

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Закарина Н.Қ.

Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша
деректерді талдау

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы


Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
КҚФжС кафедра меңгерушісі,
канд.тех.наук, ассистент
профессор
 Н.А.Сейлова
« 24 » мамыр 2019 ж.


ДИПЛОМДЫҚ ЖҰМЫС


Тақырыбы «Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау»

5B070300 – «Ақпараттық жүйелер» мамандығы бойынша

Орындаған

Закарина Н.К.

Пікір беруші:
PhD докторы, аға ғылыми оқытушы
 А.Х. Қозбақова

Ғылыми жетекші :
Техн. ғыл. магистр. лектор
 А.Н. Дуйсенбаева

« 6 » мамыр 2019ж.

« 8 » мамыр 2019ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 – «Ақпараттық жүйелер» мамандығы

БЕКІТЕМІН

КҚАӨЖС кафедра меңгерушісі,
тех.ғыл. канд, ассистент
профессор

 Н.А.Сейлова
« 4 » мамыр 2019 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Закарина Нұрзада Қожайқызы

Тақырыбы: Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау

Университет Ректоры 2018 жылғы «16» қазан № 1162 -бет бұйрығымен бекітілген

Аяқталған жұмысты тапсыру мерзімі: 2019 « 4 » мамыр

Дипломдық жұмыстың бастапқы берілістері: диплом алдындағы практикалық жұмыс қорытындысы, тақырып бойынша әдебиеттерге шолу нәтижелері, теориялық мәліметтердің жиыны

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) қойылған мәселенің қазіргі жағдайын пайымдау

ә) ақпараттық қамтаманы құру

б) программалық қамтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар



Сызба материалдар: 15 слайдпен көрсетілген

Ұсынылатын негізгі әдебиет: 14 атау

Дипломдық жобаны дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мәселенің қазіргі жағдайына шолу және оны талдау	11.01.2019 - 07.03.2019.	
Ақпараттық қамтаманы құру	07.02.2019-12.03.2019.	
Программалық қамтаманы құру	12.03.2019-28.04.2019.	

Дипломдық жұмысының бөлімдерінің кеңесшілері мен норма бақылаушыларының аяқталған жобаға қойған
қолтаңбалары

Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Норма бақылаушы	А.А.Кабдуллин, ассистент	13.05.19	
Программалық қамтама	М.Б. Бауыржан, тьютор, тех.ғыл магистрі	8.05.19	

Ғылыми жетекші 

Дуйсенбаева А.Н.

Тапсырманы орындауға алған білім алушы 

Закарина Н.К.

Күні

« 08 » маусым 2019

Ғылыми жетекшінің пікірі

Дипломдық жұмысқа
Закарина Нұрзада
5В070300- Ақпараттық жүйелер

Тақырыбы: Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау

Закарина Нұрзаданың алдына машина оқыту алгоритмдерін қолдану арқылы бетті тану жұмысын жасау мақсаты қойылды.

Закарина Нұрзада – С# ортасында машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау, бетті тану жүйесі жасау қарастырылады.

Студент дипломдық жұмысты толығымен өз бетінше орындады. Қосымшаның артықшылықтары қазіргі заманғы қауіпсіздіктің өте жоғарылығы, қол жетімді интерфейс, функционалдылық болып табылады.

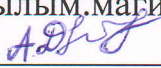
Дипломдық жобаны орындау барысында Закарина Нұрзада алдына қойылған мақсатты орындап, машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау, бетті тану ұйымдастырды.

Студент Закарина Нұрзада дипломдық жұмысты орындау барысында оқу кезінде алған теориялық және тәжірбиелік білімін, техникалық әдебиеттермен жұмыс жасап, қолдана білетіндігін көрсетті.

Дипломдық жұмыс жақсы деңгейде орындалды, студенттің дипломдық жұмысы қорғауға жіберіледі.

Ғылыми жетекші:

Техн.ғылым.магистр.лектор

 Дуйсенбаева А.Н.

« 8 » мамыр 2019ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Дипломдық жұмысқа СЫН-ПІКІР

Білім алушы: Закарина Нұрзада

Тақырыбы: Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау

Дипломдық жұмыстың тақырыбы қазіргі таңда өзекті, себебі күнделікті машина оқытуды үйренудің үлкен ауқымының арқасында біздің өмірімізде өсіп келе жатқан орынды алады. Кептелістерді талдаудан бастап, өздігінен жүретін автокөліктермен аяқталатындықтан, өздігінен жүретін машиналарға көбірек тапсырмалар қойылады.

Машина оқыту жасанды интеллекттің саласы болып саналады, оның негізгі идеясы компьютерде алдын ала жазылған алгоритмді ғана емес, сонымен қатар мәселені қалай шешуге болатынын біледі.

Кіріспеде мәселенің маңыздылығы, жобаның мақсаты және қолданылған құралдар туралы мәліметтер келтірілген. Бірінші бөлімде Машина оқытудың тәсілдері, машина оқыту міндеттері қарастырылған. Екінші бөлімде қолданылған кітапханалар маңыздылығы, Microsoft visual studio интегралданған бағдарламалау ортасын қолдану, машина оқыту әдістері көрсетілген. Үшінші бөлімде машина оқыту алгоритмдерін қолданып бетті тану программасын жасау, жалпы бағдарламалар сипатталған. Дипломдық жоба жоғарғы ғылыми техникалық деңгейде орындалған.

Дипломдық жұмыста елеулі кемшіліктер анықталмады. Белгілі бір семантикалық және стилистикалық дәлсіздіктер сияқты шағын кемшіліктер, сондай-ақ әртүрлі әдебиетке деген сілтемелер аз. Алайда, бұл кемшіліктер орындалатын жұмыстардың жалпы деңгейін төмендетпейді.

Дипломдық жұмыс Ақпараттық жүйелер мамандығының бітіру жұмыстарына қойылатын талаптарға сәйкес.

Қорытындылай келе, Закарина Нұрзаданың дипломдық жұмысына «90%» деген баға беруге және оның орындаушысына 5B070300 – «Ақпараттық жүйелер» мамандығы бойынша бакалавр біліктілігін беруге болады деп есептеймін.

Сын-пікір беруші:

PhD докторы, аға ғылыми оқытушы  Қозбақова А.Х



Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Закарина Нұрзада

Название: Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау

Координатор: Асемгуль Дуйсенбаева

Коэффициент подобия 1:0

Коэффициент подобия 2:0

Тревога:1

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....
.....
.....
.....
.....

Дата 14.05.19

Подпись заведующего кафедрой

начальника структурного подразделения




Окончательное решение в отношении допуска к защите, включая обоснование:

Вопросе не берется

Дата 14.05.1972

Подпись заведующего кафедрой /



начальника структурного подразделения



Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Закарина Нұрзада

Название: Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау

Координатор: Асемгуль Дуйсенбаева

Коэффициент подобия 1:0

Коэффициент подобия 2:0

Тревога:1

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....
.....
.....
.....
.....

14.05.2013г.

Дата

.....
.....
.....

Подпись Научного руководителя

Краткий отчет



Университет:	Satbayev University
Название:	Машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау
Автор:	Закарина Нұрзада
Координатор:	Асемгуль Дуйсенбаева
Дата отчета:	2019-05-03 11:32:43
Коэффициент подобия № 1:	0,0%
Коэффициент подобия № 2:	0,0%
Длина фразы для коэффициента подобия № 2:	25
Количество слов:	3 574
Число знаков:	29 272
Адреса пропущенные при проверке:	
Количество завершенных проверок:	21



К вашему сведению, некоторые слова в этом документе содержат буквы из других алфавитов. Возможно - это попытка скрыть позаимствованный текст. Документ был проверен путем замещения этих букв латинским эквивалентом. Пожалуйста, уделите особое внимание этим частям отчета. Они выделены соответственно.

Количество выделенных слов 1

>>

Документы, в которых найдено подобные фрагменты: из RefBooks



Не обнаружено каких-либо

заимствований

>>

Документы, содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

>>

Документы, содержащие подобные фрагменты: Из внешних баз данных

Не обнаружено каких-либо заимствований



Документы,содержащие подобные фрагменты: Из интернета

Не обнаружено каких-либо заимствований

АҢДАТПА

Дипломдық жобаның тақырыбы: "машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау".

Машина оқыту - бұл оқуға қабілетті алгоритмдерді құрастыру әдістерін зерттейтін жасанды интеллекттің кең тарауы. Оқудың екі түрі бар:

- прецедент
- индуктивті

Дефективті оқыту біліктілік білімін ресімдеуді және оларды компьютерге білім базасы түрінде беруді қамтиды. Үздіксіз оқыту әдетте сарапшы жүйелердің саласына жатқызылады, сондықтан машықтану және прецеденттерді оқыту терминдері синоним ретінде қарастырылуы мүмкін. Классикалық статистикалық тәсілдерге балама ретінде индуктивті оқытудың көптеген әдістері жасалды.

Дипломдық жобаның мақсаты: «машина оқыту алгоритмдерін қолдану арқылы өмір саласы бойынша деректерді талдау» байланысты бағдарлама құру.

Қойылған мақсатқа жету үшін келесі міндеттер орындалуы тиіс:

- машина оқытудың негізгі ұғымын қарастыру;
- машина оқыту әдістерін зерттеу;
- машина оқыту алгоритмдерін қолданып бағдарламаны

қамтамасыздандыру.

Осы мақсатқа сәйкес машина оқыту алгоритмдердің негізінде бағдарлама жасалуда.

Зерттеу объектісі. С# тілінде жазылған бағдарлама.

Мәселені талдау дәрежесі. Дипломдық жобаның тақырыбын зерттеу барысында отандық және шетелдік ғылыми еңбектері мен оқулықтар, электронды басылымдардағы ғылыми мақалалар қолданылды.

Зерттеу тәжірибесі: Салыстырмалық, аналитикалық, логикалық, құрылымдық жүйе, функционалдық жүйе, жалпылау тәжірибесі.

АННОТАЦИЯ

Тема дипломного проекта: «Анализ данных по жизненной сфере с использованием алгоритмов машинного обучения».

Машинное обучение представляет собой широкий спектр искусственного интеллекта, в котором исследуются методы обучения алгоритмов. Есть два типа чтения:

- прецедент;
- индуктивный;

Дефектное обучение включает в себя формализацию квалификаций и их передачу на компьютер в качестве базы знаний. Непрерывное обучение обычно относится к области экспертных систем, поэтому условия обучения и прецедентное обучение могут рассматриваться как синонимы. Многие методы индуктивного обучения были разработаны в качестве альтернативы классическим статистическим подходам.

Целью дипломного проекта является создание программы, связанной с «Анализом данных о жизненной сфере с использованием алгоритмов обучения».

Для достижения этой цели необходимо решить следующие задачи:

- рассмотреть основную концепцию машиностроения;
- изучение методов машинного обучения;
- разработка программного обеспечения с использованием алгоритмов машинного обучения.

На основе цели разрабатывается программа на с использованием алгоритмов машинного обучения.

Объект исследования. Программное обеспечение написанная на языке C#.

Степень анализа проблемы. При изучении темы дипломного проекта, отечественных и зарубежных научных работ и учебников использовались научные статьи в электронных изданиях.

Опыт исследования: относительная, аналитическая, логическая, структурная система, функциональная система, опыт обобщения.

ANNOTATION

Theme of the graduation project: "Analysis of data on the life sphere using machine learning algorithms."

Machine learning is a wide range of artificial intelligence, which explores the methods of learning algorithms. There are two types of reading:

- precedent;
- inductive;

Defective learning includes the formalization of qualifications and their transfer to a computer as a knowledge base. Continuous learning is usually in the area of expert systems, so learning conditions and case study can be considered synonymous. Many methods of inductive learning have been developed as an alternative to classical statistical approaches.

The aim of the graduation project is to create a program related to the "Analysis of data on the life sphere using learning algorithms."

To achieve this goal it is necessary to solve the following tasks:

- consider the basic concept of engineering;
- the study of machine learning methods;
- software development using machine learning algorithms.

Based on the goal, a program is developed on using machine learning algorithms.

Object of study. Software written in C #.

The degree of analysis of the problem. When studying the topic of the graduation project, domestic and foreign scientific works and textbooks, scientific articles were used in electronic editions.

Research experience: relative, analytical, logical, structural system, functional system, generalization experience.

МАЗМҰНЫ

КІРІСПЕ	9
1 ПӘНДІК САЛАНЫ ЗЕРТТЕУ	10
1.1 Машина оқыту түсінігіне жалпы шолу	10
1.2 Машина оқыту міндеттерінің сыныптары	11
1.3 Машина оқытудың тәсілдері	12
2 МАШИНА ОҚЫТУ БАҒДАРЛАМАСЫН ҚҰРУДЫҢ КОМПОНЕНТТЕРІ	16
2.1 OpenCV кітапханасын қолдану тиімділігі	16
2.2 Microsoft visual studio интегралданған бағдарламалау ортасын қолдану	17
2.3 Машина оқыту әдістерін қолдану	18
3 ПРОГРАММАЛЫҚ ҚАМТАМАНЫ ҚҰРУ	22
3.1 Программалау тілін таңдауды негіздеу	22
3.2 Программаның баяндалуы	22
3.3 Функционалдық тағайындалуы	23
3.4 Қажетті техникалық жабдықтар	26
3.5 Шақыру және жүктеу	26
3.6 Кіріс мәліметтер	26
3.7 Шығыс мәліметтер	26
ҚОРЫТЫНДЫ	27
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	28
А Қосымшасы	29

КІРІСПЕ

Жасанды интеллект - біздің уақыттағы инновациялық дамудың жетекші бағыттарының бірі. Кез-келген интеллекттің негізі - бұл оқыту. «Машина оқыту» технологиясы көптеген қызмет бағыттары үшін белсенді құрал болып табылады.

Learning (ML) - адамның араласусыз машинаны үйрену үрдісі. Жалпы алғанда, бұл тұтас алгоритмдер және шешімдер қабылдау жүйесі. Мәтінді өңдеу, визуалды бейнені тану, үлкен деректер массивтерін өңдеу Microsoft, Google, Apple [2] сияқты алыптар әзірлеген ML технологиялары болып табылады.

Жоғарыда айтылғандай, ML технологиясы IT-индустриясында белсенді дамудың мәні болып табылады. Бірақ оны пайдалану нақты емес. Өмірлік қызметі барысында адамзат өз қызметін әртүрлі салаларда белсенді жинақтайды. Бұл ақпаратты өңдеу арқылы кәсіпкерлер тұтынушылық әрекеттердің болжамын алады.

Мәселен, ML технологиясы (Data Mining-мен бірге) онлайн-дүкендердің, сақтандыру компанияларының иелері арасында кең таралған. [2] ML сандар жиынтығымен жұмыс істейтін алгоритмдер, математикалық құралдар болып келеді. Бұл қызмет түрі үшін математикалық аппаратты (Matlab, SPSS, SAS) енгізетін компьютер мен бағдарламалық қамтамасыз ету керек.

Жасанды интеллект және машина оқытуды әркім әртүрлі түсінеді. Тұтастай алғанда, адамға қарапайым тапсырмаларды шешетін бағдарламалар жасанды интеллект деп атауға болады.

Үлкен мөлшерде деректерді өңдеу үшін бағдарламаны үйренеміз. Нәтижесінде, карталармен ойын ойнау, бір уақытта бірнеше маршруттарды жоспарлау, бүкіл ел бойынша энергияны тұтынуды оңтайландыру және қажетті ақпаратты Google-да іздеу сияқты күрделі міндеттерді шешеді.

Машина оқыту - бұл 18 ғасырда статистика теориясының бір бөлігі ретінде басталған пән. Бірақ бүгінде бұл ешқашан болған емес.

Өз қызметтерімізді әрдайым жетілдіріп отырамыз, сондықтан оларды нақты болжамдар жасауға және адамға қиындық тудырмайтын мәселелерді орындауға үйретуіміз керек. Интернеттегі машина оқытуда қажетті деректердің іс жүзінде сарқылмайтын көзі болып табылады.

Соңғы бірнеше жылда әр түрлі IT-компаниялар, соның ішінде Google, машинада оқудың керемет және күрделі бағдарламаларын құруға көмектесті.

Дегенмен, олар әлі де жетілдірілмеген, компьютерлер сияқты, сондықтан біз әлі күнге дейін олардың жұмысына көптеген өзгерістер енгізуге тура келеді. Бұл параллельді бағдарламаны іске асыру үшін зор есептеу қуатын және арнайы ресурстарды қажет етеді. Бақытымызға орай, бағдарламалық жасақтама мен аппараттық құралдардың соңғы жетістіктері бізге көмектеседі.

1 Пәндік саланы зерттеу

1.1 Машина оқыту түсінігіне жалпы шолу

Күнделікті машина оқытуды үйренудің үлкен ауқымының арқасында біздің өмірімізде өсіп келе жатқан орынды алады. Кештелістерді талдаудан бастап, өздігінен жүретін автокөліктермен аяқталатындықтан, өздігінен жүретін машиналарға көбірек тапсырмалар қойылады.

Кейде тіпті машиналық оқыту әдістеріне негізделген кейбір қосымшалардың қалай жұмыс істейтіні туралы қате пікірлер бар.

Машина оқыту жасанды интеллекттің саласы болып саналады, оның негізгі идеясы компьютерде алдын ала жазылған алгоритмді ғана емес, сонымен қатар мәселені қалай шешуге болатынын біледі.

Кез-келген жұмыс істейтін машина оқыту технологиясы қол жетімділіктің үш деңгейінің біріне шартты түрде жатқызылуы мүмкін. Бірінші деңгей - тек Google немесе IBM деңгейіндегі түрлі технологиялық алыптар үшін қол жетімді. Екінші деңгей - білімнің белгілі бір деңгейін пайдаланатын студент. Үшінші деңгей - бұл тіпті нағыз әжеймен күресуге қабілетті.

Машина оқыту қазіргі уақытта екінші және үшінші деңгейлердің қиылысында, осы технология көмегімен әлем өзгеруінің жылдамдығы күннен-күнге артып келеді.

Машиналарды оқытудың практикалық бағыттары.

Машина оқытудың мақсаты - адам қызметінің түрлі салаларында күрделі кәсіби мәселелерді шешуді ішінара немесе толық автоматтандыру.

Машина оқытуда кең ауқымды қосымшалар бар:

- сөйлеуді тану;
- қимылдарды тану;
- қолжазбаны тану;
- үлгіні тану;
- техникалық диагностика;
- медициналық диагностика;
- уақыт серияларын болжау;
- биоинформатика;
- алаяқтықты анықтау;
- спамды анықтау;
- құжаттарды санаттау;
- қордың техникалық сараптамасы;
- қаржылық қадағалау;
- несие ұпайлары;
- клиенттерге қызмет көрсетуді болжау;
- хемоинформатика;
- ақпаратты іздеуде оқу рейтингі.

Машиналық қосымшаларды қолдану ауқымы үнемі кеңейуде. Кең таралған ақпараттандыру ғылым, индустрия, бизнес, көлік және денсаулық сақтау саласындағы көптеген деректер жинақталуына әкеледі. Бұл жағдайда туындайтын болжау, басқару және шешім қабылдау проблемалары көбінесе прецеденттерден үйренуге азаяды. Бұрын мұндай деректер болмаған кезде, бұл тапсырмалар мүлдем басқа әдістермен шешілмеген немесе мүлде басқа әдістермен шешілген.

Машина оқыту математикалық статистиканың, оңтайландыру әдістері мен классикалық математикалық пәндердің түйіскен жерінде орналасқан, бірақ есептеу тиімділігі мен қайта оқыту проблемаларымен байланысты өзіндік ерекшелігі бар. Индуктивті оқытудың көптеген әдістері классикалық статистикалық тәсілге балама ретінде әзірленген. Көптеген әдістер ақпаратты алумен және деректерді интеллектуалды талдаумен тығыз байланысты (Data Mining).

Машина оқытудың теориялық бөлімдері жеке бағытқа, есептеуіш оқыту теориясы (Computational Learning Theory, COLT) біріктірілген.

Машина оқыту-математикалық ғана емес, практикалық, инженерлік пән. Таза теория, әдетте, практикада қолданылатын әдістер мен алгоритмдерге бірден әкелмейді. Оларды жақсы жұмыс істеуге мәжбүр ету үшін теорияда жасалған болжамдардың нақты есептер шарттарына сәйкес еместігін өтейтін қосымша эвристиканы ойлап табуға тура келеді. Машиналық оқытудағы бірде-бір зерттеу әдістің практикалық жұмысқа қабілеттілігін растайтын модельдік немесе нақты деректерде экспериментсіз өтпейді.

Машина оқытудың мақсаты кіріс мәліметтері бойынша нәтижені болжау. Кіріс деректері әр түрлі болған сайын, машинада заңдылықтарды табу оңайырақ және дәлірек нәтиже.

Машина оқыту (ML) жасанды интеллект әдістерінің класы болып табылады, олардың тән ерекшелігі мәселені тікелей шешу емес, сонымен қатар көптеген ұқсас міндеттерге шешім қабылдау процесінде оқыту. Осындай әдістерді құру үшін математикалық статистика құралдары, сандық әдістер, оңтайландыру әдістері, ықтималдықтар теориясы, графикалық теория және деректермен жұмыс істеудің әр түрлі әдістері цифрлық түрде қолданылады.

Оқудың екі түрі бар:

Іс-тәжірибені немесе индуктивті оқытуды деректердегі эмпирикалық үлгілерді анықтауға негізделеді. Үздіксіз оқыту - сараптамалық білімді ресімдеуді және оларды компьютерге білім базасы ретінде беруді қамтиды. Үздіксіз оқыту әдетте сарапшы жүйелердің саласына жатқызылады, сондықтан машыктану және прецеденттерді оқыту терминдері синоним ретінде қарастырылуы мүмкін.

Классикалық статистикалық тәсілдерге балама ретінде индуктивті оқытудың көптеген әдістері жасалды. Көптеген әдістер ақпарат алу (ақпарат алу), деректерді өндіруге тығыз байланысты.

1.2 Машина оқыту міндеттерінің сыныптары

Регрессия міндеті: нақты жауапты болжау үшін түрлі белгілер негізінде. Басқаша айтқанда, жауап 1, 5, 23,575 немесе кез-келген басқа нақты сан болуы мүмкін, мысалы, пәтердің құнын көрсете алады. Мысалдар: алты айдағы қордың құнын болжау, келесі айда дүкен пайдасын болжау, соқыр тестілеу кезінде шараптың сапасын болжау.

Жіктелу міндеті: категориялық жауапты болжау үшін түрлі белгілер негізінде. Басқаша айтқанда, пациенттің қатерлі ісік бар-жоғын немесе хаттың спам екенін анықтаған жағдайда, осындай тапсырмада соңғы жауаптар саны бар. Мысалдар: мәтінді тану қолжазбалары, адам немесе мысық фотосуретте болғанын анықтау.

Кластерлеу міндеті: деректерді ұқсас санаттарға бөлу. Мысалдар: ұялы байланыс операторының клиенттерін төлем қабілеттілігіне сәйкес бөлу, ғарыш объектілерін ұқсас (галактикалар, планеталар, жұлдыздар және т.б.) бөлу.

Өлшемді азайту міндеті: біздің деректерді N белгілерімен емес сипаттауға үйрену керек, бірақ одан да аз санымен (әдетте 2-3 визуализация үшін). Мысал ретінде, визуализация қажеттілігіне қоса, деректерді қысуыңызға болады.

Аномалияларды анықтау міндеті: белгілер негізінде, ауытқуларды «ауытқулардан» ажыратуды үйренеді. Меніңше, бұл тапсырма жіктеу тапсырмасынан өзгеше емес. Бірақ аномалияларды анықтау ерекшелігі, бізде модельді үйретудегі ауытқулардың өте аз немесе жоқ екендігі, сондықтан біз бұл мәселені жіктеу мәселесі ретінде шеше алмаймыз. Мысал: банктік карточкалар бойынша алаяқтық операцияларды анықтау.

Іс-тәжірибелік тапсырмалардың жалпы тұжырымы

Көптеген нысандар (жағдайлар) және көптеген мүмкін жауаптар (жауаптар, реакциялар) бар. Жауаптар мен нысандар арасында қандай да бір корреляция бар, бірақ ол белгісіз. Тек прецеденттердің түпкілікті жиынтығы - оқыту нысаны деп аталатын «нысан, жауап» жұптары белгілі. Осы деректер негізінде, көзге көрінбейтін тәуелділікті қалпына келтіру қажет, яғни кез келген ықтимал кіріс нысаны үшін негізді дәл жіктеу жауап бере алатын алгоритм салу. Бұл тәуелділік міндетті түрде аналитикалық түрде көрсетілмейді және мұнда нейрондық желілер эмпирикалық қалыптасқан шешімнің принципін іске асырады. Маңызды ерекшелігі - дайындалған жүйенің жинақталу қабілеттілігі, яғни, қолданыстағы жаттығу жиынынан тыс деректерге барабар жауап беру. Жауаптардың дұрыстығын өлшеу үшін бағалаудың сапа функциясы енгізіледі.

Бұл тұжырым функциялардың жуықтауының классикалық мәселелерін қорыту болып табылады. Аппроксимацияның классикалық мәселелерінде объектілер нақты сандар немесе векторлар болып табылады. Нақты қосымшаларда нысандар туралы деректер енгізу толық емес, дәл емес, сандық емес, гетерогенді болуы мүмкін. Бұл мүмкіндіктер машинада оқыту әдістерінің алуан түріне әкеледі.

1.3 Машина оқытудың тәсілдері

Бір жағынан, машина оқыту бөлімі нейрондық желілерді ғылымды бөлу әдістеріне және олардың архитектурасының топологияларының түрлеріне бөлу арқылы қалыптасты, екінші жағынан математикалық статистика әдістерін меңгерді. Машиналық оқытудың келесі әдістері нейрондық желілерді қолдану жағдайында пайда болады, дегенмен, оқу үлгісі ұғымын қолданатын басқа әдістер бар - мысалы, байқалған статистиканың жалпыланған дисперсиясы мен ковариациясымен жұмыс істейтін дискриминант анализі немесе Байес классификаторлары. Перцепрон және көп қабатты перцепрон сияқты (сондай-ақ оларды модификациялау) сияқты нейрондық желілердің негізгі түрлері оқытушымен немесе оқытушысымен, күшейтумен және өзін-өзі ұйымдастыруымен оқытылуы мүмкін. Бірақ кейбір нейрондық желілер мен көптеген статистикалық әдістерді оқытудың тек бір жолына ғана жатқызуға болады. Сондықтан, оқыту әдісіне байланысты машиналық оқыту әдістерін жіктеу қажет болса, нейрондық желілерді белгілі бір түрмен жіктеу қате болар еді, нейрондық желілердің оқу алгоритмдерін елестету дұрысырақ болар еді.

Мұғаліммен сабақ беру - әр жағдайда пайдалану жағдайында жұп «жағдай, қажетті шешім» орнатылады:

- жасанды нейрондық желі;
- терең оқыту;
- қатені түзету әдісі;
- артқы таралу әдісі қатесі;
- векторлық машина қолдауы;

Мұғалімсіз оқыту - әр жағдайда тек «жағдай» орнатылады, объектілерді параллельді ұқсастығы немесе деректер өлшемін төмендету туралы деректерді пайдалана отырып, кластерлерге топтарды топтастыру қажет:

- альфа күшейту жүйесі
- гамма күшейту жүйесі
- көршілес көршілес әдісі

Арматурамен жаттығу - әрбір жағдай үшін «жағдай, шешім» жұбы бар:
Генетикалық алгоритм.

Белсенді оқыту оқытудың алгоритмі келесі жағдайды өз бетімен анықтауға мүмкіндік беретіндіктен ерекшеленеді, онда дұрыс жауап болады:

Мұғалімнің ішінара қатысуымен сабақ беру (сабақты меңгеруді үйрену) - прецеденттердің бір бөлігі үшін «жағдай, қажетті шешім» және ішінара «жағдай» ғана беріледі. Тастықты оқыту жаттығу мұғалімнің ішінара қатысуымен жаттығу болып табылады, егер болжау тек алдын-ала сынақ үлгісі

Күшейту (ағылшын тілін жетілдіру - жақсарту) - машина оқыту алгоритмдерін үнемі құрастырудың тәртібі, онда әрбір алгоритм барлық алдыңғы алгоритмдердің құрамының кемшіліктерін өтеуге тырысады.

байесовский желісі

Машина оқыту арқылы шешілетін классикалық мәселелер:

– жіктеу, әдетте, нақты оқыту кезеңінде мұғаліммен бірге оқыту арқылы жүзеге асырылады;

– кластерлеу әдетте бақылаусыз оқыту арқылы орындалады;

– регрессия, әдетте, тестілеу сатысында мұғаліммен жаттығу арқылы орындалады, бұл тапсырманы болжаудың ерекше жағдайлары;

– деректердің өлшемін азайту және оларды визуализация мұғалімсіз оқыту арқылы жүзеге асырылады;

– деректер жиынтығының ықтималдықтың таралуын қалпына келтіру;

– одноклассная жіктеу және сәйкестендіру жаңалық;

– құрылыс деңгейінің тәуелділігі.

Оқуға арналған деректердің түрлері:

– объектілердің сипаттамалық сипаттамасы ең көп таралған жағдай;

– объектілер арасындағы қарым-қатынастың сипаттамасы, көбінесе жұптық ұқсастық, қашықтық матрицасы, ядро немесе деректер кестесі арқылы анықталған;

– уақыт сериясы немесе сигнал;

– кескін немесе бейнежазба.

Сапа функционалының түрлері:

Мұғаліммен жаттығу кезінде сапа функционалдығы жауаптардың орташа қателігі ретінде анықталуы мүмкін. Қалаған алгоритм оны барынша азайту керек деп есептеледі. Қайта даярлауды болдырмау үшін, регуляторды көбінесе сапа функционалына барынша азайтылуы үшін анық немесе жанама түрде қосылады.

Мұғалімсіз жаттығу кезінде сапа функционалдығын әртүрлі түрде анықтауға болады, мысалы, орташа кластерлер мен іштей қашықтыққа қатынасы.

Арматурамен жаттығулар кезінде сапалы функционалды заттар агентті бейімдеудің сапасын көрсететін физикалық ортада анықталады.

Машина оқыту көптеген бағыттарда кеңінен қолданылады, ол деректерді жинау және талдау бойынша жұмыстар жүргізеді. Машина оқыту - шын мәнінде жақындату әдістерінің дамуы, бірақ күрделі объектілер, күрделі сипатталған кеңістіктердің элементтері ұпай ретінде әрекет етеді, ал жауаптар тек сандар ғана емес, сонымен бірге белгіленуі мүмкін.

Машиналарды оқыту міндеттері белгілі нысандар жиынтығы мен жауаптардың жиынтығы арасындағы белгісіз қатынастарды табу мәселесіне дейін азаяды. Яғни объектілер жиынтығының нүктелеріне және кеңістіктің қалған бөлігіне жауаптардың жиынтығының мәндерін дәлірек жақындататын осындай функцияны құру қажет.

Нысандар түрінде кез-келген ақпарат болуы мүмкін: веб-беттер, елдер, адамдар, өнімдер, компаниялар.

Белгілеулер зерттелген кеңістіктегі заттардың сипаттамаларын өлшеу әдісі ретінде түсіндіріледі. Мәндердің жиынына байланысты осындай белгілерге бөлуге болады:

- бинарлық;
- номиналды;
- Тапсырыс;
- сандық;
- сапа.

Жауаптардың негізінде машина оқытудың міндеттері түрлерге бөлінеді.

Машиналарды оқытудың негізгі түрлері:

- сыныптау міндеттері;
- регрессиялық мәселелер
- ранжирлеу тапсырмалары.

2 Машина оқыту бағдарламасын құрудың компоненттері

2.1 OpenCV кітапханасын қолдану тиімділігі

OpenCV (ағылшын тілінің ашық көзі компьютерлік көрнекі кітапханасы, ашық көзбен көруге арналған кітапхана) - компьютерді көру, бейнені өңдеу және ашық бастапқы коды жалпы мақсаттағы сандық алгоритмдер. C / C ++-те Python, Java, Ruby, Matlab, Lua және басқа тілдер үшін әзірленген [2]. Оқу және коммерциялық мақсаттар үшін еркін қолдануға болады - BSD лицензиясы шарттарында таратылады.

Осы саладағы қосымшалар үшін бірыңғай стандартты компьютерді көру интерфейсін бекіту үшін. Мұндай қосымшалардың санын көбейтуге және компьютерді пайдаланудың жаңа үлгілерін жасауға мүмкіндік береді.

Intel Intelligence Libraries және MKL (арнайы нұсқасы LAPACK және FFTPack) көмегімен OpenCV-дің қосымша жеделдетуіне байланысты Intel платформаларын осындай қосымшаларды әзірлеушілер үшін тартымды етіңіз. OpenCV IPP және MKL-тің болуын автоматты түрде анықтай алады және оларды өңдеуді жеделдету үшін пайдаланады.

Қолдау көрсетілетін тұғырлар мен құралдар
Кітапханалардың өздері:

– microsoft windows: microsoft visual c ++ компиляторлары (6.0, .net 2003), intel compiler, borland c ++, mingw (gcc 3.x).

– windows rt: arm арқылы iteez [3].

– linux: gcc (2.9x, 3.x)

– mac os x: gcc (3.x, 4.x).

– android

– ios.

– c және жеңіл c ++ қолданылады. прагма және шартты компиляция өте шектеулі.

GUI құралдары, бейне түсіру:

Linux: V4L2, DC1394, FFMPEG.

Mac OS X: QuickTime

Құжаттау: статикалық HTML, PDF.

Негізгі модульдер.

2.2 нұсқасында кітапхана қайта құрылды. Sxcore, swaux, highGUI және басқалардың әмбебап модульдерінің орнына, тар мамандандырылған бірнеше ықшам модульдер жасалды:

opencv_core негізгі функционалдылық. Негізгі құрылымдар, есептеулер (математикалық функциялар, кездейсоқ сандар генераторлары) және сызықтық алгебра, DFT, DCT, XML / YAML үшін енгізу / шығару және т.б.

opencv_imgproc - кескінді өңдеу (сүзу, геометриялық өзгерістер, түс кеңістігін қайта құру және т.б.).

opencv_highgui - қарапайым UI, суреттерді және бейнені енгізу / шығару.

opencv_ml - машина оқыту модельдері (SVM, шешім ағаштары, ынталандыру және т.б.).

opencv_features2d - жалпақ примитивтерді тану және сипаттау (SURF (Eng.) орысша, FAST және басқалар, соның ішінде мамандандырылған негіз).

opencv_video - қозғалыс талдауы және нысандарды бақылау (оптикалық ағын, қозғалыстың үлгілері, өңді жою).

opencv_objdetect - суреттегі объектілерді анықтау (Виола-Джонс алгоритмін қолданып, Хоогдағы адамдарды тану және т.б.).

opencv_calib3d - камераны калибрлеу, стереополимерлерді іздеу және үш өлшемді деректерді өңдеу элементтері.

opencv_flann - ең жақын көршілерге жылдам іздеу (FLANN 1.5) және OpenCV орауыштары.

opencv_contrib - қолдануға дайын емес компаньон коды.

opencv_legacy - ескірген сыйысымдылық үшін сақталған ескірген код.

opencv_gpu - NVidia қолдауымен құрылған CUDA арқасында OpenCV кейбір функцияларын жеделдету.

2.2 Microsoft visual studio интегралданған бағдарламалау ортасын қолдану

Visual Studio - Microsoft ұсынатын әр түрлі жоғары деңгейлі тілдер үшін біріктірілген әзірлеу ортасы.

Visual Studio бағдарламашыларға Win32 / Win64 бағдарламаларын, сонымен қатар .NET Framework қосымшаларын жасауға мүмкіндік береді. Сонымен қатар, Visual Studio Windows бар бағдарламалар Интернет / Интранет немесе Azure қызметтер үшін динамикалық веб-сайттарды және веб-қызметтер дамиды. Тағы бір фокус Xamarin құралдар көзделген қызметтермен Windows Phone, Android және түйініне арналған мобильдік қосымшаларымен дамыту болып табылады.

Xamarin негізінде Mac үшін Visual Studio, бұл мүмкін, бірге MacOS үшін Visual Studio бірінші нұсқасы. NET Ол Xamarin Studio негізделген 10 мамыр, 2017 шығарылды [4], Негізгі және бірлік, қосымшалар мен ойындарды дамыту. Мақсатты платформалар - Android, macOS, iOS, Web және Cloud. Фокус C# және F#, сондай-ақ HTML, CSS және JSON.

Visual Studio көптеген мүмкіндіктерге ие ыңғайлы даму ортасы болып саналады. Редактор z қолдайды. Мысалы, курсор деңгейіндегі желідегі анықтама, код блоктарының жоғарылауын және жойылуын, кілт сөздерді түспен бөлектеуді, автоматты синтаксисті тексеруді және IntelliSense, автоматты әдісті және код енгізу кезінде функцияны аяқтауды қамтиды. Бұдан басқа, әзірлеу ортасы веб-қызметтер, ActiveX және .NET кітапханаларын интеграциялауға арналған графикалық интерфейстерді, деректер көздеріне қол жеткізу үшін «серверді зерттеуші» ұсынады: Мысалы, Microsoft SQL Сервері, Windows Жаңалықтар қызметі немесе WMI, сондай-ақ WYSIWYG редакторлары Windows және Веб бағдарламаларының пайдаланушылық интерфейсін жасау үшін белгішелер мен XML құжаттарын өңдеу үшін.

Visual Studio .NET (2002) болғандықтан, қолдау көрсетілетін тілдердің ауқымы айтарлықтай кеңейтілді. Microsoft корпорациясының аттестаттау бағдарламасы, әсіресе, үшінші тарап жеткізушілеріне өздерінің тілдерін Visual Studio бағдарламасына қосу ретінде ұсынуға мүмкіндік береді. Мысалдар Delphi, Eiffel, F# немесе Prolog. Visual Studio 2008 нұсқасында, Microsoft корпорациясының Java Visual J# нұсқасына қолдау көрсетілмеді.

Visual Studio кіріктірілген түзету құралын қамтиды. Бұл «өңдеу және жалғастыру» функциясын қамтиды және кейіннен жергілікті компьютерде және желіде іске қосылған процестерге қосымша мүмкіндік береді.

ASP.NET қосымшаларын әзірлеу үшін, Visual Studio Version 2005 немесе кейінгі нұсқасы біріктірілген веб-серверді қамтиды, сондықтан жеке IIS орнатылымы талап етілмейді. Сондай-ақ, осы нұсқаға арналған Reactor, ReSharper немесе Refactor сияқты қосымша өнімдермен («қондырмалар») жұмыс істейтін Refactoring! Pro-ді жаңартуға тура келді.

Visual Studio бағдарламасы қосымшаларды әзірлеу үшін көптеген редакторлар ұсынады. Кейбір маңыздылары:

- жоғарыда аталған бағдарламалау тілдеріне арналған код редакторлары;
- графикалық интерфейс редакторлары;
- дерекқор / дерекқор схемасының;
- HTML веб-беттеріне арналған редакторлар;
- XML, XSLT, стиль кестелері және түрлі xml негізіндегі теңшелім файлдары үшін редакторлар;
- XSD редакторы;
- графикалық редакторлар;
- екілік файлдар редакторлары;
- құрылғыны қолдануды орнатушыларға арналған редакторлар (графикалық);
- класс диаграммасының редакторы;
- схема схемасын өңдегіші.

Барлық мәтіндік редакторлар синтаксисті тексеруді (бастапқы кодтың дұрыстығын), синтаксисті бөлектеуді, IntelliSense (элементтердің автоматты ұсынысын) және автотолтыруды (элементтерді автоматты түрде толтыру) қолдайды.

2.3 Машина оқыту әдістерін қолдану

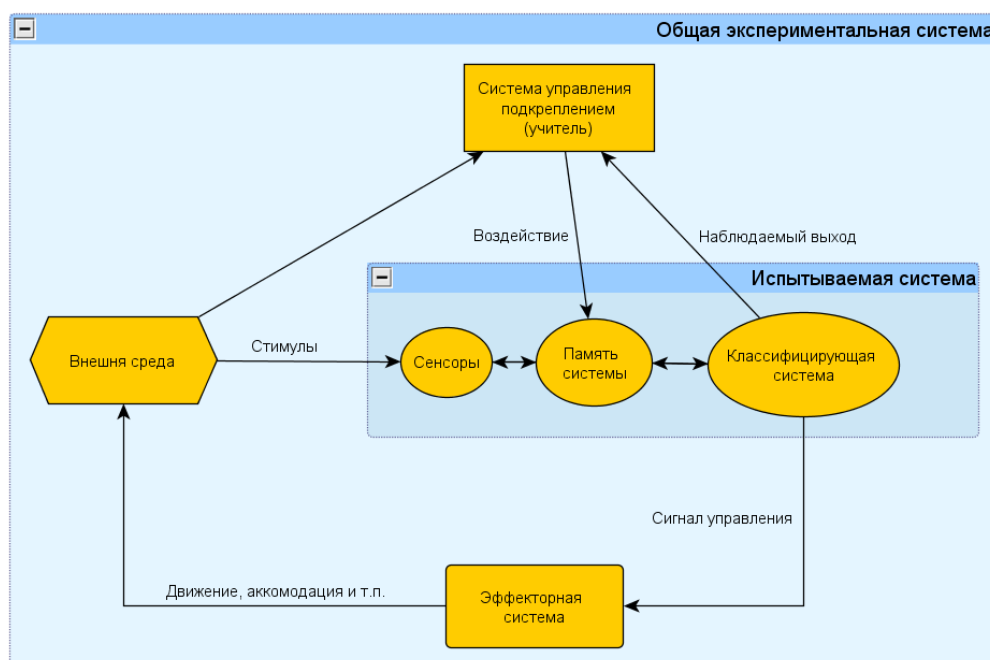
Оқытушымен сабақ беру (ағылшын тілін меңгеруді үйрену) машина оқытудың әдістерінің бірі болып табылады, оның барысында тест жүйесі «ынталандыру-жауап» мысалдарын қолдану арқылы күштеп үйренеді. Кибернетика тұрғысынан кибернетикалық эксперимент түрлерінің бірі. Кіріс және анықтамалық нәтиже арасында (ынталандыру-жауап) кейбір тәуелділік болуы мүмкін, бірақ ол белгісіз. Тек прецеденттердің соңғы жиынтығы ғана - оқыту жиынтығы деп аталатын ынталандыру-жауапты жұптар белгілі. Осы

деректерге сүйене отырып, тәуелділікті қалпына келтіру (болжау үшін қолайлы ынталандыру-жауап қарым-қатынас моделін құрастыру), яғни кез келген объектіге негізделген дәл жауапты шығаруға қабілетті алгоритм құру қажет. Жауаптардың дұрыстығын өлшеу үшін, сондай-ақ мысалды оқытуда сапа функционалдығын енгізуге болады.

Бұл эксперимент кері байланысқа ие кибернетикалық эксперименттің ерекше жағдайы болып табылады. Бұл эксперименттің формуляциясы эксперименттік жүйенің болуы, оқу әдісі және жүйені немесе өлшеу сипаттамаларын тексеру әдісін білдіреді.

Эксперименттік жүйе, өз кезегінде, сыналатын (қолданылатын) жүйе, сыртқы ортадан шығарылатын ынталандыру кеңістігін және ішкі бақылау параметрлерін басқару жүйесін (ішкі параметрлерді реттегіш) тұрады. Арматураны басқару жүйесі ретінде жүйедегі жадтың жай-күйін өзгертетін арнайы арматуралау ережелерін қолдану арқылы автоматты түрде реттеу құрылғысы (мысалы, термостат) немесе адамдық оператор (мұғалім) тестілеу кезінде жүйенің реакцияларына жауап беру үшін пайдаланылуы мүмкін.

Екі нұсқа бар: (1) сыналатын жүйенің реакциясы сыртқы орта жағдайын өзгертпесе және (2) жүйенің реакциясы сыртқы ортаны ынталандырады. Бұл схемалар биологиялық жүйеке жүйесімен жалпы түрдегі мұндай жүйенің іргелі ұқсастығын көрсетеді.



Мұғалімсіз сабақ беру (өзін-өзі үйрену, өзін-өзі үйрету, ағылшын тілін үйренуші оқыту) машина оқытудың әдістерінің бірі болып табылады, онда тестілеу жүйесі тағайындалған тапсырманы тәжірибешістан кедергісіз орындауға үйретеді. Кибернетика тұрғысынан бұл кибернетикалық эксперимент. Әдетте, ол нысандардың жиынтығы (оқыту үлгісі) белгілі болатын міндеттерге ғана жауап береді және объектілер арасындағы ішкі өзара байланысты, тәуелділікті, заңдылықты анықтау қажет.

Мұғалімсіз оқыту көбінесе мұғаліммен танысуға қайшы келеді, әр оқу нысаны үшін «дұрыс жауап» күшпен берілсе және ынталандыру мен жүйелік жауаптар арасында байланыс табу қажет.

Физиологиямен қарым-қатынас

Көптеген қолданбалы жетістіктерге қарамастан, мұғаліммен сабақ беру оның биологиялық қабілетсіздігі үшін сынға ұшырады. Мидағы оқытудың тетігін елестету қиын, ол қажетті және нақты шығу мәндерін салыстырады, кері байланыс арқылы түзету жүргізеді. Егер мидағы ұқсас механизмге жол берсек, онда қажетті нәтиже қайдан келеді? Мұғалімсіз оқыту - биологиялық жүйеде оқудың анағұрлым шынайы үлгісі. Кохонен және басқалар әзірлеген, шығудың мақсатты векторын қажет етпейді, демек, алдын-ала анықталған мінсіз жауаптармен салыстыруды талап етпейді [1].

Пішінді тану теориясында «мұғалімсіз оқыту» түсінігі

Теорияны құру және кибернетикалық эксперименттен әртүрлі теориялардан алыстау үшін, мұғалімсіз математикалық түрде оқу тәжірибесін ресімдеуге тырысады. Бұл ресімдеудің тұжырымдамасы мен анықталуының көптеген әртүрлі подтиптері бар, олардың біреуі үлгіні тану теориясында көрініс табады.

Эксперимент пен теорияның құрылуы осындай көзқарастар мамандардың көзқарастары бойынша әртүрлі пікірлермен байланысты. Әсіресе, сұраққа жауап беру кезінде пайда болған айырмашылықтар: «Бірегей қағидалар әртүрлі табиғат бейнелерін лайықты түрде сипаттау үшін мүмкін бе, әлде осындай сипаттама әрбір нақты білімі бар мамандар үшін міндетті ме?».

Бірінші жағдайда, мәлімдеме бейнелерді барабар сипаттауды дайындауда априорная ақпараттарды қолданудың жалпы қағидаттарын анықтауға бағытталуы тиіс. Мұнда әртүрлі табиғаттың бейнелеріне қатысты априориалдық ақпарат әртүрлі және оларды ескеру принципі бірдей. Екінші жағдайда, жалпы сипаттағы сипаттаманы алу мәселесі алынып тасталды, ал үлгіні танудың статистикалық теориясының тұрғысынан оқыту үлгісін тану теориясы теориясының шешім қабылдаудың ерекше класындағы орташа тәуекелді азайту проблемасына дейін азайтылуы мүмкін [2].

Үлгіні тану теориясында осы мәселеге негізінен үш тәсіл бар [3]:

Эвристикалық әдістер;

Математикалық әдістер;

Лингвистикалық (синтаксистік) әдістер.

Болжау бойынша тапсырмалар - болжауда классикалық болжау міндеттерінің әртүрлі түрлері бар. Осындай тапсырмаларды бірыңғай түрде қалыптастыру әртүрлі пәндермен ұсынылған әр түрлі әдістерді салыстыруға мүмкіндік береді.

Машина оқыту көптеген бағыттарда кеңінен қолданылады, ол деректерді жинау және талдау бойынша жұмыстар жүргізеді. Машина оқыту - шын мәнінде жақындату әдістерінің дамуы, бірақ күрделі объектілер, күрделі сипатталған кеңістіктердің элементтері ұпай ретінде әрекет етеді, ал жауаптар тек сандар ғана емес, сонымен бірге белгіленуі мүмкін.

Машиналарды оқыту міндеттері белгілі нысандар жиынтығы мен жауаптардың жиынтығы арасындағы белгісіз қатынастарды табу проблемасына дейін азаяды. Яғни объектілер жиынтығының нүктелеріне және кеңістіктің қалған бөлігіне жауаптардың жиынтығының мәндерін дәлірек жақындататын осындай функцияны құру қажет.

3 ПРОГРАММАЛЫҚ ҚАМТАМАНЫ ҚҰРУ

3.1 Программалау тілін таңдауды негіздеу

C# (анықталған C-Sharp) - объектілі-бағдарланған және контекстік-бағдарланған ұғымдарды біріктіретін бағдарламалау тілі. 1998-2001 жж. Microsoft корпорациясының Anders Heilсберг басшылығымен инженерлер тобы Microsoft корпорациясының Microsoft .NET платформасына арналған қосымшаларды әзірлеудің негізгі тілі ретінде жасалды. C # құрастырушысы .NET-дің стандартты орнатылымына қосылған, сондықтан оған арналған бағдарламалар Visual Studio сияқты құралдарсыз құрастырылып, құрастырылуы мүмкін.

C# тілі C тіліндегі синтаксисі бар C тіліндегі синтаксисі бар, оның синтаксисі C++ және Java-ге ең жақын. Тілде қатаң статикалық теру, полиморфизмді, оператор жүктемесін қолдайды, сыныптың мүшелік функцияларына, атрибуттарға, оқиғаларға, қасиеттерге, ерекшеліктерге, XML пішіміндегі түсініктемелерге нұсқайды. C++, Delphi, Modula және Smalltalk - C # бағдарламаларын қолдану тәжірибесіне сүйене отырып, бағдарламалық қамтамасыз етуді әзірлеу кезінде қиындықтар туындаған кейбір модельдерді жоққа шығарады: мысалы, C# бірнеше сыныпты иеленуді қолдамайды немесе Haskell-қа қарағанда басқаша түрін енгізіңіз.

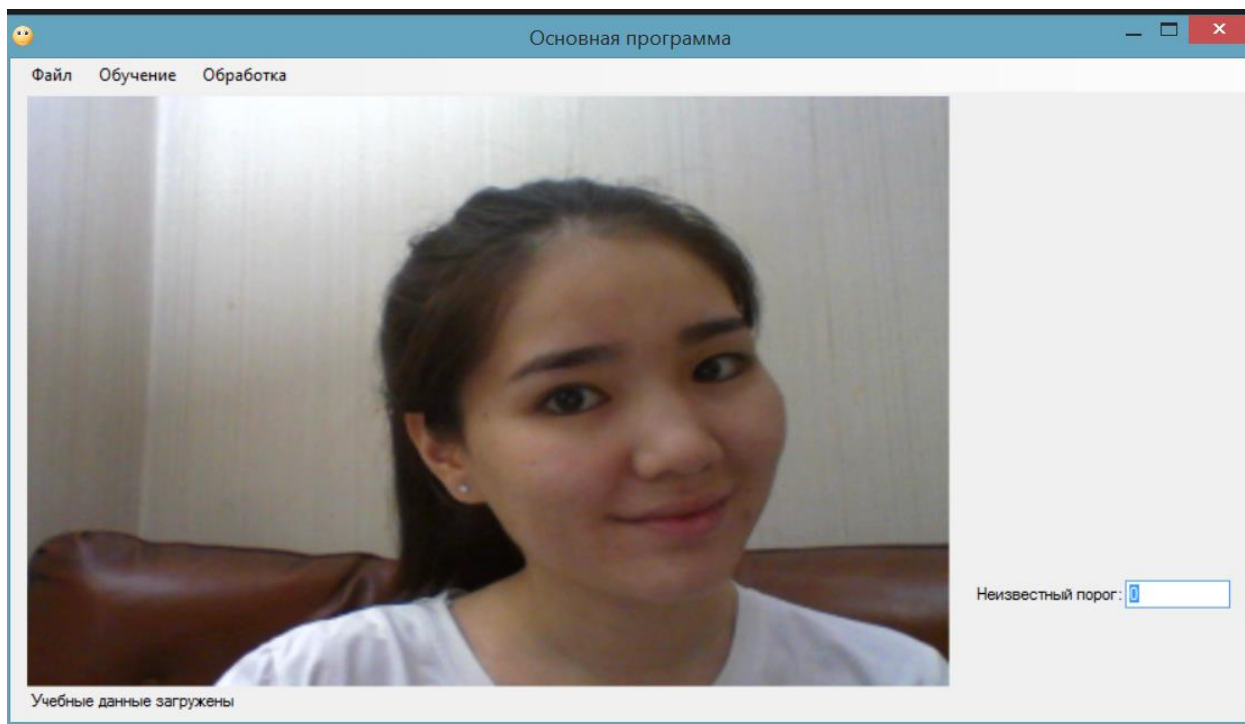
3.2 Программаның баяндалуы

Бет тану жүйесі - бұл адамды сандық кескіннен бейнебақылау немесе бейне магнитофоннан сәйкестендіруге немесе тексеруге қабілетті технология болып табылады. Бет тану жүйелерінің жұмыс жасайтын бірнеше әдістері бар, бірақ олар жалпы деректер базасындағы тұлғалармен берілген кескіннен таңдалған бет-бейнесін салыстыру арқылы жұмыс істейді. Сондай-ақ, адамның беті текстурасына және формасына негізделген үлгілерді талдап, адамды бірегей түрде анықтауға мүмкіндік беретін биометриялық жасанды интеллект негізіндегі қосымша ретінде сипатталады.

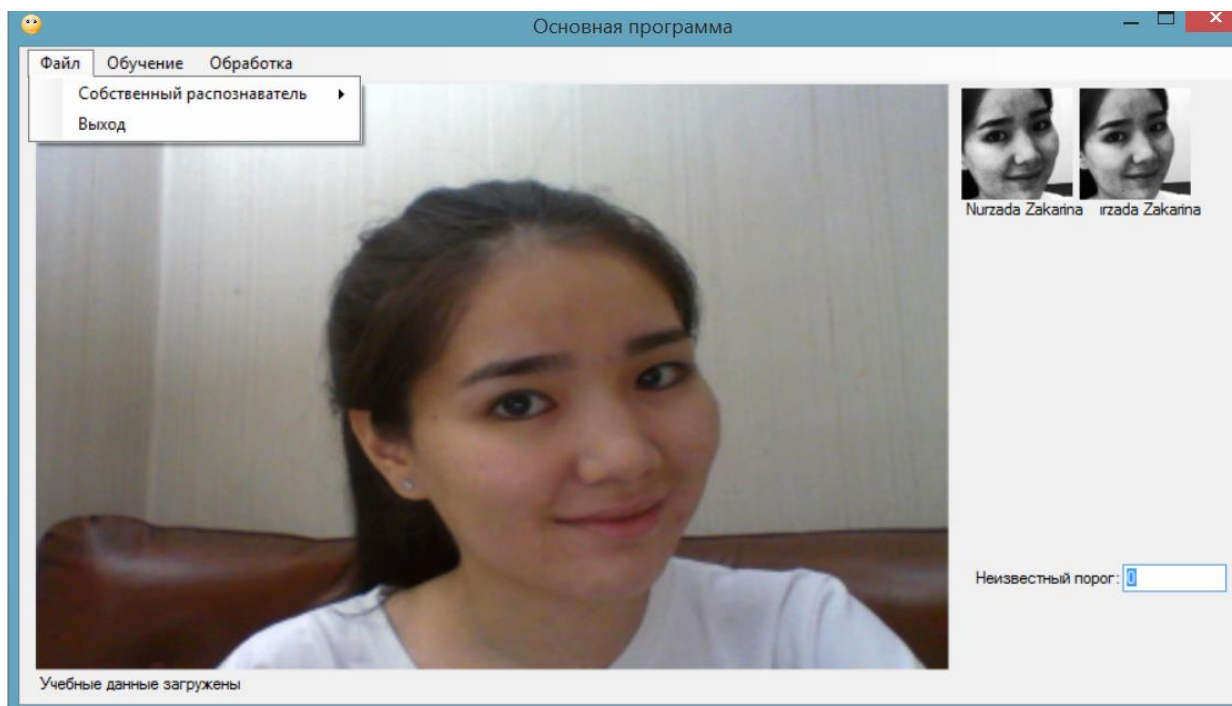
Алғашында компьютерді қолданудың нысаны, ол соңғы кездері мобильді платформаларда және робототехника секілді технологиялардың басқа түрлерінде кең қолданды. Ол әдетте қауіпсіздік жүйесіндегі кіруді бақылау ретінде пайдаланылады және саусақ ізі немесе көздің иристі тану сияқты басқа биометриямен салыстыруға болады. Биометриялық технология ретінде бет тану жүйесінің дәлдігі ирис танудан және саусақ іздерін танудан төмен болғанымен, ол бесконтактсыз және инвазивті емес процесс арқылы кеңінен қабылданады. Жақында ол коммерциялық сәйкестендіру және маркетинг құралы ретінде танымал болды. Басқа бағдарламалар арасында адамның компьютерлік өзара әрекеті, бейнебақылау, суреттерді автоматты түрде индекстеу және бейне дерекқоры бар.

3.3 Функционалдык тағайындалуы

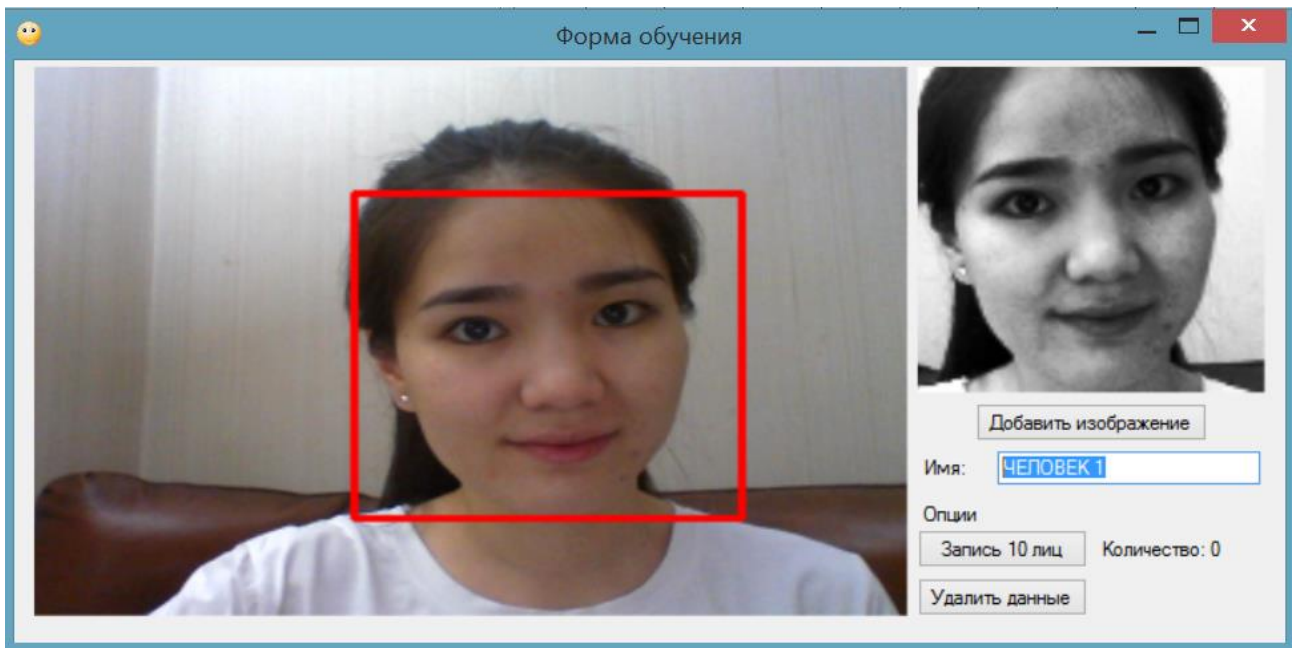
Visual Studio ортасын қолданып с# бағдарламалау тілінде бетті тану бағдарламасын жасау.



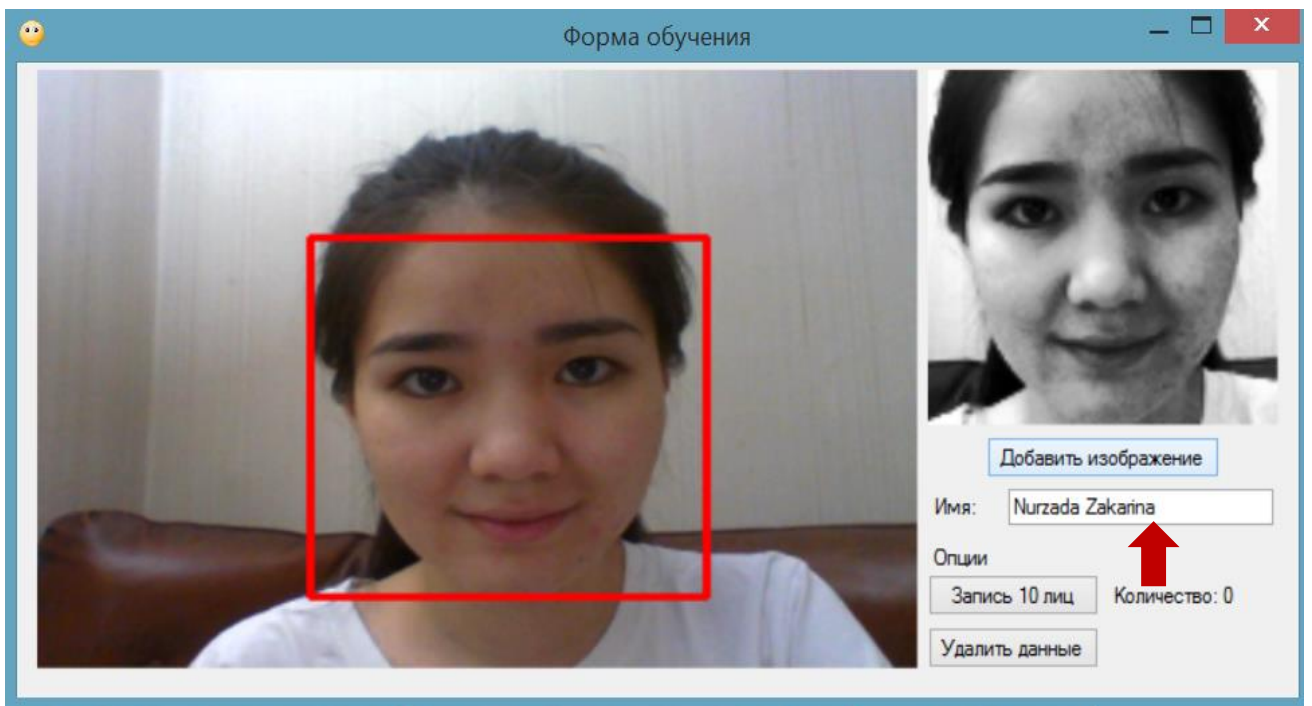
3.1 – сурет. Негізгі беті



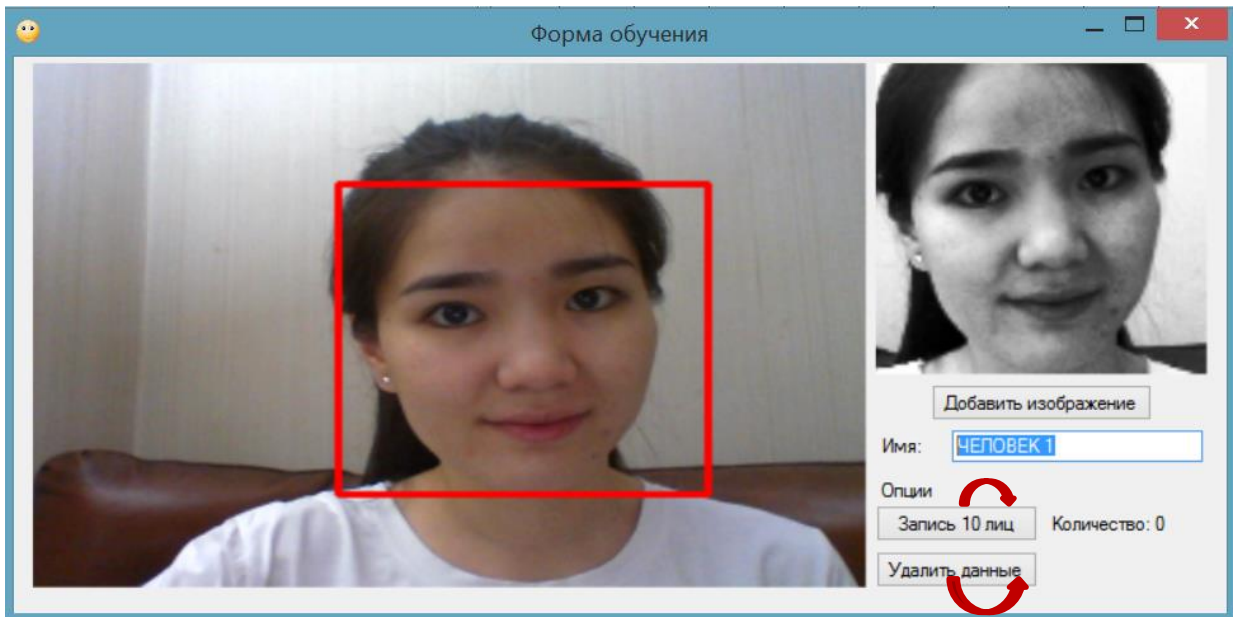
3.2 – сурет. Бағдарламаның негізгі бөлімі



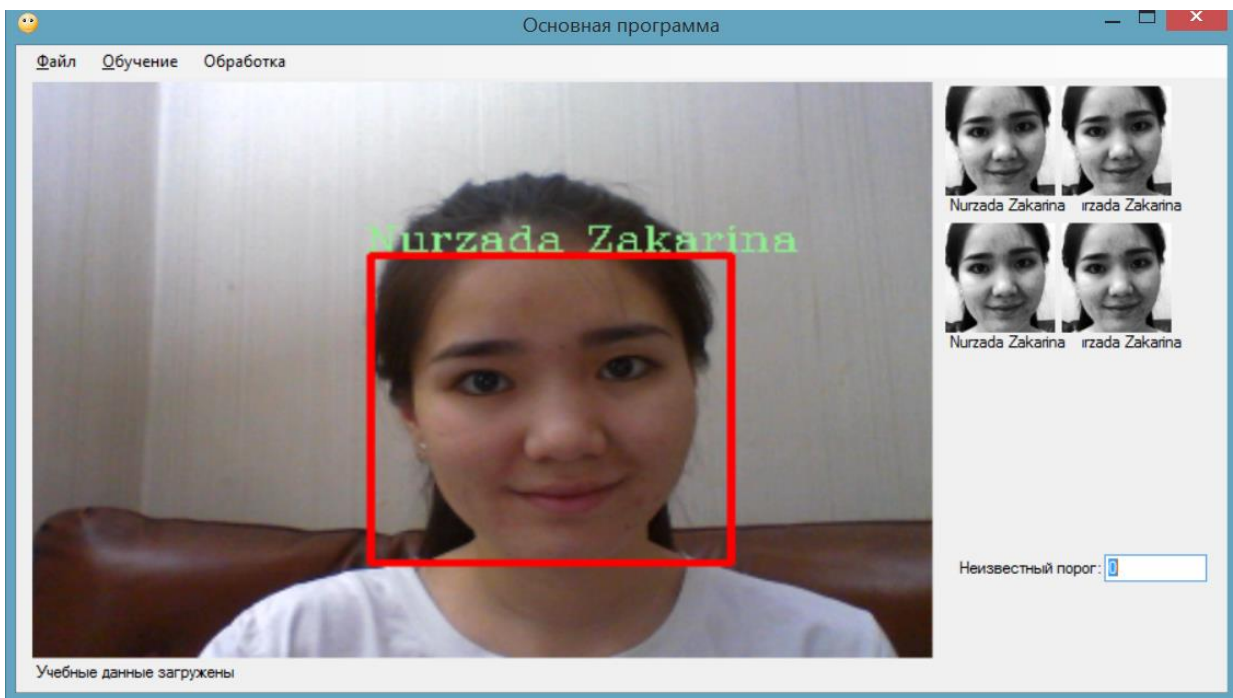
3.3 – сурет. Машина оқытуды жүргізу



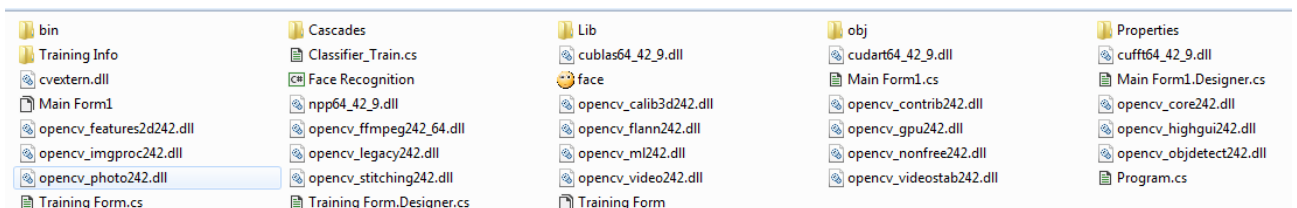
3.4 – сурет. Адамның аты-жөнін енгізу



3.5 – сурет. Беттік бейнелердің жазбалары және мәліметтерді жою



3.6 – сурет. Бетті тану нәтижесін шығару



3.6 – сурет. Бағдарламаның физикалық құрылымы

3.4 Қолданылған техникалық жабдықтар

«Машина оқыту» бағдарламасын құру үшін Windows 8 операциялық жүйесінде Asus in search of incredible 500 GB ноутбугы қолданылды.

3.5 Шақыру және жүктеу

«Машина оқыту» бағдарламасын Microsoft visual studio ортасында ашу үшін Пуск → Программы → Microsoft visual studio → FaceRecognition жобасын ашамыз.

3.6 Кіріс мәліметтер

Бағдарламаның кіріс мәліметтері арқылы бағдарламаға адам келбетін енгіземіз.

3.7 Шығыс мәліметтер

Шығыс мәліметтерге аты-жөні енгізілген адамның беттік бейнесі арқылы тануы және нәтижесін шығару.

ҚОРЫТЫНДЫ

Мен бұл жұмыста бетті тану жүйесін жасадым. Бетті тану - бұл адамның бет-әлпетін технология арқылы тану тәсілі. Бетті тану жүйесі фотосуреттер мен бейнелердің бет-әлпетін бейнелеу үшін биометрияны пайдаланады. Ол деректерді табу үшін белгілі тұлғалардың деректер базасымен ақпаратпен салыстырылады. Бет-әлпетті тану жеке сәйкестендіруді тексеруге көмектеседі, бірақ ол сонымен бірге құпиялылық мәселелерін көтереді.

Адамның беті біздің өзара қарым-қатынасымызда маңызды рөл атқарады, ол адамдардың жеке басын жеткізеді. Адамның бет-жүзін қауіпсіздіктің кілті ретінде қолдану арқылы адам құқықтарын қорғаудың биометриялық технологиясы соңғы бірнеше жылда құқық қорғау органдарында да, құқық қорғау органдарында да кең ауқымды қолдану мүмкіндігіне байланысты үлкен көңіл бөлді.

Бағдарламалық қамтамасыз ету адамның бетіндегі 80 түйін нүктесін анықтайды. Осыған байланысты түйін нүктелері адамның бетінің айнымалы мәндерін, мысалы, мұрынның ұзындығы мен ені, көздің розеткалары тереңдігі және бет жағының пішіні сияқты өлшеуге арналған соңғы нүктелер болып табылады. Жүйе адамның бетінің сандық кескіні бойынша түйін нүктелері үшін деректерді жинау және алынған деректерді беткі із ретінде сақтау арқылы жұмыс істейді. Бет суреті кейін бейнені немесе бейнедегі тұлғалардан алынған деректермен салыстыру үшін негіз ретінде пайдаланылады.

Бет тану жүйесі 80 түйін нүктесін ғана пайдаланады, бірақ бұл жағдай қолайлы болған кезде мақсатты тұлғаларды жылдам және дәл анықтауы мүмкін. Дегенмен, егер субъектінің беті алдын-ала қаралмай, жартылай қараңғылық немесе профильде болса немесе жарық жеткіліксіз болса, бағдарламалық жасақтаманың бұл түрі кем сенімдірек болады. Ұлттық стандарттар мен технологиялар институтының (NIST) деректері бойынша, 1993 жылдан бері бет-бейнені тану жүйелеріндегі жалған оң жағдайлар жиілігі екі жылда екі есе азайды. Сондықтан бетті тану жүйесі қолайлы болып табылады.

ПАЙДАЛАНҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: основы моделирования и первичная обработка данных. — М.: Финансы и статистика, 1983.
2. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. — М.: Финансы и статистика, 1985.
3. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.
4. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.
5. Журавлев Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006. ISBN 5-7036-0108-8.
6. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. ISBN 5-86134-060-9.
7. Флах П. Машинное обучение. — М.: ДМК Пресс, 2015. — 400 с. — ISBN 978-5-97060-273-7.
8. Шлезингер М., Главач В. Десять лекций по статистическому и структурному распознаванию. — Киев: Наукова думка, 2004. ISBN 966-00-0341-
9. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 p. — ISBN 978-0-387-84857-0..
10. Mitchell T. Machine Learning. — McGraw-Hill Science/Engineering/Math, 1997. ISBN 0-07-042807-7.
11. Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell (1983), Machine Learning: An Artificial Intelligence Approach, Tioga Publishing Company, ISBN 0-935382-05-4 (Machine Learning: An Artificial Intelligence Approach в «Книгах Google»).
12. Vapnik V. N. Statistical learning theory. — N.Y.: John Wiley & Sons, Inc., 1998. [1]
13. Bernhard Schölkopf, Alexander J. Smola Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. — MIT Press, Cambridge, MA, 2002 ISBN 978-0-262-19475-4 [2]
14. Liang Wang, Li Cheng, Guoying Zhao. Machine Learning for Human Motion Analysis. — IGI Global, 2009. — 318 p. — ISBN 978-1-60566-900-7.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Emgu.CV.UI;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
using System.Xml;
using System.Runtime.InteropServices;
using System.Threading;
using System.Security.Principal;
using System.Threading.Tasks;
using Microsoft.Win32.SafeHandles;
namespace Face_Recognition
{
public partial class Form1 : Form
{
#region variables
Image<Bgr, Byte> currentFrame;
Image<Gray, byte> result, TrainedFace = null;
Image<Gray, byte> gray_frame = null;
Capture grabber;
public HaarCascade Face = new HaarCascade(Application.StartupPath +
"/Cascades/haarcascade_frontalface_alt2.xml");//haarcascade_frontalface_alt_tree.xml");
MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_COMPLEX, 0.5,
0.5);
int NumLabels;
//Classifier with default training location
Classifier_Train Eigen_Recog = new Classifier_Train();
#endregion
public Form1 ()
{
InitializeComponent();
//Load of previous trained faces and labels for each image
if (Eigen_Recog.IsTrained)
{
message_bar.Text = "Учебные данные загружены";
}
}
}
}

```

```

}
else
{
message_bar.Text = "Тренировочные данные не найдены, пожалуйста, обучайте
программу, используя опцию меню Обучение";
}
initialise_capture();
}
//Open training form and pass this
private void trainToolStripMenuItem_Click(object sender, EventArgs e)
{
//Stop Camera
stop_capture();
//OpenForm
Training_Form TF = new Training_Form(this);
TF.Show();
}
public void retrain()
{
Eigen_Recog = new Classifier_Train();
if (Eigen_Recog.IsTrained)
{
message_bar.Text = "Учебные данные загружены";
}
else
{
message_bar.Text = "Тренировочные данные не найдены, пожалуйста, обучайте
программу, используя опцию меню Обучение";
}
}
//Camera Start Stop
public void initialise_capture()
{
grabber = new Capture();
grabber.QueryFrame();
//Initialize the FrameGraber event
if (parrellelToolStripMenuItem.Checked)
{
Application.Idle += new EventHandler(FrameGrabber_Parrellel);
}
else
{
Application.Idle += new EventHandler(FrameGrabber_Standard);
}
}

```

```

}
private void stop_capture()
{
if (parallelToolStripMenuItem.Checked)
{
Application.Idle -= new EventHandler(FrameGrabber_Parrellel);
}
else
{
Application.Idle -= new EventHandler(FrameGrabber_Standard);
}
if (grabber != null)
{
grabber.Dispose();
}
}
//Process Frame
void FrameGrabber_Standard(object sender, EventArgs e)
{
//Get the current frame form capture device
currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
//Convert it to Grayscale
if (currentFrame != null)
{
gray_frame = currentFrame.Convert<Gray, Byte>();
//Face Detector
MCvAvgComp[][] facesDetected = gray_frame.DetectHaarCascade(Face, 1.2, 10,
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new
Size(50, 50));

//Action for each element detected
foreach (MCvAvgComp face_found in facesDetected[0])
{
result = currentFrame.Copy(face_found.rect).Convert<Gray, byte>().Resize(100,
100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
//draw the face detected in the 0th (gray) channel with blue color
currentFrame.Draw(face_found.rect, new Bgr(Color.Red), 2);
if (Eigen_Recog.IsTrained)
{
string name = Eigen_Recog.Recognise(result);
int match_value = (int)Eigen_Recog.Get_Eigen_Distance;

//Draw the label for each face detected and recognized

```

```

currentFrame.Draw(name + " ", ref font, new Point(face_found.rect.X - 2,
face_found.rect.Y - 2), new Bgr(Color.LightGreen));
ADD_Face_Found(result, name, match_value);
}
}
//Show the faces procesed and recognized
image_PICBX.Image = currentFrame.ToBitmap();
}
}
void FrameGrabber_Parrellel(object sender, EventArgs e)
{
//Get the current frame form capture device
currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
//Convert it to Grayscale
//Clear_Faces_Found();
if (currentFrame != null)
{
gray_frame = currentFrame.Convert<Gray, Byte>();
//Face Detector
MCvAvgComp[][] facesDetected = gray_frame.DetectHaarCascade(Face, 1.2, 10,
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new
Size(50, 50));
//Action for each element detected
Parallel.ForEach(facesDetected[0], face_found =>
{
try
{
result = currentFrame.Copy(face_found.rect).Convert<Gray, byte>().Resize(100,
100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
result._EqualizeHist();
//draw the face detected in the 0th (gray) channel with blue color
currentFrame.Draw(face_found.rect, new Bgr(Color.Red), 2);
if (Eigen_Recog.IsTrained)
{
string name = Eigen_Recog.Recognise(result);
int match_value = (int)Eigen_Recog.Get_Eigen_Distance;
//Draw the label for each face detected and recognized
currentFrame.Draw(name + " ", ref font, new Point(face_found.rect.X - 2,
face_found.rect.Y - 2), new Bgr(Color.LightGreen));
ADD_Face_Found(result, name, match_value);
}
}
}
}
}

```



```

catch
{
//do nothing as parallel loop buggy
//No action as the error is useless, it is simply an error in
//no data being there to process and this occurs sporadically
}
});
//Show the faces processed and recognized
image_PICBX.Image = currentFrame.ToBitmap();
}
}
//ADD Picture box and label to a panel for each face
int faces_count = 0;
int faces_panel_Y = 0;
int faces_panel_X = 0;
void Clear_Faces_Found()
{
this.Faces_Found_Panel.Controls.Clear();
faces_count = 0;
faces_panel_Y = 0;
faces_panel_X = 0;
}
void ADD_Face_Found(Image<Gray, Byte> img_found, string name_person, int
match_value)
{
PictureBox PI = new PictureBox();
PI.Location = new Point(faces_panel_X, faces_panel_Y);
PI.Height = 80;
PI.Width = 80;
PI.SizeMode = PictureBoxSizeMode.StretchImage;
PI.Image = img_found.ToBitmap();
Label LB = new Label();
LB.Text = name_person + " " + match_value.ToString();
LB.Location = new Point(faces_panel_X, faces_panel_Y + 80);
//LB.Width = 80;
LB.Height = 15;
this.Faces_Found_Panel.Controls.Add(PI);
this.Faces_Found_Panel.Controls.Add(LB);
faces_count++;
if (faces_count == 2)
{
faces_panel_X = 0;
faces_panel_Y += 100;
faces_count = 0;
}
}
}

```

```

}
else faces_panel_X += 85;
if (Faces_Found_Panel.Controls.Count > 10)
{
Clear_Faces_Found();
}
}
//Menu Opeartions
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
this.Dispose();
}
private void singleToolStripMenuItem_Click(object sender, EventArgs e)
{
parallelToolStripMenuItem.Checked = false;
singleToolStripMenuItem.Checked = true;
Application.Idle -= new EventHandler(FrameGrabber_Parrellel);
Application.Idle += new EventHandler(FrameGrabber_Standard);
}
private void parrellelToolStripMenuItem_Click(object sender, EventArgs e)
{
parallelToolStripMenuItem.Checked = true;
singleToolStripMenuItem.Checked = false;
Application.Idle -= new EventHandler(FrameGrabber_Standard);
Application.Idle += new EventHandler(FrameGrabber_Parrellel);
}
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
SaveFileDialog SF = new SaveFileDialog();
SF.Filter = "XML File *.xml| *.xml";
if (SF.ShowDialog() == DialogResult.OK)
{
Eigen_Recog.Save_Eigen_Recogniser(SF.FileName);
}
}
private void loadToolStripMenuItem_Click(object sender, EventArgs e)
{
OpenFileDialog OF = new OpenFileDialog();
OF.Filter = "XML File *.xml| *.xml";
if (OF.ShowDialog() == DialogResult.OK)
{
Eigen_Recog.Load_Eigen_Recogniser(OF.FileName);
}
}
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
}
//Unknow face calibration
private void Eigen_threshold_txtbxChanged(object sender, EventArgs e)
{
try
{
Eigen_Recog.Set_Eigen_Threshold
Math.Abs(Convert.ToInt32(Eigne_threshold_txtbx.Text));
message_bar.Text = "Собственный порог";
}
catch
{
message_bar.Text = "Ошибка в пороговом входе, пожалуйста, используйте целое
число";
}
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Emgu.CV.UI;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
using System.Drawing.Imaging;
using System.Xml;
using System.Threading;
namespace Face_Recognition
{
public partial class Training_Form : Form
{
#region Variables
//Camera specific
Capture grabber;

```

```

//Images for finding face
Image<Bgr, Byte> currentFrame;
Image<Gray, byte> result = null;
Image<Gray, byte> gray_frame = null;
//Classifier
HaarCascade Face;
//For aquiring 10 images in a row
List<Image<Gray, byte>> resultImages = new List<Image<Gray, byte>>();
int results_list_pos = 0;
int num_faces_to_acquire = 10;
bool RECORD = false;
//Saving Jpg
List<Image<Gray, byte>> ImagestoWrite = new List<Image<Gray, byte>>();
EncoderParameters ENC_Parameters = new EncoderParameters(1);
EncoderParameter ENC = new EncoderParameter(EncoderParameter(System.Drawing.Imaging.Encoder.Quality, 100));
ImageCodecInfo Image_Encoder_JPG;
//Saving XAML Data file
List<string> NamestoWrite = new List<string>();
List<string> NamesforFile = new List<string>();
XmlDocument docu = new XmlDocument();
//Variables
Form1 Parent;
#endregion
public Training_Form(Form1 _Parent)
{
InitializeComponent();
Parent = _Parent;
Face = Parent.Face;
//Face = new HaarCascade(Application.StartupPath +
"/Cascades/haarcascade_frontalface_alt2.xml");
ENC_Parameters.Param[0] = ENC;
Image_Encoder_JPG = GetEncoder(ImageFormat.Jpeg);
initialise_capture();
}

private void Training_Form_FormClosing(object sender, FormClosingEventArgs e)
{
stop_capture();
Parent.retrain();
Parent.initialise_capture();
}

```

```

//Camera Start Stop
public void initialise_capture()
{
grabber = new Capture();
grabber.QueryFrame();
//Initialize the FrameGraber event
Application.Idle += new EventHandler(FrameGrabber);
}
private void stop_capture()
{
Application.Idle -= new EventHandler(FrameGrabber);
if (grabber != null)
{
grabber.Dispose();
}
//Initialize the FrameGraber event
}
//Process Frame
void FrameGrabber(object sender, EventArgs e)
{
//Get the current frame form capture device
currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
//Convert it to Grayscale
if (currentFrame != null)
{
gray_frame = currentFrame.Convert<Gray, Byte>();
//Face Detector
MCvAvgComp[][] facesDetected = gray_frame.DetectHaarCascade(Face, 1.2, 10,
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new
Size(20, 20));
//Action for each element detected
foreach (MCvAvgComp face_found in facesDetected[0])
{
result = currentFrame.Copy(face_found.rect).Convert<Gray, byte>().Resize(100,
100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
result._EqualizeHist();
face_PICBX.Image = result.ToBitmap();
//draw the face detected in the 0th (gray) channel with blue color
currentFrame.Draw(face_found.rect, new Bgr(Color.Red), 2);

}
if (RECORD && facesDetected.Length > 0 && resultImages.Count <
num_faces_to_acquire)

```

```

{
resultImages.Add(result);
count_lbl.Text = "Количество: " + resultImages.Count.ToString();
if (resultImages.Count == num_faces_to_acquire)
{
ADD_BTN.Enabled = true;
NEXT_BTN.Visible = true;
PREV_btn.Visible = true;
count_lbl.Visible = false;
Single_btn.Visible = true;
ADD_ALL.Visible = true;
RECORD = false;
Application.Idle -= new EventHandler(FrameGrabber);
}
}
image_PICBX.Image = currentFrame.ToBitmap();
}
}
//Saving The Data
private bool save_training_data(Image face_data)
{
try
{
Random rand = new Random();
bool file_create = true;
string facename = "face_" + NAME_PERSON.Text + "_" + rand.Next().ToString() +
".jpg";
while (file_create)
{
if (!File.Exists(Application.StartupPath + "/TrainedFaces/" + facename))
{
file_create = false;
}
else
{
facename = "face_" + NAME_PERSON.Text + "_" + rand.Next().ToString() +
".jpg";
}
}
}

if(Directory.Exists(Application.StartupPath + "/TrainedFaces/"))
{

```

```

face_data.Save(Application.StartupPath + "/TrainedFaces/" + facename,
ImageFormat.Jpeg);
}
else
{
Directory.CreateDirectory(Application.StartupPath + "/TrainedFaces/");
face_data.Save(Application.StartupPath + "/TrainedFaces/" + facename,
ImageFormat.Jpeg);
}
if (File.Exists(Application.StartupPath + "/TrainedFaces/TrainedLabels.xml"))
{
//File.AppendAllText(Application.StartupPath + "/TrainedFaces/TrainedLabels.txt",
NAME_PERSON.Text + "\n\r");
bool loading = true;
while (loading)
{
try
{
docu.Load(Application.StartupPath + "/TrainedFaces/TrainedLabels.xml");
loading = false;
}
catch
{
docu = null;
docu = new XmlDocument();
Thread.Sleep(10);
}
}
//Get the root element
XmlElement root = docu.DocumentElement;
XmlElement face_D = docu.CreateElement("FACE");
XmlElement name_D = docu.CreateElement("NAME");
XmlElement file_D = docu.CreateElement("FILE");
//Add the values for each nodes
//name.Value = textBoxName.Text;
//age.InnerText = textBoxAge.Text;
//gender.InnerText = textBoxGender.Text;
name_D.InnerText = NAME_PERSON.Text;
file_D.InnerText = facename;
//Construct the Person element
//person.Attributes.Append(name);
face_D.AppendChild(name_D);
face_D.AppendChild(file_D);
//Add the New person element to the end of the root element

```

```

root.AppendChild(face_D);
//Save the document
docu.Save(Application.StartupPath + "/TrainedFaces/TrainedLabels.xml");
//XmlElement child_element = docu.CreateElement("FACE");
//docu.AppendChild(child_element);
//docu.Save("TrainedLabels.xml");
}
else
{
FileStream    FS_Face    =    File.OpenWrite(Application.StartupPath    +
"/TrainedFaces/TrainedLabels.xml");
using (XmlWriter writer = XmlWriter.Create(FS_Face))
{
writer.WriteStartDocument();
writer.WriteStartElement("Faces_For_Training");
writer.WriteStartElement("FACE");
writer.WriteElementString("NAME", NAME_PERSON.Text);
writer.WriteElementString("FILE", facename);
writer.WriteEndElement();
writer.WriteEndElement();
writer.WriteEndDocument();
}
FS_Face.Close();
}

return true;
}
catch (Exception ex)
{
return false;
}

}
private ImageCodecInfo GetEncoder(ImageFormat format)
{
ImageCodecInfo[] codecs = ImageCodecInfo.GetImageDecoders();
foreach (ImageCodecInfo codec in codecs)
{
if (codec.FormatID == format.Guid)
{
return codec;
}
}
return null;
}

```



```

}
//Delete all the old training data by simply deleting the folder
private void Delete_Data_BTN_Click(object sender, EventArgs e)
{
if (Directory.Exists(Application.StartupPath + "/TrainedFaces/"))
{
Directory.Delete(Application.StartupPath + "/TrainedFaces/", true);
Directory.CreateDirectory(Application.StartupPath + "/TrainedFaces/");
}
}
//Add the image to training data
private void ADD_BTN_Click(object sender, EventArgs e)
{
if (resultImages.Count == num_faces_to_acquire)
{
if (!save_training_data(face_PICBX.Image))    MessageBox.Show("Ошибка",
"Ошибка при сохранении информации о файле. Тренировочные данные не
сохранены", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
{
stop_capture();
if (!save_training_data(face_PICBX.Image))    MessageBox.Show("Ошибка",
"Ошибка при сохранении информации о файле. Тренировочные данные не
сохранены", MessageBoxButtons.OK, MessageBoxIcon.Error);
initialise_capture();
}
}
private void Single_btn_Click(object sender, EventArgs e)
{
RECORD = false;
resultImages.Clear();
NEXT_BTN.Visible = false;
PREV_btn.Visible = false;
Application.Idle += new EventHandler(FrameGrabber);
Single_btn.Visible = false;
count_lbl.Text = "Count: 0";
count_lbl.Visible = true;
}
//Get 10 image to train
private void RECORD_BTN_Click(object sender, EventArgs e)
{
if (RECORD)
{

```

```

RECORD = false;
}
else
{
if (resultImages.Count == 10)
{
resultImages.Clear();
Application.Idle += new EventHandler(FrameGrabber);
}
RECORD = true;
ADD_BTN.Enabled = false;
}
}
private void NEXT_BTN_Click(object sender, EventArgs e)
{
if (results_list_pos < resultImages.Count - 1)
{
face_PICBX.Image = resultImages[results_list_pos].ToBitmap();
results_list_pos++;
PREV_btn.Enabled = true;
}
else
{
NEXT_BTN.Enabled = false;
}
}
private void PREV_btn_Click(object sender, EventArgs e)
{
if (results_list_pos > 0)
{
results_list_pos--;
face_PICBX.Image = resultImages[results_list_pos].ToBitmap();
NEXT_BTN.Enabled = true;
}
else
{
PREV_btn.Enabled = false;
}
}
private void ADD_ALL_Click(object sender, EventArgs e)
{
for(int i = 0; i<resultImages.Count;i++)
{
face_PICBX.Image = resultImages[i].ToBitmap();
}
}

```

```

if (!save_training_data(face_PICBX.Image))    MessageBox.Show("Ошибка",
"Ошибка при сохранении информации о файле. Тренировочные данные не
сохранены", MessageBoxButtons.OK, MessageBoxIcon.Error);
Thread.Sleep(100);
}
ADD_ALL.Visible = false;
}
private void Training_Form_Load(object sender, EventArgs e)
{
}
}
}
}

```

```

void FrameGrabber(object sender, EventArgs e)
{
    //Get the current frame form capture
    device
    currentFrame =
grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
    //Convert it to Grayscale
    if (currentFrame != null)
    {
        gray_frame =
currentFrame.Convert<Gray, Byte>();
        //Face Detector
        MCvAvgComp[][] facesDetected =
gray_frame.DetectHaarCascade(Face, 1.2, 10,
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_C
ANNY_PRUNING, new Size(20, 20));
        //Action for each element detected
        foreach (MCvAvgComp face_found in
facesDetected[0])
        {
            result =
currentFrame.Copy(face_found.rect).Convert<Gr
ay, byte>().Resize(100, 100,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
            result._EqualizeHist();
            face_PICBX.Image = result.ToBitmap();
            //draw the face detected in the 0th
(gray) channel with blue color
            currentFrame.Draw(face_found.rect,
new Bgr(Color.Red), 2);
        }
    }
}

```

```

        if (RECORD && facesDetected.Length > 0
&& resultImages.Count < num_faces_to_acquire)
        {
            resultImages.Add(result);
            count_lbl.Text = "Количество: " +
resultImages.Count.ToString();
            if (resultImages.Count ==
num_faces_to_acquire)
            {
                ADD_BTN.Enabled = true;
                NEXT_BTN.Visible = true;
                PREV_btn.Visible = true;
                count_lbl.Visible = false;
                Single_btn.Visible = true;
                ADD_ALL.Visible = true;
                RECORD = false;
                Application.Idle -= new
EventHandler(FrameGrabber);
            }
        }
        image_PICBX.Image =
currentFrame.ToBitmap();
    }
}
}

```