасының белгілерді тиімді бөліп алу қабілетімен түсіндіруге болады. Оқыту дәуірлері артқан сайын шығын функциясы жылдам төмендегені анықталып отыр. Үшінші дәуірден кейін шамамен 0.01 деңгейінде тұрақтанатындығы анықталып отыр.

MobileNet негізіндегі модельмен салыстырғанда, бұл жағдайда шығын функциясының кейінгі кезеңдерде өсуі байқалмайтындығын атап өткен жөн. Бұл ResNet архитектурасының деректерді жақсы жалпылау қабілетін және қайта оқытылуға (overfitting) аз бейімділігін көрсететін болып отыр.



Сурет 2.13. ResNet негізіндегі Faster R-CNN моделінің шығын функциясы

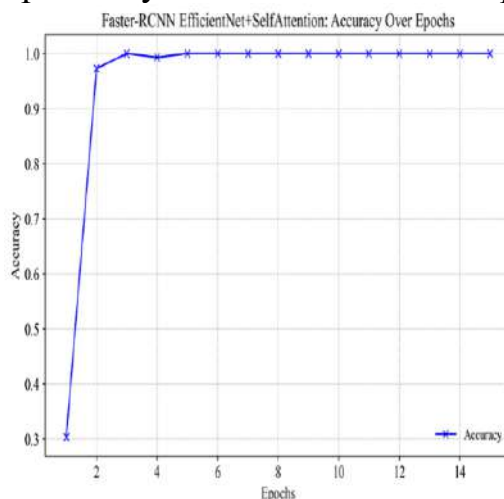
EfficientNetB0 архитектурасы мен Self-Attention механизмін біріктіре қолданатын Faster R-CNN моделін оқыту процесі жоғары тиімділік пен жылдам жинақталуды көрсететіндігін анықтадық. Бұл модель дәлдігі мен шығын

функциясының өзгерісін сипаттайтын графиктер арқылы расталатындығын атап өткен маңызды.

2.14-суретте оқыту дәуірлерінің саны артқан сайын модель дәлдігінің өзгеру динамикасы көрсетейік. Бірінші дәуірде модель дәлдігі 30,34% шамасында болған. Бұл салыстырмалы түрде алғандағы төмен көрсеткіш екені белгілі. Осы кезеңде модель тек оқыта алу процесін бастай отырып, деректерге бейімделе алатын кезеңінен өте алатыны аңғарылады.

Келесі дәуірлерде модельдің дәлдігі әлдеқайда өскен. Мысалы, екінші дәуірде модельдің дәлдік мәні 97,30% деңгейіне дейін жетті. Нәтижелер қолданып отырған архитектураның тиімді болатынын және модельдің деректерді тез меңгере алу қабілеттілігін көрсете алады.

Маңызды нәтижелердің бірі ретінде модельдің үшінші дәуірінде 100% дәлдікке жетуі арқылы байланыстыруды айтамын. Көрсеткіш оқытудың кейінгі барлық дәуірлерінде тұрақты түрде сақталып отырады. Модельдің сәтті оқытылғанын және деректердегі мүмкін болатын өзгерістерге қарамастан жоғары дәлдікпен объектілерді тануға қабілетті екенін көрсетті.

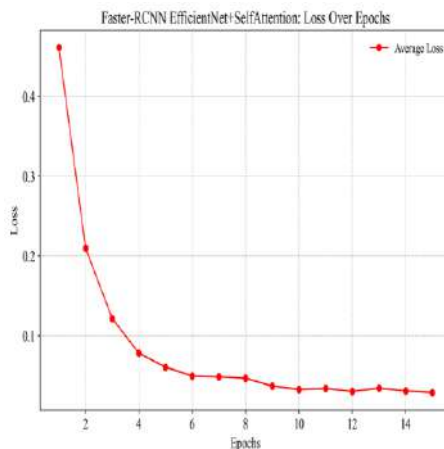


Сурет 2.14. Оқыту дәуірлерімен модельдің дәлдігінің өзгеру динамикасы

2.15-суретте 15 оқу дәуірінде шығын функциясының орташа мәнінің өзгеруі сатысы бейнеленген.

Бірінші дәуірде шығын функциясының орташа мәні 0.4611 болды. Оқытудың бастапқы кезеңі үшін қалыпты жағдайдағы көрсеткіш екені анық. Келесі дәуірлерде шығын функциясының айтарлықтай деңгейде төмендеуін байқауға болады. Мысалы, екінші дәуірде орташа шығынның мәні 0.2097-ге дейін азайды. Өзгеріс модельдің тиімді түрде оқытып жатқанын және танудың қателіктерінің азайып отырғанын көрсетеді. Үшінші дәуірде шығын функциясының мәні 0.1216-ға дейін төмендеді. Модель 100% дәлдікке дәл осы кезеңде жетті. Модельдің тиімді деңгейде оқытылғанына көз жеткізуге болады. Және жүйенің қателіктерді қысқарта ала алуы деңгейіне жеткенін көруге болады. Төртінші дәуірден бастап шығын функциясы бірте – бірте төмендей береді. Он бесінші дәуірде 0.0289 шамасында минималды мәніне жеткенін көрсетті. Модельдің оқыту процесі ары қарай жалғасады. Нәтижеленің барлығының біртіндеп жақсарып отырғаны байқалады. Өзгерістердің баяу жүріп отырғаны

белгілі. Бұл оқытудың соңғы кезеңдеріне тән келетін динамика. Себебі модель жоғары дәлдікке қол жеткізгеннен кейін, қалған қателіктерді азайту процесі жүреді.



Сурет 2.15. Шығын функциясы мәнінің өзгеру динамикасы

Әрбір оқу дәуірінің оқыту уақытына да назар аударылуы керек. Бірінші дәуірдегі модельді оқыту шамамен 28 минут уақытты алып отыр. Мұндай архитектураларға орташа көрсеткіш болып саналады.

Кейінгі дәуірлерде оқыту уақыты осы деңгейде сақталды. Алайда 13-ші және 14-ші дәуірлерде оқыту уақыты айтарлықтай өсті. Сәйкесінше 45 және 52 минутты құрайтынын көрсетті. Оқыту уақытының артуы негізінен Self-Attention механизмінің есептеу кезінде күрделілігінің жоғары болуымен түсіндіруге болады. Механизм белгілер арасындағы жаһандық тәуелділіктерді өңдеу үшін есептеудің қосымша ресурстарын талап ететінін атап өту керек.

Self-Attention алгоритмі квадраттық есептеу күрделілігіне ие болғандықтан, деректер арасындағы байланыстарды терең түрде талдау кезеңінде есептеу шығындары арта түсетіндігі белгілі. Оқыту процесін оңтайландыруда бірнеше тәсілді қарастыруға болады. Мысал ретінде соңғы оқу кезеңдерінде кіріс деректерінің ажыратымды болуын азайту немесе Self-Attention механизмінің әлдеқайда тиімдірек нұсқаларын қолдануды келтіруге болады.

Уақытының артуы шығын функциясын минимизациялаудағы күрделілігімен және модельдің деректердегі күрделі жағдайларға бейімделе алу қажеттілігімен де байланысты болуы мүмкін. Оқыту уақытының айтарлықтай артуына қарамастан модель жоғары дәлдікті сақтап, шығын функциясының мәнін төмендетуді жалғастырып отырады. Бұл модельдің тұрақты бола алатынын және күрделі деректермен тиімді түрде жұмыс істеу қабілетін көрсетіп отыр.

Осылайша EfficientNetB0 архитектурасы мен Self-Attention механизмі қолданылған Faster R-CNN моделі объектілерді анықтау міндеттерінде жоғары нәтижелер көрсетті.

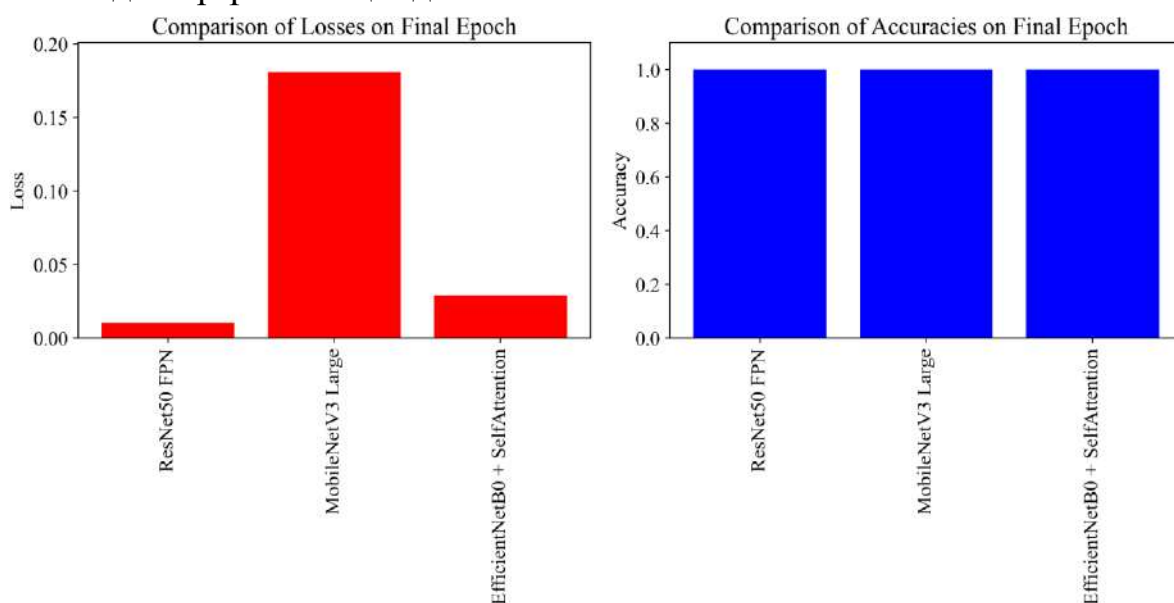
Модельдің жылдам жинақталуы, жоғары дәлдігі және шығын функциясының тұрақты түрде төмендеуі таңдалған архитектураның тиімді екенін көрсетті. Self-Attention механизмі модельге кескіннің жаһандық

контекстін тиімді ескеруге және маңызды белгілерге назар аударуға мүмкіндік береді. Бұл жүйенің жалпы өнімділігін арттырып қана қоймай, оны күрделі тану міндеттері үшін тиімді құралға айналдыратындығын аңғаруға болады.

2.16-суретте ResNet негізіндегі Faster R-CNN, MobileNetV3 Large және Self-Attention механизмімен толықтырылған EfficientNetB0 модельдерінің нәтижелері салыстырмалы түрде келтірілген. Сол жақтағы графиктен 15-ші дәуірдегі шығын функциясының орташа мәндерін көруге болады. Оң жақтағы графиктен сол дәуірдегі модельдердің дәлдік көрсеткіштері берілгенін байқауға болады.

Әрбір модельдің өзіндік артықшылықтары мен ерекшеліктері бар екен. ResNet50 FPN backbone қолданылған Faster R-CNN моделі шығын функциясының ең төмен мәнін – 0.0102 көрсеткені анықталды. Бұл ResNet50 FPN архитектурасының терең құрылымы мен күрделі резидуалды блоктарының арқасында модель оқыту барысында қателіктерді тиімді азайта алатындығын көрсетеді.

Мұндай нәтиже модельдің жоғары деңгейде тұрақты бола алатынын және деректерді жалпылау қабілетін дәлелдейді. Жоғары дәлдік пен сенімділікті талап ететін міндеттер үшін маңызды.



Сурет 2.16. Модельдердің салыстырмалы талдауы.

Self-Attention механизмі арқылы EfficientNetB0 моделі толықтырылған шығынның орташа мәнін 0.0289 деп көрсетті. Бұл жоғары деңгейдегі тиімділікті көрсетіп отыр. Нәтиже салыстыра келгенде төмен есептеу ресурстарын талап ететіні ескеріліп отырған жағдайлар үшін өте жақсы көрсеткіш.

Соңғы дәуірде MobileNetV3 Large архитектурасының шығын функциясының ең жоғары мәні – 0.1807 мәніне ие. Бұл оның ерекше архитектуралық тұстарымен байланысты.

MobileNetV3 Large мобильді және ендірілген құрылғыларда жұмыс жасау үшін қолданылады. Модельдің параметрлерінің санын азайтып, есептеудегі

жылдамдығын арттыруға бағытталады. Модельдің күрделі деректерде тиімді түрде оқыту қабілеті белгілі бір деңгейде төмендеуі мүмкін.

Шығын функциясының графигінде модель оқытудың бастапқы кезеңдерінде жақсы нәтижелерді көрсете білді. Бірақ соңғы дәуірлерде шығын функциясының белгілі бір деңгейде ауытқуы көрініп отыр. Бұл модельдің қайта оқытылуы (overfitting) мүмкін болатынын білдіреді. Мұндай жағдайда модель оқытудың деректерін өте дәл «есте сақтауы». Бірақ жаңа деректерге жалпы қорытындылау қабілетінің төмендеп тұруы болады. MobileNetV3 Large моделінің жылдамдығы мен есептеу қабілеті төмен деңгейде болуы жүктемесіне қарамастан, қателіктерге төзімділігі мен жалпылау қабілетінің төмен болатынын көрсетеді. Модельдердің дәлдік көрсеткіштерін қарастыруда үш модель де соңғы дәуірде 100% дәлдікке жеткені байқалады. Барлығы объектілерді анықтаудағы міндетін жоғары дәлдікпен орындай ала алатынын көрсетіп отыр. Дәлдік көрсеткіштері бірдей болғанымен, шығын функциясының мәндеріндегі айырмашылықтар бар. Ол модельдердің тиімді болуы мен жалпылау қабілеті туралы ақпаратты қосымша бере алады.

ResNet50 FPN және Self-Attention механизмімен толықтырылған EfficientNetB0 модельдері MobileNetV3 Large моделіне қарағанда деректерді жалпылау қабілеті жоғары деңгейде екендігі анықталды. Бұл қасиет жоғары сенімділікті талап ететін есептерде маңызды болып отыр. Модельді таңдау нақты міндеттің талаптары мен қолжетімді есептеу ресурстарына байланысты болады. Егер басты мақсат максималды дәлдік пен тұрақтылық болатын болса, ең тиімді шешім осы ResNet50 FPN архитектурасы. Басты талап жоғары жылдамдық пен есептеу ресурстарын үнемдеу болса, MobileNetV3 Large немесе Self-Attention механизмімен толықтырыла алатын EfficientNetB0 модельдерін қолдануға болады. Тіпті белгілі бір ауытқулар шығын функциясында байқалатын жағдайда да.

Зерттеу барысында Faster R-CNN архитектурасына негізделген және әртүрлі backbone-желілерді қолданатын үш модельдің оқыту нәтижелері талданып отыр. ResNet50 FPN, MobileNetV3 Large және Self-Attention механизмі арқылы толықтырылған EfficientNetB0 архитектураларын қарастырылды. Бұл модельдердің әрқайсысы әртүрлі мақсаттарға бағытталған. Жоғары дәлдікке қол жеткізу немесе есептеу кезіндегі ресурстарын барынша үнемдеу. ResNet50 FPN негізіндегі Faster R-CNN моделі оқыту барысында жоғары дәлдік пен тұрақтылықты көрсетті. ResNet50 архитектурасы құрамындағы резидуалды блоктар градиенттің жоғалу мәселесін тиімді шеше отырып, терең нейрондық желілерді тұрақты оқытуға мүмкіндік бере алады. Сонымен қатар, Feature Pyramid Network (FPN) архитектурасының интеграциясы модельдің әртүрлі масштабтағы белгілерді тиімді анықтау қабілетін арттырады. Әртүрлі өлшемі бар объектілер мен күрделі көріністерді тану кезінде маңызды болып табыдаы. Оқыту процесінде үшінші дәуірде модель 100% дәлдікке жеткенін көруге болады. Шығын функциясының мәні бірінші дәуірде 0,0922 болғанын байқасақ, он бесінші дәуірде 0,0102 деңгейіне дейін төмендегенін көрсетеді. Бұл модельдің тұрақты түрде оқытылғанын және қателіктердің тиімді түрде азайта алғанын

көрсетіп отыр. Алайда модельдің архитектуралық жағынан күрделі және параметерлер санының көп болуына байланысты оқыту уақыты айтарлықтай ұзақ болады. Бір дәуірді оқытуда орта есеппен 12-13 минут уақыт алады. Ал жалпы оқыту уақыты 183 минутты құрағанын көруге болады. Ал MobileNetV3 Large backbone қолданылған Faster R-CNN моделі жылдам оқыту және есептеу кезіндегі ресурстарын тиімді пайдалану қажет міндеттер үшін арнайы әзірленді.

ResNet50 FPN архитектурасынан айырмашылығы, MobileNetV3 Large моделі бөлінген тереңдік конволюцияларын (depthwise separable convolutions) қолдануды қамтамасыз ете алады. Тәсіл модельдегі параметрлер саны мен есептеу кезіндегі операцияларын айтарлықтай азайтып, жеткілікті дәлдікті сақтауға мүмкіндік бере алады. Модель оқыту барысында жоғары деңгейдегі дәлдікке жылдам жетті. Екінші оқу дәуірінде-ақ 100% дәлдікке қол жеткізе алғаны көрініп отыр. Жоғары дәлдікке қарамай, оқыту процесінде шығын функциясының белгілі бір деңгейде ауытқуын байқауға болады. Бұл модельдің деректерде кездесетін өзгерістерге сезімтал болуы мүмкін екенін көрсетеді. Бірінші дәуірде шығын функциясының мәні 0,2581. Екінші дәуірде ол 0,1348-ге дейін төмендеп отыр. Бірақ кейінгі дәуірлерде 0,1435–0,1862 аралығында өзгерісте болатыны анықталды. Соған қарамастан модель барлық оқу кезеңдерінде жоғары дәлдікті сақтай алды. Есептеу кезіндегі ресурстарды аз қолдану және оқыту жылдамдығының жоғары деңгейде болуы модельдің негізгі артықшылықтары болып табылады. Бір оқу дәуірінің ұзақтығы шамамен 2–3 минут деп алатын болсақ, толық оқыту процесі 40 минут уақыт аралығында аяқталып қоятынын көрсетті.

EfficientNetB0 backbone және Self-Attention механизмі қолданылатын Faster R-CNN моделі дәлдік пен тиімділікті үйлестіре алатын көпкезеңді шешім болады.

EfficientNetB0 архитектурасы желінің тереңдігін, ені және кіріс кескінін ажырата алатындығы арасындағы теңгерімді масштабтау принципіне негізделген болады.

EfficientNetB0 құрылымы есептеу кезіндегі шығындарын азайтады. Жоғары дәлдікке қол жеткізуге мүмкіндік береді.

Self-Attention механизмін қосу модельдің кескіннің әртүрлі бөліктері арасындағы жаһандық бір-бірімен байланыстарды ескеріп отыру қабілетін арттырады. Маңызды белгілерге назар аудара алуына және сол белгілерді ескере алуына мүмкіндік береді. Оқыту барысында модель 2 – дәуірде 97.30% шамасында дәлдік көрсетсе, 3 – дәуірде 100% дәлдік нәтижелерін көрсетті. Бұл көрсеткіштердің кейінгі барлық дәуірлерде тұрақты түрде сақталып отырғанын көруге болады.

Бірінші дәуірде шығын функциясының орташа мәні салыстырмалы жағдайда жоғары – 0,4611. Бірақ кейінгі кезеңдерде өте тез төмендеп отыр. 15-дәуірде 0,0289 деңгейінде болды. Бір оқу дәуірінің орташа уақыты шамамен 28 минут. 13 және 14 дәуірлерде оқыту уақыты айтарлықтай өседі. Сәйкесінше 45 және 52 минутты құрайтыны анықталды. Оқыту уақытының артуы шығын

функциясын азайта алу процесінің күрделене түсуі мен модельдің деректердегі күрделі жағдайларға бейімделуі қажет екенін түсіндіруге болады.

EfficientNetB0 және Self-Attention механизмі қолданылған Faster R-CNN моделі дәлдік пен есептеудің тиімділігі арасындағы оңтайланған тепе-теңдікті қамтамасыз етіп отыр. Бұл модель жоғары дәлдік қажет болатын, бірақ есептеу кезіндегі ресурстардың шектеулі болуы қолдануға қолайлылығын айтамыз. Кейбір кезеңдерде оқыту уақытының артуына қарамастан, модель тұрақтылық пен сенімділікті сақтап отырғанын аңғаруға болады. Жүргізілген зерттеудің нәтижелерін негізге ала отырып, әрбір модельдің артықшылықтары мен шектеулерін анықтауға:

- жоғары дәлдік пен тұрақтылық талап етілетін есептер үшін ResNet50 FPN моделі ең тиімді шешім болады. Бұл модель жоғары дәлдікке тез жеткенін көрсе, шығын функциясының ең төмен деген мәндерін көрседі. Алайда архитектурасының күрделі болуына байланысты едәуір есептеу ресурстарын және ұзақ уақыт оқытуын талап ететіні анықталды.

- жылдам оқыту мен төмен есептеу шығындарын қамтамсыз ететін модель MobileNetV3, мобильді және ендірілген жүйелер үшін тиімді шешім болды. Дегенмен деректерге сезімтал болуы және шығын функциясының ауытқуы оның тұрақтылығын төмендетіп отыр.

- дәлдік пен есептеу тиімділігі арасындағы теңгерімді қамтамасыз ететін модель Self-Attention механизмімен толықтырылып отырған EfficientNetB0. Бұл архитектура жоғары өнімділікті және сенімді тануды талап етіп отыратын қолданбалар үшін ұсынылғанын анықтадық.

- Модельді таңдау нақты міндеттің талаптарына және есептеу кезіндегі қолжетімді ресурстарына байланысты. Егер негізгі мақсат максималды дәлдік пен тұрақтылық болатын болса, ең қолайлы нұсқа ол ResNet50 FPN екенін көрсетеді. Ал егер жоғары деңгейдегі жылдамдық пен ресурстарды тиімді түрде пайдалану маңызды болса, онда MobileNetV3 Large немесе EfficientNetB0, қоса Self-Attention модельдері тиімді шешім отыр. Жалпы алғанда, объектілерді анықтау міндеттерінде дәлдік, жылдамдық және есептеу шығындары арасындағы тепе-теңдікті сақтай алу маңызды болып келеді. Бұл әрбір нақты қолдану жағдайында оңтайлы нәтижеге қол жеткізе алуға мүмкіндік беріп отыр. Зерттеудің негізгі қорытындыларын келесідей тұжырымдауға болады:

- нақты уақыт режимінде қозғалатын объектілерді анықтауға арналған модель ұсынылып отыр. Бұл модель Faster R-CNN архитектурасына негізделген және әртүрлі базалық архитектураларды (ResNet50, MobileNetV3 Large, Self-Attention механизмімен толықтыратын EfficientNetB0) қолдануды қамти алатынын көрсетті.

- эксперименттік нәтижелер MobileNetV3 Large негізіндегі Faster R-CNN моделінің екінші оқу дәуірінде 95.70% шамасы дәлдікке жеткенін көрсетті. Бұл модельдің жылдам оқыту қабілетін дәлелдейді. Ал ResNet50 және Feature Pyramid Network (FPN) біріктірілген архитектурасы жоғары деңгейдегі тұрақтылық пен дәлдікті қамтамасыз ете отырып, үшінші дәуірде-ақ 100%

дәлдікке жетіп отыр. Сонымен қатар, 15-дәуірде шығын функциясы 0.0102 деңгейіне дейін орташа мәнге төмендегенін көрсетті.

- Self-Attention механизмімен толықтырылып отырған EfficientNetB0 моделі жоғары дәлдік пен есептеу тиімділігі арасындағы тепе-теңдікте бола алуын көрсетті. Модель үшінші дәуірде 100% дәлдікке жете отырып, 15-дәуірде 0.0289-ға дейін шығын функциясының орташа мәні төмендегенін көрсетеді.

- Базалық архитектураны таңдау нақты есептің талаптарына байланысты болады:

- ResNet50 FPN жоғары дәлдік пен тұрақтылық қажет болатын міндеттер үшін тиімділігі бойынша;

- MobileNetV3 Large есептеу ресурстары шектеулі жүйелерге арналғандығымен;

- Self-Attention механизмі бар EfficientNetB0 дәлдік пен ресурстар шығыны арасындағы оңтайлы теңгерімде болатынын қамтамасыз етеді.

Екінші бөлім бойынша қорытынды

Диссертациялық зерттеу барысында кескіндерді алдын ала өңдеуге және бетті тануға арналған конволюциялық нейрондық желілер мен Retinex алгоритмдері негізінде математикалық модель әзірленді. Ұсынылған модель аясында біркелкі емес жарықтандыруды түзету, контраст сипаттамаларын қалыпқа келтіру және шу бұрмалануларын басу арқылы кіріс деректерінің сапасын арттыру процестері формалданды. Retinex алгоритмін алдын ала өңдеу кезеңінде қолдану белгілер кеңістігінің ақпараттылығын арттырып, әсіресе жарықтандыру жағдайлары өзгермелі күрделі визуалды ортада тұлғаларды тану дәлдігін едәуір жоғарылататыны дәлелденді.

Классикалық математикалық модель бейне кадрын $I_t = S_t + \varepsilon_t$ түрінде сипаттайды, мұнда S_t — пайдалы сигнал, ал ε_t — шу компоненті. Retinex алгоритмін енгізгеннен кейін модель жарықтандыруды түзету функциясымен толықтырылды: $\tilde{I}_t = \varphi(\text{Retinex}(I_t))$.

Retinex теориясы бойынша кескін жарықтығы жарықтандыру және шағылысу компоненттерінен тұрады: $I(x, y) = L(x, y) \cdot R(x, y)$. Сондықтан алгоритмнің негізгі мақсаты жарық әсерін азайтып, пайдалы белгілерді күшейту болды: $R(x, y) = \log I(x, y) - \log L(x, y)$.

Жақсартылған кескін кейін CNN моделіне беріліп, эмбеддинг-векторлар алынды: $z_t = E(\tilde{I}_t; \theta_E)$. Эмбеддингтер косинустық ұқсастық арқылы салыстырылды: $\text{sim}(\hat{z}_a, \hat{z}_b) = \hat{z}_a^T \hat{z}_b$.

Нәтижесінде ұсынылған модельдің құрылымы келесі тізбек түрінде жүзеге асырылды: бейне кадры → Retinex арқылы жарықтандыруды түзету → нормализация → CNN арқылы белгілерді шығару → эмбеддингтерді салыстыру → сәйкестендіру. Эксперимент нәтижелері бойынша 1 м қашықтықта және 20% жарық кезінде тану дәлдігі 60.98%-дан 86%-ға дейін, ал 3 м қашықтықта және 50% жарық кезінде 61.09%-дан 85%-ға дейін артты. Сонымен қатар Assurance көрсеткіші 6.52%-ға, Precision 9.78%-ға және F1-Score 6.82%-ға жақсарды. Бұл

Retinex алгоритмінің күрделі жарықтандыру жағдайларында кескін сапасын жақсартып, тану тұрақтылығын арттыратынын дәлелдейді.

Бейне ағындарындағы кескіндерді тану дәлдігін арттыру және кадрларды өңдеу уақытын оңтайландыру мақсатында CNN негізіндегі көпкөзденді модель құрылды. Мұнда Faster R-CNN, EfficientNetB0, Self-Attention және FaceNet модельдері пайдаланылды. Faster R-CNN күрделі көріністерде объектілерді жоғары дәлдікпен анықтап, ResNet50 FPN бэкбонымен бірге 100% дәлдік пен ең төменгі 0,0102 loss мәнін көрсетті. Алайда оның толық оқыту уақыты 183 минут болғандықтан, есептеу шығындары жоғары екені анықталды. MobileNetV3 Large ең жылдам архитектура болып, толық оқыту уақыты небәрі 40 минутты құрады, бірақ соңғы эпохаларда loss көрсеткіші 0,1807-ге дейін өсіп, тұрақтылығы төмен екенін көрсетті. EfficientNetB0, Self-Attention моделі үшінші эпохада-ақ 100% дәлдікке жетіп, соңғы эпохада loss мәні 0,0289 жетті. Self-Attention механизмі модельге кескіннің маңызды аймақтарына назар аударуға мүмкіндік берді, күрделі жарықтандыру жағдайларында тану сапасын арттырды. FaceNet архитектурасы тұлғаларды евклидтік қашықтық арқылы сенімді салыстыруға мүмкіндік беріп, бет кескіндерін 128 өлшемді эмбеддингке түрлендірді.

Нәтижесінде ұсынылған көпкөзденді модель бейне ағындарындағы тұлғаларды нақты уақыт режимінде жоғары дәлдікпен және тұрақты түрде тануды қамтамасыз етті.

3 Nexiface биометриялық сәйкестендіру жүйесінің архитектуралық шешімдерін әзірлеу және бағдарламалық жүзеге асыру

3.1 Бейне ағындарда тану дәлдігін арттырып, өңдеу уақытын қысқартатын көпкезеңді модельді іске асыру

Қазіргі заманғы терең оқыту жүйелері нейрондық желілерді немесе есептеу механизмі ретінде пайдаланады. Алайда тиімді инженерлік іске асыру үшін оның негізінде жатқан математикалық механизмді түсіну қажет. Нейрондық желілердің математикалық принциптері жадыға қойылатын талаптарды, есептеу күрделілігін және оңтайландыру кеңістігін анықтайды. Бұл өз кезегінде жүйені жобалау шешімдеріне тікелей әсер етеді.

Градиенттік ағын, активация функциясы және кері тарату сияқты негізгі ұғымдарды түсінбей, инженерлер жүйенің мінез-құлқын болжай алмайды. Оқытудағы ақауларды анықтай алмайды немесе ресурстарды тиімді бөле алмайды. Әрбір математикалық операция аппараттық ресурстарға белгілі бір талаптар қояды: мысалы, матрицаларды көбейту жоғары жады өткізу қабілетін қажет етеді. Ал активация функциясын таңдау мобильді процессормен үйлесімділікті анықтайды. Осы операцияларды терең түсіну нейрондық желілерді "қара жәшіктен" болжамды және басқарылатын инженерлік жүйеге айналдырады.

Конволюциялық нейрондық желілер (CNN) негізіндегі бет-әлпетті тану жүйелері сәйкестендірудің жоғары дәлдігін көрсететін көп кезеңді процесс. Барлық процесс кескінді алу кезеңінен басталады, мұнда алдын ала өңдеу алгоритмдері жарықтандыруды нормализациялап, цифрлік шуды жояды.

Одан кейін бет детекторы іске қосылып, кадрды сканерлеу арқылы "Bounding box" бет аймағының координаттарын анықтайды. Бет анықталғаннан кейін теңестіру (alignment) кезеңі орындалады - бұл көз, мұрын, ауыз сияқты негізгі нүктелерді бірыңғай канондық күйге келтіруге мүмкіндік беретін геометриялық түрлендіру процесі.

Кейін нормаланған кескін терең конволюциялық нейрондық желіге беріледі, онда қабаттар арқылы қарапайым сызықтардан бастап күрделі текстураларға дейін иерархиялық белгілер бөлініп алынады. Осы кезеңнің нәтижесінде embedding - көп өлшемді кеңістіктегі бет бейнесінің ықшам математикалық көрінісі қалыптасады.

Конволюциялық нейрондық желіге негізделген бет-әлпетті тану моделі бірнеше кезеңдерден тұратын күрделі өңдеу конвейері (pipeline) ретінде жүзеге асырылады.

Әрбір кадр келесі алгоритмдік кезеңдерден өтеді. Модельдің жұмыс процесін келесі түрде жіктеуге болады:

1. Кескінді алдын ала өңдеу моделі

Процесс құрылғы камерасынан алынған бастапқы кадрды (Raw Frame) қабылдаудан басталады. Бастапқы кескін көбінесе цифрлік шу мен жарықтандыру жеткіліксіз аймақтарды қамтиды, бұл нейрондық желінің

жұмысына кері әсер етеді. Осы кемшіліктерді түзету үшін арнайы Retinex алгоритмі қолданылады.

Бұл әдіс түстің тұрақтылығы теориясына негізделген және жүйеге жарықтың спектрлік құрамына тәуелсіз түрде көріністі дұрыс интерпретациялауға мүмкіндік береді. Алгоритм кескінді екі құрамдас бөлікке бөледі: жарықтандыру (illumination) және шағылысу (reflectance).

Өңдеу барысында жарықтық автоматты түрде теңестіріліп, терең көлеңкелер тиімді жойылады. Нәтижесінде қараңғы аймақтардағы жасырын бөлшектер анық әрі талдауға қолжетімді болады.

Алгоритм нәтижесінде жоғары контрастты және түстері дұрыс берілген оңтайландырылған кескін алынады. Осындай дайындалған кадр объектілерді тану модельдерінің дәлдігін айтарлықтай арттырады.

Алдын ала өңдеу кезеңі күрделі түсіру жағдайларында да жүйенің тұрақты жұмысын қамтамасыз етеді.

2. Faster R-CNN негізінде бетті анықтау (Face Detection).

Retinex алгоритмімен жақсартылған кескін Faster R-CNN нейрондық желісінің кірісіне беріледі. Модель нақты уақыт режимінде объектілерді дәл анықтауға арналған тиімді архитектуралардың бірі. Әуелі базалық желі (Backbone, ResNet) кескіннен белгілер картасын (Feature Map) шығарады. Картада текстуралар, шекаралар, бетке тән негізгі визуалды ерекшеліктер туралы ақпарат болады.

Модельдің негізгі компоненттерінің бірі - ұсыныстар желісі (Region Proposal Network, RPN). RPN белгілер картасын «жылжымалы терезе» әдісі арқылы сканерлейді. Ол әртүрлі өлшемдегі мыңдаған виртуалды аймақтарды (anchors) орналастырады.

Әр аймақ үшін желі «Бұл жерде адам беті бар ма?» деген ықтималдықты есептейді. Егер ықтималдық жоғары болса, аймақ кандидат ретінде белгіленеді де оның координаттары нақтыланады.

Модель бос фонды талдау үшін ресурстарды жұмсамай маңызды аймақтарға назар аударады.

Таңдалған аймақтар RoI Pooling қабатынан өтіп, бірдей өлшемге келтіріледі. Содан кейін бұл аймақтар толық байланысқан қабаттарға жіберіледі. Соңғы классификация жүргізіледі: «Бұл объект бет пе?, әлде фон элементі ме?».

Регрессия модулі шекараларды дәлдеп, анықталған бетке bounding box (шектеу рамкасы) сызады. Егер кадрда бірнеше адам болса, модель әрқайсысын жеке анықтайды.

3. EfficientNetB0 және Self-Attention көмегімен бет белгілерін терең талдау.

Алдыңғы қадамда бөлінген бет кескіні EfficientNetB0 екінші нейрондық желісіне беріледі. EfficientNetB0 - дәлдік пен есептеу жылдамдығы арасындағы тепе-теңдікті сақтайтын оңтайландырылған архитектура.

Модель бет құрылымын талдайды. Көз, мұрын, ерін сияқты визуалды белгілерді ықшам математикалық векторларға түрлендіреді. Сыртқы өзгерістерге (жарық, шаш үлгісі) қарамастан тұрақты белгілерді анықтау желінің негізгі мақсаты.

Негізгі желімен қатар қосымша интеллектуалды қабат - Self-Attention механизмі іске қосылады. Механизм нейрондық желіге дәл сол сәтте беттің қай бөліктері неғұрлым мазмұнды болатынын өздігінен анықтауға мүмкіндік береді.

Self-Attention механизмі кескіннің барлық пиксельдері арасындағы өзара байланыстарды есептеп, салмақ коэффициенттері көмегімен олардың «маңыздылығын» таратады. Мысалы, адам медициналық маска киген, назар механизмі беттің төменгі бөлігі жасырылғанын және пайдалы ақпарат бермейтінін түсінеді. Мұндай жағдайда жүйе автоматты түрде негізгі назарды көз тұсына, қас және маңдай формасына аударады. Егер бет көзілдірік немесе шашпен жартылай жабылып тұрса, алгоритм терінің ашық және анық жерлерін назарға алады.

Динамикалық назар бөлу механизмі модельдің сыртқы кедергілерге тұрақтылығын арттырады. EfficientNet және Self-Attention бірге істеп, терең белгілер картасын (Future Map) қалыптастырады. Карта жүздеген параметрден тұратын тұлғаның бірегей цифрлық сипаттамасы болады.

Нәтижесінде адам беті күрделі көпөлшемді кодқа айналады. Оны басқа тұлғалармен шатастыру немесе қолдан жасау өте қиын. Бұл бірегей белгілер жиынтығы талдаудың соңғы нәтижесі болып табылады. Әрі қарай деректер базасымен салыстыруға дайын болады.

4. Жеке тұлғаны сәйкестендіру және тексеру (FaceNet+Facebase)

Алынған белгілер картасы FaceNet нейрондық желісіне беріледі. FaceNet-тің негізгі мақсаты бет бейнесінің ықшам математикалық көрінісін (embedding) қалыптастыру. Модель визуалды ақпаратты 128 саннан тұратын векторға түрлендіреді. Бұл вектор көпөлшемді кеңістіктегі нүкте координаттары.

Вектор тұлғаның бірегей «цифрлық қолтаңбасы». Жарық немесе бұрыш өзгерсе де тұрақты сақталады. Алынған вектор Firebase бұлттық деректер базасына жіберіледі. Деректер базасында бұрын тіркелген пайдаланушылардың векторлары сақталады. Салыстыру Евклидтік қашықтықты (d) есептеу арқылы жүргізіледі. Егер қашықтық аз болса бұл бір адам, қашықтық үлкен болса әртүрлі адамдар. Жүйе белгілі бір шекті мәнді (threshold) пайдаланады. Егер қашықтық шектен төмен болса – «Жеке тұлға расталды», егер жоғары болса – «Бейтаныс адам». Жаңа тұлғалар үшін жүйе автоматты түрде жаңа жазба жасайды. Сонымен қатар олардың векторларын базаға сақтай алады.

FaceNet-тің жоғары дәлдігі Triplet Loss функциясының негізінде орындалады. Ол әрбір адамдардың векторларын бір-бірінен барынша алшақ орналастыруға үйретеді. Жүйе камерадан алынған пиксельдік деректерді нақты уақыт режимінде тез, дәл сәйкестендіруге айналдырады.

5. Оқиғаларды журналдау және интерфейсте бейнелеу (Firestore және Firebase Storage)

Жеке тұлға сәтті сәйкестендірілгеннен соң немесе «Бейтаныс» ретінде анықталғанда, жүйе деректерді сақтау кезеңіне өтеді. Оқиғаға қатысты мәтіндік метадеректер Cloud Firestore реляциялық емес деректер базасына жазылады. Деректер базасына пайланушының бірегей идентификаторы (UserId), аты, тіркелген уақыты (Timestamp) енгізіледі.

Мәтіндік деректермен бірге визуалды контент те өңделеді. Бастапқы кадр немесе қиылып алынған бет кескіні Firebase Storage бұлттық қоймасына жүктеледі. Firebase Storage үлкен көлемдегі файлдарды негізгі деректер базасын жүктемей ақ сенімді сақтауға мүмкіндік береді.

Кескін сәтті жүктелгеннен кейін файлға тікелей сілтеме (URL) генерацияланады. Ол автоматты түрде Firestore жазбасына байланыстырылып сақталады. Firestore архитектурасы жазылым (Real-time Listeners) қағидасы бойынша жұмыс істейді, бұл интерфейс үшін аса маңызды. Деректер базасында жаңа жазба қосылған сәтте клиенттік қосымша бұл туралы бірден хабарлама алады.

Пайдаланушы интерфейсі (UI) автоматты түрде жаңарып бетті қайта жүктемей-ақ өзекті ақпаратты көрсетеді. Мониторинг экранында адамның аты, фотосуреті және өту уақыты көрсетілген карточка пайда болады. Егер жүйе бөгде адамды анықтаса, интерфейс қауіпсіздік қызметкерлерінің назарын аудару үшін оқиғаны қызыл түспен ерекшелеуі мүмкін.

Барлық деректер әкімшіге келушілер тарихын күн немесе аты бойынша оңай сүзуге мүмкіндік беретін құрылымда сақталады. Мұндай екі құрамды сақтау жүйесі (деректер базасы+бұлттық қойма) жоғары өнімділік пен масштабталуды қамтамасыз етеді.

Осылайша, процесс толық циклмен аяқталады: камерадан кадрды түсіруден бастап жүйе архивінде тексерілген жазбаның пайда болуына дейін.

Толық өңдеу конвейерінің жұмыс тізбегі.

Камерадан секундына 30 кадр (30 fps) жиілігімен түсетін әрбір кадр келесі кезеңдерден өтеді. Алдымен кадр RGB форматынан YCbCr түстік кеңістігіне түрлендіріледі. Бұл процесс өте жылдам, себебі әр пиксельге тек қарапайым арифметикалық операциялар қолданылады. 1920x1080 өлшемді кадр үшін шамамен 6 миллион операция орындалады және бұл миллисекундтар ішінде жүзеге асады.

Келесі кезеңде кадр Faster R-CNN моделіне жіберіледі. Модель Backbone CNN арқылы кадрдан белгілер картасын (Future Map) шығарады. RPN (Region Proposal Network) картаның әр нүктесіне 9 түрлі аймақ (рамка) ұсынып, олардың ықтималдығын бағалайды. NMS (Non-Maximum Suppression) алгоритмі қабаттасқан аймақтарды біріктіреді. Шамамен 300 ең жақсы аймақты таңдайды. RoI Pooling әр аймақты 7x7 бірдей өлшемге келтіреді. Ал соңғы классификатор мен регрессор әр аймақты нақтылап, жіктейді. Процесс шамамен 20-25 миллисекунд уақыт алады.

Егер кадрда бірнеше адам болса, әрбір бет жеке өңделеді. Өңдеу параллель түрде жүзеге асады. Бірінші бет EfficientNetB0 моделіне жіберілгенде, екінші бет кезек күтпей өңделе бастайды. Бұл GPU-дың көпядролы архитектурасының негізінде орындалады.

EfficientNetB0 моделінен өткенде әрбір бет бірнеше MBConv блоктары арқылы өңделеді. Блоктар белгілер картасы кеңейтіледі және терең талданады. SE блоктары маңызды белгілерді күшейтеді. EfficientNetB0-ден кейін Self-Attention механизмі қосылады. Бет бөліктері арасындағы өзара байланысты

модельдейді. Бұл бет жартылай жабылған немесе әртүрлі ракурста түсірілгенде маңызды.

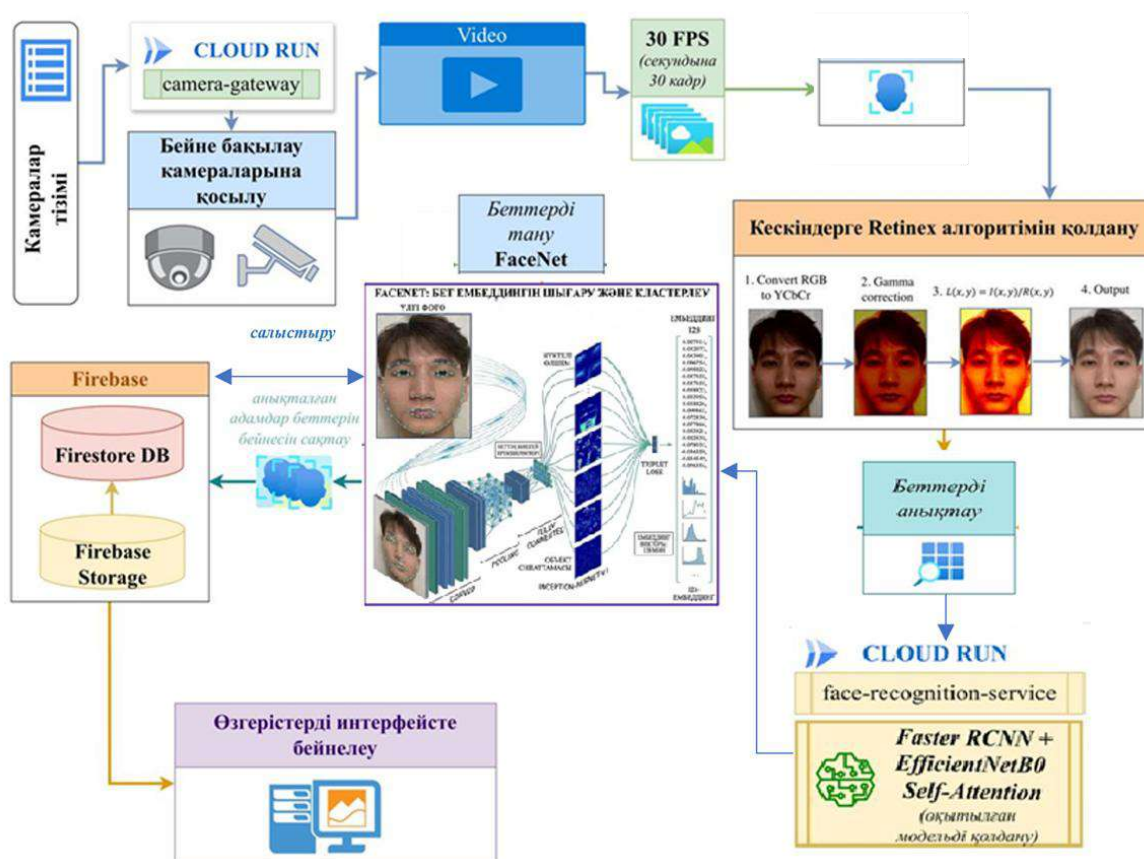
Соңында әрбір бет FaceNet моделіне жіберіледі. FaceNet бет кескінін 128 саннан тұратын векторға (embedding) түрлендіреді. Вектор әр адамның бірегей "математикалық қолтаңбасы". Жаңа алынған вектор Firebase жүйесінде сақталған векторлармен салыстырылады. Егер вектор базадағы қандай да бір векторға өте жақын орналасса (арақашықтық белгілі бір шектен төмен болса), адам танылды деп саналады. Егер сәйкес вектор табылмаса, адам "бейтаныс" деп белгіленеді. Қажет болған жағдайда жаңа жазба ретінде базаға сақталады.

Нәтижелер Firebase жүйесінің екі түрлі сақтау ортасына жазылады. Метадеректер (кім, қашан, қай камера) Firestore DB-де, ал бет кескіндері Firebase Storage жүйесінде сақталады. Осы деректер негізінде интерфейстегі "Мониторинг" бетті нақты уақыт режимінде жаңартылады. Ақпарат пайдаланушыға көрсетіледі. Кадрды толық өңдеу шамамен 45-50 миллисекундты құрайды, бұл секундтың 25-30 кадр өңдеуге мүмкіндік береді.

Модель бейне ағын кадрларын көпдеңгейлі өңдеу (алдын-ала жақсарту, бет аймағын анықтау, терең нейрондық желілер негізінде сәйкестендіру) арқылы жұмыс істейді. Retinex алгоритмін қолдану жарық әсерін азайтады. Faster R-CNN және EfficientNetB0 архитектураларын біріктіру бет аймақтарын дәл анықтауға мүмкіндік береді. FaceNet моделімен алынған эмбеддингтер жоғары дәлдікті сәйкестендіруді қамтамасыз етеді. Нәтижесінде жүйе нақты уақыт режимінде жоғары дәлдік пен төмен кідірісті қамтамасыз етеді.

3.2 Бетті тану тиімділігін арттыруға арналған нейрондық желі архитектурасын әзірлеу

Nexiface мониторинг жүйесі микросервистік, бұлттық (Cloud Native) архитектура негізінде serverless технологияларын қолдана отырып құрылған. Жүйе бейне ағындарды нақты уақыт режимінде талдайды. 3.1-суретте архитектураның әрбір қабатының толық сипаттамасы бейнеленген.



Сурет 3.1. Nexiface архитектурасы

1. Деректер көзі қабаты (Camera Layer). Қорғалатын нысандарға орнатылған бейнебақылау камералары деректерді жинау жүйесінің ақпараттық негізін құрайды. Құрылғылар әр арна үшін секундына 30 кадр жиілігімен үздіксіз бастапқы бейне ағынды қалыптастырады. Платформа архитектурасы әртүрлі жабдықтарды қолдайды. Атап айтқанда, Ethernet арқылы қосылатын заманауи IP-камералар, жергілікті тапсырмаларға қолданылатын USB-камералар және қолданыстағы бейнебақылау жүйелерімен интеграцияға арналған RTSP-ағындар. Физикалық камера жұмысын бастарда бұлттық платформада тіркеліп, бірегей жолдық идентификаторды алады. Орталық Firebase платформасына камералардан бейнедеректер тек трафикті шифрлаумен қорғалған хаттамалар бойынша жүзеге асады. Қауіпсіздікті қамтамасыз ету үшін TLS-тің үстінен HTTPS және WebSocket хаттамалары пайдаланылады. Хаттамалар деректерді ұстап қалуды және алмастыруды болдырмайтынын көрсетті. Бейне ағын кейін тану сервистері арқылы өңделу үшін нақты уақытта бұлтқа үздіксіз жіберіліп отырады. Кідірістер детекцияның дәлдігіне кері әсер етуі мүмкін болғандықтан жүйенің байланыс сапасына қоятын талабы жоғары. Байланыстағы уақытша үзілістерді өтеу үшін камера немесе жергілікті шлюз жағында буферлеудің интеллектуалды механизмі қарастырылғанын байқауға болады. Байланыс қалпына келген кезде жүйе үзіліс кезінде жоғалған бейне фрагменттерін қайта жіберуді автоматты түрде іске қосады. Камералардың қосылуы мен ажыратылуы туралы барлық оқиғалар кейінгі аудит үшін Firestore деректер базасының жеке коллекциясында тіркеледі. Техникалық мамандар басқару панелі арқылы

камераның күйін және кіріс сигналының сапасын бақылап отыра алады. Мұндай деректерді жинау тәсілі желі тұрақсыз болған жағдайда да мониторингтің үздіксіздігін және ақпараттың сақталуын қамтамасыз етеді.

2. Деректерді жинау және басқару қабаты (Ingestion & Storage Layer - Firebase). Firebase платформасы мониторинг жүйесінің ішінде барлық ақпараттық ағындарды бағыттайтын орталық деректер шинасы қызметін атқарады. Бейнебақылау камералары Firebase-пен тікелей байланыс ортанып, бейне ағынды аралық серверлерсіз жібереді. Бұлтты тиімді өңдеуді қамтамасыз ету үшін бейнедеректер оңтайландырылған қысу форматында беріледі. Firebase техникалық инфрақұрылымы камералардан өңдеуші сервистерге дейін бейнені жеткізу кезінде минималды кідірісті қамтамасыз етеді. Firebase-та Cloud Run ортасында орналастырылған екі негізгі микросервис жұмыс істейді. Бірінші микросервис - camera-gateway. Ол кіріс бейне ағындағы қабылдау және бастапқы өңдеуге жауап береді. Шлюз үздіксіз бейнеден секундына 30 кадр жиілігімен жеке кадрларды бөліп алады. Camera-gateway алдын ала сүзгіден өткізеді. Бұлыңғыр немесе ақпараттық құндылығы төмен кескіндерді автоматты түрде алып тасайды. Сүзгіден өткен кадрлар тану сервисіне жіберу үшін қажетті форматқа келтіріледі. Cloud Run жүйеге түсетін жүктеме артқанда camera-gateway қызметінің автоматты түрде масштабталуын орындайды. Екінші микросервис - face-recognition-service. Барлық мониторинг платформасының есептеу ядросы болып табылады. Сервис құрамында кескін сапасын жақсартуға арналған Retinex алдын ала өңдеу моделі бар. Үш нейрондық модель жұмыс істейді. Беттерді анықтау үшін Faster R-CNN моделі. Белгілерді шығару үшін Self-Attention механизмімен толықтырылған EfficientNetB0. Сәйкестендіру үшін FaceNet моделі. Face-recognition-service алынған эмбедингтерді Firestore деректер базасында сақталған вектормен салыстыру логикасын орындайды. Сервис экожүйенің басқа компоненттерімен өзара әрекеттесу үшін бағдарламалық API интерфейсін ұсынады. Face-recognition-service-ке жіберілетін барлық сұраныстар жүктемені теңгеру жүйесі (load balancer) арқылы өтеді. Ол есептеу тапсырмаларын сервис даналары арасында тиімді бөледі.

3. Өңдеу қабаты (Processing Layer). Жүйенің негізгі есептеу компоненті болып табылады. Бейнедеректерді талдау мен бет-әлпетті тануға жауап береді. Өңдеу қабаты екі негізгі модульден тұрады. Кескін сапасын жақсарту модулі және нейрондық желілер жиынтығы.

Retinex кескін сапасын күшейтуге арналған модуль. Ол күрделі жарықтандыру жағдайында тану дәлдігін анықтауға бағытталған. Өңдеу процесінің алғашқы қадамы бастапқы RGB кескінін YCbCr түстік кеңістігіне түрлендіру. Бұл түрлендіру жарықтық пен түстік ақпаратты бөлуге мүмкіндік береді. Кейінгі талдау үшін аса маңызды. Компоненттердің бөлу кескіні арнайы өңдеуге дайындайды.

Кескіннің барлық бөліктеріндегі жарықтықты теңестіру үшін (гамма-коррекция) қолданылады. Бұл әдіс көлеңкелі аймақтарды күшейтеді. Қараңғыда жасырын қалған бөлшектерді айқын көрсетеді. Кескін математикалық түрде жарықтандыру (illumination) және шағылысу (reflectance) құрамдастарына

бөлінеді. Retinex модулінің соңғы кезеңі кескінді қашықтан біркелкі емес жарықтандырумен қалпына келтіру болады. Мұнда тері құрылымы және ұсақ белгілер сияқты беттегі барлық маңызды бөлшектер өзгертілмеген түрде сақталады. Өңдеу нәтижесінде жарықтың өзгеруіне төзімді және дәл тануға жарамды кескін шығады.

Сапасы өңделгеннен кейін кескін беттерді танып, анықтау үшін нейрондық желілер конвейеріне түседі. Конвейерде бірінші болып өңделген кескіндегі беттерді анықтауға жауап беретін Faster R-CNN моделі жұмыс істейді. Faster R-CNN моделінің кіріс деректеріне Retinex модулімен дайындалған кескін беріледі. Шығыс деректеріне анықталған барлық беттердің шекараларын белгілейтін тікбұрышты аймақтардың координаттары қалыптасады. Faster R-CNN екі кезеңді архитектураны пайдаланып, объектілерді жоғары дәлдікпен локализациялауды қамтамасыз етеді.

Конвейердегі екінші модель - Self-Attention механизмімен біріктірілетін EfficientNetB0. Модель бет белгілерін шығарып алуға жауап береді. Бұл модельдің кірісіне бет аймақтары кесіліп алынған кескін бөліктері беріледі. EfficientNetB0 осы аймақтарды тұлғаның бірегей сипаттамаларын кодтайтын кішкентай белгілер картасына (feature map) түрлендіреді. EfficientNetB0-дің жеңілдетілген архитектурасы есептеу жылдамдығын жоғары деңгейде сақтай отырып, сапаны төмендетпейді. Кіріктірілген Self-Attention механизмі модельге беттегі ең маңызды аймақтарға назар аударуға мүмкіндік береді. Нәтижесінде модель көз, мұрын және жақ сызықтары сияқты негізгі элементтерді дәл ажыратады.

Конвейердің соңғы моделі - FaceNet, ол алынған белгілер негізінде бет-әлпетті тікелей тануды жүзеге асырады. FaceNet моделінің кірісіне Self-Attention механизмімен толықтырылған EfficientNetB0 арқылы алынған белгілер картасы беріледі. Модельдің шығысында тұлғаның бірегей математикалық қолтаңбасын сипаттайтын 128 өлшемді эмбединг векторы қалыптасады. FaceNet embedding кеңістігін ұйымдастыру үшін triplet loss негізінде метрикалық оқытуды қолданады. Triplet loss бір адамға тиесілі векторды жақындатып, әртүрлі адамдардың векторларын бір-бірінен алыстатуды қамтамасыз етеді.

Алынған 128 өлшемді вектор Firestore деректер базасында сақталған барлық векторлармен салыстырылады. Салыстыру үшін косинустық ұқсастық немесе Евклидтік қашықтық қолданылады. Егер вектор эталондардың біріне жеткілікті жақын болса, тұлға танылды деп есептеледі. Үш модель бірізді жұмыс істеп, бастапқы кадрдан бастап сәйкестендірілген тұлғаға дейін толық өңдеу конвейерін қалыптастырады.

4. Деректер базасы қабаты (Database Layer - Firebase). Firebase деректер базасының қабаты екі қосымша қоймадан тұрады: Firebase құжатқа-бағытталған базасы және Firebase Storage объектілік қоймасы. Firestore DB танудың әрбір оқиғасы туралы барлық метадеректерді сақтайды. Метадеректерде адамның бірегей идентификаторы, анықтаудың дәл уақыт белгісі (timestamp) сақталатыны анықталып отыр. Базаға әрбір анықталған бет үшін камераның идентификаторы, кадрдағы беттің орналасуын көрсететін координаттар, тану сенімділігінің мәні

енгізілетіндігі көрінді. Firestore базасы барлық оқиғалардың толық тарихын жүргізіп, уақыт бойынша тіркелген барлық анықталған тізбегін қалыптастырады. Осы деректер негізінде жүйе бақылау аймағына адамдардың кіруі мен шыққан уақыттарын автоматты түрде анықтайтынын көруге болады. Жүйе келушілер статистикасын жинақтайды. Firebase Storage қоймасы кейінгі тексеру үшін беттері бар бастапқы кадрлар сияқты барлық графикалық деректерді сақтайды. Жылдам қарауға, тексеруге пайдаланылатын беттердің қиылған кескіндері қоймада сақталады. FaceNet моделіне арналған векторлық эмбеддингтер құрылатын эталондық кескіндердің базасына бөлек қолдау көрсетіледі. Қоймада орналасқан барлық кескіндер адамдардың идентификаторлары мен уақыт белгілеріне тіркелген иерархиялық құрылым бойынша ұйымдастырылғаны көрінеді.

5. Пайдаланушы интерфейсінің қабаты (UI Layer). Жүйенің пайдаланушы интерфейсі танудың барлық процесстерін визуалды бақылай алатын «Мониторинг» парақшасы арқылы ұсынылып отыр. Парақшаның негізгі функциясы нақты уақыт режимінде кез келген қосылған камерадан live-трансляциялау мүмкіндігімен бейне ағынды көрсетуінде. Бейне ағында тану жағдайына сәйкес келетін түрлі түсті жақтаулар көмегімен анықталған беттер автоматты түрде бейнеленеді. Ағымда анықталған деректердің уақыты мен идентификаторлары көрсетілген өзекті тізім бейне ағын жанындағы ақпараттық панельде бейнеленетіні көрінеді. Пайдаланушы таңдалған уақыт аралығы бойынша толық статистиканы көре алады. Панельде оқиғалардың жалпы саны мен бірегей келушілер саны да көрсетіледі. Маңызды оқиғалар туралы хабарламаларды жүйе автоматты түрде қалыптастыратынын анықталды. Мысалы, бейтаныс адамдар немесе "қара тізімдегі" адамдар пайда болғанда орындалатынын байқауға болады. Пайдаланушы интерфейсі арқылы детекцияның сезімталдығы мен сенімділік шектерін орнату сияқты тану параметрлерін басқару мүмкіндігі де қолжетімді. Әкімшілер жаңа эталондарды қосу немесе бар жазбаларды түзете отырып, танымал тұлғалардың базасын басқару мүмкіндіктеріне ие. Сыртқы талдау және архивтеу үшін есептерді әр түрлі форматтарда экспорттау мүмкіндігі қарастырылғанын аңғаруға болады. Деректер базасындағы барлық өзгерістер оқиғаларды бірден жеткізетін тұрақты WebSocket – байланыстарының негізінде интерфейсте көрініп тұрады. Оқиғалардың жылдам жеткізілетіндігін байқауға болады.

6. Жүйе архитектурасының ерекшеліктері. Nexiface жүйесінің құрылымы микросервистік тәсіл негізінде әзірленген. Жүйедегі camera-gateway және face-recognition-service компоненттері толықтай тәуелсіз жұмыс істейді. Мұндай оқшаулану жүйені толық тоқтатпай-ақ тек жүктемені жоғары бөліктерді оңай масштабтауға мүмкіндік береді. Бір сервистегі ақау қалған жүйенің жұмысына әсер етпеуін жүзеге асырады. Cloud Native қағидалары бойынша барлық сервистер контейнерлерге оралған. Олардың кез келген ортада бірдей жұмыс істеуін ұйымдастырады. Контейнерлерді басқару Kubernetes платформасы арқылы орындалады. Істен шыққан сервистерді автоматты түрде қайта іске қосып, жүктемені теңестіреді. Cloud Run платформасындағы serverless

технологиялардың арқасында тапсырыс беруші бос тұрған ресурстарға шығындалмай, кодтың нақты орындалу уақытын ғана төлейді. Серверлерді басқарудың әдеттегі тапсырмаларынан команданы босатып, инфрақұрылым бұлтты провайдермен толық басқарылады. Жүйенің жоғары сенімділігіне әртүрлі қолжетімділік аймақтарында сервистерді автоматты түрде қайталау арқылы қол жеткізіледі. Нақты уақыт режимінде жұмыс істеуді қамтамасыз ету үшін жүйе әр қосылған камерадан секундына 30 кадрға дейін өңдейді. Камерадан кадрды түсірген сәттен бастап интерфейсте нәтижені көрсетуге дейінгі жалпы кідіріс 100 миллисекундтан аспайды. Барлық есептеу процестері өнімділікті жоғалтпай бір уақытта бірнеше бейнағынды параллель өңдеуге арналған.

Nexiface архитектурасы - әрбір қабаты нақты белгіленген функция-ларды орындайтын, заманауи бұлттық бейнебақылау жүйесі. Мұндай модульдік жүйені жеңіл масштабтауға, жаңа мүмкіндіктер қосуға және бейнебақылаудың маңызды деректерімен жұмыс істеуде жоғары сенімділікті қамтамасыз етуге мүмкіндік береді.

3.3 Nexiface веб-платформасында бетті тану жүйесінің функционалдық жүзеге асырылуы

Интеллектуалды компоненттерді (соның ішінде Faster R-CNN, EfficientNetB0 және Self-Attention механизмдерін) оқыту және валидациядан өткізгеннен кейін, модельдерді нақты ұйымдық ортада енгізуге арналған қолданбалы Nexiface веб-платформасы әзірленді. Платформа білім беру ұйымдарына және онымен байланысты құрылғыларға бағытталған, мұнда бейне ағындарды үздіксіз мониторингтеу, қатысу оқиғаларын тіркеу және аналитика қалыптастыру талап етіледі. Зерттеу прототиптерінен айырмашылығы, Nexiface толық қолданбалы контурды жүзеге асырады: камераларды қосу және бейне ағынды қабылдаудан бастап, интеллектуалды талдау, нәтижелерді сақтау, визуализация және әкімшілендіруге дейінгі барлық кезеңдерді қамтиды.

Шешім архитектурасы клиенттік интерфейс, бұлттық сақтау инфрақұрылымы, серверлік есептеу сервистері арасында жауапкершілікті бөлу қағидасына сай құрылған. Flutter Web-қосымша ретінде іске асырылған. Интерфейстің бірізділігі, модульдігі, көптілділікті қолдау, мониторинг, оқиғалармен жұмыс, сондай ақ, камералар мен пайдаланушылар профилін басқару сценарийлерінде жылдам әрекет етуді орындайды. Жүйені браузер арқылы күрделі орнатусыз пайдалану мүмкіндігі де қарастырылған. Жүйеде бұл функцияны Веб-формат іске асырады. Аталған функция пайдаланушылар саны көп әрі техникалық дайындығы әртүрлі ұйымдар үшін маңызды.

Инфрақұрылымдық ядро ретінде Firebase экожүйесі пайдаланылған. Firebase Authentication қауіпсіз қолжетімділік пен рөлдерді айқындауға жауапты. Cloud Firestore объектілерді (адамдардың профильдері, камералар, кампустардың орындары, оқиғалар журналдары, жинақталған аналитика) және интеллектуалды өңдеу нәтижелерін сақтау үшін негізгі құжатқа-бағытталған деректер қоры ретінде пайдаланылады. Firebase Storage медиа – деректер

(түсірілімдер, бейненің фрагменттері, техникалық артефакттар) үшін қолданылады. Негізгі деректер базасының жүктемесін азайта отырып, жүйені масштабтауды жеңілдетеді. Firebase Hosting веб-клиентті орналастырып, статикалық ресурстардың қауіпсіз жеткізілуін орындайды.

Google Cloud Run ортасында серверлік қабат орналастырылған. Жүктемеге сәйкес масштабталатын оқшауланған контейнерлер (микросервистер) жиынтығы ретінде жүзеге асырылады. Жеке сервис құрылғылармен және бейне ағындармен жұмыс істеуді қамтамасыз етеді (WebRTC арқылы кадрларды интерфейске жеткізу). Интеллектуалды өңдеу сервистері беттерді тану және сәйкестендіруді (соның ішінде бейтаныс тұлғаларды өңдеу), модельдер мен артефакттерді автономды түрде оқыту және жаңартуды, сондай-ақ қауіпті әрекеттерді (мысалы, буллинг немесе төбелес) анықтауды жүзеге асырады. Сервистерге бөлу модульдерді тәуелсіз жаңартуға және есептеуге жүктемесін тиімді бөлуге мүмкіндік береді.

Nexiface жүйесінің маңызды ерекшеліктерінің бірі - нақты уақыт режимін және фондық есептеулерді қолдау. Пайдаланушы әрекеттері (тіркеу, камераларды баптау, мониторингті қарау) ұзақ уақытты қажет ететін машиналық оқыту процестерімен бөгелмейді. Ауыр есептеулер Cloud Run ортасында автономды түрде орындалып, нәтижелері Firestore және Storage жүйелерінде сақталып, әкімші панелінде бірден қолжетімді болады. Қолданбалы деңгейде платформа бейне ағынды құрылымдалған оқиғалар мен көрсеткіштерге түрлендіреді: кіру/шығу, аймақтардағы қатысу, бейтаныс тұлғалардың пайда болуы, аномалияларды тіркеу және келушілер мен белсенділік бойынша аналитика қалыптастыру.

Nexiface - Flutter Web (Dart), Firebase және Google Cloud Run технологияларын біріктіреді. Нақты ортада оқытылған модельдерді қолдануға арналған тұрақты әрі масштабталатын жүйе.

Firebase - Google компаниясының Backend-as-a-Service (BaaS) бұлттық платформасы. Платформа дайын серверлік сервистер арқылы қосымшаларды әзірлеу мен пайдалануды жеңілдетуге арналған.

Төмендегі 3.2-суретте клиент пен Firebase арасында өзара әрекеттесудің базалық логикасы көрсетілген: қосымша API арқылы сұраныстарды (оқу/жазу операциялары) жіберіледі, жауаптарды JSON форматында қабылдайды, ал деректер бұлттық қоймада сақталып, сол жерден алынады.

Мұндай архитектура клиенттік бөліктің бұлтпен интеграциясын жеңілдетеді және деректер мен сервистерге бірыңғай стандартталған қолжетімділікті қамтамасыз етеді.



Сурет 3.2. Клиенттің Firebase-пен API және бұлттық деректер қоймасы арқылы өзара әрекеттесуінің жалпыланған схемасы

3.2-суреттен көрініп тұрғандай Firebase орталық инфрақұрылымдық қабат ретінде қызмет атқарады: ол деректерді сақтау, синхрондау және қолжетімділікті басқару функцияларын өзіне алады, ал клиенттік қосымша деректермен бірыңғай интерфейстер арқылы жұмыс істейді. Нақты жүйелерде бұл схема қауіпсіздік ережелерімен, нақты уақыт механизмдерімен және медиа-объектілерді бөлек сақтаумен толықтырылған.

Firebase-тің тәжірибеде пайдаланылатын негізгі компоненттері: Firebase Authentication-аутентификация және рөлдерді басқарады; Cloud Firestore -NoSQL құжаттық деректер базасы; Firestore Storage-медиаға арналған объектілік қойма; Firebase Hosting -веб-клиентті орналастырады; Cloud Function-серверлік себептер, фондық тапсырмалар үшін; Firebase Cloud Messaging- push-хабарламалар үшін; сонымен қатар пайдалану құралдары - Analytics, Crashlytics, Performance Monitoring және Security Rules, App Check- қорғау механизмдері.

Клиенттік веб-платформа мен серверлік есептеу сервистерін сенімді бұлттық сақтау үшін, қолжетімділік механизмдері арқылы байланыстыратын бірыңғай инфрақұрылымдық ядро ретінде Firebase таңдалды. Firebase пайдаланушылар құқықтарын басқару, объектілер мен оқиғаларды құрылымдап сақтау, өзгерістерді бірден көрсетуге мүмкіндік береді. Сонымен қатар, медиа-деректермен жұмыс, веб-клиентті орналастыру, дайын қауіпсіздік, мониторинг құралдарының негізінде операциялық күрделілікті төмендету мүмкіндіктерін орындайды.

Firebase қауіпсіздігі көпдеңгейлі модель ретінде іске асырылады. Қорғаныс пайдаланушыны аутентификациялау, деректерге қолжетімділік құқықтарын басқару, сұраныстарды тексеру және сенімді серверлік логика деңгейлерінде орындалады. Жүйе жеке деректерді, қатысу оқиғаларын және медиа-артефакттарды (сурет/кадрлар/бейне үзінділері) өңдейді. Рөлдер мен ұйымға тиесілілік бойынша қатаң ажыратуды талап етеді. Платформа үшін берілген талап маңызды рөл атқарады. Қауіпсіздік "келісімдермен" емес, Firebase Authentication, Security Rules, Cloud Functions формалды механизмдерімен, сонымен қатар, ең артықшылықтар мен журналдау қағидаларымен берілетіні анықталды.

Деректерді жазудың типтік қауіпсіз сценарийі 3.3-суреттен байқауға болады. Клиенттің деректер базасына тікелей жазуды орындамай, серверлік функцияға сұраныс жіберетінін көре аламыз. Токен мен деректер осы функцияда тексеріледі. Cloud Firestore деректер базасына жазба жасау сәтті валидациядан кейін ғана орындалады. Маңызды операциялар үшін (пайдаланушы профилін тіркеу, рөлдерді өзгерту, сезімтал оқиғаларды тіркеу) берілген тәсіл тиімді. Басқаруды сенімді серверлік ортаға көшіреді.



Сурет 3.3 – Токен мен деректерді тексеру арқылы Cloud Function арқылы Firestore-қа қауіпсіз қолжетімділік схемасы

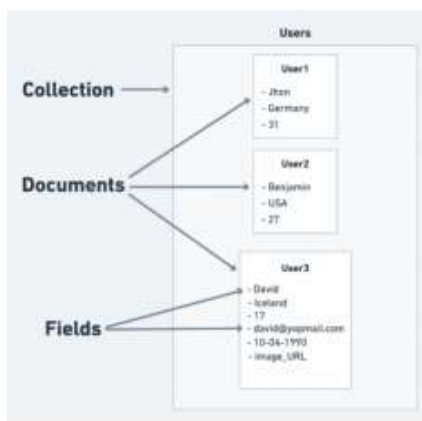
Пайдаланушының алдымен аутентификациядан өтуі 3.3-суретте көрсетілген. Жеке басын растайтын ID-токен алады. Cloud Functions (HTTPS) сервисіне сұраныс жіберіледі.. Токеннің нақтылығын тексеру, рөлдерді сәйкестендіру, сұраныс құрылымын валидациялау, ұйымдық контексті (scope) бақылау және операцияның рұқсат етілуін анықтау сияқты қауіпсіздік тексерулері орындалады. Серверлік функция Cloud Firestore деректер базасына жазбаны тексеруден кейін жасайды. Орындалған тәсіл рұқсатсыз өзгерістерді, деректерді бұрмалау қаупін айтарлықтай төмендетеді.

Жүйеде пайдаланылатын Firebase қауіпсіздігінің кілттік механизмдері:

- Authentication – пайдаланушыларды басқару, токендерді беру;
- Firestore Security Rules – коллекциялар мен құжаттар деңгейінде оқу-жазу саясаты;
- Cloud Function – тексеру және бизнес логиканы орындауға арналған сенімді орта;
- Storage Rules – медиа-файлдарды қорғау;
- App Check – сенімсіз қосымшалардан келетін жалған сұраныстардан қорғау;
- функциялар логтары мен оқиғалар жазбалары арқылы аудит және бақылау.

Ұйым деректерінің оқшаулануы мен рөлдердің қатаң моделінің қамтамасыз етілуі, маңызды операциялардың басқарылатын серверлік контур арқылы орындалуы, медиа-деректердің қолжетімділік ережелері мен шектеулермен қорғалуы, дайын қауіпсіздік механизмдері есебінен қателердің азаюы, журналдау мен бақылау арқылы жүйе сенімділігінің артуы платформаның практикалық нәтижелері болады.

Cloud Firestore - да деректерді ұйымдастырудың базалық логикасы 3.4-суретте ұсынылған. Жоғарғы деңгей ішінде құжаттар (User1, User2, User3) орналасатын коллекциямен (мысалы, Users) беріледі. Әр құжат объектінің қасиеттерін сипаттайтын өрістер (атрибуттар) жиынынан тұрады. Коллекция бір типтегі объектілердің контейнері, құжат жеке объектіні (мысалы, пайдаланушы профилі немесе камера жазбасы) береді, ал өріс атрибуттардың мәндерін (идентификаторлар, аттары, мәртебелері, уақыт белгілері, медиаға сілтемелер және басқа параметрлер).



Сурет 3.4. Cloud Firestore жүйесінің логикалық құрылымы

Firestore жүйесінің негізгі ерекшелігі алдын ала анықталған құрылыммен емес, еркін түрде ұйымдастырылуы. Әртүрлі өрістер жиынтығына бір коллекциядағы құжаттар ие бола алады. Деректер моделін миграциясыз кеңейту (жаңа сапа параметрлерін, алгоритм нұсқаларын, қызметтік жағдайларды қосу) мүмкіндігін береді. Бұрынғы жазбалардың тұтастығын бұзбайды.

Құжаттық модель бейнебақылау және аналитика жүйелерінде пайдаланған ыңғайлы. Онда әртүрлі типтегі адамдардың профильдері, камералар, кампус локациялары (ғимарат/қабат/аудитория), тану оқиғалары, "бейтаныс" бақылаулар сондай-ақ, агрегатталған көрсеткіштер сияқты объектілер сақталады. Firestore жүйесі өрістермен сұраныстарды (сүзгілеу, сұрыптау, шектеу, индекстер) құрады. Сұраныстар интерфейске тек қажетті деректерді алу мүмкіндігін береді. Белгілі бір уақыт аралығында болған оқиғаларды, нақты аймақтағы камералар немесе кейін тексерілетін "бейтаныс" тұлғалардың тізімін мысалға келтіруге болады.

Firestore жүйесін NoSQL деректер базасына жатқызуға болады. Құрылым кестелер түрінде емес, JSON-ға ұқсас құжаттар түрінде сақталады. Жаңа атрибуттарды қосу, схеманы өзгерту және реляциялық деректер базасына тән күрделі миграцияларды қажет етпейді.

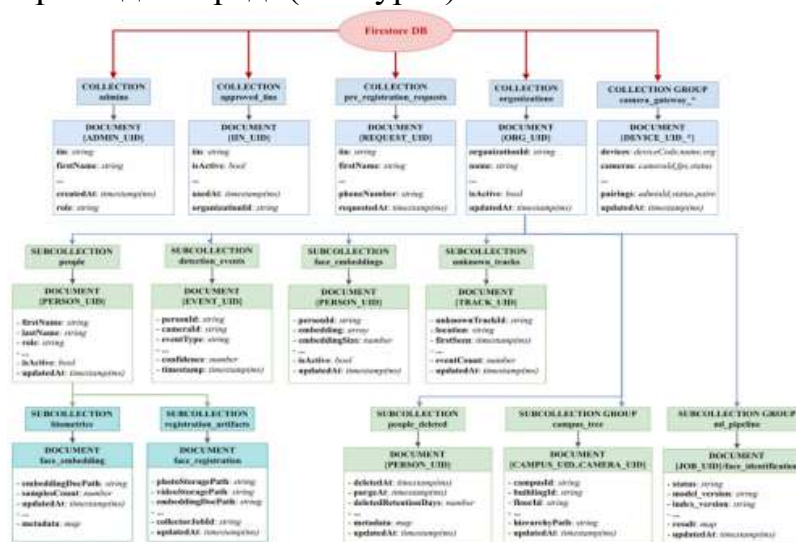
Firestore құжатқа бағытталған NoSQL қоймасы. Ішкі құрылымдарды, массивтерді және құжаттарға сілтемелерді қолдайды. SQL тіліндегі JOIN операцияларын қолданбай-ақ объектілер арасындағы байланыстарды модельдеуді жеңілдетеді.

Firestore коллекция Firestore – құжат – ішкі коллекция (subcollection) түріндегі логикалық иерархияны қамтамасыз етеді. Деректерді ұйым бойынша бөлуге (multi-tenant) өте қолайлы. Жазбаларды ұйымдар бойынша топтастыруға және қолжетімділікті қауіпсіздік ережелері арқылы нақты шектеуге болады. Платформа контекстіндегі бұл тәсіл оқиғалар мен жүйе күйлерін масштабталатын түрде сақтайды. Интерфейсте деректер жедел жаңартылып, жүйе дамуына қарай құрылымды тиімді басқарылады.

Деректер базасы Cloud Firestore негізінде жүзеге асырылған. Бейнебақылау мен интеллектуалды аналитика міндеттеріне оңтайландырылған иерархиялық, құжатқа бағытталған модель ретінде жобаланған. Архитектуралық идея деректерді екі деңгейге бөлуден тұрады: жүйелік деңгейдегі әкімшілік коллекциялар, ұйымдық контур. Оның ішінде мекеменің нақты құрылымын

бейнелейтін ішкі коллекциялар (адамдар, камералар, локациялар, оқиғалар және ML-процестердің күйлері). Мұндай тәсіл масштабтауды және жүйенің эволюциялық дамуын қамтамасыз етеді. Firestore-да схемаға тәуелсіз жаңа өрістер мен құжаттарды қосу мүмкіндігі бар. Функционал мен метадеректер құрамын біртіндеп кеңейту кезінде өте маңызды.

Қолжетімділікті басқару, тіркеу сценарийлерін орындайтын коллекциялар жоғарғы деңгейде орналасқан. admins коллекциясы әкімшілердің құжаттары (ADMIN_UID идентификаторы) мен атрибуттарын сақтайды. approved_iins коллекциясы бастапқы тексеру (whitelist) рөлін атқарады. Жазбалар идентификатор мәнінен, белсенділік пен қолдану белгілерінен, сондай-ақ ұйымға тіркелгендігінен тұратыны мәлім болды. Бақылаусыз тіркелу қаупі төмендейді. Өтініштерді буферлеуде pre_registration_requests коллекциясы қолданылады. Өтініштер (REQUEST_UID) базалық деректермен және беру уақытымен тіркеледі. Пайдаланушы профилін бірден құрмай, өтінімдерді тексеріп, кейін өңдеуді өткізу мүмкіндіктері бар. Multi-tenant моделінің негізгі элементі - organizations коллекциясы екенін байқауға болады. ORG_UID құжаты ұйымның идентификаторын, атауын, белсенділік күйін және жаңарту уақыт белгілерін қамтиды. Сипаттамалар деректерді оқшаулап, қауіпсіздік ережелерін жеңілдетеді. Шлюз құрылғыларының қабаты бейне ағын көздерімен байланыстыру үшін бөлек қарастырылып отыр. Бұл camera_gateway_*collection group арқылы жүзеге асырылып, құрылғылардың метадеректері, камералар тізімі, байланыстыру параметрлері мен жұмыс күйі сақталады. Архитектура бағдарламалық контурды бейне ағынның физикалық көздерімен тиімді байланыстыруға мүмкіндік береді (3.5-сурет).



Сурет 3.5. Платформаға арналған Cloud Firestore ДҚ құрылымы

Ұйымдық контур ішінде операциялық ішкі коллекциялар орналасқан. people коллекциясы тұлға профильдерін (PERSON_UID) сақтайды, онда жеке атрибуттар мен белсенділік дәрежелері көрсетіледі; бұл идентификаторлар тану оқиғалары үшін бейтаныс кілті ретінде қолданылады. Деректердің өмірлік циклін дұрыс басқару үшін people_deleted (soft-delete) механизмі қолданылады:

мұнда жою белгілері мен сақтау саясаттары тіркеледі, бұл деректерді толық жоғалтпай, басқарылатын "тазалау" процесін қамтамасыз етеді. Бақылау және талдау нәтижелері `detection_events` коллекциясында сақталады: `EVENT_UID` құжатты адамға/камераға сілтемелерді, оқиға түрін, сенімділік мәнін (`confidence`) және уақыт белгісін (`timestamp`) қамтиды. Бұл аналитика мен есеп беру үшін аудит жүргізуге болатын журналды қалыптастырады.

Биометриялық қабат профильден бөлек ұйымдастырылған. `face_embedding` эмбеддингтер мен олардың өлшемін сақтайды. Белгілердің сапасы мен шығу тегі туралы метадеректерді (үлгілер саны, сілтемелер, қызметтік параметрлер) `biometrics` ішкі коллекциясы тіркейді. `registration_artifacts` ішкі коллекциясы тіркеу процесінің қайта жаңғыртылуын қамтамасыз етеді. Сапаны бақылау мен даулы жағдайларды талдау үшін өзекті. Бейтаныс тұлғалардың қайталануын болдырмау үшін `unknown_tracks` қолданған тиімді. Сенімді тректер бір адамның әртүрлі уақытта, тіпті әртүрлі камераларда пайда болуын жинақтайды. `ml_pipeline`-мен (модель нұсқалары/индекстері, нәтижелер) фондық есептеулерді басқару орындала алатындығымен анықталады. Мекеменің физикалық құрылымына тіркеу (`campus/building/floor/room` сатылық құрылымы және `hierarchyPath`) `campus_tree` арқылы іске асады.

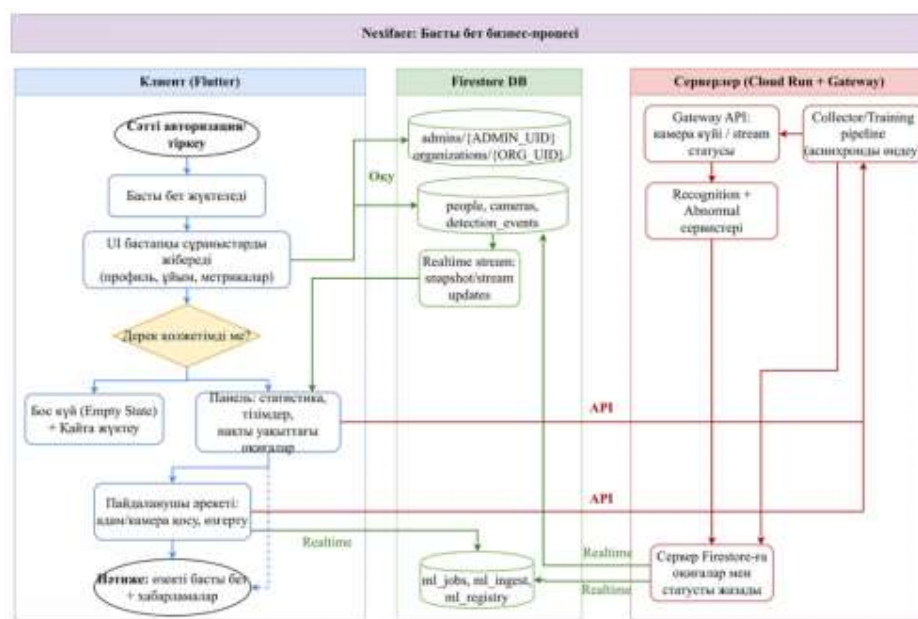
Схема ұйымдар бойынша деректерді оқшаулауды, оқиғаларды бақылау және тану нәтижелерін платформа интерфейсімен сенімді интеграциялануын қамтамасыз етеді.

3.6-суретте берілген схема веб-платформадағы `Firebase` инфра-құрылымының логикалық ұйымдастырылуы анықталды. Үш кілттік компоненттердің `Authentication`, `FirestoreDB` және `Hosting` өзара байланысын көруге болады. Диаграммасының жоғарғы бөлігінде `Firebase` бір уақытта пайдаланушыларды басқаруды, деректерді сақтауды және клиенттік қосымшаны жариялауды қамтамасыз ететін бірыңғай бұлттық орта ретінде көрсетілгенін байқауға болады. Мұндай тәсіл платформаның мынадай маңызды функцияларын орталықтандыруға мүмкіндік беретіні анықталды: аутентификация және қолжетімділікті басқару, пайдаланушы профильдерін қалыптастыру және сақтау, веб-хостинг арқылы интерфейстің қолжетімділігін іске асыру. Сурет пайдаланушы енгізген деректердің (мысалы, телефон нөмірі, бір реттік код) қалай расталған есептік жазбаға және ұйымның деректер базасындағы құрылымдалған жазбаға айналатынын түсіндіреді.

3.6-суретте келтірілген схема веб-платформада `Firebase` экожүйесі үш өзара толықтыратын компонентті біріктіретін бірыңғай бұлттық контур ретінде пайдаланатынын көрсетеді. Аталған үш компонент `Authentication` - пайдаланушыны сәйкестендіреді және растайды, `FirestoreDB` - профильге қатысты қолданбалы деректерді сақтайды, `Hosting` - веб-интерфейсті жариялап, жеткізу қызметін атқарады.

Hosting веб-қосымшаның қолжетімділігін сервис ретінде жүзеге асырады: әкімші жүйеге кіреді, профиль деректер базасында тіркеледі және интерфейс ары қарай жұмыс істеуге (платформаның объектілері мен оқиғаларын басқару) дайын болады.

3.7-суретте Nexiface басты парақшасының бизнес-процесі көрсетіледі. Үш өзара байланысты контур бір схемаға біріктірілген: клиенттік интерфейс, Firestore деректер базасы, Gateway қабатымен бірге контейнерлердегі серверлік сервистер (Cloud Run) .



Сурет 3.7. Nexiface басты парақшасы

Диаграммадан сәтті авторизациядан кейін пайдаланушы бастапқы сұраныстар (профиль, ұйым, негізгі метрикалар) арқылы интерфейснің бастапқы күйі қалай алатынын байқауға болады. Ары қарай экран динамикалық бақылау режиміне өтеді.

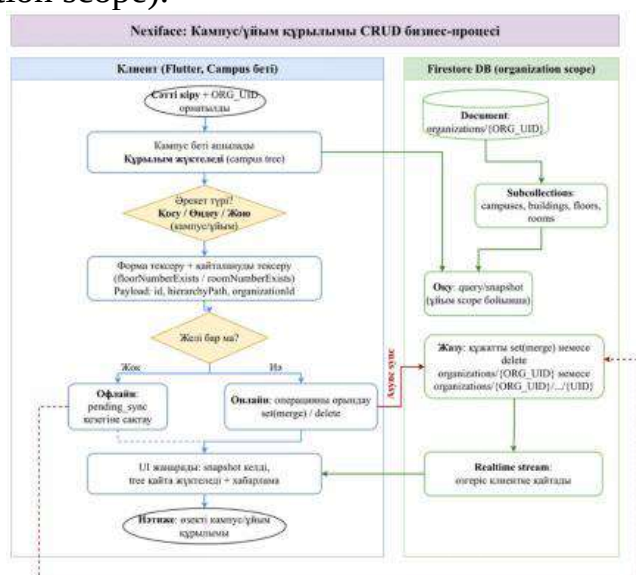
Клиенттік бөлік бастапқы жүктеудің үйлестірушісі ретінде әрекет ететіні анықталды. Ол кілттік құжаттарды (әкімші мен ұйым контексті) оқуды бастайды, деректердің қолжетімділігін тексереді және қажет болған жағдайда интерфейсін басқаратын empty state күйіне ауыстырады. Әрекет қайта жүктеу мүмкіндігін береді. Деректер сәтті алынғаннан кейін негізгі панель қалыптасады. Панельде статистика, объектілер тізімі және өзекті оқиғалар блоктары көрсетіледі. Олар парақшаны қолмен қайта жүктеусіз жаңарып отыруы қажет.

Схемада Firestore келесі екі функцияны атқарады: орталықтандырылған деректер қоймасы, нақты уақыттағы синхронизация арнасы. Әкімшілік құжаттар мен объектілерден (адамдар, камералар, оқиғалар) бөлек, ML-конвейерге қатысты жазбалар (тапсырмалар, ingest/registry операциялары) белгіленген. Бұл фондық өңдеу жағдайларын тіркеуге және оларды бақыланатын күй ретінде интерфейсте бейнелеу мүмкіндігін береді.

Серверлік контур функцияларына сәйкес бөлінеді. Gateway API камералар мен ағындардың күйін бақылайды, Recognition/Abnormal сервистері

бейнедеректерді құрылымдалған оқиғаларға түрлендіреді, Collector/Training pipeline асинхронды өңдеуді және артефакттарды жаңартуды жүзеге асырады. Маңызды ерекшелік - сервер нәтижелерді (оқиғалар мен күйлер) Firestore-қа жариялайды, ал клиент оларды нақты уақыттағы жазылу (realtime subscriptions) немесе командалық операциялар үшін API шақырулар арқылы алады. Қорытындысында басты парақша желілік кідірістерге төзімді және нақты уақыт режимінде мониторинг жүргізуге арналған операциялық панельге айналады.

3.8-суретте Nexiface платформасында кампус/ұйым құрылымын басқарудың бизнес-процесі CRUD (Create/Read/Update/Delete) форматында көрсетілді. Схема мына екі контурға бөлінген: әкімші локациялармен операцияларды бастайтын клиенттік интерфейс және деректердің реактивті жаңартылуын қамтамасыз ететін, өзекті күйінің бірыңғай көзі ретінде қызмет атқаратын Firestore DB (organization scope).



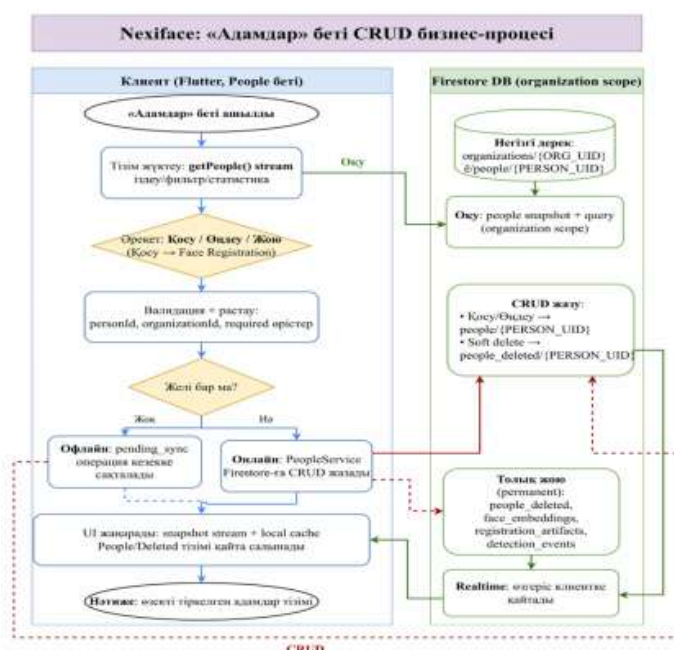
Сурет 3.8. Nexiface: ұйымдық контекст, офлайн-кезек және нақты уақыттағы синхронизацияны ескере отырып, кампус/ұйым құрылымын CRUD-операциялар арқылы басқару бизнес-процесі

Процесс пайдаланушы сәтті жүйеге кіргеннен кейін басталады, осы кезде ұйым идентификаторы (ORG_UID) бекітіледі және қолжетімді деректер шекарасы анықталады. Одан кейін клиент кампус құрылымының ағашын (campus tree) жүктейді → "кампус → ғимарат → қабат → бөлме" сатылық құрылымы, ол камераларды, оқиғаларды және аналитиканы нақты физикалық аймақтармен дұрыс байланыстыру үшін қажет. Пайдаланушы әрекет түрін таңдайды (қосу/өзгерту/жою), содан кейін қолданбалы валидация орындалады: форма тексеріледі және бірегейлік бақыланады (мысалы, бір түйін ішінде қабат немесе аудитория нөмірінің қайталануына жол берілмейді). Осы кезеңде унификацияланған payload қалыптасады, оның құрамына: объект идентификаторы, hierarchyPath контекст кіреді. Бұл жазбалар бірімділігін және ағаш құрылымының келісімділігін қамтамасыз етеді.

Процестегі маңызды тармақ - желінің қолжетімділігі. Онлайн режимде операция бірден Firestore-да орындалады: set(merge) арқылы (құжатты толық

кайта жазбай, нүктелік жаңарту), немесе delete арқылы. Офлайн режимде әрекет жоғалмайды - ол pending_sync кезегінде сақталады. Байланыс қалпына келгенде, асинхронды синхронизация іске қосылып, сол операциялар бұлттағы деректерге қолданылады. Жазба сәтті орындалғаннан кейін Firestore realtime stream (snapshot жаңарту) іске қосады, соның нәтижесінде интерфейс автоматты түрде ағашты жаңартып, операцияның орындалғанын көрсетеді. Нәтижесінде жүйеде кампус/ұйымның өзекті, қайшылықсыз құрылымы қалыптасады, ол платформаның келесі сценарийі үшін (камераларды тағайындау, мониторинг, есеп беру және навигация) дайын болады.

3.9-суретте Nexiface платформасында "Адамдар" (People) объектісін CRUD-операциялар арқылы басқарудың бизнес-процесі көрсетілген. Диаграмма екі жауапкершілік контурын ажыратады: People бетінің клиенттік интерфейсі (тізімді басқару, іздеу/сүзгілеу, операцияларды бастау), Cloud Firestore (organization scope), мұнда деректер тек таңдалған ұйым шегінде сақталып, синхронизацияланады. Схема персоналдық профильдермен жұмыс істеудің толық өмірлік циклін көрсетеді: тізімді ағындық режимде оқу, қосу/өңдеу, өмірлік циклді бақылау арқылы жою, офлайн сценарийлерді өңдеу, деректер базасында өзгерістер тіркелгеннен кейін интерфейсін автоматты жаңартылуы.



Сурет 3.9. Nexiface: "Адамдар" бетінде CRUD-операцияларды басқару бизнес-процесі (ұйымдық контекст, офлайн-кезек, soft delete және realtime жаңарту)

Процесс "Адамдар" парақшасын ашудан басталады. Клиент getPeople() stream-ге жазылады және бір уақытта іздеу, сүзгілеу, статистика функцияларын қолдайтыны анықталды. Оқудың ағындық моделі тізімді жедел көрсетуді қамтамасыз етеді және өзгерістерді қолмен жаңартусыз бейнелеу мүмкіндігін береді. Жаңа профиль пайда болғанда, деректер өзгертілгенде немесе статус жаңартылғанда, жазба snapshot бірден көрінетіні белгілі болды.

Келесі қадамда пайдаланушы Қосу/Өңдеу/ Жою әрекетін таңдайды:.. Қосу және редакциялау үшін валидациялау және payload дайындау (personID,organizationID және міндетті өрістер) қарастырылды. Бұл жазбалардың сәйкессіздігін болдырмайды және профильді ұйымдық контурмен дұрыс тіркеуді қамтамасыз етеді. Қосу процесі Face Registration процедурасымен тығыз байланысты. Nexiface жүйесінде профиль тек анкета ғана емес, идентификацияның субъектісі болатыны анықталды. Кейінгі тану процестері биометриялық деректер мен тіркеу артефактілеріне сүйенеді.

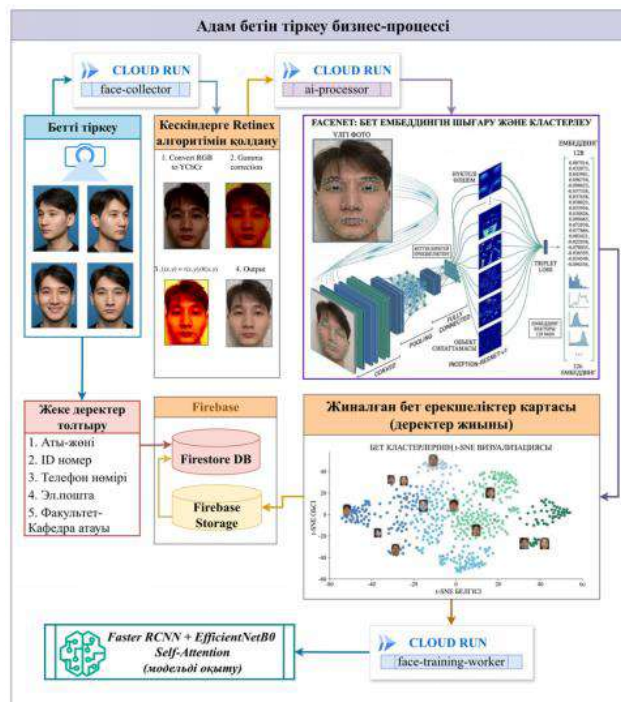
Кілттік нүкте – желіні тексеру. Байланыс болмаса операция pending_sync орналастырылып, байланыс қалпына келгенге дейін семантикасын сақтайды. Онлайн режимде сервис қабаты Firestore-ға жазба жасайды. people/{PERSON_UID} ішінде профиль құрылады немесе жаңартылады, ал жою soft delete арқылы жүзеге асырылады (people_deleted/{PERSON_UID} көшіру). Тәсіл басқарылатын өмірлік циклды және реттелген сақтау мүмкіндігін қолдайды. permanent delete сценарийі бөлек көрсетіледі, онда байланысты объектілер (эмбеддингтер, registration_artifacts және релевантты жазбалар) тазартылады, бұл "жетім" деректерді болдырмайды.

Операцияны қолданғаннан кейін Firestore realtime – жаңартуды іске қосады. UI snapshot stream және жергілікті кэш негізінде People/Deleted тізімдерін қайта жинайды. Процестің қорытындысы – биометриялық контурмен келісілген және кейінгі идентификацияның модульдеріне, бақылау және оқиғалық аналитикаға жарамды ұйым адамдарының өзекті реестрі.

3.10-суретте Nexiface жүйесінде адамды тіркеудің толық конвейері қорытындыланған – бастапқы бет кадрларын алудан бастап, оларды ары қарай идентификациялау мен аналитика үшін жарамды деректерді дайындауға дейін. Процесс түсіру және бастапқы верификация кезеңінен басталады. Пайдаланушы белгілері бір әрекеттерді (басын әртүрлі бұру, бейтарап мимика сақтау) орындайды. Бұл ракурстардың әртүрлілігін арттырады және биометрияның дене қалпы мен жартылай жабылуларға тұрақтылығын қамтамасыз етеді.

Ары қарай Retinex фотометриялық нормализация блогы көрсетіледі. Ол жарықтандыру мен контрастты теңестіру үшін қолданылады. Рөлі - нақты бейнебақылау жағдайында (артық жарық, көлеңке, біркелкі емес жарық) кескін сапасын тұрақтандыру. Бұл кезеңде кадрлар біркелкі көрініске келтіріліп, әртүрлі камералар мен локациялар арасында айырмашылықтар азайтылады.

Келесі кілттік модуль - FaceNet тектес нейрондық желі арқылы бет эмбеддингтерін шығару. Эмбеддинг - тұлғаның бірегей ерекшеліктерін сипаттайтын ықшам сандық вектор. Ол тұлғаларды метрикалық ұқсастық бойынша салыстыру және тез іздеу индекcін құру үшін қолданылады. 3.10-суретте сапаны бақылау құралы сияқты кластерлеу/визуализация (t-SNE) идеясы жеке көрсетіледі. Ол бір адамның белгілері үйлесімді түрде топтастырылғанына және әртүрлі беттер кеңістікте бөлінгенін тексеруге көмектеседі.



Сурет 3.10. Nexiface: "Адам/бет тіркеу" бизнес-процесі (профильді тексеру, биометрия жинау, транзакция сақтау, rollback-тазалау және өңдеу кезегіне тапсырма қою)

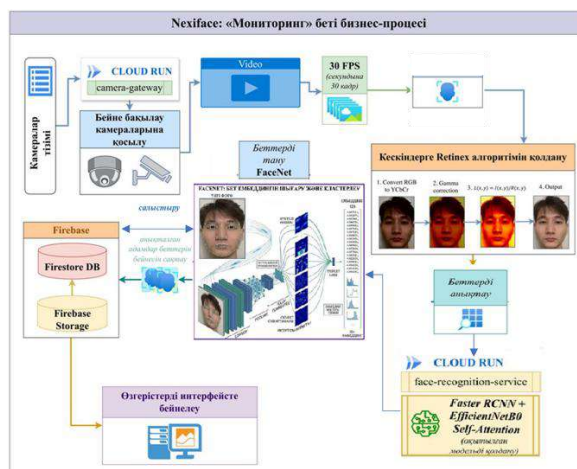
Сонымен қатар қолданбалы сақтау контуры қалыптасады: Firestore профильді және қызметтік күйлерді сақтайды, ал Storage медиа-артефакттарды (фото/бейне/алдын ала өңдеу нәтижелері) сақтауға жауап береді. Қорытынды кезең - фондық есептеу контурына (Cloud Run worker) тапсырма жіберу, мұнда индекстер жаңартылады және тану модельдері қайта оқытылады немесе өзектендіріледі. Осылайша, тіркеу процесі жай ғана "сурет жүктеу" емес, деректер сапасын, қадағалануын (трассируемость) және кейінгі қолдануға дайындықты қамтамасыз ететін толыққанды өңдеу тізбегіне айналады.

3.11-суретте Nexiface жүйесіндегі "Мониторинг" бетінің жұмыс логикасы көрсетілген. Бұл бет камералары нақты уақыт режимінде бақылау және бейне ағынды интеллектуалды өңдеу негізінде оқиғаларды жедел қалыптастыруға арналған.

Процесс камералар тізімінен деректер көзін танудан басталады. Одан кейін camera-gateway сервисі (Cloud Run) арқылы бейнағын арнасына қосылу жүзеге асырылады және кадрларды тұрақты жеткізу контуры қалыптасады. Схемада секундына шамамен 30 кадр (30 FPS) жиілігі ерекше көрсетілген, бұл жылдам өзгерістерді анықтауға және қысқа уақыттық эпизоттарды дұрыс тіркеуге жеткілікті уақыттық дәлдік береді.

Кейін бейнағын екі параллель есептеу бағытына бөлінеді. Бірінші бағыт - беттерді тану және сәйкестендіру: кадрлар face-recognition-service (Cloud Run) сервисіне жіберіледі, мұнда беттер анықталып, белгілер (эмбеддингтер) шығарылады және тіркелген профильдермен салыстырылады. Екінші бағыт - агрессивті/аномалиялық әрекеттерді анықтау. Бейне ағын abnormal-action-service (Cloud Run) сервисінде талданады, ол мінез-құлық үлгілерін (мысалы, төбелес,

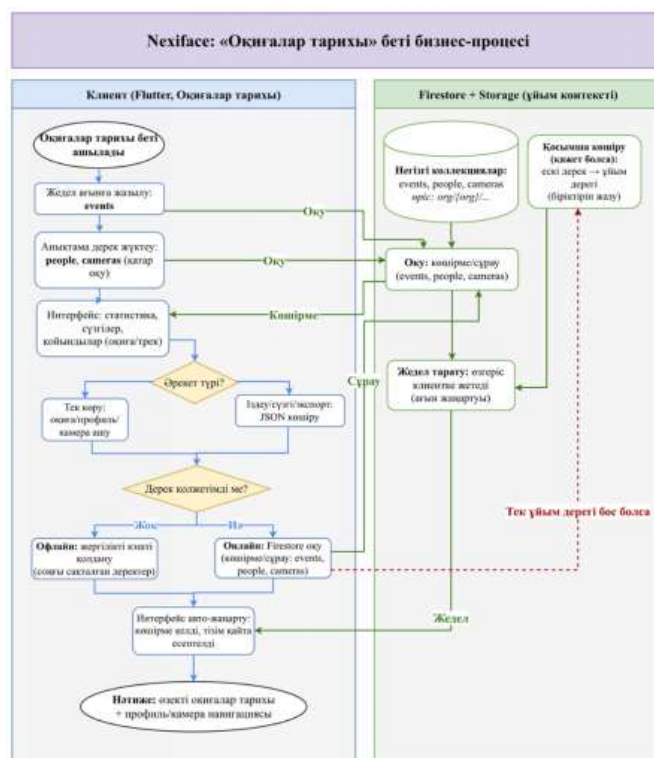
буллинг) классификациялап, сенімділік шегі асқан жағдайда оқиға қалыптасады. Бұл екі контурлы архитектура бір уақытта "кадрда кім бар", "кадрда не болып жатыр" деген екі маңызды сұраққа жауап береді. Сонымен қатар клиенттік интерфейске артық есептеу жүктемесін түсірмейді.



Сурет 3.11. Nexiface: "Мониторинг" бетіннің бизнес-процесі (бейне ағынды қосу, беттерді және агрессивті әрекеттерді онлайн-инференс арқылы тану, нәтижелерді Firebase-те сақтау)

Қорытынды шығару нәтижелері (оқиға метадеректері, камера/локация идентификаторлары, сенімділік мәндері, мәртебелер) Firestore DB-ға жазылады, ал медиа-артефакттар (кадрлар, бейне үзінділері, бейтаныс тұлғалардың суреттері) Firestore Storage-та сақталады. Деректер тіркелгеннен кейін өзгерістер интерфейске realtime – жаңарту арқылы таралады, соның арқасында оператор өзекті оқиғалар мен күйлерді іс жүзінде кідіріссіз көреді. Нәтижесінде "Мониторинг" парақшасы жүйенің операциялық қабаты ретінде қызмет атқарады. Қайталанатын бақылау тарихын және ыңғайлы басқаруды қамтамасыз ете отырып бейне ағынды, жасанды интеллект талдауын және орталықтандырылған сақтауды біріктіреді.

3.12-суретте Nexiface жүйесінде "Оқиғалар тарихы" парақшасының бизнес-процесі көрсетілген. Парақша ұйымдағы барлық оқиғалардың бірыңғай лентасын қалыптастыруға және оларды аналитикалық қарауға арналады. Схема екі үйлесімді контурдан тұрады. Контурлар навигация, визуализация және пайдаланушы әрекеттеріне жауап беретін клиенттік интерфейсден және оқиғалар журналын оқуды, байланысты контентті жүктеуді және деректердің ағындық жеткізілуін қамтамасыз ететін Firestore + Storage контурдан тұрады. Парақшаның кілттік ерекшелігі - оперативті режиммен (өзектілікке бағытталған) аналитикалық режимнің (сүзгілеу, таңдау және экспортқа бағытталған) біріктірілуі. Сондықтан процесте ағындық жазылымдар мен сұраныс операциялары бір уақытта пайдаланылатыны анықталды.



Сурет 3.12. Nexiface: «Оқиғалар тарихы» парақшасының бизнес-процесі

Сценарий парақшаны ашудан басталады. Интерфейс events ағынына қосылып, жаңа жазбаларды пайда болған сәтте алады. Бұл бейнебақылау жүйелері үшін аса маңызды, мұнда оқиғалар үздіксіз қалыптасып отырады. Параллель түрде people және cameras анықтамалық деректері жүктеледі. Деректер оқиғаларды дұрыс интерпретациялау мүмкіндігін береді. Оқиғалар әдетте адамға немесе камераға сілтемелерден (идентификатор) тұрады, Нәтижесінде лента келесі толық контекстпен көрсетіледі: аты, рөлі, камера түрі, орналасқан орны.

Ары қарай интерфейстің аналитикалық қабаты құрылады. Статистика, сүзгілер, бөлімдер және басқару элементтері қосылады. Бұл журналды жай «шикі тізімнен» толыққанды талдау құралына айналдырады. Пайдаланушы қарау сценарийін таңдай алады (оқиға карточкасын ашу және адамның профиліне немесе камераға өту), болмаса іздеу/сүзгілеу/экспорт жасау жоспарын таңдайды. Мұнда мақсаттық таңдау қалыптастырылады және қажет болса, бастапқы деректерді өзгертпестен құрылымдалған жүктеу (мысалы, JSON-көшірме) құрылады.

Схеманың жеке бөлігі деректердің қолжетімділігін тексеруге арналады. Желінің тұрақсыздығы кезінде интерфейс жұмыстың үздіксіздігін және базалық навигацияны сақтай отырып, соңғы сақталған жазбалардың жергілікті кэшін қолданады. Желі қолжетімді болған жағдайда Firestone-ға (events/people/cameras) өзекті сұраныстар орындалып, интерфейс автоматты түрде деректерді қайта есептеп, көріністерді жаңартады. Ұйымдастырушылық контурда сирек қолданылатын сценарий де қарастырылған – деректер болмаған жағдайда

қосымша көшіру жүргізіледі. Бұл басқарылатын процесс болып табылады және ағымдағы деректер тұтастығын бұзбайды.

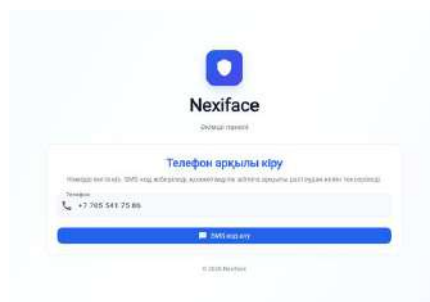
Процесс нәтижесінде пайдаланушы өзекті оқиғалар лентасын алады, байланысты объектілерге жылдам өтеді және аналитикалық мүмкіндіктерді пайдалана алады. Бұл бетті күнделікті мониторингке де, есеп беру мақсаттарына да тиімді құрал етеді.

3.13-суретте Nexiface әкімшілік веб-платформасын кірудің бастапқы экраны көрсетілген, онда пайдаланушы телефон нөмірі арқылы аутентификациядан өтіп, кейін SMS-код арқылы растайды.

автоматты түрде деректерді қайта есептеп, жаңартады. Ұйымдастыру деңгейде сирек қолданылатын сценарий де қарастырылған – деректер болмаған жағдайда бастапқы көшіру (инициализация) жүргізіледі. Бұл басқарылатын процесс болып табылады және ағымдағы деректер тұтастығын бұзбайды.

Процесс нәтижесінде пайдаланушы өзекті оқиғалар лентасын алады, байланысты объектілерге жылдам өтеді және аналитикалық мүмкіндіктерді пайдалана алады. Бұл бетті күнделікті мониторингке де, есеп беру мақсаттарына да тиімді құрал етеді.

3.13-суретте Nexiface әкімшілік веб-платформасын кірудің бастапқы экраны көрсетілген, онда пайдаланушы телефон нөмірі арқылы аутентификациядан өтіп, кейін SMS-код арқылы растайды.



Сурет 3.13. Nexiface: телефон нөмірі арқылы кіру экраны (SMS-код сұрау және интерфейстің көптілді қолдауы)

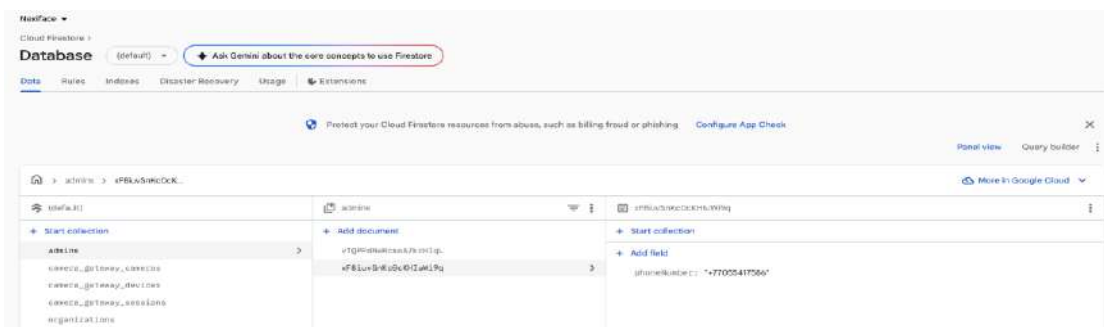
Бұл экран жүйеге кірудің бірыңғай «кіру нүктесі» ретінде қызмет атқарады және сессияның бастапқы иелік ететінін растайды, содан кейін жүйе оны профильмен және ұйымдық қолжетімділік құқықтарымен байланыстыра алады.

Осы себепті интерфейс әдейі минималистік түрде жасалған және пайдаланушыны тексеру аяқталғанға дейін деректерді басқару функцияларын қамтымайды.

Визуалдық композиция орталық блокқа — платформа логотипті мен атауына — негізделген. Бұл сервисті дұрыс тануға көмектеседі және қате кіру ықтималдығын төмендетеді.

KAZ/RUS/ENG тілдерін ауыстыру батырмасы жоғарғы оң жақ бұрышта орналасқан.

Форманың функционалдық бөлігі халықаралық форматтағы телефон нөмірін енгізу, SMS-код сұрау батырмасынан («SMS код алу») тұрады. Процестің айқындығы мен басқаруын қамтамасыз ете отырып және пайдаланушының когнитивтік жүктемесін азайтады. Мұндай құрылым «енгізу → код алу → растау») сценарийінің логикалық реттілігін сақтайды 3.14-суретте Cloud Firestore консолінің фрагменті көрсетілген. Ол тіркелген телефон нөмірі арқылы жүйеге рұқсат беру механизмінің практикалық іске асырылатынын бейнелейді.



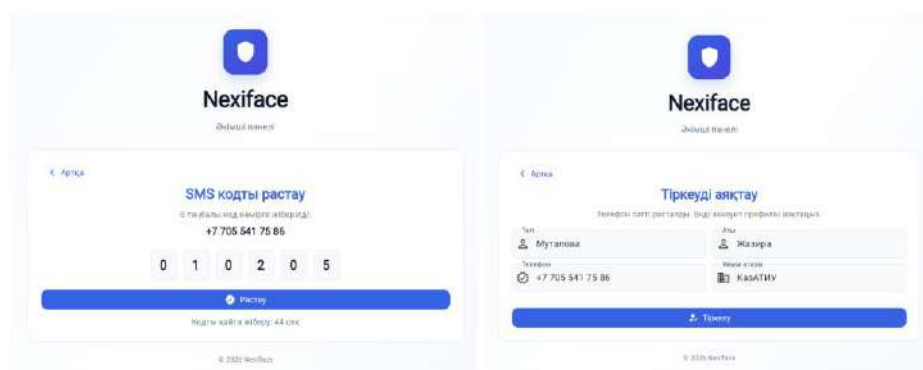
Сурет 3.14. Cloud Firestore-тағы жазбаның мысалы ретінде: авторизацияға рұқсат ететін басты атрибут әкімші құжаты ретінде көрсетілген телефон нөмірі

Firestore интерфейсында әкімшілер коллекциясы көрсетіледі. Мұнда әрбір құжат белгілі бір есептік жазбаға сәйкес келеді. Сондай-ақ негізгі идентификациялаушы атрибут ретінде `phoneNumber` өрісі қолданылған. Деректерді ұйымдастыру келесі ережені формализациялауға мүмкіндік береді. Платформаға кіру және авторизация тек қана алдын ала деректер базасында тіркелген, белсенді қолданылатын телефон нөмірлері үшін ғана мүмкін болады. Бұл басқарылатын қолжетімділік моделін қолдайды.

`Admins` коллекциясы ерекше белгіленген коллекциялар тізімі консольдің сол жақ панелінде көрсетілген. Осы коллекцияның құжаттар тізімі (бірегей жазба идентификаторлары) орталық бөлікте көрсетілген. Ал оң жақ панельде таңдалған құжаттың мазмұны көрсетіледі. Ұсынылған мысалда жүйе тіркеуге және кейінен авторизациялауға жарамды нөмір ретінде пайдаланатын мәні бар `phoneNumber` өрісі берілген. Практикалық жағынан осындай құжат тек телефон нөмірін ғана емес, сондай-ақ профильдің қосымша атрибуттарын да қамтиды. Олар: пайдаланушының рөлі, ұйымға сілтеме, уақыт белгілері. Осы атрибуттар сәтті верификациядан кейін құқықтарды нақты бөлуге мүмкіндік береді.

3.14-суретте көрсетілгендей аталған экран телефон нөмірі арқылы кіру моделі басқарылатын деректер қоймасына негізделген. Жүйе алдымен SMS-код арқылы нөмірді растайды, содан кейін оны Firestore-тағы әкімші жазбасымен сәйкестігін тексереді. Бұл платформа функционалына қауіпсіз және бақыланатын қолжетімділікті қамтиды.

3.15-суретте көрсетілгендей Nexiface әкімшілік веб-платформасына телефон нөмірін енгізгеннен кейінгі екі негізгі кезеңі бар. Біріншісі бір реттік кодты (OTP) растау. Екіншісі пайдаланушы профилін бастапқы тіркеуді аяқтау. Бұл сценарий үнемі басқарылатын қолжетімділік моделін жүзеге асырады. Мұнда телефон нөміріне иелік ету SMS арқылы расталады. Ал пайдаланушының кейінгі әрекеті жүйеде бұрын тіркелген есептік жазбаның бар-жоқтығына байланысты анықталады.



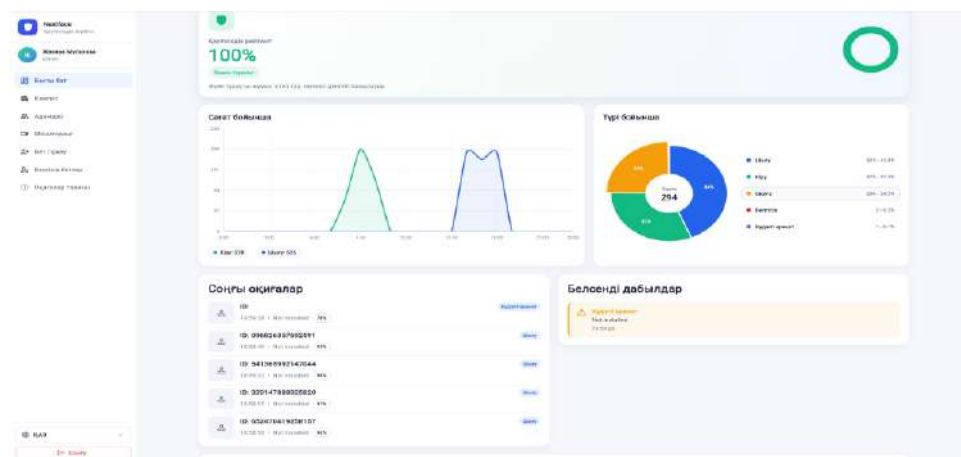
Сурет 3.15. Nexiface: OTP-кодты растау және алғашқы кіру кезінде профильді алғашқы тіркеуді аяқтау

3.15-суреттің сол жақ бөлігінде верификация экраны көрсетілген: пайдаланушыға 6 таңбалы OTP-код жіберілген телефон нөмірі, сол кодты енгізуге арналған форма ұсынылады. Код енгізілгеннен кейін жүйе оны дұрыстығын тексереді. Бұл телефон нөміріне иелік етуді криптографилық тұрғыдан растау болады. Белсенді сессия сәтті верификация жағдайында бұрын тіркелген пайдаланушы үшін қалыптасады. Сол сияқты қосымша әрекеттерсіз платформаның негізгі интерфейсіне өту іске асырылады.

3.15-суреттің оң жақ бөлігінде пайдаланушы жүйеде алғаш рет тіркелген кездегі «алғашқы кіру» сценарийі көрсетілген. Осы жағдайда пайдаланушыға минималды қажет деректерді енгізу арқылы профильді тіркеуді аяқтау сұралады. Олар: аты, тегі, расталған телефон нөмірі (алдыңғы қадамда енгізілген) және ұйымның атауы. Көрсетілген өріс жиынтығы деректер әкімшілік профильді дұрыс қалыптастыруға және пайдаланушыны белгілі бір ұйым контекстіне байланыстыруды іске асуын қамтамасыз етеді. Деректерге қолжетімділікті (камералар, оқиғалар, пайдаланушылар) кейіннен дұрыс ажырату үшін қажет.

3.15-суретте көрсетілген механизм қауіпсіз және басқарылатын аутентификацияның орындалуын жүзеге асырады. Оған телефон нөмірін растау – жүйеге кірудің міндетті шарты, ал «бар пайдаланушы / жаңа пайдаланушы» тармақталуы. Бұл жүйеге тек дұрыс анықталған және контекстке байланысты есептік жазбалардың ғана енуін жүзеге асырып отырады.

3.16-суретте Nexiface әкімшілік панелінің басты беті көрсетілген. Бұл экран пайдаланушы алғашқы тіркеуді аяқтағаннан, сәтті авторизациядан өткен соң автоматты түрде ашылады. Бұл жедел басқарудың орталық нүктесі болады. Экран ұйымның негізгі көрсеткіштерін біріктіреді. Жүйенің басты модульдеріне жылдам қолжетімділікті қамтамасыз етеді. Сол жақ навигациялық мәзір арқылы келесі бөлімдерге өтуге болады. Кампус құрылымды, адамдардың тізімі, мониторинг, бетті тіркеу, белгісіз тұлғалар, оқиғалардың тарихы және т.б. Бұл интерфейс пайдаланушы тұлғаға жүйенің барлық негізгі деген функцияларын бір орталықтан басқа алуға мүмкіндік береді. Және уақыт режимінде деректерді бақылауды қамтамасыз ете алады.

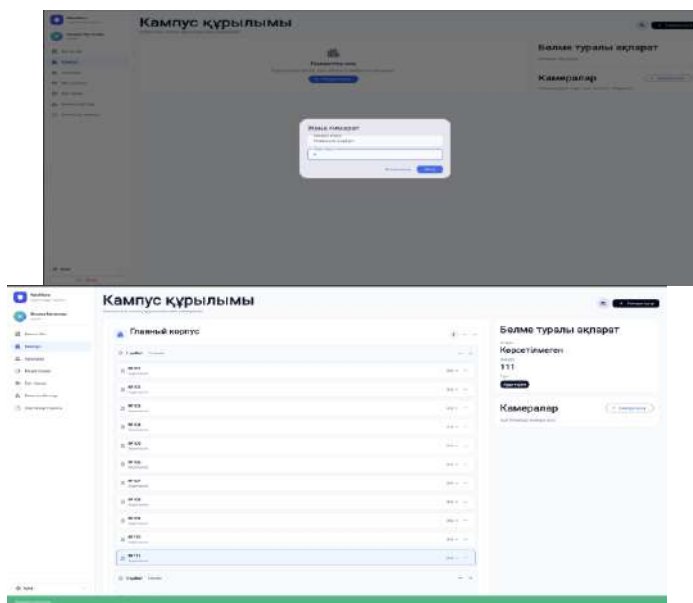


Сурет 3.16. Nexiface: әкімшілік панельдің негізгі экраны. Басты КРІ көрсеткіштері, жүйенің ағымдағы күйі мен ұйым модульдерінің бөлігінің навигациясы

Интерфейстің жоғарғы жағында қауіпсіздіктің жиынтық көрсеткіші мен платформаның қазіргі кездегі күйі бейнеленеді. Бұл жүйенің жалпы тәуекел деңгейін және ішкі жүйелердің дайындығы бар-жоқтығын әкімші тез бағалай алуға жағдай жасайды. Төменгі жағында аналитикалық виджеттер орналасқан. Олар уақыт бойынша оқиғалар динамикасы (белсенділік графиктері), оқиғалардың түрлері бойынша тарала алуы (дөңгелек диаграмма). Көрсетілген құралдар бейнебақылау ағындары мен интеллектуалды өңдеу нәтижелерін базалық деңгейде талдау жасай алуға жағдай жасайды.

«Соңғы оқиғалар» мен «Белсенді дабылдар» блоктар жедел журнал қызметін атқара отырады. Олар соңғы тіркелген оқиғаларды, олардың идентификаторларын және ықтималдық бағаларын көрсетеді. Сондай-ақ аса назарды қажет ететін күмәнді немесе маңызды хабарламаларды бөлек-бөлек көрсетеді. Бұл экран әкімші үшін «бір терезе» қағидатын іске асырып отырады. Статистика, ағымдағы инциденттер және навигация бір жерде жинақталады. Бұл жүйені басқаруын арттырады, оқиғаларға реакцияны тездетеді.

3.17-суретте «Кампус құрылымы» модулі көрсетілген. Модуль ұйымның орналасқан жерінің иерархиялық моделін қалыптастырады (корпус → қабат → бөлме). Бұл модель камераларды кейіннен байланыстыруға және бақылау оқиғаларын дұрыс түсіндіру үшін кеңістіктік контекст орнатады. Бұл тәсіл мекеменің физикалық инфрақұралымын формалданған деректер құрылымына ауыстырады. Навигация біркелкілігін, аймақтарды талдауды жеңілдейді, оқиғаларды нақты бөлмелермен салыстыруды қамтамасыз етіп отырады.

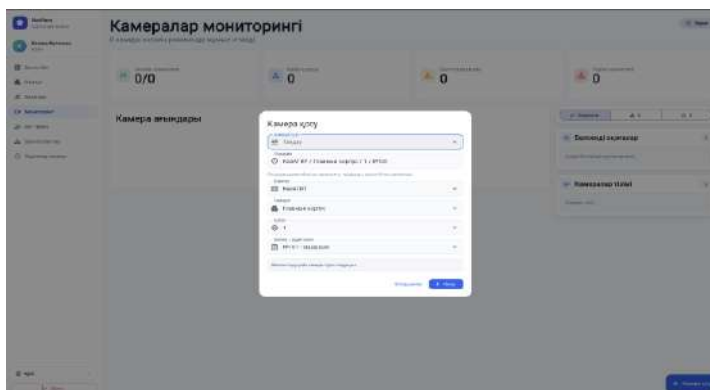


Сурет 3.17. Nexiface: «Кампус құрылымы» модулі (корпусты қосу, қабаттарды автоматты түрде құру. Камераларды кейіннен тағайындау үшін бөлмелер иерархиясын басқару)

Бірінші терезеде корпус диалогтық терезе арқылы корпусты қосу оқиғалар тізбегі көрсетілген. Әкімші корпус атауын және қабаттар санын орнатады. Кейін жүйе құрылымының негізгі «қаңқасын» автоматты түрде дайындайды. Бұл алғашқы баптаудың еңбек сыйымдылығын азайтады. Тағы да қолмен енгізу кезіндегі қателер мүмкіндіктерін төмендетеді. Екінші кескінде қалыптасқан бөлмелер ағашы және таңдалған аудиторияға арналған контекстік панель көрсетіліп тұр. Осы панельдің арқасында камера қосу операцияларын орындау мүмкіндігі болады. «Локацияны сипаттау → бөлмені таңдау → бақылау жүйесін қосу» сияқты толық сценарий жүзеге асырылады (3.17-сурет).

3.18-суретте ұйым құрылымындағы нақты локация (аудитория, дәліз, кіру/шығу аймағы) таңдалады. Кейін іске қосылатын камера қосу диалогтық сценарийі көрсетілген. Бұл интерфейс объектіні контекстке байланысты құру қағидатына негізделген. Камера «бос кеңістікке» емес, алдын ала анықталған нақты аймаққа қосылады. Бұл тәсіл камераның кеңістіктік байланысын дұрыс орнатуға және конфигурация қателерінің мүмкіндіктерін азайтуға жол береді. Әкімші камераны таңдалған бөлме аясында қосады. Ал жүйе ұйымының анықтамалық деректеріне сүйене отырып негізгі өрістерді автоматты түрде толтырады.

3.18-суреттің орталық элементі – «Камера қосу» модельді терезесі болады. Мұнда камера түрін таңдау және орналасуын баптау параметрлері қарастырылған. «Локация» өрісі орналастыру жолын толық көрсетеді (мысалы: ұйым → корпус → қабат → аудитория). Камераны құру операциясы бұрын таңдалған инфрақұрылым нүктесінің контекстінде орындалатынын дәлелдейді.



Сурет 3.18. Nexiface: таңдалған аудитория/аймақ және ұйымның анықтамалық деректерді негізге ала отырып локацияны автоматты түрде толтырып камера қосу

Төменде иерархияны егжейлеуге арналған ашылмалы тізімдер берілген (кампус, ғимарат, қабат, бөлме/аудитория). Бұл ұйым деректер базасында сақталған ақпарат негізінде, өрістер пайдаланушының ағымдағы таңдауымен автоматты түрде толтырылады. Қажет болған жағдайда оларды қолмен түзетуге болады. Маңыздылығы камера екі аймақтың шекарасында орналасқан немесе әкімшілік модельді нақтылау қажет болған жағдайларда.

Әдіснамалық тұрғыдан бұл процесс деректердің келісімділігін жүзеге асырады. Камера құру кезінде локация идентификаторларымен байланысады. Мониторинг панельдері дұрыс қалыптасады, оқиғаларды бөлмелер бойынша сүзу мүмкіндігі болады, келушілер ағыны бойынша аналитика құру жеңілдейді. Камера түрін таңдау маңызды параметр болады. Себебі ол жүйеде тіркелетін оқиғалардың семантикасын анықтайды (мысалы, кіру камерасы, шығу камерасы немесе аудитория ішіндегі бақылау камерасы). Терезенің соңғы элементтері бұл растау және бас тарту әрекеттері. Операцияны басқарылатын түрде бекітуді қамтамасыз етеді. Пайдаланушының нақты растауынан кейін ғана камера құрылады.

3.18-суретте көрсетілген сценарий бейнебақылау жүйесін конфигурациялаудың архитектуралық жағынан дұрыс тәсілін көрсетеді. Әкімші камераны нақты локация контекстінде қосады. Ал жүйе кампус құрылымын пайдаланып параметрлерді автоматты жағдайда толтырады. Бұл оқиғаларды талдаудың сапасын жақсартып, локациямен байланыс дәлдігін арттырады, .

3.19-суретте Nexiface жүйесінде бақылау камераларын іске қосудың унификацияланған сценарийі кескінделген. Сценарий бірыңғай форма арқылы іске асырылады және бейне көздерінің үш түрлі санатын қолдайды. Олар IP-камера, шлюздік (Gateway) камера, жергілікті камера. Қосылу әдістерінің айырмашылығына қарамай, барлық нұсқалар бірдей конфигурация қағидатына негізделген. Камера контекстте алдын ала таңдалған локация жасайды (кампус → корпус → қабат → бөлме). Кейін ұйымның кеңістіктік құрылымына оқиғаларды бақылау, тану және байланыстыру үшін қажет деген техникалық сипаттамасын ала алады. Бұл тәсіл деректердің салыстырмалы және аппараттық деректер түріне қарамай оқиғаларды интерпретацияның бірыңғай ережелерін жүзеге асыра алады.

The image displays three sequential screenshots of the 'Камера қосу' (Add Camera) interface in the Nexiface system, illustrating the configuration process for different camera types. Each screen shows a hierarchical selection of location (Location, Campus, Building, Floor, Room) and camera details (Camera type, Camera name, Camera ID, Camera URL, PIN code, Gateway configuration, Camera ID/Code, Gateway camera name, Camera type, and Stream URL).

- Left Screenshot (Шлюз):** Shows the configuration for a Shutter camera. The location is set to 'KazATIY / Main building / 1 / NP101'. The camera type is 'KazATIY'. The camera name is 'Главный корпус'. The camera ID is '1'. The camera URL is 'NP101 - classroom'. The PIN code is '101_1'. The Gateway configuration is 'Gateway configuration'. The Camera ID/Code is 'KazATIY ID / code'. The Gateway camera name is 'Gateway camera name'. The Camera type is 'Auditorium (OBSERVATION)'. The Stream URL is 'Stream URL (optional)'.
- Middle Screenshot (IP-камера):** Shows the configuration for an IP camera. The location is set to 'KazATIY / Main building / 1 / NP101'. The camera type is 'KazATIY'. The camera name is 'Главный корпус'. The camera ID is '1'. The camera URL is 'NP101 - classroom'. The PIN code is '101_1'. The Gateway configuration is 'Gateway configuration'. The Camera ID/Code is 'KazATIY ID / code'. The Gateway camera name is 'Gateway camera name'. The Camera type is 'Auditorium (OBSERVATION)'. The Stream URL is 'Stream URL (optional)'.
- Right Screenshot (Локалды камера):** Shows the configuration for a Local camera. The location is set to 'KazATIY / Main building / 1 / NP101'. The camera type is 'KazATIY'. The camera name is 'Главный корпус'. The camera ID is '1'. The camera URL is 'NP101 - classroom'. The PIN code is '101_1'. The Gateway configuration is 'Gateway configuration'. The Camera ID/Code is 'KazATIY ID / code'. The Gateway camera name is 'Gateway camera name'. The Camera type is 'Auditorium (OBSERVATION)'. The Stream URL is 'Stream URL (optional)'.

Сурет 3.19. Nexiface-тің үш камера түрін қосу. Олар бірыңғай және типтік конфигурация параметріне байланысты (IP-камера, Gateway, жергілікті камера) болады

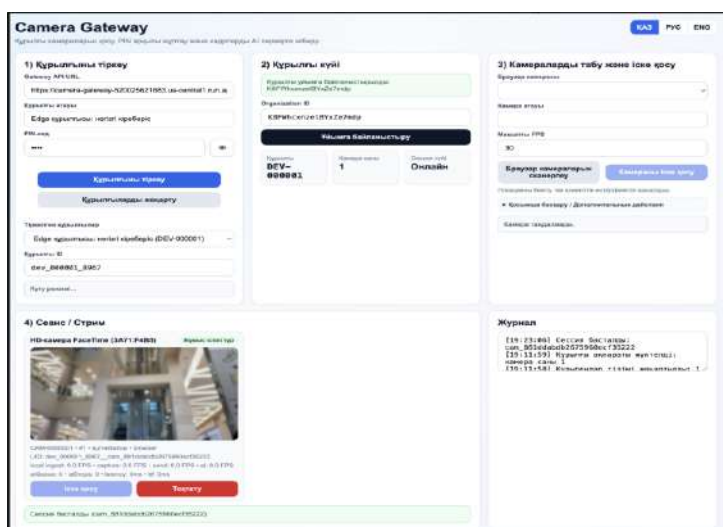
Gateway (шлюз) нұсқасы бейне ағын бір ғана емес бірнеше камераны біріктіруге мүмкіндік беретін құрылғы-шлюздан келе отыратын жағдайларға арналған. Оларға қорғалатын қызмет арқылы қол жеткізе алуға болады. Формада сервис адресі (Gateway URL), байланыстыру үшін арналған PIN-код, нақты таңдала алған құрылғы, қолжетімді болған камералар тізімін алу шлюз параметрлері көрсетіледі. Камера таңдалған соң рөлі (мысалы, аудиториядағы бақылау жүргізу) беріледі. Бұл режим басқаруды масштабтауға және орталықтандыруға ыңғайлы болады. Себебі бір шлюз бірнеше камераға қызмет көрсете алады.

Ұйымның инфрақұрылымының төңірегінде IP-камера нұсқасы желілік құрылғыны тікелей түрде іске қосуға есептелген. Бұл жағдайда форма камераның атауының өрісін, IP-мекенжайы, порттар (көбіне RTSP-порт) және қажет болса қосымша Stream URL өрістерін қамти алады. Бұл режим бұрын орнатылған бейнебақылау жүйелерімен икемді түрде интеграцияны жүзеге асырады. Дұрыс параметрлер болғанда желілік маршруттың қолжетімді болатынын, камераны аралық шлюздің көмегінсіз тікелей қосуға мүмкіндік болады.

Жергілікті камера құрылғының/браузердің камерасын (мысалы, ноутбук веб-камерасы) іске қосуға арналған нұсқа. Формада қолжетімді болатын камералар тізімі қолданылады. Бейне ағын тікелей түрде көрсетіледі. Осы ағын берілген ортада тікелей түрде көріне алады. Камера жайлы жазба жүйеде тіркеле отырады. Таңдалған аудиториямен байланыстырылады. Сыртқы жабдықтың көмегінсіз бейне көзін тез қосуға мүмкіндік талап етілген жағдайда, режим уақытша немесе мобильді түрде бақылау сценарийлері үшін тиімді.

Дереккөз технологиясына тәуелсіз болған камера біртаңбалы орналасуды және дұрыс техникалық атрибуттарды алады. Камера түріне қарамай, әрбір құрылғы нақты локациямен байланыстырылады. Керек техникалық параметрлермен толықтырылады. Бұл мониторингті тиімді басқаруға, оқиғаларды реттік интерпретациялауға, аналитиканы салыстыруға мүмкіндік беруді жүзеге асырады.

3.20-суретте Camera Gateway серверлік модулінің пайдаланушы интерфейсі көрсетілген. Бұл сессияларды басқаруға, модуль құрылғыларды қосуға, камералардан бейнеағынды Nexiface клиенттік бөлігіне жеткізуге жауапты. Экран «шлюздік» сценарийдің толық циклін көрсетеді. Олар құрылғыны PIN-код арқылы тіркеу, ұйымға байланыстыру, бейне ағын сессиясын іске қосу, қолжетімді камераларды анықтау. Осыдан кейін бейнеағынды беру сапасын бақылау үшін клиент тірі бейнені және қызметтік метрикаларды алады (FPS), кезектердің жай-күйі, кадрларды жіберіп алу).



Сурет 3.20. Camera Gateway: шлюз-құрылғыны тіркеу, камераларды анықтау және клиенттік бөлігіне бейне ағындық сессиясын бастау

Бұл сценарийде мультимедияны жеткізу технологиялық тұрғыдан нақты уақыттағы ағын (real-time streaming) моделіне негізделген, мұнда басқару командалары мен сигнализация медиа-арнадан бөлек ұйымдастырылады. Басқару бөлігі (құрылғыны тіркеу, ұйымға байланысты, сессия құру, камераны таңдау, мақсатты FPS орнату) шлюздің HTTO(S) API интерфейсі арқылы жүзеге асырылады. Сессия құрылғаннан кейін бейне деректеріне арналған нақты уақыттағы арна іске қосылады. Негізгі жеткізу механизмі ретінде WebRTC қолданылады, себебі ол: төмен кідірісті қамтамасыз етеді, желі жағдайына автоматты түрде бейімделеді, браузерлерде кросс-платформалық қолдауға ие. WebRTC архитектурасында сигнализация (SDP/ICE параметрлерімен алмасы) қорғалған HTTP(S) сұраныстары арқылы Gateway арқылы өтеді, ал бейне ағын SRTP (шифрланған RTP) протоколы арқылы беріледі. Бұл камера деректермен жұмыс істеу кезінде қауіпсіздік талаптарына сәйкес келеді.

Қолданбалы деңгейде процесс келесідей жүзеге асады: камера немесе шлюз-құрылғы кадрларды түсіреді, оларды аппараттық жеделдетуі бар кодектер арқылы кодтайды, медианы WebRTC сессияны жібереді, браузер клиенті бұл ағынды қабылдап, декодтайды. Желі жағдайына байланысты тұрақты қосылуды қамтамасыз ету үшін ICE/STUN/TURN механизмдері пайдаланылады. Шлюз сапа телеметриясын және журнал жүргізеді. FPS мәні, қызметтік есептегіштер, кадр жоғалту немесе кідіріс көрсеткіштері жатады. Бұл көрсеткіштер жүйедегі тар жерлерді (мысалы, кезек жүктемесі, кадр жоғалуы немесе кешігуі) анықтауға мүмкіндік береді. Бұл дизайн жауапкершілікті бөлінеді. Gateway – бейне ағынды жеткізу мен сапаны бақылауды іске асырады. Ал клиенттік бөлік – камераным ұйым локациясымен байланыстыра отырып, бейнені визуализациялайды. Бұл жүйені масштабталатын және операциялық басқарылатын алатын етеді.

Бұл шлюздік әдістің архитектуралық тұрғыдан тура ұйымдастырылғанын көрсетеді. Камераны басқару API арқылы іске асады. Ал бейнеағын WebRTC арқылы төмен кідірісті, төмен қорғалған медиаканал ретінде жеткізіледі. Мониторинг және интеллектуалды тапсырмалар үшін тұрақты нақты уақыттағы бейнені жеткізуді қамтамасыз етеді

3.21-суретте Google Cloud Run → Service көрсетілген. Nexiface платформасының серверлік бөлігінде контейнерлік сервистер көрсетілген. Бұл тізім микросервистік архитектураны бейнелейді. Мұнда әрбір компонент жеке endpoint-ке ие. Ол тәуелсіз түрде масштабталады, жүйеде нақты функцияны орындайды. Бұл сервистер алдын ала өңдеу (preprocessing), бейне ағынды жеткізу (streaming), модельдерді оқыту (training), бет тану (recognition), қауіпті жағдайларды анықтау (risk detection) міндеттерін қамтиды. Қызметтердің бір-біріне әсерін азайтады, жүйені сүйемелдеуді жеңілдетін архитектуралық бөліну, ағымдағы жүктемеге сәйкес ресурстарды ыңғайлы бөлуде болады.

You are now incurring charges in your billing account **My Billing Account**, as of February 9, 2026. [Learn more](#)

Cloud Run Services Deploy container Connect repo Write a function

Overview
Services
 Jobs
 Worker pools
 Domain mappings

A service exposes a unique endpoint and automatically scales the underlying infrastructure to handle incoming requests. Deploy a container image, source code or a function to create a service.

Filter Filter services

☐	Name ↑	Deployment type	Req/sec ?	Region	Cost ?	Authentication ?	Ingress ?
☐	abnormal-actions-service	(A) Source	0	us-central1	—	Require authentication	All
☐	ai-processor	(A) Source	0.02	us-central1	—	Public access	All
☐	camera-gateway	(A) Source	0.46	us-central1	—	Public access	All
☐	face-collector	(A) Source	0	us-central1	—	Public access	All
☐	face-recognition-service	(A) Source	0	us-central1	—	Require authentication	All
☐	face-training-worker	(A) Source	0	us-central1	—	Require authentication	All

Сурет 3.21. Nexiface үшін Google Cloud Run-де орналастырылған сервистер тізімі. Бейне ағын шлюзі, тану/аналитика, ML-өңдеу сервистері

Жүйедегі негізгі көлік (транспорт) рөлін camera-gateway сервисі атқарады. Ол сессияларды басқаруға, құрылғыларды тіркеуге, камера күйін жариялауға жауап береді. Сондай-ақ әрбір көзбен тікелей байланыс орнатпай-ақ бейнағын мен метадеректерді клиенттік интерфейске жеткізуге жауап береді. Ai-processor сервисі кіріс кадрларын қолданбалы өңдеуді қамтиды (деректерді дайындау, нормализация, маршрутизация). Бұл клиенттік бөлікті жүктемеден босатады. Кейінгі модульдер үшін тек кіріс форматын қалыптастырады. Face-collector тіркеу сценарийлерін қолдайды. Ол фото/видео материалдары және сапа көрсеткіштерін жинақтайды. Тіркеу деректерінің пакетін қалыптастырады және оларды әрі қарай өңдеуді бастайды.

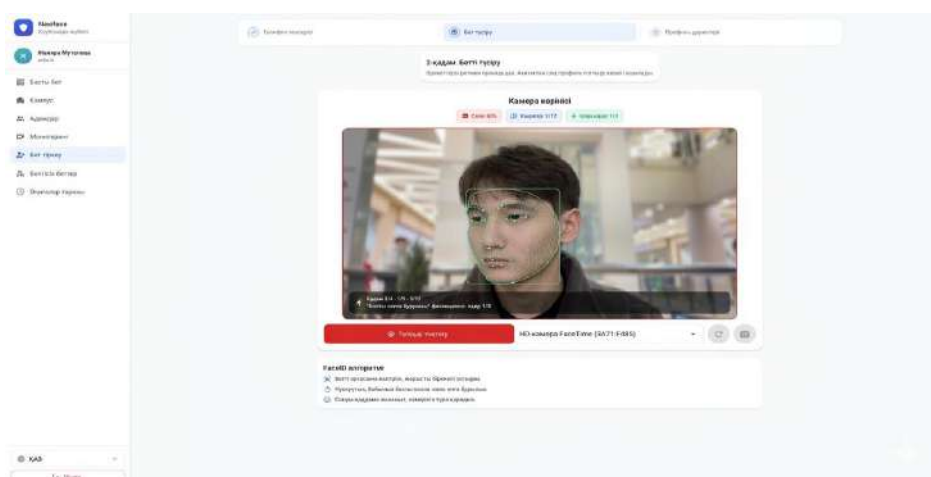
Face-recognition-service тұлғаны сәйкестендіруді жүзеге асырады. Эмбеддингтерді деректер базасымен салыстырады. Белгісіз тұлғаларды өңдейді және сенімділік деңгейімен бірге оқиғаларды қалыптастарыда. Биометриялық деректерді қорғау негізінде бұл сервис міндетті аутентификациямен қорғалған. Face-training-worker асинхронды түрде модельдерді оқыту/қайта оқытуды орындайды. Артефакттарды (индекстер, модель нұсқалары) жаңарту жұмыстарын орындайды. Процестер жүйенің интерактивтілігіне әсер етпейді. Abnormal-actions-service қауіпті әрекеттерді (мысалы, төбелес немесе буллинг) анықтайды. Мониторинг пен хабарландыру үшін тәуекел оқиғаларды қалыптастырады. Аталған сервистер жиынтығы Nexiface жүйесінің толық серверлік контурын құрайды (3.21-сурет).

3.22-суретте бет тіркеу сценарийінің алғашқы кезеңі көрсетілген. Адамның ұйым контекстіндегі негізгі идентификаторы ретінде телефон нөмірін тексеру. Интерфейс қадамдық шебер (wizard) түрінде ұйымдастырылған. Ағымдағы кезеңде пайдаланушы телефон нөмірін енгізіп, тексер бастырмасын жасайды. Осыдан кейін жүйе енгізілген мәнді ұйымның тіркелген тұлғалар базасындағы жазбалармен салыстырады.

Егер нөмір табылмайтын болса, интерфейс профильдің жоқ екені туралы хабарлама көрсетеді. Сондай-ақ биометриялық тіркеу аяқталғаннан кейін жаңа профиль құрылатыныны жайлы хабарлама жасайды. Ал егер нөмір табылатын болса, жүйе процесті жаңарту режиміне ауыстырады. Пайдаланушыға жеке деректерді жаңарту ниетін растау ұсынылады. Профильді қайта жазу немесе нақтылау арқылы тіркеуді жалғастыру мүмкіндігі беріледі. Осы уақытта жазбалардың қайталануын болдырмау қамтамасыз етіледі. Бір телефон идентификаторы мен байланысты бір адамның жеке тұлғасына қолдау көрсетеді. Әрбір адам үшін бір ғана телефон идентификаторына байланысатын біртұтас профиль жасалады. 3.22-суретте көрсетілген механизм келесі кезеңдердің (биометриялық деректерді жинау және профильді сақтау) нақты негізін қалыптастыра алады. Ол деректердің сәйкес болуын қамтамасыз етеді. Жеке тұлғалар туралы жазбалардың өмірлік циклін басқаруды жүзеге асыруды жалғастырады.

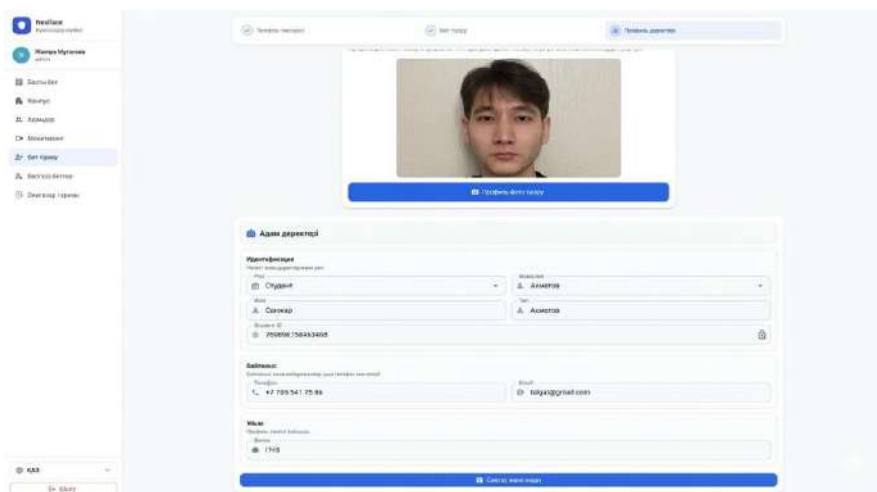
Сурет 3.22. Nexiface: Face Registration — 1-қадам: Телефон нөмірін тексеру және «Жаңа профиль жасау» немесе «Бар профильді жаңарту» опциясын таңдау

Кезеңнің негізгі артефактісі-эмбеддингтер. Бет бейнесі бойынша нейрондық желі экстракторы есептейтін шағын сандық векторлар. Эмбеддинг ішкі белгілер кеңістігіндегі жеке текстуралық-семантикалық ерекшеліктерді кодтайды. Бұл ұқсастық/қашықтық көрсеткіштері арқылы сәйкестендіруді. Әрі қарай орындауға және пайдаланушының тұрақты профилін қалыптастыруға мүмкіндік береді. Өзгергіштікті арттыру және позаға сезімталдықты төмендету үшін жинау реттелетін әрекеттерге сәйкес жүзеге асырылады. Басын солға және оңға бұру, тіке қарау және күлімсіреу. Әр әрекет үшін бірнеше кадрлар жазылады (әдетте 12-ге дейін). Содан кейін кадрлар бірыңғай биометриялық тіркеу пакетіне біріктіреді. Олар эмбеддингтерді есептейді. Тағы да қосымша аугментация әдістері қолданылады отырады. Кадрларға жарықтық пен контрастты өзгерту сияқты фотометриялық түрлендірулер енгізіле отырады. Бұл белгілердің түсіру шарттарына және камера типтеріне төзімділігін арттырады. Адамның бірегейлігін бұзбайды. 3.23-суретте көрсетілген кезеңнің нәтижесі биометриялық профиль құруға және оны кейін сәйкестендіру жүйелерінде пайдалануға жеткілікті болатын стандартталған белгілер жиынтығы болады.



Сурет 3.23. Nexiface: кадрды ұлғайту арқылы ендірулерді жасау. Негізгі нүктені анықтау, поза бойынша қадамдық нұсқаулар және мүмкіндігі бар биометриялық бетті түсіру кезеңі.

3.24-суретте бет тіркеу бизнес-процесінің соңғы кезеңі кескінделген. Биометриялық деректер сәтті жиналғаннан кейін профильдік мәліметтерді қалыптастыру және растау кезеңі. Бұл кезеңде жүйе белгілерді (эмбеддингтерді) алудан қолданбалы деңгейдегі «есептік жазбаны» құруға өтеді. Осы тәсіл тану нәтижелерін нақты бір адаммен дұрыс байланыстыруды жүзеге асырады. Тіркелген профиль білім беру ұйымының контекстінде қолданылады және: мониторинг интерфейсінде, оқиғалар журналында, есеп беру жүйелерінде адамды сәйкестендіруді қамтамасыз етеді.



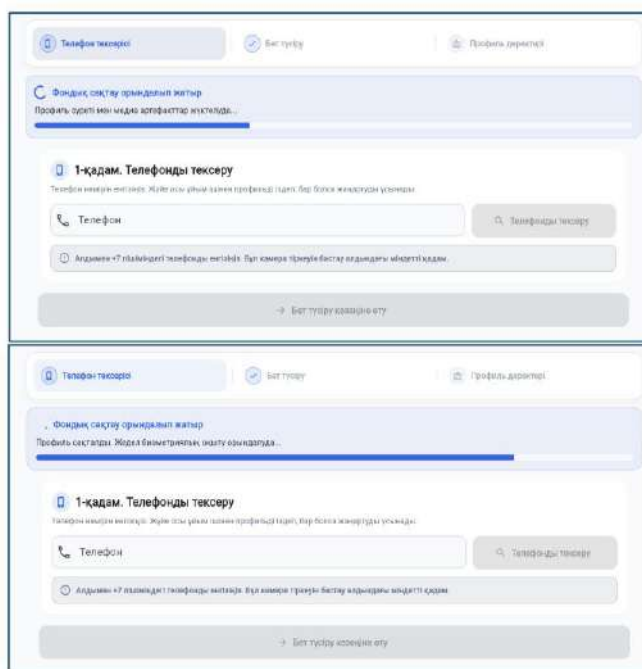
Сурет 3.24. Nexiface: биометриялық тіркеуден кейін профильдік мәліметтерді толтыру (рөлді таңдау, профильді сақтау)

Экранның жоғарғы жағында профильдік фотосурет көрсетіледі. Ол әкімшілік панельде және оқиғаларды қарау кезеңінде визуалды идентификатор қызметін атқарады. Одан кейін «Адам деректерді» блогы орналады. Мұнда субъектінің рөлі таңдалады (мысалы, студент, оқытушы, қызметкер, уақытша қонақ). Рөлді таңдау жазбаны пайдалану семантикасын анықтайды. Қандай өрістер міндетті болатынын (мысалы, студент идентификаторы), қатысу аналитикасында қалай жіктелетінін, қандай бақылау сценарийлері қолданылатын анықтайды.

Негізгі идентификациялық атрибуттар (аты, тегі, ішкі ID), байланыс деректерді (телефон, e-mail); - ұйым құрылымына (бөлім/подразделение) байланыс деректерін қамтиды. Қорытынды әрекет-растау және сақтау — дерекқордағы Профильді жазады, содан кейін кез-келген тану және бақылау оқиғалары осы субъект дұрыс салыстырылуы мүмкін, ал интерфейстегі тізімдер мен карталар оны тіркелген қатысушы ретінде көрсетеді.

3.25-суретте пайдаланушы биометрияны жинауды және профильді толтыруды аяқтаған және жүйе фондық сақтау және есептеу операцияларына көшкен тіркеуден кейінгі күй интерфейсі бейнеленген. Визуализацияның негізгі элементі-фондық сақтаудың орындалуы туралы хабарламасы бар жоғарғы прогресс-бар. Ол әкімшіге профильдік деректер мен медиа артефактілердің жүктеліп, бекітіліп жатқандығы туралы хабарлайды. Сондай-ақ биометриялық профильді дайындаудың есептеу кезеңдері қатар іске қосылады. Платформа пайдаланушының тіркеу сценарийін (ол тез және "қатып қалмай" аяқталуы керек) автономды түрде орындалатын және прогресс күйі ретінде мөлдір түрде көрсетілетін кейінгі есептеу шығындарынан анық ажыратады.

3.25-суретте көрсетілген прогресс-бар қатарынан екі әрекет тобын көрсетеді. Біріншісі - деректерді сақтау: сауалнама атрибуттарын жазу, профильдік суретті және байланысты медиа компоненттерді жүктеу, тіркеу артефактілерін қалыптастыру. Екіншісі-биометриялық өңдеу және оқыту/жаңарту: тіркеу кезінде алынған бет белгілері (негізгі нүктелер/бет геометриясы және эмбеддингтер) тану және сәйкестендіру үшін тұлғаның тұрақты көрінісі қалыптасатын есептеу контурына жіберіледі. Интерфейс деректердің сақталғанын немесе сақталу кезеңінде екенін көрсетеді, содан кейін "жеделдел биометриялық оқу" орындалады, яғни жаңа адам мониторингте ұзақ кідіріссіз дұрыс таныла бастауы үшін модель/индексті жедел дайындау.



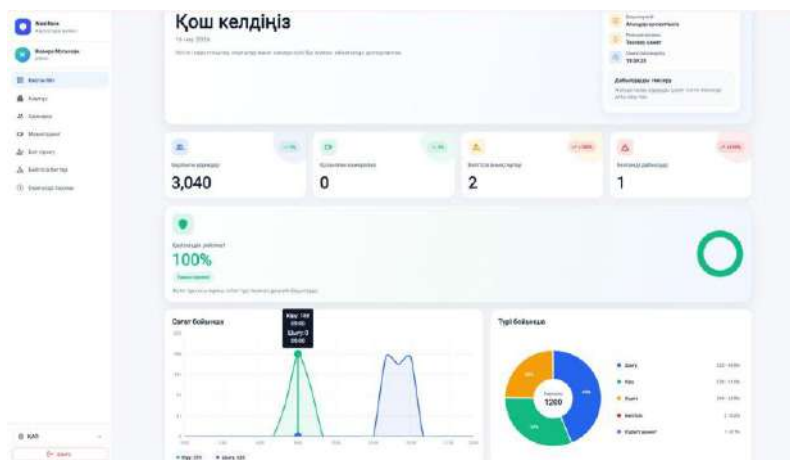
Сурет 3.25. Nexiface: тіркеуден кейінгі прогресс-бар фондық сақтау және жеделдетілген биометриялық өңдеу бар (кескіндерді күшейту және негізгі нүктелер + Фото белгілері бойынша гибридті оқыту)

Аталған процесте аугментацияның рөлі ерекше маңызды екенін ескере кеткен жөн. Аугментация дегеніміз бақыланатын кескін түрлендірулерінің жиынтығы. Бұл адамның бет-әлпетінің сәйкестігін өзгертпестен кеңейтетін оқу жиынтығын. Біздің жүйенің контекстінде модельдің нақты бақылау жағдайларына төзімділігін арттыратын түрлендірулер қолданылады: жарықтандыру мен контрасттың өзгеруі, әртүрлі жарықтандыруды имитациялау, жеңіл геометриялық вариациялар және тұлғаның ақпараттық құрылымын сақтайтын басқа да рұқсат етілген түрлендірулер.

Аугментацияның мақсаты-камераның өзгеруіне, көру бұрышына, көлеңкелерге және біркелкі емес жарықтандыруға тану сезімталдығын азайту, бұл әсіресе аудиториялар мен дәліздер бойынша түсіру жағдайлары өзгертін білім беру мекемелері үшін өте маңызды.

Көрсетілген механизм жүйенің эксплуатациялық тұрғыдан дұрыс архитектурасын көрсетеді. Пайдаланушы тіркеуді қиын есептеулерді күтпей-ақ тез аяқтайды. Ал жүйе биометриялық өңдеуді фондық режимде сақтап, автономды түрде орындайды. Бұл кейін тану дәлдігін және жүйенің тұрақтылығын ұйымдастыру нақты шартында орындалуын арттырады.

Мұндай интерфейс ашықтықты қамтамасыз етеді. Әкімші тек жеке тұлғалар тізімін емес, мониторинг жүйесінде дұрыс сәйкестендіру үшін қажетті биометриялық артефакттардың дайындық күйін де көре алады. 3.26-суреттің жоғарғы бөлігінде тану/сәйкестендіру жаңартуын қолмен бастауға мүмкіндік беретін басқару түймелері орналасқан. Бұл операция индексті жинақталған мәліметтер бойынша (эмбеддингтер, геометрия, тіркеу артефактілері) қайта оқытудың немесе өзектендірудің есептеу циклін іске қосады. Бұл әсіресе адамдарды жаппай тіркеуден кейін, жоғары сапалы жаңа үлгілер пайда болған кезеңде маңызды болады.



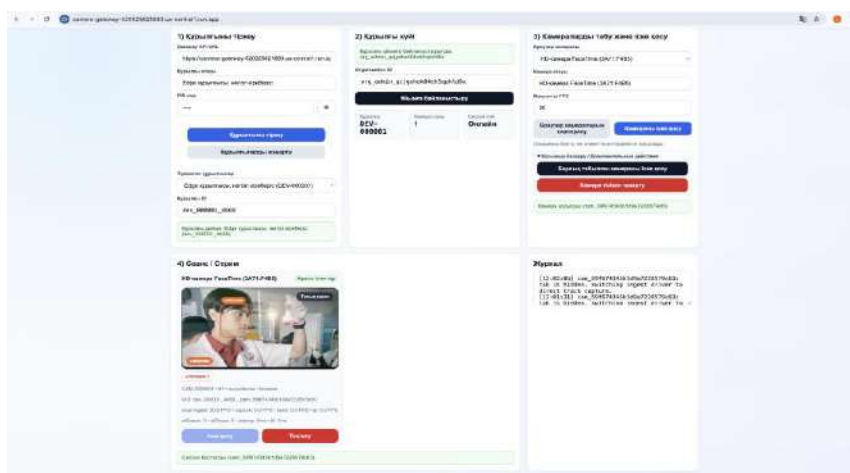
113

Қолмен жаңарту режимінен басқа фондық регламенттік режим қарастырылған. Күндізгі мониторингпен ресурстар үшін бәсекеде болмау үшін оқыту түнгі уақытта автоматты түрде орындалады. Ал жұмыс күнінің басында индекс пен модель өзекті болады. Прогресс тақтасында ағымдағы кезектегі тапсырманы орындау қадамдары (job), өңдеу үлесі және жаңарту процесінің бақылануда болуы мен қайталануын қамтамасыз ететін ықтимал ерекшеліктер көрсетіледі.

Интерфейстің төменгі жағында рөлдердің жиынтық карталары (мысалы, студенттер, оқытушылар, қызметкерлер). Іздеу жолағы мен тіркелген адамдардың тізімі бар. Әрбір жазбада рөл, мәртебе және қызмет белгілері қамтылады. Бұл деректерді тексеру жұмысын және ақпаратты нақтылау барысын немесе сәйкестендіру нәтижелерін тексеру үшін тез адамның профиліне өтуді жеңілдетеді.

Модуль биометриялық жүйенің басқарылатын сүйемелдеуін қамтамасыз етеді. Әкімші сұраныс бойынша модельді жаңартуды іске қоса алады. Сонымен бірге нақты жұмыс кезеңінде тану іс-әрекетінің өзектілігін сақтай отырады. Автоматты түрде түнгі уақытта қайта оқытуға сүйене алады.

3.27-суретте құрылғыны белсендіру, ағындық сеансты бастау және жергілікті сеанс негізіндегі қорытынды — уақытша моделді кэштеу және нәтижелерді клиент жағына жіберу мүмкіндігі интерфейсі бейнеленген. Экран толықтай белсендіру циклін көрсетеді. Құрылғы PIN кодпен расталады. Содан кейін ұйымның контексті анықтала отырады. Кейін пайдаланушы қол жетімді камераны таңдайды. Мақсатты FPS орната алады. Төменгі жағында детекция белгілері бар белсенді болатын бейне ағыны (мысалы, белгісіз адамдар) журналы көрсетіле отырады. Ал оң жағында ауысулардың дұрыстығын және өңдеу күйін бақылауға мүмкіндік беретін сессия оқиғаларының журналы көрсетіледі.



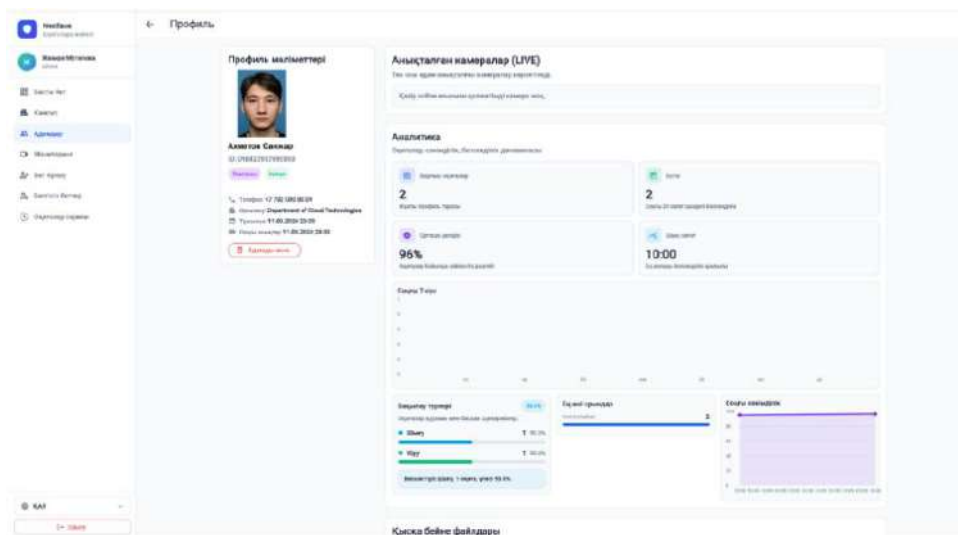
Сурет 3.27. Camera Gateway: құрылғыны белсендіру, ағындық сеансты бастау және жергілікті сеанс негізіндегі қорытынды — уақытша моделді кэштеу және нәтижелерді клиент жағына жіберу мүмкіндігі

Ұсынылған тәсіл шеңберінде есептеу компоненттері (бетті тану, агрессивті әрекеттерді анықтау модельдері) шлюз-сілтеме іске қосылатын құрылғының браузерінің ортасына уақытша жүктеледі. Сессия деңгейіндегі кэште сақталады. Бұл механизм төмен кідірісті қамтамасыз етеді. Модельдер кіріс кадрларға жылдам әсер ету үшін жергілікті қол жетімді. Бұл жүйенің жаңа тұлғалардың пайда болуына және ықтимал қауіпті әрекеттерге реакциясының тиімділігін арттырады. Қауіпсіздікті қамтамасыз ету үшін қорғалған деректерді беру және көлік деңгейінің криптографиялық қорғанысы қолданылады. Артефактілердің өмірлік циклі сеанстың ұзақтығымен шектеледі — қосылу аяқталған кезде кэш тазаланады. Және жүктелген модельдер автоматты түрде жойылады. Бұл сезімтал компоненттерге рұқсатсыз кіру қаупін азайтады.

Онлайн режиміндегі бейне ағыны параллель түрде клиент бөлігіне жіберіледі. Онда детекция/сәйкестендіру нәтижелері іс-шаралар ретінде көрсетіледі, жазылады. Қалыптасқан оқиғалар (камера идентификаторы, уақыт, оқиға түрі, сенімділік және ілеспе метадеректер) ұйымның дерекқорында сақталады. Бұл бақылаулардың кейінгі аналитикалық өңделуін және қайталануын жүзеге асырады.

Онлайн режиміндегі бейне ағыны клиент бөлігіне параллельді жіберіледі. Онда детекция/сәйкестендіру нәтижелері іс-шаралар ретінде көрсетіледі және жазылады. Қалыптасқан оқиғалар (камера идентификаторы, уақыт, оқиға түрі, сенімділік және ілеспе метадеректер) ұйымның дерекқорында сақталады. Бұл бақылаулардың кейінгі аналитикалық өңделуін және қайталануын жүзеге асырады. Архитектура шифрлау және артефактілердің шектеулі өмірлік циклі арқылы қажетті қауіпсіздік деңгейін сақтай отырып, сессия шеңберіндегі төмен патенттік жергілікті байланыстарды (модельдерді уақытша кэштеу) және жүйеде нәтижелерді орталықтандыра отыратын бекітуді (панельдік визуализация және оқиғаларды дерекқорда жадында сақтау) біріктіреді.

3.28-суретте Nexiface әкімшілік тақтасында жүйеге тіркелген адамның профиль экраны бейнеленген. Ол бетті тіркеу бизнес-процесі аяқталғаннан соң автоматты түрде қалыптаса алады (биометриялық жинау → профиль жасау → базада сақтау). Бұл экран "контекст орталығы" рөлін атқарады. Ол тану және бақылау жүйесін кейіннен пайдалану үшін керек сәйкестендіру мәліметтерін, жедел мәртебелер мен аналитикалық көрсеткіштерді біріктіре алады.

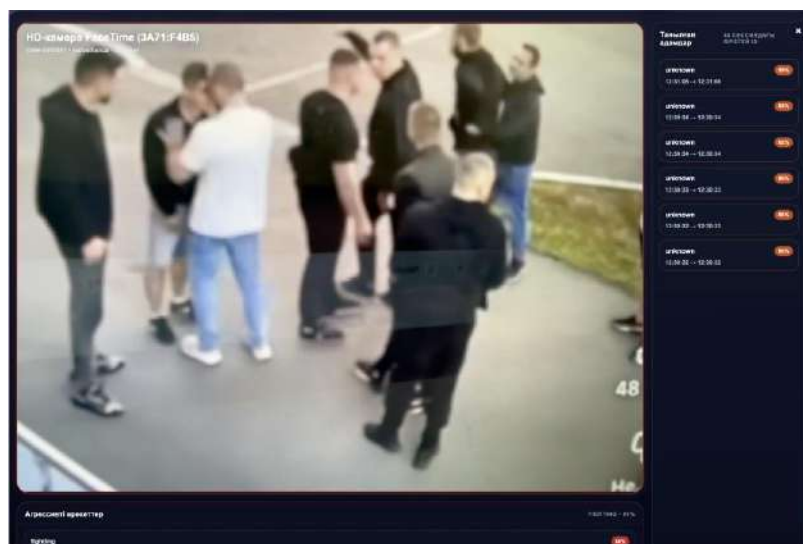


Сурет 3.28. Nexiface: LIVE-камера контекстімен, оқиғаларды талдаумен және жазудың өмірлік циклін басқарумен тіркелген адамның профиль экраны

Сол жақта фотосурет, аты-жөні, бірегей идентификаторы және рөл/күй атрибуттары бар профиль картасы көрсетіледі. Мұнда негізгі метадеректер (байланыс деректері, бөлімше/ұйым, тіркеу уақыты және соңғы жаңарту) жазылады, бұл белгілі бір есептік жазбаға танылатын оқиғалардың бір мәнді байланысын қамтамасыз етеді. Профильді жою түймесі жазудың өмірлік циклін басқаруды көрсетеді және деректерге қол жеткізу және сақтау ережелері аясында жүзеге асырылады.

Интерфейстің оң жағы бақылау нәтижелерін біріктіреді: LIVE камералары блогы нақты уақытта анықталған жеке тұлғаның белсенді көздерін көрсетеді (немесе қолжетімді ағындардың жоқтығын көрсетеді), ал “Аналитика” бөлімі оқиға тарихын, күнделікті белсенділікті, сәйкестік сенімділік ұпайларын және уақытша көрсеткіштерді қорытындылайды. Төменде оқиға түрлері мен трендтерінің көрнекі жиынтықтары, сондай-ақ оқиғаны тексеру мен тексеруді жеңілдететін жылдам медиа клиптер тақтасы орналасқан.

3.29-суретте агрессивті әрекеттерді анықтау, бет-әлпетті тану және модульдері бір уақытта іске қосылған толық экранды бейне ағынын бақылау режимі көрсетілген.



Сурет 3.29. Nexiface: белгісіз беттерді тану және агрессивті әрекеттерді анықтау арқылы толық экранды бақылау (fighting сыныбының мысалы) ықтималдықты бағалаумен экранды мониторинг режимі

Негізгі аймақта бақылау камерасынан алынған бейне ағын көрсетіледі. Ал бейне үстіндегі интерфейс элементтері интеллектуалды өңдеу нәтижелерін визуализациялайды. Сәйкестендірілген және сәйкестендірілмеген (белгісіз) тұлғалар, олардың кадрда пайда болу уақыт аралықтары, сенімділік деңгейінің бағасын визуализациялайды. Мінез-құлықтың қауіпті оқиға ретінде жіктелуі көрсетеді. Бұл көрсету форматы әкімшілік үшін жедел жұмысқа бағытталған. Оператор «шикі» бейнені емес, нақты уақыт режимінде интерпретацияланған қауіпсіздік контекстін алады.

Оң жақ панельде ағымдағы сессияда анықталған адамдар тізімі беріледі. Мұнда деректер базасында жоқ тұлғалар unknown ретінде белгіленеді. Әрбір жазба үшін уақыт белгісі (анықталған кезең аралығы), сенімділік көрсеткіші (пайыздық мөлшердегі мәні) көрсетіледі. Бұл тұрақты анықтауларды шу әсерінен немесе қысқа мерзім аралығында пайда болған жалған детекциялардан ажыратып алуға мүмкіндік береді. Аталған панель бетті тану модулінің нәтижесін көрсете отырады. Жүйе бет белгілерін шығара отырып, оларды эталондық деректермен салыстыра алады. Сәйкестік жоқ болған жағдайда уақыт бойынша оқиғаларды жинақтай отырып, «белгісіз тұлға» ретінде тіркейді.

Төменгі блок агрессивті әрекеттерді талдауға арналған. Мінез-құлық моделі бейне фрагменттерін жіктеп, оларды тәуекел класына (мысалы, fighting) жатқызу ықтималдығын есептейді. Ықтималдықтың көрсетілуі нәтижені бинарлы «бар/жоқ» емес. Сандық сенімділік деңгейі ретінде қарастыруға мүмкіндік береді. Бұл жалған дабылдарды азайтады. Және кейінгі сараптамалық талдау үшін маңызды. Белгісіз тұлғаларды тану мен агрессияны анықтау бірігіп, бірыңғай қауіпсіздік сценарийін қалыптастырады. Күмәнді әрекет байқалған жағдайда жүйе қатысушыларды (белгілі/белгісіз) тіркеп, оқиғаны қауіп түрі бойынша белгілейді.

Үшінші бөлім бойынша қорытынды

Деректерді жіктеу дәлдігін арттыру және өңдеу тиімділігін жоғарылату мақсатында көпдеңгейлі конволюциялық нейрондық желі архитектурасы жасақталды. Ұсынылған Nexiface архитектурасында Retinex алдын ала өңдеу модулі, Faster RCNN детекторы, EfficientNetB0, Self-Attention белгілерді шығару модулі және FaceNet сәйкестендіру моделі біртұтас өңдеу конвейеріне біріктірілді. Retinex алгоритмі жарықтың біркелкі емес әсерін азайтып, кескін сапасын жақсарту арқылы нейрондық желінің тұрақтылығын арттырды. Faster RCNN екісатылы детекция архитектурасы арқылы бет аймақтарын жоғары дәлдікпен анықтауға мүмкіндік берді. EfficientNetB0 жеңіл архитектурасының арқасында есептеу шығындарын азайтып, өңдеу жылдамдығын сақтай отырып, тұлғаның маңызды белгілерін тиімді шығарды. Self-Attention механизмі көз, мұрын және бет контурлары сияқты маңызды аймақтарға фокус жасап, белгілер кеңістігінің ақпараттылығын күшейтті.

FaceNet моделі 128 өлшемді эмбеддинг-векторлар құрып, triplet loss негізінде бір тұлғаға тиесілі векторларды жақындату және әртүрлі тұлғаларды ажырату арқылы жіктеу дәлдігін арттырды. Архитектура нақты уақыт режимінде жұмыс істеуге бейімделіп, әр камерадан секундына 30 кадрға дейін өңдеуді және жалпы кідірісті 100 мс-тан төмен деңгейде сақтауды қамтамасыз етті. Сонымен қатар жүйе бірнеше бейне ағындарын параллель өңдеп, өнімділікті жоғалтпай масштабталатын cloud-native микросервистік архитектура негізінде жүзеге асырылды.

Нәтижесінде ұсынылған CNN архитектурасы деректерді жіктеу дәлдігін, тұрақтылығын және өңдеу тиімділігін жоғарылататын кешенді интеллектуалды жүйе ретінде тиімділігін дәлелдеді.

Жүйе клиенттік интерфейс, Firebase бұлттық инфрақұрылымы және серверлік микросервистер арасында жауапкершілікті тиімді бөлу негізінде құрылған. Firebase қолдану арқылы аутентификация, деректерді сақтау және қолжетімділікті басқару қауіпсіз ірі сенімді түрде ұйымдастырылды. Өзірленген платформа интеллектуалды бейнемониторинг және қолжетімділікті басқару міндеттері үшін толыққанды, қауіпсіз және масштабталатын шешім болады.

ҚОРЫТЫНДЫ

Диссертациялық жұмысты орындау барысында жасанды интеллект және терең оқыту әдістеріне негізделген математикалық модельді әзірлеудің өзекті ғылыми-техникалық мәселесі қарастырылды. Ұсынып отырған модель жарықтандырудың өзгеруі, шу және динамикалық көріністер сияқты визуалды ортаның өзгеру жағдайларында жұмыс істеуге бағытталған және кескінді тану дәлдігі мен жүйенің қалыпты жұмыс жасауына жағдай жасайды.

Кескінді алдын ала өңдеу әдістері, терең нейрондық желілер және көпкезеңді архитектуралар маңыздылығын баса айтқан орынды. Олар кіріс деректерінің ақпараттық толықтығын әжептәуір жоғарлатады, алынатын белгілердің сапасын жақсартады. Сонымен бірге тану жүйелерінің күрделі жағдайларда дұрыс жұмыс жасауына септігін тигізеді.

Зерттеу жұмысы тақырыбының шеңберінде келесі жұмыстар жасалды:

6. нейрондық желілер негізінде бейне ағындардағы кескінді тану саласындағы отандық және шетелдік ғылыми зерттеулер егжей-тегжейлі салыстырмалы талданды;

7. конволюциялық нейрондық желілер мен Retinex алгоритмдері негізінде бейне ағындардағы шу әсері бар кескіндерді алдын ала өңдеуге және бетті тануға арналған математикалық модель жасақталды;

8. бейне ағындардағы кескіндерді тану дәлдігін арттыру және кадрларды өңдеу уақытын қысқарту мақсатында конволюциялық нейрондық желіге негізделген көпкезеңді модель құрастырылды;

9. деректерді жіктеу дәлдігін арттыратын және өңдеу тиімділігін жоғарлатуға бағытталған конволюциялық нейрондық желі архитектурасы тұрғызылды;

10. ұсынылған математикалық модельдер мен алгоритмдер негізінде бейне ағындардағы кескіндерде бетті тану жүйесі әзірленіп, іске асырылды.

Диссертациялық зерттеу барысында келесі негізгі нәтижелер алынды:

1. Кескіндерді алдын ала өңдеуге және бетті тануға арналған конволюциялық нейрондық желілер және Retinex алгоритмдері негізінде математикалық модель құра отырып, зерттелді. Ұсынылған модель аясында біркелкі емес жарықтандыруды түзету, контраст сипаттамаларын қалыпқа келтіру және шу бұрмалануларын басу арқылы кіріс деректерінің сапасын арттыру процестері формалданды. Retinex алгоритмдерін алдын ала өңдеу кезеңінде қолдану белгілер кеңістігінің ақпараттылығын арттырып, әсіресе жарықтандыру жағдайлары өзгермелі күрделі визуалды ортада тұлғаларды тану дәлдігін едәуір жоғарылататыны дәлелденді.

Классикалық математикалық модель бейне кадрын $I_t = S_t + \varepsilon_t$ түрінде сипаттайды, мұнда S_t — пайдалы сигнал, ал ε_t — шу компоненті. Retinex алгоритмін енгізгеннен кейін модель жарықтандыруды түзету функциясымен толықтырылды: $\tilde{I}_t = \varphi(\text{Retinex}(I_t))$

Retinex теориясы бойынша кескін жарықтығы жарықтандыру және шағылысу компоненттерінен тұрады: $I(x, y) = L(x, y) \cdot R(x, y)$. Сондықтан

алгоритмнің негізгі мақсаты жарық әсерін азайтып, пайдалы белгілерді күшейту болды: $R(x, y) = \log I(x, y) - \log L(x, y)$.

Жақсартылған кескін кейін CNN моделіне беріліп, эмбеддинг-векторлар алынды: $z_t = E(\tilde{I}_t; \theta_E)$ Эмбеддингтер косинустық ұқсастық арқылы салыстырылды: $\text{sim}(\hat{z}_a, \hat{z}_b) = \hat{z}_a^T \hat{z}_b$

Нәтижесінде модель құрылымы келесі тізбекке айналды: бейне кадры $\rightarrow$ Retinex арқылы жарықтандыруды түзету $\rightarrow$ нормализация $\rightarrow$ CNN арқылы белгілерді шығару $\rightarrow$ эмбеддингтерді салыстыру $\rightarrow$ сәйкестендіру. Эксперимент нәтижелері бойынша 1 м қашықтықта және 20% жарық кезінде тану дәлдігі 60.98%-дан 86%-ға дейін, ал 3 м қашықтықта 50% жарық кезінде 61.09%-дан 85%-ға дейін артты. Сонымен қатар Accuracy көрсеткіші 6.52%-ға, Precision 9.78%-ға және F1-Score 6.82%-ға жақсарды. Сондықтан Retinex алгоритмі күрделі жарықтандыру жағдайларында тану дәлдігін арттыратын тиімді алдын ала өңдеу әдісі ретінде таңдалды.

2. Бейне ағындардағы кескіндерді тану дәлдігін арттыру және кадрларды өңдеу уақытын оңтайландыру мақсатында CNN негізінде кешенді модель құрылды. Эксперименттік нәтижелер негізінде гибридті модель үшін Faster R-CNN, EfficientNetB0 + Self-Attention және FaceNet архитектуралары таңдалды, себебі олар дәлдік, тұрақтылық және есептеу тиімділігі арасындағы оңтайлы тепе-теңдікті қамтамасыз етті. Faster R-CNN күрделі көріністерде объектілерді жоғары дәлдікпен анықтап, ResNet50 FPN бэкбонымен бірге 100% дәлдік пен ең төменгі 0,0102 loss мәнін көрсетті. Алайда оның толық оқыту уақыты 183 минут болғандықтан, есептеу шығындары жоғары екені анықталды. MobileNetV3 Large ең жылдам архитектура болып, толық оқыту уақыты небәрі 40 минутты құрады, бірақ соңғы эпохаларда loss көрсеткіші 0,1807-ге дейін өсіп, тұрақтылығы төмен екенін көрсетті. EfficientNetB0, Self-Attention моделі үшінші эпохада-ақ 100% дәлдікке жетіп, соңғы эпохада loss мәні 0,0289 жетті. Self-Attention механизмі модельге кескіннің маңызды аймақтарына назар аударуға мүмкіндік берді, күрделі жарықтандыру жағдайларында тану сапасын арттырды. FaceNet архитектурасы тұлғаларды евклидтік қашықтық арқылы сенімді салыстыруға мүмкіндік беріп, бет кескіндерін 128 өлшемді эмбеддингке түрлендірді. Ұсынылған көпкезеңді модель бейне ағындарындағы тұлғаларды нақты уақыт режимінде жоғары дәлдікпен және тұрақты түрде тануды қамтамасыз етті.

3. Деректерді жіктеу дәлдігін арттыруға және өңдеу тиімділігін жоғарылатуға бағытталған конволюциялық нейрондық желі архитектурасы жасақталды.

Деректерді жіктеу дәлдігін арттыру және өңдеу тиімділігін жоғарылату мақсатында көпдеңгейлі конволюциялық нейрондық желі архитектурасы жасақталды. Ұсынылған Nexiface архитектурасында Retinex алдын ала өңдеу модулі, Faster RCNN детекторы, EfficientNetB0, Self-Attention белгілерді шығару модулі және FaceNet сәйкестендіру моделі біртұтас өңдеу конвейеріне біріктірілді. Retinex алгоритмі жарықтың біркелкі емес әсерін азайтып, кескін сапасын жақсарту арқылы нейрондық желінің тұрақтылығын арттырды. Faster RCNN екісатылы детекция архитектурасы арқылы бет аймақтарын жоғары

дәлдікпен анықтауға мүмкіндік берді. EfficientNetB0 жеңіл архитектурасының арқасында есептеу шығындарын азайтып, өңдеу жылдамдығын сақтай отырып, тұлғаның маңызды белгілерін тиімді шығарды. Self-Attention механизмі көз, мұрын және бет контурлары сияқты маңызды аймақтарға фокус жасап, белгілер кеңістігінің ақпараттылығын күшейтті. FaceNet моделі 128 өлшемді эмбеддинг-векторлар құрып, triplet loss негізінде бір тұлғаға тиесілі векторларды жақындату және әртүрлі тұлғаларды ажырату арқылы жіктеу дәлдігін арттырды. Архитектура нақты уақыт режимінде жұмыс істеуге бейімделіп, әр камерадан секундтың 30 кадрға дейін өңдеуді және жалпы кідірісті 100 мс-тан төмен деңгейде сақтауды қамтамасыз етті. Сонымен қатар жүйе бірнеше бейне ағындарын параллель өңдеп, өнімділікті жоғалтпай масштабталатын cloud-native микросервистік архитектура негізінде жүзеге асырылды. Нәтижесінде ұсынылған CNN архитектурасы деректерді жіктеу дәлдігін, тұрақтылығын және өңдеу тиімділігін жоғарылататын кешенді интеллектуалды жүйе ретінде тиімділігін дәлелдеді.

Теориялық маңыздылығы бейне ағындардағы кескіндерді танудың математикалық моделін және нейрондық желілер негізінде кескіндерді өңдеу әдістерін дамытуға бағытталғандығын аңғаруға болады.

Практикалық маңыздылығы ұсынылған алгоритмдерді бейнебақылау және интеллектуалды бейне аналитика жүйелерінде қолдану арқылы кескіндерді нақты уақыт режимінде танудың тиімділігін арттыруға болатындығымен түсіндіруге болады.

Диссертациялық жұмыстың негізгі нәтижелері Қазақстан Республикасы Ғылым және жоғары білім министрлігінің 2024–2026 жылдарға арналған гранттық қаржыландыруы аясында жүзеге асырылып жатырған АР23486538 «Жасанды интеллект негізінде бейне ағындардағы бейнелерді тану жүйесін зерттеу және әзірлеу» ғылыми жобасы аясында орындалып жатырғандығын атап өткен маңызды (Қосымша А).

Зерттеу нәтижелерінің практикалық маңыздылығы мен қолданбалы сипатын растайтын диссертациялық жұмыс шеңберінде «Нейрондық желі негізіндегі тұлғаны тану жүйесі» атты пайдалы модельге №9870 патент (2024 ж.) алынды (Қосымша В).

Диссертациялық жұмыс жарықтандырудың, шудың, динамикалық көріністердің өзгеруі сияқты өзгермелі визуалды орта жағдайында нейрондық желілерге негізделген бейне ағындарындағы кескіндерді танудың математикалық модельдерін әзірлеудің ғылыми және әдіснамалық негіздерін зерттеуге бағытталған негіз қалыптастырды.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Постановление Правительства Республики Казахстан. Об утверждении Концепции развития искусственного интеллекта на 2024 – 2029 годы: утв. 24 июля 2024 года, № 592 // <https://adilet.zan.kz/rus/docs/P2400000592>.
2. A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of Human Faces", Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.
3. Zhao W., Chellappa R., Phillips P.J., Rosenfeld A. Face recognition: A literature survey // ACM computing surveys (CSUR). 2003. vol. 35. no. 4. pp. 399-458.
4. Заварикин Д.Н., Кадейшвили А.А., Коробкова С.В., Соколов А.Ю., Степаненко О.В. Система некооперативной биометрической идентификации людей // Вестник МГТУ им. Н.Э. Баумана. Сер. "Приборостроение", 2012. С. 40-45.
5. Zhou H., Mian A., Wei L., Creighton D., Hossny M., Nahavandi S. Recent advances on singlemodal and multimodal face recognition: a survey // Proceedings of IEEE Transactions on Human-Machine Systems. 2014. vol. 44. no. 6. pp. 701-716.
6. Taigman Y., Yang M., Ranzato M.A., Wolf L. Deepface: Closing the gap to human-level performance in face verification // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. pp. 1701-1708.
7. Tomasi C., Kanade T. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.
8. Детинич Г. ЕС может запретить технологии распознавания лиц в общественных местах. [Электронный ресурс] -URL: <https://3dnews.ru/1001694>.
9. Viola, P.; Jones, M.J. Robust real-time face detection. Int. J. Comput. Vis. 2004, 57, 137–154.
10. Viola, P.; Jones, M.J. Rapid Object Detection using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, USA, 8–14 December 2001; pp. I-511–I-518.
11. Bao, Y.; Tang, Z.; Li, H. Computer vision and deep learning-based data anomaly detection method for structural health monitoring. Struct. Health Monit. 2018, 18, 401–421.
12. Xue, J.; Han, J.; Zheng, T.; Gao, X.; Guo, J. A Multi-Task Learning Framework for Overcoming the Catastrophic Forgetting in Automatic Speech Recognition. arXiv 2019, arXiv:1904.08039.
13. Imran, J.; Raman, B. Deep motion templates and extreme learning machine for sign language recognition. Vis. Comput. 2019.
14. Ravi, S.; Suman, M.; Kishore, P.V.V.; Kumar, K.; Kumar, A. Multi Modal Spatio Temporal Co-Trained CNNs with Single Modal Testing on RGB–D based Sign Language Gesture Recognition. J. Comput. Lang. 2019, 52, 88–102.
15. Al-Emadi, S.; Al-Ali, A.; Mohammad, A.; Al-Ali, A. Audio Based Drone Detection and Identification using Deep Learning. In Proceedings of the 2019 15th

International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 459–464.

16. Prasad, P.S.; Pathak, R.; Gunjan, V.K.; Rao, H.V.R. Deep Learning Based Representation for Face Recognition; Springer: Berlin, Germany, 2019; pp. 419–424.

17. Amirgaliyev, Y.; Shamiluulu, S.; Serek, A. Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods. In Proceedings of the 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 17–19 October 2018; pp. 1–4.

18. Khoroshilov, A.D.A.; Musabaev, R.R.; Kozlovskaya, Y.D.; Nikitin, Y.A.; Khoroshilov, A.A. Automatic Detection and Classification of Information Events in Media Texts. *Autom. Doc. Math. Linguist.* 2020, 54, 202–214.

19. Dyusembayev, A.; Grishko, M. On correctness conditions for the algebra of recognition algorithms with the initial set of μ -operators over the set of problems with binary information. *Rep. Acad. Sci.* 2018, 482, 128–131.

20. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* 2022, 22, 464.

21. Tippins, N.T.; Beaty, J.; Drasgow, F.; Gibson, W.M.; Pearlman, K.; Segall, D.O.; Shepherd, W. Unproctored internet testing in employment settings. *Pers. Psychol.* 2006, 59, 189–225.

22. Boonsuk, W.; Saisin, Y. The Application of a Face Recognition System for the Criminal Database. *J. Appl. Inform. Technol.* 2021, 3, 14–21. [CrossRef] 23. Turk, M.; Pentland, A. Eigenfaces for Recognition. *J. Cogn. Neurosci.* 1991, 3, 71–86.

23. Huang, D.; Shan, C.; Ardabilian, M.; Wang, Y.; Chen, L. Local binary patterns and its application to facial image analysis: A survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 2011, 41, 765–781.

24. Zhu, Q.; He, Z.; Zhang, T.; Cui, W. Improving Classification Performance of Softmax Loss Function Based on Scalable Batch Normalization. *Appl. Sci.* 2020, 10, 2950.

25. Hong, J.H.; Kim, H.; Kim, M.; Nam, G.P.; Cho, J.; Ko, H.S.; Kim, I.J. A 3D model-based approach for fitting masks to faces in the wild. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 235–239. *Computation* 2022, 10, 136

26. Nefian, A.V.; Hayes, M.H. Maximum likelihood training of the embedded HMM for face detection and recognition. In Proceedings of the 2000 International Conference on Image Processing (Cat. No. 00CH37101), Vancouver, BC, Canada, 10–13 September 2000; Volume 1, pp. 33–36.

27. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.

28. Муталова Ж.С., Шаушенова А.Г., Нурпейсова А.А. Алгоритм сверточной нейронной сети, основанный на улучшении функций глубокого обучения. Қазақстан Республикасы Ұлттық инженерлік академиясының

академигі, техника ғылымдарының докторы, профессор Ахметов Бақытжан Сражатдинұлының 70 жылдығына арналған «Жаһандық цифрландыру дәуірінде жасанды интеллекттің білім беру үдерісіне интеграциясы: стратегиялар, инновациялар және киберқауіпсіздік мәселелері» атты Халықаралық ғылыми-практикалық конференция материалдары, Алматы қ. 5 сәуір, 2024 жыл, 1148-1153 б.

29. Mutalova, Z.; Shaushenova, A.; Nurpeisova, A.; et.al. Development of a mathematical model for detecting moving objects in video streams in real-time. IEEE Access. 2024, 10, 136. <https://doi.org/10.1109/ACCESS.2024.3487783>.

30. Nurpeisova, A., Shaushenova A., Kuznetsov O., Ispussinov A., Mutalova Z., Kassymova A. Deep Residual Learning for Face Anti-Spoofing: A Mathematical Framework for Optimized Skip Connections. Technologies, 2025, Volume 13, Issue 9, 413; <https://doi.org/10.3390/technologies13090413>.

31. Муталова Ж.С., Шаушенова А.Г., Исакова Г.О., Нурпейсова А.А., Онгарбаева М.Б., Абдыкаликова Г.А. Метод распознавания человека по изображению лица на основе перемещения точки по направляющим. Известия Национальной Академии Наук Республики Казахстан. Физико-математическая серия, ISSN 1991-346X Том 3. Номер 351 (2024). 118-137 <https://doi.org/10.32014/2024.2518-1726.296>.

32. Муталова Ж. С., Шаушенова А.Г., Нурпейсова А.А., Айнагулова А. С., Онгарбаева М. Б. Нейрондық желілер негізінде окклюзиялы бет-әлпетті тану әдістерін салыстырмалы талдау. Университет еңбектері. – 2026. №1 (102) 374-380 б. https://doi.org/10.52209/1609-1825_2026_1_374.

33. Муталова Ж.С. Шаушенова А.Г. Нурпейсова А.А. Онгарбаева М.Б. Исследование алгоритма распознавания лиц на основе YOLO. Актуальные вопросы современной экономики. №6, 2025. <https://avse-journal.ru/Editions>.

34. Муталова Ж.С. Шаушенова А.Г. Нурпейсова А.А. Онгарбаева М.Б. Сверточная нейронная сеть, основанная на функции распознавания лиц. Актуальные вопросы современной экономики. №6, 2025. <https://avse-journal.ru/Editions>.

35. Муталова З.С., Нурпейсова А.А., Шаушенова А.Г., Испусинов А.М., Жумалиева Л.Д. Нейрондық желі негізінде тұлғаны тану жүйесі, Қазақстан Республикасының патенті. №9870, 2024 (Қазақстан). <https://gosreestr.kazpatent.kz/Utilitymodel/Details?docNumber=409046>

36. Zhou S , Chellappa R. Beyond a single still image : Face recognition from multiple still images and videos/ / Zhao W et al eds. Face Processing : Advanced Modeling and Methods. New York : Academic Press , 2005

37. Shakhnarovich G , Fisher J W , Darrell T. Face recognition from long-term observations/ / Proceedings of the European Conference on Computer Vision. Bari , 2002 : 8512868

38. Liu X M , Chen T , Thornton S M. Eigenspace updating for nonstationary process and its application to face recognition. Pattern Recognition , 2003 , 36 (9) : 194521959

39. Liu X M , Chen T. Video2based face recognition using adap2 tive hidden Markov models/ / Proceedings of t he IEEE Inter2 national Conference on Computer Vision and Pattern Recog2 nition. Madison , 2003 : 3402345
40. Lee K C , Ho J , Yang M H , Kriegman D. Video2based face recognition using probabilistic appearance manifolds/ / Pro2 ceedings of t he International IEEE Conference on Computer Vision and Pattern Recognition. Madison , 2003 : 3132320
41. Lee K C , Ho J , Yang M H , Kriegman D. Visual tracking and recognition using probabilistic appearance manifolds. Computer Vision and Image Understanding , 2005 , 99 (3) : 3032331
42. Zhou S , Krueger V , Chellappa R. Probabilistic recognition of human faces from video. Computer Vision and Image Un2 derstanding , 2003 , 91 (1) : 2142245
43. Zhou S , Chellappa R , Moghaddam B. Visual tracking and recognition using appearance2adaptive models in particle fil2 ters. IEEE Transactions on Image Processing , 2004 , 13 (11) : 143421456
44. Aggarwal G , Chowdhury A K R , Chellappa R. A system identification approach for video2based face recognition/ / Pro2 ceedings of t he IEEE International Conference on Pattern Recognition. Cambridge , 2004 : 23226
45. Arandjelovi'c O , Cipolla R. Face recognition from face motion manifolds using robust kernel resistor2average distance/ / Pro2 ceedings of t he IEEE Conference on Compute Vision and Pat2 ter Recognition workshop. Washington D. C , 2004 : 88293
46. Arandjelovi'c O , Shakhnarovich G , Fisher G , Cipolla R , Darrell T. Face recognition wit h image sets using manifold density divergence/ / Proceedings of t he IEEE Conference on Computer Vision and Pattern Recognition. San Diego , 2005 : 5812588
47. Arandjelovi'c O , Cipolla R. A pose2wise linear illumination manifold model for face recognition using video. Computer Vision and Image Understanding , 2009 , 113 (1) : 1132125
48. Yamaguchi O , Fukui K , Maeda K. Face recognition using temporal image sequence/ / Proceedings of t he IEEE Interna2 tional Conference on Automatic Face and Gesture Recogni2 tion. Nara , 1998 : 3182323
49. Fukui K , Yamaguchi O. Face recognition using multi2view2 point patterns for robot vision/ / Proceedings of t he Interna2 tional Symposium of Robotics Research. Siena , Italy , 2003 : 1922201
50. Nishiyama M , Yamaguchi O , Fukui K. Face Recognition wit h t he multiple constrained mutual subspace met hod/ / Pro2 ceedings of t he 5t h International Conference on Audio2 and Video2Based Biometric Person Aut hentication. New York , 2005 : 71280
51. Li J W , Wang Y H , Tan T N. Video2based face recognition using a metric of average Euclidean distance/ / Proceedings of t he 5t h Chinese Conference on Biometric Recognition. Guan2 gzhou , China , 2004 : 2242232
52. Li J W , Wang Y H , Tan T N. Video2based face recognition using eart h mover's distance/ / Proceedings of t he Interna2 tional Conference on Audio2 and Video2based person Aut hen2 tication. New York , 2005 : 2292239

53. Fan W , Wang Y H , Tan T N. Video2based face recognition using Bayesian inference model/ / Proceedings of t he International Conference on Audio2 and Video2based Person Aut hen2 tication. New York , 2005 : 1222130
54. Jeremiah R. Barr , Kevin W. Bowyer , “Face recognition from video: a review”, International Journal of Pattern Recognition and Artificial Intelligence Vol. 26, No. 5 (2012) 1266002
55. Yongzhong Lu, Jingli Zhou, Shengsheng Yu, ” a survey of face detection, extraction and recognition”, Computing and Informatics, Vol. 22, 2003.
56. L. Juan and O. Gwun, “A Comparison of SIFT , PCA-SIFT and SURF,” International Journal of Image Processing, Vol. 65, pp. 143-152, 2009.
57. R. Muralidharan, C. Chandrasekar , “Combining local and global feature for object recognition using SVM 2004. KNN,” Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering, March 2012.
58. Kelvin Salton Do Prado, Facial Recognition: Understanding LBPH Algorithm, November 11, 2017.
59. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012
60. D. Ciresan, A. Giusti, J. Schmidhuber, et al. Deep neural networks segment neuronal membranes in electron microscopy images. In NIPS, 2012. 1
61. L. N. Clement Farabet, Camille Couprie and Y. LeCun. Learning hierarchical features for scene labeling. PAMI, 35(8), 2013.
62. A Krenker, J Bešter and A Kos,” Introduction to the Artificial Neural Networks”, Edited Kenji Suzuki, Published by InTech,, Janeza Trdine, Croatia, (2011), pp 3-18.
63. N Jindal, V Kumar,” Enhanced Face Recognition Algorithm using PCA with Artificial Neural Networks”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, (2013),pp. 864-872.
64. S Sudhakar Farfade, M Saberian, L-J Li, “Multi-view Face Detection Using Deep Convolutional Neural Networks”, International Conference on Multimedia Retrieval, Shanghai, China, (2015).
65. M.Nandini, P.Bhargavi, G.Raja Sekhar,”Face Recognition Using Neural Network”, International Journal of Scientific and Research Publications, Volume 3, Issue 3, (2013), pp. 1-5
66. H A. Rowley, Student Member,” Neural Network-Based Face Detection”,IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA , (1996), pp. 203 – 208.
67. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems25 (2012): 1097-1105.
68. He, Kaiming, et al. “Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
69. Radosavovic, Ilija, et al. “Designing network design spaces.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

70. Howard, Andrew, et al. "Searching for mobilenetv3." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
71. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International Conference on Machine Learning. PMLR, 2019.
72. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.
73. Li, Yawei, et al. "Spatio-Temporal Gated Transformers for Efficient Video Processing.", NeurIPS ML4AD Workshop, 2021
74. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
75. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
76. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 2016, 39, 1137–1149.
77. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Proceedings, Part I 14, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
78. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
79. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
80. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
81. S. Almabdy and L. Elrefaei, "Deep convolutional neural network-based approaches for face recognition," Applied Sciences (Switzerland), vol. 9, no. 20, Oct. 2019, doi: 10.3390/app9204397.
82. Politeknik Negeri Batam, Institute of Electrical and Electronics Engineers. Indonesia Section. CSS/RAS Joint Chapter, and Institute of Electrical and Electronics Engineers, Proceedings of the 2019 2nd International Conference on Applied Engineering (ICAE) : Batam, Indonesia, Oct 2-3, 2019.
83. Chinese Association of Automation, M. IEEE Systems, and Institute of Electrical and Electronics Engineers, Proceedings, 2019 Chinese Automation Congress (CAC2019) : Nov. 22-24, 2019, Hangzhou, China.

84. East China Jiaotong University et al., Proceedings of the 31st Chinese Control and Decision Conference (2019 CCDC) : 3-5 June, 2019, Nanchang, China.
85. K. Yadav, M. Kumar, S. Kumar, and A. K. Tiwari, "Real Time Face Recognition based on Convolution Neural Network." [Online]. Available: <https://ssrn.com/abstract=3350324>
86. S. S. Rajput and K. V. Arya, "CNN Classifier based Low-resolution Face Recognition Algorithm," in 2020 International Conference on Emerging Frontiers in Electrical and Electronic Technologies, ICEFEET 2020, Institute of Electrical and Electronics Engineers Inc., Jul. 2020. doi: 10.1109/ICEFEET49149.2020.9187001.
87. H. Lee, S. H. Park, J. H. Yoo, S. H. Jung, and J. H. Huh, "Face recognition at a distance for a stand-alone access control system," *Sensors (Switzerland)*, vol. 20, no. 3, Feb. 2020, doi: 10.3390/s20030785.
88. S. P. R. Borra, N. V. S. S. Pradeep, N. T. S. Raju, S. Vineel, and V. Karteek, "Face Recognition based on Convolutional Neural Network," *Int J Eng Adv Technol*, vol. 9, no. 4, pp. 192–196, Apr. 2020, doi: 10.35940/ijeat.D6658.049420.
89. S. Xu, S. Xu, Z. Wang, C. Pu, and Q. Meng, "Face Recognition Algorithm Based on Convolutional Neural Network," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1742-6596/1550/2/022023.
90. L. Ning, Z. Huang, X. Xi, and Y. Zhang, "Research on Face Recognition Based on Improved Convolutional Neural Network Using Raspberry Pi," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Aug. 2021. doi: 10.1088/1742-6596/2002/1/012063.
91. Amjoud, A.B. and Amrouch, M. (2023) Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access*, 11, 35479-35516. <https://doi.org/10.1109/ACCESS.2023.3266093>
92. Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma, A Review of Yolo Algorithm Developments, *Procedia Computer Science*, Volume 199, 2022, Pages 1066-1073, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2022.01.135>. (<https://www.sciencedirect.com/science/article/pii/S1877050922001363>)
93. Zhou Z, Zheng Y, Ye H, Pu J, Sun G. Satellite image scene classification via convnet with context aggregation. In *Pacific Rim Conference on Multimedia*. Springer, Cham. 2018 (pp. 329–339).
94. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* 2004, 57, 137–154.
95. Viola, P.; Jones, M.J. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Kauai, HI, USA, 8–14 December 2001; pp. I-511–I-518.
96. Amirgaliyev, Y.; Shamiluulu, S.; Serek, A. Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods. In *Proceedings of the 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, Almaty, Kazakhstan, 17–19 October 2018; pp. 1–4.

97. Khoroshilov, A.D.A.; Musabaev, R.R.; Kozlovskaya, Y.D.; Nikitin, Y.A.; Khoroshilov, A.A. Automatic Detection and Classification of Information Events in Media Texts. *Autom. Doc. Math. Linguist.* 2020, 54, 202–214.
98. Dyusembayev, A.; Grishko, M. On correctness conditions for the algebra of recognition algorithms with the initial set of operators over the set of problems with binary information. *Rep. Acad. Sci.* 2018, 482, 128–131.
99. Boonsuk, W.; Saisin, Y. The Application of a Face Recognition System for the Criminal Database. *J. Appl. Inform. Technol.* 2021, 3, 14–21.
100. Turk, M.; Pentland, A. Eigenfaces for Recognition. *J. Cogn. Neurosci.* 1991, 3, 71–86.
101. Xiaoli, Y.; Guangda, S.; Jiansheng, C.; Nan, S.; Xiaolong, R. Large Scale Identity Deduplication Using Face Recognition Based on Facial Feature Points. In *Proceedings of the 6th Chinese Conference on Biometric Recognition: CCBR 2011*, Beijing, China, 3–4 December 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 25–32.
102. Hasan, M.K.; Ahsan, M.S.; Newaz, S.H.S.; Lee, G.M. Human Face Detection Techniques: A Comprehensive Review and Future Research Directions. *Electronics* 2021, 10, 2354.
103. T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models," *Proc. Fifth European Conf. Computer Vision*, H. Burkhardt and B. Neumann, eds., vol. 2, pp. 484–498, 1998..
104. Matthew A. Turk and Alex P. Pentland, "Face Recognition Using Eigenfaces", MIT Vision and Modeling Lab, CVPR '91.
105. Cortes C. and Vapnik V., "Support vector networks", *Machine Learning*, 20:1--25, 1995.
106. Guo G., Li S.Z., Chan K. // *IEEE Int. Conf. on Automatic Face and Gesture Recognition*. 2000. P. 196–201.
107. Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *CoRR abs/1502.00873*, 2015.
108. E. Zhou, Z. Cao, and Q. Yin, "Naive-deep face recognition: Touching the limit of LFW benchmark or not?," *CoRR abs/1501.04690*, 2015.
109. Stanko I. The Architectures of Geoffrey Hinton. In: Skansi S. (eds) *Guide to Deep Learning Basics*. Springer, Cham, 2020.
110. Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, H. Chen, Y. Miao, Z. Huang, and J. Liang, "Masked face recognition dataset and application," 2020, *arXiv:2003.09093*.
111. Y. Li, L. Song, X. Wu, R. He, and T. Tan, "Anti-makeup: Learning a bilevel adversarial network for makeup-invariant face verification," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
112. S. Gong, Y. Shi, N. D. Kalka, and A. K. Jain, "Video face recognition: Component-wise feature aggregation network (C-FAN)," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2019, pp. 1–8.
113. Z. Li, U. Park, and A. K. Jain. A discriminative model for age invariant face recognition. *IEEE transactions on Information Forensics and Security (TIFS)*, 2011.

114. D. Gong, Z. Li, D. Lin, J. Liu, and X. Tang. Hidden factor analysis for age invariant face recognition. In International Conference on Computer Vision (ICCV), 2013.
115. P. Jonathon Phillips, W. Todd Scruggs, Patrick J. Flynn et al. FRVT 2006 and ICE 2006 Large-Scale Experimental Results. IEEE Transactions on Pattern Analysis and Machine Intelligence; 2010, p.834-846.
116. K. Wu, J. Huang, Y. Ma, F. Fan и J. Ma, «Cycle-Retinex: non-paired image enhancement in low light using Retinex-Inline CycleGAN», *IEEE Transactions on Multimedia* , I. 26, pp. 1213-1228, 2024, doi:10.1109/TMM.2023.3278385.
117. Zhang Jing yun, Ding Ying, Yang Yi et al., "ICE Workshops - Real-time defrag model based on visible and near-infrared information", 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) , pg. 1, (2016); doi:10.1109/icmew.2016.7574749
118. Kunlun Li, Guoyan Zhang, Xia Li, Jing Xie, Face recognition based on improved Retinex and sparse representation. Procedia Engineering. Volume 15, 2011, Pages 2010-2014, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2011.08.375>.
119. R. Liu, L. Ma, J. Zhang, X. Fan, and Z. Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10561–10570, 2021.
120. Hong Yan, Pang Rong, Wei Qing, Su Jingming, Zhao Feng. Nonlinear Adaptive Enhancement Algorithm for Uneven Illumination Images. Advances in Lasers and Optoelectronics. Volume 60, Issue 16/August 2023. DOI : 10.3788/LOP222380
121. Kilicaslan M, Tanyeri U, Demirci R. Image retrieval using one-dimensional color histogram created with entropy[J]. Advances in Electrical and Computer Engineering, 2020, 20(2): 79-88
122. Bhandari A K, Ghosh A, Kumar I V. A local contrast fusion based 3D Otsu algorithm for multilevel image segmentation[J]. IEEE/CAA Journal of Automatica Sinica, 2020, 7(1): 200-213.
123. Hassan F S, Gutub A. Improving data hiding within colour images using hue component of HSV colour space[J]. CAAI Transactions on Intelligence Technology, 2021, 7(1): 56-68.
124. R. A. Waelen, "The struggle for recognition in the age of facial recognition technology," *AI and Ethics*, vol. 3, no. 1, pp. 215-222, 2023, doi: 10.1007/s43681-022-00146-8.
125. T. V. Dang, "Smart attendance system based on improved facial recognition," *Journal of Robotics and Control (JRC)*, vol. 4, no. 1, pp. 46-53, 2023, doi: 10.18196/jrc.v4i1.16808.
126. V. M. Opanasenko, S. K. Fazilov, S. S. Radjabov, and S. S. Kakharov, "Multilevel face recognition system," *Cybernetics and Systems Analysis*, vol. 60, no. 1, pp. 146-151, 2024, doi: 10.1007/s10559-024-00655-w.

127. D. J. Carragher and P. J. Hancock, "Simulated automated facial recognition systems as decision-aids in forensic face matching tasks," *Journal of Experimental Psychology: General*, vol. 152, no. 5, p. 1286, 2023, doi: 10.1037/xge0001310.
128. P. B. Jha, A. Basnet, B. Pokhrel, G. K. Thakur, and S. Chhetri, "An automated attendance system using facial detection and recognition technology," *Apex Journal of Business and Management*, vol. 1, no. 1, pp. 103-120, 2023.
129. F. Boutros, V. Struc, J. Fierrez, and N. Damer, "Synthetic data for face recognition: Current state and future prospects," *Image and Vision Computing*, vol. 135, p. 104688, 2023, doi: 0.1016/j.imavis.2023.104688.
130. V. M. Opanasenko, S. K. Fazilov, O. N. Mirzaev, and S. Sa'dullo ugli Kakharov, "An ensemble approach to face recognition in access control systems," *Journal of Mobile Multimedia*, pp. 749-768, 2024, doi: 10.13052/jmm1550-4646.20310.
131. H. O. Ikromovich and B. B. Mamatkulovich, "Facial recognition using transfer learning in the deep CNN," *Open Access Repository*, vol. 4, no. 3, pp. 502-507, 2023, doi: 10.17605/OSF.IO/NRMK2.
132. S. Hangaragi, T. Singh, and N. Neelima, "Face detection and recognition using face mesh and deep neural network," *Procedia Computer Science*, vol. 218, pp. 741-749, 2023, doi: 10.1016/j.procs.2023.01.054.
133. D. Mamieva, A. B. Abdusalomov, M. Mukhiddinov, and T. K. Whangbo, "Improved face detection method via learning small faces on hard images based on a deep learning approach," *Sensors*, vol. 23, no. 1, p. 502, 2023, doi: 10.3390/s23010502.
134. S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics recognition using deep learning: A survey," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8647-8695, 2023, doi: 10.1007/s10462-022-10237-x.
135. P. Melzi et al., "Multi-ive: Privacy enhancement of multiple soft-biometrics in face embeddings," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 323-331, 2023.
136. J. C. L. Chai, T. S. Ng, C. Y. Low, J. Park, and A. B. J. Teoh, "Recognizability embedding enhancement for very low-resolution face recognition and quality estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9957-9967, 2023.
137. D. V. Kumar and R. M. Shafi, "Analysis and fast feature selection technique for real-time face detection materials using modified region optimized convolutional neural network," *Materials Today: Proceedings*, vol. 81, pp. 563-569, 2023, doi: 10.1016/j.matpr.2021.04.011.
138. V. P. Lysechko, B. I. Sadovnykov, O. M. Komar, and O. S. Zhuchenko, "A research of the latest approaches to visual image recognition and classification," *Radio Electronics, Computer Science, Control*, no. 1, pp. 140-140, 2024, doi: 10.15588/1607-3274-2024-1-13.
139. G. M. Kumar and A. Pandian, "A constructive deep convolutional network model for analyzing video-to-image sequences," *Data & Knowledge Engineering*, vol. 144, p. 102119, 2023, doi: 10.1016/j.datak.2022.102119.

140. Y. Zhang and X. Hou, "Application of video image processing in sports action recognition based on particle swarm optimization algorithm," *Preventive Medicine*, vol. 173, p. 107592, 2023, doi: 10.1016/j.ypmed.2023.107592.
141. H. P. Nguyen and B. Ribeiro, "Video action recognition collaborative learning with dynamics via PSO-ConvNet Transformer," *Scientific Reports*, vol. 13, no. 1, p. 14624, 2023, doi: 10.1038/s41598-023-39744-9.
142. R. Singh, S. Saurav, T. Kumar, R. Saini, A. Vohra, and S. Singh, "Facial expression recognition in videos using hybrid CNN & ConvLSTM," *International Journal of Information Technology*, vol. 15, no. 4, pp. 1819-1830, 2023, doi: 10.1007/s41870-023-01183-0.
143. Kushwaha, A. Khare, and O. Prakash, "Human activity recognition algorithm in video sequences based on the fusion of multiple features for realistic and multi-view environment," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 22727-22748, 2024, doi: 10.1007/s11042-023-16364-z.
144. F. Chen, Z. Luo, L. Zhou, X. Pan, and Y. Jiang, "Comprehensive survey of model compression and speed up for vision transformers," *arXiv preprint*, arXiv:2404.10407, 2024.
145. B. Yang, X. Wang, Y. Xing, C. Cheng, W. Jiang, and Q. Feng, "Modality Fusion Vision Transformer for Hyperspectral and LiDAR Data Collaborative Classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 2024.
146. L. Roquet, F. F. dos Santos, P. Rech, M. Traiola, O. Sentieys, and A. Kritikakou, "Cross-Layer Reliability Evaluation and Efficient Hardening of Large Vision Transformers Models," 2024.

Анықтама

«С.СЕЙФУЛЛИН атындағы ҚАЗАҚ
АГРОТЕХНИКАЛЫҚ ЗЕРТТЕУ
УНИВЕРСИТЕТІ» КОММЕРЦИЯЛЫҚ
ЕМЕС АКЦИОНЕРЛІК ҚОҒАМЫ



SEIFULLIN
UNIVERSITY

NON-COMMERCIAL
JOINT STOCK COMPANY
«SAKEN SEIFULLIN KAZAKH
AGROTECHNICAL RESEARCH
UNIVERSITY»

010011, Астана қаласы, Жәніс даңғылы, 62-үй
Тел.: (7172) 393-918, факс: (7172) 316-072
e-mail: office@kazatu.edu.kz, www.kazatu.edu.kz

010011, Astana city, 62, Zhenis avenue
tel.: (7172) 393-918, fax: (7172) 316-072
e-mail: office@kazatu.edu.kz, www.kazatu.edu.kz

28.01.2026 № 7.5.7.1-14/292

№ _____ ге

СПРАВКА

Дана докторанту 3 курса по образовательной программе 8D06101 «Аналитика больших данных» Муталовой Жазире Саткановне в том, что ее диссертационная работа на тему «Разработка математической модели распознавания изображений в видеопотоках на основе нейронных сетей» выполнена в рамках бюджетной программы 217 «Развитие науки», подпрограммы 102 «Грантовое финансирование научных исследований» по проекту АР23486538 «Исследование и разработка системы распознавания изображений в видеопотоках на основе искусственного интеллекта» на 2024–2026 годы.

Член Правления - Проректор
по исследовательской
деятельности

Сыргалиев Е.О.

0000852

Оқу процесіне енгізу актісі

«БЕКІТЕМІН»
 Жәңгір хан атындағы БҚАТУ
 Басқарма төрағасы-ректордың
 орынбасары – оқу жұмыстары
 жөніндегі проректоры, а.ш.ғ.д., доцент
 Губашев Н.М.
 « 12 » 2026 ж.

«КЕЛІСІЛДІ»
 Цифрлық экономика және
 тұрақты даму институтының
 директоры
 Ибыжанова А.Д.
 12.01.2026ж

«С. Сейфуллин атындағы Қазақ агротехникалық зерттеу университеті»
 КеАҚ, 8D06101 «Үлкен деректер аналитикасы» білім беру бағдарламасының
 3-курс докторанты Ж.С. Муталованың ғылыми-зерттеу жұмысының
 нәтижелерін оқу үрдісіне енгізу туралы
АКТ-і

1. Ғылыми-зерттеу жұмысының атауы: «Нейрондық желілер негізінде бейне ағындардағы кескінді танудың математикалық моделін әзірлеу»

2. Қысқаша аннотация:

Ғылыми-зерттеу жұмысының жекелеген нәтижелері Жәңгір хан атындағы Батыс Қазақстан аграрлық-техникалық университетінің оқу үдерісінде қолданылуда. Атап айтқанда, 7M06110 – «Ақпараттық жүйелер және технологиялар» білім беру бағдарламасы бойынша «Ақпараттық технологиялар қызметін ұйымдастыру» пәнінің екінші модулі – «Нейрондық желілер мен жасанды интеллекттің арнайы салаларын терең зерттеу» аясында дәріс және практикалық сабақтарды жүргізу барысында пайдаланылуда.

Ғылыми-зерттеу жұмысы аясында әзірленген келесі алгоритмдер мен модельдер оқу процесіне енгізілді:

- бейнеағындарды өңдеудің математикалық моделі;
- кескіндерді алдын ала өңдеу кезеңінде қолданылатын Retinex алгоритмі (жарықтандыруды түзету, шуды азайту);
- бет аймақтарын анықтауға арналған Faster R-CNN моделі;
- ерекшелік белгілерін тиімді шығаратын EfficientNetB0 нейрондық желісі;
- маңызды аймақтарды күшейтетін Self-Attention механизмі;
- тұлғаны сәйкестендіруге арналған FaceNet эмбеддинг моделі.

3. Енгізу тиімділігі:

Оқу процесіне енгізу нәтижесінде:

- білім алушылардың жасанды интеллект және компьютерлік көру саласындағы теориялық білімдері тереңдетілді;
- нақты уақыттағы бейнеағындарды өңдеу бойынша практикалық дағдылары қалыптасты;
- заманауи нейрондық желілерді қолдану арқылы есептерді шешу қабілеті артты;

ҚОСЫМША Ә (жалғасы)

Оқу процесіне енгізу актісі

— ғылыми зерттеу нәтижелерін білім беру процесінде қолдану арқылы оқытудың сапасы жақсарды.

Ғылыми-зерттеу жұмысының нәтижелері қазіргі заманғы жасанды интеллект жүйелерінің талаптарына сәйкес келеді және оқу процесінде тиімді қолданылуда.

4. Енгізу орны мен уақыты: Жәңгір хан атындағы БҚАТУ СТН 270100216151 Орал к., Жәңгір хан көшесі-51, 2025-2026 жж.

5. Енгізу формасы: теориялық, практикалық сабақтар мазмұнына енгізілді

«Дербес деректер және оларды қорғау туралы» Қазақстан Республикасы Занының 8-бабына сәйкес дербес деректерді жинауға және өңдеуге келісім.

Қолдар:

1. Субъект (салық төлеуші, ғылыми-зерттеу жұмысын жүзеге асырған тұлға) немесе оның заңды өкілі:

«С. Сейфуллин атындағы Қазақ агротехникалық зерттеу университеті» КеАҚ, Ақпараттық жүйелер БББ жетекшісі, т.ғ.к., қауымдастырылған профессор

Шаушенова Анаргүл Гимрановна 

2. Ғылыми-зерттеу жұмысының нәтижелері енгізілген субъект немесе оның заңды өкілі:

Жәңгір хан атындағы БҚАТУ Басқарма төрағасы-ректордың орынбасары – оқу жұмыстары жөніндегі проректоры, а.ш.ғ.д., доцент

Губашев Нүркен Маратович 

Өндіріске енгізу актісі

«УТВЕРЖДАЮ»
 Директор ТОО «QUARES»
 _____ Д. Есенулы
 «08» апреля 2026 г.

«СОГЛАСОВАНО»
 Системный администратор
 _____ А. Бершимбаев
 «2» апреля 2026 г.

АКТ ВНЕДРЕНИЯ

результатов научно-исследовательской работы докторанта 3 курса образовательной программы 8D06101 «Аналитика больших данных» НАО «Казахский агротехнический исследовательский университет имени С. Сейфуллина» Муталовой Ж.С. в производственную деятельность

1. Наименование научно-исследовательской работы: «Разработка математической модели распознавания изображений в видеопотоках на основе нейронных сетей».

2. Краткая аннотация: Программный модуль интеллектуального анализа видеопотоков, реализующий математическую модель распознавания изображений на основе конволюционных нейронных сетей, предназначенный для детекции и идентификации лиц в режиме реального времени.

В процессе внедрения реализованы следующие алгоритмы и методы обработки данных:

— на этапе предварительной обработки видеок кадров внедрён алгоритм повышения качества изображений на основе алгоритма Retinex, обеспечивающий компенсацию неравномерного освещения и подавление шумов, что повышает информативность входных данных;

— для детекции лиц в видеопотоке применена модель Faster R-CNN, обеспечивающая высокоточную локализацию областей интереса в условиях шумов, вариативного освещения и различных ракурсов;

— в качестве экстрактора признаков использована модель EfficientNetB0, позволяющая формировать информативные эмбединг-векторы при относительно низких вычислительных затратах;

— дополнительно интегрирован механизм Self-Attention, обеспечивающий адаптивное усиление значимых областей лица (глаза, нос, контуры), что повышает дискриминативную способность признакового пространства;

— на этапе представления признаков и последующего сравнения применена модель FaceNet, преобразующая изображения лиц в компактные эмбединг-векторы, что позволяет эффективно измерять степень сходства между лицами и повышает точность их идентификации.

3. Эффективность внедрения

Комплексное применение указанных алгоритмов и моделей позволило:

— повысить точность распознавания изображений с 92% до 98% (прирост 6%, $\approx 6.52\%$);

Өндіріске енгізу актісі

— обеспечить устойчивость системы к изменениям внешних факторов (освещение, шум, условия съёмки), что подтверждается улучшением метрик: F1-score на 6.82% и Precision на 9.78%;

— обеспечить обработку видеопотока в режиме реального времени со скоростью до 30 кадров/сек (≈ 33 мс на кадр).

Полученные результаты подтверждены экспериментальными исследованиями, проведёнными в различных условиях освещённости, и свидетельствуют об улучшении ключевых метрик качества распознавания.

Предложенная математическая модель распознавания изображений в видеопотоках и её программная реализация соответствуют современным требованиям к системам искусственного интеллекта и успешно интегрированы в технологический стек ТОО «QUARES».

4. Место и время внедрения: ТОО «QUARES» БИН 090240000677 г.Алматы бул.Бухар-Жырау д.75/2 2025-2026 гг.

5. Форма внедрения: Результаты научного исследования внедрены в виде программного модуля интеллектуального анализа видеопотоков, интегрированного в информационную систему организации.

Согласие на сбор и обработку персональных данных в соответствии со статьей 8 Закона Республики Казахстан «О персональных данных и их защите».

Подписи:

1. Субъект или его законный представитель заявителя (налогоплательщик, проводивший научно-исследовательскую деятельность):

Руководитель ГОП «Информационные системы», к.т.н., ассоц. Профессор, НАО «Казахский агротехнический исследовательский университет имени С. Сейфуллина»

Шаушенова Анаргүл Гимрановна

2. Субъект или его законный представитель организации, в производственную деятельность которой были внедрены результаты научно-исследовательской деятельности:

Директор ТОО «QUARES» к.п.н. Есенулы Данияр



Патент

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ  РЕСПУБЛИКА КАЗАХСТАН

REPUBLIC OF KAZAKHSTAN

ПАТЕНТ
PATENT

№ 9870

ПАЙДАЛЫ МОДЕЛЬГЕ / НА ПОЛЕЗНУЮ МОДЕЛЬ / FOR UTILITY MODEL



(21) 2024/1274.2

(22) 09.10.2024

(45) 29.11.2024

(54) Нейрондық желіге негізделген беттерді тану жүйесі
Система распознавания лиц на основе нейронной сети
Face recognition system based on neural network

(73) Нурпейсова Ардак Алданышовна (KZ)
Nurpeisova Ardak Aldanyshovna (KZ)

(72) Муталова Жазира Саткановна (KZ) Mutalova Zhazira Satkanovna (KZ)
Нурпейсова Ардак Алданышовна (KZ) Nurpeisova Ardak Aldanyshovna (KZ)
Шаушенова Анаргүл Гимрановна (KZ) Shaushenova Anargul Gimranovna (KZ)
Испусинов Айдар Манасович (KZ) Ispussinov Aidar Manasovich (KZ)
Жумалиева Ляззат Дауренбаевна (KZ) Zhumaliyeva Lyazzat Daurenbayevna (KZ)



ЭЦҚ қол қойылды
Подписано ЭЦП
Signed with EDS

С. Ахметов
С. Ахметов
S. Akhmetov

«Ұлттық зияткерлік меншік институты» РМК директоры
Директор РГП «Национальный институт интеллектуальной собственности»
Director of RSE «National institute of intellectual property»

Nexiface бетті тану жүйесінің бағдарламалық коды

```

import 'dart:async';
import 'dart:convert';
import 'dart:math' as math;
import 'dart:ui' as ui;
import 'package:camera/camera.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:provider/provider.dart';
import 'package:permission_handler/permission_handler.dart';

import '../110n/localization_provider.dart';
import '../models/person.dart';
import '../services/cloud_run_service.dart';
import '../services/face_detection_service.dart';
import '../services/local_storage_service.dart';
import '../services/mediapipe_face_service.dart';
import '../services/organization_scope.dart';
import '../services/people_service.dart';
import '../services/storage_service.dart';
import '../utils/constants.dart';

enum _EnrollmentGesture {
  lookStraight,
  turnLeft,
  turnRight,
  lookUp,
  lookDown,
  smile,
  done,
}

enum _RegistrationStage {
  phoneCheck,
  faceCapture,
  personDetails,
  uploading,
  success,
}

enum _LiveFaceLinkStatus {
  idle,
  checking,
  unavailable,
  cameraUnavailable,
  noKnownMatch,
  matchedSameProfile,
  matchedDifferentProfile,
  failed,
}

enum _ViewportClass {
  mobile,
  tablet,
  desktop,
}

```

```

enum _BackgroundSaveStatus {
    idle,
    running,
    success,
    error,
}

class _RegistrationBiometricSnapshot {
    final List<_EnrollmentGesture> gestureFlow;
    final Map<_EnrollmentGesture, Uint8List> gestureSamples;
    final Map<_EnrollmentGesture, Map<String, dynamic>> gestureLandmarks;
    final Map<String, dynamic> latestOverlayLandmarks;
    final Uint8List? profileImageBytes;
    final Uint8List? registrationVideoBytes;
    final String registrationVideoExtension;
    final String? registrationVideoError;
    final int qualityScore;
    final List<Uint8List> sequenceFrames;
    final List<Map<String, dynamic>> sequenceFrameDiagnostics;
    final String? capturedImagePath;
    final int serverProcessedFrames;
    final bool serverPreprocessingDone;
    final bool isGestureFlowCompleted;

    const _RegistrationBiometricSnapshot({
        required this.gestureFlow,
        required this.gestureSamples,
        required this.gestureLandmarks,
        required this.latestOverlayLandmarks,
        required this.profileImageBytes,
        required this.registrationVideoBytes,
        required this.registrationVideoExtension,
        required this.registrationVideoError,
        required this.qualityScore,
        required this.sequenceFrames,
        required this.sequenceFrameDiagnostics,
        required this.capturedImagePath,
        required this.serverProcessedFrames,
        required this.serverPreprocessingDone,
        required this.isGestureFlowCompleted,
    });
}

class _RegistrationFormSnapshot {
    final String firstName;
    final String lastName;
    final String email;
    final String phoneInput;
    final String normalizedPhone;
    final String studentId;
    final String department;
    final String selectedRole;
    final Person? existingPersonByPhone;
    final bool overwriteApprovedForExistingPhone;
    final String? lastResolvedPhone;
    final _LiveFaceLinkStatus liveFaceLinkStatus;
}

```



```

final String? lastLiveFaceCheckPhone;
final String? generatedPersonId;
final String? generatedPersonIdPhone;
final _RegistrationBiometricSnapshot biometric;

const _RegistrationFormSnapshot({
  required this.firstName,
  required this.lastName,
  required this.email,
  required this.phoneInput,
  required this.normalizedPhone,
  required this.studentId,
  required this.department,
  required this.selectedRole,
  required this.existingPersonByPhone,
  required this.overrideApprovedForExistingPhone,
  required this.lastResolvedPhone,
  required this.liveFaceLinkStatus,
  required this.lastLiveFaceCheckPhone,
  required this.generatedPersonId,
  required this.generatedPersonIdPhone,
  required this.biometric,
});
}

class FaceRegistrationScreen extends StatefulWidget {
  const FaceRegistrationScreen({super.key});

  @override
  State<FaceRegistrationScreen> createState() => _FaceRegistrationScreenState();
}

class _FaceRegistrationScreenState extends State<FaceRegistrationScreen>
  with WidgetsBindingObserver {
  static String _phoneDraftCache = "";
  static _RegistrationStage _stageDraftCache = _RegistrationStage.phoneCheck;
  static const String _backgroundSaveContextKey =
    'face_registration_background_save_context_v1';

  final _formKey = GlobalKey<FormState>();
  final _firstNameController = TextEditingController();
  final _lastNameController = TextEditingController();
  final _emailController = TextEditingController();
  final _phoneController = TextEditingController();
  final _studentIdController = TextEditingController();
  final _departmentController = TextEditingController();

  final PeopleService _peopleService = PeopleService();
  final FaceDetectionService _faceDetectionService = FaceDetectionService();
  final MediaPipeFaceService _mediaPipeFaceService = MediaPipeFaceService();
  final LocalStorageService _localStorageService = LocalStorageService();
  final StorageService _storageService = StorageService();
  final CloudRunService _cloudRunService = const CloudRunService();

  CameraController? _cameraController;
  List<CameraDescription> _availableCameras = const [];
  int _selectedCameraIndex = 0;

```

```

String _selectedRole = AppConstants.roleStudent;
XFile? _capturedImage;
Uint8List? _capturedImageBytes;

bool _isCameraInitializing = false;
String? _cameraError;
bool _cameraPermissionPermanentlyDenied = false;
Timer? _resizeDebounce;
Timer? _phoneLookupDebounce;
Timer? _autoCaptureTimer;
Timer? _liveOverlayTimer;
Timer? _webBurstCaptureTimer;
Timer? _passiveOverlayTimer;
bool _autoCaptureInProgress = false;
bool _liveOverlayInProgress = false;
bool _passiveOverlayInProgress = false;
bool _isRegistrationSessionActive = false;
bool _isFaceDataReady = false;
bool _isFinalizingFaceData = false;
bool _isRecordingRegistrationVideo = false;
bool _isSavingPerson = false;
bool _isCheckingPhone = false;
bool _isCheckingLiveFace = false;
bool _isProfilePhotoCaptureMode = false;
bool _isCapturingProfilePhoto = false;
bool _overwriteApprovedForExistingPhone = false;
String? _profilePhotoCaptureError;
double? _profilePhotoAspectRatio;
Person? _existingPersonByPhone;
_LiveFaceLinkStatus _liveFaceLinkStatus = _LiveFaceLinkStatus.idle;
String? _lastLiveFaceCheckPhone;
String? _lastResolvedPhone;
bool _serverPreprocessingDone = false;
int _serverProcessedFrames = 0;
bool _isMediaPipeReady = false;
bool _isGestureConditionSatisfied = false;
MediaPipeFaceResult? _latestLiveMediaPipeResult;
DateTime? _lastMediaPipeDiagnosticAt;
bool _isFormattingPhoneInput = false;
int _qualityScore = 0;
String _qualityFeedback = "";
final List<String> _qualityChecks = [];
_EnrollmentGesture _currentGesture = _EnrollmentGesture.lookStraight;
List<_EnrollmentGesture> _activeGestureFlow = const [];
int _currentGestureIndex = 0;
int? _leftTurnDirection;
int? _verticalTurnDirection;
double? _neutralPitch;
final Map<_EnrollmentGesture, Uint8List> _gestureSamples = {};
final Map<_EnrollmentGesture, Map<String, dynamic>> _gestureLandmarks = {};
Map<String, dynamic> _latestOverlayLandmarks = const <String, dynamic>{};
Uint8List? _profileImageBytes;
final List<Uint8List> _currentGestureBurstFrames = <Uint8List>[];
Uint8List? _registrationVideoBytes;
String _registrationVideoExtension = 'mp4';
String? _registrationVideoError;
static const int _framesPerGesture = 3;

```

```

static const int _registrationVideoSeconds = 3;
static const int _persistedSequenceFrameLimit = 8;
static const String _faceGeometrySchema = 'mediapipe-facepack-v1';
static const String _faceSequenceSchema = 'face-sequence-v1';
final List<Uint8List> _sequenceFrames = <Uint8List>[];
final List<Map<String, dynamic>> _sequenceFrameDiagnostics =
    <Map<String, dynamic>>[];
_RegistrationStage _registrationStage = _RegistrationStage.phoneCheck;
double _saveProgress = 0;
String _saveProgressMessage = "";
double _faceFinalizeProgress = 0;
String _faceFinalizeMessage = "";
String _recentlySavedPersonId = "";
bool _showTransitionLoading = false;
String _transitionLoadingTitle = "";
String _transitionLoadingMessage = "";
String? _generatedPersonId;
String? _generatedPersonIdPhone;
Timer? _backgroundSaveNoticeTimer;
_BackgroundSaveStatus _backgroundSaveStatus = _BackgroundSaveStatus.idle;
StreamSubscription<Person?>? _backgroundSavePersonSubscription;
@override
void initState() {
    super.initState();
    WidgetsBinding.instance.addObserver(this);
    if (_phoneDraftCache.isNotEmpty) {
        _phoneController.text = _phoneDraftCache;
    }
    if (_stageDraftCache != _RegistrationStage.uploading &&
        _stageDraftCache != _RegistrationStage.success) {
        _registrationStage = _stageDraftCache;
    }
    _phoneController.addListener(_onPhoneChanged);
    _qualityFeedback = "";
    WidgetsBinding.instance.addPostFrameCallback((_) {
        if (!mounted) return;
        unawaited(_restoreBackgroundSaveContext());
    });
}
Future<void> _initializeMediaPipe() async {
    final ready = await _mediaPipeFaceService.initialize();
    if (!mounted) return;
    await _syncPreferredCameraForMediaPipe();
    if (!mounted) return;
    setState(() {
        _isMediaPipeReady = ready;
    });
    _refreshPassiveOverlayLoop();
    if (!ready || !_isRegistrationSessionActive) return;
    final localization = context.read<LocalizationProvider>();
    _stopAutoCaptureLoop();
    _startLiveOverlayLoop(localization);
    _startWebBurstCaptureLoop(localization);
}
@override
void didChangeMetrics() {
    if (!mounted || kIsWeb || !_isRegistrationSessionActive) return;

```

```

_resizeDebounce?.cancel();
_resizeDebounce = Timer(const Duration(milliseconds: 220), () async {
  final controller = _cameraController;
  if (!mounted || controller == null) return;
  if (!controller.value.isInitialized || controller.value.isTakingPicture) {
    return;
  }
  try {
    await controller.pausePreview();
    await controller.resumePreview();
  } catch (_) {}
});
}
@override
Widget build(BuildContext context) {
  _phoneDraftCache = _phoneController.text;
  _stageDraftCache = _registrationStage;

  final localization = context.watch<LocalizationProvider>();
  final viewportWidth = MediaQuery.sizeOf(context).width;
  final viewportClass = _viewportClassForWidth(viewportWidth);
  final effectiveStage = _registrationStage;
  return Scaffold(
    backgroundColor: AppConstants.backgroundColor,
    body: Stack(
      children: [
        Column(
          children: [
            if (viewportClass != _ViewportClass.desktop)
              _buildHeader(localization),
            Expanded(
              child: LayoutBuilder(
                builder: (context, constraints) {
                  final stageBody = switch (effectiveStage) {
                    _RegistrationStage.phoneCheck =>
                      _buildPhoneCheckStage(localization, constraints),
                    _RegistrationStage.faceCapture =>
                      _buildFaceCaptureStage(localization, constraints),
                    _RegistrationStage.personDetails =>
                      _buildPersonDetailsStage(localization, constraints),
                    _RegistrationStage.uploading =>
                      _buildUploadingStage(localization, constraints),
                    _RegistrationStage.success =>
                      _buildSuccessStage(localization, constraints),
                  };
                  final showProgressHeader =
                    effectiveStage != _RegistrationStage.uploading &&
                    effectiveStage != _RegistrationStage.success;
                  final horizontalPadding = switch (viewportClass) {
                    _ViewportClass.mobile => 12.0,
                    _ViewportClass.tablet => 16.0,
                    _ViewportClass.desktop => 24.0,
                  };
                  final verticalPadding = switch (viewportClass) {
                    _ViewportClass.mobile => 12.0,
                    _ViewportClass.tablet || _ViewportClass.desktop => 16.0,
                  };

```

```

final layoutMaxWidth = _layoutMaxWidthForStage(
  effectiveStage,
  constraints.maxWidth,
  viewportClass,
);
return Padding(
  padding: EdgeInsets.fromLTRB(
    horizontalPadding,
    verticalPadding,
    horizontalPadding,
    verticalPadding,
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
      if (showProgressHeader) ...[
        Align(
          alignment: Alignment.topCenter,
          child: ConstrainedBox(
            constraints:
              BoxConstraints(maxWidth: layoutMaxWidth),
            child: _buildStageProgressHeader(
              localization,
              stage: effectiveStage,
            ),
          ),
        ),
      ],
      const SizedBox(height: 12),
      if (_backgroundSaveStatus !=
        _BackgroundSaveStatus.idle) ...[
        Align(
          alignment: Alignment.topCenter,
          child: ConstrainedBox(
            constraints:
              BoxConstraints(maxWidth: layoutMaxWidth),
            child:
              _buildBackgroundSaveBanner(localization),
          ),
        ),
      ],
      const SizedBox(height: 12),
    ],
  ),
  Expanded(
    child: Align(
      alignment: Alignment.topCenter,
      child: ConstrainedBox(
        constraints:
          BoxConstraints(maxWidth: layoutMaxWidth),
        child: AnimatedSwitcher(
          duration: const Duration(milliseconds: 260),
          switchInCurve: Curves.easeOutCubic,
          switchOutCurve: Curves.easeInCubic,
          transitionBuilder: (child, animation) {
            final slide = Tween<Offset>(
              begin: const Offset(0.02, 0),
              end: Offset.zero,
            ).animate(animation);

```



```

if (bytes == null || bytes.isEmpty) return;
try {
    final codec = await ui.instantiateImageCodec(bytes);
    final frame = await codec.getNextFrame();
    final image = frame.image;
    final width = image.width.toDouble();
    final height = image.height.toDouble();
    image.dispose();
    if (!mounted || width <= 0 || height <= 0) return;
    final ratio = (width / height).clamp(0.56, 2.4).toDouble();
    setState(() {
        _profilePhotoAspectRatio = ratio;
    });
} catch (_) {
    // Keep previous aspect ratio if bytes cannot be decoded.
}

bool _isFaceStepCompleted() {
    return _isBiometricReadyForDetails &&
        _isGestureFlowCompleted &&
        !_isRegistrationSessionActive;
}

static const List<_RegistrationStage> _orderedWizardStages =
    <_RegistrationStage>[
        _RegistrationStage.phoneCheck,
        _RegistrationStage.faceCapture,
        _RegistrationStage.personDetails,
    ];

_ViewportClass _viewportClassForWidth(double width) {
    if (width < 760) return _ViewportClass.mobile;
    if (width < 1200) return _ViewportClass.tablet;
    return _ViewportClass.desktop;
}

double _layoutMaxWidthForStage(
    _RegistrationStage stage,
    double viewportWidth,
    _ViewportClass viewportClass,
) {
    final target = switch (stage) {
        _RegistrationStage.phoneCheck => switch (viewportClass) {
            _ViewportClass.mobile => viewportWidth,
            _ViewportClass.tablet => 920.0,
            _ViewportClass.desktop => 980.0,
        },
        _RegistrationStage.faceCapture => switch (viewportClass) {
            _ViewportClass.mobile => viewportWidth,
            _ViewportClass.tablet => 1120.0,
            _ViewportClass.desktop => 1320.0,
        },
        _RegistrationStage.personDetails => switch (viewportClass) {
            _ViewportClass.mobile => viewportWidth,
            _ViewportClass.tablet => 1140.0,
            _ViewportClass.desktop => 1280.0,
        },
        _RegistrationStage.uploading || _RegistrationStage.success => switch (
            viewportClass) {
            _ViewportClass.mobile => viewportWidth,
            _ViewportClass.tablet => 860.0,
            _ViewportClass.desktop => 920.0,
        },
    };
}

```



```

    return math.min(viewportWidth, target); }
int _stageSequenceIndex(_RegistrationStage stage) =>
    _orderedWizardStages.indexOf(stage);

bool _isWizardStage(_RegistrationStage stage) =>
    _orderedWizardStages.contains(stage);

bool _isStagePrerequisiteCompleted(_RegistrationStage stage) {
    return switch (stage) {
        _RegistrationStage.phoneCheck => true,
        _RegistrationStage.faceCapture => _isPhoneStepCompleted(),
        _RegistrationStage.personDetails => _isFaceStepCompleted(),
        _RegistrationStage.uploading || _RegistrationStage.success => false,    };
    }
bool _isBackwardStageNavigation(_RegistrationStage targetStage) {
    if (!_isWizardStage(targetStage) || !_isWizardStage(_registrationStage)) {
        return false;    }
    return _stageSequenceIndex(targetStage) <
        _stageSequenceIndex(_registrationStage);    }
bool _isStageAvailableFromCurrent(_RegistrationStage targetStage) {
    if (!_isWizardStage(targetStage) || !_isWizardStage(_registrationStage)) {
        return false;    }
    final currentIndex = _stageSequenceIndex(_registrationStage);
    final targetIndex = _stageSequenceIndex(targetStage);
    if (targetIndex == currentIndex) return true;
    if (targetIndex != currentIndex + 1) return false;
    return _isStagePrerequisiteCompleted(targetStage);    }
String _stageLabel(
    LocalizationProvider localization,
    _RegistrationStage stage,
) {
    return switch (stage) {
        _RegistrationStage.phoneCheck => switch (localization.languageCode) {
            AppConstants.languageRussian => 'Проверка телефона',
            AppConstants.languageEnglish => 'Phone verification',
            _ => 'Телефон тексерісі',    },
        _RegistrationStage.faceCapture => switch (localization.languageCode) {
            AppConstants.languageRussian => 'Съёмка лица',
            AppConstants.languageEnglish => 'Face capture',
            _ => 'Бет түсіру',    },
        _RegistrationStage.personDetails => switch (localization.languageCode) {
            AppConstants.languageRussian => 'Данные профиля',
            AppConstants.languageEnglish => 'Profile details',
            _ => 'Профиль деректері',    },
        _RegistrationStage.uploading => switch (localization.languageCode) {
            AppConstants.languageRussian => 'Сохранение',
            AppConstants.languageEnglish => 'Saving',
            _ => 'Сақтау',    },
        _RegistrationStage.success => switch (localization.languageCode) {
            AppConstants.languageRussian => 'Готово',
            AppConstants.languageEnglish => 'Done',
            _ => 'Дайын',    },
    };
}

```