



Университет:	Satbayev University
Название:	manarov_shami.docx
Автор:	Макалов Шамил
Координатор:	Алия Адильбекова
Дата отчета:	2019-05-06 10:00:09
Коэффициент подобия № 1: <input type="checkbox"/>	1,9%
Коэффициент подобия № 2: <input type="checkbox"/>	0,0%
Длина фразы для коэффициента подобия № 2: <input type="checkbox"/>	25
Количество слов:	2 794
Число знаков:	22 093
Адреса пропущенные при проверке:	
Количество завершенных проверок: <input type="checkbox"/>	21

Ғылыми жетекші

А. К. Адильбекова

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Мамандығы 5В070400 – Есептеу техникасы және бағдарламалық
қамтамасыз ету

Студент Манапов Шамил Ялкунжанович

Тақырыбы: “Тұтынушының тапсырыс беру жүйесін автоматтандыру”
web-қосымшасын құру

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ



Диплом жобасын жасаушы Манапов Шамилдің алдына тұтынушының тапсырыс беру жүйесін автоматтандыру web-қосымшасын құру тапсырмасы қойылған.

Дипломдық жобада ресторан, асхана және т.б. жерлерде тұтынушыға деген ыңғайлылық және жұмысшылардың міндеттерін оңайлатуға деген мүмкіндіктер беретін кешен құрылған.

Жұмыс барысындағы басты мәселелер: отырыс орындарында тамақтың құнын автоматты түрде есептеу жүйесін құруда қолданылатын әдістерді талдау, ең қолайлы аппараттық платформаны таңдау, тұтынушыларға деген интерфейсі үшін бағдарламалық қамтамасыз етуді әзірлеу құралдарын таңдау, объектінің басқару алгоритмдерінің жұмыстары мен тұтынушы үшін бағдарламалық қамтамасыз етудің алгоритмдерін іске асыру болып табылады. Бұл қазіргі заманауи ең өзекті және келешегі бар технологияларды қолданудағы маңызды мәселелердің бірі.

Менің пікірімше, диплом жазушы алдына қойылған тапсырманы барлық талаптарға сай, біліктілігімен, толық орындады және аппараттық жүйелердің заманауи технологияларын меңгергендігін көрсетті.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып Манапов Шамилға 5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ  А.К. Адильбекова
«Программалық инженерия»
кафедрасының лекторы « 8 »  2019 жыл

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Манапов Шамил Ялкунжанович

«Тұтынушының тапсырыс беру жүйесін автоматтандыру»
web-қосымшасын құру

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

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

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,

ассистент-профессор

 Р. Юнусов

" 30 " мае 2019 ж.

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

Тақырыбы: «Тұтынушының тапсырыс беру жүйесін автоматтандыру»
web-қосымшасын құру

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Орындаған

Ш. Я. Манапов

Ғылыми жетекші

техн. ғыл. магистрі, лектор

 А. К. Адильбекова

" 30 " 04 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты


Программалық инженерия кафедрасы

5B070400 – «Есептеу техникасы және бағдарламалық камтамасыз ету»

БЕКІТЕМІН

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
" " " 2019 ж.

**Дипломдық жоба орындауға
ТАПСЫРМА**

Білім алушы: Манапов Шамил Ялқунжанович

Тақырыбы: «Тұтынушының тапсырыс беру жүйесін автоматтандыру» веб-қосымшасын құру

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

Орындалған жобаның өткізу мерзімі «13» мамыр 2019 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Тұтынушының тапсырыс беру жүйесін автоматтандыруға арналған жүйенің веб-қосымшасын құру, қосымшаны қолданушыларға үнемді және тиімді пайдалануды ұйымдастыру.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) нейронды желі және веб-қосымшасыны талдау

б) жобалау бөлімі;

в) өндеу құралдары және жобаны талдау;

г) қолданбалы бөлім;

д) А Қосымшасы – техникалық тапсырма.

е) Б Қосымшасы – бағдарлама мәтіні.

Жобаның презентациялық 13 слайды ұсынылған.

Ұсынылған негізгі әдебиеттер: 10 атау.

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Диплом жұмысының жоспар-құрылымын құру.	14.01.2019	орындалған
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	орындалған
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім беру бойынша есеп беру жазбасын дайындау	01.02.2019	орындалған
4. Дипломның екінші бөлімі – жобалау сызбаларын дайындау.	15.02.2019	орындалған
5. Жобаның веб-қосымшасын тестілеуден өткізу.	18.03.2019	орындалған
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2019	орындалған

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Таурбекова А. А. лектор	04.05.19	
Бағдарламалық бөлім	Қалдыбеков С. Б. сениор-лектор	06.05.19	

Ғылыми жетекші Адильбекова А. К.

Тапсырманы орындауға қабылдап алған студент Манапов Ш. Я.

Күні

«5» 10 2018 ж.

АҢДАТПА

Бұл мақалада кез келген бір тамақты детектеу сипаттамасы келтіріледі. Нейрон желісі қазіргі кезде заманауи технологиялардың ең маңызды бөлігі болып табылады. Нейрон желісін үйренуге қалай жіберу керек екенін және қалай тестілеу екені туралы айтылған.

Камера көмегімен объектіні тану – қазіргі кезде маңызды жұмыстардың біріне жатады. Оның күші әр түрлі салаларда маңызды орын алады олар: робототехникаға арналған, ендірілген құрылғыларында, автоматтандырылған басқару жүйелерінде және т.б. салаларда қолданылады. Қазіргі уақытта детектеу және объектіні тану қиын тапсырма болып табылады.

Жоба үш бөлімнен тұрады: бірінші бөлімде жобаны, жобаның жаңа моделі жобаланды; екінші бөлімінде жоба жүйесінің жұмыс істеу принциптері UML-де жобаланып, деректер қоры құрылымы келтірілді; үшінші бөлімде жоба web-қосымшаның нәтижелері, жобаның құрылымы, талдау қызметтерінің жүзеге асырылуы бойынша есеп берілген.

Жобаның түсініктеме жазбасы жалпы 43 бет, оның ішінде 25 сурет, 2 міндетті қосымшадан тұрады. Жұмысты жазу кезінде 10 әдебиеттер мен мақалалар қолданылды.

АННОТАЦИЯ

В данной статье приводится описание детекции еды. Нейронная сеть в настоящее время самая важная часть современной технологии. Внизу показано как следует отправлять нейронные сети на тренировку и как их тестировать.

Распознавание объекта – относится к одной из важнейших задач искусственного интеллекта. Его мощь используют в разных сферах как робототехника, автоматизированные системы управления и внедрённые техники. В настоящее время детектирование и распознавание объектов трудные задачи.

Проект состоит из трех разделов: в первой части проведен анализ аналогичных систем и проектирована новая модель проекта; во второй части принципы функционирования системы проекта в UML и приведены структура базы данных; в третьей части представлены результаты web-приложения проекта, структура проекта.

Пояснительная записка проекта состоит из 43 страниц, в том числе 25 рисунков, 2 обязательных приложения. При написании работы использовались 10 литературы и статьи.

ANNOTATION

This article describes the detection of an foods. The neural network is the most important part of modern technology at present. Below you can see how to sent neural networks to train and to test them.

Object recognition with camera is one of the important tasks of artificial intelligence at the present time. The power of it is used in various fields, such as robotic technology, automated control system and embedded technologies. Today detecting and recognizing an object is difficult task.

The project consists of three sections: in the first part, the analysis of similar systems and designed a new model of the project; in the second part, the principles of the system Help Desk are designed in UML and the structure of the database; in the third part, the results of the web-application of the project, the structure of the project, the report on the implementation of services for obtaining, analysis, storage of data.

Explanatory note of the project consists of 43 pages, including 25 figures, 2 mandatory annexes. When writing the work used 10 literature and articles.

МАЗМҰНЫ

	Кіріспе	8
1	Негізгі бөлім	9
1.1	Жобаның өзектілігі	9
1.2.1	Жарық өзгеруіндегі туындайтын қиындықтар	9
1.2.2	Объектіні тану кезінде туындайтын қиындықтар	10
1.3	Web – технологиясы	10
2	Жобалау бөлімі	12
2.1	Унифицирленген моделдеу тілі	12
2.2	Rational Rose құралы	12
2.3	Прецеденттер диаграммасын құру	12
2.4	Тізбек диаграммасын құру	13
2.5	Кооперация диаграммасын құру	14
2.6	Күй диаграммасын құру	15
2.7	ER диаграммасын құру	16
3	Жобалау бөлімі	18
3.1	Python программалау тілі	18
3.1.2	Python программалау тілін қолдану ерекшелігі	18
3.2	Tensorflow кітапханасы	18
3.3	Keras кітапханасы	19
3.4	Flask веб-беті	19
3.5	NumPy кітапханасы	20
3.6	OpenCV компьютерлік көзқарас кітапханасы	20
3.7	YOLO моделі	20
3.8	DarkFlow кітапханасы	21
3.9	Нейрондық желі	21
3.9.1	Машиналық оқытылу	22
3.10	Веб бетті құру және машиналық оқыту жұмысын іске асыру	22
3.11	Кескін ішіндегі объектіні табу	23
3.12	Жаңа жиын оқыту үшін деректердің жаңа сыныптары	23
3.13	Кескін координаттарын алу құралы	23
3.14	YOLO v2 баптауларын жасау	25
3.15.1	Үйретілген моделдің нәтижесі	28
3.15.2	Қолданушы интерфейсі	29
	Қорытынды	30
	Пайдаланылған әдебиеттер тізімі	31
	А қосымшасы. Техникалық тапсырма	32
	Б қосымшасы. Бағдарлама мәтіні	34

КІРІСПЕ

Біздің өміріміздің әр түрлі салаларын автоматтандыру – өте керек процесстердің бірі. Көптеген процесстер компьютерлік жүйелерде және көп құрылғыларда да көп кездеседі. Бұл деректерді тиімді және табысты өңдеуге, оларды сақтауға және тез іздеуге мүмкіндік береді.

Нейронды желі – бұл кең ауқымды саласы. Кез келген объектіні әртүрлі түрде анықтап және әрбір нәтижелерді аламыз. Дегенмен, машина жасаудағы көптеген тәсілдер және оны тесттеу арқылы өте қызықты нәтижелерге қол жеткізуге болады.

Дипломдық жобаның өзектілігі – қазіргі кезде барлық адамдарға автоматтандыру құрылғылары өте көп. Кез келген асхана отырыс орындары немесе асханаларада автоматтандыру құрылғылары бар. Олардың ішінде банк картасымен төлеуге арналған құрылғыны алсақ болады.

Дипломдық жобаға алған жұмысымның мақсаты – отырыс орындарының тұтынушы және қызметкерлерге деген жеңілдік жасау. Тұтынушының тамақ құнын есептейтін кезде бұл программа автоматты түрде құнын шығарып, қызметшіге деген ыңғайлы болып табылады. Сонымен бірге, жұмыс істеу механизімін және қолданушылар мен орындаушылар арасындағы байланысты барынша жеңілдетуді жүзеге асыру және уақыт үнемдеу болып табылады.

Соңғы бірнеше жылда нейрон желілеріне деген қызығушылық артып келеді. Нейрон желісі қазіргі кезде заманауи технологиялардың ең маңызды бөлігі. Камера көмегімен объектіні тану – қазіргі кезде маңызды жұмыстардың бірін жатады. Ол әртүрлі салаларда және де атқаратын қызметі де жоғары. Робототехникаға арналған құрылғылар, автоматтандыру жүйелерінде және көптеген салаларда қолданылады. Қазіргі уақытта детектеу және объектіні тану қиын тапсырмалардың бірі. Нейронды желі күрделі моделдерді үйреніп, автоматтандыруда қоланылады. Мысалы, кез келген бір болжамдарда және де автоматтандыру жағдайларында қолданылады.

Жобамның бірінші тарауында жүйені қолдану және кездесетін қиындықтарға байланысты мағлұматтарды зерттедім.

Екінші тарауда жобаның жұмыс істеу принциптерін жобалау және деректер қоры құрылымы келтірілді.

Қосымшаның нәтижелері, жобаның құрылымы, қызметінің сипаттамасы үшінші тарауда келтірілді.

Жоба веб-қосымша және нейронды желі негізінде жасалып, отырыс орындарының қызметкерлері үшін өте тиімді және ыңғайлы интерфейс те құрылды.

1 Негізгі бөлім

1.1 Жобаның өзектілігі

Жасанды нейрон желілері адам миіне ұқсас келеді, олардың басты айырмашылығы компьютерлер ақпарат өңдеу тәсілін сандық түрде жүзеге асады. Адам миі үлкен жүйке клеткалары және нейрондардан тұрады. Нейрон басқа да нейрондармен байланысып, олардың саны 10-нан 100000 нүктелерге дейін қосылуы мүмкін. Адамның миі нақты міндеттер орындай алатын нейрондар жұмысын ұйымдастыруға және олар компьютерге қарағанда тез орындауға қабілеті бар. Ақпаратты өңдеудің қалыпты үлгісі көру бола алады. Көру бұл дүние жүзімен байланыстау болуын қадағалайды және бірқатар міндеттерді орындай алады. Мысалы, таныс бейтаныс адамды ажырата алады. Компьютер адам миіне қарағанда 1000 есе көп нейрондар болады.

Қазіргі кезде барлық адамдарға автоматтандыру құрылғылары өте көп. Кез келген асхана отырыс орындары немесе асханаларада автоматтандыру құрылғылары бар. Олардың ішінде банк картасымен төлеуге арналған құрылғыны алсақ болады. Соған байланысты, қазір күніне мыңдаған автоматты құрылғылары жарыққа шығып жатыр. Әрқайсысы, неше түрлі бағытқа бағытталған. Сондықтан, «Тұтынушының тапсырыс беру жүйесін автоматтандыру» проектісінің түпкі мақсаты кез келген отырыс орындарында сатып алынған тауардың құнын автоматты түрде шығару болып табылады. Жоба нейронды желі және веб-сайттан құралады.

1.2.1 Жарық өзгеруіндегі туындайтын қиындықтар

Сыртқы жарықтың түсуіне байланысты тану барысында жүйелердің жұмысына қатты әсер етеді. Сондықтан кезе келген бір модельді үйретуге жіберген жағдайда кескіндерді дайындау барысында, бір сурет әр түрлі жарықтарда неғұрылым көп болса, солғұрылым тану процесі жоғары болады. Объектті табу барысында, объектке түсетін камера тікелей жарыққа байланыстылығында, ол 1.1-суретте көрсетілген.



1.1-сурет - Жарықтың объектке түсуі

1.2.2 Объектіні тану кезінде туындайтын қиындықтар

Объектіні тану – бұл күрделі жұмыстардың бірі десекте болады. Объектіні тану үшін, кескін жоғары деңгейде көрініс алуы керек. Объектіні жақсы тану үшін біріншіден датасет жақсылап дайындау керек. Датасет неғұрылым көп болса, модел солғұрылым жақсы үйретіледі. Солардың ішіндегі әр класстардың кескіндер саны бір-біріне жақын түрде болса, моделдің тану жүйесі де жоғары болады.

1.3 Web – технологиясы

Internet қазіргі кезде біздің өмірімізге мықтап кірді десекте болады. Ол компьютерді ғаламшарда орналасқан басқа да компьютерлермен байланыстырып отыратын әлемдік желі. Интернеттің көмегімен адамдар бір қалада тұрмасада бір-бірімен ақпарат алмаса алады. Ғалымдар үлкен есептеу ресурстарын қажет ететін мәселелерді шешуге арналған компьютерлерді біріктіріп, керекті тәжірбиелерін жүргізе алды. Интернет технологиясын даму барысында, қашықтықтан жұмыс істеуге деген адамдар саны артты. Бұл оң нәтижесін көрсетті. Ақыр соңында, көбіміз жұмысқа және үйге оралу үшін, күн сайын бірнеше сағатымызды үнемдейміз. Ақпараттандыруды дамыта отырып, адамдардың көбісі үйден шықпай-ақ, дербес компьютерде отырып жұмыс істей алады.

Осыдан кейін интернетті пайдаланудың толық тізбесі жақын арада біздің өмірімізде қаншалықты маңызды рөл атқара алатыны анық. Сондықтан,

заманауи мамандар ақпараттық технологиялар мен коммуникациялар саласында мамандандырылған адамдар интернет-технологияларда, атап айтқанда веб-технологияларда, соның ішінде HTML және CSS-ні пайдалана отырып, веб-сайттарды құру жеткілікті түрде білуі керек. Бұл технологияларды игере отырып, интернетте өзіңіздің ақпараттарыңызды орналастыру мүмкіндігімен қатар, технологияларды игеру сізге дүниежүзілік желінің жоғары деңгейінде орналасқан көптеген ақпараттық ресурстарын пайдалануға көмектеседі.

Веб-сайтты зерттеу интернеттегі сайттар құрылымын неғұрлым жақсы түсінуге, интернет ішінде және жеке веб-сайттар арасында байланысты неғұрлым тиімді пайдалану мүмкіндігін береді. веб-сайтыңызды іздеуге жүйелеріне оңайлату, сондай-ақ оны интернет-каталогтарда тіркеу арқылы сіздің сайтыңызды басқа да адамдар көре алады.

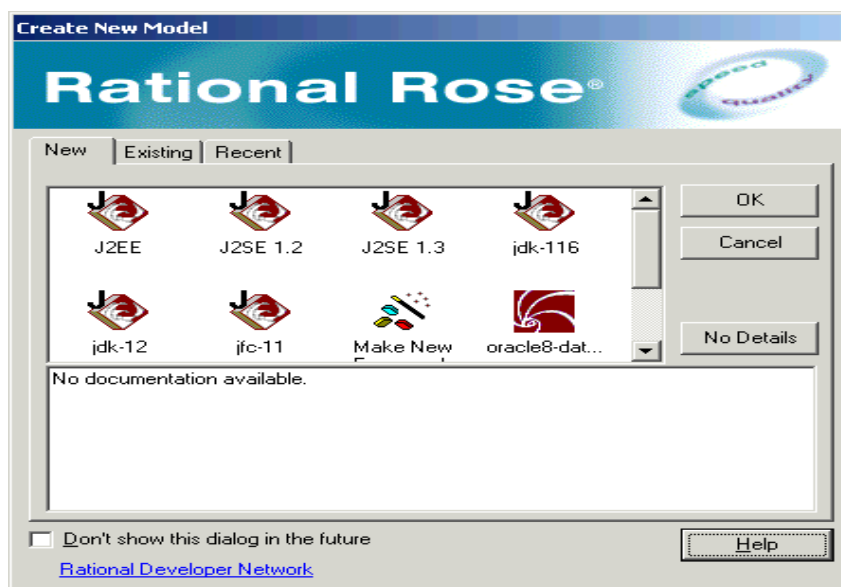
2 Жобалау бөлімі

2.1 Унифицирленген моделдеу тілі

UML – бағдарламалық қамтамасы графикалық объекті модельдеп, сипаттау үшін, бизнес-процесті модельдеу болып табылады. UML моделі программалық жүйені анықтау, жобалау негізінде құрылған. UML – бағдарламалау тілі емес, бірақ UML моделі негізінде кодты жасауы мүмкін.

2.2 Rational Rose құралы

Rational Rose – визуалды модельдеу үшін және кәсіпорын деңгейіндегі бағдарламалық жасақтама қосымшаларын компонентті құрастыру үшін және де бизнес-процестерді сипаттау үшін арналған (UML) бағдарламалық жасақтама құралы келесі 2.1-суретте көрсетілген.

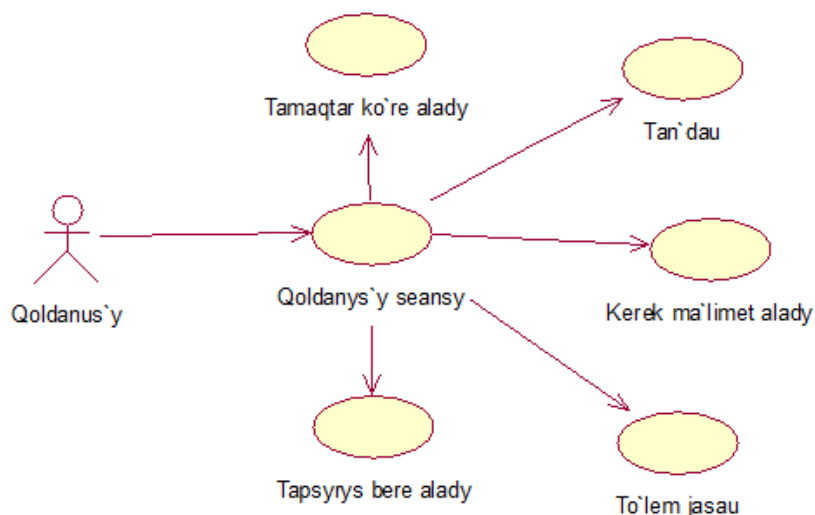


2.1-сурет – Rational Rose құралы

2.3 Прецеденттер диаграммасын құру

Актерлер мен прецеденттер арасындағы қарым-қатынасты көрсететін диаграмма және жүйенің тұжырымдамалық деңгейде сипатталуына мүмкіндік беретін, үлгелі заң моделінің бөлігі. Жүйенің жасалатын жұмысы онымен қалай айналысатынына және не қажет болуына байланысты болады. Объект моделі

өзгергіштік тараудағы прецедент моделдерімен байланыс жасайды. Модельдерді құру негізінде қолдана алады. Орналасқан жүйелерден басқа жерлерде болуы мүмкін пайдаланушы зерттеуінен, сонымен қоса ақпараттың қимыл бөліктерін ашу жолынан басталады. Осындай тұрыптан ақпарат құрылымы пайда болады, яғни, қажет ететін іс-әрекеттен болады. Прецедент диаграммасы 2.2-суретте көрсетілген.

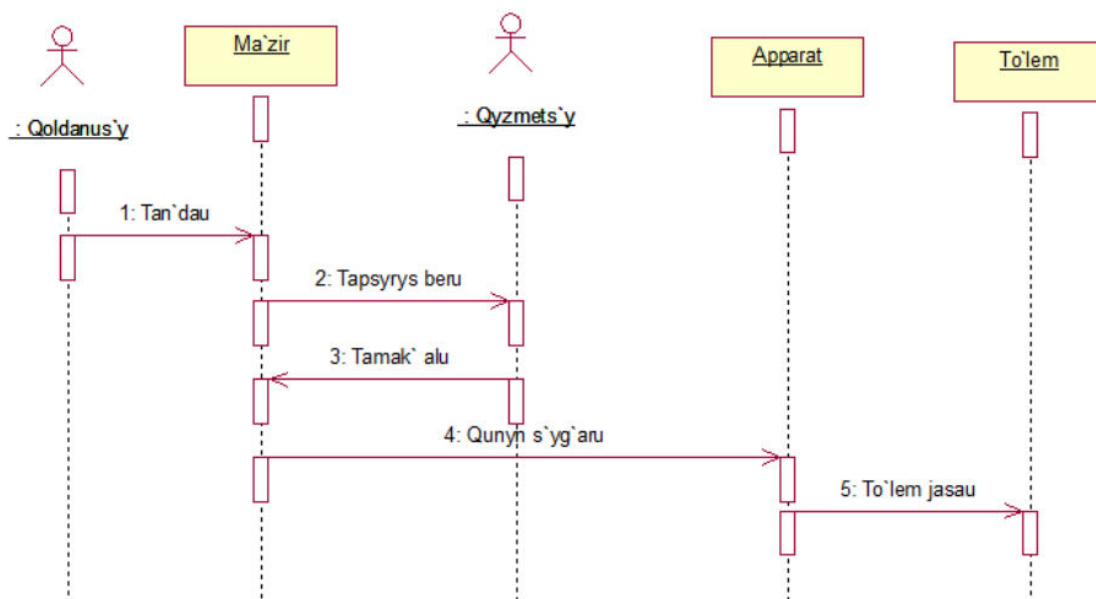


2.2-сурет – Қолданушының прецеденттер диаграммасы

2.4 Тізбек диаграммасын құру

Тізбек диаграммасы уақыт бойынша берілген объектілердің өзара ұйымдасу операцияларының жасалу ретін безендіреді және жоспарлармен қарастырылған мүмкіндіктердің орындалу процесіндегі өзара алмасатын объектілердің хабарлама тізбектерінің реттелген класстар және объектілер бейнеленеді.

UML талаптарына байланысты объектінің тізбектелген диаграммасында тіктөртбұрыш кейпінде көрсетіледі. Объектіні 3 түрлі есіммен атауға болады: тек оның атауын көрсетіп кету қажет, объект пен класстың атауын беру қажет, не класстың атауымен шектелу керек. Тізбектер диаграммасы 2.3-суретте көрсетілген.

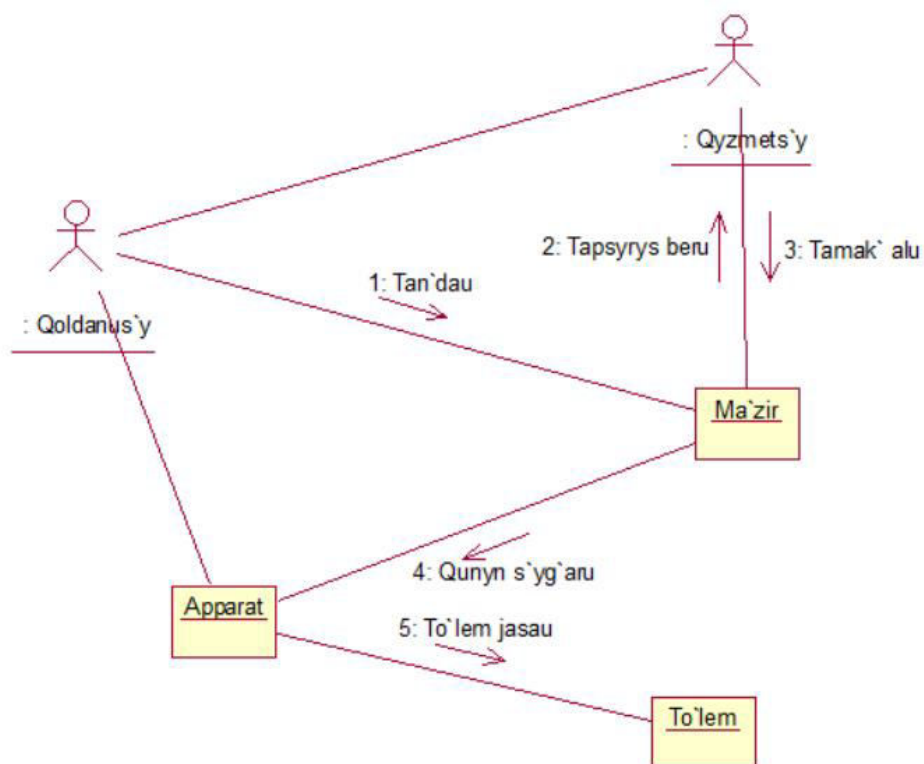


2.3-сурет – Тізбек диаграммасы

2.5 Кооперация диаграммасын құру

Кооперация диаграммасы – бұл хаттарды жібереді және қабылдайды. Субъектіні құрылымдық жағынан әрекеттесу диаграммасы. Кооперация диаграммасы жүйе жұмысы кезінде субъектілер арасында баяндайды. Мұндай диаграмма жүйе жоспары сценарийін бейнелеп көрсетеді. Объект атының астын сызылады, ал қасиеттері көрсетіледі.

Кооперация сызбасы жүйе жұмысының барысында субъектіні өзара бейнелейді. Сызба жүйенің тәртібі жоспары модельдеп көрсетеді. Кооперация диаграммасы 2.4-суретте көрсетілген.

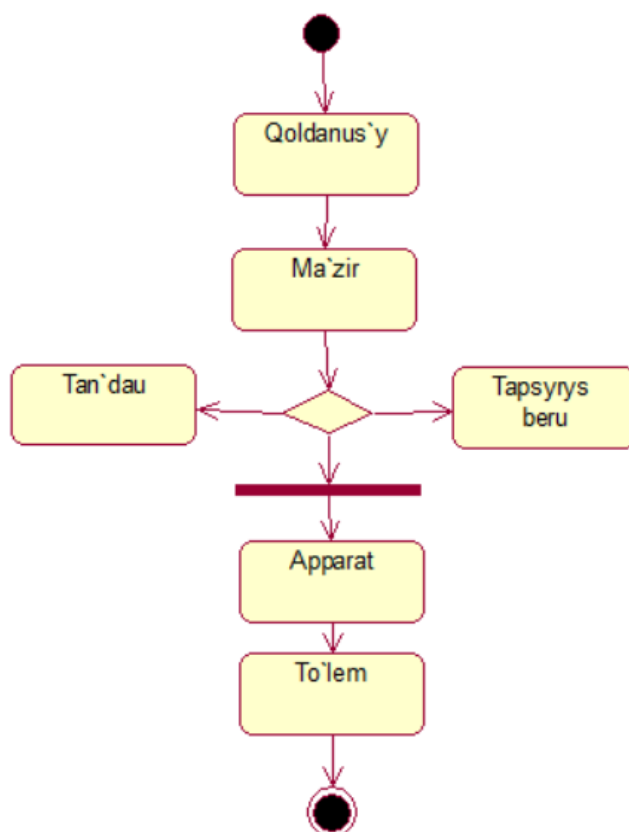


2.4-сурет – Кооперация диаграммасы

2.6 Күй диаграммасын құру

Күй (State) – бұл кез келген бір әрекеттер орындау кезінде белгіленген әрекет орындау шарты болады. Объектінің күйі класы бірнеше немесе бір деректермен бейнеленеді.

Күй диаграммасы – жиынның бір қалыптан басқасына ауысуын айыратын, және іс-әрекеттері қалпыларын алмастырумен ескертілінген, оқиғалардың, болмаса хабарлаулардың объекттік жағдайларының графикалық түрдегі ұсынысы. Күй диаграммасы 2.5-суретте көрсетілген.

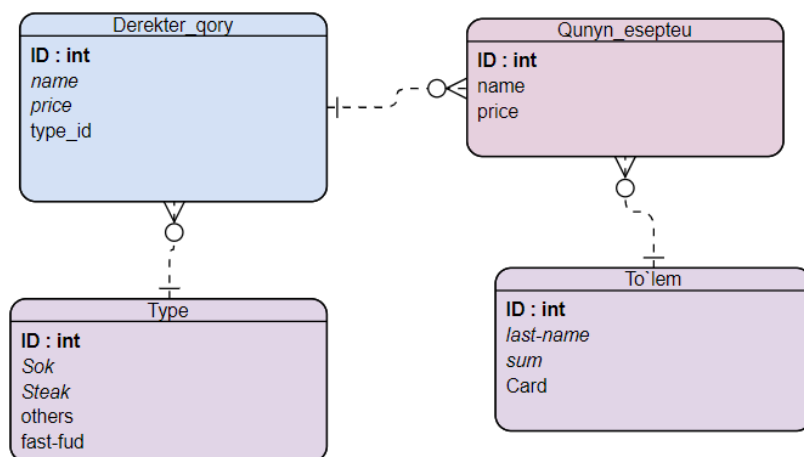


2.5-сурет – Күй диаграммасы

2.7 ER диаграммасын құру

ER-модель базалық деректерді жобалауды қолданылады. Оның көмегімен кілттерді белгілеуге және байланысын көрсетуге болады, олар осы қасиеттерге ие. Деректерді жасақтау кезінде ER үлгісі таңдалған деректер үлгісіне негізделген нақты дерекқор схемасына түрленеді.

ER-моделі – кез-келген графикалық тәсілін анықтайды және оны кескін ретінде көрсетеді. ER-модельді визуализациялауға болатын стандартты графикалық белгілер ретінде, «субъект-қатынас диаграммасы» ұсынылды. ER диаграммасы 2.6-суретте көрсетілген.



2.6-сурет – ER диаграммасы

3 Жобалау бөлімі

3.1 Python программалау тілі

Python – қарапайым, икемді және өте танымал, ол заманауи дамудың барлық салаларында қолданылады. Оған қоса, сіз веб-қосымшаларды жасай аласыз, ойын жаза аласыз, жүйелік әкімшілік тапсырмаларды автоматтандыруға және тағы басқаларды жасай аласыз. 1991 жылы Г.В.Россум Python тілін шығарған болатын. Бұл тіл басқа тілдерден ерекшелігіне мыналар жатады:

1. тілді үйрену оңай және алғашқы программа ұзақ уақытқа созылмайды
2. бірнеше кітапханалары (SciPy, NumPy)
3. Web framework-тары (Django, Flask т.б)

Тілді енді үйреніп жатқан тұлғаларға оңай болып табылады.

Python тілі 1980-ші жылдарда Голландиялық Г.В.Россум бастаған болатын. Сол кезде Г.В.Россум ABC тілін жасауды қарастырып жүрген болатын. 1991 жылы ақпан айында alt.sources жаңалықтарында түпкі кодын жариялады. Python-ның шығу кезіңі осы уақыттарда болатын.

Ван Россомдың MCSscience орталығында жұмыс істеген соңғы нұсқасы Python 1.2 болды. 1995 жылдан бастап Ван Россум Вирджиниядағы Рестон қаласында Python компаниясында жұмыс істеді, ол жерде бірнеше тілдер бар.

Python3 (Py3K) бұл іргелі кемшіліктерді жоюға арналған. Бұл өзгертулер 2 нұсқасымен толық кері қарай сыйысымдылықты сақтай отырып жасалмады, сондықтан негізгі нұсқа нөмірін өзгертуді талап етті. Python 3-нің дамуының негізгі қағидасы: «қайталанатын функционалдылықты азайту және оның орындаудағы ескірген жолдарын жою» [10].

3.1.2 Python программалау тілін қолдану ерекшелігі

Көптеген адамдарға Python тілінің артықшылығы оқуға түсінікті түрде болуы, және оның сапасы жоғарылығы оны оны басқа тілдерден ерекшелендіреді. Кодты оқуға жеңілдірек, яғни жазылған кодты пайдалану жеңіл түрде түсіне аласыз. Python тілінде жазылған кодты, тіпті оны жасауға қатыспаған адамдар үшін түсіну процессін жеңілдетеді. Python кодын пайдалануды заманауи механизмдер қолдайды.

3.2 Tensorflow кітапханасы

TensorFlow – бұл адамның қабылдау сапасы сияқты нәтижеге жету үшін кескіндерді автоматты түрде табуға және жіктеуге, нейрондық желіні құруға

және оқытуға арналған проблемаларды шешу үшін Google әзірлеген компьютерлік оқытуға арналған ашық бағдарлама кітапханасы.

TensorFlow 2015 жылдың 9 қарашасында адамдар үшін тегін қол жетімді болды. TensorFlow – бұл машина оқыту жүйесі Google Brain екінші ұрпағы. TensorFlow графикалық процессорларда жалпы мақсаттағы есептеулерді қолдау үшін CUDA архитектурасына сүйене отырып, көптеген процессорлар мен процессорлар сияқты параллель процессорларда жұмыс істей алады. TensorFlow 64-биттік Linux, macOS, Windows және ұялы есептеуіш платформаларында, соның ішінде Android және iOS үшін қол жетімді. Графикалық процесстер кодты немесе машиналық үйретуді жіберген кезде CUDA және CUDnn бұлар Nvidia компаниясының бағдарламалық қамтамалары болып табылады. Бұл қамтаманы қолданған жағдайда процессті әлдеқайда жылдамдатады [9].

3.3 Keras кітапханасы

Keras – бұл python-да жазылған ашық нейрондық кітапхана. Бұл Deeplearning, TensorFlow және Theano жоғары кітапханалардың баптамаларының үстем құрылысы. Ол тереңдетілген оқыту желілерімен жедел жұмысқа бағытталған, модульдік және кеңеюге арналған. Ол ONEIROS зерттеу жұмыстарының бөлігі ретінде құрылды. Оның негізгі авторы – Франсуа Чолле (Google инженері).

Google Keras-ті негізгі TensorFlow кітапханасында қолдайтыны анықталды, алайда Scholle Keras-ты бөлек қосуға ұсынды, өйткені тұжырымдамаға сай, Keras – бұл компьютерлік оқытудың түпкілікті жүйесі емес, ал интерфейс болып табылады. Keras компьютерлік есептеуіш ретінде пайдаланылатын ғылыми есептеу кітапханасына қарамастан, нейрондық желілерді құруға мүмкіндік беретін жоғары деңгейлі, интуитивті абстракция жиынтығын ұсынады. Microsoft Keras және CNTK төмен деңгейлі кітапханаларға қосу бойынша жұмыс жасады [8].

3.4 Flask веб-беті

Flask – фреймворк құралдар жиынтығын пайдаланып, python бағдарламалау тілінде веб-қоосымшасын жасауға арналған. Python пакеттері, Flask сияқты, жалпы репозиторийге қол жеткізе аламыз, оларды шығарып, оларды орнатуға болады. PyPI пакетінің орнатылуы өте оңай, өйткені Python құрамында pip деп аталатын құрал бар, ол осы жұмысты pip install flask командасы арқылы орната аламыз. Қарапайым көрсететін веб-бағдарламаның мысалы 3.1-суретте көрсетілген.

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
if __name__ == "__main__":
```

3.1-сурет – Қарапайым Flask коды

3.5 NumPy кітапханасы

Python интерпретаторлық тілі, математикалық алгоритм көбінесе C++ және Java тілдерінде баяу жұмыс істейді. Сондықтан NumPy кітапханасы жасауға тура келді. NumPy – бұл Python тілінің кеңейтілген түрінде, өлшемді массивтер мен матрицалар қолдау жасауға және осы массивпен жұмыс істеуге арналған, жоғары деңгейлік математикалық функциялар жиынтығының кітапханасы болып табылады.

3.6 OpenCV компьютерлік көзқарас кітапханасы

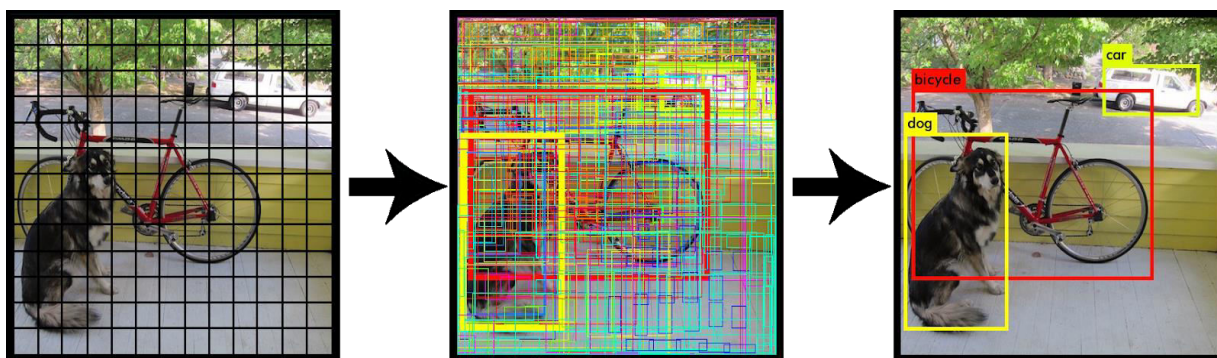
OpenCV – бұл нақты уақытта компьютерлік көзқарас бағдарламалау функциялар кітапханасы. Библиотекада 2500-ден астам оңтайландырылған алгоритмдермен жабдықталған, олар өз кезегінде толық жиынтық және қазіргі заманғы компьютерлік алгоритмдер мен машина оқулықтарына қосылады.

Кітапхана Windows, Linux, Android және Mac OS қолдауымен C ++, C, Python, Java және MATLAB секілді көптеген танымал тілдерде қол жетімді. OpenCV C ++ тіліне жазылған және шаблон интерфейсі бар, ол STL контейнерлерінің мәселесі бойынша жұмыс істейді.

3.7 YOLO моделі

YOLO – нақты уақыт режимінде объектіні анықтаудың алдыңғы қатарлы жүйесі. Ресми сайтында YOLOv2 және Tiny YOLO екі түрлі деректер жиынтығымен дайындалған: VOC 2007 + 2012 және COCO. Интернеттегі машиналық үйретуге арналған конфигурациялар мен деректер жинақтары үшін қосымша параметрлерді табуға болады (мысалы, YOLO9k). Қол жетімді

опциялардың кең спектрі арқасында, қажеттіліктеріңізге сай келетін нұсқаны таңдай аласыз. Мысалы, Tiny YOLO смартфондарда тез жұмыс істей алатын ең ықшам нұсқасы. YOLO моделінің детекттеу мысалы 3.2-суретте көрсетілген.



3.2-сурет – Yolo детекциясы

Yolo моделі DarkNet – ке негізделген нейрондық желілер үшін әзірленген, біз үшін осы шешімдердің кейбір ерекшеліктері жарамайды. DarkNet үйрену коэффициенттерін салмақ түрінде сақтайды және әртүрлі платформаларда түрлі әдістермен тануға болатын тәжірибие жүзінде көре аламыз. Бұл мәселе кез келген бір автоматтандыру салаларында қолдануы мүмкін, сондықтан керекті үлгі дайындау арқылы оны басқа жабдыққа пайдалану керек [5].

3.8 DarkFlow кітапханасы

Darkflow – бұл yolo моделінің C тілінде үйретілген салмағын python бағдарламалау тіліне келтіреді. Darkflow-ды бағдарламашылар орнатуға өте оңай етіп жасаған. Ол үшін репозиторийын жүктеп алып, `python3setup.pybuild_ext – inplace` терменалдан осы команданы орындаймыз және оны `import` жасаймыз[6].

3.9 Нейрондық желі

Қазіргі кезде нейрондық желілердің көптеген түрі бар. Нейрондық желілер оқыту арқылы үлгісі алынады. Нейрондық желілер адамның миына ұқсас аналитикалық есептеулерді қажет ететін күрделі мәселені шешу арқылы қолданылады. Нейрондық желілердің ең көп қолданылатын түрлері:

- классификациялау - деректерді параметрлер бойынша бөлу;
- болжау - келесі қадамды болжауы мүмкін;
- тану – қазіргі уақытта нейрондық желілердің ең кең танымал мәселесі

[4].

3.9.1 Машиналық оқытылу

Машина оқытылу – бұл деректерден күрделі функциялардан үйренетін және болжамдар жасайтын алгоритмдерді жасау. Үйрену - тәжірибе немесе оқу арқылы білім мен дағдыларды игеру. Машиналық оқыту уақыт өте келе дамып келе жатқан нәрсе. Машиналық оқытылу мына жерлерде қолданылады:

- ауруханаларда дәрігерлер үшін әр түрлі болжамдар жасау;
- әлеуметтік желілерде танысу сайтының үйлесімділігін болжау;
- кредиттік карталарда қаржыны алаяқтық мәселелерге болжау жасау.

1952 жылы Артур Сэмюэль алғашқы дойбы ойынын жасады және ойынға өздігінен үйрену мүмкіндігін қосты. Яғни компьютерге дойбы ойнауды үйреткен болатын. Артур Самуэль жасанды интеллект саласына ізашар деп атасақ болады. Ғалымдар көптеген алгоритмдермен жұмыс жасап, біраз жобаларды жасады. 1959 жылы олар алғашқы нейрондық желіні жасап шығарды. 60-70 жылдары жасанды интеллект өте жақсы дами бастады. Компьютерлер мықты шахматшылады жеңе бастады, ал жасанды интеллект саласын зерттеу ге деген көп ақша бөліне бастады. 1973 жылдары бәрі ойлағандай жеңіл емес екені туралы айтыла бастады. Сол жылдары Лайтхал "Жасанды интеллект: Жалпы шолу" баяндамасын шығарды. Бұл баяндамада Жасанды интеллектке деген біраз ойлар айтылды. Осы баяндамадан кейін жасанды интеллектке деген ақша азайды [1].

3.10 Веб бетті құру және машиналық оқыту жұмысын іске асыру

Бұл проекттің толық құрылымы машиналы үйрету және Веб қосымшасынан тұрады. Веб-сайтпен машиналық үйретудің қызметі – жұмысшыларға ыңғайлап, жылдамдатып, жеңілдетіп тауардың құнын шығару. Тек жұмысшыларға ғана емес, қарапайым қолданушыларға кез келген зат алудың жолын тездету болып табылады.

3.11 Кескін ішіндегі объектіні табу

Кескін ішіндегі объектіні тану көптеген қосымшаларда оның пайдалы болуына байланысты зерттеудің маңызды саласына айналғаны ақиқат. Мұндай объектіні тану жүйесі кез келген жағдайларда автоматтандыру арқылы адамдарға жеңілдік жасау түрінде қолданылуы мүмкін. Қазіргі кезде бірнеше объектілерді табуға арналған алгоритмдер бар. Олардың арасында ең танымалы

YOLO. YOLO – нақты уақытта объектілерді анықтауға арналған салмағы. Бұл моделі C тілінде жазылған. Алайда объектіні табу өте қиын мәселенің бірі. Сол сияқты қиындықтарға толығырақ үңіле кетсек.

3.12 Жаңа жиын оқыту үшін деректердің жаңа сыныптары

YOLO жиынын үйретуде деректер қоры әр түрлі сыныптардан тұрады. Олардың әрқайсысы суреттер және оның координаттары болады.

Darknet – бұл ресми түрде жарияланған және YOLO салмағын іске асыру үшін құрылған моделі. Бұл C тілінде жазылған және есептеуді GPU бейне картасында қолдайды. Darknet-тің txt форматы қарапайым түрде 3.3-суретте көрсетілген.

```
[category number] [object center in X] [object center in Y] [object width in X] [object width in Y]
```

3.3-сурет – Кескін координаттары

3.13 Кескін координаттарын алу құралы

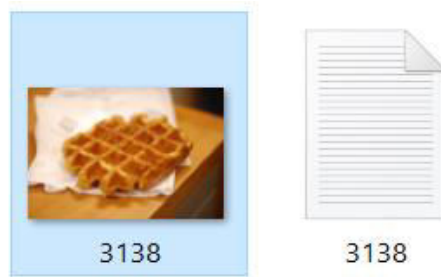
Кескін координаттарын алудың бірнеше құралдары бар. Бұлардың ішінде VBox құралын пайдалану арқылы кескін ішіндегі объектілерді әр түрлі сыныптарға нөмір жазу арқылы орындалады. Бұл құрал келесі баптаулар арқылы жұмыс істейді:

- суреттерді белгілеу үшін / Images / 001, / Images / 002 бұл құралдың стандартты баптаулары. Егер сіз суреттерді басқа жерде белгілегіңіз келсе, кодты өзгертуіңіз керек;

- папканың орналасқан жерін көрсетеміз;

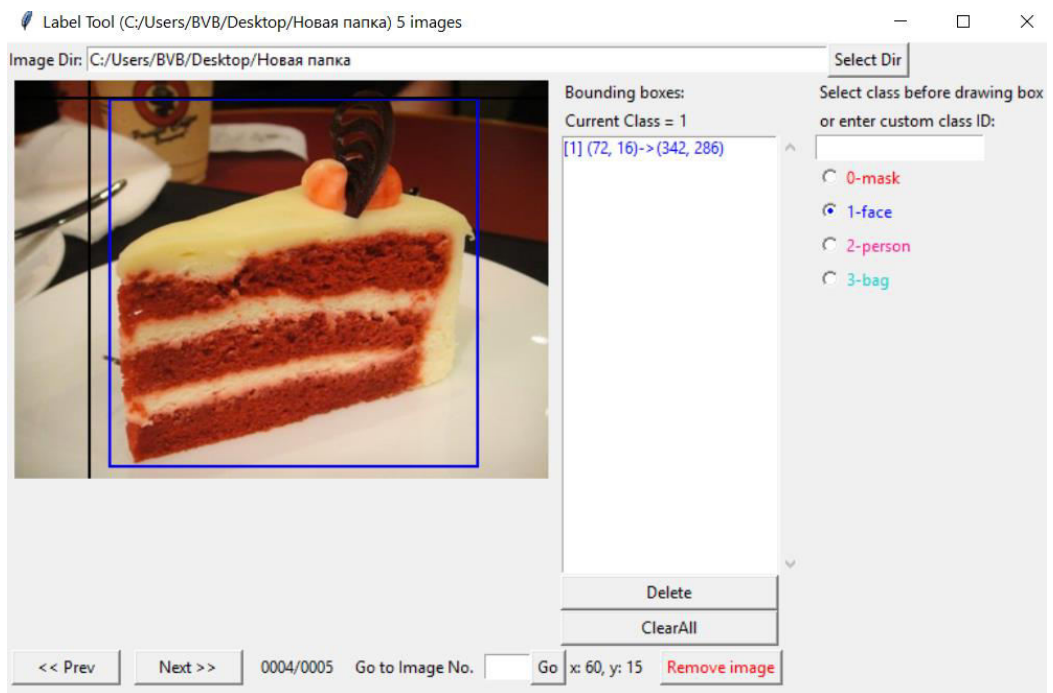
- координаталарын көрсету үшін тышқан сол батырмасын басып, екінші рет координатасын созып тышқан сол батырмасын басамыз;

- келесі батырмасын басқан жағдайда сол кескіннің txt форматы сол папкаға сақталады ол 3.4-суретте көрсетілген.



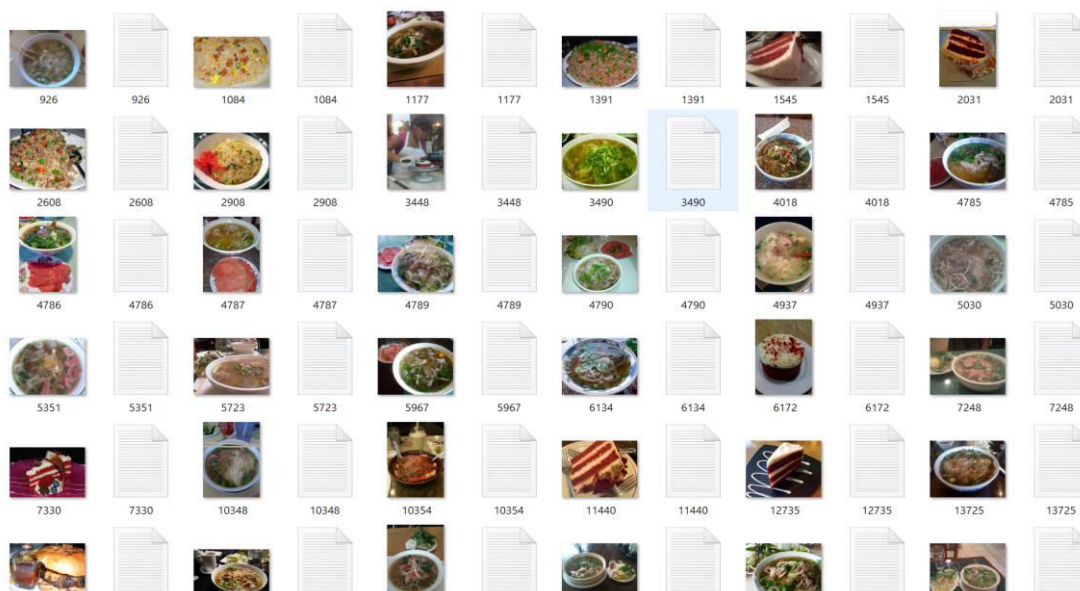
3.4-сурет – Кескін және оның координатасы

ВBox Label құралының жалпы интерфейсі 3.5-суретте көрсетілген.



3.5-сурет – VBox Label Tool құралы

Осы құралмен барлық кескіндердің координаталарын көрсеткен соң оларды бір папкаға кескін және txt салуымыз керек. Бұл кескіндер 3.6-суретте көрсетілген.

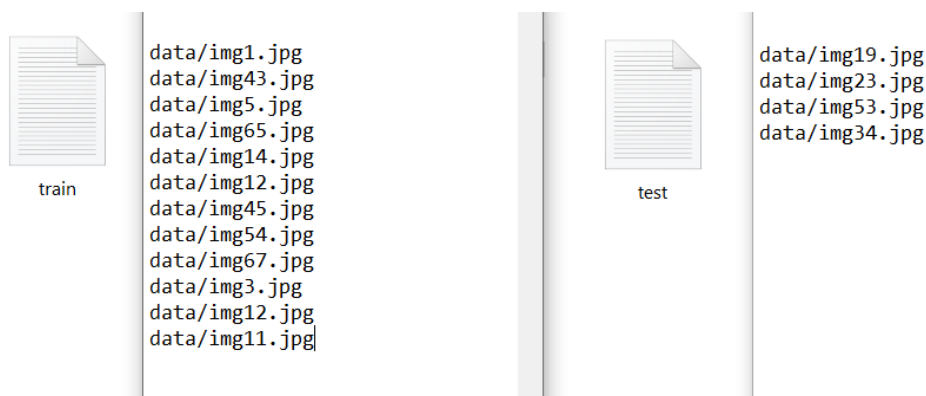


3.6-сурет – Кескін және координатасы

Әр сурет қасында міндетті түрде txt туруы керек.

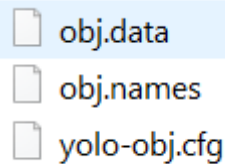
3.14 YOLO v2 баптауларын жасау

Енді YOLO салмағына суреттердің белгілі бір пайызын тесттеуге және үйретуге береміз. Мен осы суреттерді қолмен жазбау үшін алгоритм жаздым. Ол жерде сурет 10% пайызын test.txt тесттеуге файлына, қалған 90% пайызын train.txt үйретуге арналған суреттердің тізімін жаздым. Бұл 3.7-суретте көрсетілген.



3.7-сурет – Train және test-ке бөлу

YOLOv2 арналған белгілі үйренуге арналған әр түрлі форматта баптауларын құрамыз. Олар 3.8-суретте көрсетілген.



3.8-сурет – Баптау файылдары

Бұл жерде obj.data баптау файылы іші 3.9-суретте көрсетілген. Ал obj.names файылда кескін аттары тұрады.

```
classes = 4
train = train.txt
valid = test.txt
names = obj.names
backup = backup/|
```

3.9-сурет – Obj.data файылы

Келесі obj.names файылының ішінде әр класстың аты жазылады, яғни тамақ аттары. Бұл 3.10-суретте көрсетілген.

```
rice
burger
cake
pho|
```

3.10-сурет – obj.names файылы

Соңғы файл obj.cfg бұл файл ішінде стандартты түрде жазылған командалар, олардың ішінде кейбір командаларға өзгерістер енгіземіз. 3.11-суретте керекті баптаулар көрсетілген ол жерде класстар саны, кескін размері және графикалық картаға жіберілетін көлемін көрсетеміз.

```
# Testing
batch=1|
# Training
# batch=64
width= 320
height = 320
classes = 4
```

3.11 –сурет – Yolo-obg.cfg файылы

Деректер қорымыз дайын, енді бізге үйретілген darknet.conv.23 салмағы арқылы үйретуге жібереміз. Мына команда арқылы ./darknetdetector train objес/obj.data objес/obj.cfg darknet.conv.23 орындалады. Бұл жерде басты obj.data және obj.cfg. Ол 3.12-суретте көрсетілген.

```

PS C:\Users\ibcn\Documents\darknet\windows\darknet\build\darknet\x64> ./darknet.exe detector train cfg/obj.data cfg/yolo-obj.cfg darknet19_448.conv.23
yolo-obj
layer   filters  size      input                output
0 conv  32  3 x 3 / 1  416 x 416 x 3  ->  416 x 416 x 32
1 max   2  2 x 2 / 2  416 x 416 x 32  ->  208 x 208 x 32
2 conv  64  3 x 3 / 1  208 x 208 x 32  ->  208 x 208 x 64
3 max   2  2 x 2 / 2  208 x 208 x 64  ->  104 x 104 x 64
4 conv  128 3 x 3 / 1  104 x 104 x 64  ->  104 x 104 x 128
5 conv  64  1 x 1 / 1  104 x 104 x 128 ->  104 x 104 x 64
6 conv  128 3 x 3 / 1  104 x 104 x 64  ->  104 x 104 x 128
7 max   2  2 x 2 / 2  104 x 104 x 128 ->  52 x 52 x 128
8 conv  256 3 x 3 / 1  52 x 52 x 128   ->  52 x 52 x 256
9 conv  128 1 x 1 / 1  52 x 52 x 256   ->  52 x 52 x 128
10 conv 256 3 x 3 / 1  52 x 52 x 128   ->  52 x 52 x 256
11 max   2  2 x 2 / 2  52 x 52 x 256   ->  26 x 26 x 256
12 conv 512 3 x 3 / 1  26 x 26 x 256   ->  26 x 26 x 512
13 conv 256 1 x 1 / 1  26 x 26 x 512   ->  26 x 26 x 256
14 conv 512 3 x 3 / 1  26 x 26 x 256   ->  26 x 26 x 512
15 conv 256 1 x 1 / 1  26 x 26 x 512   ->  26 x 26 x 256
16 conv 512 3 x 3 / 1  26 x 26 x 256   ->  26 x 26 x 512
17 max   2  2 x 2 / 2  26 x 26 x 512   ->  13 x 13 x 512
18 conv 1024 3 x 3 / 1  13 x 13 x 512   ->  13 x 13 x 1024
19 conv 512 1 x 1 / 1  13 x 13 x 1024  ->  13 x 13 x 512
20 conv 1024 3 x 3 / 1  13 x 13 x 512   ->  13 x 13 x 1024
21 conv 512 1 x 1 / 1  13 x 13 x 1024  ->  13 x 13 x 512
22 conv 1024 3 x 3 / 1  13 x 13 x 512   ->  13 x 13 x 1024
23 conv 1024 3 x 3 / 1  13 x 13 x 1024  ->  13 x 13 x 1024
24 conv 1024 3 x 3 / 1  13 x 13 x 1024  ->  13 x 13 x 1024
25 route 16
26 conv 64  1 x 1 / 1  26 x 26 x 512   ->  26 x 26 x 64
27 reorg / 2  26 x 26 x 64    ->  13 x 13 x 256
28 route 27 24
29 conv 1024 3 x 3 / 1  13 x 13 x 1280  ->  13 x 13 x 1024
30 conv 30  1 x 1 / 1  13 x 13 x 1024  ->  13 x 13 x 30
31 detection
Loading weights from darknet19_448.conv.23...Done!
Learning Rate: 0.001, Momentum: 0.9, Decay: 0.0005
Resizing
448
  
```

3.12-сурет – Yolo-ны үйретуге жіберу

Үйрету өте көп уақытар аралығында өтеді. Ол бірнеше күн емес апталарға созылуы мүмкін. Сондықтан мен үйретуді серверде nohup арқылы жібердім. Ол жерде тіпті компьютер өшкен жағдайда да тоқтамай жіберіле береді. Үйрету процесі нақты режимде көре алмаймыз, бірақ txt файлға жазу арқылы көруге болады. Бұл 3.13-суретте көрсетілген.

```

log_textd — Блокнот
Файл Правка Формат Вид Справка
layer   filters  size      input                output
0 conv  32  3 x 3 / 1  512 x 320 x 3  ->  512 x 320 x 32
1 max   2  2 x 2 / 2  512 x 320 x 32  ->  256 x 160 x 32
2 conv  64  3 x 3 / 1  256 x 160 x 32  ->  256 x 160 x 64
3 max   2  2 x 2 / 2  256 x 160 x 64  ->  128 x 80 x 64
4 conv  128 3 x 3 / 1  128 x 80 x 64   ->  128 x 80 x 128
5 conv  64  1 x 1 / 1  128 x 80 x 128  ->  128 x 80 x 64
6 conv  128 3 x 3 / 1  128 x 80 x 64   ->  128 x 80 x 128
7 max   2  2 x 2 / 2  128 x 80 x 128  ->  64 x 40 x 128
8 conv  256 3 x 3 / 1  64 x 40 x 128   ->  64 x 40 x 256
9 conv  128 1 x 1 / 1  64 x 40 x 256   ->  64 x 40 x 128
10 conv 256 3 x 3 / 1  64 x 40 x 128   ->  64 x 40 x 256
11 max   2  2 x 2 / 2  64 x 40 x 256   ->  32 x 20 x 256
12 conv 512 3 x 3 / 1  32 x 20 x 256   ->  32 x 20 x 512
13 conv 256 1 x 1 / 1  32 x 20 x 512   ->  32 x 20 x 256
14 conv 512 3 x 3 / 1  32 x 20 x 256   ->  32 x 20 x 512
15 conv 256 1 x 1 / 1  32 x 20 x 512   ->  32 x 20 x 256
16 conv 512 3 x 3 / 1  32 x 20 x 256   ->  32 x 20 x 512
17 max   2  2 x 2 / 2  32 x 20 x 512   ->  16 x 10 x 512
18 conv 1024 3 x 3 / 1  16 x 10 x 512   ->  16 x 10 x 1024
19 conv 512 1 x 1 / 1  16 x 10 x 1024  ->  16 x 10 x 512
20 conv 1024 3 x 3 / 1  16 x 10 x 512   ->  16 x 10 x 1024
21 conv 512 1 x 1 / 1  16 x 10 x 1024  ->  16 x 10 x 512
22 conv 1024 3 x 3 / 1  16 x 10 x 512   ->  16 x 10 x 1024
23 conv 1024 3 x 3 / 1  16 x 10 x 1024  ->  16 x 10 x 1024
24 conv 1024 3 x 3 / 1  16 x 10 x 1024  ->  16 x 10 x 1024
25 route 16
26 conv 64  1 x 1 / 1  32 x 20 x 512   ->  32 x 20 x 64
27 reorg / 2  32 x 20 x 64    ->  16 x 10 x 256
28 route 27 24
29 conv 1024 3 x 3 / 1  16 x 10 x 1280  ->  16 x 10 x 1024
30 conv 30  1 x 1 / 1  16 x 10 x 1024  ->  16 x 10 x 30
31 detection
  
```

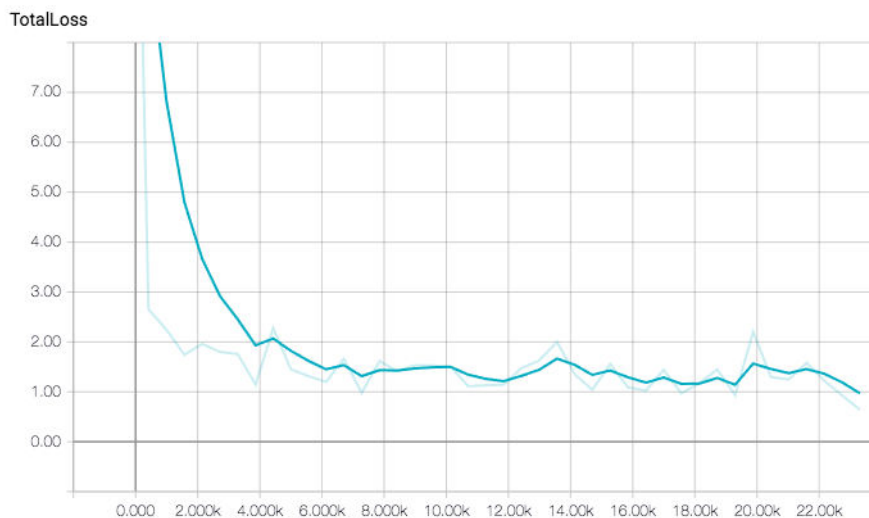
3.13-сурет – Nohup форматындағы файл

Бұл үйретуге кеткен уақыт 24 сағат аралығында өтті. Нақты нәтиже алу үшін мен 7400 итерациядан өткіздім. Осы уақыт аралығында менде 46 салмағы түсті, ол 3.14-суретте көрсетілген.

```
(base) [sustudent4@supermicro2 weights]$ ls
yolo-food_1000.weights  yolo-food_2400.weights  yolo-food_3900.weights
yolo-food_100.weights  yolo-food_2500.weights  yolo-food_4000.weights
yolo-food_1100.weights  yolo-food_2600.weights  yolo-food_400.weights
yolo-food_1200.weights  yolo-food_2700.weights  yolo-food_4100.weights
yolo-food_1300.weights  yolo-food_2800.weights  yolo-food_4200.weights
yolo-food_1400.weights  yolo-food_2900.weights  yolo-food_4300.weights
yolo-food_1500.weights  yolo-food_3000.weights  yolo-food_4400.weights
yolo-food_1600.weights  yolo-food_300.weights   yolo-food_4500.weights
yolo-food_1700.weights  yolo-food_3100.weights  yolo-food_4600.weights
yolo-food_1800.weights  yolo-food_3200.weights  yolo-food_500.weights
yolo-food_1900.weights  yolo-food_3300.weights  yolo-food_600.weights
yolo-food_2000.weights  yolo-food_3400.weights  yolo-food_700.weights
yolo-food_200.weights   yolo-food_3500.weights  yolo-food_800.weights
yolo-food_2100.weights  yolo-food_3600.weights  yolo-food_900.weights
yolo-food_2200.weights  yolo-food_3700.weights
yolo-food_2300.weights  yolo-food_3800.weights
```

3.14-сурет – Yolo салмақтары

Оның ішінде нақтылығы жоғарысын табу үшін бір алгоритм жаздым. Бұл алгоритмде әр итерация сайын avg мәні төмендеп немесе жоғарылайды, соның ішінде ең төмені нақтылығы жоғары болады. 3.15-суретте графикалық түрінде салмақтардың нақтылығы көрсетілген.

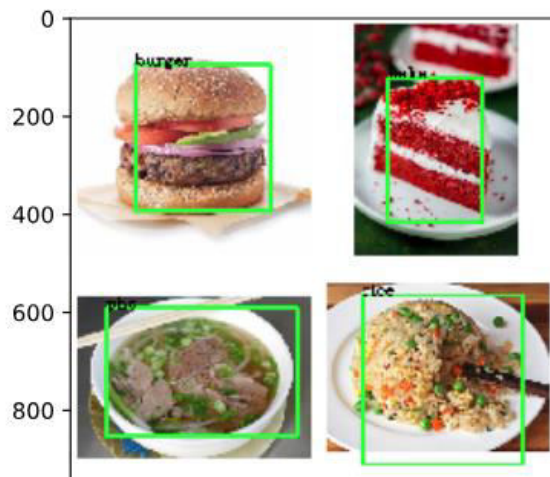


3.15-сурет – Салмақтардың нақтылығы

3.15.1 Үйретілген моделдің нәтижесі

Модел үйретіліп болған соң, салмақты алып, тесттеу арқылы жүзеге асырайық. 3.16-суретте тамақтарды тану арқылы детектеу және олардың құнын есептеу алгоритімі жазылған.

```
Rice price: 400тг  
Burger price: 1050тг  
Cake price: 350тг  
Pho price: 450тг  
Сумма = 2250
```

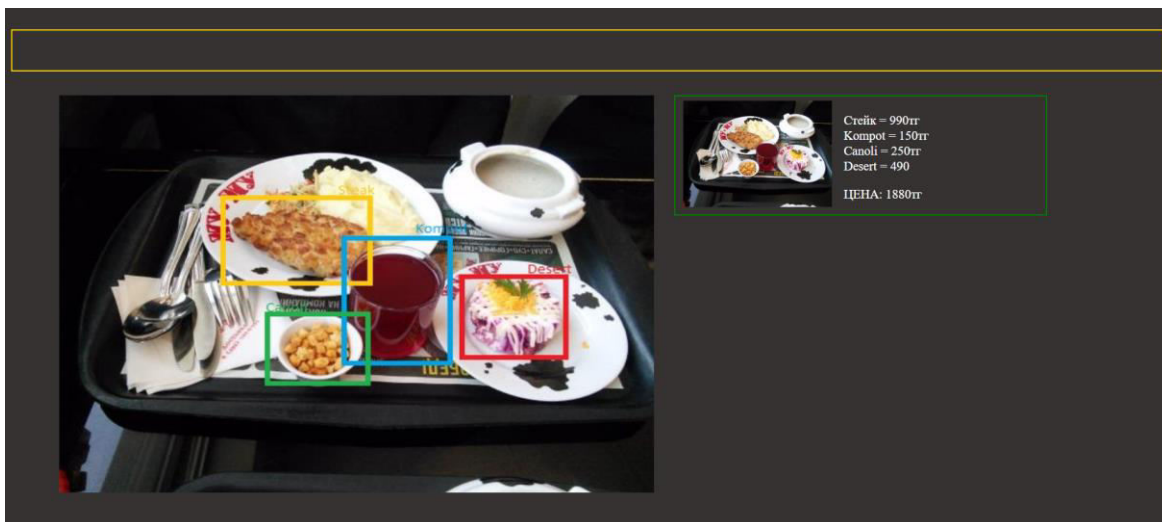


3.16-сурет – Нәтижені тестілеу

3.15.2 Қолданушы интерфейсі

Бұл проектіні жасаудағы түпкі мақсаты тұтынушының тапсырыс беру жүйесін автоматтандыру болып табылады. Веб-бетті қарауға кез-келген браузер жарайды, Opera, Explorer, немесе Google Chrome. Веб-бет тоқтамай жұмыс жасап тұру үшін веб-бет толықтай веб-серверде орналасып, желіге хостинг арқылы таратылуы қажет.

Веб-бетке кірген бойда тамақтар түрін камераға көрсетіп, құнын автоматты түрде шығарады. Веб-беттің басты беті 3.17-суретте көрсетілген.



3.17-сурет – Веб-беттің басты беті

Камераға науаны әкелген соң, сол веб бетінің оң жағында осы тамақтың құны автоматтыңдыру арқылы жасалынған моделімеzdің нәтижесі. Тамақ құнын шығару 3.18-суретте көрсетілген.



3.18-сурет – Тамақтар құны

ҚОРЫТЫНДЫ

Нейрондық желілердің дамыуы көптеген адамдарының санасы жыне ынтаның жоғарлауы себеп болды. Кескінді тану сияқты көптеген тапсырмалар қазіргі кезде дамып бара жатыр. Әртүрлі тәсілдерді қолдану және керекті алгоритмдерді жазуды қалай пайдалануды түсінуге тырысу керек.

Осы тарауда біз нейронды желі не екенін, қайда қолданылатындығын, олардың артықшылықтары қандай, кемшіліктері қандай екенін қарастырдық. Енді сіз жасанды интеллект пен машина жасауды қалай жұмыс жасайтынын елестете аласыз.

Бүгінгі күнде нейрондық желілер көп жерлерде жұмыс істеуге үшін арналған. Сонымен қатар, нейрондық желілер біздің өмірімізге сенімді түрде енуде. Құрылған моделдер әртүрлі бағытта қолданылады, атап айтсақ медицинада автоматтандырылған болжамдар, қаржы жерлерінде ұсыныс жасау және басқа да жерлерде үнемді қолдайды. бағыттарда қолданылады. Дегенмен, нейрондық желі кең ауқымды мәселелерді шешуге қабілеті болса да, бұл механизмдерді болжамдары әр қашан оң нәтиже көрсете бермейді. Бірақ көптеген мәселелерді автоматтандыру арқылы жұмыс уақытын әлде қайда жылдамдатады.

Бұл проект жасалуы барысында қолданушыға деген барлық қолайлы түрде негізделген. Адам баласына автоматтандыру саласы өзінің ыңғайлылығы және уақытты үнемдеуімен көзге түседі. Тұтынушының тапсырысын автоматтандыру бұл қазір біраз жерлерде жұмыс істейді. Сол сияқты бұл жоба тұтынушыға деген ыңғайлылық және заманға сай есептеу әрекетімен көзге түседі.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Петер Флах. Машинное обучение. Наука и искусство построения Алгоритмов. – USA, 2015. – 130 с.
2. Фрэнк Кейн. Наука о данных, глубокое обучение и машинное обучение с помощью Python. – Лондон, 2016. – 116 с.
3. Как работает нейронная сеть: алгоритмы, обучение, функции активации и потери // Сайттың электронды нұсқасы <https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/>
4. Нейронные сети // Сайттың электронды нұсқасы <http://www.aiportal.ru/articles/neural-networks/neural-networks.html>
5. YOLO: Real-Time Object Detection // Сайттың электронды нұсқасы <https://pjreddie.com/darknet/yolo/>
6. Dependencies Getting started Darkflow // Сайттың электронды нұсқасы <https://github.com/thtrieu/darkflow>
7. Darknet: Open Source Neural Networks in C // Сайттың электронды нұсқасы <https://github.com/pjreddie/darknet>
8. Keras: The Python Deep Learning library // Сайттың электронды нұсқасы <https://keras.io/>
9. TensorFlow is an end-to-end open source platform for machine learning. // Сайттың электронды <https://www.tensorflow.org/>
10. Самоучитель Python // Сайттың электронды нұсқасы <https://pythonworld.ru/samouchitel-python>

А қосымшасы **(міндетті)**

«Тұтынушының тапсырыс беру жүйесін автоматтандыру» web-қосымшасын құруға арналған техникалық тапсырма

А.1 Кіріспе

Кез келген бір жүйені детектеу қиын жұмыстардың бірі. Нейрон желісі қазіргі кезде заманауи технологиялардың ең маңызды бөлігі болып табылады.

Нейрон желісінің көмегімен тамақтарды танып, оның құнын автоматты түрде есептеу болып табылады. Тұтынушының тапсырыс беру жүйесін автоматтандыру қазіргі кезде өте маңызды және қажетті салалардың бірі болып табылады.

А.1.1 Жобаның мақсаты мен міндеті

Бұл бағдарламалық қамтама отырыс орындарының құнын автоматты түрде шығару үшін бағытталған.

Асханалар немесе кафе сияқты жерлерде тұтынушы өзінің тамақтарын және де басқа өнімдер болсын құнын есептейтін кезінде автоматты есептеу әдісімен есептеп қана қоймай, жұмыскерлердің уақтын үнемдейді.

Жүйе кез келген өнімдерді есептеу үшін келесі шарттарды орындауы қажет:

- өнімдерді алып науа(поднос) салу;
- науаны апарып камераға көрсету;

А.1.2 Қолдану саласы

Кез-келген бір ресторан, кафе және асханалар т.б. басқа жерлерде қолданылады. Оның мүмкіндіктерін келесідей көруге болады:

- тамақтарды таниды;
- әр тамақты класстарға бөледі;
- классқа бөлінген тамақты есептейді;
- құнын автоматты түрде шығарады.

А қосымшасының жалғасы

А.2 Жүйенің жалпы сипаттамасы

А.2.1 Тұтынушының тапсырыс беру жүйесін автоматтандыру жүйесінің жалпы сипаттамасы

Бұл проекттің негізгі жұмыстарының бірі автоматтандыру арқылы клиенттің және жұмыскерлердің уақытын өнімдейді.

А.2.3 Аппараттық интерфейстер

Компьютер бойынша қойылатын жалпы талаптар:

- процессор – intel inside CORE i3 версиясынан жоғары;
- оперативті жады – көлемі 4 Гб-тан жоғары;
- интернет желісіне және wi-fi желісіне қатынау;
- видеокарта – NVIDIA grafics.

3 Жүйелік талаптар

3.1 Жалпы жүйелік талаптар

3.1.1 Есептеуді автоматтандыру жүйесіне қойылатын жалпы талаптар

Тұтынушының тапсырыс беру жүйесін автоматтандыру жобасы келесі типтегі есептеуге және құрылғыларға қызмет көрсетеді:

- асхана орындары, кафе және де басқа өнімдерді есептеу үшін арналған жерлерге жұмыс көрсетеді.

Сіз оператор қолмен есептеп отырмау үшін автоматтандыру әдісін қолдану арқылы жүзеге асыру.

А.3.1.2 Жүйелік талаптарды ұйымдастыру

Жобаны ұйымдастыру, жоспарлау, орындау қабілеті, ресторан және асханаларда тұтынушының тапсырыс беруі бойынша жұмыстар ұйымдастырылады.

Б Қосымшасы
(міндетті)

Бағдарлама мәтіні

```
//app.py
#!/usr/bin/env python
from importlib import import_module
import os
from flask import Flask, render_template, Response, jsonify
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
# import camera driver
from camera import Camera
app = Flask(__name__)
@app.route('/')
def index():
    """Video streaming home page."""
    return render_template('index.html')
def gen(camera):
    """Video streaming generator function."""
    while True:
        try:
            frame = camera.get_frame()
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
        except Exception as e:
            print("error : ", e)
@app.route('/face_video')
def video_feed():
    """Video streaming route. Put this in the src attribute of an img tag."""
    return Response(gen(Camera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
if __name__ == '__main__':
    Camera.set_video_source(0)
    #Camera.set_video_source('./idet_out.mp4')
    gen(Camera())
    #app.run(host='172.16.3.142', port=8839, threaded=True)
    app.run(host='localhost', port=8844, threaded=True)

//
//camera.py
import cv2
```

Б Қосымшасының жалғасы

```
from base_camera import BaseCamera
from darkflow.net.build import TFNet
import numpy as np
import time
import os
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
options = {
    'model': 'carplate/food.cfg',
    'load': 'carplate/yolo-food_4400.weights',
    'threshold': 0.2,
    'gpu': 0,
    'labels': 'carplate/labels.txt'
}
tfnet = TFNet(options)
colors = [tuple(255 * np.random.rand(3)) for _ in range(10)]
lastRecognitionTime = -1
tresholdTime = 5
class Camera(BaseCamera):
    video_source = 0
    @staticmethod
    def set_video_source(source):
        Camera.video_source = source
    @staticmethod
    def frames():
        camera = cv2.VideoCapture(Camera.video_source)
        while True:
            stime = time.time()
            ret, frame = camera.read()
            if ret:
                results = tfnet.return_predict(frame)
                for color, result in zip(colors, results):
                    tl = (result['topleft']['x'], result['topleft']['y'])
                    br = (result['bottomright']['x'], result['bottomright']['y'])
                    label = result['label']
                    confidence = result['confidence']
                    text = '{ }: {:.0f} %'.format(label, confidence * 100)
                    frame = cv2.rectangle(frame, tl, br, color, 5)
                    frame = cv2.putText(frame, text, tl,
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 2)
                with tf.device('/gpu:2'):
                    img, (newX,newY) = pd.plateDetector(frame[300:-100,300:-250,:])
```

Б Қосымшасының жалғасы

```
        cv2.rectangle(frame,(300,300),(frame.shape[1]-250,frame.shape[0]-
100),(255,0,0),2)
        outputImg[:,:(width),:] = cv2.resize(frame, (width, height))
        currentTime = time.time()
        if (currentTime-lastRecognitionTime) < tresholdTime:
            try:
                addLabel(outputLabels,new_img,strftime("%d.%m.%Y
%H:%M:%S", localtime()),res)
                photo_time = datetime.utcnow().strftime("%Y-%m-
%dT%H:%M:%S")
                outputImg = generateOuputFrame(frame, outputLabels, width,
height)
            yield cv2.imencode('.jpg', frame)[1].tobytes()

//
//plateDetector.py
import os
#os.environ["CUDA_VISIBLE_DEVICES"]="1"
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
import cv2
options = {
    'model': 'carplate/food.cfg',
    'load': 'carplate/yolo-food_4400.weights',
    'threshold': 0.3,
    'gpu': 0.95,
    'labels': 'carplate/labels.txt'
}
class Detector():
    def __init__(self):
        from darkflow.net.build import TFNet
        self.tfnet3 = TFNet(options)
        #df = RunDarkflow()
    def plateDetector(self, img): #Plate Detector
        try:
            #img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            X = 0
            Y = 0
            result = self.tfnet3.return_predict(img)
            crop_img = None
            for res in result:
                print("res",res)
```


Б Қосымшасының жалғасы

```
#d = int((res['bottomright']['x']-res['topleft']['x'])*0.1)
d=0
if res['label']=='001':
    d = 5
f_h, f_w = img.shape[:2]
tl = (res['topleft']['x']-d, res['topleft']['y'])
br = (res['bottomright']['x']+d, res['bottomright']['y'])
if(res['topleft']['x'] - d < 0 or res['topleft']['y'] - d < 0 or
res['bottomright']['x'] + d > f_w or res['bottomright']['y'] + d > f_h):
    d = min(res['topleft']['x'], res['topleft']['y'], f_w -
res['bottomright']['x'], f_h - res['bottomright']['y'])
label = res['label']
# add the box and label and display it
if crop_img is None:
    crop_img = img[res['topleft']['y']:res['bottomright']['y'],
res['topleft']['x']:res['bottomright']['x']+d]
#crop_img = cv2.cvtColor(crop_img, cv2.COLOR_BGR2RGB)
X = res['topleft']['x']
Y = res['topleft']['y']
if res['label'] == '001':
    cv2.rectangle(img,tl,br,(0,0,255),3)
else:
    cv2.rectangle(img,tl,br,(255,0,0),3)
#break
except Exception as e:
    print("plateDetector.py line 55: ",e)
return crop_img
```

<i>Формат</i>	<i>Зона</i>	<i>Поз.</i>	<i>Обозначение</i>	<i>Наименование</i>	<i>Кол.</i>	<i>Примечание</i>
						<i>Лист</i>
						49

Дипломный проект