

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Асқаров Е.А.

Қол қимылдарын анықтау

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

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

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедра меңгерушісі

техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
"13"  2019ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: “ Қол қимылдарын анықтау”

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

Орындаған

Асқаров Е.А.

Ғылыми жетекші, лектор

 I.N. Ersari
"13"  05 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӨТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

БЕКІТЕМІН

ПИ кафедра меңгерушісі,
техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
"13" мамыр 2019ж.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға Асқаров Елдос Абзалбекұлы

Тақырыбы: Қол қимылдарын анықтау

Университет ректоры бұйрығының № 1162-б "16" қазан 2018 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі "14" мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

a) MNIST технологиясын қолдана отырып қимылдарды анықтау;
b) тұжырымдамаға сәйкес толыққанды оқытуға арналған ақпараттар қорын жинау;

в) бағдарлама дәлділігін дамыту, жақсарту;

г) бағдарламаны құру, қателіктерді анықтау және түзету, тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):

36 графикалық материалдар бар.

Ұсынылған негізгі әдебиеттер: 13 әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Кіріспе	12.12.2018	орындауға
2. Тапсырма қойылымы	15.01.2019	орындауға
3. Бағдарламалық қамтаманы жобалау.	08.02.2019	орындауға
4. Бағдарламаны әзірлеу	14.02.2019	орындауға
5. Бағдарламаның мүмкіндіктерін дамыту	15.03.2019	орындауға
6. Дипломдық жобаға түсіндірме жазба жазу	30.03.2019	орындауға

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

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

Ғылыми жетекші



I.N. Ersari

Тапсырманы орындауға қабылдап алған студент



E.A. Askarov

Күні

«1» қазан 2018 ж.

АҢДАТПА

Бұл жоба есту қабілеті төмен адамдардың қоршаған ортамен және де цифрланған технологиялармен қарым – қатынасты дамытуға арналған. Ұсынылған шешім ым тілінің әріптерін қимыл арқылы тануға мүмкіндік береді. Адамдар мен машиналар қарым-қатынасы, адамдар үшін табиғи қарым-қатынас деңгейі секілді жұмыс жасауы керек. Бұл дипломдық жобаның тақырыбы «Қозғалыстағы қимылдарды тану». Ақпараттық жүйе толықтай Python, OpenCV, Tensorflow, Keras технологияларын қолдану арқылы жасалды.

Дипломдық жоба құрылымы кіріспеден, екі бөлімнен, қорытындыдан және қосымшадан тұрады. Кіріспеде жобаның мақсаты, өзектілігі мен міндеттері көрсетіледі.

Бірінші бөлімде қазіргі таңдағы образдарды анықтау технологиялардың маңыздылығы, анықтау технологияларын адамдардың қарым-қатынас ортасында қолдану артықшылықтары, ақпараттық басқару жүйесінің классикалық моделі.

Екінші бөлімде Python бағдарламалау тілімен жұмыс, аталған бағдарламада модельденген диаграммалар, жобаның логикалық және бағдарламалық құрылымы туралы сипаттамалар берілген.

Құрастырылған ақпараттық жүйені жасау мен ендіру жолдары сәйкестендіріліп, жобада анық айтылған. Жобаны іске асыру барысында қойылған мақсатқа қол жеткізіліп, сәйкес мәселелері шешілді – «Қол қимылдарын анықтау» жүйесі құрастырылды.

Бұл мақалада цифрландыру мақсатында қабілеті төмен адамдарды қоршаған ортамен қарым – қатынасты автоматтандыру мәселелері қаралады. Жүйені дамыту үшін MNIST деректер қоры негізінде модельді оқыту қарастырылады, Tensorflow технологиясы арқылы модель негізінде болжау жасау мүмкіндіктері жасалған. PyCharm бағдарламалау ортасында Python бағдарламалау тілінде және көптеген қосымшалардың көмегімен әзірленген бағдарлама.

АННОТАЦИЯ

Этот проект предназначен для развития общения с людьми с нарушениями слуха и цифровыми технологиями. Предлагаемое решение позволяет распознавать язык жестов индивидуально. Отношения между людьми и машинами должны работать как уровень естественных отношений для людей. Тема этого дипломного проекта - «Признание движения». Информационная система была разработана технологиями Python, Opencv, Tensorflow, Keras.

Структура дипломного проекта состоит из введения, двух глав, выводов и приложений. Во введении указывается цель, актуальность и задачи проекта.

Первая часть посвящена важности современных технологий обнаружения изображений, преимуществам использования технологий в коммуникационной среде и классической модели управления информацией.

Во второй части, работа с языком программирования Python, смоделированные диаграммы в этой программе, описания логики проекта и структуры программного обеспечения.

Способы создания и внедрения скомпилированной ИТ-системы были определены и выделены в проекте. В ходе реализации проекта были достигнуты цели и решены соответствующие проблемы - разработана система «определения жестов».

Данная статья посвящена автоматизации общения с людьми с низким интеллектуальным потенциалом. Разработка системы основана на разработке базы данных MNIST, моделировании на основе технологии Tensorflow. Среда программирования PyCharm на языке программирования Python и программное обеспечение, разработанное для многих приложений.

ANNOTATION

This project is designed to develop communication with people with hearing impairments and digital technology. The suggested solution allows you to recognize the sign language individually. Relationships between people and machines need to work as the level of natural relationships for people. The theme of this diploma project is "Recognition of Movement." The information system was developed by Python, OpenCV, Tensorflow, Keras technology.

The structure of the diploma project consists of an introduction, two chapters, conclusions and annexes. In the introduction, the purpose, relevance and objectives of the project are indicated.

The first part focuses on the importance of modern technology for detecting images, the advantages of using technology in the communication environment, and the classical model of information management.

In the second part, work with Python programming language, modeled diagrams in this program, descriptions of project logic and software structure.

Ways to create and implement a compiled IT system have been identified and highlighted in the project. During the project implementation the goals were achieved and the corresponding problems were solved - a system of "hand gestures determination" was developed.

This article deals with automation of communication with people with low intellectual potential. System development is based on MNIST database development, modeling based on Tensorflow technology. PyCharm programming environment in the Python programming language and software developed with many applications.

МАЗМҰНЫ

	Кіріспе	8
1	Негізгі бөлім	9
1.1	Жүйеде қолданылатын ақпарат қоры (dataset)	9
1.2	Қол қимылдарың таңу жүйесінің әдістерін құру	12
1.2.1	Кескін таңу негіздері	13
1.2.2	Қимылды анықтау негіздері	15
1.2.3	Қол қимылдарын анықтау әдістеріне арналған теориялық материалдар	17
1.2.4	Қол қимылдарын анықтау әдісін құру	22
2	Құрылымдық - техникалық бөлім	30
2.1	Бағдарламалау ортасы	30
2.2	Жүйені толық тексеру	30
3	Экология және еңбекті қорғау	31
3.1	Жұмыс орнының талаптары	31
3.2	Жұмыс орнындағы жарық	33
3.3	Жүйенің қоршаған ортаға деген әсері	33
	Қорытынды	35
	Әдебиеттер тізімі	36
	А қосымшасы	37
	Б қосымшасы	39
	Ерекшеліктер	41

КІРІСПЕ

Жобаның өзектілігі. Адам-компьютермен өзара іс-қимыл әрекеттерін мүмкіндігінше жеңілдету керек. Адамдар мен машиналар қарым-қатынасы, адамдар үшін табиғи қарым-қатынас деңгейі секілді жұмыс жасауы керек. Адам табиғаты ақпарат алу үшін аралық құрылғыларды пайдаланбайды. Барлық ақпарат миға тікелей енеді, онда өңделеді және нәтиже беріледі. Машинаны өзара әрекеттесуді, суреттерді тануға, оларға жауап беруге және қажетті нәтижеге қол жеткізуге үйрету - бұл қазіргі кездегі ғылыми зерттеулер үшін танымал мақсат.

Болашақта образдарды динамикалық түрде шынайы және тез анықтау үшін көптеген зерттеу, амалдар құру жұмыстары жүргізіліп жатыр.

Өкінішке орай, бүгінгі күнге дейін адам-машиналық өзара әрекеттесу толықтай жұмыс істеуге қиындық тудыратын көптеген ақаулар бар: түсіру кезінде белгілі бір жарықтандыру жағдайларын қамтамасыз ету, жүйені әрбір операторға бейімдеу, сапасыз және жылдамдығы төмен анықтау жүйелері. Аралық жетістіктер бар, бірақ үш өлшемді датчиктер түрінде бейнекамералар және олардың күрделі аналогтары арқылы қол және саусақ қимылын танудың алгоритмдер сапасы төмен.

Енді басты міндет келесідей: жаңа және қарапайым әдістерді, динамикалық бейнелерді тану модельдері мен алгоритмдерін жасау, кейінірек адам мен машина арасындағы байланыссыз әрекет жүйелерін құру үшін пайдалана алу.

Жоба мақсаты. Жүйенің мақсаты ым тіліне негізделген қол қимылын анықтау. Мақсатқа жету барысында келесідей тапсырмалар орындалады:

- қадағалау мен үлгіні тану (қолдың қимылдары) үшін әзірленген және қолданылатын әдістерді талдау, сондай-ақ қашықтықтағы адам-машиналық өзара әрекеттесу;

- қолдың конфигурациясын сипаттау, алу және тану әдістерін зерттеу мен әзірлеу.

Ғылыми жаңалығы. Кескінді тану жүйесі тірі организмдердегі сияқты тиімді болуы керек.

Осы диссертацияда осындай идеяны іске асыру үшін орташа WEB камера көмегімен динамикалық объектіні тану арқылы, ым тілінің қимылдарын тану үшін балама әдістер жасалды.

Кеңістіктегі динамикалық объектілерді анықтауға арналған қарапайым алгоритмдер әзірленді.

Теориялық маңыздылығы. Жұмыстың теориялық маңыздылығы адам-компьютерлік өзара әрекеттесудің жоғары сапасын қамтамасыз ететін қол қимылдарын алу және тану әдістерін және алгоритмдерін әзірлеуде қажетті қасиеттер мен абсолютті жиынтығы бар объектілерді сыныптау және анықтау әдістерін әзірлеу болып табылады.

1 Негізгі бөлім

Адамзат тарихында бұрын-соңды болмаған күнделікті өмірде елестете алмайтын бірнеше ашылымдар болған. Бұл жаңалықтар, әдетте, адам өмірінің сапасын жақсартуға бағытталған. Адамзаттың қарым-қатынасын жақсарту үшін жазу үлгісі ойлап табылды. Адамды ұзақ қашықтыққа жылжыту мүмкіндігін доңғалақ бере алды. Ғылыми прогрестегі нақты серпіліс - электр энергиясы. XX ғасыр көптеген жаңалықтардың ғасыры болды. Ең қызығы, қазір адамзат осының бәрін болуы тиіс секілді қабылдайды. Адамның миы сан алуан ойларға толы, сол себепті қол жеткізілген нәтижеге тоқталып қалмас.

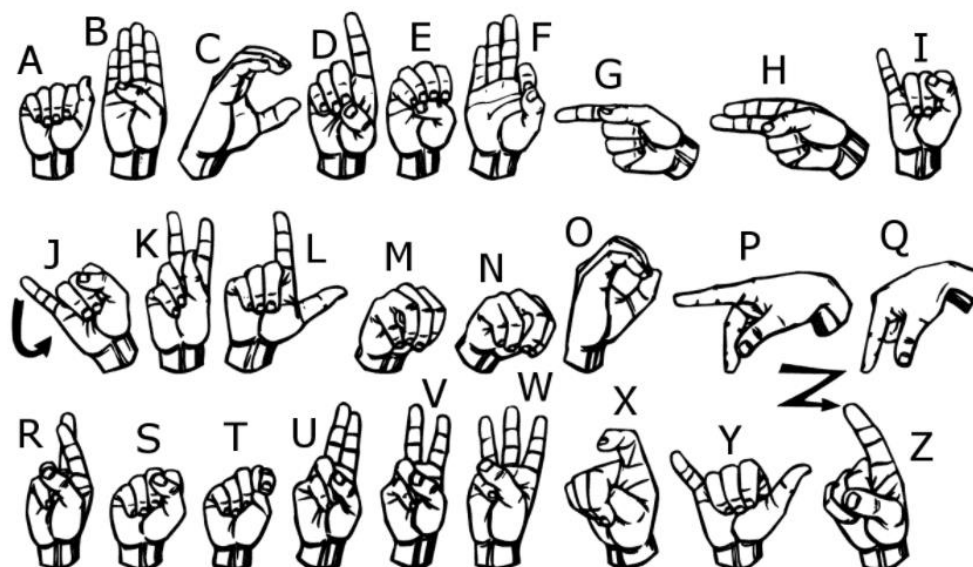
Есту қабілеттері төмен немесе мүлдем естімейтін адамадардың басқа адамдармен қарым қатынасын қарастырайық. Бұл мәселені бір жақты тарапта ым тілі шешіп тұр. Алайды жер шарымызда ым тілін білетін және білмейтін адамдардың сандық қатынасы өте жоғары. Сол себепті адам – компьютер – адам қатынасын құрып, есту қабілеті төмен адамдардың қоршаған ортамен байланысуына жаңа серпіліс бере аламыз.

1.1 Жүйеде қолданылатын ақпарат қоры (dataset)

Бастапқы MNIST қолжазбалық цифрланған кескін жиынтығына негізделген компьютерлік оқыту әдістеріне арналған танымал анықтама болып табылады, бірақ зерттеушілер компьютердің көруі үшін неғұрлым күрделі және түпнұсқа қосымшалар үшін түпнұсқалық алмастыруларды жаңартуға және дамытуға тырысады. Fashion-MNIST деректер жиынтығы деп аталғандай, Заландо зерттеушілері «MNIST сандарының көп жұптарын (бір үлгіге тек 784 пиксель) бір пиксель арқылы өте жақсы ажыратуға болады» деп айтқан болатын. Қосымша дамытуларға қауымдастықты ынталандыру үшін, мұнда белгілер және әрбір жолдағы пиксель мәндері бар сол CSV форматындағы ым тілінің MNIST ақпарат қоры ұсынылған. Американдық қолтаңба тілі - 24 класстағы әріптермен қамтылған.

конвейер, пайдаланылған жаңа кескін деректерін жасау үшін. Түрлендіру және кеңейту стратегиясы 5% кездейсоқ пиксельдендіру, +/- 15% жарықтылық, контраст және сүзгі (Mitchell, Robidou, Katrom, Spline, Hermite) дәрежесі Кескіндердің кішкентай өлшемдеріне байланысты, бұл өзгертулер қызықты, басқарылатын жолдардағы ажыратымдылық пен ажыратымдылықты тиімді түрде өзгертеді.

Бұл деректер жиынтығы Fashion-MNIST 2-тен және Sreehari-4 компьютерлік оқыту құралдарынан дем алған.



8-сурет – ASL ым тілі

Көрнекі түрде тану алгоритмі жаңа конструктивті нейрондық желілер сияқты қазіргі заманғы компьютерлік оқыту әдістеріне тап болатын жаңа сынақтарды ғана емес, сондай-ақ, саңырау және есту қабілеті нашар адамдарға компьютерлік көру бағдарламалары арқылы жақсы қарым-қатынас жасауға көмектеседі. Ауыртпалық және басқа да байланыс бұзылуларының ұлттық институты (NIDCD) 200 жастағы американдық ым тілін толық, әрі күрделі тіл (оның әріптік қимылдар бөлігі ғана) болып табылады, бірақ Солтүстік Америкадағы көптеген саңырау адамдарға арналған негізгі тіл болып табылады. ASL - Америка Құрама Штаттарында «Үлкен төрттік» жетекші тілдер: испан, итальян, неміс және француз тілдерінен кейінгі тарапта тұр.



9-сурет – Класстарға бөлінген суреттер

1.2 Қол қимылдарың таңу жүйесінің әдістерін құру

Қазіргі қоғамда ақпаратқа үлкен рөл беріледі. Ақпаратқа деген қажеттілік жоғары деңгейге жетті, адамдар үздіксіз - ақпарат алмасумен айналысады. Сондықтан оны алудың, өндеудің, ақпаратты жүйелеудің жоғары жылдамдығы қажет. Ақпаратты беру процесінде пайда болатын ақпаратқа қол жетімділік проблемасы бар, оны жоғалту қаупі бар.

«Ақпаратқа қолжетімділік» термині пайдаланушының жеке қажеттіліктерін ескерместен ақпаратқа рұқсатсыз кіру мүмкіндігі. Жеке мұқтаждықтар: қолданылған аппараттық құралдардың, бағдарламалық жасақтаманың немесе пайдаланушының жеке қабылдауларының ерекшеліктері болуы мүмкін. Ақпаратты орналастыру ортасы ақысыз ақпаратты жүзеге асыру үшін пайдаланушының жеке қажеттіліктерін әрдайым қанағаттандырмайды.

Сөйлейтін технологияларды, суреттерді және бейнені дамыту арқылы адамның дербес компьютермен өзара қарым-қатынасы басқа деңгейге ауысады. Жақында адаммен компьютердің өзара әрекеттесуі кеңейтіліп, адаммен адамның байланысымен ұқсас байланыс орнату. Пайдаланушы интерфейстеріне мыналар жатады: дыбыс, сенсор, қимыл, өзара әрекеттесу. Барлық өзгерістер мен жетілдірулер адамның сезімін тиімді пайдалануға қол жеткізуге бағытталған.

Адамдар үшін қалыпты өзара әрекеттесу бірдей болмауы керек. Мұндай жағдайларда барлық қатысушылар үшін айқын және логикалық байланыс қамтамасыз ететін коммуникативті байланыс болып табылатын перцептуалды есептеу негізіндегі құралдар қолданылады.

Төменде ыммен, әдістермен және алгоритмдермен жұмыс істейтін қолданыстағы технологиялар талқыланады және мүмкіндіктері шектеулі адамдармен сыртқы әлеммен - автоматты ымдау тілі жүйелерінің өзара әрекеттесу шекараларын жою үшін қолданылатын, ұсынатын нақты мүмкіндіктер қарастырылады.

1.2.1 Кескін таңу негіздері

Біздің бүкіл өміріміз кескіндерді тануға негізделген. Бұл мәселені шешпесек, біздің өміріміз өте қиын болар еді. Біздің миымыз, жүйке процестерінің саны онның он дәрежесіне тең, әр секундта қарапайым заттардан - объектілерден, сигналдардан және бізде пайда болатын құбылыстармен, үрдістермен және жағдайлармен аяқталатын барлық ақпаратты өңдеуге жұмыс жасайды.

Бұл мидың жұмысы бірден-бір нәтижеге бағытталған - шешімді қабылдау және қажетті әрекетті орындау. Бұл ойлау процесін іске асыру кезінде барлық сезімдер қатысады, олардың көмегі арқылы біз қоршаған айналамыздағы шындық туралы ақпарат аламыз. Одан кейін орталық жүйке жүйесі ағындарды өңдеуге байланысты, шешім қабылдау үшін және шешімді іске асыру мақсатында адамның физиологиялық қабығы іске қосылады. Барлық осы жүйені қосу үшін жалғыз ғана кескін болып табылады.

Біздің миымыз керемет жаратылыс. Көптеген параллель процестерді бақылау мүмкін емес жылдамдықпен бақылайды. Автокөлікті тығыз және карама – қарсы бағытта келе жатқан автокөліктерді ескере отырып, жүргізушінің азғана секундтар аралығында шешімдер қабылдап және сол шешімдерді жүзеге асырулары.

Қазіргі кезеңдегі есептеу техникасының дамуы, машинаны кескінді тану мәселелерін шешуге үйретуге болады деп болжауға болады. Электрокардиограммалардың жіктелуінде кейбір прогресс жасалды, саусақ іздері бар дерекқорлар жасалды және бастапқы сынама мен қолда бар үлгі салыстыру негізінде машина адамды анықтай алады. Фотосуреттер, сөйлеу, жазба және қолжазба мәтіндері туралы да айтуға болады. Көрсетілген параметрлерге сәйкес бағдарлама ақауларды анықтау үшін күрделі құрылғылар мен жүйелердің диагностикасын жүргізе алады.

Объектінің қасиеттерін тану, сурет, дыбыс жазу, объектіге тән өзге де сипаттамалар арқылы сәйкестендіру - бұл үлгіні танудың семантикалық жүктемесі. Барлығы - құрылымдық топтарда анықталатын және біріктіруші сипаттамаларға сәйкес объектілерді жіктеу. Маргиналды минимумға дейін сығылған құбылыстардың барлық түрлерін қоса алғанда, тән қасиеттердің алдын ала анықталған үлгілерін қалыптастыру, суреттердің кез-келген санының мәнін оқшаулауға мүмкіндік береді. Үлгіні танудың үш әдісі, қарау үшін ең қызықты: іздеу әдісі, сурет сипаттамаларын талдау, жасанды нейрондық желілерді.

Іздеу әдісі. Белгілі бір деректер массиві бар. Мысалы, әртүрлі кеңістіктегі бағдар мен тұтастық нысанды сурет немесе қаріптер жиыны, сөз тіркестері. Бағдарлама көмегімен қолда бар ақпаратпен салыстыру жүргізіледі, нәтижесінде алынған кескін таңылады.

Екінші әдісте суреттің сипаттамаларына терең талдау жүргізіледі, осы әдіс негізі (оптикалық тану) объектінің геометриялық сипаттамаларына байланады.

Дыбысты тану кезінде жиіліктер мен амплитудаларды талдау қажет.

Егер суреттің жоғары сапасын немесе объектіні өзгерту ықтималдығын қамтамасыз ете алмасаңыз, сипатталған әдіс өзектілігін жоғалтады.

Жасанды нейрондық желілерді пайдалану (ННН). Егер жалпы жүйедегі нейрондық желіні сипаттайтын болсақ, бұл нақты тапсырманы шешу сәтінде миға ұқсас ақпаратты өңдейтін машина. Бұл әдіс негізінде жатқан негізгі идея - мидың иірімділігі, кіріс туралы ақпаратты өңдеу - маңызды рөл атқарады. Адам миында шексіз саны бар нейрондар құрылады және олар бір-бірімен байланысты, жасанды нейрондық желілерде есептеулерді жүргізетін және олардың арасындағы байланысты ұйымдастыратын көптеген жасушаларды жасау қажет. Жасанды нейрондық желілер, алдымен оны өңдейтін ақпаратты алған аккумуляторлар болып табылатын қарапайым бірліктер жиынтығы. Жасалған процесті жүзеге асыру үшін әзірлеушілер, оқыту алгоритмін тағайындайды. Нейронды өзара байланыстыру үшін нейрондық желінің синоптикалық байланысын ұйымдастыру процедурасын жетілдіруге іс - шаралары жүргізіледі.

Жасанды нейрондық желілік әдістердің жұмыс істеуін қамтамасыз ету үшін оқыту барысында немесе нақты тапсырманы орындау үшін нақты құрылымдалған нейрондық желіні тану проблемасының үлгілік негіздері қажет.

Белгіленген әдісті жұмысқа кіргізу үшін әсерлі дайындық жұмыстарын жүргізу қажет болғанымен, қазіргі уақытта ең тиімді және өнімді болып табылады.

Тану міндеттерінің ең тән ерекшеліктерін келтіреміз. Бұл екі кезеңнен тұратын ақпараттық міндеттер:

- 1) түпнұсқа деректерді тануға ыңғайлы пішінге айналдыру;
- 2) тиісті тану (объектінің белгілі бір сыныпқа тиесілі екенін көрсету).

Тану мынадай міндеттерді шешеді:

- 1) тану - берілген сыныптың біріне оның сипаттамасына сәйкес ұсынылған объектіні беру;
- 2) автоматтандырылған жіктеу - бөлінбейтін сыныптар жүйесіне (таксономия, кластерлік талдау, өзін-өзі зерттеу) сипаттама, объектілер жиынтығын, жағдайларды, құбылыстарды бөлу;
- 3) тану ерекшеліктерінің жиынтығын таңдау;
- 4) болжау.

1.2.2 Қимылды анықтау негіздері

Адамдар сөйлесу қабілеті арқылы бір-бірімен сөйлеседі, сондай-ақ қимылдарды қолданады. Егер біз күнделікті өмірдегі бұл қимыл қажеттігін талдайтын болсақ. Бір жағынан, адамдар сөйлеудің айқындығы мен эмоционалдылығын көрсету үшін қимылдарды қолданады. Екінші жағынан, сөйлеу қабілетінің бұзылыстары бар адамдар санаты бар, олар үшін қимыл қажетті талап.

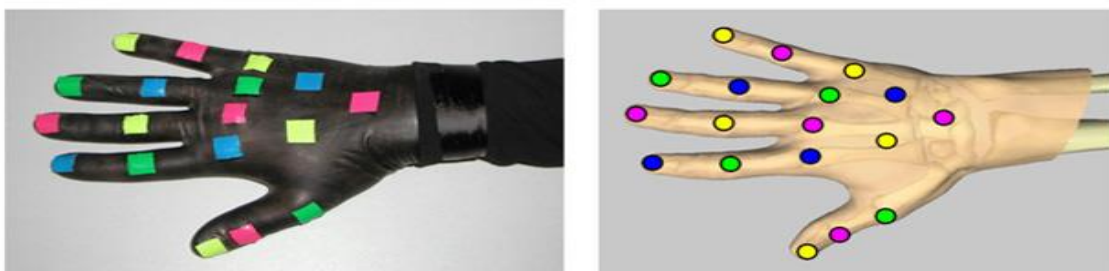
Қимыл ретінде жіктелетін әр түрлі қимылдар бар. Пішіні мен семантикасы бойынша өте әртүрлі болатын ымдау тілін толық зерттеу адам миы үшін де өте ауыр. Сондықтан, қазіргі кезде автоматты түрде тану міндеті генерацияланған дерекқорда алдын ала жазылған қимылдарды анықтауға негізделген.

Әртүрлі қимыл сыныптарын танудың көптеген әдістерін және тәсілдерін сипаттайды.

Әдістер мен әдістерді жіктеу, енгізу деректеріне және қолмен қабылдау үшін қолданылатын датчиктерге сәйкес жүзеге асырылуы мүмкін. Тану әдістерін және тәсілдерін жіктеу:

- 1) маркерлерге негізделген қозғалысты қабылдау;
- 2) қолғап арқылы;
- 3) көрнекі қимылдарды тану әдістері:
 - 3.1) кірістің кескі арқылы қол моделін құру;
 - 3.2) қолдың кіріс кескінінің сипаттамасын құру;
 - 3.3) метрикалық қимылдарды тану әдістері.

Маркерлерге негізделген қозғалысты анықтау. Маркерлік жүйені пайдалану үшін сізге арнайы жабдық: сенсорлар, камералар қажет.



10-сурет – Маркерлерге негізделген қозғалысты анықтау

Датчиктер адамның қолына бекітіледі, қолдың кез-келген қозғалысы камералар көмегімен жазылады. Жиналған ақпарат ағыны компьютерге енеді. Берілген ақпаратты өңдеу және талдаудан кейін, адамның кескіні негізінде анимация ретінде жазылған адамның қимылдарын дұрыс көрсететін үшөлшемді толық үлгі пайда болады.



11-сурет – Маркерлерге негізделген қозғалысты анықтау нәтижесінде пайда болған кескін

Жоғарыда аталған жүйелерде пайдалану ыңғайлылығына тән артықшылықтар бар, кемшіліктер көп камералы қондырғының жоғары құны.

Бұл әдістің алдыңғы жағында экзоскелетикалық құрылғыларда дайындалған және адам қолының биомеханикасын қайталауға мүмкіндік беретін әртүрлі сенсорлармен (температура, діріл, қысым және т.б.) жабдықталған қолғаптарды қолдану болып табылады. Қолдардың қозғалысы туралы ақпарат сенсорлардан оқылады және компьютерге өңдеу үшін беріледі. Алынған ақпарат кейін талданып, бейнелерге түрлендіріледі. Жеке кескіндердің сипаттамалары туралы деректер жүйе үлгілері ретінде пайдаланылады.



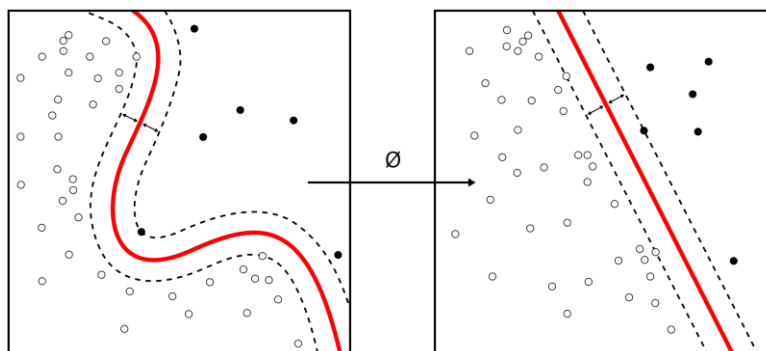
12-сурет – Экзоскелетикалық қолғап

Көрнекі қимылдарды тану. Қазіргі уақытта визуалды қимылдарды тану әдістерін зерттеу саласы ең қарқынды дамыған. Негізінде, бұл әдіс - қолды қадағалау үшін. Қосымша жабдықты қажет етпейді. Қолдың күйін анықтау үшін түс, сурет, контур сияқты сипаттамалар қолданылады.

Әзірге бұл сигналдар жалпы және әр түрлі жарық жағдайларына төзімді болып табылады, яғни олар қолдың көп өлшемді кеңістіктік тетіктерін іздейтін күрделі алгоритмдерді анықтауды талап етеді.

1.2.3 Қол қимылдарын анықтау әдістеріне арналған теориялық материалдар

CNN моделі. 1988 жылы Ян Лекун ұсынған жасанды нейрондық желілер (CNN) - бұл терең оқу технологиясының бір бөлігі. Түрлі бұрыштардан қарау және қарапайым жасушалардың белгілі бір жиынтығын белсендіруге реакциясы бар, күрделі клеткалар түзу сызықтарға әрекет ететін қарапайым жасушалар деп аталатын визуалды қыртыстың кейбір ерекшеліктерін анықтайды. Осылайша, конволюциялық нейрондық желілер идеясы конволитациялық қабаттардың және іріктемелі қабаттардың ауысу арқылы жұмыс істету болып табылады. Желі құрылымы бір жақты (кері байланыссыз) және көп қабатты. Стандартты әдістер оқыту үшін қолданылады, көбінесе қателерден артылу әдісі. Нейрондардың активтендіру функциясы (трансфер функциясы) - зерттеушінің таңдауы бойынша.



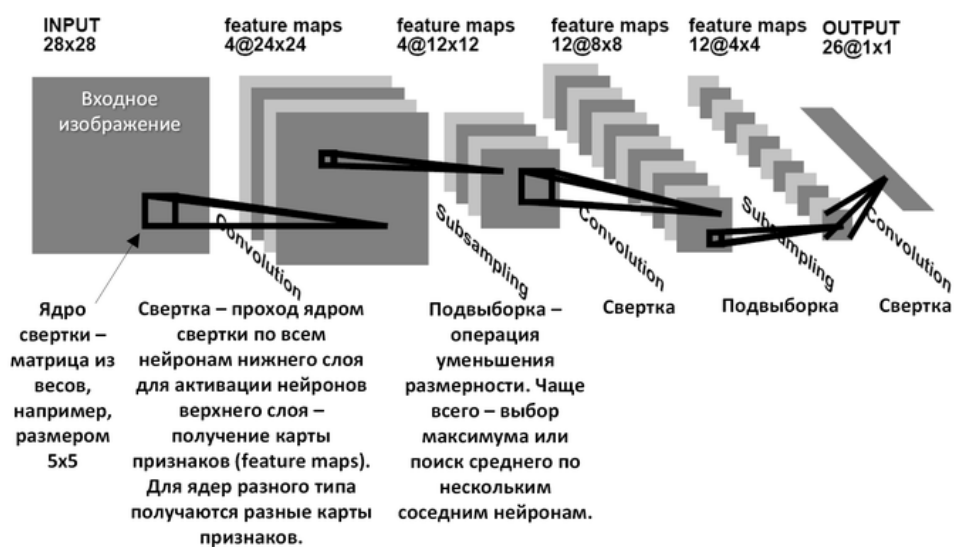
13-сурет – Машинаны оқыту және data mining

Желі архитектурасы, өз кезегінде, әрбір сурет фрагментің матрицаның элементіне көбейтіліп, нәтижелер қосылып, жинақталып шығыс кескінінің сәйкес орнына жазылғандықтан, конвульсиялық операцияға байланысты өз атын алды.

Түсіндірме. Конволитациялық нейрондық желінің жұмысы, әдетте, суреттің нақты ерекшеліктерінен көбірек дерексіз бөлшектерге көшу, әрі жоғары дәрежелі ұғымдарды анықтауға дейін көбірек дерексіз мәліметтерді түсіндіреді. Сонымен қатар, желі өздігінен реттеледі және абстрактілі ерекшеліктердің иерархиясын жасайды, маңызды емес мәліметтерді сүзеді және маңыздылығын бөліп көрсетеді.

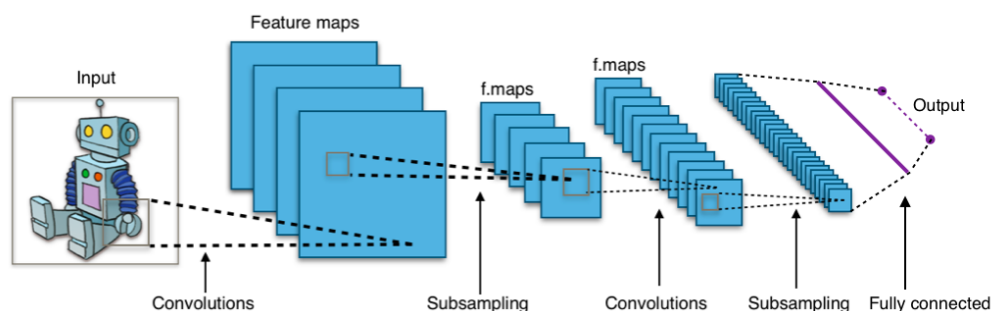
Мұндай түсіндіру метафоралық немесе иллюстративті болып табылады. Іс жүзінде, күрделі желі арқылы шығарылған «белгілер» түсініксіз және қиын түсіндіріледі, практикалық жүйелерде бұл белгілердің мазмұнын түсінуге немесе оны түзетуге тырысу ұсынылмайды, оның орнына жақсы нәтиже алу үшін желі құрылымы мен сәулетін жақсарту ұсынылады. Осылайша, жүйенің кез-келген маңызды құбылыстарына назар аудармау, оқытудың жеткіліксіздігі немесе желілік құрылымның кемшіліктері бар екендігін көрсетуі мүмкін.

Сәулеті мен жұмыс принципі. Нейрондық, толығымен байланысқан әрбір нейрон алдыңғы қабаттың барлық нейрондарына қосылған, әр қосылыс өзінің жеке салмақ факторына ие. Конволитациялық нейрондық желіде кішігірім, конвульция жұмысында, шектеулі салмақтардың матрицасын өңделіп жатқан қабатта «жылжыту» арқылы пайдаланылады, әрбір ауысымнан кейін келесі деңгейдегі ұқсас орындағы нейронды іске қосу сигналын қалыптастырады. Яғни, шығу қабатының әртүрлі нейрондары үшін салмақтың бірдей матрицасы пайдаланылады, ол сондай-ақ конволюция ядросы деп аталады. Ол белгішенің графикалық кодтауы ретінде түсіндіріледі, мысалы, белгілі бір бұрышқа көлбеу сызықтың болуы. Содан кейін, осындай матрицалар салмағы бар конвульсиялық операциядан туындайтын, өңделетін қабаттағы, бұл мүмкіндікті және оның координаттарын көрсетеді, бұл функционалды карта деп аталады. Сонымен қатар, мұндай конверсия ядролары зерттеушіге алдын ала қойылмайды, бірақ желіні қате таратудың классикалық әдісін оқыту арқылы дербес түрде қалыптасады. Таразылардың әрқайсысын өту функционалды картаның өзіндік көшірмесін жасайды, нейрондық желіні көп арналы жасайды. Сонымен қатар, қабат салмақ матрицасы арқылы өлшенгенде, әдетте ол толық қадаммен емес, кішкене қашықтықта жылжытылады. Мысалға, егер 5×5 салмақ матрицасының өлшемі болса, қалаған функцияны «басып өту» үшін бес орнына бір не екі нейрон ауыстырылады.



14-сурет – Конволитациялық нейрондық желінің сәулеті

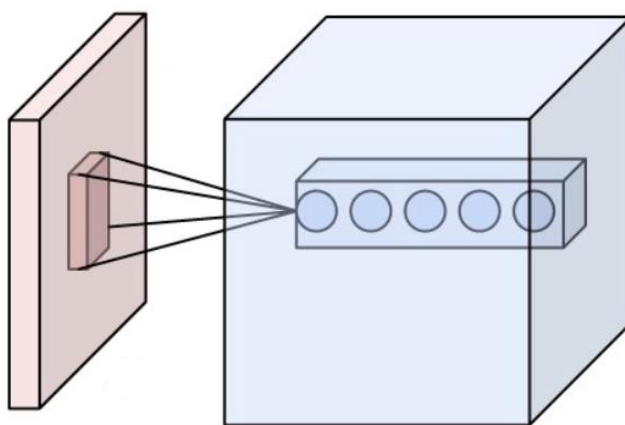
Субдискретизация операциясы генерацияланған ерекшеліктер картасының өлшемін азайтады. Осы желілік архитектурада қажетті функцияның болуы туралы ақпарат оның координаттарын дұрыс білуден гөрі маңызды деп есептеледі, сол себепті мүмкіндігінің ең көбі функционалды картаның бірнеше көршілес нейрондарынан таңдалады және кішірек өлшемділік ерекшелігі картасының бір қысылған нейроны ретінде қабылданады.



15-сурет – Конволитациялық нейрондық желінің сәулеті түрі

Конволитациялық нейрондық желінің типтік құрылымын егжей-тегжейлі қарастырайық. Желі көптеген қабаттардан тұрады. Бастапқы қабаттан (кіріс сурет) кейін, сигнал конвективті қабаттардың сериясын өткізеді, онда конволюция және іріктеу (біріктіру) ауыстырылады. Қабаттардың ауысуы ерекшеліктер карталарын жасауға мүмкіндік береді, әр кейінгі қабатта карта өлшемі азаяды, бірақ арналардың саны артады. Іс жүзінде, бұл күрделі ерекшеліктер иерархиясын тану мүмкіндігін білдіреді. Әдетте, бірнеше қабаттан өткеннен кейін, функция картасы векторға немесе скалярға айналады, бірақ жүздеген осындай ерекшелік карталары бар. Желілік қабаттарының шығыс ақпараты одан әрі болатын қабаттың кіріс белгілері картасына қызмет етеді, онда бірнеше қабаттары толығымен қосылған нейрондық желі, құру қамтылады.

Конволитациялық қабат конвективті нейрондық желінің негізгі бірлігі болып табылады. Конвульция қабаты әрбір арна үшін сүзгіден тұрады, оның фрагменттері бойынша алдыңғы қабатты өңдейтін конвульциялық ядросы. Конвульция ядросының салмағы белгісіз және оқу үдерісінде белгіленеді.



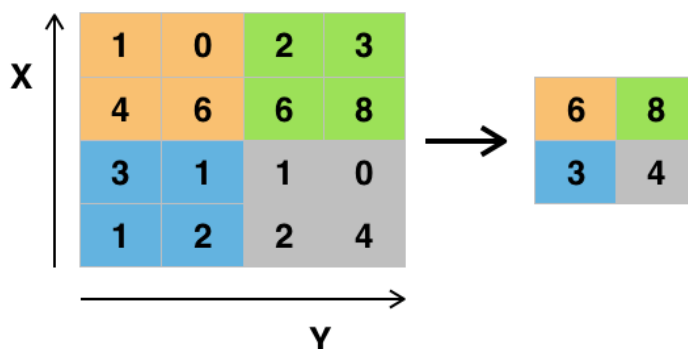
16-сурет – Конволитациялық қабат нейрондары

Конволитациялық қабаттың ерекшелігі - жаттығу кезінде белгіленген параметрлердің салыстырмалы түрде аз саны. Мысалы, егер түпнұсқалық кескін үш арнада 100×100 пиксель өлшемі болса бұл 30,000 кіріс нейронды білдіреді және конволитациялық қабатта 6 каналды шығатын 3×3 пиксель ядросы бар

сүзгілер қолданылса, онда оқу процесінде тек 9 негізгі салмақ анықталады, барлық арналық комбинациялар, яғни $9 \times 3 \times 6 = 162$, бұл жағдайда бұл қабат тек 162 параметрді табуды талап етеді, бұл толығымен байланысты нейрондық желінің қажетті параметрлерінің санынан айтарлықтай аз.

Әрбір конвульцияның скалярлық нәтижесі кейбір сызықты емес функция болып табылатын активация функциясына келеді. Активация қабаты, әдетте, қисынды қабатпен біріктіріледі. Сызықты емес функция зерттеушілердің таңдауы бойынша кез-келген болуы мүмкін, дәстүрлі түрде осы мақсатта гиперболалық тангенстік түрдегі функциялар $f(x)=\tanh(x)$, $f(x)=|\tanh(x)|$ немесе $f(x)=(1+e^{-x})^{-1}$ қолданылады. Алайда, 2000-шы жылдары жаңа оқу процесін белсендіру функциясы (Rectus Linear Unit) үшін ұсынылды және зерттелді, бұл оқу үдерісін едәуір жеделдетті және сонымен қатар жеңілдетілген есептеулерді $f(x)=\max(0,x)$ функциясын есептейтін сызықты түзету бірлігін білдіреді. Яғни, мәні бойынша, бұл скалярдың теріс бөлігін кесу операциясы. 2017 жылғы жағдай бойынша бұл функция және оның модификациясы (Noisy ReLU, Leaky ReLU және басқалары) терең нейрондық желілерде, атап айтқанда конволитациялық желілерде жиі қолданылатын белсендіру функциялары болып табылады. Сызықты түзету блоктарының оңтайлы санын анықтаудың әдісі бар.

Біріктірудің қабаты (басқаша субгумуляциялау, субтабляциялау) сызықтық емес трансформациядан өтіп, бір пиксельге дейін қысылған (әдетте өлшемі 2×2) пиксель тобы бар атрибут картасының сызықты емес қысымы. Максимум функциясы ең жиі пайдаланылады. Трансформациялар әрқайсысы бір пиксельге қысылған және максималды мәні бар пиксель таңдалмаған қиылыспайтын төртбұрыштарға немесе квадраттарға әсер етеді. Біріктіру операциясы, кескіннің кеңістіктік көлемін едәуір азайтады. Біріктіру келесідей түсіндіріледі: егер бұрынғы конвульсиялық операцияда кейбір белгілер анықталған болса, онда одан әрі өңдеу үшін мұндай егжей-тегжейлі кескін қажет емес және ол аз детальді болады. Сонымен қатар, қажетсіз бөліктерді сүзуді қайтадан оқытуға болмайды. Біріктіру қабаты, әдетте, келесі қабықтың қабатының алдындағы қабырғаның қабатынан кейін енгізіледі.



17-сурет – Біріктірудің қабаты, 2×2 қадамымен ең үлкен мән функциясы

Максималды функциямен біріктіруден басқа, басқа функцияларды пайдалана аласыз - мысалы, орташа мән немесе L2-бағалау. Дегенмен, тәжірибе,

типтік жүйелерге кіретін максималды функциямен біріктірудің артықшылықтарын көрсетті.

Алынған нәтижелердің мөлшерін неғұрлым агрессивті түрде төмендету үшін кішірек сүзгілерді қолдану идеясы немесе біріктіру қабаттарын толық қабылдамау идеясы кеңінен тарайды.

Біріктірілген жүйені пайдаланып, суреттің икемділігі мен қысылуын бірнеше рет өтегеннен кейін, жүйе жоғары ажыратымдылығы жоғары абсолютті сипаттамалар картасына дейін бетон торынан қалпына келтіріледі, әдетте, әрбір келесі қабатта арналардың саны артып, әр арнаның сурет өлшемі азаяды. Ақыр соңында, түпнұсқалық кескіннен анықталған ең дерексіз ұғымдар ретінде түсіндірілетін деректердің аз мөлшерін (тіпті бір параметр) сақтайтын арналардың үлкен жиынтығы қалады.

Бұл деректер біріктіріліп, бірнеше қабаттан тұратын әдеттегі толығымен байланысқан нейрондық желіге беріледі. Бұл жағдайда толығымен қосылған қабаттар пикселдердің кеңістіктік құрылымын жоғалтады және салыстырмалы түрде кішкентай өлшемге ие (бастапқы кескіннің пикселдер санына қатысты).

Ең қарапайым және ең танымал оқыту әдісі - мұғаліммен (белгіліленген деректер бойынша) оқыту әдісі - қатені қайта тарату әдісі және оны модификациялау. Сонымен қатар, бірқатар оқытушылардан тыс білім беру технологиялары бар. Мысалы, Гофрдың операция сүзгілерін бақыланбайтын оқыту (мысалы Autoencoder немесе тіпті K-құралдары) үшін кез келген белгілі алгоритм, оларды оқыту үлгідегі бастапқы суреттердің кездейсоқ бөліктерін кесетін және олар үшін өтініш бере отырып, жеке және дербес оқытылған болуы мүмкін - patch-based training техникасына негізделген оқыту. Тиісінше, желілік конволюцияның келесі қабаты желінің қазірдің өзінде дайындалған бірінші қабатынан дайындалады. Сондай-ақ, конвективті нейрондық желіні басқа терең білім беру технологияларымен біріктіруге болады.

Кездейсоқ бір нейрондарды лақтыру әдісі ішкі желі оқыту - желілік өнімділігін жақсарту, оның тұрақтылығын арттыру және қайта оқытуды алдын алу үшін қолданылады.

Артықшылықтары:

- толық-нейрондық желі (перцептрон түрі) суреттің әрбір пикселін сурет үшін қолданғаннан гөрі теңшелетін салмақ саны аз, суретті толық пайдаланған тиімді;

- суретті тану мен классқа бөлудегі ең пайдалы алгоритм;

- есептеулердің ыңғайлы параллелизациясы, графикалық процессорларға алгоритмдерді және желілік оқытуды енгізу мүмкіндігі;

- көрінетін кескіннің айналуына және жылжуына салыстырмалы қарсылық;

- қатені қайта таратудың классикалық әдісімен танысу.

Кемшіліктері:

- айнымалы желілік параметрлері өте көп; Қандай тапсырма және есептегіш қуаттылық қандай параметрлер қажет екендігі анық емес. Осылайша, айнымалы параметрлер үшін мыналарды қамтиды: өлшем азайту қабаттарының

санын, қабаттарының әрбірі бойынша Гофрдың өлшемі, қабаттарының әрбірі үшін ядролардың саны, өңдеу қабатының қадам ығысу қабаттарының субдискретизациясы қажет, функцияның мөлшерін төмендету дәрежесі (ең ауқымды, орта және т.б.), нейрондардың берілу функциясы, конволюцияның шығуындағы толық байланысты нейрондық желінің болуы және параметрлері. Барлық осы параметрлер нәтижеге айтарлықтай әсер етеді, бірақ олар эмпирикалық түрде зерттеушілермен таңдалады. Бірнеше тексерілген және өте жақсы жұмыс істейтін желілік конфигурациялар бар, бірақ жаңа тапсырма үшін желіні құру бойынша ұсыныстар жеткіліксіз.

1.2.4 Қол қимылдарын анықтау әдісін құру

CSV файлындағы оқытуға және тестілеуге арналған ақпараттарды жүктеп аламыз.

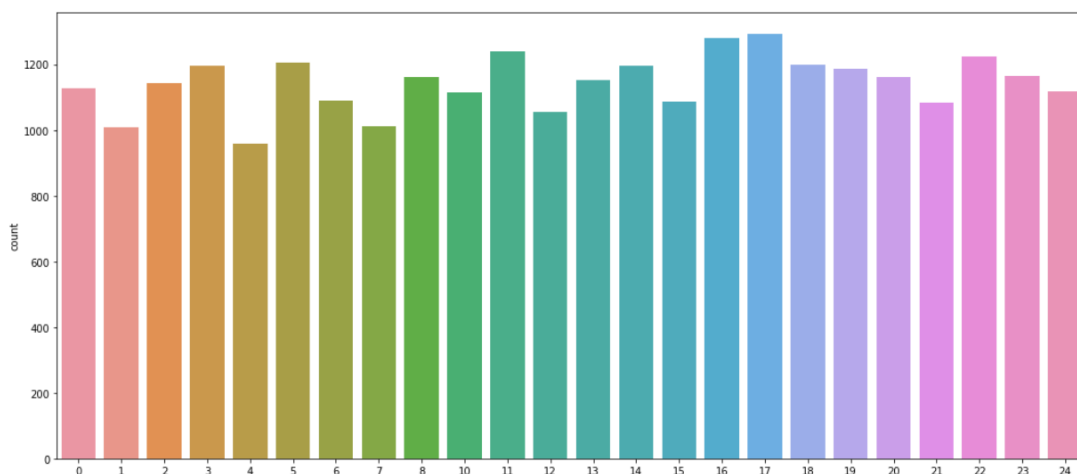
```
train = pd.read_csv('sign_mnist_train.csv')
```

```
test = pd.read_csv('sign_mnist_test.csv')
```

Модельді үйренуге кіріспес бұрын деректер жиынтығындағы деректер пішімін қарастырайық. Shape көмегімен біз деректер жиынтығы көлемінің 60 000 бейнесін 28 x 28 пиксельден тұратынына көз жеткізе аламыз:

```
test_images.shape
```

```
(7172, 28, 28, 1)
```

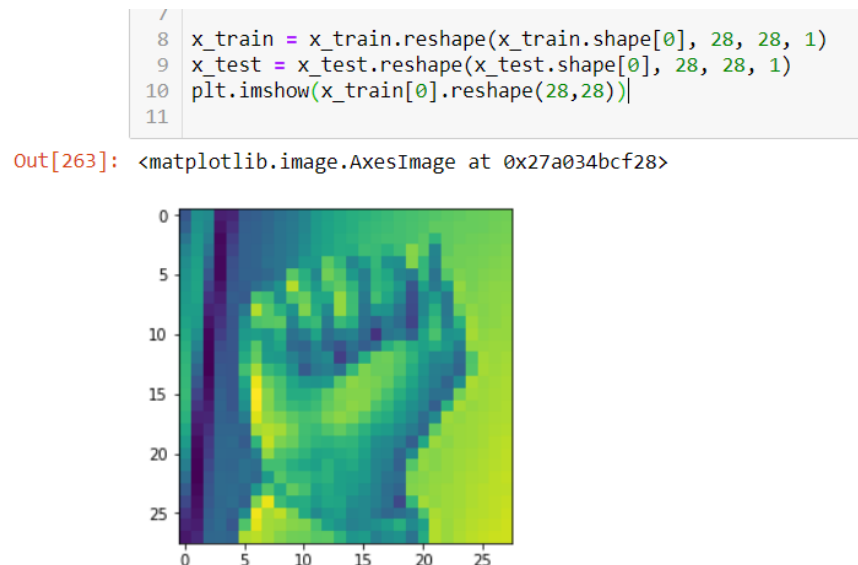


18-сурет – Класстарға бөлінген суреттер диаграммасы

Барлық деректерді нейрондық желіні үйретуге дейін бірдей түрде дайындау және өңдеу керек. Егер жаттығу жиынынан бірінші кескінді қарастырсақ, пиксель мәні 0-ден 255-ге дейінгі диапазонда екенін көреміз.

Нейрондық желіні үйренуден бұрын, бұл мәндерді 0-ден 1-ге дейін ауқымға бөлу керек. Ол үшін суреттің деректер түрін бүтін сандардан өзгермелі сандарға түрлендіріп, 255-ке бөліңіз.

Тренинг пен тест жиынтығы тең дәрежеде дайындалғаны маңызды. Бұл функция барлық кескіндерді дайындау сияқты көрінеді:

$$x_{train} = x_{train} / 255$$
$$x_{test} = x_{test} / 255$$


19-сурет – Тестіге арналған жиынның бірінші элементі

Нейрондық желінің моделін құру әрбір қабаттың дұрыс конфигурациясын және одан кейін модельді құрастыруды талап етеді.

Нейрондық желінің негізгі ғимараты - бұл қабат. Қабаттар олар оқытылған деректерден нақты көріністерді шығарады. Және, әрине, біз ақыр соңында олар практикалық тапсырмаларымыз үшін пайдалы нәрсе үйренеді деп үміттенеміз.

Ең терең оқу модельдері тізбектегі қарапайым топтардан тұрады. Көптеген қабаттар, мысалы `tf.keras.layers.Dense`, жаттығу кезінде қолданылатын параметрлерге ие.

Нейрондық желідегі бірінші қабат `tf.keras.layers.Flatten` (`flatten`, `trans. «Align»`) $28 * 28 = 784$ пикселді бір өлшемді массивке екі өлшемді массивтерден (28×28 пиксель) түрлендіреді. Бұл қабаттағы бүктелген пикселдердің қатарын талдап, оларды қатарға қойып жатқандай етіп елестетіп көріңізші. Бұл қабатта оқу үшін параметрлер жоқ; ол тек деректер пішімін түрлендіреді.

Барлық пикселдерді бөлшектегеннен кейін, нейрондық желі екі тығыз `tf.keras.layers.Dense` қабаттардан тұрады. Олар толығымен байланысты қабаттар деп аталады. Бірінші тығыз қабат 128 тораптан немесе нейроннан тұрады. Екінші және соңғы 10 түйін `softmax` қабаты 24 ықтималдық бағалауларының массивін қайтарады, олардың сомасы 1-ге тең. Әрбір нейронда сурет 24 сыныптың біріне жататындығын көрсететін рейтингі бар.

Модель дайындыққа дайын болмас бұрын, біз бірнеше параметрді көрсетуіміз керек. Loss function («жоғалту функциясы») - жаттығу кезінде үлгінің дәлдігін өлшейді. Біз бұл функцияның бағасын дұрыс бағытта «бағыттау» үшін азайтқымыз келеді.

Optimizer - деректерді енгізу және жоғалту функциясының негізінде моделі қалай жаңартылғанын көрсетеді

Metrics - модельді дайындау және сынаудың қадамдарын көрсету. Біздің мысалымызда біз дәлдікті қолданамыз, яғни дұрыс бейнеленген бейнелердің пайызы.

Нейрондық желінің моделін оқыту келесі қадамдарды қажет етеді:

Оқу деректерін үлгіге, біздің жағдайда - train_images және train_labels массивтеріне жүктеңіз. Модель суреттерді дұрыс сыныптармен байланыстыруды үйренеді. Оқуды бастау үшін model.fit әдісін шақырыңыз (fit, trans «Fit», «Fit») - модель оқу деректерінен үйренуді бастайды:

```
In [264]: 1 model = Sequential()
2 model.add(Conv2D(64, kernel_size=(3,3), activation = 'relu', input_shape=(28, 28, 1) ))
3 model.add(MaxPooling2D(pool_size = (2, 2)))
4
5 model.add(Conv2D(64, kernel_size = (3, 3), activation = 'relu'))
6 model.add(MaxPooling2D(pool_size = (2, 2)))
7
8 model.add(Conv2D(64, kernel_size = (3, 3), activation = 'relu'))
9 model.add(MaxPooling2D(pool_size = (2, 2)))
10
11 model.add(Flatten())
12 model.add(Dense(128, activation = 'relu'))
13 model.add(Dropout(0.20))
14 model.add(Dense(num_classes, activation = 'softmax'))
15
16 model.compile(loss = keras.losses.categorical_crossentropy, optimizer=keras.optimizers.Adam(), metrics=['accuracy'])
17 history = model.fit(x_train, y_train, validation_data = (x_test, y_test), epochs=epochs, batch_size=batch_size)
```

```
Train on 19218 samples, validate on 8237 samples
Epoch 1/50
19218/19218 [=====] - 28s 1ms/step - loss: 2.5463 - acc: 0.2139 - val_loss: 1.5415 - val_acc: 0.5498
Epoch 2/50
19218/19218 [=====] - 26s 1ms/step - loss: 1.2368 - acc: 0.5947 - val_loss: 0.7870 - val_acc: 0.7618
Epoch 3/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.7273 - acc: 0.7576 - val_loss: 0.5140 - val_acc: 0.8361
Epoch 4/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.4805 - acc: 0.8345 - val_loss: 0.3650 - val_acc: 0.8722
Epoch 5/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.3370 - acc: 0.8844 - val_loss: 0.2123 - val_acc: 0.9377
Epoch 6/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.2303 - acc: 0.9247 - val_loss: 0.1265 - val_acc: 0.9672
Epoch 7/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.1722 - acc: 0.9443 - val_loss: 0.0976 - val_acc: 0.9712
Epoch 8/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.1227 - acc: 0.9613 - val_loss: 0.0610 - val_acc: 0.9835
Epoch 9/50
19218/19218 [=====] - 26s 1ms/step - loss: 0.0871 - acc: 0.9743 - val_loss: 0.0331 - val_acc: 0.9962
```

20-сурет – Модельді оқыту

Оқу барысында модель шығын (жоғалту) және дәлдік (акс) көрсеткіштерін көрсетеді. Нәтижесінде, үлгі жаттығу жиынтығында оқу барысында шамамен 0,88 (88%) дәлдікке жетеді.

10000/10000 [=====] - 1s 64us/step

Точность после проверки: 0.8745

Енді оқытылған моделді сақтап алу тиімдірек болғаннан кейін, оны біз JSON файлына сақтап аламыз, ал салмағын . h5 форматындағы файлға сақтаймыз.

```
In [265]: 1 model_json = model.to_json()
2 # Записываем модель в файл
3 json_file = open("mnist_model.json", "w")
4 json_file.write(model_json)
5 json_file.close()
```

21-сурет – Модельді json файлға сақтап аламыз

```
In [266]: 1 model.save_weights("mnist_model.h5")
```

22-сурет – Салмақты сақтау

Енді модель оқытылғандықтан, жаңа суреттерге негізделген сурет кластарын болжауға тырысамыз:

```
1 test_labels = test['label']
2 test.drop('label', axis = 1, inplace = True)
3
4 test_images = test.values
5 test_images = np.array([np.reshape(i, (28, 28)) for i in test_images])
6 test_images = np.array([i.flatten() for i in test_images])
7
8 test_labels = label_binrizer.fit_transform(test_labels)
9 test_images = test_images.reshape(test_images.shape[0], 28, 28, 1)
10
11 test_images.shape
```

23-сурет – Тестіге арналған ақпарат

Төменде көрсетілген бөлікте файлға жазылған модельді оқып, салмаған сақтау.

```
In [268]: 1 from keras.models import model_from_json
2 json_file = open("mnist_model.json", "r")
3 loaded_model_json = json_file.read()
4 json_file.close()
5 loaded_model = model_from_json(loaded_model_json)
6 loaded_model.load_weights("mnist_model.h5")
```

24-сурет – Файлдағы модельді оқу

```
In [271]: 1 #compile model
2 loaded_model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=["accuracy"])
3 scores = loaded_model.evaluate(test_images, test_labels, verbose=0)
4 print("Точность модели на тестовых данных: %.2f%%" % (scores[1]*100))
```

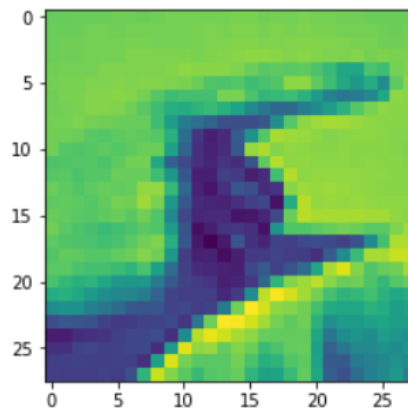
Точность модели на тестовых данных: 83.41%

25-сурет – Тестіні модельге тексеру

Тестіге арналған файлдың бірінші тұрған суретін 28 x 28 өлшемінде шығарып көрейк.

```
In [272]: 1 img = test_images[0]
          2 plt.imshow(img.reshape(28,28))

Out[272]: <matplotlib.image.AxesImage at 0x27a0583ea90>
```



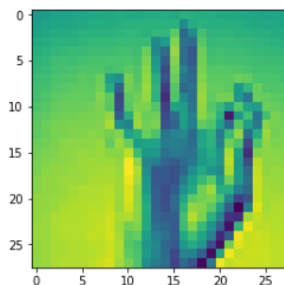
26-сурет – Тест файлындағы бірінші сурет

Тестіге арналған файлдың кез - келген жерінен сурет алып, сол суретке болжау жасайық.

Нәтижесінде 5 деген сан шықты. Бұл сан суреттегі әріптің, әріптер қатарындағы реттік саны.

```
In [273]: 1 img = test_images[45]
          2 plt.imshow(img.reshape(28,28))
          3 print(img.shape)

(28, 28, 1)
```



```
In [274]: 1 img = (np.expand_dims(img,0))
          2 print(img.shape)

(1, 28, 28, 1)
```

```
In [275]: 1 loaded_model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=["accuracy"])
          2 predictions_single = loaded_model.predict(img)
          3 print(predictions_single)

[[0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

```
In [276]: 1 np.argmax(predictions_single[0])

Out[276]: 5
```

27-сурет – Модель болжамы

Ендігі мақсат кірістегі суреттерді MNIST форматындағы ақпаратқа айналдыру, яғни әр пикселді 0-255 түс белгісінің арасында аңықтап, оқытылған модель арқылы болжау.

```
jupyter image2mnist.py ✓ 8 minutes ago
File Edit View Language

1 from PIL import Image, ImageFilter
2 def imagetomnist(argv):
3
4     im = Image.open(argv).convert('L')
5     width = float(im.size[0])
6     height = float(im.size[1])
7     newImage = Image.new('L', (28, 28), (255))
8
9     if width > height:
10         nheight = int(round((20.0 / width * height), 0))
11         if (nheight == 0):
12             nheight = 1
13         img = im.resize((20, nheight), Image.ANTIALIAS).filter(ImageFilter.SHARPEN)
14         wtop = int(round(((28 - nheight) / 2), 0))
15         newImage.paste(img, (4, wtop))
16     else:
17         nwidth = int(round((20.0 / height * width), 0))
18         if (nwidth == 0):
19             nwidth = 1
20         img = im.resize((nwidth, 20), Image.ANTIALIAS).filter(ImageFilter.SHARPEN)
21         wleft = int(round(((28 - nwidth) / 2), 0))
22         newImage.paste(img, (wleft, 4))
23
24     tv = list(newImage.getdata())
25     tva = [(255 - x) * 1.0 / 255.0 for x in tv]
26     print(tva)
27     return tva
28
29 x=imagetomnist('win_test.jpg')
30 print(len(x))
```

28-сурет – Image2mnist скрипті

Жоғарыда айтылған процесті, менің жобамда “image2mnist.py” скрипті орындайды. Бұл функция пиксель мәндерін қайтарады, енгізу - бұл png файлының орны.

28 x 28 өлшемді пикселдерден тұратын ақ қаңқас:

```
im = Image.open(argv).convert('L')
width = float(im.size[0])
height = float(im.size[1])
newImage = Image.new('L', (28, 28), (255))
```

29-сурет – 28 x 28 өлшемді қаңқас

Қай өлшем үлкен екенін тексеріледі, егер ені үлкенірек болса, онда ені 20 пикселге айналады. Ені бойынша биіктігінің өлшемін өзгертеміз. Сирек жағдайда, бірақ минимум - 1 пиксель болады. Ары қарай өлшемін өзгертеміз және бұрмалаймыз және көлденең позицияны есептеледі де, ақ қаңқасқа өлшемі кескінделген суретті қоямыз.

```

if width > height:
    nheight = int(round((20.0 / width * height), 0))
    if (nheight == 0):
        nheight = 1
    img = im.resize((20, nheight), Image.ANTIALIAS).filter(ImageFilter.SHARPEN)
    wtop = int(round(((28 - nheight) / 2), 0))
    newImage.paste(img, (4, wtop))

```

30-сурет – Ені үлкен болған жағдайда

Егер де биіктігі үлкенірек болса, онда биіктігі 20 пикселге айналады. биіктігі бойынша енінің өлшемін өзгертеміз. Сирек жағдайда, бірақ минимум - 1 пиксель болады. Ары қарай өлшемін өзгертеміз және бұрмалаймыз және көлденең позицияны есептеледі де, ақ қаңқасқа өлшемі кескінделген суретті қоямыз.

```

else:
    nwidth = int(round((20.0 / height * width), 0))
    if (nwidth == 0):
        nwidth = 1
    img = im.resize((nwidth, 20), Image.ANTIALIAS).filter(ImageFilter.SHARPEN)
    wleft = int(round(((28 - nwidth) / 2), 0))
    newImage.paste(img, (wleft, 4))

```

31-сурет – биіктігі үлкен болған жағдайда

Пиксель мәндерін алу. 0-ге және 0-ге дейін пикселдерді қалыпқа келтіру. 0 таза ақ, 1 таза қара.

```

24     tv = list(newImage.getdata())
25     tva = [(255 - x) * 1.0 / 255.0 for x in tv]
26     print(tva)
27     return tva

```

32-сурет – Қалыпқа келтіру және мәнді қайтару

```
In [277]: 1 from IPython.display import Image
          2 img = Image("test/win_test.jpg")

In [278]: 1 import image2mnist

In [279]: 1 dir(image2mnist)

Out[279]: ['Image',
            'ImageFilter',
            'builtins',
            'cached',
            'doc',
            'file',
            'loader',
            'name',
            'package',
            'spec',
            'image2mnist',
            'x']

In [280]: 1 #test real image
          2 x = image2mnist.image2mnist('win_test_a2.jpeg')
          3 print(x)
```

33-сурет – Сурет салып, мәнің экранға шығару

2 Құрылымдық - техникалық бөлім

2.1 Бағдарламалау ортасы

Rycharm өнімді қосымшаларды әзірлеуге арналған құралдар жиынтығы. Бұл орта келесі себептер бойынша таңдалды:

- a) бағдарламаның мәтіндік редакторында көптеген пайдалы функциялар бар;
- b) екі консольдық және графикалық қосымшаларды дамытуға қолдау көрсетіледі;
- c) көп қабатты қосымшаларды дамытуға орнатылған мүмкіндігі бар;
- d) бірліктерді тестілеу үшін функционалды отладчик бар;
- e) жақсы анықтамалық материалға ие;
- g) әртүрлі программалау тілдеріндегі бағдарламалардың көптеген мысалдары.

2.2 Жүйені толық тексеру

Windows ОЖ-ны басқару үшін құрылған кіші жүйелер негізінде әртүрлі жағдайларда әдістерді пайдалану уақытын өлшеу жүргізілді, сынақ нәтижелері салыстырмалы кестеде көрсетілген.

1-кесте – тестілеу нәтижесінің кестесі

Көрсеткіштер	Динамикадағы объектілерді түс негізінде анықтау модельдері	Адамның басын анықтау тәсілі	Қолдың қимылдарын бір тонды фонда анықтау	Қолды кеңістікте анықтау
Жұмыс уақыты	0,5 сек.	1 сек.	1 сек.	2 сек.
Жарыққа тәуелділік	Жоғары	Төмен	Төмен	Төмен
Фонның әсері	Орташа	Төмен	Орташа	Төмен
Сенсорға дейінгі қашықтық	3 м. Дейін	0,5 м. дейін	0,5 м. дейін	3 м. дейін

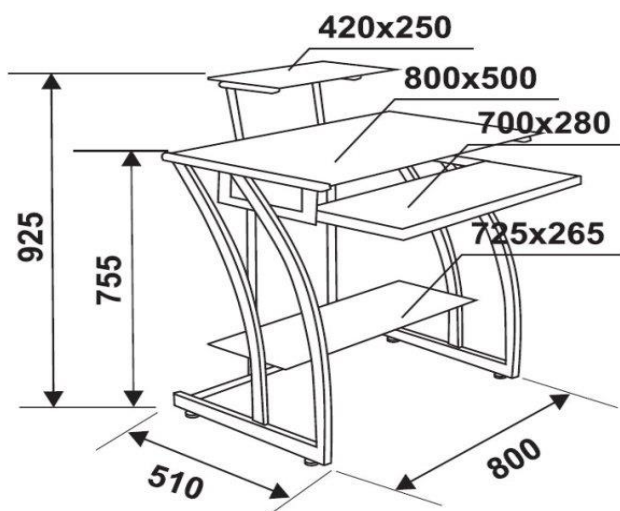
3 Экология және еңбекті қорғау

Кез келген жұмыс берушінің міндеті - қызметкерлерге қауіпсіз еңбек жағдайларын жасау. Қызметкердің жұмыс орнына қойылатын талаптарды және стандарттарды стандарттау мақсатында ГОСТ, СанПиН, СНиП және басқалары үшін ұсынылған құжаттар әзірленеді.

3.1 Жұмыс орнының талаптары

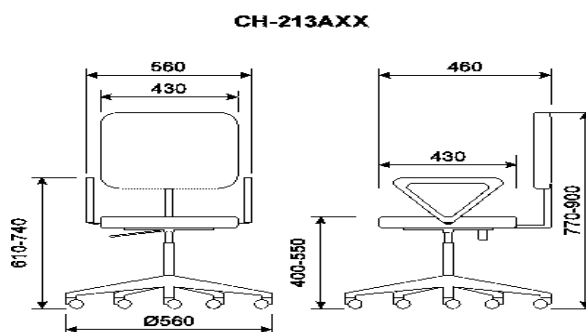
Бір адамдағы жұмыс алаңының көлемі кемінде 4,5 м² (СанПиН 2.2.2 / 2.4.1340-03-де көрсетілген норма) болуы тиіс.

Жұмыс үстелінің көлемі мен формасы барлық қажетті жұмыс жабдығын орналастыруға, сондай-ақ құжаттарды орналастыру мен жұмыс жазбаларын сақтауға мүмкіндік береді.



34-сурет – Негізгі параметрлерге сай келетін ұсынылатын жұмыс үстелі

Жұмыс креслолары келесі элементтерден тұруы керек: орындық, артқы және қолтаңба. Сондай-ақ, бас киім мен аяқ киім де болуы мүмкін.

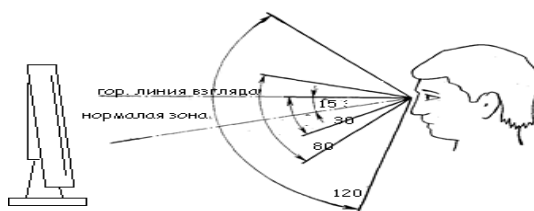


35-сурет – Орындық құрылысы

Жұмыс орнында аяқ қоюға арналған параметрлермен жабдықталуы мүмкін: ені кемінде 300 мм, тереңдігі кемінде 400 мм, биіктігі 150 мм-ге дейін және тіреудің тірек бетінің 20° дейін бұрылу бұрышы болуы мүмкін. Тұғырдың бетіне қойылатын талаптар: бедерлі және 10 мм жоғары жағымен жабдықталған.

Пернетақта орналасуы: қолданушы тарапынан арнайы жұмыс бетінде, биіктіктен реттелетін стендте, негізгі жұмыс үстелінде емес, жиектен 100 - 300 мм қашықтықта орналасқан жөн.

Орналасқан жерді және пайдаланушының орналасуын бақылау:



36-сурет – Бақылау бұрыштары

Бағдарламалық инженер жұмыс істейтін бөлмеде санитарлық ережелермен реттелетін микроклимат параметрлері сақталуы керек.

Инженер-бағдарламашының жұмыс орнында макро-климатты сақтау мәселесінде маңызды компонент - жылу және ауа баптау жүйелерінің болуы. Еденнің және бастың ауданындағы ауа температурасының айырмашылығы 50° С-тан жоғары емес болуы тиіс. Кондиционерлік жүйенің дұрыс жұмыс істеуінің негізгі индикаторы сағатына қанша рет бөлмеде ауаның толық ауысуы болып табылады.

Сонымен қатар, бағдарламалық жасақтама инженерінің жұмыс орнында шу деңгейін қолайлы шегінде ұстау керек. 71, 61, 54, 49, 45, 42, 40, 38 дБ диапазонында болатын дыбыс қысымының нормативтік деңгейлерінің шекті спектрі анықталады. Жоғарыда көрсетілген талаптар мен талаптарға сай жұмыс орны еңбек өнімділігі жоғары, жұмысты орындау кезінде тиімділікті сақтау үшін жұмыс жасау үшін қолайлы жағдайлар жасайды.

3.2 Жұмыс орнындағы жарық

Бөлме табиғи және жасанды жарықпен жарықтандырылуы тиіс. Қамтамасыз етіңіз: жеткіліксіз немесе артық жарық; жарықтың дұрыс емес бағыты.

Бағдарламалық жасақтама инженерінің жұмыс орнының жарықтылығын есептеу мыналардан тұрады:

- 1) жарықтандыру жүйесін таңдау;
- 2) қажетті арқандар санын анықтайды;
- 3) оларды түр бойынша таңдау;
- 4) оларды орналастыру жүйесін айқындайды.

Сондай-ақ, құжаттарды бөлектеу үшін жергілікті жарықтандыру үшін жеке үстел шамдарын орнату рұқсат етіледі. Жеке шамдардың жұмыс бетіне орналастырылған кезде, монитордың жұмыс бетінде шағылыстың пайда болуына байланысты жұмысқа араласпау керек.

3.3 Жүйенің адамға әсері

Компьютерде ұзақ жұмыс адам денсаулығына айтарлықтай әсер етуі мүмкін. Қазіргі уақытта компьютерлерді өмірдің барлық салаларында қолдану кеңейе түсуде, демек, көп адам компьютерді бақылаушыларда күндерін өткізуге мәжбүр.

Компьютерді пайдалану мәселесіне дұрыс емес көзқарас көбінесе бағдарламалық жасақтама инженерінің денсаулығына зиян тигізуі мүмкін. Егер байланыссыз компьютерлік басқару әдісін жасау және енгізу орын алса, онда карпаль-ульнар синдромы сияқты ауру сырқатқа түсіп кетуі мүмкін. Гиподинамикалық мәселе де шешілуі мүмкін, қызметкер жұмысын қатаң түрде «мәжбүрлеу» жағдайында орындауға мәжбүр етпейді, «тұру» позициясынан белсенді әрекеттер жасау үшін көбірек мүмкіндіктер пайда болады және т.б. Монитордан мәтінді көзге дейінгі қашықтықты өзгерту үшін оқу кезінде, бұл өз кезегінде тұрақты кернеуді жоюға көмектеседі. Монитордан тұрақты түрде статикалық оқуға байланысты визуалды бұзылыс өткен нәрсе болады.

Әлемде қоршаған ортаны қорғаушылар ұзақ уақыт бойы дабыл қағып кетті. Ластану ауқымы барлық мүмкін шектеулерден өтті. Компьютерлік техниканы өндіру сатысында көптеген табиғи ресурстар пайдаланылады және қоршаған ортаға теріс әсер ететін қалдықтардың үлкен саны қалыптасады. Су, ауа және тағы басқа заттар ластанған. Сондай-ақ, компьютерлік техниканы тастаған кезде, табиғатта адамдар аз. Көпшілігіміз ескірген жабдықты қоқысқа тастаймыз, дұрыс өңдеу тұжырымдамасы бізге таныс емес. 1991 жылдан бастап әртүрлі дерек көздеріне сәйкес, Ресейге әртүрлі өндірушілерден 10 000 000-нан астам техника әкелінгенін ескере отырып, қорғасын, сурьма, сынап, кадмий,

мышьяк және түрлі полимерлер сияқты көптеген зиянды металдар қазірдің өзінде қоқыстарды және біздің экологияны уларды.

Біздің өмірімізде де, үйде де, біздің компьютерімізсіз өмір сүруді ойламаймыз. Ол барлық жерде және барлық жерде көмекші. Сондықтан, компьютерлік басқаруды қаңылтыр арқылы кеңінен енгізу арқылы және компьютерлік тінтуірді пайдаланбаудың арқасында, өндіріс кезеңінде де, кәдеге жарату сатысында да қоршаған ортаға жүктемені аздап мүмкін.

КОРЫТЫНДЫ

Жобада қол қимылдарын анықтау және тану әдістері жасалды. Осы әдістердің негізінде адамның компьютермен байланысы үшін бағдарламалық жасақтама жасалды.

Қазіргі уақытта осы технологиялар дамудың жоғары әлеуетіне ие, себебі олар мүгедектігі бар адамдарды ақпарат әлеміне интеграциялау және олардың әлеуметтену деңгейі мәселелері болып табылады.

Анықтау негізінде жасалған операциялық жүйені басқаруды практикада енгізгенде, бірнеше айқын артықшылықтар пайда болады: компьютерлік тінтуір секілді қосымша перифериялық құрылғыларды пайдалану қажеттілігі, бір уақытта пайдаланушылардың ұжымдық жұмыс істеу мүмкіндігі, жұмыс орнына қатты сілтеме болмауы. пайдаланушылардың еркіндік дәрежесін арттыру.

Дипломдық жобада келесі жұмыстар орындалды:

1) Адам-компьютерлік өзара әрекеттесу жүйелерінде қимылдарды тануды жүзеге асыратын қарапайым WEB-камера мен құрылғылармен қолданыстағы бағдарламалық жасақтаманы талдау. Осы жүйелердің артықшылықтары мен кемшіліктері анықталды.

2) Ең танымал модельдерді тану әдістеріне салыстырмалы талдау жасалды, қолдың қаңылтыр тану әдістерінің жіктелуі қарастырылды.

3) Жүргізілген зерттеудің негізінде объектілерді - қол мен саусақтарды танудың өз әдістерін жасау үшін теориялық негіз болды. Оларды сыныптау жүйесі анықталған. Жасалған әдістер мен алгоритмдер қосымша маркерлерді қажет етпейді және сенімділікке ие.

4) Өзірленген тану әдістерінің негізінде қол қимылдарын анықтау әдісі ұсынылды.

5) Пайдаланушы жүйемен өзара іс-қимыл жасау үшін қажетті көптеген командаларды талдады және анықтады.

6) Орындалатын сынақтар негізінде ұсынылатын әдістерді тану сапасын бағалау.

Дипломдық жобаны жүзеге асыру нәтижесінде динамикалық объектілерді тану және оның негізіндегі адам-машина интерфейстерін құру міндеті толығымен шешілді.

ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Абдуллин Ю.Е., сандық бейне ағынында қозғалысты анықтаудың түс сегменті // Томск мемлекеттік университеті 2008. - 248 б.
- 2 Абдуллин Ю.Е. Техникалық көзқарас жүйелерінде қозғалысты анықтау мәселесінде фондық кадр қалыптастыру // «Ресейде зерттеу» электронды журналы, № 124, 2007, 1289 б., [Http://zhurnal.ape.relarn.ru/articles/2007/124.pdf](http://zhurnal.ape.relarn.ru/articles/2007/124.pdf).
- 3 Альфимцев А.Н. Динамикалық қимылдарды түсіру, қадағалау және тану әдістерін әзірлеу және зерттеу // МСТУ. Н.Е. Баумана. - Мәскеу, 2008.
- 4 Анисимов Б.В. Тану және цифрлық кескінді өңдеу - М.: Жоғары. Мектеп, 1983 - 295 с.
- 5 Баженов А. Морфологиялық кескінді өңдеу [Мәтін] / Softline - Мәскеу, 2012. - 20 б.
- 6 Визилтер Ю В., Желтов С.Ю., Княз В. А., Ходарев А.Н., Моржин А.В. LabVIEW IMAQ Vision мысалдарымен сандық суреттерді өңдеу және талдау. - Мәскеу: ДМК, 2008. - 233 б.
- 7 Гонсалес Р., Вудс Р. Сандық кескінді өңдеу. 3-ші басылым, қайта қаралған және кеңейтілген. - Мәскеу: Техносфера, 2012. - 1104 б.
- 8 Ковалева И.Л. «Түсті үлгілерді түрлендіру» зертханалық жұмысына арналған нұсқаулар // BNTU.- Минск, 2008. - 31 б.
- 9 Колдаев В.Д. Процесті автоматтандыру жүйелерінің контурлық кескін сегменттеуін дамытудың теориялық негізі [Мәтін]: автор. дис. конкурста ғалымдар қадам Техника ғылымдарының докторы (05.13.06) / Колдаев Виктор Дмитриевич; МИЭТ - Мәскеу, 2014. - 354 б.
- 10 Кудрявцев Л.В. Математикалық талдаудың қысқаша курсы - М.: Ғылым, 1989 - 736 б.
- 11 Куракин А.В. Сурет формасын медиа-бейнелеу негізінде динамикалық қимылдарды тану // Мәскеу мемлекеттік университеті. Ломоносова.- Мәскеу, 2014. - 26 б.
- 12 Малашин П.О., аспирант; Люцив В.Р., дәрігерлік техника. Екілік бейненің бейімделу морфологиялық сүзгісін қолдану арқылы қимылдарды тану міндетіне қолдың түсінің қалпына келтірілуі. // «Мемлекеттік оптикалық институты» ААҚ. С.В. Вавилова, Санкт-Петербург, 2013. - 8 б.
- 13 Нагапетян В.Е. Қашықтағы суреттерді талдау негізінде қол қимылдарын тану әдістері [Мәтін]: автор. дис. конкурста ғалымдар қадам Канд. Физ.-матема. Ғылымдар (05.13.17) / Ваагн Едвардович Нагапетян; RUDN - Мәскеу, 2013. - 108 б.

А қосымшасы **(міндетті)**

Техникалық тапсырма

А.1.5 Қол қимылдарын анықтауға арналған жүйені жасақтауға техникалық тапсырма

Техникалық тапсырма қол қимылдарын анықтауға арналған жүйені жасақтауға бағытталған. Жүйе есту қабілеті төмен адамдар қолданатын ым тілінің әріптерін анықтау мақсатында құрылған. Бұл жүйе есту қабілеті төмен адамдардың қоршаған ортамен қиындықсыз қатынау шараларын қолдану үшін. Адам мен адам қарым – қатынасын ғана емес, адам мен компьютер қарым–қатынасын автоматтандыру, жүйелеу процесіне негізделген. Бұл жүйе есту қабілеті шектеулі адамдарға қоршаған ортадағы оларға бейімделмеген бұйым, технология, техникаларды басқаруға, тындауға, үйренуге мүмкіндік береді.

А.1.5.1 Мақсаты

Жүйе қол қимылдары негізінде жасалған суреттердегі ым тілінің әріптерін анықтап, модель негізінде болжау жасау мақсатына арналған.

А.1.5.2 Жүйенің функционалды талаптары

Жүйе келесі функцияларды орындай алу керек:

- жүйенің инициализациясы;
- кіріске берілген суреттің дұрыс өңделуі және берілген сурет негізінде жасалған қорытынды ақпарат;
- кіріске берілген сурет негізінде керекті ақпаратты көрсету.

Бастапқы деректер:

- ым тілі негізінде жасалған қол қимылдарының суреті;
- модельге арналған оқыту ақпараттары.

Нәтиже:

- класс атауын экранға шығару.

А қосымшасының жалғасы

А.1.5.3 Қауіпсіздік талаптары

Кіріс туралы ақпаратты бақылауды қамтамасыз етіңіз. Жүйемен жұмыс істеу кезінде қате пайдаланушы әрекеттерін блоктауды қамтамасыз етіңіз. Сақталған ақпараттың тұтастығын қамтамасыз етіңіз.

А.1.5.4 Техникалық талаптар

Жүйе IBM үйлесімді дербес компьютерлерде жұмыс істеуі керек. Ең аз конфигурация: процессор түрі - Pentium және одан жоғары; Жад сыйымдылығы - 32 МБ және одан жоғары.

А.1.5.5 Бағдарламалық үйлесімділік талаптары

Жүйе Win32 API-ді іске асыратын Windows операциялық жүйелерінің бақылауында жұмыс істеуі керек.

Б қосымшасы (міндетті)

Бағдарлама коды

```
from IPython.display import Image
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

train = pd.read_csv('sign_mnist_train.csv')
test = pd.read_csv('sign_mnist_test.csv')

labels = train['label'].values
unique_val = np.array(labels)
np.unique(unique_val)
print(plt.figure(figsize = (18,8)))
print(sns.countplot(x =labels))
train.drop('label', axis = 1, inplace = True)

images = train.values
images = np.array([np.reshape(i, (28, 28)) for i in images])
images = np.array([i.flatten() for i in images])

from sklearn.preprocessing import LabelBinarizer

label_binrizer = LabelBinarizer()
labels = label_binrizer.fit_transform(labels)
print(labels)

plt.imshow(images[0].reshape(28,28))

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(images, labels, test_size = 0.3,
random_state = 101)

import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Dropout
batch_size = 128
num_classes = 24
epochs = 50
```

Б қосымшасының жалғасы

```
x_train = x_train / 255
x_test = x_test / 255

x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
plt.imshow(x_train[0].reshape(28,28))

model = Sequential()
model.add(Conv2D(64, kernel_size=(3,3), activation = 'relu', input_shape=(28,
28 ,1) ))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Conv2D(64, kernel_size = (3, 3), activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Conv2D(64, kernel_size = (3, 3), activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Flatten())
model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.20))
model.add(Dense(num_classes, activation = 'softmax'))

model.compile(loss = keras.losses.categorical_crossentropy,
optimizer=keras.optimizers.Adam(), metrics=['accuracy'])
history = model.fit(x_train, y_train, validation_data = (x_test, y_test),
epochs=epochs, batch_size=batch_size)
test_labels = test['label']
test.drop('label', axis = 1, inplace = True)
test_images = test.values
test_images = np.array([np.reshape(i, (28, 28)) for i in test_images])
test_images = np.array([i.flatten() for i in test_images])

test_labels = label_binrizer.fit_transform(test_labels)
test_images = test_images.reshape(test_images.shape[0], 28, 28, 1)
test_images.shape
y_pred = model.predict(test_images)
from sklearn.metrics import accuracy_score
print(test_labels)
print(accuracy_score(test_labels, y_pred.round()))
```