

ҒЫЛЫМИ ЖЕТЕКШІНІҢ

ПІКІРІ

Дипломдық жұмыс

(жұмыс түрінің атауы)

Жумадил Біржан Айдосұлы

(білім алушының Т.А.Ә.)

6B06103 - Математикалық және компьютерлік модельдеу

(мамандық атауы мен шифрі)

Тақырыбы: Графикалық үлгілерді тану үшін машиналық көру жүйесін
өзірлеу

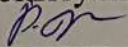
Дипломдық жұмыстың теориялық бөлімінде жұмыс тақырыбы бойынша аналитикалық шолу жасалып, зерттеулер жүргізілген.

Адам бетіндегі эмоцияларды тануды үшін FER – бет әлпетін тану кітапханасы, OpenCV – компьютерлік көру алгоритмдерінің кітапханасы қолданылған. Дипломдық жұмыстың программалық нәтижесінде камераны пайдаланып бет-әлпетті анықтауға және эмоцияны нақты уақытта тануға арналған жүйен жасалған. Жұмыста қойылған мақсат, міндеттер толық көлемде орындалған.

Жумадил Біржан өзінің еңбекқорлығымен, зерттеу және талдау қабілеттілігімен, тапсырылған жұмысты уақытымен орындауымен ерекшеленеді. Тақырып бойынша анықталған тапсырламалар жоғары дәрежеде орындалған, 97 % бағалануға лайықты.

Ғылыми жетекші:

аға оқытушы



Ергазина Р.А.

(қолы)

« 31 »

05

2024 ж.

РЕЦЕНЗИЯ

Дипломдық жұмыс

(жұмыс түрінің атауы)

Жумадил Біржан Айдосұлы

(білім алушының Т.А.Ә.)

6B06103 - Математикалық және компьютерлік модельдеу

(мамандық атауы мен шифрі)

Тақырыбы: Графикалық үлгілерді тану үшін машиналық көру жүйесін әзірлеу

Орындалды:

а) графикалық бөлім 12 бет

б) түсініктеме 18 бет

ЖҰМЫСҚА ЕСКЕРТУ

Дипломдық жұмыстың тақырыбы цифрлық технологияны қолдану саласындағы графикалық деректерді өңдеудің қазіргі таңдағы өзекті мәселелерін қамтиды. Қарастырылған алгоритмдерді одан әрі дамытуға және жетілдіруге болады. Жалпы жұмыс бойынша ешқандай ескертулер жоқ.

ЖҰМЫСТЫҢ БАҒАСЫ

Дипломдық жұмысты графикалық деректерді өңдеу барысында нақты уақыт режимінде бет пен эмоцияны тануға арналған жүйе жасау мәселесін шешудегі өзекті жұмыс ретінде санауға болады. Жумадил Біржан жұмыс тақырыбы бойынша машиналық көру мәселелеріне аналитикалық шолу жасап, тақырып аймағында зерттеулер жүргізіп, суреттердегі беттерді анықтау үшін OpenCV кітапханасын пайдалану арқылы графикалық үлгіні тану жүйесін жасаған.

Дипломдық жұмыс 6B06103 – “Математикалық және компьютерлік модельдеу” мамандығына қойылатын талапқа және ережелерге сай. Жұмыс авторы толығымен 97% өте жоғары бағалануға лайықты.

РЕЦЕНЗИЯ БЕРУШІ

Абай атындағы ҚазҰПУ

«Информатика және білімді ақпараттандыру»

кафедрасының қауымдастырылған профессоры,

педагогика ғылымдарының кандидаты

Г.А. Асқарова



**Университеттің жүйе администраторы мен Академиялық мәселелер департаменті
директорының ұқсастық есебіне талдау хаттамасы**

Жүйе администраторы мен Академиялық мәселелер департаментінің директоры көрсетілген еңбекке қатысты дайындалған Плагияттың алдын алу және анықтау жүйесінің толық ұқсастық есебімен танысқанын мәлімдейді:

Автор: Жумадил Біржан Айдосұлы

Тақырыбы: Графикалық үлгілерді тану үшін машиналық көру жүйесін әзірлеу

Жетекшісі: Рыскуль Ергазина

1-ұқсастық коэффициенті (30): 3.9

2-ұқсастық коэффициенті (5): 0.8

Дәйексөз (35): 0.4

Әріптерді ауыстыру: 0

Аралықтар: 0

Шағын кеңістіктер: 0

Ақ белгілер: 0

Ұқсастық есебін талдай отырып, Жүйе администраторы мен Академиялық мәселелер департаментінің директоры келесі шешімдерді мәлімдейді :

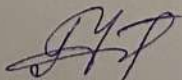
Ғылыми еңбекте табылған ұқсастықтар плагиат болып есептелмейді. Осыған байланысты жұмыс өз бетінше жазылған болып санала отырып, қорғауға жіберіледі.

Осы жұмыстағы ұқсастықтар плагиат болып есептелмейді, бірақ олардың шамадан тыс көптігі еңбектің құндылығына және автордың ғылыми жұмысты өзі жазғанына қатысты күмән тудырады. Осыған байланысты ұқсастықтарды шектеу мақсатында жұмыс қайта өңдеуге жіберілісін.

Еңбекте анықталған ұқсастықтар жосықсыз және плагиаттың белгілері болып саналады немесе мәтіндері қасақана бұрмаланып плагиат белгілері жасырылған. Осыған байланысты жұмыс қорғауға жіберілмейді.

Негіздеме:

Күні



Кафедра меңгерушісі

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Қ. И. Сатбаев атындағы ғылыми-техникалық университеті» коммерциялық емес
акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Жоғарғы математика және модельдеу кафедрасы

Жумадил Біржан Айдосұлы

Графикалық үлгілерді тану үшін машиналық көру жүйесін әзірлеу

ДИПЛОМДЫҚ ЖҰМЫС

6В06103 – «Математикалық және компьютерлік модельдеу»

Алматы 2024

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ
«Қ. И. Сатбаев атындағы ғылыми-техникалық университеті» коммерциялық емес
акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Жоғарғы математика және модельдеу кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

«Жоғарғы Математика және
Модельдеу» кафедрасының
менгерушісі физика-математика
ғылымдарының, қауымдастырылған
профессор Тулешева Г.А.

«31» 05 2024ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Графикалық үлгілерді тану үшін машиналық көру жүйесін әзірлеу»

6B06103 – «Математикалық және компьютерлік модельдеу»

Орындаған

Рецензент

Абай атындағы ҚазҰПУ

«Информатика және білімді

ақпараттандыру»

кафедрасының қауымдастырылған

профессоры педагогика

ғылымдарының кандидаты

Г.А. Асқарова

«31» маусым 2024ж.

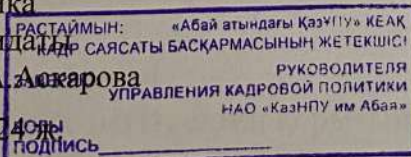
Жумадил Біржан

Ғылыми жетекші

«Жоғарғы математика және
модельдеу» кафедрасының
аға оқытушысы

Р. А. Ергазина

«28» маусым 2024 ж.



Алматы 2024

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

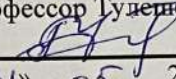
«Қ. И. Сатбаев атындағы ғылыми-техникалық университеті» коммерциялық емес
акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Жоғарғы математика және модельдеу кафедрасы

БЕКІТЕМІН

«Жоғарғы Математика және
Модельдеу» кафедрасының
меңгерушісі физика-математика
ғылымдарының, қауымдастырылған
профессор Тудетиева Г.А.


« 30 » 05 2024 ж.

6B06103 – «Математикалық және компьютерлік модельдеу»

Дипломдық жұмыс орындауға

ТАПСЫРМА

Білім алушы: Жумадил Біржан Айдосұлы

Тақырыбы: Графикалық үлгілерді тану үшін машиналық көру жүйесін әзірлеу
Университет ректорының 2024 жылғы «4» желтоқсандағы №548-П/Ө
бұйрығымен бекітілген.

Аяқталған жұмысты тапсыру мерзімі: « 5 » маусым 2024ж.

Дипломдық жұмыстың бастапқы деректері: Мен дипломдық жұмысымды Python
программалау тілінде жазамын. Және де Python программалау тілінің OpenCV
кітапханасын қолданамын.

Дипломдық жұмыста әзірленуге жататын мәселелердің тізбесі немесе
дипломдық жұмыстың қысқаша мазмұны:

А Жұмысқа жалпы шолу

Б Жұмыстың құрылым бөлімі

В Алынған мәліметтерге негізделген қорытынды

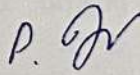
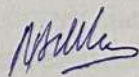
Ұсынылатын негізгі әдебиеттер: 7 кітап.

Дипломдық жұмысты дайындау
КЕСТЕСІ

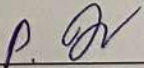
Бөлімдердің атаулары, әзірленетін мәселелердің тізбесі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Тақырыпқа байланысты арнайы әдибеттерді қарастыру	12.02.2024	орындалды
Дипломдық жұмыстың жоспарын құру	11.03.2024	орындалды
Негізгі бөлімді қарастыру	13.04.2024	орындалды
Дипломдық жұмысты қорытындылау	17.05.2024	орындалды

Қолтаңбалар

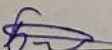
Аяқталған дипломдық жұмысқа консультанттар мен нормоконтролер оларға қатысты жұмыс бөлімдерін көрсете отырып

Бөлімдердің атаулары	Кеңесшілер Т.А.Ә. (мұғ. дәрежесі, атағы)	Қол қойылған Күні	Қолы
Негізгі бөлім	Ерғазина Р. А. “Жоғары математика және модельдеу” кафедрасының аға оқытушысы	28.05.24	
Норма бақылаушы	Шатманов Ж.Ж. физ.-мат. ғылымдарының кандидаты, қауымдастырылған профессор	31.05.2024	

Ғылыми жетекшісі

 Ерғазина Р. А.

Білім алушы тапсырманы орындауға алды

 Жумадил Біржан

Күні

«28» 05. 2024ж.

Аңдатпа

Менің бұл дипломдық жұмысым машиналық көруді қолдана отырып, графикалық бейнелерді, дәлірек айтқанда адам бейнесін (бетін) тануға, анықтауға арналған машиналық көру жүйесін жасауға арналған. Кодта Python программалау тіліндегі OpenCV кітапханасын бет детекторы арқылы суреттегі, ағымдағы бейнетаспадағы адам бетін анықтау үшін пайдаланатын жүйені енгізеді, және де беттерді анықтағаннан кейін жүйе бет эмоцияларын тану үшін терең оқыту алгоритмдерін қолданады. Кодта анықталған адам бетінің көлемін есептейтін функция енгізілген.

Бұл әдістерді қазіргі заманғы ғылым мен технологияның маңызды бағыты болып табылатын нақты уақыттағы графикалық деректерді талдау үшін пайдалануға болады. Бұл жүйені қолдана отырып алған нәтижелерді әртүрлі салаларда қолдануға мүмкіндік бар, соның ішінде бетті автоматты түрде анықтау, эмоцияларды талдау, мінез-құлықты бақылау және т.б. Жұмыс графикалық деректерді талдаудың әмбебап жүйесін құру үшін машиналық оқыту мен машиналық көрудің бірнеше әдістерін біріктіруді қамтиды.

Аннотация

Эта дипломная работа посвящена созданию системы машинного зрения для распознавания, идентификации графических изображений, а точнее человеческого изображения (лица), с использованием машинного зрения. В языке Python реализует систему, которая использует библиотеку OpenCV на языке программирования Python для обнаружения человеческого лица на изображении, видео в потоке с помощью детектора лица, и после обнаружения лиц система использует алгоритмы глубокого обучения для распознавания лицевых эмоций. В код встроена функция, которая вычисляет объем человеческого лица, определенный в коде.

Эти методы можно использовать для анализа графических данных в реальном времени, что является важной областью современной науки и технологий. Результаты, полученные с помощью этой системы, могут быть использованы в различных областях, включая автоматическое распознавание лиц, анализ эмоций, мониторинг поведения и т. д. Дипломная работа включает в себя интеграцию нескольких методов машинного обучения и машинного зрения для создания универсальной системы анализа графических данных.

Annotation

This thesis is devoted to the creation of a machine vision system for the recognition and identification of graphic images, or rather a human image (face), using machine vision. In Python, it implements a system that uses the OpenCV library in the Python programming language to detect a human face in an image, video in a stream using a face detector, and after detecting faces, the system uses deep learning

algorithms to recognize facial emotions. The code has a built-in function that calculates the volume of a human face defined in the code.

These methods can be used to analyze graphical data in real time, which is an important area of modern science and technology. The results obtained using this system can be used in various fields, including automatic face recognition, emotion analysis, behavior monitoring, etc. The thesis includes the integration of several machine learning and machine vision methods to create a universal graphical data analysis system.

МАЗМҰНЫ

КІРІСПЕ	8
1 Машиналық көру	9
2 Графикалық деректерді өңдеудің қолданыстағы әдістеріне шолу	11
2.1 Бет пен эмоцияны танудың негізгі тәсілдері	20
3 Графикалық үлгіні тану жүйесін әзірлеу	23
3.1 Суреттердегі беттерді анықтау үшін OpenCV кітапханасын пайдалану	23
3.2 Беттердегі эмоцияларды тану үшін терең оқыту алгоритмін қолдану	27
4 Тәжірибелер мен нәтижелер	31
5 Қолдану және перспективалар	32
ҚОРЫТЫНДЫ	33
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	34

КІРІСПЕ

Заманауи компьютерлік технологияларды дамуымен және де жасанды интеллекттің дамуымен графикалық деректерді талдауға және түсінуге қабілетті жүйелерді құру қажеттілігі артып келеді. Бұл саладағы негізгі бағыттардың бірі адамдардың бет-әлпеті сияқты графикалық бейнелерді тану және оларда көрсетілген эмоцияларды талдау. Бұл Бейнебақылау жүйелерін, фотосуреттерге тегтерді автоматты түрде қосуды, пайдаланушылардың реакцияларын талдауды және басқа салаларды қоса алғанда, қолданудың кең спектріне ие.

Бұл зерттеуде машиналық көруді қолдана отырып, графикалық кескіндерді тануға арналған машиналық оқыту жүйесін ұсынамын. Дипломдық жұмыс екі негізгі модульден тұрады: суреттердегі беттерді анықтау және сол беттерде көрсетілген эмоцияларды тану. Беттерді анықтау үшін біз машиналық көрудің Хаар Каскады әдістерін қолданамын, ал эмоцияларды тану үшін фотосуреттердегі адамның эмоционалды жағдайын дәл анықтайтын терең оқыту алгоритмін қолданамын.

Бұл дипломдық жұмысымды мен Python программалау тілінде жазып шықтым. Өйткені Python программалау тілі көптеген машиналық көруге байланысты жасалынатын жобаларға арналған өте функционалды, әмбебап кітапханалары бар. Python программалау тілінің кітапханаларының арасынан менің дипломдық жұмысымдағы мәселелерді шешетін OpenCV (Open Source Computer Vision Library) кітапханасын қолдандым.

Бұл жұмыстың мақсаты-нақты уақыт режимінде жұмыс істей алатын және әртүрлі салаларда қолданылатын жүйені құруға тырысу: бейнебақылау, медицина, психология, ойын-сауық индустриясы салаларында қолданысқа ие болатын. Дипломдық жұмыста әзірленген жүйенің тиімділігін бағалау бойынша эксперименттердің нәтижелерін ұсынамын және оның одан әрі жетілдіру әлеуетін талқылаймыз.

1 Машиналық көру

Машиналық көру бұл компьютерлерге суреттерді адамның көру қабілетімен бірдей көруге, анықтауға және өңдеуге, содан кейін тиісті нәтиже беруге мүмкіндік беретін жасанды интеллект саласы. Басқаша айтқанда, бұл адамның интеллектіне және оның компьютерге деген инстинктивті көзқарасына әсер етеді.

Машиналық көрудің қандай мүмкіндіктері бар және де машиналық көру қандай мәселелерді шешеді.

1 Суреттердің жіктеуі (Классификация):

Бүкіл кескінді "адамдар", "жануарлар" сияқты сыныптарға бөлу.

2 Объектілерді оқшаулау:

Суреттегі нысандардың бар-жоғын анықтау және олардың орналасқан жерін көрсету үшін фигураларды пайдалану

Кіріс: санаты / тегі бірдей бір немесе бірнеше нысанды қамтитын кескіндер.

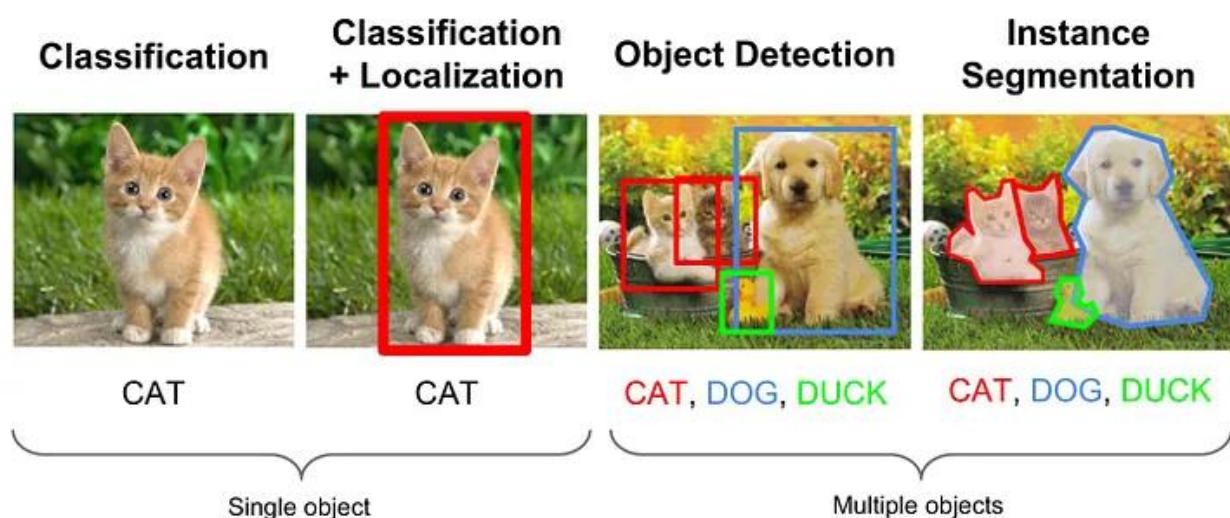
Нәтиже: екі нүктемен немесе бір нүктемен, ені мен биіктігімен анықталған бір немесе бірнеше жақтаулар.

3 Нысандарды анықтау:

Шектеу жақтауы арқылы объектілердің болуын анықтайды және суретте орналасқан объектілердің түрлерін немесе кластарын болжайды. Осылайша, бұл кескін классификациясы мен объектілерді локализациялауды біріктіру сияқты.

Кіріс: бір немесе бірнеше санаттағы / белгілердегі бір немесе бірнеше нысандары бар сурет.

Нәтиже: бір немесе бірнеше шектеу тіктөртбұрыштары (мысалы, нүктемен, енімен және биіктікпен анықталады) және әр шектеу тіктөртбұрышына арналған сынып белгісі.



Сурет – 1

Кескіннің бөліктерін анықтау және олардың қай объектіге жататынын түсіну

Бет пен эмоцияны талдауды қоса алғанда, графикалық бейнені тану саласындағы зерттеулер компьютерлік көру және жасанды интеллект саласында белсенді дамып келе жатқаны анық. Бұл технологиялардың бейнебақылау жүйелерінде, Медициналық диагностика, психологиялық зерттеулер мен ойын-сауық қолданбаларын қоса алғанда, көптеген қолданысқа ие.

Зерттеудің өзектілігі:

Цифрлық технологияны қолданудың кеңеюімен және машиналық көруге деген қызығушылықтың артуымен бет-әлпетті тану және эмоцияларды талдау міндеттері маңызды бола бастады. Мысалы, бейнебақылау саласында адамдарды өз бетінше автоматты түрде анықтауға, қылмыстың алдын алу немесе төтенше жағдайларда көмек көрсету мақсатында олардың эмоционалдық жағдайын талдауға қабілетті жүйелерге қажеттілік артып келеді.

Медициналық тәжірибеде бетті тану және эмоцияны талдау жүйелерін психологиялық жағдайларды диагностикалау, пациенттердің эмоционалды әлауқатын бақылау және де жеке медициналық көмек көрсету үшін пайдалануға болады. Ойын-сауық индустриясында бұл технологияларды пайдаланушылардың эмоционалды жағдайына жауап беретін интерактивті ойындар мен қосымшалар жасау үшін пайдалануға болады.

Осылайша, бет-әлпетті танудың және бейнелердегі эмоцияларды талдаудың тиімді әдістерін әзірлеу өзекті болып қала береді және адам қызметінің әртүрлі салаларына айтарлықтай әсер етуі мүмкін.

2 Графикалық деректерді өңдеудің қолданыстағы әдістеріне шолу

Графикалық деректерді өңдеу кескіндерді талдауға, өзгертуге және түсіндіруге бағытталған әдістер мен алгоритмдердің кең ауқымын қамтиды. Төменде компьютерлік көруде және басқа да байланысты салаларда қолданылатын графикалық деректерді өңдеудің негізгі әдістеріне шолу берілген.

Графикалық деректерді өңдеу оңай болу үшін көптеген өңдеу әдістерін қолданады. Солардың арасында маңыздыларының бірі түсті модельдер. Түсті модельдер компьютерлік көру мен кескінді өңдеуде маңызды рөл атқарады. Олар сандық кескіндерде түрлі-түсті ақпаратты кодтау мен ұсынудың әртүрлі тәсілдерін ұсынады. Ең жиі қолданылатын түсті модельдерге - RGB, HSV түсті модельдері жатады. Бұл модельдердің әрбірінің өзіндік артықшылықтары бар және әртүрлі қосымшаларда кеңінен қолданылады.

1 RGB (Red, Green, Blue)

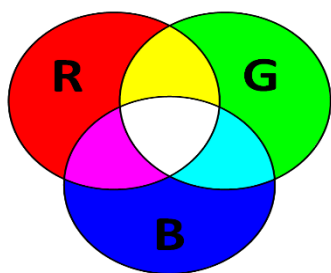
RGB моделі-сандық камераларда, мониторларда және басқа да көптеген дисплей құрылғыларында қолданылатын ең танымал түсті модельдердің бірі. Ол түстерді үш негізгі түстің тіркесімі ретінде ұсынады: қызыл (қызыл), жасыл (жасыл) және көк (көк). сурет-1

Артықшылықтары:

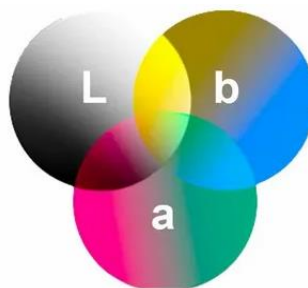
- Іске асыру және пайдалану оңай.
- Көптеген дисплей құрылғыларына қолдау көрсетіледі.

Кемшіліктері:

- Адамның түс қабылдауына әрдайым сәйкес келе бермейді.
- Реңктермен және түс қанықтылығымен жұмыс істеудегі қиындықтар.



Сурет – 2



Сурет – 3

Бұл түсті модель кеңінен қолданылады.

2 HSV (Hue, Saturation, Value)

HSV моделі адамға ыңғайлы, өйткені ол түс туралы ақпаратты (реңк) жарықтық (мән) және қанықтылық (қанықтылық) туралы ақпараттан тұрады. Бұл оны нысандарды бөлектеу және түсті сүзу сияқты кескінді өңдеуге қатысты тапсырмалар үшін ыңғайлы болады. сурет-2

Артықшылықтары:

- Адамның қабылдауы үшін интуитивті.
- Кескіндерді сегменттеу және өңдеу тапсырмалары үшін ыңғайлы.

Кемшіліктері:

- Әрқашан аппараттық қолдау көрсетілмейді.
- RGB-ден HSV-ге көшу қосымша есептеулерді қажет етеді.

RGB-ден HSV-ге түрлендірудің математикалық моделі:

RGB-ден HSV-ге түрлендіру барысын төменгі қадамдармен сипаттауға болады:

RGB мәндерін (1) нормалау: RGB мәндерін $[0, 255]$ диапазонынан $[0, 1]$ диапазонына түрлендіру.

$$R' = \frac{R}{255}, \quad G' = \frac{G}{255}, \quad B' = \frac{B}{255} \quad (1)$$

R' , G' және B' арасындағы максималды (2) және минималды мәндерді анықтаймыз:

$$C_{\max} = \max(R', G', B'), \quad C_{\min} = \min(R', G', B') \\ \Delta = C_{\max} - C_{\min} \quad (2)$$

Hue (реңк) мәнін (3) есептеу:

$$H = \begin{cases} 0, & \text{егер } \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right), & \text{егер } C_{\max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right), & \text{егер } C_{\max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right), & \text{егер } C_{\max} = B' \end{cases} \quad (3)$$

Saturation мәнін (4) есептеу (қанықтылық):

$$S = \begin{cases} 0, & \text{егер } C_{\max} = 0 \\ \frac{\Delta}{C_{\max}}, & \end{cases} \quad (4)$$

Value мәнін (5) есептеу (жарықтық):

$$V = C_{\max} \quad (5)$$

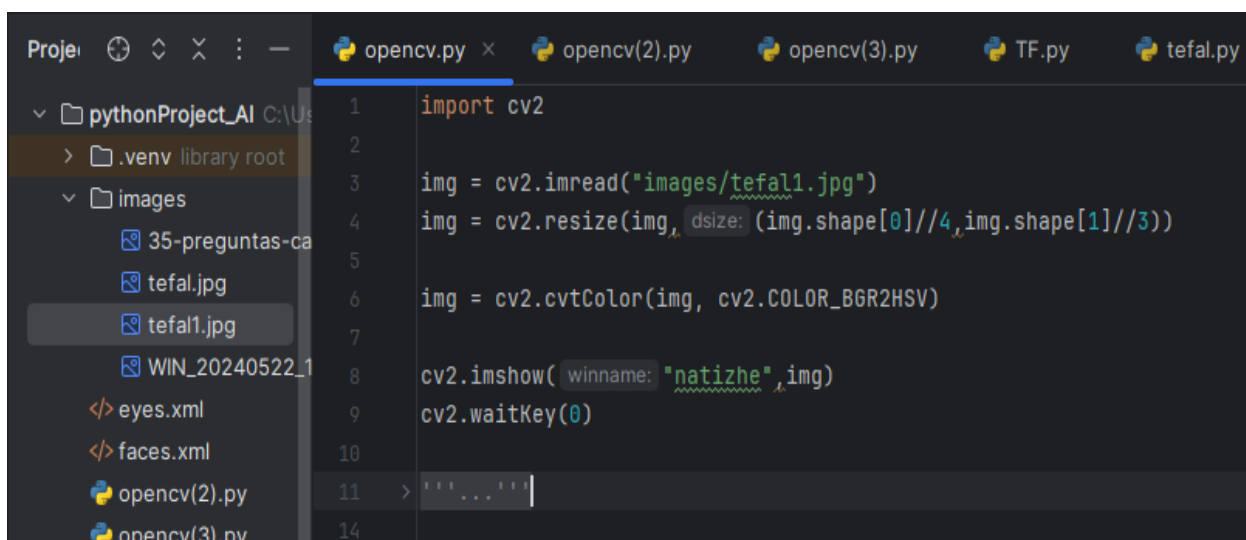
Бұл теңдеулер R , G және B мәндеріне негізделген Hue, Saturation және Value мәндерін есептеуге қолданылатын математикалық түрлендіру болып саналады.

RGB түстік моделінен HSV-ге түрлендіру компьютерлік көру тапсырмалары үшін пайдалы, өйткені ол түстермен тиімдірек жұмыс істеуге, нысандарды олардың түстерін таңдауға және кескінді сегменттеу мен сүзудің әртүрлі операцияларын орындауға мүмкіндік береді.

Ендігі кезекте бұл операцияларды PyCharm бағдарламасында жасап көруге болады.

Бұл түсті модельді қолдану үшін PyCharm бағдарламасын қолданамын. PyCharm бағдарламасына OpenCV кітапханаасын жүктейміз. Бұны біз `pip install python-opencv` кодын PyCharm бағдарламасының терминалына енгізу арқылы жүзеге асырамыз.

OpenCV кітапханасы көшірілген соң кодты жазуды бастаймыз.



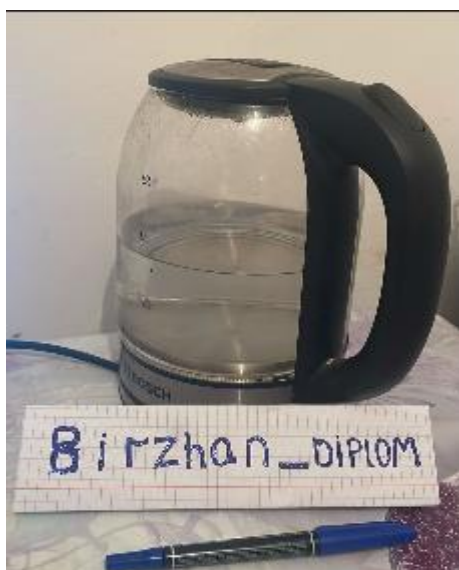
```
1 import cv2
2
3 img = cv2.imread("images/tefal1.jpg")
4 img = cv2.resize(img, dsize: (img.shape[0]//4, img.shape[1]//3))
5
6 img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
7
8 cv2.imshow( winname: "natizhe", img)
9 cv2.waitKey(0)
10
11 > '''...'''
14
```

Сурет – 4

- 1 `import cv2` коды арқылы OpenCV кітапханасын импорттаймыз.
- 2 `img = cv2.imread("images/tefal1.jpg")` әдісі арқылы `images` папкасындағы `tefal1.jpg` суретін PyCharm бағдарламасына оқытамыз.
- 3 `img = cv2.resize()` әдісі арқылы суреттің размерін өзімізге ыңғайлы жасаймыз.

4 `img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)` коды арқылы RGB форматындағы түсті HSV форматына аударамыз.

Содан алатын нәтижеміз:



Сурет – 6

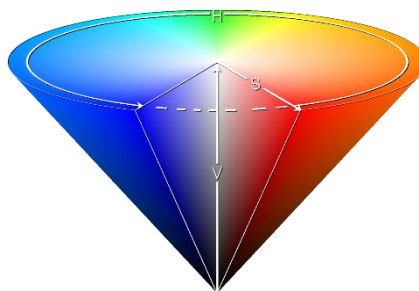


Сурет – 7

3 LAB (немесе CIELAB) түс моделі-компьютерлік көру және кескінді өңдеу саласындағы ең танымал түс модельдерінің бірі. Бұл модельде түс үш компонентпен ұсынылады: L (Lightness) жарықтықты, a және b түс координаттарын білдіреді. LAB моделі адамның түс қабылдауына жақынырақ және түстердің жоғары дәлдігін қажет ететін тапсырмаларда қолданылады.

Lab моделінің артықшылықтары:

- Перцептивті сызықтық: зертхананың түс мәндеріндегі өзгерістер адамның көзі қабылдайтын өзгерістерге көбірек сәйкес келеді.
- Жарықтықты түс ақпаратынан бөлу: бұл жарықтыққа немесе түске негізделген кескінді бөлек өңдеуге мүмкіндік береді.



Сурет – 8

Сұр реңктерге айналдыру

Сұр реңктердегі кескіндерді өңдеу компьютерлік көру мен машиналық оқытудың негізгі кезеңі болып табылады. Түсті кескінді сұр түске түрлендіру талдауды жеңілдетуге, деректер көлемін азайтуға және кескіннің текстуралық және жарықтылық ерекшеліктеріне назар аударуға мүмкіндік береді. Бұл бөлімде сұр реңктердегі кескіндерді өңдеудің негізгі әдістері мен принциптерін қарастырыңыз.

Сұр реңкте кескінді өңдеуді қолдану

Сұр реңктердегі кескіндерді өңдеу машиналық көрудің әртүрлі алгоритмдерін тиімді қолдануға мүмкіндік береді.

Түсті кескінді сұр реңктерге түрлендіру қызыл (R), жасыл (G) және көк (B) арналардың өлшенген орташа мәндерін қолданып жүзеге асырамыз. Адам көзі әртүрлі қарқындылықтағы әртүрлі қабылдайды, сондықтан бұл арналар әртүрлі өлшенуі керек. Ең көп таралған түрлендіру (6) формуласы келесідей көрсетілген:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (6)$$

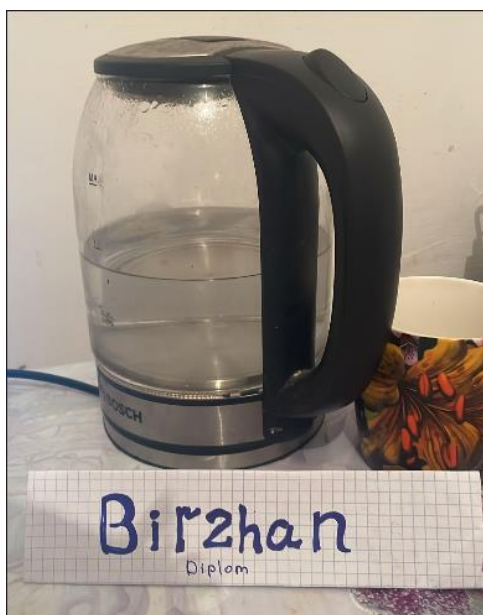
- Y-сұр түсті пиксель жарықтығының мәні.
- R-қызыл арна қарқындылығының мәні.
- G-жасыл арна қарқындылығының мәні.
- B - көк арнаның қарқындылығының мәні.

Контурларды табу

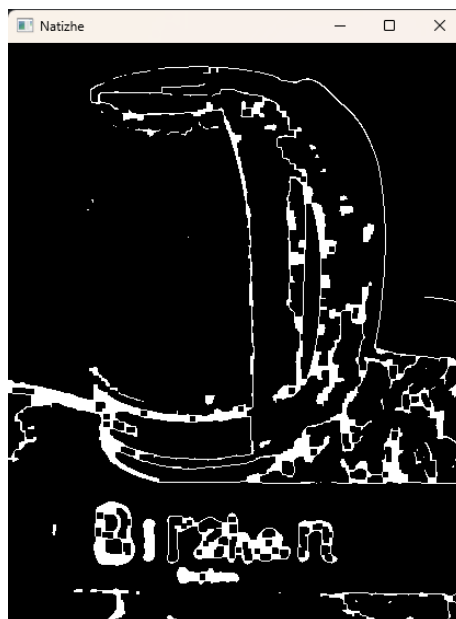
Суреттегі контурларды табу көптеген машиналық көру тапсырмаларында маңызды қадамы. Контурлар кескіндегі нысандардың шекараларын табуға көмектеседі, бұл әрі қарай өңдеу мен талдауды жеңілдетеді. Контурларды табудың пайдалы және қажет болуының бірнеше себептері:

- Нысандарды бөлектеу: контурлар суреттегі нысандардың шекараларын анықтауға көмектеседі. Бұл нысанды тану және сегменттеуге көмектеседі.
- Кескінді жеңілдету: контурлар тек маңызды бөліктерді бөлектеу және қажет емес бөлшектерді жою арқылы кескінді жеңілдетуге мүмкіндік береді. Бұл енгізген деректер көлемін азайтады және оларды өңдеуді жеңілдетеді.
- Пішінді талдау: контурлар объектілердің пішінін талдауға мүмкіндік береді, бұл жіктеуге, күйін келтіруге тағы басқа тапсырмалар үшін қажет болады.
- Өлшемдер мен көрсеткіштер: контурлар әр түрлі қосымшаларда маңызды болуы мүмкін аудан, периметр және басқа көрсеткіштер сияқты объектінің параметрлерін өлшеуге мүмкіндік береді.

- Алдын ала өңдеу: контурды анықтау көбіне бетгі тануға, нысандарды қадағалау және тағы басқа күрделі тапсырмаларды орындамағаннан бұрын алдын-ала өңдеу кезеңі ретінде пайдаланамыз.



Сурет – 9



Сурет – 10

Біз суреттегі дененің контурын табу үшін жоғарыда көрсетілген операцияларды қайталаймыз. Ол үшін контурын анықтауға арналған бейнемізді PyCharm бағдарламасына енгіземіз. Менде бейне тefальдың бейнесі.

```

1  import cv2
2  import numpy as np
3
4  img = cv2.imread('images/tefal.jpg')
5  # imread әдісі арқылы images файлындағы tefal.jpg суретін оқимыз
6  img = cv2.resize(img, dsize: (img.shape[1]//3, img.shape[0]//3))
7  # Разерін өзгертеміз
8  img = cv2.GaussianBlur(img, ksize: (5,5), sigmaX: 0)
9  # Бұлдырлатамыз. Бұл бағдарламаға сурет қиынға түспеу үшін
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
11 # Түсті моделін өзгертеміз
12 img = cv2.Canny(img, 200, 200)
13 # Матрица жасаймыз 5*5
14 kernel = np.ones( shape: (5,5), np.uint8)
15 # Контурды анықтаймыз
16 img = cv2.dilate(img, kernel, iterations=1)
17 # Дефекттерді азайтамыз
18 img = cv2.erode(img, kernel, iterations=1)
19 #print(img.shape)
20 # Нәтижені экранға шығарамыз
21 cv2.imshow( winname: 'Nativze', img)
22 # көру уақыты
23 cv2.waitKey(0)

```

Сурет – 11

Жоғарыда көрсетілген операциялар арқылы осындай нәтиже ала аламыз. Менде берілген суреттегі тefальдың контурларын OpenCV кітапханасы арқылы анықтадым.

Кескінді бұлыңғырлау

Gaussian Blur (гаусс бұлыңғырлығы) - кескінді тегістеу (бұлыңғырлау) және шуды жою үшін қолданылатын кескінді сүзу әдісі. Бұл әдіс пиксель мәндерін есептеу үшін Гаусс үлестірімін қолдануға негізделген, бұл пикселдер арасындағы ауысуларды тегістеу мен жиектердің айқындылығын азайтуға мүмкіндік береді.

Екі өлшемді кеңістіктегі Гаусс үлестірімінің (ядросының) (7) формуласы төменде көрсетілген:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (7)$$

Бұл жерде:

$G(x, y)$ - Гаусс ядросының нүктедегі мәні

σ - Гаусс функциясының "қоңырауының" енін анықтайтын Гаусс үлестірімінің стандартты ауытқуы

Гаусс ядросының кескінін жинақтау (8) үшін келесі формула қолданылады:

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x + i, y + j) \cdot G(i, j) \quad (8)$$

Мұнда - бұлыңғыр суреттегі нүктедегі пиксель мәні

$I(x + i, y + j)$ - нүктедегі бастапқы кескіннің пиксель мәні

$G(i, j)$ - Гаусс ядросының нүктедегі мәні

k - ядро өлшемі (әдетте ядро ені мен биіктігінің жартысы).

Егер осы әдісті 5*5 матрицасымен (9) бағдарламамызда қолдансақ Гаусс ядросы шамамен мынандай түрге ие болады:

$$G = \frac{1}{2\pi} \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix} \quad (9)$$

Енді PyCharm бағдарламасына салып көреміз.

Код іске асуы келесідей астында көрсетілген. Кодты жазып бастамай тұрып алдымен images файлына өзімізге керек бейнені енгізіп аламыз. Ол үшін кез-келген өзіңізге керек суретті персоналды компьютерге жүктейміз. Одан кейін жүктелген суретті жұмыс үстелінен PyCharm бағдарламасына тінтуір арқылы енгіземіз.

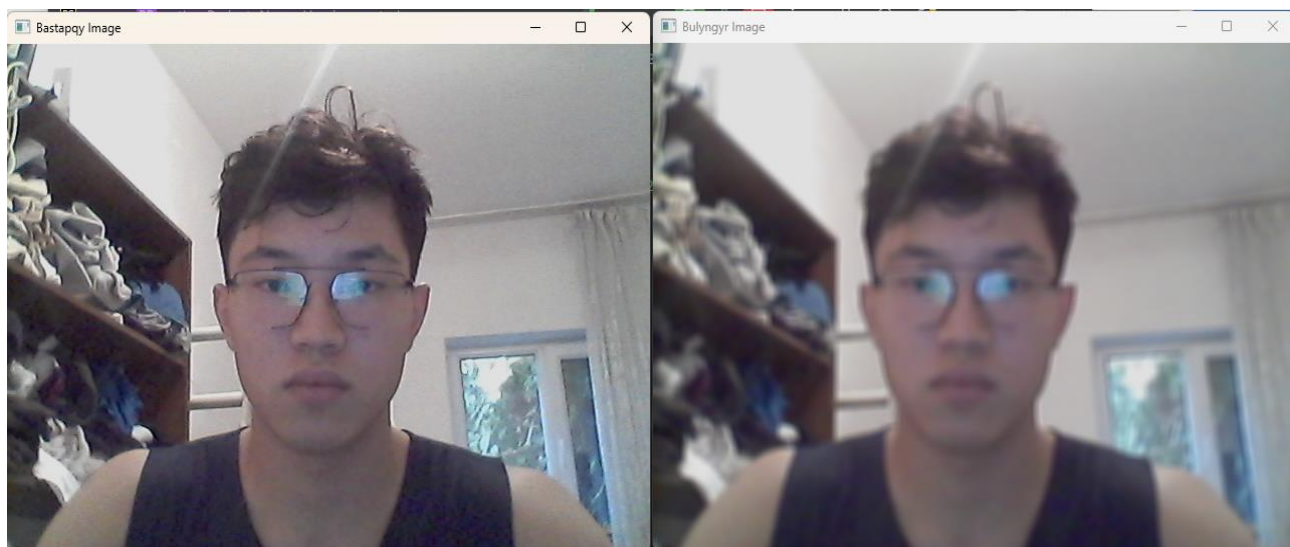
```

1
2 import cv2
3
4 # Суретті жүктейміз
5 img = cv2.imread('images/WIN_20240522_17_51_12_Pro.jpg')
6
7 # Gaussian Blur әдісін қолдану
8 bulynqyr_img = cv2.GaussianBlur(img, ksize: (9, 9), sigmaX: 7)
9
10 # Нәтижелерін шығару
11 cv2.imshow( winname: 'Bastapqy Image', img)
12 cv2.imshow( winname: 'Bulynqyr Image', bulynqyr_img)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15 >

```

Сурет – 12

Нәтижесінде осындай сурет аламыз:



Сурет – 13

Біз жоғарыда сурет пен бейнероликті өңдеу тәсілдерін шолып өттік. Бұл әдістер жұмыстарға қажет болады.

2.1 Бет пен эмоцияны танудың негізгі тәсілдері

Менің дипломдық жұмысымда эмоцияларды анықтайтын бағдарлама жасау үшін тек графикалық деректерді өңдеу жеткіліксіз. Сондықтан біз бет пен эмоцияны танудың негізгі тәсілдерін қолданамыз. Ол үшін мен бетті анықтайтын Каскад Хаара бейнелерді өңдеу және объектілерді тану үшін жсалған әдісін қолданамын. Бұл әдіс маған суреттен, видеодан, ағымдағы видеодан адам бетін анықтауға көмектеседі.

Ал адам бетіндегі эмоцияларды анықтау үшін мен Python бағдарламалау тілінің FER кітапханасын қолдандым. Бұл кітапхана менің жобамдағы мәселерді шеше алатын көтеген функциялары бар. FER (Facial Expression Recognition) – бұл адам бетінің эмоцияларын тану үшін қолданылатын кітапхана. Python тілінде жазылған және танымал машиналық оқыту кітапханаларын пайдаланып жасалған. FER кітапханасы қарапайым және қолдануға ыңғайлы болып келеді, сонымен қатар нақты уақытта бет эмоцияларын тануға мүмкіндік бере алады.

Алдымен Каскад Хаара-ға тоқталайық.

Каскад Хаара

Хаар каскады – бұл бет, көз немесе машина сияқты өнертабыстардағы заттарды тануға, анықтауға арналған машиналық оқыту әдісі. Ол Хаар әдістерін қолдана отырып дайындалған каскадты жіктеуіштерді қолдануға негізделіп жасалған.

Каскад Хаара бейнелерді өңдеу және объектілерді тану үшін бірнеше математикалық түсініктер мен формулаларды қолданады. Міне, олардың негізгісі:

1 Хаар ерекшеліктері (Haar Features)

Хаар ерекшеліктері көршілес тікбұрышты аймақтардың арасындағы жарықтық айырмашылығын білдіреді.

Бұл характеристикаларды есептеу үшін келесі формула (10) қолданылады:

$$H = \sum_{i \in R_1} I(i) - \sum_{i \in R_2} I(i) \quad (10)$$

Мұнда:

- H — хаар ерекшелігінің мәні
- $I(i)$ — i -ші пикселдің интенсивтілігі
- R_1 және R_2 - қарқындылық айырмашылығы есептелетін аймақтар

2 Интегралды бейне

Интегралды бейне кез келген тікбұрышты аймақтағы пикселдердің қосындысын жылдам есептеуге (11) мүмкіндік береді. Ол келесі түрде анықталады:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (11)$$

Мұнда:

- $ii(x, y) — (x, y)$ нүктесіндегі интегралды бейненің мәні
- $I(x', y') — (x', y')$ нүктесіндегі пикселдің қарқындылығы

Тікбұрышты аймақтағы пикселдердің қосындысы (12) интегралды бейне мәндері арқылы анықталады:

$$S(A, B, C, D) = ii(D) - ii(B) - ii(C) + ii(A) \quad (12)$$

Мұнда:

- $S(A, B, C, D) —$ тікбұрышты аймақтағы пикселдердің
- $A, B, C, D —$ тікбұрышты аймақтың бұрыштық нүктелері

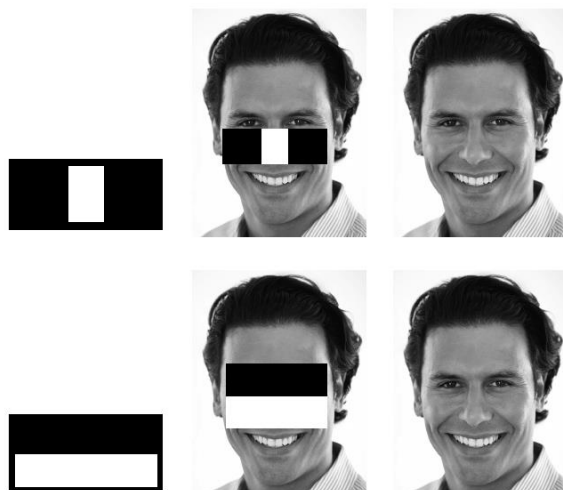
3 Adaboost алгоритмі

Adaboost алгоритмі Хаар ерекшеліктерінің ішінен ең маңыздысын таңдау және күшті классификаторды құру үшін қолданылады. Adaboost алгоритмінің негізгі идеясы әлсіз классификаторлардың көптеген түрін бір күшті классификаторға біріктіру болып табылады. Салмақтық коэффициенттер әр кезеңде (13) жаңартылып отырады:

$$w_i = w_i \cdot e^{\alpha \cdot y_i \cdot h(x_i)} \quad (13)$$

Мұнда:

- $w_i — i$ -ші үлгінің салмақтық коэффициенті
- $\alpha —$ әлсіз классификатордың ықпалын анықтайтын коэффициент
- $y_i — i$ -ші үлгінің нақты белгісі
- $h(x_i) —$ әлсіз классификатордың i -ші үлгіге арналған болжамы



Сурет – 14 – Хаар Каскадының белгілері бейнеленген

FER кітапханасы

FER кітапханасы эмоцияларды тану үшін алдын ала үйретілген нейрондық желілерді пайдаланады. Бұл модельдер көптеген суреттерді пайдаланып үйретіледі, сондықтан нақты формула жоқ, бірақ нейрондық желілердің негізгі принциптері қолданылады.

Нейрондық желілердің негізгі операциялары:

1 Аффиндік (14) түрлендірулер:

$$z = W \cdot x + b \quad (14)$$

2 Белсендіру (15) функциялары (activation functions):

- Sigmoid:

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (15)$$

- ReLU:

$$\text{ReLU}(z) = \max(0, z)$$

- Softmax (көп класстық классификация үшін):

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (16)$$

FER кітапханасы осындай нейрондық желілердің архитектураларын пайдалана отырып, эмоцияларды тануды орындайды.

3 Графикалық үлгіні тану жүйесін әзірлеу

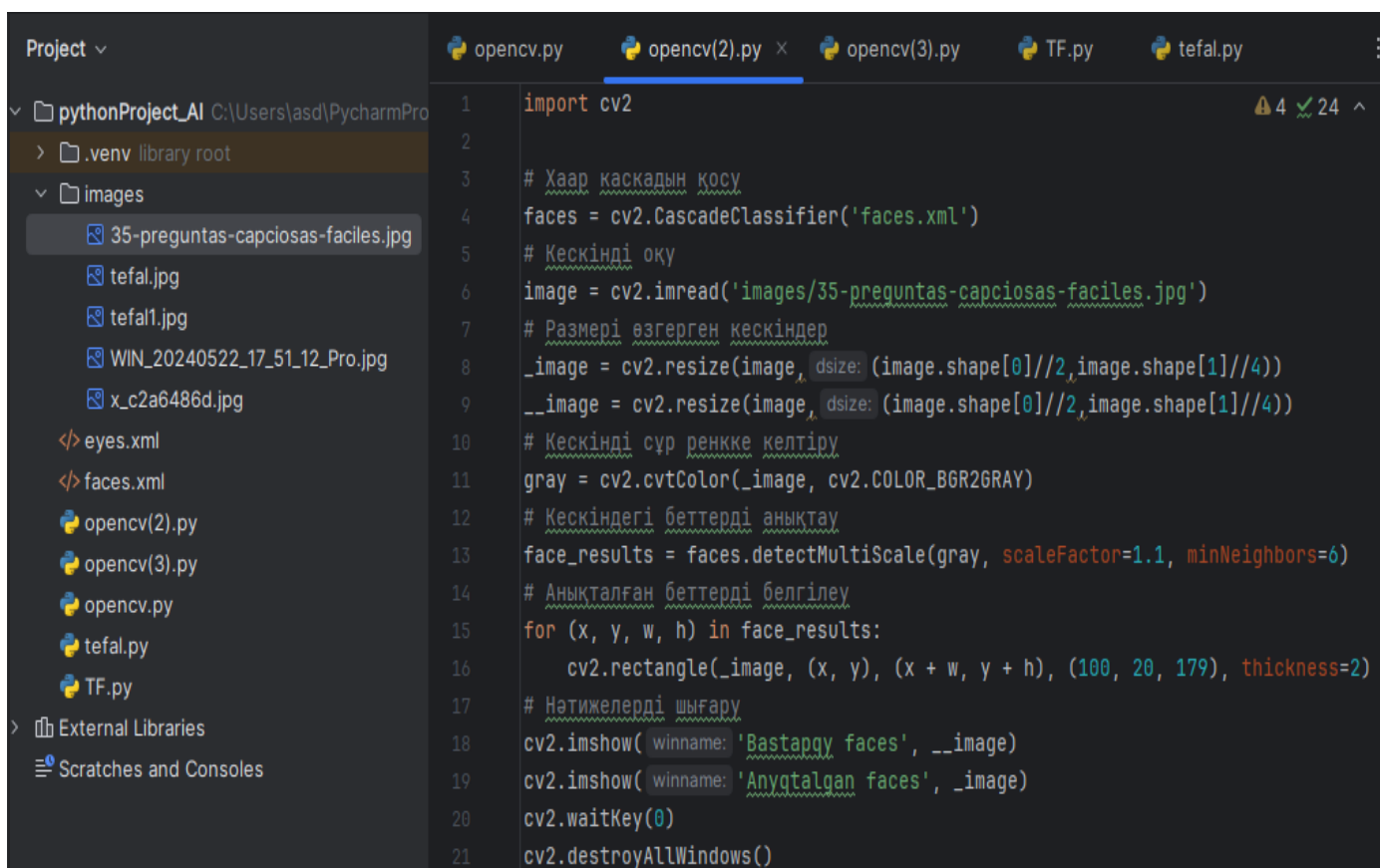
Ендігі кезекте біз жоғарыда көрсетілген әдістер мен графикалық өңдеулерді пайдаланып адам бетінің эмоцияларын анықтайтын бағдарлама жасаймыз. Ол үшін ең алдымен бізге керекті Python бағдарламау тілінің керекті кітапханаларын жүктейміз.

3.1 Суреттердегі беттерді анықтау үшін OpenCV кітапханасын пайдалану

Кескіндегі беттерді анықтау үшін OpenCV кітапханасын пайдаланамыз. Оны жүктеу үшін PyCharm бағдарламасының терминалына мына кодты енгіземіз: `pip install opencv-python`

OpenCV кітапханасы жүктеліп болған соң, PyCharm бағдарламасында бізге керек суреттерді персоналды компьютерден жүктеп аламыз.

Келесі қадам Хаар Каскадын қолдану. Ол үшін <https://github.com/opencv/opencv/tree/master/data/haarcascades> сайтынан Хаар Каскадын аламыз. Хаар Каскады бізге бейнедегі адам беттерін анықтауға мүмкіндік береді. Сайттағы файлды оқу үшін PyCharm бағдарламасында `faces.xml` деген `.xml` форматындағы файл ашамыз. Файлға сайттағы модельді жүктейміз.



```
1 import cv2
2
3 # Хаар каскадын қосу
4 faces = cv2.CascadeClassifier('faces.xml')
5
6 # Кескінді оқу
7 image = cv2.imread('images/35-preguntas-capciosas-faciles.jpg')
8
9 # Размери өзгерген кескіндер
10 _image = cv2.resize(image, dsize=(image.shape[0]//2, image.shape[1]//4))
11 __image = cv2.resize(image, dsize=(image.shape[0]//2, image.shape[1]//4))
12
13 # Кескінді сұр ренкке келтіру
14 gray = cv2.cvtColor(_image, cv2.COLOR_BGR2GRAY)
15
16 # Кескіндегі беттерді анықтау
17 face_results = faces.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=6)
18
19 # Анықталған беттерді белгілеу
20 for (x, y, w, h) in face_results:
21     cv2.rectangle(_image, (x, y), (x + w, y + h), (100, 20, 179), thickness=2)
22
23 # Нәтижелерді шығару
24 cv2.imshow( winname: 'Bastapgy faces', __image)
25 cv2.imshow( winname: 'Anyqtalqan faces', _image)
26
27 cv2.waitKey(0)
28
29 cv2.destroyAllWindows()
```

faces = cv2.CascadeClassifier('faces.xml') коды арқылы Хаар Каскадын іске қосамыз. Хаар Каскады әдісін faces. арқылы функцияларын қолдана аламыз. Кодты орындағаннан соң төмендегідей нәтиже аламыз:



Сурет – 16

Нәтижесінде маған кескіндегі анықталған адам беттерін шығарып берді. Анықталған адам беттері тіктөртбұрыш қоршаулармен белгіленген. Қоршаулар түсін cv2.rectangle(_image, (x, y), (x + w, y + h), (100, 20, 179), thickness=2) осы кодтағы 3 сан арқылы кез-келген қалаған түске бояса болады.

Хаар Каскадын ағымдағы видеоға қолдану

Осы операцияларды қолдана отырып ағымдағы видеодан адам бетін анықтайтын бағдарлама жасауға болады. Ол үшін:

```
1 cap = cv2.VideoCapture(0)
```

Камерадағы видеоны алу үшін төмендегі кодты жазамыз. 0 саны персоналды компьютердегі камераның нумерациясы.

```
2 cap.set(3, 500) # Ені  
   cap.set(4, 600) # Биіктігі
```

Көрсетілетін бейненің көлемі қандай болатынын осы код арқылы белгілейміз.

```
3 while True:  
   success, img = cap.read()  
   if not success:  
       print("Қате")  
       break
```

Цикл арқылы ағымдағы кадрды оқуға мүмкіндік береміз.

```
4 key = cv2.waitKey(1)  
   if key == ord('g'):  
       break
```

Кадрды аяқтау үшін "g" батырмасын қосамыз. "g" батырмасымен кадрды тоқтатамыз.

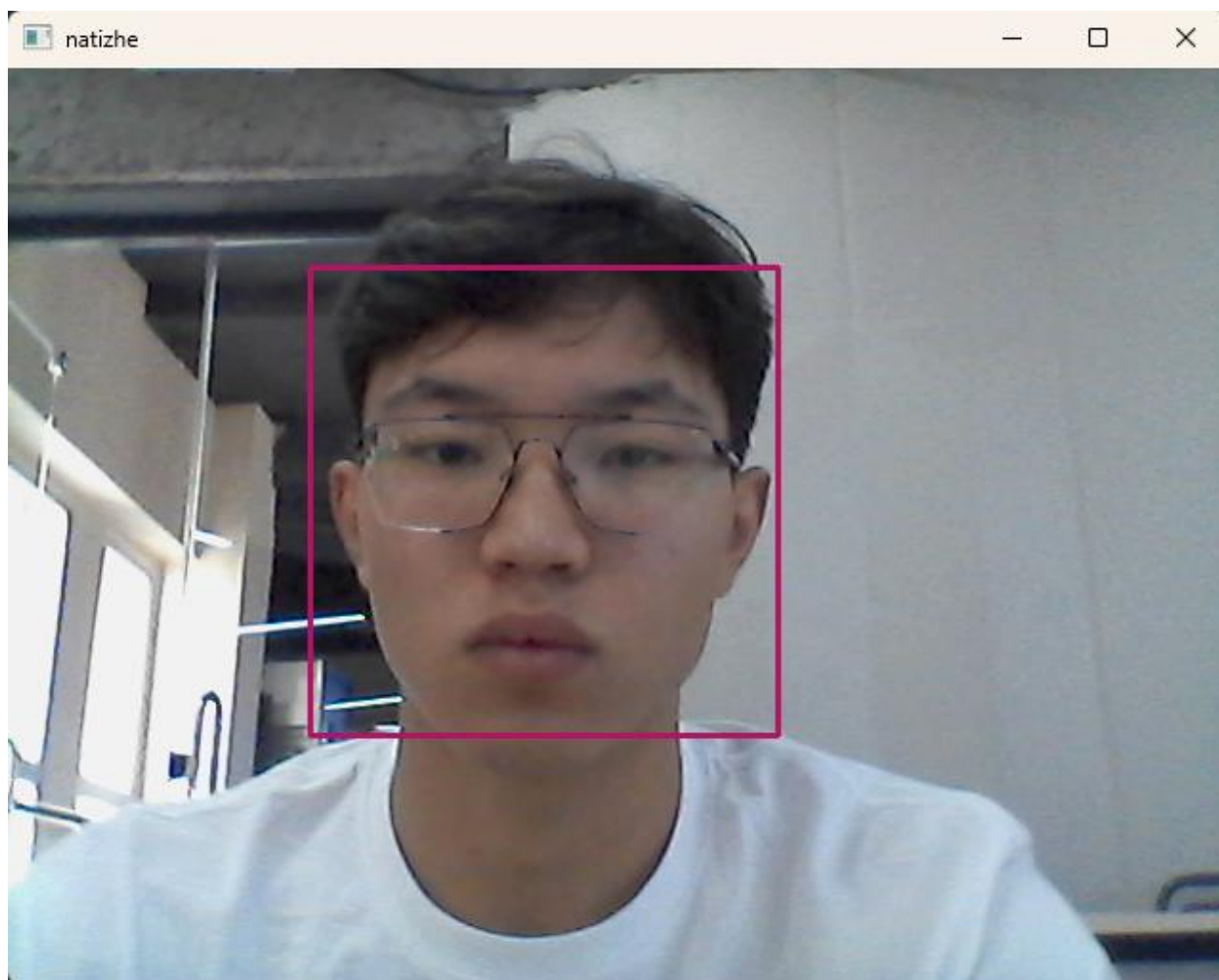
```

1
2 import numpy as np
3 import cv2
4
5 # Камерадағы видеоны алу
6 cap = cv2.VideoCapture(0)
7
8 # Кәрсетілетін бейненің көлемі
9 cap.set( propId: 3, value: 500) # Ені
10 cap.set( propId: 4, value: 600) # Биіктігі
11
12 # Адам бетін анықтауға арналған модель
13 faces = cv2.CascadeClassifier('faces.xml')
14
15 while True:
16     # кадрды оқу
17     success, img = cap.read()
18     if not success:
19         print("Кәте")
20         break
21
22 # Ағымдағы кадрды сұр ренкке келтіреміз
23 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
24
25 # Кадрдағы адамдардың беттерін анықтаймыз
26 results = faces.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=6)
27
28 # Анықталған бетті белгілеу үшін тіктөртбұрыштарды қолдану
29 for (x, y, w, h) in results:
30     cv2.rectangle(img, (x, y), (x + w, y + h), (100, 20, 179), thickness=2)
31
32 # Отображение кадра с распознаванием лиц в окне
33 cv2.imshow( winname: 'natizhe', img)
34
35 # Кадрды аяқтау үшін "g" батырмасын қосамыз
36 key = cv2.waitKey(1)
37 if key == ord('g'):
38     break
39
40 # камераны тоқтатамыз
41 cap.release()
42 cv2.destroyAllWindows()

```

Сурет – 17

Нәтижесінде ноутбуктың камерасына рұқсат аламыз. Және де камера арқылы камераға түскен адамдардың бейнелерін анықтаймыз. Нәтижесі төменде көрсетілген.



Сурет – 18

Хаар Каскады әдісі арқылы ағымдағы видеодан дәлірек ноутбуктың камерасынан OpenCV кітапханасының көмегімен адам бейнесін анықтадым. Бұл бағдарлама адам бейнесін анықтап, оның қозғалысын бақылай алады. OpenCV кітапханасы камераны PyCharm бағдарламасына қосса, Хаар Каскады адам бейнесін анықтауға мүмкіндік береді.

3.2 Беттердегі эмоцияларды тану үшін терең оқыту алгоритмдерін қолдану

Нақты уақыт ағымындағы адам бейнесін анықтайтын бағдарлама іске асқаннан кейін, адам бейнесіндегі эмоцияны анықтайтын бағдарлама жасаймын. Бұл бағдарламаны іске асыру үшін FER кітапханасы қажет болады. FER кітапханасы адам бейнесіндегі эмоцияны анықтауға мүмкіндік береді. FER кітапханасын қосу үшін PyCharm бағдарламасының терминалына мына кодты енгіземіз: `pip install fer`

Кодты жазған соң FER кітапханасы жүктеледі. Бұл кітапхананы қолдану үшін `from fer import FER` кодын жазамыз. Осы код FER кітапханасының функцияларын қолдануға мүмкіндік береді.

```
1 detector = FER() – коды арқылы FER() кітапханасын іске қосу
```

```
2 face_img = img[y:y + h, x:x + w]
```

Коды арқылы кескіннен адам бейнелерін алу, оны `face_img` айнымалысына жинайды.

```
3 emotions = detector.detect_emotions(face_img)
```

`emotions`: бұл `detector` функциясы қайтаратын `detector.detect_emotions(face_img)`. Тізімнің әрбір элементі бір адамға танылған эмоциялар туралы ақпараты бар сөздік болып табылады.

```
4 if emotions:
```

```
    emotion_dict = emotions[0]['emotions']
```

`emotions[0]`: тізімнің бірінші элементі (`face_img`-де тек бір тұлға бар деп есептеңіз).

`emotions[0]['emotions']`: әр түрлі эмоциялардың ықтималдығы бар кірістірілген сөздікті шығарады.

```
5 emotion, score = max(emotion_dict.items(), key=lambda item: item[1])
```

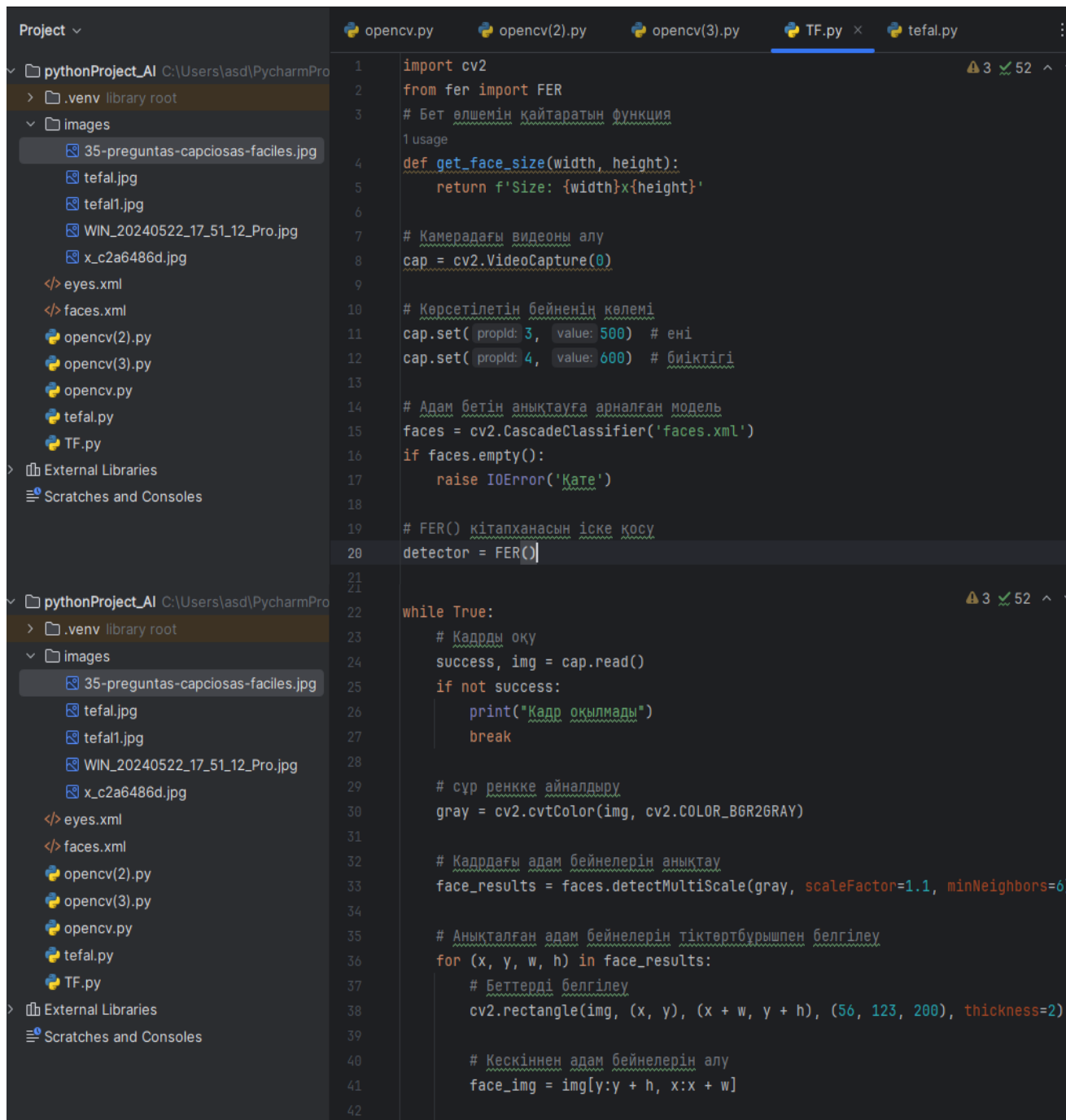
`emotion_dict.items()`: эмоциялар сөздігінен жұптарды (эмоция, ұпай) қайтарады.

`max(..., key=lambda item: item[1])`: максималды `score` мәні бар жұпты табады (ықтималдық). `item[1]` әрбір эмоция үшін ықтималдық мәнін білдіреді. `emotion, score`: тиісінше, ең жақын эмоция және оның ықтималдығы.

```
6 cv2.putText(img, f'{emotion}: {score:.2f}', (x, y - 10),  
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255, 12), 2)
```

`f'{emotion}: {score:.2f}'` - көрсетілетін мәтін. `f`-жол эмоцияның атауын және үтірден кейінгі екі таңбалы ықтималдықты қамтитын жолды пішімдеу үшін қолданылады.

cv2.FONT_HERSHEY_SIMPLEX: мәтін қарпіі. 0.9 масштабы. (36, 255, 12) түсі. 2 қалыңдығы.



```
Project
├── pythonProject_AI C:\Users\asd\PycharmPro
│   ├── .venv library root
│   └── images
│       ├── 35-preguntas-capciosas-faciles.jpg
│       ├── tefal.jpg
│       ├── tefal1.jpg
│       ├── WIN_20240522_17_51_12_Pro.jpg
│       ├── x_c2a6486d.jpg
│       ├── eyes.xml
│       ├── faces.xml
│       ├── opencv(2).py
│       ├── opencv(3).py
│       ├── opencv.py
│       ├── tefal.py
│       └── TF.py
├── External Libraries
└── Scratches and Consoles

pythonProject_AI C:\Users\asd\PycharmPro
├── .venv library root
└── images
    ├── 35-preguntas-capciosas-faciles.jpg
    ├── tefal.jpg
    ├── tefal1.jpg
    ├── WIN_20240522_17_51_12_Pro.jpg
    ├── x_c2a6486d.jpg
    ├── eyes.xml
    ├── faces.xml
    ├── opencv(2).py
    ├── opencv(3).py
    ├── opencv.py
    ├── tefal.py
    └── TF.py

External Libraries
Scratches and Consoles

opencv.py opencv(2).py opencv(3).py TF.py tefal.py
1 import cv2
2 from fer import FER
3 # Бет өлшемін қайтаратын функция
4 usage
5 def get_face_size(width, height):
6     return f'Size: {width}x{height}'
7
8 # Камерадағы видеоны алу
9 cap = cv2.VideoCapture(0)
10
11 # Көрсетілетін бейненің көлемі
12 cap.set( propId: 3, value: 500) # ені
13 cap.set( propId: 4, value: 600) # биіктігі
14
15 # Адам бетін анықтауға арналған модель
16 faces = cv2.CascadeClassifier('faces.xml')
17 if faces.empty():
18     raise IOError('Қате')
19
20 # FER() кітапханасын іске қосу
21 detector = FER()
22
23 while True:
24     # Кадрды оқу
25     success, img = cap.read()
26     if not success:
27         print("Кадр оқылмады")
28         break
29
30     # сұр ренкке айналдыру
31     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
32
33     # Кадрдағы адам бейнелерін анықтау
34     face_results = faces.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=6)
35
36     # Анықталған адам бейнелерін тіктөртбұрышпен белгілеу
37     for (x, y, w, h) in face_results:
38         # Беттерді белгілеу
39         cv2.rectangle(img, (x, y), (x + w, y + h), (56, 123, 200), thickness=2)
40
41         # Кескіннен адам бейнелерін алу
42         face_img = img[y:y + h, x:x + w]
```

Сурет – 19

```
> .venv library root
  images
  35-preguntas-capciosas-faciles.jpg
  tefal.jpg
  tefal1.jpg
  WIN_20240522_17_51_12_Pro.jpg
  x_c2a6486d.jpg
  eyes.xml
  faces.xml
  opencv(2).py
  opencv(3).py
  opencv.py
  tefal.py
  TF.py
  External Libraries
  Scratches and Consoles

43 # Эмоцияны анықтау
44 emotions = detector.detect_emotions(face_img)
45 if emotions:
46     # Ең дәл болатын эмоцияны анықтау
47     emotion_dict = emotions[0]['emotions']
48     emotion, score = max(emotion_dict.items(), key=lambda item: item[1])
49     cv2.putText(img, text=f'{emotion}: {score:.2f}', org=(x, y - 10), c
50
51 # Адам бейнесінің өлшемін есептеу
52 face_size_text = get_face_size(w, h)
53 cv2.putText(img, face_size_text, org=(x, y + h + 20), cv2.FONT_HERSHEY_
54
55 # Кадрды көрсету
56 cv2.imshow('winname: 'result', img)
57
58 # Бағдарламадан шығу
59 if cv2.waitKey(1) & 0xFF == ord('g'):
60     break
61
62 # Камерадан шығу
63 cap.release()
```

Сурет – 20



Сурет – 21

FER кітапханасы (Facial Expression Recognition) жеті негізгі эмоцияны тани алады. Бұл эмоцияларға мыналар жатады:

- Angry (ашулы)
- Disgust (жиіркеніш)
- Fear (қорқыныш)
- Happy (бақыт)
- Sad (мұңды)
- Surprised (таңдану)
- Neutral (бейтарап мимика)

4 Тәжірибелер мен нәтижелер

Жоғарыда сипатталған код пен FER кітапханасының жұмысы негізінде нәтижелерді келесідей қорытындылауға болады:

Кодтың мақсаты:

Камераны пайдаланып бет-әлпетті анықтауға және эмоцияны нақты уақытта тануға арналған жүйені әзірлеу.

Бейне ағынында анықталған беттер мен танылған эмоциялардың өлшемдерін көрсету.

- Бағдарлама жұмысының негізгі кезеңдері:
- Камерадан бейне ағынын инициализациялау.
- Бейне ағыны терезесінің өлшемін орнату.
- Бетті анықтау моделін жүктеу (Хаар каскады).
- Эмоция детекторын инициализациялау (FER кітапханасы).
- Нақты уақыттағы кадрларды оқу және өңдеу:
- Кадрларды сұр реңктерге түрлендіру.
- Кадрдағы адамдарды анықтау.
- Табылған беттердің айналасындағы тіктөртбұрыштарды көрсету.
- Беттерді шығару және эмоцияларды тану.
- Кадрда беттердің өлшемін және танылған эмоцияларды көрсету.
- "g" батырмасын басқан кезде бағдарламаны аяқтау.

Негізгі нәтижелер:

- Бағдарлама нақты уақыттағы бейне ағынындағы беттерді сәтті анықтайды.
- Жүйе жеті негізгі эмоцияны таниды (ашу, жиіркеніш, қорқыныш, бақыт, мұң, таңдану және бейтарап көрініс).
- Бейне ағындағы анықталған беттердің өлшемдері мен танылған эмоцияларды көрсетеді.
- "g" батырмасы арқылы бағдарламаны аяқтау мүмкіндігі.

Эмоцияны нәтижеге айналдыру:

- FER кітапханасы жеті эмоцияның әрқайсысының ықтималдығын қайтарады. 0-1-ге дейін.
- Бағдарлама эмоцияны және оның бағасын көрсетеді

Осылайша, бұл код нақты уақыттағы бейне ағынын талдау үшін машиналық көру және эмоцияны тануды сәтті қолданылуын көрсетеді.

5 Қолдану және перспективалар

Қауіпсіздік және бақылау:

Бейнебақылау жүйелерінде күдікті мінез-құлық пен ықтимал қауіпті адамдарды анықтау үшін бет пен эмоцияны тануда қолдана алады. Мысалы, жүйе адамдар көп жиналатын жерлерде агрессиясы анық бар адамның болуын білдіре алады.

Маркетингтік және тұтынушылық зерттеулерде:

Сауда орталықтары мен дүкендердегі эмоцияларды тану клиенттердің әртүрлі тауарлар мен маркетингтік материалдарға қалай реакция беретінін түсінуге көмектеседі. Бұл тауарлар мен жарнамалық материалдарды орналастыру стратегиясын жақсартуға мүмкіндік береді.

Медицина және психологияда:

Бет пен эмоцияны тану технологиясын психикалық ауытқулары бар науқастардың жағдайын диагностикалауға және бақылау үшін пайдалануға болады. Бұл эмоционалды жағдайды бағалауға және терапевтік әрекеттерді реттеуге көмектесуі мүмкін.

Білім беру және қашықтықтан оқыту кезінде:

Білім беру мекемелерінде эмоцияны тану жүйелерін оқушылардың сабаққа қатысуын және олардың эмоционалды жағдайын бағалау үшін пайдалануға болады деп ойлаймын. Бұл мұғалімдерге тиімділікті арттыруға және де оқыту әдістерін оқушыларға қарай бейімдеуге көмектесе алады.

Бұл жұмыста машиналық оқыту әдістерін қолдана отырып, нақты уақыт режимінде бет пен эмоцияны тануға арналған жүйе жасалды және енгізілді. Графикалық бейнелерді анықтау, кескінді, видеороликті қабылдау OpenCV кітапханасының көмегімен іске асты. Ал жүйенің негізі бетті анықтауға арналған Хаар каскадтық алгоритмі және эмоцияны тануға арналған FER кітапханасы болды. Ал код PyCharm бағдарламасында жүзеге асты.

Бет пен эмоцияны тану технологиясын қолдану қауіпсіздіктен ойын-сауыққа дейін әртүрлі салаларда кең мүмкіндіктерге ие бола алады. Бұл технологиялардың даму перспективалары дәлдік пен жылдамдықты және ыңғайлылықты айтарлықтай жақсартуға алады. Дегенмен, осы технологияларды қауіпсіз және жауапты пайдалануды қамтамасыз ету үшін этикалық және құқықтық аспектілерді ескеру маңызды болуы керек.

Әзірленген жүйе өзінің тиімділігі мен нақты жағдайларда қолданылуын көрсете алды. Алайда, алгоритмдерді одан әрі дамыту және жетілдіру, сондай-ақ жеке деректерді қорғау сияқты этикалық мәселелерді ескеру осы технологияларды күнделікті өмірге сәтті енгізудің негізгі аспектілері болып табылады.

Әлі де машиналық көрудің болашақта үлкен перспективалары бар. Машиналық көру күнделікті адам өмірін жеңілдетеді алады. Болашақ алыс емес, болашақ орын алды.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.
- 2 Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, pp. I-I.
- 3 Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. IEEE Transactions on Affective Computing, 10(1), 18-31.
- 4 Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- 5 Zhang, Y., & Chen, D. (2018). Efficient Emotion Recognition from Face Images Using Deep Convolutional Neural Networks. IEEE Access, 6, 41398-41406.
- 6 OpenCV Documentation. (n.d.). OpenCV Library Documentation.
- 7 Barsoum, E., Zhang, C., Ferrer, C. C., & Zhang, Z. (2016). Training deep networks for facial expression recognition with crowd-sourced label distribution. Proceedings of the 18th ACM International Conference on Multimodal Interaction.
- 8 FER Documentation. (n.d.). FER Library Documentation.
- 9 Хаар Каскадының коды алынған сайт:
<https://github.com/opencv/opencv/tree/master/data/haarcascades>.