

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Жұмабек Жанат

Бет әлпеті бойынша танып білу жүйесіне арналған программалық қамтама
өндеу

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Алматы 2019


СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедра меңгерушісі
тех. ғыл. кандидаты, доцент,
ассистент-профессор


Р. Юнусов
" 14 " май 2019ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: Бет әлпеті бойынша танып білу жүйесіне арналған программалық
қамтама өңдеу


5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Орындаған

Жұмабек Ж.

Ғылыми жетекші

лектор


Ж.М.Алибиева
" 02 " 05 2019 ж.

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

БЕКІТЕМІН

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,
ассистент-профессор

 Р. Юнусов

" 19 "  2019 ж.

**Дипломдық жоба орындауға
ТАПСЫРМА**

Білім алушы: Жұмабек Жанат

Тақырыбы: Бет әлпеті бойынша танып білу жүйесіне арналған программалық қамтаманы өңдеу

Университет академиялық мәселелер жөніндегі проректорының бұйрығымен «14» наурыз 2018ж. № 1841-б шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі «13» мамыр 2019 ж. Дипломдық жұмыстың бастапқы мәліметтері: Ұсынылатын дипломдық жобаны құру барысында сұлбаусты автоматты түрде құрастырып беретін конструкторлық веб қосымша жасау

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) аналитикалық шолу;

б) жобалау бөлімі;

в) өңдеу құралдары және жобаны талдау;

г) қолданбалы бөлім;

д) А Қосымшасы – техникалық тапсырма.

е) Б Қосымшасы – бағдарлама мәтіні.

Жобаның презентациялық 26 слайды ұсынылған.

Ұсынылған негізгі әдебиеттер: 11 атау.

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Кіріспе	10.01.2019	9000
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	9000
3. Бағдарламалық қамтаманы жобалау	01.02.2019	9000
4. Бағдарламаны әзірлеу	14.02.2019	9000
5. Бағдарламаның мүмкіндіктерін дамыту	18.03.2019	9000
6. Дипломдық жобаға түсіндірме жазба жазу	26.04.2019	9000

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Рамазан А.Б. тьютор	13.05.19	
Бағдарламалық бөлім	Байгаринов А.М. лектор	08.05.19	

Ғылыми жетекші Алибиева Ж.М

Тапсырманы орындауға қабылдап алған студент Жұмабек Ж.

Күні 08 01 2018 ж.

Abstract

Face recognition is one of the key areas in the field of pattern recognition and artificial intelligence. It has been used in a wide range of applications, such as identity authentication, biometrics, and surveillance. This work points out the difficulties to perform such a research and testing process. For detection, the influence of facial attributes such as pose, size or blur is explored. To have more reliable results with regards to the project, I designed a specific dataset on which the same algorithms are tested. Nowadays many algorithms are available for solving this problem in Computer Vision. This work studies the mathematical fundamentals of this technique to understand how information is processed to perform face recognition. Then, some tests have been performed to check the reliability of the application with several databases of facial images. In this way, it is possible to determine the strengths and weaknesses of the algorithm. This work introduces an implementation based on Python using the OpenCV library. The characterization of hardware and the description of software is presented. Next, results, limitations, works, and conclusions over the job development are presented.

Keywords: Face Recognition, face, detection, recognition, system, opencv, eigenface, neural networks, Face detection algorithm, Computer Vision, Image processing, Python.

Аннотация

Распознавание лиц является одним из ключевых направлений в области распознавания образов и искусственного интеллекта. Он был использован в широком диапазоне применений, как удостоверение личности, биометрия, и наблюдение. Эта работа указывает на трудности, связанные с проведением такого исследования и тестирования. Для обнаружения изучается влияние атрибутов лица, таких как поза, размер или размытие. Чтобы получить более сопоставимые результаты в отношении проекта, я разработала конкретный набор данных, на котором тестируются те же алгоритмы. В настоящее время многие алгоритмы доступны для решения этой проблемы в компьютерном зрении. Эта работа изучает математические основы этой техники, чтобы понять, как обрабатывается информация для выполнения распознавания лиц. Затем были проведены некоторые тесты для проверки надежности приложения с несколькими базами данных изображений лица. Таким образом, можно определить сильные и слабые стороны алгоритма. В этой работе представлена реализация на основе Python с использованием библиотеки OpenCV. Представлена характеристика аппаратного обеспечения и описание программного обеспечения. Далее представлены результаты, ограничения, работы и выводы по развитию рабочих мест.

Ключевые слова: распознавание лиц, распознавание лиц, система, opencv, собственное лицо, нейронные сети, алгоритм распознавания лиц, компьютерное зрение, обработка изображений, Python.

Аңдатпа

Бет-әлпетті тану жасанды интеллект саласындағы негізгі бағыттардың бірі болып табылады. Ол жеке куәлік, биометрия, және бақылау ретінде кең ауқымда қолданылған. Бұл жұмыста зерттеу мен тестілеуден өткізуге байланысты туындаған мәселелерді көрсетеді. Бет-әлпетті анықтау үшін бет атрибуттарын, өлшем немесе түсі сияқты аргументтер зерттеледі. Жобаға қатысты салыстырмалы нәтижелер алу үшін мен алгоритмдер мен тестілетін деректердің нақты жиынтығын әзірледім. Қазіргі уақытта көптеген алгоритмдер компьютерлік көзде осы мәселені шешу үшін қол жетімді. Бұл жұмыс тану үшін ақпаратты қалай өңдейтінін түсіну үшін осы техниканың математикалық негіздерін зерттейді. Осылайша, алгоритмнің күшті және әлсіз жақтарын анықтауға болады. Бұл жұмыста OpenCV кітапханасын пайдалану арқылы Python бағдарламалық тілі негізінде жүзеге асыру ұсынылған. Аппараттық қамтамасыз етудің сипаттамасы және бағдарламалық қамтамасыз етудің сипаттамасы берілген. Бұдан әрі жұмыс орындарын дамыту бойынша нәтижелер, шектеулер, жұмыстар мен қорытындылар ұсынылған.

Түйін сөздер: бет тану, бет тану, жүйе, opencv, жеке тұлға, нейрондық желілер, бет тану алгоритмі, компьютерлік көру, бейнелерді өңдеу, Python

МАЗМҰНЫ

Кіріспе	8
1 Бейнелерді тану міндеттерінің жалпы сипаттамасы және олардың түрлері	11
1.1 Бағдарламалық модельдерді оқыту алгоритмдерінің сипаттамасы және түрлері	12
1.2 Бет әлпетті айырып тану әдістер	13
1.3 Бейнелерді танудың қазіргі жағдайы мен перспективалары	14
1.4 Бет әлпетті тану тапсырманың өзектілігі	15
2 Жобалау бөлімі	16
3 Бет әлпетті тану алгоритімі	21
3.1 Бет кеңістігінің базисін құру	22
3.2 Абстракты қойылымы	22
4 Бағдарлама жасау үшін пайдаланылатын құралдар	24
4.1 Бағдарламалық қамтаманың көрсетілімі	26
Қорытынды	31
Пайдаланылған әдебиеттер тізімі	32
А қосымшасы	33
Б қосымшасы	37
Ерекшеліктері	46

КІРІСПЕ

Бұл дипломдық жұмыста адамдардың үш өлшемді бейнелерін пайдалана отырып, адамдарды бейнелерін тану мен сәйкестендірудің әдісі ұсынылады. Тану әдісі тек үш өлшемді бейнелерді ғана пайдаланады және бет - бейнесінің түсі мен текстурасы туралы ақпаратты пайдаланбайды. Мимикалық экспрессиялардың кең спектрін және субъектілердің бас жағдайының өзгеруін қамтитын адамдардың үш өлшемді беттерінің белгілі деректер базасын пайдалана отырып, адамдарды тану нәтижелері келтірілген.

Мен жасаған бейнелерді тану жүйесіне арналған бағдарламалық қамтаманың негізгі мақсаты ұсынылған бейнелердің кейбір ерекшелктерін қолданып қауіпсіздік саласына бағытталған, яғни ент, экзамен тапсырар алдында студенттерге арналған тексеріс құралы болып табылады. Сонымен қатар қол жетімділікті бақылау жүйелерінің сенімділігін және оған әсер ететін факторларды, сондай - ақ белгіленген факторлардың әсерін төмендету мүмкіндігін зерттеу болып табылады, бейнепотоктан адамдарды тану бағдарламасын жазу.

Тапсырманың маңызды тәсілдерінің бірі ортақ функцияларды пайдалануды көздейді. Біз танылатын образдар туралы аз ғана априорлы мәліметтер болған жағдайда тану жүйесін құру кезінде оқыту рәсімін қолданған жөн. Бірінші кезеңде ерікті бөлу функциялары таңдап алынады және одан кейін оқытудың итеративтік қадамдарын орындау процесінде бөлу функциялары қолайлы түрге жеткізіледі.

Адамдарды тану - адам үшін өте қарапайым міндет. Эксперимент барысында 1 - 3 күндік балалар таныс адамдарды ажырата алатынын көрсетті. Бірақ бұл тапсырма компьютер үшін оңай емес. Сыртқы (шаш, бас пішіні) немесе ішкі (көз, мұрын, ауыз) қандай белгілері бар? Суретті қалай талдауға және оны қалай кодтауға болады? Ғалымдардың ойына келген алғашқы тәсіл беттің геометриялық ерекшеліктеріне негізделген. Және адамдарды айырудың осындай алғашқы автоматтандырылған жүйелерінің бірі сипатталған, онда белгіленген нүктелер (көздің, құлақтың, мұрынның жағдайы) белгілер векторын құру үшін пайдаланылған (нүктелер арасындағы қашықтық, олардың арасындағы бұрыш). Танудың өзі Евклидовтың кіріс және тірек бейнелер белгілерінің векторлары арасындағы қашықтықты табу үшін есептеулерді орындады. Бұл әдіс Жарық өзгерістеріне төзімді, бірақ белгіленген нүктелерді дәл анықтай алмайды. Бұл әдістің негізгі тәсілі басты компоненттер (PCA - Principle Component Analysis) әдісін пайдалана отырып, ақпараттық мазмұнды айтарлықтай жоғалтпай бастапқы бейненің ақпаратын қысу болып табылады. Көп өлшемсіздікті болдырмау үшін белгілерді тек бейнеге жақын жерде сипаттай бастады, бұл белгілер кедергілерге, жарыққа және іріктеменің кіші мөлшеріне неғұрлым төзімді болуы тиіс. Мұндай алгоритмдерге мыналар жатады: габаритті вейвлеттер, дискретті косинусты түрлендіру, жергілікті бинарлы паттерналар.

Адамдарды бейнелеін тану негізіндегі процедураларға деген қызығушылық, әсіресе өсіп келе жатқан практикалық қажеттіліктерге байланысты: қауіпсіздік жүйелерін тексеру, сот сараптамасы, телеконференция және т.б. Күнделікті факт адамның бет - әлпетін айқындайтынына қарамастан, бұл процедураны орындау үшін компьютерді адамдардың сандық суреттерін кодтау және сақтау әдістерінің жолын табу керек болды.

1 Бейнелерді тану міндеттерінің жалпы сипаттамасы және олардың түрлері

Бейнелерді тану – белгісіз объект, сигнал, түрлі оқиға, құбылыс немесе процесстерден құрылған кластарды (бейнелерді) тануға немесе анықтауға арналған жүйелерді жасаудың және оны жүзеге асырудың принциптері мен теориясын зерттейтін информатиканың бөлімі. Бұның мәселесі автоматтарды классификациялық есептерді шешуге пайдалану мүмкіндігін зерттеуден шыққан. Бейнелерді тану көбінесе қорғаныс жағдайларында қолданылады. Бейнелерді танудағы объект негізгі сипаттамалардың (белгілердің, қасиеттердің) жиынтығымен сипатталады. Негізгі сипаттамалар әртүрлі ерекшеліктерге ие болуы мүмкін: олар заттай тік реттелген жиынынан немесе дискретті жиыннан болуы мүмкін. Объектіні мұндай түсіну бейнелерді танып - білудің практикалық қосымшаларының қажеттілігімен де, адамның объектіні қабылдау механизімен біздің түсінуімізбен де келісіледі. Компьютерде көптеген қайталанбас бір типті элементтер жиынтығы бар. "Қайталанбас" элементтің бар немесе жоқ екенін білдіреді.

Адамның жеке сипаттамаларын қолдану идеясы жаңалыққа жатпайды. Қазіргі таңда қауіпсіздік жүйелерінде тұлғаны сәйкестендірудің көптеген тәсілдері қолданылатын технологиялар бар. Олар:

- саусақ таңбасы (жеке және толық қол);
- бет келбеті (оптикалық және инфрақызыл суреттер негізінде);
- таңбаларды тану;
- штрих-кодтарды тану;
- автомобиль нөмірлерін тану;
- сөйлеуді тану;
- алаяқтықты анықтау;
- мәтінді тану;
- спамды анықтау;
- құжаттарды санаттау;
- қолжазбаны тану;
- суреттерді тану;
- пайдалы қазбалар кен орындары орналасқан жер қыртысының жергілікті учаскелерін тану;
- дауыс т.б. өзге де сипаттамалар бар.

Барлық биометриялық технологиялардың сәйкестендіру мәселелерін шешуде өз әдістері мен тәсілдері бар. Алайда, барлық әдістер бір - бірінен пайдалану тиімділігі мен нәтиженің дәлдігімен бөлінген. Кез келген биометриялық технология кезең - кезеңмен қолданылады:

- объектіні сканерлеу;

- жеке ақпаратты алу;
- шаблонды құрастыру;
- ағымдағы шаблонды мәліметтер қорымен салыстыру.

Айырып тану биометриялық жүйесі белгілі бір берілген үлгімен, қолданушының нақты физиологиялық немесе мінез - құлықтық сипаттамасы сәйкестігін орнатады .

Биометриялық жүйелерді қолдану кезінде, әсіресе бет - әлпет бойынша айырып тану жүйелері, дұрыс биометриялық сипаттамаларды енгізген жағдайда да сәйкестендіру жайлы шешім әрқашанда оң болып шықпайды. Бұл жағдай, біріншіден, көптеген биометриялық сипаттамалардың өзгеруімен тығыз байланысты. Жүйеде белгілі бір қате ықтималдылығының деңгейі бар. Сонымен бірге әркілі технологияларды қолдану кезінде қате өзгеше болуы мүмкін. Енуді бақылау жүйелері үшін, биометриялық технологияларды қолдану кезінде, алдын ала анықтап алған жөн, не маңызды «бөгдені» өткізбеу әлде «өз адамымыздың» барлығын өткізу. Қауіпсіздік жүйелеріндегі биометриялық технологиялар қолданушылары үшін маңызды фактор қолдану қарапайымдылығы болып табылады. Сканерден өтетін тұлға сипаттамалары ешқандай жайсыздықты сезінбеуі тиіс. Бұл жағдайда, сөзсіз, бет - әлпет бойынша айырып тану технологиясы айрықша қызықты әдіс болып табылады. Адамдардың бейнесін тану қауіпсіздікті қамтамасыз ету талаптары жоғары объектілерде, мысалы, банктерде немесе режимдік кәсіпорындарда пайдаланылады. Модульдің басқа маңызды қолданылуы Казинода, қонақ үйлерде, мейрамханаларда және басқа да осындай объектілерде фейс - бақылауды автоматтандыру болып табылады. Бірқатар компаниялар 40 жылдан астам уақыт бойы автоматтандырылған, қазіргі адам тұлғаларын айырудың автоматты жүйелерін белсенді түрде әзірлеуде: Smith & Wesson (ASID – Automated Suspect Identification System жүйесі); ImageWare (face жүйесі); Imagis, Epic Solutions, Spillman, Miros (Trueface жүйесі); Vissage Technology (Vissage Gallery жүйесі); Visionics (face жүйесі).

1.1 Бағдарламалық модельдерді оқыту алгоритмдерінің сипаттамасы және түрлері

Айқын артықшылықтарға қарамастан, биометриялық мәліметтер тұлғалардың өзіне түсу және жеке өмірлеріне қатысты олардың құқықтарын бұзуы мүмкін бе деген сұрақтар туындататын биометрияға қарсы жағымсыз қате түсініктер тізбегі бар. Көңіл аударарлық өтініштер және қисынсыз даурықпалар арқасында биометриялық технологиялар шынайы іс жағдайынан өзгешелінеді. Соған қарамастан, соңғы жылдары биометриялық теңдестіру әдістерін қолдану ерекше өзектілікке ие болды. Әсіресе берілген мәселеге АҚШ - та 11 - қыркүйекте

болған оқиғалардан соң көңіл бөліне басталды. Әлемдік бірлестіктер бүкіл әлемдегі лаңкестік қауіп - қатерінің арту деңгейін және дәстүрлі әдістер арқылы сенімді қорғанысты ұйымдастыру күрделілігін мойындады. Дәл осы қайғылы оқиғалар, заманауи біріктірілген қауіпсіздік жүйелеріне назарды арттыру үшін себеп болды. Жалпыға мәлім түсінік, егер әуежайлардағы бақылау қаталдау болса, онда көптеген келеңсіз жағдайлар - дың алдын алуға болушы еді. Қазіргі таңда оқиғаларға қатысты кінәлілерді іздестіру, бет - әлпетті анықтап білу жүйелерімен біріктірілген заманауи бейнебақылау жүйелерін қолдану арқылы айтарлықтай жеңілдеуі мүмкін.

1.2 Бет әлпетті айырып тану әдістер

Қазіргі таңда бет - әлпетті айырып танудың негізгі төрт түрі бар:

- eigenfaces;
- айрықша белгілерді саралау;
- нейрондық желі негізінде саралау;
- бет - әлпет кескінін автоматты түрде өңдеу әдісі жоғарыда келтірілген барлық әдістер жүзеге асыру қиындығы және қолдану мақсатымен өзгешелінеді.

«Eigenface» сөзін «өзіндік бет - әлпет» - деп аударуға болады. Бұл технология бет - әлпет кескінінің айрықша сипаттамаларын ұсынатын сұр градациясындағы екіөлшемді кескіндерді қолданады. «Eigenface» әдісі өзге бет - әлпетті айырып тану әдістері үшін негіз ретінде қолданылады.

«Eigenface» 100 - 120 сипаттамаларын құрамдастыра отырып, көп тұлғалар санын қалпына келтіре алады. Тіркеу кезінде әр нақты адамның «eigenface» коэффициенттер тізбегі ретінде ұсынылады. Кескін сәйкестікті тексеру үшін қолданылатын түпнұсқалықты анықтау тәртібінде, өзгелік коэффициентін анықтау мақсаты үшін «нақты» үлгі тіркелінген үлгімен салыстырылынады. Үлгілер арасындағы өзгелік деңгейі теңдестіру коэффициентін анықтайды. Бет әлпетті сканерлеу мүмкіндігі болған жағдайда, «eigenface» технология сын жақсы жарықталған ғимараттарда қолдану оңтайлы. «Айрықша белгілерді» саралау әдістемесі кеңінен қолданылатын сәйкестендіру технологиясы. Бұл технология «eigenface» әдістемесіне ұқсас, алайда жоғарғы дәрежеде тұлғаның ымын немесе сыртқы келбетін (күлімсіреген немесе тұнжыраған бет әлпет) өзгертуге бейімделген. «Айрықша белгілер» технологиясында бет - әлпеттің әрқилы облысындағы ондаған өзіндік ерекшеліктер қолданылады, сонымен қатар салыстырмалы орналасқан жерін есепке ала отыра осы параметрлердің жеке қисындасуы әр нақты тұлғаның өзгешеліктерін анықтайды. Тұлғаның бет - әлпеті қайталанбас болып келеді, бірақ өзгермелі. Себебі тұлға күле алады, көзілдірік кие алады, мұрт пен сақал жібере алады – осының барлығы сәйкестендіру тәртібінің қиыншылдығын арттырады. Сөйтіп, мысалы, күлген мезетте бет - әлпет бөлігінің жылжуы байқалынады. Осындай жылжуларды есепке

ала отырып, тұлға бет - әлпетінің әр түрлі ымдық өзгерістері кезінде сәйкестендіруге болады. Осы саралау бет - әлпеттің жергілікті аумағын қарастыратындықтан, мүмкін ауытқулар жазық бетте 25° дейін болуы ықтимал, және вертикал жазықтықта 15° шамасында және жеткілікті түрде қуатты және қымбат аппаратураны қажет етеді, осыған сәйкес берілген әдістің таралу деңгейін азайтады. Нейрондық желіге негізделген әдісте, екі бет - әлпеттің, тіркелінген және салыстырылатын, өзіндік ерекшеліктері сәйкестікке салыстырылады. «Нейрондық желілер» дерекқорда орналасқан үлгі параметрлерімен және тексерілетін тұлға бет - әлпетінің қайталанбас параметрінің сәйкестігін орнататын алгоритмді қолданады, сонымен қатар мүмкін болатын параметрлер саны қолданылады. Сәйкестендіруге қарай дерекқордағы үлгі мен тексерілетін тұлға арасындағы сәйкессіздік анықталынады, одан кейін лайықты салмақ коэффициенттерінің көмегімен тексерілетін тұлғаның дерекқордағы үлгімен сәйкес келу деңгейін анықтайтын механизм жүзеге асырылады. Бұл әдіс қиын жағдайларда тұлғаның сәйкестендіру сапасын арттырады.

1.3 Бейнелерді танудың қазіргі жағдайы мен перспективалары

«Бет әлпет кескінін автоматты түрде өңдеу» әдісі—көз, мұрын ұшы, ауыз бұрыштары іспеттес оңай анықталынатын бет - әлпет нүктелері арасындағы арақашықтық қатынасын және арақашықты қолданатын айрықша қарапайым технология.

Алайда берілген әдіс «eigenfaces» немесе «нейрондық желі» сияқты соншалықты қуатты емес, ол жеткілікті түрде әлсіз жарықталған жағдайда қолданылуы мүмкін. Дәл осы әдіс көптеген жағдайларда қолданылады, мысалы, ұсынылған құжаттар ұсынған адамға тиістілігін анықтаған жағдайда. Мүмкін болу ортасы: шекарашы халықаралық әуежайда төлқұжаттағы фотоны төлқұжат иесінің бет - әлпетімен салыстыру кезі. Осындай алгоритм бойынша компьютерлі ену жүйесі жұмыс атқарады. Екеуінің арасындағы өзгешелік, компьютерлі ену жүйесінде фотография дерекқорда сақталынған үлгімен салыстырылады.

Сонымен қатар, инфрақызыл жарықта бет әлпетті айырып тану технологиялары пайда бола бастады. Жаңа технология беттің қан тамырларының жылу шығаруымен немесе, басқаша айтқанда, тұлға бетінің термограммасы, құрылған жылулық сурет. Бұл сурет жүйенің әр қолданушысы үшін қайталанбас болып табылады, және енуді бақылау жүйелері үшін биометриялық сипаттамалары ретінде қолданылуы мүмкін. Берілген термограмма, бет геометриясына қарағанда, тұрақты сәйкестендіргіш болып табылады. Себебі ол тұлға келбетінің өзгеруіне байланысты болмайды.

1.4 Бет әлпетті тану тапсырманың өзектілігі

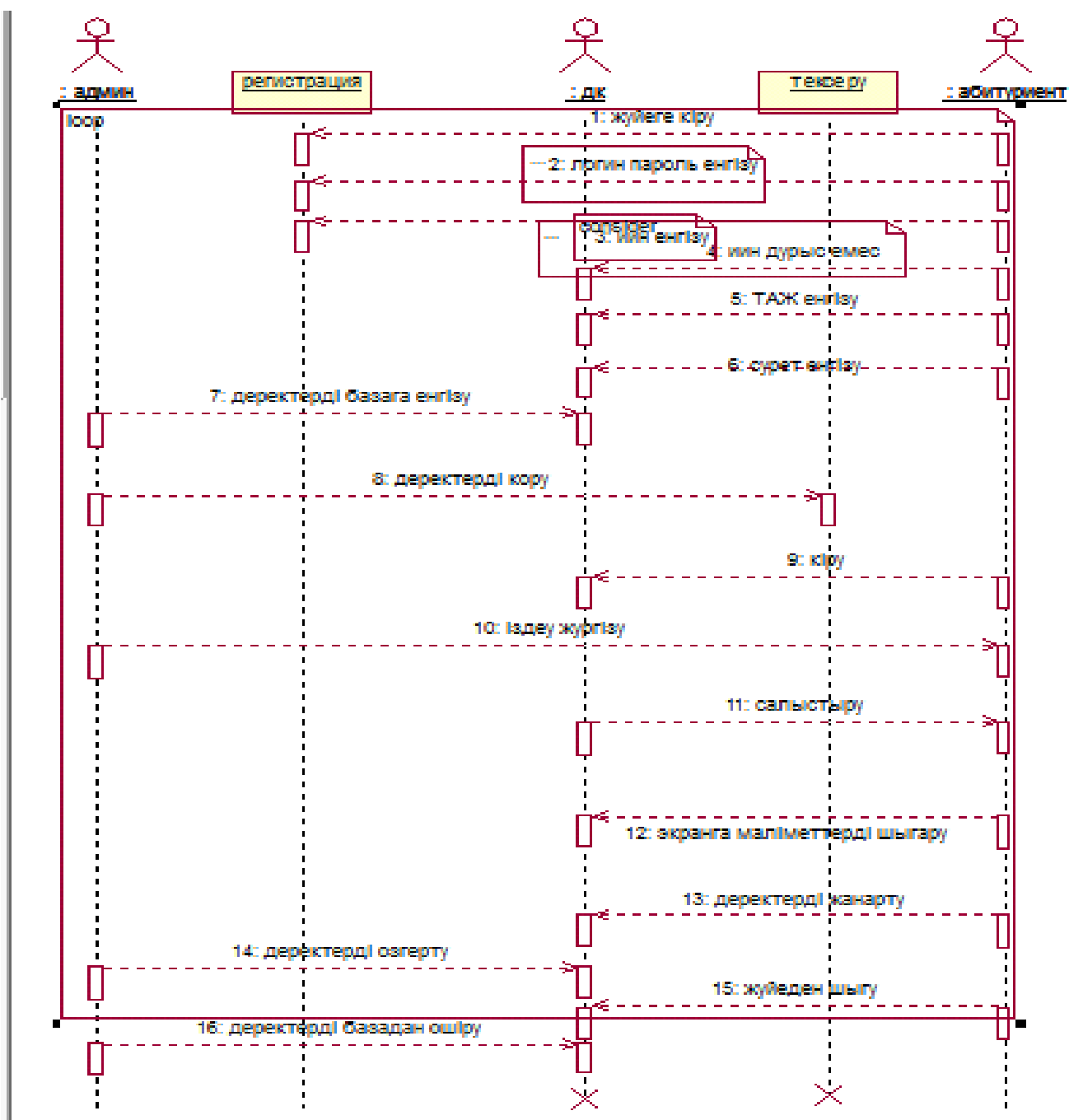
Адамдарды тану міндетін шешу кезінде екі мәселе туындайды. Біріншіден, кез келген сурет пиксель массиві болып табылады. Сонымен қатар, бір пиксель суреттер ештеңе білдірмейді (оның түсі өзгертуге болады және ешкім айырмашылықты байқамайды). Бұл суреттердің мұндай көрінісін анық және экономикалық емес етеді. Осылайша, тұлғаларды тиімді тану үшін суреттерді ұсынудың кейбір ықшам және ыңғайлы пішімін әзірлеу қажет. Бүгінгі күні жоғалтулары бар суреттерді қысудың көптеген тәсілдері белгілі, бірақ онда пайдаланылатын пішім адамдардың суреттерін жіктеу үшін ыңғайлы емес, себебі адамдарды тану мәселесін шешу үшін тағы да ақпарат қажет.

Бұл бірінші кезекте осы адамның коллекциядан қалай көрінетінін анықтаудың қажеті жоқ, ал кері тапсырманы шешу қажет: коллекциядан қай адам осы түрде көрінеді.

Екінші мәселе - бір бейне жарыққа байланысты, эмоцияға байланысты, позицияға байланысты әр түрлі факторларда түсірілуі мүмкін.

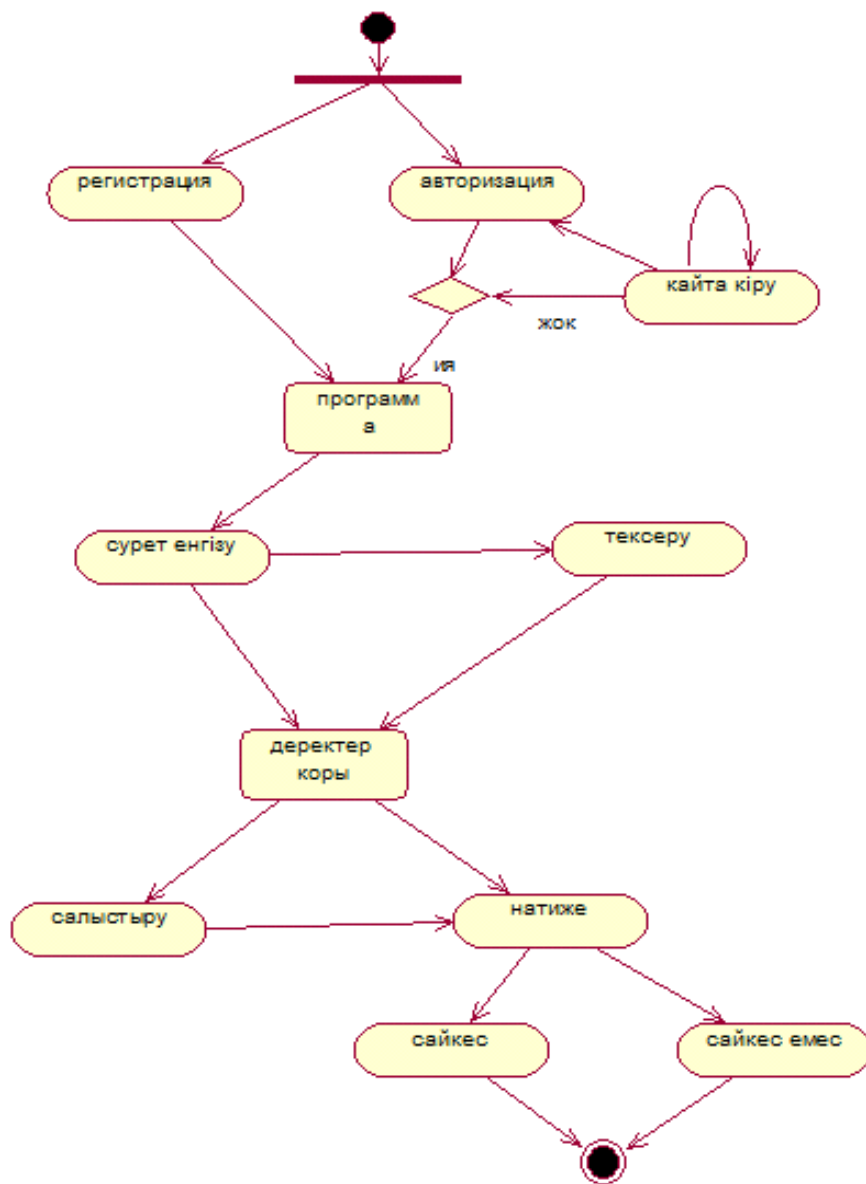
Шын мәнінде, біз адамның нысанды бақылағанда (өлшеуде), ол туралы мәліметтер миға сенсорлардың (талданатын арналардың) соңғы саны бойынша келіп түседі және әрбір сенсорға объектінің тиісті сипаттамасын салыстыруға болады деп ойлаймыз. Біздің объектінің өлшемдеріне сәйкес келетін белгілерден басқа, сондай - ақ бөлінген белгі немесе біз жіктеуші белгілер деп аталатын белгілер тобы бар және X векторындағы олардың мәндерін анықтағанда және табиғи және жасанды тану жүйелерін орындайтын міндеттен тұрады. Бейнені тану процесі кезінде нышандарды бөліп жазу және біріктіру, яғни, мәтіннің жолын графикалық үлгіде жеке нышандарға фрагментациялау – жазба мәтіндерін тану кезінде туындайтын кең ауқымды мәселелердің бірі.

Тізбек диаграммасында объектінің өмірлік циклі (құру - іс - әрекет - белгілі бір мәнді жою) және белгілі бір прецедент шеңберінде АЖ актерлерінің өзара іс - қимылы (сұрауларды жіберу және жауап алу) көрсетілген (2 - сурет).



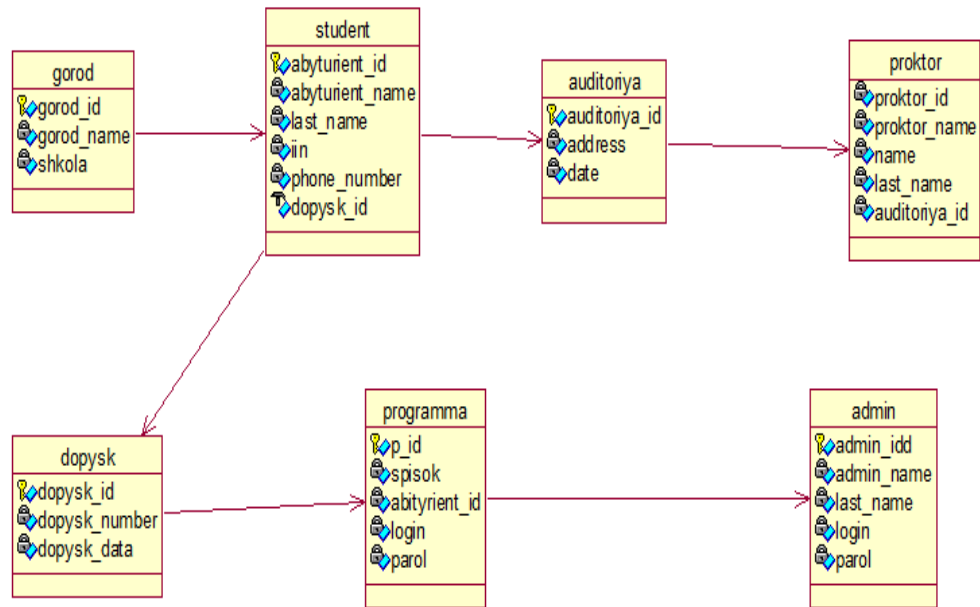
2 - сурет – Тізбек диаграммасы

Күй диаграммасында күрделі жүйені жобалау кезінде оны бөліктерге бөлу қабылданған, олардың әрқайсысын бөлек қарау. Осылайша, объектілік декомпозицияда жүйе хабар алмасу арқылы бір - бірімен өзара әрекеттесетін объектілерге немесе компоненттерге бөлінеді. Хабарлар кейбір оқиғаларды сипаттайды немесе білдіреді (3 - сурет).



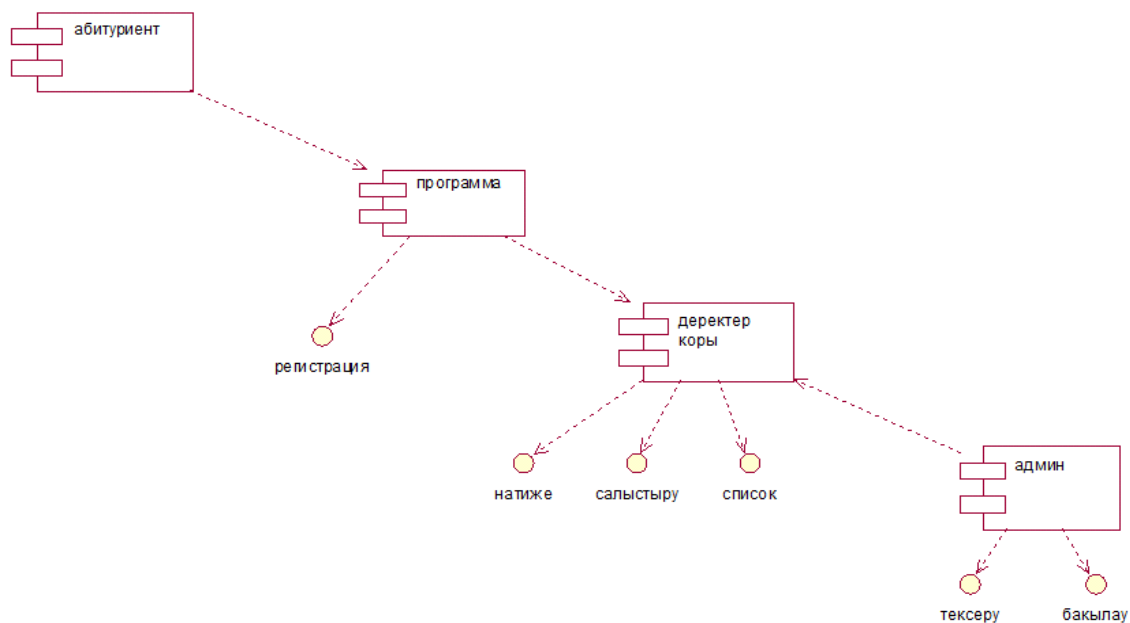
3 - сурет – Күй диаграммасы

ER - модель деректер базасында жоғары деңгейде жобалау кезінде негізгі мәндерді ерекшелеуге және осы мәндердің арасында орнатылуы мүмкін байланыстар белгіленген (4 - сурет).



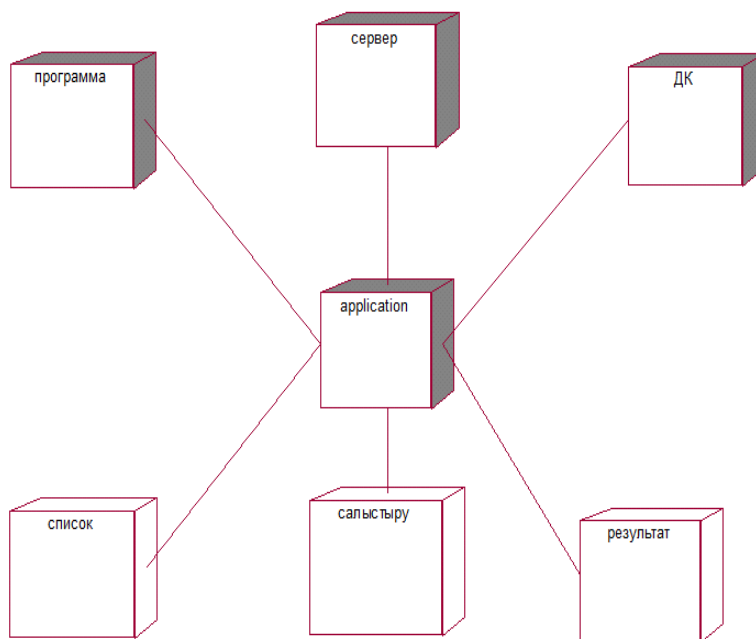
4 - сурет – ER - модель

Компонент диаграммасында жүйе құрауыштарға бөлінген, қолдану қажет және олардың өзара қарым - қатынасын интерфейстер арқылы немесе жүйенің төменгі деңгейлі құрылымындағы құрауыштарының сызбасы көрсетілген (5 - сурет).



5 - сурет – Компонент диаграммасы

Өрістеу диаграммасында ІТ инфроқұрылымының графикалық көрінісі жүйенің компоненттерін желі тораптары бойынша неғұрлым ұтымды бөлуге көмектеседі, оған қоса жүйенің өнімділігі да байланысты (6 - сурет).



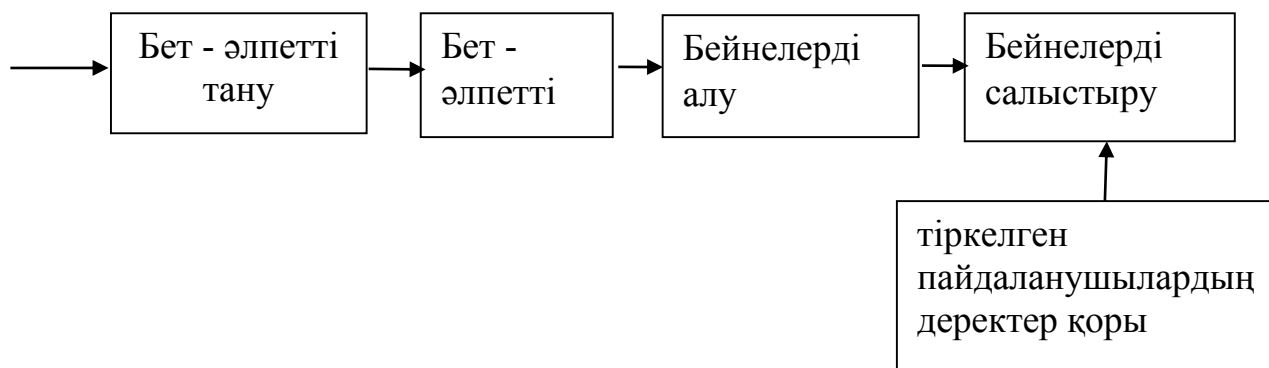
6 - сурет – ІТ инфроқұрылым

3 Бет әлпетті тану алгоритімі

«Бет - әлпетті тану» функциясы бар бейнебақылау жүйесінен алынған кескінді дерекқорда қол жетімділікпен салыстыру қағидаты бойынша жұмыс істейді, ол келесі бір – бірімен байланысқан міндеттерден тұрады:

- а) инициализация;
- б) жоспарлау;
- в) бақылау және мониторинг;
- г) сұраныс талдауы;
- д) дизайн;
- е) кіші жобаларды әзірлеу және енгізу;
- ж) қосалқы жобаларды тексеру;
- з) сілтеме бейнелеу дерекқорынан беттерді қосу және алып тастау;
- и) танылған тұлғалардың фотосуреттерін bmp немесе jpeg файлына басып шығарыңыз және сақтаңыз;
- к) дерекқорда іздеу және көрсету тану статистикасын жүргізу;
- л) бетті тану сәтіне сәйкес келетін бейнені көру;
- м) фотосуреттерді автоматты түрде сәйкестендіру жүйелерінің стандарттарына сәйкестігін тексеріңіз, жобаның маңызды және сыни бағыттары бойынша күш - жігердің маңыздылығын;
- н) барлық жобаны бағалау үшін тұрақты итеративті тестілеу;
- о) қажеттіліктерді, модельдерді және жобаларды іске асыру арасындағы қақтығыстарды анықтау;
- п) біркелкі жүктелген жобаға қатысушылар;

Дипломдық жоба бойынша бағдарламалық қамтаманың алгоритімі 6-суретте көрсетілген.

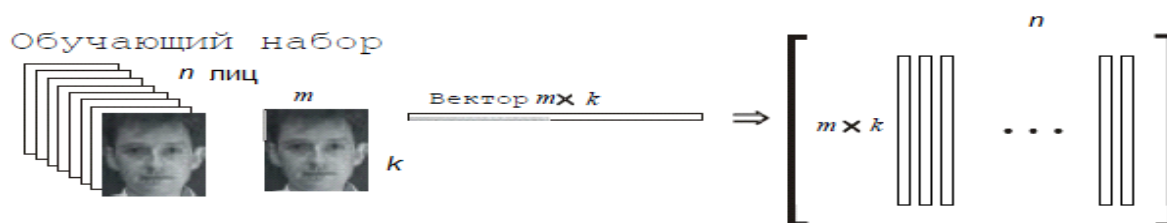


6 - сурет – Бағдарламалық қамтаманың алгоритімі

3.1 Бет кеңістігінің базисін құру

Бізде N портреттер және оларға сәйкес x_i векторы бар. Біз бет кеңістігінің базисін (қырлар кеңістігі, / қырлар кеңістігі / = 100) біртіндеп құрамыз. Базис U_1 бірінші векторын қолда бар адамдарды неғұрлым күшті "ажырататындай" таңдаймыз. $N_i P_{N_i=1} X_i - Y_1 N$. Space Y_1 Space суреттер кеңістігі және әрекетті қайталаймыз. Осылайша біз базис Y_i , $1 = i = 100$ құрамыз (7 - сурет).

Бізге Image Space кеңістігінің z векторын ұсынады. Оны FaceSpace кеңістігінің базисіне қарай ыдыратуды табамыз. Енді біз қолда бар коллекцияның қай тұлғасына осы вектор сәйкес келетінін анықтау қажет. Ол үшін k әдісін пайдалана аламыз.



7 - сурет – Бет кеңістігінің базисі

Бет әлпетті айырып тану технологиясы стандартты бейнекамералармен жұмыс жасайды. Салыстыру үшін – бейне конференция үшін оңтайлы сапа секундны 15 кадрдан бастап бейнеағымның жылдамдығын талап етеді. Бейне - ағымның жоғары жылдамдығы сәйкестендірудің сапасын арттыруға мүмкіндік береді. Алыс арақашықтықта бет - әлпетті айырып тану кезінде, бейнекамера сапасы мен сәйкестендіру нәтижесі арасында тәуелділік пайда болады. Стандартты дербес компьютерлерді қолдану кезінде дерекқордың көлемі 10000 кескіннен аспайды. Қазіргі таңда ұсынылатын бет - әлпетті айырып тану әдістері қызықты және кеңінен қолданысқа енгізуге жақын, алайда осы әдістерге ғана жүгініп отыруымыз дұрыс болмайды. Ол енді бақылау жүйелерінің көмекшісі ретінде таптырмайтын зат.

3.2 Абстракты қойылымы

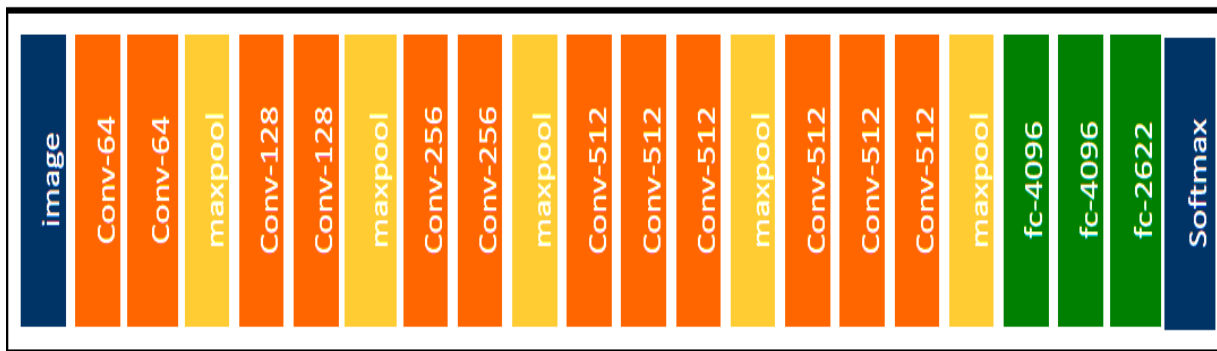
Болашақта барлық фотосуреттер 100×100 пиксель өлшемі 256 сұр реңкі бар деп есептейміз. (8 - сурет) Мысалы, 40 фотосуреттен тұратын кейбір жаттығу коллекциясы бар (әр түрлі жағдайда 4 адамға 10 фотосурет).

Бірінші кезеңде алгоритм жалпы суреттен бейне бөлінеді. Содан кейін суретті қалпына келтіру бар. Суретті қалпына келтіруге келесі әрекеттер жатады:

- Сурет форматын 100×100 дейін өзгерту
- 256 сұр реңкке түстерді түрлендіру
- Суреттің жалпы жарықтығын кейбір орташа мәнге дейін өзгерту

Кейбір алгоритмдер үшін суреттегі бет мүмкіндігінше тік болуы қажет. Бұл жағдайда бетті дұрыс бұрышқа бұру қалыпқа келтіру кезінде жүзеге асырылады.

Тану алгоритмінің келесі кезеңі қолда бар тұлғаның сипаттамасын бөлу болып табылады. Таңдалған сипаттамалар тану алгоритміне қатты байланысты, сондықтан олардың мысалдары кейінірек келтірілетін болады. Сипаттамаларды бөлгеннен кейін сурет қажет емес. Танудың соңғы кезеңі қолда бар сипаттамалар бойынша есепке жауап беретін жіктеуішті қолдану болып табылады.



8 - сурет –VGG - Face моделі

4 Бағдарлама жасау үшін пайдаланылатын құралдар

Python бағдарламалау тілі - Бұл тілді таңдау себебтерім: ол қандай да бір операциялық жүйемен немесе машинамен байланысты емес, бұл тіл осы жұмысты орындау үшін қажетті бағдарламалаудың негізгі парадигмаларын қолдайды (ООП, функционалдық бағдарламалау, процедуралық бағдарламалау);

- оңай оқылатын программалау тілі;
- пайдалы кітапханалары көп.

NumPy кітапханасы - бұл үлкен кітапханамен бірге Python үлкен көп өлшемді массивтер мен матрицалардың қолдауын қосатын жоғары деңгейлі және математикалық функциялардың үлкен кітапханасымен C, C++ және Fortran кірістірмелерін пайдалана отырып, өте жылдам жұмыс істейтін операцияларды орындайды. Мүмкіндіктері:

- көп өлшемді массивтерді қолдау (матрицаларды қоса алғанда));
- көп өлшемді массивтермен жұмыс істеуге арналған жоғары деңгейлі математикалық функцияларды қолдау.

OpenCV кітапханасы - бұл компьютерлік көру, бейнелерді өңдеу және жалпы мақсатты сандық алгоритмдердің ашық бастапқы коды бар кітапхана. (EigenFaces, FisherFaces, LBPН);

dlib кітапханасы - машинамен оқыту алгоритмдері және әр түрлі қосымша көмекші құралдар кітапханасы. Бұл кітапханадан қыр нүктелерін іздеу үшін дайын оқытылған ААМ (Active Appearance Model) қолданылады. Оның негізгі мүмкіндігі, әрине, нейрондық желілерді (inference) жүктеу және іске қосу болып табылады. Бұл ретте модель терең оқытудың кез келген үш фреймворкінде құрылуы мүмкін — Caffe, TensorFlow немесе Torch. Үш танымал фреймворк қолдау арқылы, біз бір жалғыз фреймворкта бәрін қайта жасау қажетсіз, олардан жүктелген модельдердің жұмыс нәтижелерін біріктіре аламыз. Жүктеу кезінде модельдерді Caffe - де қолданушыға жақын ішкі көрініске айырбастау жүргізіледі.

openface кітапханасы

FaceNet технологиясына - негізделген терең түрде анықтайтын нейрондық желіні пайдаланатын ашық бастапқы коды бар тұлғаларды тану кітапханасы.

Keras кітапханасы - бұл кітапхана ең аз операциялар санымен нейрондық желілерді іске қосады. Нейрондық желі моделі ретінде keras модулінен Sequential қолданылады. Бұл кітапханадан қолданылған моделдері:

Dense - Keras қабатының тапсырмасы бар моделі. Біздің нейрондық желі тығыз (Dense) құрылымынан тұрады, әрбір нейрон келесі қабаттың барлық нейрондарымен байланысты.

Conv2d - толық екі өлшемді дискретті орауды есептеу үшін қолданады. Бұл функция матрица нысанын қабылдайды және әрбір ұяшық үшін көбейтеді. Соңында, ол әрбір ұяшық бойынша өлшенген соманың орташа мәнін қайтарады.

ZeroPadding2D - бұл сурет тензорының жоғарғы, төменгі, сол және оң бөліктерінде жолдар мен бағандарды қосуға арналған.

MaxPooling2D - кеңістіктік деректерді біріктірудің максималды әрекеті.

Flatten - бұл әдіс `rows()` және `columns()` сияқты көптеген әдістермен жұмыс істеу кезінде пайдаланады. 2D - құрылымын дейін 1D - массив құрылымына азайтады.

Dropout - оқыту кезінде әрбір жаңарту кезінде 0 - де енгізу бірліктерінің үлесін кездейсоқ орнатудан тұрады, бұл қайта жарактандыруды болдырмауға көмектеседі.

Activation - әрбір жаңарту кезінде 0 - де енгізу бірліктерінің үлесін кездейсоқ орнатудан тұрады, бұл қайта жарактандыруды болдырмауға көмектеседі.

PIL кітапханасы - PIL (Python сурет кітапханасы) - Python сурет графикасымен жұмыс істеуге арналған.

Мүмкіндіктері:

- бинарлық, жартылай тонды, индекстелген және толық түсті суреттерді қолдау;

- bmp пішімдерін қолдау BMP, EPS, GIF, JPEG, PDF.PNG, PNM, TIFF форматтарында пішімінде және басқа да оқу және жазу;

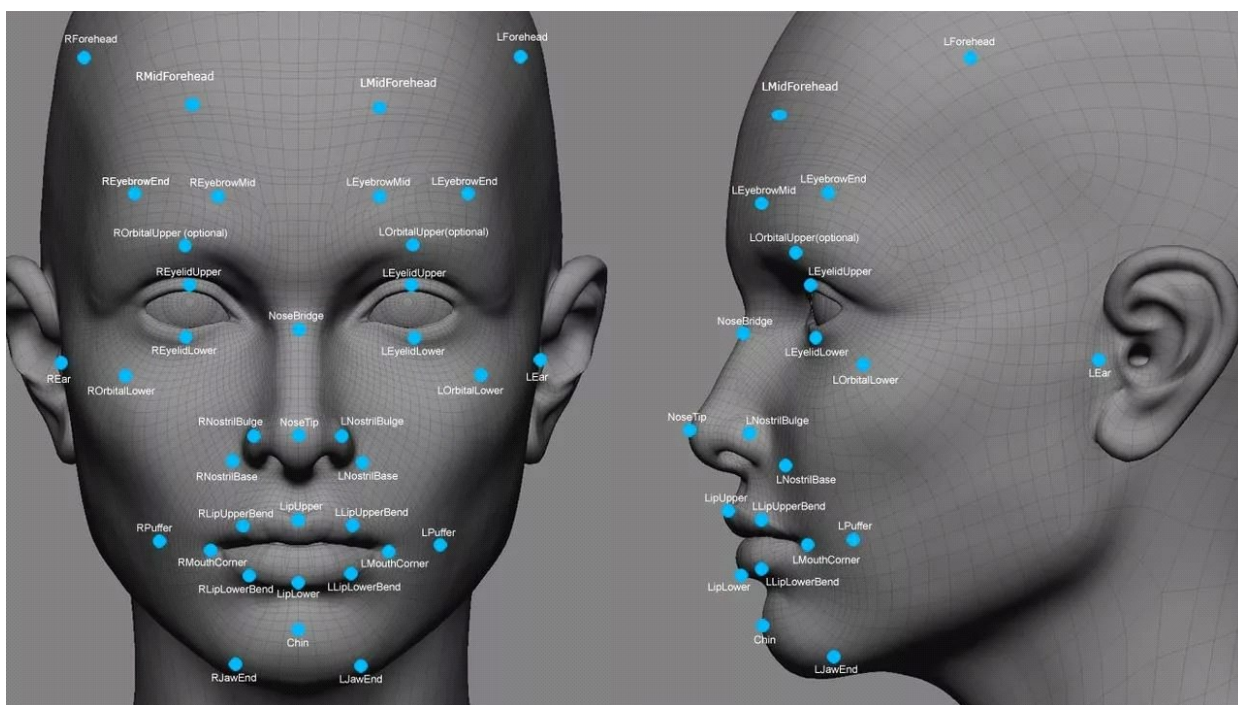
- көптеген пішімдерді қолдану ((ICO, MPEG, PCX, PSD, WMF форматында және т. б.) тек оқу үшін;

- суреттерді бір пішімнен екіншісіне түрлендіру;

- суреттерді өңдеу (түрлі сүзгілерді пайдалану, масштабтау, сурет салу, матрицалық операциялар және т. б.);

TensorFlow адам қабылдауының сапасына жетіп, бейнелерді автоматты табу және жіктеу мақсатында нейрондық желіні құру және жаттықтыру мәселелерін шешу үшін Google компаниясы әзірлеген машиналық оқытуға арналған ашық бағдарламалық кітапхана. Зерттеу үшін де, Google өз өнімдерін әзірлеу үшін де қолданылады. Кітапханамен жұмыс істеу үшін негізгі API Python үшін іске асырылды, сондай - ақ c++, Haskell, Java, Go және Swift үшін іске асыру бар.

Бағдарламалық қамтаманы құру барысында назар аударылатын белгілер, барлық адамдар көз арасындағы мұрын биіктігіне ара қашықтық шамамен бірдей. Сондықтан, бет жағында мұрын, көз, қас, ауыз, жақ, иек сияқты кейбір бақылау нүктелерін бөліп алған жөн. Осыдан кейін сол немесе басқа адамды идентификациялау үшін габаритті сүзгілердің мәнін дәл бақылау нүктелерінде есептеу жеткілікті. Кейбір бақылау нүктелерін еркін бетке бөлектеңіз (9 - сурет). Берілген адамның осы нүктелерінің қайда екенін анықтау үшін олардың орналасуы туралы кейбір түсініктері болуы қажет. Ол үшін "бірыңғай портрет" жасаймыз.



9 - сурет – Бақылау нүктелері

4.1 Бағдарламалық қамтаманың көрсетілімі

Бет әлпеті бойынша танып білу жүйесіне арналған программалық қамтаманың жұмысын айта кетер болсам. Адамды тану үшін оны кем дегенде бір рет көру керек. Дәлірек айтқанда, оның кем дегенде бір фотосуреті болуы немесе оны кейбір бейне түрінде есте сақтау. Бізге жаңа, белгісіз фотосуретті көрсеткенде, біз оны есте қалған барлық суреттермен салыстырамыз және жауап бере аламыз: біз осы адамды көрдік және оны анықтай аламыз немесе бұл адаммен біз кездеспедік. Осылайша, жоғарыда айтылған міндет келесіге негізделеді: екі фотосуретке ие (p_1 ; p_2) олардың бір адамға жататынын немесе жатпайтынын анықтау. (10 - сурет)Басқаша айтқанда, олар бір адамға тән қасиеттерге ие ма соны анықтайды.



10 - сурет – Суреттерді салыстыру

Бағдарлама жүйені нақты тұлғаны теңдестіруге «үйретеді». Бейнекамера және өзге құрылғылар, тіркеу кезеңінде, тұлға кескінің сандық көрінісін құру үшін тұлғаны сканерден өткізеді. Бетті сканерден өткізу 20 - 30 секундқа созылады, нәтижесінде бірнеше кескіндер құрылады. Егер камера алдында ешкім болмаса, камера жабылады. Мінсіз жағдайда, бұл кескіндер сәл әрқилы ракурстар мен бет - әлпетіне ие болады. Бұл жағдай нақты мәліметтерді алуға мүмкіндік береді. Арнайы бағдарламалық модуль бұл көріністі өңдеп, тұлғаның өзіндік ерекшеліктерін анықтайды, содан соң үлгі құрады. Беттің кейбір бөліктері уақыт өте келе өзгермейді, мысалы, көздің жоғарғы кескіндері, ауыздың шеті және бет сүйегі облысы. Биометриялық технологиялар үшін құрылған көптеген алгоритмдер, шаш ұзындығынан жоғары шекара облысын саралау үшін қолданылмайтындықтан, тұлға шашының мүмкін болатын өзгерістерін ескеруге мүмкіндік береді. Әр қолданушы кескінінің үлгісі биометриялық жүйенің дерекқорында сақталынады. Бейнекамерадан тұлға кескінін қабылдап, бет - әлпетті бағдарлама суретке айналдырады бет бейнесі 100x100 өлшемде болған кезде камера оны суретке тусіреді де, оны 224x224 қылып киып алып жақындатады. Үлгі сақталынатын сандық форматқа ауыстырады. Биометриялық жүйелерді қолдану кезінде, әсіресе бет - әлпет бойынша айырып тану жүйелері, дұрыс биометриялық сипаттамаларды енгізген жағдайда да сәйкестендіру жайлы шешім әрқашанда оң болып шықпайды. Бұл жағдай, біріншіден, көптеген биометриялық сипаттамалардың өзгеруімен тығыз байланысты. Жүйеде белгілі бір қате ықтималдылығының деңгейі бар. Егер камераға қараған кезде сурет анықтылығы 0,16 жоғары болған кезде бейнені таниды (11 - сурет).

Сонымен бірге әрқилы технологияларды қолдану кезінде қате болуы мүмкін. Енуді бақылау жүйелері үшін, биометриялық технологияларды қолдану кезінде, алдын ала анықтап алған жөн, не маңызды «бөгдені» өткізбеу әлде «өз

адамның барлығын өткізу. Қауіпсіздік жүйелеріндегі биометриялық технологиялар қолданушылары үшін маңызды фактор қолдану қарапайымдылығы болып табылады. Сканерден өтетін тұлға сипаттамалары ешқандай жайсыздықты сезінбеуі тиіс. Бұл жағдайда сөзсіз бет - әлпет бойынша айырып тану технологиясы айрықша қызықты әдіс болып табылады.

Қабылданған мәліметтер бір - бірімен сәйкестігін анықтау үшін дерекқорда сақталынатын үлгімен салыстырылады. Тексеріске қажетті ұқсастық деңгейі бір шек іспеттес және ол ДК қуаттылығы, тәулік уақыты, қызметкердің әрқилы типі және өзге факторлар үшін реттелуі мүмкін. Теңдестіруді анықтау, сәйкестендіру немесе айырып тану түрінде жүзеге асырылуы мүмкін. Анықтау кезінде дерекқорда сақталынатын үлгінің және қабылданған мәліметтердің теңдестігі расталынады.

Сәйкестендіру – дерекқорда сақталынатын үлгілердің бірінші жаңа кадр болған кезде камера ағынымен шақырылады.

```
Администратор: Anaconda Prompt - python app.py
0.09793174266815186
[0.24144625663757324, 0.09793174266815186 ]
0.188792884349823
[0.31399524211883545, 0.188792884349823 ]
0.09712719917297363
[0.28705596923828125, 0.09712719917297363 ]
0.10366976261138916
[0.2951921820640564, 0.10366976261138916 ]
0.06964915990829468
[0.2383684515953064, 0.06964915990829468 ]
0.08855223655700684
[0.2574577331542969, 0.08855223655700684 ]
0.07402026653289795
[0.25647395849227905, 0.07402026653289795 ]
0.06877243518829346
[0.23748505115509033, 0.06877243518829346 ]
0.07096356153488159
[0.2464160919189453, 0.07096356153488159 ]
0.067172110080719
[0.24198198318481445, 0.067172110080719 ]
0.15437012910842896
[0.33146703243255615, 0.15437012910842896 ]
0.06647729873657227
[0.25954848527908325, 0.06647729873657227 ]
```

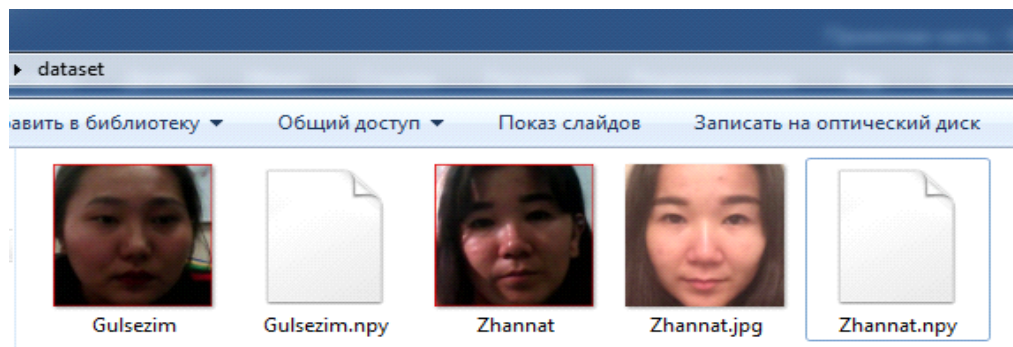
11 - сурет – ықтималдылығының деңгейі

Бағдарлама бет - әлпетті суретке тусіріп, оны базаға тіркейді. (12 - сурет) Бет әлпетті суретке айналдырып оны қырқып dataset папкасына жібереді. (13 - сурет)

```
Администратор: Anaconda Prompt - python script.py
(base) C:\Users\Lanos>cd Desktop
(base) C:\Users\Lanos\Desktop>cd project
(base) C:\Users\Lanos\Desktop\project>anaconda activate script.py
"anaconda" эх ты хсё тэсЕхээхц шыш тэхЭхц
ьвьрэфюц, шёяюыз хьюц яЕюуЕрььюц шыш ярьхСэЪь Ырщьюь.
(base) C:\Users\Lanos\Desktop\project>conda activate face
(face) C:\Users\Lanos\Desktop\project>python script.py
Using TensorFlow backend.
[ WARN:0] terminating async callback
2019-04-29 01:57:54.065918: W T:\src\github\tensorflow\tensorflow\core\framework
\allocator.cc:108] Allocation of 411041792 exceeds 10% of system memory.
2019-04-29 01:57:56.504058: W T:\src\github\tensorflow\tensorflow\core\framework
\allocator.cc:108] Allocation of 411041792 exceeds 10% of system memory.

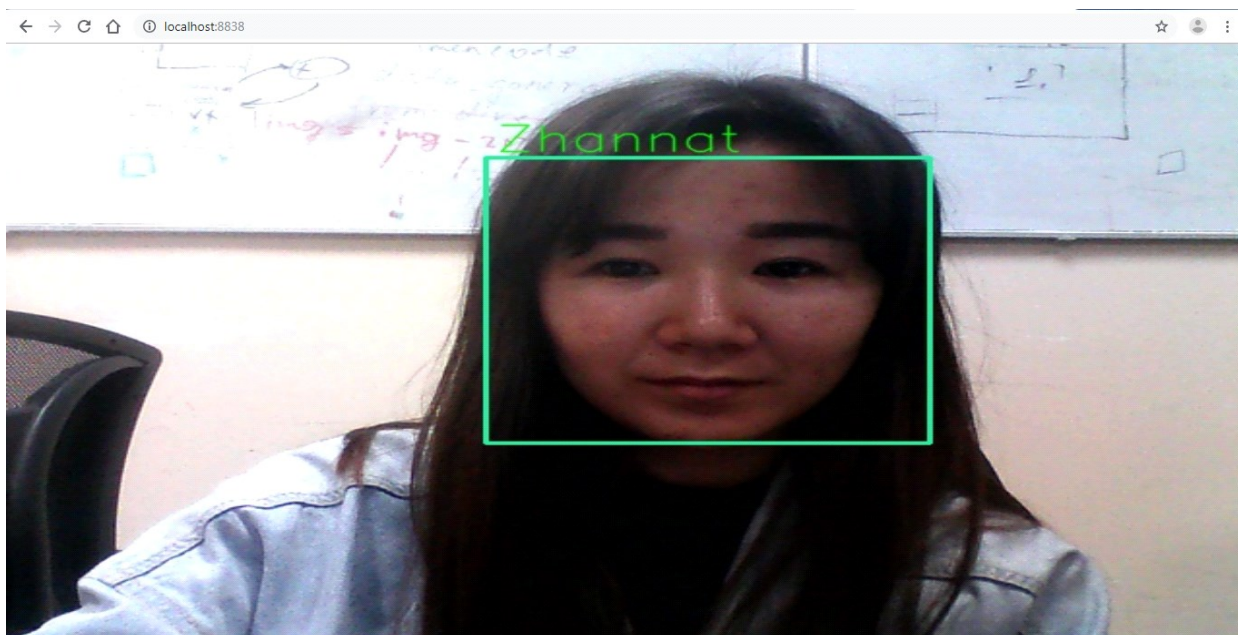
Enter the name: Zhannat_
```

12 - сурет – Суретті базаға тіркеу

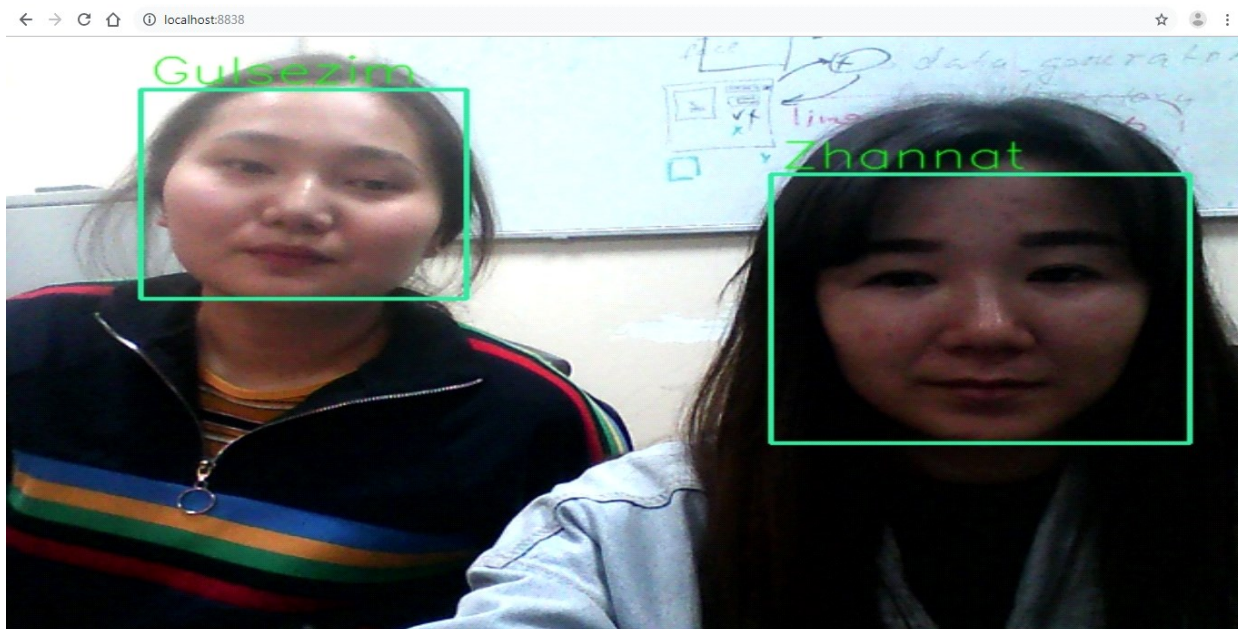


13 - сурет – dataset папкасы

Бейнені сурет ретінде алып оны камера арқылы 8838 адресімен вебке шығарып берді. Ары қарай бет - әлпетті тану жүреді (14 - сурет).



14 - сурет - Бет – әлпетті тану



15 – сурет – Бірнеше тұлғаны тану

ҚОРЫТЫНДЫ

Бұл дипломдық жұмыста бүгінгі күні бейнелерді тану саласының жағдайына арналған жүйеге талдау жүргізілді.

Қазіргі уақытта бейнелерді тану және автоматты жүйе оптикалық ақпаратты талдау проблемаларының әртүрлі аспектілеріне арналған көптеген жүйелер бар. Бейнелерді тану әдістері көп жерде толық қарастырылған, сондықтан бейнелерді тану жүйесінің математикалық моделін құру қиын емес. Бұл математикалық модельді зерттеу кезінде тану жүйесінің негізгі сипаттамаларын, атап айтқанда, әрекет алыстығын, дұрыс тану ықтималдығын және тану міндетін орындау үшін қажетті уақытты алуға болады.

Нейрондардың желілеріндегі есептеулер көп жағдайда ақпаратты мимен өңдеуге ұқсайды. Соңғы онжылдықта нейрокомпьютинг батыста өте танымал болды, онда ол коммерциялық өнімдерді өндірумен тығыз байланысты инженерлік тәртіпке айналып үлгерді. Жыл сайын нейрокомпьютингтің практикалық аспектілеріне арналған ондаған кітап шығады. Нейроқұрылымдарға арналған жаңа – аналогтық элементтік базаны құру бойынша жұмыстар қарқынды жүргізілуде. Барлық модельдерде сыртқы стимул қолдану үшін арнайы әзірленген бағдарламалық, өлшеуіш тор. Модельдерде өткізілген эксперименттер сериясының нәтижелері: Белгілі бір функцияны модельдеудің персептрон және кері тарату желісінің қабілеті желінің рұқсат етілген жалпы қатесіне байланысты. Хопфилд желісінің топологиялық құрылымы стимулды релаксациялау, прототип өндіру, қабылдау бистабильділігі ретінде түсіндіруге болатын оның қасиеттерін негіздейді. Одан әрі бейнелерді тану міндеті үшін неғұрлым тиімді алгоритмдерді алу мақсатында неғұрлым күрделі нейрондық желілердің бағдарламалық модельдерін және олардың комбинацияларын әзірлеу жоспарлануда.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Jain A.K., Bolle R., Pankanti S., eds., Biometrics: Personal Identification in Networked Society, Kluwer Academic, Norwell, Mass., 1999.
2. . Brin D., Transparent Society: Will Technology Force Us to Choose Between Privacy and Freedom? Addison-Wesley, New York, 1998.
3. Айзерман М.А., Браверман Э.М., Розоноэр Л.И. Метод потенциальных функций в теории обучения машин. - М.: Наука, 2004. - 384 с.
4. Горбань А., Россиев Д. Нейронные сети на персональном компьютере. //Новосибирск, Наука, 1996. – С 114 – 119.
5. Мазуров В.Д. Комитеты систем неравенств и задача распознавания // Кибернетика, 2004, № 2. С. 140-146.
6. Потапов А.С. Распознавание образов и машинное восприятие. - С-Пб.: Политехника, 2007. - 548 с
7. Растрингин Л. А., Эренштейн Р. Х. Метод коллективного распознавания. 79 с. ил. 20 см., М. Энергоиздат, 2006. – 80 с.
8. Рудаков К.В. Об алгебраической теории универсальных и локальных ограничений для задач классификации // Распознавание, классификация, прогноз. Математические методы и их применение. Вып. 1. - М.: Наука, 2007. - С. 176-200.
9. <https://ru.wikipedia.org/>
10. <https://www.itv.ru/products/intellect/faceintellect/>
11. <https://securityrussia.com/blog/face-recognition.html>
12. Jones, M., Viola, P. (2001) Robust Real-Time Face Detection. URL: International Journal of Computer Vision, 57(2), 137-154, 2004. Available from: <http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf> (ссылка проверена 28 мая 2013)
13. Turk, M., Pentland, A. (1991) Face Recognition Using Eigenfaces URL: <http://www.cs.ucsb.edu/~mturk/Papers/mturk-CVPR91.pdf> (ссылка проверена 28 мая 2013)
14. A novel approach for human face detection from color images under complex background <http://visgraph.cs.ust.hk/biometrics/Papers/Face/pr2001-10-1.pdf>(ссылка проверена 28 мая 2013)

А қосымшасы (міндетті)

Техникалық тапсырма

А.1 Жүргізілетін жұмыстардың мақсаты мен негізгі міндеттері

Бет-әлпетті анықтау және тану және сәйкестендіру теориясын қалыптастыру мен дамытуды ынталандыратын алғашқы тәжірибелік міндеттердің бірі болып табылады.

Аталған мақсатқа қол жеткізудің мынадай негізгі міндеттері бар:

- қоғамдық орындардағы жағдайды бақылаудың тиімділігін арттыру;
- барлық адамдардың көмегімен қылмыстарды жедел ашуға жәрдемдесу

Адамдарды тану және тану үрдісіне негізделген процедураларға қызығушылық, әсіресе практикалық қажеттіліктерге байланысты: қауіпсіздік жүйелерін тексеру, сот сараптамасы, телеконференция және т.б. Адамның бет-әлпетін айқындайтын дүниелік фактінің айқын екеніне қарамастан, компьютерді осы рәсімді өткізуге қалай үйрету керек екенін, соның ішінде жеке тұлғалардың сандық суреттерін кодтау және сақтау әдістерін білмейміз. Адамдардың ұқсастығын, оның ішінде оларды кешенді өңдеуді қоса есептегенде, неғұрлым айқын емес. Бетті тану мәселесіне зерттеудің бірнеше саласы бар:

- нейробиологиялық модельдер;
- нейрофизиологиялық модельдер;
- аппараттық-процедуралық үлгілер;
- танудың компьютерлік үлгілері.

А.2 Қолдану саласы

Бетті тану технологиялары графикалық файлдар мен бейне ағындарын автоматты іздеу және тану мүмкіндігін береді. Бетті тану технологиялары графикалық файлдар мен бейне ағындарын автоматты іздеу және тану мүмкіндігін береді. Қоғамдық қауіпсіздік. Қауіпсіздікті қамтамасыз ету-биометриялық сәйкестендіру жүйесін енгізу басталған бастапқы нүкте. Адамдарды қашықтықтан тану жүйелері адамдар жаппай болатын объектілердің қауіпсіздігін қамтамасыз ету үшін қолданылады.

А қосымшасының жалғасы

Ең күрделі міндет – топтағы адамды сәйкестендіру. Адам жүйемен өзара іс-қимыл жасамаса, камера объективіне қарамаса, бетін бұрады немесе жасыруға тырысады. Мысалы, көлік-ауыстырып салу тораптарында, метрода, ірі халықаралық іс-шараларда. Қашықтықтан биометриялық тану жүйесінде арнайы камералар қолданылды.

Видеоаналитиканың көмегімен онлайн режимінде ірі аумақтық-бөлінген объектілерде күдіктілерді, жоғалған адамдарды іздейді, оқиғалар мен инциденттерді тексереді, жолаушылар ағымына талдау жүргізеді. Олар бірқатар артықшылықтарға ие: тану жоғары шынайылығын қамтамасыз етеді, оларды алдау, көшіру немесе идентификаторды ұрлау мүмкін емес, оларды қолданыстағы қорғау жабдығымен оңай интеграциялау. Сіз тіпті бар бақылау камераларын пайдалануға болады. Адамдарды биометриялық сәйкестендіру жүйелері мұрағатта оқиғаларды тіркеп, қашықтықтан және өте тез жұмыс істейді.

Кассада билетті сатып алу кезінде әрбір сатып алушының тұлғасы автоматты түрде суретке түсіріледі және жүйеге жүктеледі. Осылайша матчқа келушілердің базасы қалыптасады. Егер сатып алу интернет немесе мобильді қосымша арқылы болса, онда авторизация алыстан "селфи" көмегімен мүмкін. Одан әрі, адам стадионға келгенде, жүйе оны ешқандай төлқұжатсыз таниды. Стадионға билет сатып алған, билетті басқа адамға беру немесе жалған билет арқылы өту мүмкін емес. Стадиондарда адамдарды қашықтықтан тану ірі аумақтық-бөлінген көлік объектілеріндегі сияқты принцип бойынша жұмыс істейді: егер адам стадионға кіруге тыйым салынған тұлғалар тізіміне енгізілсе, жүйе оны жібермейді.

А.3 Пайдаланушы интерфейстер

Тағы бір жаңа тренд стандартты сандық немесе сканерленген кескіндерде түсірілген көрнекі тері бөлшектерін пайдаланады. Тері құрылымының талдауы деп аталатын бұл әдіс адамның терісінде көрінетін бірегей сызықтарды, үлгілер мен кемшіліктерді математикалық кеңістікке айналдырады. Сынақтар көрсеткендей, бұл технологияны қосу арқылы бет тану тиімділігі 20-25% артады. Жылу камералары бет тану үшін кірістің тағы бір түрі термалды бейнекамера камераларын пайдалану болып табылады. Осы процедураның арқасында камералар тек бастың пішінін анықтайды, сонымен бірге көзілдірік, бас киім немесе макияж тәрізді заттарды елемейді.

А қосымшасының жалғасы

А.4 Аппараттық интерфейстер

Бетті тану үшін термалды бейнелерді пайдалану проблемасы дерекқорлар үшін шектелген. Қазіргі уақытта зерттеушілер осы технологияны нақты өмірде және операциялық пейзаждарда пайдалануды зерттеп, сондай-ақ термалды бейнелердің жаңа дерекқорын жасап жатыр. Зерттеуде ұзақ уақытты толқынды термиялық инфрақызыл суреттерді (LWIR) қабылдауға қабілетті төмен сезгіштік төмен ажыратымдылықты ферроэлектрлік электр сенсорлары пайдаланылады. Нәтижелері LWIR және кәдімгі визуальды камералардың біріктіруі сыртқы зондтарда жақсы нәтижелерге ие екендігін көрсетеді. Бұл комбинация камераға арналған өте қуатты тұлға тану бағдарламалық жасақтамасымен жұмыс істей алады.

А.5 Анықтамалар, терминдер және қысқартулар

Тарихи себептерге байланысты бейнені тану саласының негізі "бейне" ұғымымен байланысты. Ол ағылшын тіліндегі жұмыстардан алынған "pattern recognition" терминінде "pattern" терминінің "образ" мәнінен басқа "модель", "стиль", "режим", "зандылық", "әрекет бейнесі" мағынасына ие. Қазіргі заманғы тану және әсіресе жасанды интеллект оны кең мағынада қолданады, "бейне" - бұл біршама құрылымға жақын зерттелетін объектінің, құбылыстың немесе процестің сипаттамасы (эскизі).

Яғни, сипаттаманың ішінара анықталуы бейненің принциптік қасиеті болып табылады.

Сипаттамалардың (бейнелердің) негізгі мақсаты - оларды объектілердің сәйкестігін анықтау процесінде, яғни олардың ұқсастығын дәлелдеген кезде пайдалану, ол салыстыру арқылы жүзеге асырылады. Егер олардың сәйкестігін анықтау мүмкін болса, екі бейне ұқсас болып саналады. Атап айтқанда, егер олардың ұқсастығына қол жеткізілсе, сәйкестік орын алады деп есептеуге болады.

Егер тегіс объект болып табылатын сурет (көркем мата) арасындағы айырмашылықты қарастырса, және оның фотосуреттік немесе компьютерлік, телекамерамен немесе сканермен енгізілген суреті болса, объект арасындағы айырмашылық анық және теориялық және іс жүзінде түсініледі.

$f(a,b)$ - объект;

$g(x,y)$ - нысанның бейнесі.

А қосымшасының жалғасы

Нысан f әрпімен, ал сурет- g әрпімен белгіленеді.

Нысан белгілерінің x векторы арқылы белгілейміз. Бұл вектордың әрбір құрамдас бөлігі кейбір заңға сәйкес бөлінген i -ші белгінің мәні деп санаймыз. Демек, x векторы кездейсоқ Вектор болады. Мұндай векторлар көп n -өлшемді бейнелер кеңістігін құрайды, ал бұл кеңістіктің әрбір нақты нүктесі белгілі бір объектіге сәйкес келеді. Бейнелер кластары осы кеңістіктің аймағына сәйкес болады.

Бейнелерді тану міндеті әр түрлі класстарға сәйкес келетін облыстардың шекараларын анықтауға негізделеді. Егер өткен тәжірибеден x векторының осы екі таралуы белгілі болса, онда олардың арасында екі өлшемді кеңістікті екі аймаққа бөлетін $G(x_1, x_2)$ шекарасын орнатуға болады. Осылайша, $G(x_1, x_2)$ функциясының белгісіне байланысты жаңа бейнені қарау кезінде бұл бейненің I класына немесе II класына тиесілі екенін шешуге болады.

$G(x_1, x_2)$ функциясын дискриминанттық функция деп атайды, ал $g(x_1, x_2)$ белгісін анықтайтын техникалық құрылғы - бейнелерді тану блогы немесе классификатор деп аталады. N -Өлшем кеңістігіндегі жіктеушінің блок-схемасы бейнеленген.

Б қосымшасы (міндетті)

Бағдарламалық код үлгісі

App.py-бет-әлпетті камера арқылы тану

```
#!/usr/bin/env python
from importlib import import_module
import os
from flask import Flask, render_template, Response, jsonify
# import camera driver
from camera import Camera
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
def gen(camera):
    while True:
        try:
            frame = camera.get_frame()
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
        except Exception as e:
            print("error : ", e)
@app.route('/face_video')
def video_feed():
    return Response(gen(Camera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
if __name__ == '__main__':
    Camera.set_video_source(0)
    gen(Camera())
    app.run(host='localhost', port=8838, threaded=True)
```

base-camera.py-суретке түсіру жүйесі

```
import time
import threading
try:
```

В қосымшасының жалғасы

```
from greenlet import getcurrent as get_ident
except ImportError:
    try:
        from thread import get_ident
    except ImportError:
        from _thread import get_ident
class CameraEvent(object):
    def __init__(self):
        self.events = {}
    def wait(self):
        ident = get_ident()
        if ident not in self.events:
            self.events[ident] = [threading.Event(), time.time()]
        return self.events[ident][0].wait()
    def set(self):
        now = time.time()
        remove = None
        for ident, event in self.events.items():
            if not event[0].isSet():
                event[0].set()
                event[1] = now
            else:
                if now - event[1] > 5:
                    remove = ident
        if remove:
            del self.events[remove]
    def clear(self):
        self.events[get_ident()][0].clear()
class BaseCamera(object):
    thread = None
    frame = None
    last_access = 0
    event = CameraEvent()
    def __init__(self):
        if BaseCamera.thread is None:
            BaseCamera.last_access = time.time()
            BaseCamera.thread = threading.Thread(target=self._thread)
            BaseCamera.thread.start()
```

В қосымшасының жалғасы

```
        while self.get_frame() is None:
            time.sleep(0)
def get_frame(self):
    BaseCamera.last_access = time.time()
    BaseCamera.event.wait()
    BaseCamera.event.clear()
    return BaseCamera.frame
@staticmethod
def frames():
    raise RuntimeError('Must be implemented by subclasses.')
@classmethod
def _thread(cls):
    print('Starting camera thread.')
    frames_iterator = cls.frames()
    for frame in frames_iterator:
        BaseCamera.frame = frame
        BaseCamera.event.set()
        time.sleep(0)
        if time.time() - BaseCamera.last_access > 999999999:
            frames_iterator.close()
            print('Stopping camera thread due to inactivity.')
            break
    BaseCamera.thread = None
```

Camera.py-камераның тану жүйесі

```
from base_camera import BaseCamera
from func import FaceControl
from time import localtime, strftime
from datetime import datetime
from PIL import Image
import numpy as np
import time
import sys
import glob
import os
```

В қосымшасының жалғасы

```
import cv2
import random
from matplotlib.pyplot import imshow
from keras.models import Model, Sequential
from keras.layers import Input, Convolution2D, ZeroPadding2D, MaxPooling2D, Flatten, Dense, Dropout, Activation
from keras.preprocessing.image import load_img, save_img, img_to_array
from keras.applications.imagenet_utils import preprocess_input
from keras.preprocessing import image
import matplotlib.pyplot as plt
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
(width, height) = (224, 224)
idx=0
imgDim = 224
a = glob.glob('dataset/*.npy')
time_thold = 3
print(a)
class Camera(BaseCamera):
    video_source = 0
    @staticmethod
    def set_video_source(source):
        Camera.video_source = source
    @staticmethod
    def frames():
        fc = FaceControl()
        model = fc.loadVggFaceModel()
        camera = cv2.VideoCapture(Camera.video_source)
        detectface_vectors = []
        count=0
        time_e=0
        while True:
            ret, frame = camera.read()
            try:
                time_s = time.time()
                #x_sh=int(frame.shape[1]/6)
                #y_sh=int(frame.shape[0]/6)
                #rrec = cv2.rectangle(frame, (x_sh*2, y_sh*2), (x_sh*4, y_sh*4), (0, 0, 255),
```

2)

В қосымшасының жалғасы

```
#crop_rrec = frame[y_sh*2:y_sh*4, x_sh*2:x_sh*4]
faces = faceCascade.detectMultiScale(
    frame,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize = (100, 100)
)
for (x,y,w,h) in faces:
    face = frame[y:y+h,x:x+w]
    face_resize = cv2.resize(face, (224, 224))
    face_reshape = face_resize.reshape(1,224,224,3)
    cv2.rectangle(frame, (x,y), (x+w,y+h), (157,237,33), 2)
    img1_representation = model.predict(fc.preprocess_image(face_re-
shape))[0,:])
    cosines=[]
    facenames = []
    for i in range(len(a)):
        name = os.path.basename(a[i])
        base,typee = name.split('.')
        facenames.append(base)
        vectors = np.load(a[i]).item()
        cosine_similarity = fc.findCosineSimilarity(img1_representation,
vectors[0])
        cosines.append(cosine_similarity)
    minimum = cosines.index(min(cosines))
    text=""
    if cosines[minimum]<0.16:
        text = facenames[minimum]
        (text)
        print(cosines[minimum])
        print(cosines)
    else:
        text="NOT VERIFIED"
        print(cosines[minimum])
        print(cosines)
    cv2.putText(frame, text, (x+5,y-5), cv2.FONT_HERSHEY_SIMPLEX,
1,(0,255,0),1,cv2.LINE_AA)
    yield cv2.imencode('.jpg', frame)[1].tobytes()
```

В қосымшасының жалғасы

```
except:  
    pass
```

func.py-бет-әлпетті тану функциясы

```
from keras.models import Model, Sequential  
from keras.layers import Input, Convolution2D, ZeroPadding2D, MaxPooling2D, Flat-  
ten, Dense, Dropout, Activation  
from PIL import Image  
import numpy as np  
from keras.preprocessing.image import load_img, save_img, img_to_array  
from keras.applications.imagenet_utils import preprocess_input  
from keras.preprocessing import image  
import matplotlib.pyplot as plt  
class FaceControl():  
    def loadVggFaceModel(self):  
        model = Sequential()  
        model.add(ZeroPadding2D((1,1),input_shape=(224,224, 3)))  
        model.add(Convolution2D(64, (3, 3), activation='relu'))  
        model.add(ZeroPadding2D((1,1)))  
        model.add(Convolution2D(64, (3, 3), activation='relu'))  
        model.add(MaxPooling2D((2,2), strides=(2,2)))  
        model.add(ZeroPadding2D((1,1)))  
        model.add(Convolution2D(128, (3, 3), activation='relu'))  
        model.add(ZeroPadding2D((1,1)))  
        model.add(Convolution2D(128, (3, 3), activation='relu'))  
        model.add(MaxPooling2D((2,2), strides=(2,2)))  
        model.add(ZeroPadding2D((1,1)))  
        model.add(Convolution2D(256, (3, 3), activation='relu'))  
        model.add(ZeroPadding2D((1,1)))  
        model.add(Convolution2D(256, (3, 3), activation='relu'))  
        model.add(ZeroPadding2D((1,1)))  
        model.add(Convolution2D(256, (3, 3), activation='relu'))  
        model.add(MaxPooling2D((2,2), strides=(2,2)))  
  
        model.add(ZeroPadding2D((1,1)))
```

В қосымшасының жалғасы

```
model.add(Convolution2D(512, (3, 3), activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, (3, 3), activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, (3, 3), activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))

model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, (3, 3), activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, (3, 3), activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, (3, 3), activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))

model.add(Convolution2D(4096, (7, 7), activation='relu'))
model.add(Dropout(0.5))
model.add(Convolution2D(4096, (1, 1), activation='relu'))
model.add(Dropout(0.5))
model.add(Convolution2D(2622, (1, 1)))
model.add(Flatten())
model.add(Activation('softmax'))

from keras.models import model_from_json
model.load_weights('vgg_face_weights.h5')

vgg_face_descriptor = Model(inputs=model.layers[0].input, outputs=model.layers[-
2].output)
return vgg_face_descriptor
def preprocess_image(self,image_path):
    img_vector = preprocess_input(image_path)
    return img_vector
def findCosineSimilarity(self,source_representation, test_representation):
    a = np.matmul(np.transpose(source_representation), test_representation)
    b = np.sum(np.multiply(source_representation, source_representation))
    c = np.sum(np.multiply(test_representation, test_representation))
    return 1 - (a / (np.sqrt(b) * np.sqrt(c)))
```

В қосымшасының жалғасы

script.py-суретті түсіріп базаға жиберу

```
from func import FaceControl
import cv2
import os
import glob
import numpy as np
import time
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
#images = glob.glob('images/*.jpg')
cap = cv2.VideoCapture(0)
fc = FaceControl()
model = fc.loadVggFaceModel()
face_id = input('\n Enter the name: ')
count = 0
while True:
    #img = cv2.imread(i)
    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    faces = faceCascade.detectMultiScale(
        frame,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize = (180, 180)
    )
    for (x,y,w,h) in faces:
        face = frame[y:y+h,x:x+w]
        face_resize = cv2.resize(face, (224, 224))
        count+=1
        face_reshape = face_resize.reshape(1,224,224,3)
        img1_representation = model.predict(fc.preprocess_image(face_reshape))[0,:]
        b={0:img1_representation}
        np.save('dataset/'+str(face_id)+'.npy',b)
        cv2.imwrite('dataset/'+str(face_id)+'.jpg',face_resize)
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255), 2)
    time.sleep(4)
    k=cv2.waitKey(100) & 0xff
```

В қосымшасының жалғасы

```
if k==27:  
    break  
elif count >=1:  
    time.sleep(4)  
    break  
cap.release()  
cv2.destroyAllWindows()
```