

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Тергеусизова Н.Н.

Өмір сүру салаларына арналған машина оқыту әдістерін қолдану

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

КҚФЖС кафедра меңгерушісі,

канд.тех.наук, ассистент

профессор

 Н.А.Сейлова

« 15 » мамыр 2019 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы «Өмір сүру салаларына арналған машина оқыту әдістерін қолдану»

5B070300 – «Ақпараттық жүйелер» мамандығы бойынша

Орындаған

Тергеусизова Н.Н.

Пікір беруші:

Front - end Разработчик

Ж.Ш. Тілеубаев

« 08 » мамыр 2019ж.

Ғылыми жетекші :

Техн.ғыл.магистр. лектор.

 А.Н.Дуйсенбаева

« 1 » мамыр 2019ж.



Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 – «Ақпараттық жүйелер» мамандығы

БЕКІТЕМІН

КҚАӨЖС кафедра меңгерушісі,

тех.ғыл. канд, ассистент

профессор

 Н.А.Сейлова

« 15 » мамыр 2019 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Тергеусизова Назым Нурсагатқызы

Тақырыбы: Өмір сүру салаларына арналған машина оқыту әдістерін қолдану

Университет Ректоры: 2018 « 16 » 10 № 1162 -бет бұйрығымен бекітілген

Аяқталған жұмысты тапсыру мерзімі: 2019 « 16 » мамыр

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

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) қойылған мәселенің қазіргі жағдайын пайымдау

ә) ақпараттық қамтаманы құру

б) программалық қамтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар



Сызба материалдар: 15 слайдпен көрсетілген

Ұсынылатын негізгі әдебиет: 20 амау

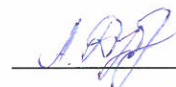
Дипломдық жобаны дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мәселенің қазіргі жағдайына шолу және оны талдау	11.01.2019 - 08.03.2019.	
Ақпараттық қамтаманы құру	06.02.2019-10.03.2019.	
Программалық қамтаманы құру	12.03.2019-28.04.2019.	

Дипломдық жұмысының бөлімдерінің кеңесшілері мен
норма бақылаушыларының аяқталған жобаға қойған
қолтаңбалары


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

Ғылыми жетекші



Дуйсенбаева А.Н.

Тапсырманы орындауға алған білім алушы



Тергеусизова Н.Н.

Күні

« 8 » маусым 2019

Ғылыми жетекшінің пікірі

Дипломдық жұмысқа
Тергеусизова Назым
5B070300- Ақпараттық жүйелер

Тақырыбы: Өмір сүру салаларына арналған машина оқыту әдістерін қолдану

Дипломдық жұмыстың мақсаты - рентген суреттерін анықтау үшін машиналық оқытудың бірнеше әдістерді енгізу және салыстыра отырып оларды жеңілдету.

Тергеусизова Назым – машиналық оқытуды өмір сүру салаларына арналған әдістерін қолдана отырып программаны әзірлеуді, алдына қойылған жұмыстарды шешуге ынтасын көрсетті. Сонымен бірге желі қауіпсіздігін қамтамасыз ету жұмысын жүзеге асыру үшін құралдар таңдалды.

Студент дипломдық жұмысты толығымен өз бетінше орындады. Заманауи технологияларға шолу жасап, жергілікті компьютерлердің негізгі мәліметтерін қарастырды, бағдарламалық қамтаманы және құрылғыларды таңдай білді. Өмір сүру салаларына арналған машина оқыту әдістерін қолдану арқылы логикалық сұлбасын құрып, қауіпсіздік шараларын ұйымдастырды.

Дипломдық жұмысты орындау барысында Тергеусизова Назым алдына қойылған мақсатты орындап, Өмір сүру салаларына арналған машина оқыту әдістерін қолдану арқылы программалық қамтамасын құрып, қауіпсіздік шараларын ұйымдастырды. Жобалап, бағдарламалық және техникалық жабдықтарды таңдап, қауіпсіздігін ұйымдастырды..

Студент Тергеусизова Назым дипломдық жұмысты орындау барысында оқу кезінде алған теориялық және тәжірибиелік білімін, техникалық әдебиеттермен жұмыс жасап, қолдана білетіндігін көрсетті.

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

Ғылыми жетекші

Техн.ғыл.магистр.лектор

(қызметі, ғыл. дәрежесі, атағы)

 Дуйсенбаева А.Н.

(қолы)

« 8 » 05 2019ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Дипломдық жұмысқа СЫН-ПІКІР

Білім алушы :Тергеусизова Назым Нурсагатқызы

Дипломдық жұмыстың тақырыбы: Өмір сүру салаларына арналған машина оқыту әдістерін қолдану

Бұл дипломдық жұмыс Кеуде рентгендік суреттерінің ақпараттық қамтамасын құру , машиналық оқытудың медицина ортасында қолданылуының жоғарғы дәлдігі қарастырылған.

Дипломдық жұмыста түсіндірме хат кіріспеден, қорытынды, әдебиеттер тізімінен тұрады. Дипломдық жұмыс барлығы 37 беттен тұрады.

Кіріспеде мәселенің маңыздылығы, дипломдық жұмыстың мақсаты және қолданылған құралдар қысқаша келтірілген. Бірінші бөлімде машиналық оқытудың өзектілігі , оның даму тарихы және машиналық оқытудың әдістері туралы толық мәлімет жазылған. Екінші бөлімде өмір сүру салаларындағы машиналық оқытудың қолданылуы жайында қарастырылған. Үшінші бөлімде программаның сипатамасы ,жалпы мәліметтер,функционалдық құрылымы,логикалық құрылымның баяндалуы, қолданылған техникалық құрал-жабдықтар, кіріс және шығыс мәліметтер жайында толық қарастырылып жазылған.Төртінші бөлімде дипломдық жұмыстың практикалық бөлімі орындалған. Дипломдық жоба жоғарғы ғылыми техникалық деңгейде орындалған.

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

Тергеусизова Назым дипломдық жұмысты орындау барысында өз беттілік көрсетіп, әдебиеттермен жұмыс істеген. Дипломдық жұмыс Ақпараттық жүйелер мамандығының бітіру жұмыстарына қойылатын талаптарға сәйкес.

Қорытындылай келе, Тергеусизова Назымның дипломдық жұмысына «Жақсы» деген баға беруге және оның орындаушысына 5B070300 – «Ақпараттық жүйелер» мамандығы бойынша бакалавр біліктілігін беруге болады деп есептеймін.

Сын-пікір беруші:

Front-end разработчик

Ж.Ш.



Тілеубаев Ж.Ш

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

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

Автор: Тергеусизова Назым

Название: применение методов машинного обучения в области жизнедеятельности

Координатор: Асемгуль Дуйсенбаева

Коэффициент подобия 1:4,9

Коэффициент подобия 2:4,3

Тревога:11

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

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

Обоснование:

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

Дата 15.06.192

Подпись заведующего кафедрой /

начальника структурного подразделения


КБС/ХУ

Окончательное решение в отношении допуска к защите, включая обоснование:

Вернуть к переделке

Дата 15.05.19.2

Подпись заведующего кафедрой / 

начальника структурного подразделения



Протокол анализа Отчета подобия Научным руководителем

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

Автор: Тергеусизова Назым

Название: применение методов машинного обучения в области жизнедеятельности

Координатор: Асемгуль Дуйсенбаева

Коэффициент подобия 1: 4,9

Коэффициент подобия 2: 4,3

Тревога: 11

После анализа Отчета подобия констатирую следующее:

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

Обоснование:

.....

.....

.....

.....

.....

.....

20.08.2022.

Дата

А.А.А.

Дубинина А.Н.

Подпись Научного руководителя

! К вашему сведению, некоторые слова в этом документе содержат буквы из других алфавитов. Возможно - это попытка скрыть позаимствованный текст. Документ был проверен путем замещения этих букв латинским эквивалентом. Пожалуйста, уделите особое внимание этим частям отчета. Они выделены соответственно.
Количество выделенных слов 11

№	Название, имя автора или адрес гиперссылки (Название базы данных)	Автор	Количество одинаковых слов
1	URL_ https://www.zakon.kz/4796475-1178aza1179standa-so11871171y-10-zhylda.html		273
2	URL_ https://www.inform.kz/kz/akmolada-tuberkulezben-kureske-2-mird-tengeden-astam-karzh-y-zhumsalmak_a3515321		15
3	URL_ https://www.inform.kz/kz/akmolada-tuberkulezben-kureske-2-mird-tengeden-astam-karzh-y-zhumsalmak_a3515321		15
4	URL_ https://www.topreferat.com/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%D0%B4%D1%8B%D2%9B-%D0%B6%D2%B1%D0%BC%D1%8B%D1%81%365-%D0%9C%D2%B1%D0%BD%D0%B0%D0%B9-%D1%82%D0%BA		5

Не обнаружено каких-либо заимствований

Не обнаружено каких-либо заимствований

Не обнаружено каких-либо заимствований

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

Copyright © Plagiat.pl 2002-2019

АНДАТПА

Бұл дипломдық жұмыста өмір сүру саласында машиналық оқыту әдістерін қолдану арқылы программалық қамтаманы жүзеге асыру қарастырылған.

Медициналық оқыту тәжірибеде ең көп таралған құралы ретінде кеуде радиографиясының ауру диагнозында маңызды клиникалық мәні бар. Осылайша, кеуде радиографиясына негізделген кеуде ауруын автоматты түрде анықтау науқастарды медициналық зерттеу жұмыстарының алдыңғы тақырыптарының бірі болды. Клиникалық қосымшаларға сүйене отырып, зерттеу компьютерлік диагностика (CAD) жүйелері бойынша кешенді зерттеу жүргізілді және әсіресе кеуде радиографиясында қолданылатын жасанды интеллект технологиясына бағытталған. Жұмыста бірнеше жалпы кеуде рентгендік деректер жиынтығы бар, сонымен қатар контрастты жақсарту және сегментация сияқты жалпы кескінді алдын ала өңдеу процедуралары, сондай-ақ, кеуде радиографиясына қолданылатын сүйек сөндіру әдістері қысқаша баяндалған. Содан кейін алгоритмнің негізгі қағидаларына, зерттеулерде қолданылатын деректерге, бағалау шараларына және басқа да ауруларға назар аударатын жүйесі нақты ауруларды (өкпе түтіндері, туберкулез және интерстициалды өкпе аурулары) және көптеген ауруларды анықтауда сипатталды. Нәтижесінде, жасанды интеллект негізінде кеуде радиографиясында АЖЖ жүйесін жинақтап, бар мәселелер мен үрдістерді талқыланды.

Машиналық оқытудың алгоритмдері арасында терең оқытудың алгоритмдері - кешенді көп қабатты нейрондық желілер болып есептеледі.

Осы жұмыстың бөлігі ретінде адамның рентгендік суреттеріне терең оқытудың машиналық оқытудың алгоритмдерін қолдану мүмкіндігі, атап айтқанда, адамның өкпесін автоматты түрде тану мүмкіндігі, кейіннен оны анықтау үшін қолдануға болады.

Бұл дипломдық жұмыстың мақсаты - рентген суреттерін анықтау үшін машиналық оқытудың бірнеше әдістерді енгізу және салыстыра отырып оларды жеңілдету.

Дипломдық жобаның түсініктеме жазба жалпы көлемі 39 бет: 3 тілде аңдатпа (мемлекеттік, орыс және ағылшын тілдерінде), 20 сурет, 3 кесте, 9 пайдаланылған әдебиеттер тізімі, 1 қосымшадан тұрады.

АННОТАЦИЯ

Как наиболее распространенный инструмент обследования в медицинской практике, рентгенография грудной клетки имеет важное клиническое значение в диагностике заболевания. Таким образом, автоматическое обнаружение заболеваний грудной клетки на основе рентгенографии грудной клетки стало одной из горячих тем в исследованиях медицинской визуализации. Основываясь на клинических применениях, в исследовании проводится комплексное исследование систем автоматизированного обнаружения (САПР), и особое внимание уделяется технологии искусственного интеллекта, применяемой в рентгенографии грудной клетки. В документе представлены несколько распространенных наборов данных рентгеновского исследования грудной клетки и кратко представлены общие процедуры предварительной обработки изображений, такие как повышение контрастности и сегментация, а также методы подавления кости, которые применяются для рентгенографии грудной клетки. Затем описывается система САД для выявления специфических заболеваний (легочных узелков, туберкулеза и интерстициальных заболеваний легких) и множественных заболеваний с акцентом на основные принципы алгоритма, данные, использованные в исследовании, меры оценки и Результаты. Наконец, в дипломной работе обобщается система САД в рентгенографии грудной клетки на основе искусственного интеллекта и обсуждаются существующие проблемы и тенденции.

Среди алгоритмов машинного обучения выделяют алгоритмы глубокого машинного обучения – сложных многослойных нейронных сетей. В рамках этой работы будет исследована возможность применения алгоритмов глубокого машинного обучения к флюорографическим снимкам человека, а именно для автоматического распознавания легких человека, что может быть использовано для последующего обнаружения туберкулеза.

Целью данной работы является упрощение и внедрение нескольких методов машинного обучения для идентификации рентгеновских изображений.

Общий объем описания дипломного проекта составляет 39 страниц: аннотация на трех языках (на государственном, русском и английском языках) , 20 рисунков, 3 таблиц, 9 использованных источников и 1 приложение.

ANNOTATION

As the most common examination tool in medical practice, chest radiography has important clinical value in the diagnosis of disease. Thus, the automatic detection of chest disease based on chest radiography has become one of the hot topics in medical imaging research. Based on the clinical applications, the study conducts a comprehensive survey on computer-aided detection (CAD) systems, and especially focuses on the artificial intelligence technology applied in chest radiography. The paper presents several common chest X-ray datasets and briefly introduces general image preprocessing procedures, such as contrast enhancement and segmentation, and bone suppression techniques that are applied to chest radiography. Then, the CAD system in the detection of specific disease (pulmonary nodules, tuberculosis, and interstitial lung diseases) and multiple diseases is described, focusing on the basic principles of the algorithm, the data used in the study, the evaluation measures, and the results. Finally, the paper summarizes the CAD system in chest radiography based on artificial intelligence and discusses the existing problems and trends.

Among machine learning algorithms, there are algorithms of deep machine learning - complex multilayered neural networks.

As part of this work, the possibility of applying deep machine learning algorithms to human x-ray images, namely, for automatic recognition of human lung, which can be used for subsequent detection tuberculosis.

The purpose of this thesis is to simplify and introduce several methods of machine learning to identify X-ray images.

The total volume of the description of the diploma project is 39 pages: an abstract in three languages (Kazakh, Russian and English), 20 images, 3 tables, 9 list of references and 1 appendix.

МАЗМҰНЫ

КІРІСПЕ	9
1 МАШИНАЛЫҚ ОҚЫТУДЫ ТАЛДАУ ЖӘНЕ АНАЛИЗДЕУ	10
1.1 Машиналық оқытудың өзектілігі	10
1.2 Машиналық оқытудың даму тарихы	11
1.3 Машиналық оқыту әдістерін қолдану	12
1.4 Жасанды нейрондық желінің негізгі ұғымдары	16
1.5 Терең оқыту (Deep learning) жұмыс бағыттары	16
2 ӨМІР СҰРУ САЛАЛАРЫНДАҒЫ МАШИНАЛЫҚ ОҚЫТУ	18
2.1 Өмір сүру салаларындағы машиналық оқыту бағыттары	18
2.2 Медицина саласында машиналық оқыту әдістерін қолдану	19
2.3 Кеуде рентгендік суреттердің ақпараттық қамтамасын құру	20
2.4 Суреттің алдын-ала өңдеу әдістері	21
2.5 Ауруды анықтаудың өзіндік ерекшелігі	23
3 ПРОГРАММАНЫҢ СИПАТТАМАСЫ	25
3.1 Жалпы мәліметтер	25
3.2 Функционалдық құрылымы	25
3.3 Логикалық құрылымның баяндалуы	25
3.4 Қолданылатын техникалық құрал-жабдықтар	26
3.5 Шақыру және жүктеу	26
3.6 Кіріс мәліметтер	26
3.7 Шығыс мәліметтер	26
4 ПРАКТИКАЛЫҚ БӨЛІМ	27
4.1 PYTHON бағдарламалау тілінің сипаттамасы	27
4.2 Anaconda Python бағдарламалау тілінің сипаттамасы	28
4.3 Программалық қамтаманың құрылымы	30
ҚОРЫТЫНДЫ	38
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	39
Қосымша А	40

КІРІСПЕ

Денсаулық сақтаудың статистикалық деректері бойынша кейінгі 10 жылда Қазақстандағы туберкулез ауруы көрсеткіші 54,9%-ға, ал өлім көрсеткіші 77,4%-ға төмендеген.

Сонымен қатар тексеру барысында статистика ақпаратын бұрмалау, профилактикалық іс-шаралардың тиісті деңгейде ұйымдастырылмауы, сапасыз дәрігерлік қызмет көрсету, науқастарды еркінен тыс емдеу шараларын қамтамасыз етпеу сияқты бірқатар олқылық анықталды. Мысалға, Алматы облысы Талғар ауданының Орталық емханасында тексеру күні 78 емделушінің флюорографиясы оң нәтиже көрсеткенімен, толықтай тексерілмегені анықталды. Ал Қостанай облысы Әулікөл ауданының орталық емханасына туберкулезге тексеруден өтуі қажет белгілерімен «О» бес айы бойы бірнеше рет қаралуға келген (2015 жылғы 28 мамыр, 8 тамыз және 14 қазан).

Соған қарамастан, дәрігерлер ауруға сақтықпен қарамай, нәтижесінде 22 қазанда ол туберкулезге қарсы ауруханаға жатқызылып, 4 күннен кейін туберкулез ауруының асқынғандығынан жан тапсырған. Осы оқиға бойынша сотқа дейінгі тергеу жүргізіліп жатыр. Жалпы тексеру нәтижесінде туберкулез ауруына шалдығу күдігі бар адамдардың тексеруі уақытылы жүргізілмеген, толық тексерілмеген сынды 1028 дерек анықталды. Көрсетілген факторлар сау адамдарға тікелей қауіп төндіреді. Сонымен қатар туберкулездің таралмауы сол ауруға шалдыққан адамдардың жауапкершілігіне де байланысты. Тексеру барысында науқастардың емделуден бас тарту фактілері анықталған, бұл жағдай аурудың ауа арқылы таралу қаупі бар қиын емделу түріне айналуына септігін тигізеді. Таңдаулы тексерулер нәтижесінде емделуден бас тартып, оңашалауды қажет ететін, елді мекендерде жүріп, басқа адамдарға ауруын жұқтыру қаупін төндіріп жүрген 327 науқас анықталды. Қолданылған шаралармен 259 науқас емделуге қайтарылды, қалғандары іздестірілуде.

Денсаулық сақтау және әлеуметтік даму министрлігі заңбұзушылықты және оның себептерін жою туралы енгізілген ұсынысты қарау нәтижесінде профилактикалық жұмыстағы немқұрайлықты болдырмай, халықтың денсаулығын қамтамасыз етуге бағытталған шаралар қабылдануда. Сонымен қатар науқас адамдардың емделуден бас тартуы бойынша әкімшілік жауапкершілікке тартылуы және сот арқылы еркінен тыс шаралар қолданылуы мүмкін екенін еске сала отырып, барлық азаматтарды өзінің және өзге адамдардың денсаулығына жауапкершілікпен қарауға шақырады.

Қазақстанда туберкулезді анықтау, алдын алу және емдеу тегін медициналық көмектің кепілдік берілген көлеміне кіреді және тегін. 2019 жылы мемлекеттік бюджеттен туберкулезбен ауратын науқастарға диагностика, емдеу және әлеуметтік көмек көрсету үшін 47,7 млрд. теңге бөлінді.

1 МАШИНАЛЫҚ ОҚЫТУДЫ ТАЛДАУ ЖӘНЕ АНАЛИЗДЕУ

1.1 Машиналық оқытудың және оның медицина саласындағы өзектілігі

Қазіргі уақытта биология, медицина, экология, экономика және басқа да салаларда көптеген ғалымдар үлкен көлемді мәліметтермен жұмыс істейді. Олар осы мәліметтерді қандай да бір түрде топтастыруға, сыныптардың белгілі бір санына, яғни классификация мәселесін шешуге тырысады. Тәжірибелік деректер әдетте нашар құрылымдалған және дәл емес болып келеді. Осындай деректерге арналған дәстүрлі математикалық жіктеу әдістері нашар жұмыс жасайды.

Соңғы жылдары осы мәселені шешу үшін машиналық оқыту әдістерінің саны артып келеді. Машиналық оқыту - аналитикалық модельде автоматтандырылған деректерді талдау әдісі болып табылады. Бұл жасанды интеллекттің жүйесі, жүйеден үйренуге болатын және адамның ең аз араласуын анықтайтын идеяға негізделген.

Machine Learning маңызды құралы Машиналық оқытудың үлкен көлемін енгізу арқылы оқуға және шешім қабылдауға үйретілген машиналарды сипаттайды. Ол осы ақпаратты талдауға негізделген есептік ұсыныстарды және/немесе деген болжамдарды жасайды және адамның ақыл-кеңесін талап ететін есептерді орындайды. Бұған сөйлеуді тану, аударма, көрнекі қабылдау және т.б. кіреді. Машиналық оқыту саласы да терең білім беру саласын қамтиды. Машиналық оқыту мен жасанды интеллект арасындағы негізгі айырмашылық «оқу» термині болып табылады.

Машиналық оқытудың алгоритмдерін күрделі пайдалану арқылы үйретіледі және ақылға қонымды түсініктер береді. Іскерлік құндылықты қамтамасыз ету үшін машина деректерден үлгілерді үйренеді, одан кейін жаңа және өзгеретілген деректерге автономды түрде қосыла алады. Бұл қосымша ресурстарды немесе адамның өзара іс-қимылын талап етпестен қосымша түсініктерді, тіпті дәлірек түсіну үшін қосымша үлгілерді тиімді түрде жасауға мүмкіндік беретін динамикалық кері байланыс циклін жасайды.

Machine Learning соңғы онжылдықта елеулі түрде дамыды және көптеген интеллектуалдық қосымшаларда табысты пайдаланылуда. Ең қызықты сұрақтардың бірі - Machine Learning медициналық диагностика саласында табысты қолдану мүмкін бе. Сонымен қатар, қандай деректер қажет екендігі туралы сұрақ туындайды. Мамандандырылған медициналық өрістерде машиналық оқыту әдістерін табысты қолданудың бірнеше мысалдары бар. Жақында терінің суреттеріне негізделген тері катерлі ісігін анықтауға қабілетті модель ұсынылды, ол дерматологтың салыстырмалы деңгейіне жетеді. Дегенмен, гематология сияқты медициналық диагностикада кең және күрделі өрістерді шешетін машиналық оқытудың табысты қолданысы жоқ.

Медициналық диагноз - бұл аурудың адамның белгілері мен осы белгілерін жақсы түсіндіруге арналған анықтау процесі. Дәрігерге

дифференциалды диагнозды анықтауға және жедел емдеуге жоспарын жасау үшін медициналық білімдер, дағдылар мен тәжірибе маңызды рөл атқарады. Диагностикалық процедурада қол жетімді ақпарат пациенттің медициналық тарихынан, физикалық тексерісінен және әр түрлі диагностикалық сынақтардан, соның ішінде клиникалық зертханалық зерттеулерден алынуы мүмкін және қосымша деректер жинаумен толықтырылады. Зертханалық сынақтар ауруларды растауға, жоюға, жіктеуге немесе бақылауға және емдеуге бағытталады. Дегенмен, зертханалық тест нәтижелерінің шынайы қуаты жиі бағаланбайды, өйткені клиникалық зертханалар сынақ нәтижелерін жеке сандық немесе категориялық мәндер деп есептейді, ал дәрігерлер негізінен берілген ауқымнан тыс құндылықтарға шоғырланады.[1]

1.2 Машиналық оқытудың даму тарихы

Бүгінгі күні машиналық оқыту алгоритмдері компьютерлермен адамдармен қарым-қатынас жасауға, автомобильдерді дербес басқаруға, спорттық матч туралы есептерді жазуға, жариялауға және террорист күдіктілерді табуға мүмкіндік береді. Сондықтан қазіргі таңда машиналық оқыту көптеген өндірістерге және олардағы жұмысқа қатты әсер ететініне сенімдімін, сондықтан әрбір менеджер кем дегенде қандай да болсын машиналық оқытудың бір әдісін және оны қалай дамитындығын білуі керек.

1950 - Алан Тьюринг компьютердің нақты ақыл-ойының бар-жоғын анықтау үшін «Тьюринг сынағын» жасайды. Сынақтан өту үшін, компьютер адамға адам болып табылады деп сену үшін адамды алдауға қабілетті болуы керек.

1952 - Артур Сэмюэл бірінші компьютерлік оқу бағдарламасын жазды. Бағдарлама дойбы ойыны үшін арналған болды, ал IBM компьютерінің ойында ойнаған сайын, оны жақсартты, оқып жатқан стратегияларға ие болды және оларды сол бағдарламаға енгізді.

1957 - Фрэнк Розенблатт адам миының ойлау үдерістерін модельдейтін компьютерлерге (перцептрон) арналған алғашқы нейрондық желіні әзірледі.

1967 ж. - «Жақын көршіміз» алгоритмі жазылды, бұл компьютерлерге өте жақсы тануды бастауға мүмкіндік береді. Бұл кездейсоқ қалада басталатын саудагерлерге арналған маршрутты картаға түсіру үшін қолдануға болады, бірақ олар қысқа мерзім ішінде барлық қалаларға баруды қамтамасыз етеді.

1981 - Джеральд Дежонг түсіндірме негізделген оқыту (EBL) тұжырымдамасын таныстырады, онда компьютер компьютерде оқыту туралы деректерді талдайды және маңызды емес деректерді алып тастау арқылы жалпы ереже жасайды.

1990 жылдар - Машина жасаудағы жұмыс білім негізінде басқарылатын тәсілнен деректерге негізделген тәсілге ауысады. Ғалымдар компьютерлерге үлкен көлемдегі деректерді талдауға және қорытындыдан - немесе нәтижелерден - «үйренуге» арналған бағдарламалар жасауды бастайды.

1997 - IBM Deep Blue шахмат бойынша әлем чемпионы атанды.

2006 - Джеффри Хинтон компьютерлерге «көруге» және суреттер мен бейнелердегі нысандарды және мәтінді ажыратуға мүмкіндік беретін жаңа алгоритмдерді түсіндіру үшін «терең білім» терминін енгізді.

2010 - Microsoft MSFT + 0% Kinect 20 адамның мүмкіндіктерін секундына 30 рет жылдамдықпен қадағалай алады, бұл адамдарға қозғалыстар мен қимылдар арқылы компьютермен өзара әрекеттесуге мүмкіндік береді.

2011 - IBM Watson өздерінің адам бәсекелестерін Jeopardy-да жеңіп шықты.

2011 - Google GOOGL + 0% миы дамыды және оның терең нейрондық желісі мысықтарды мысал ретінде анықтауға және санауға үйренді.

2012 - Google X Lab зертханалық машықтану алгоритмін дамытады, ол мысықтарды қамтитын бейнелерді анықтау үшін YouTube бейнелерін дербес карауға қабілетті болды.

2014 ж. - Facebook FB + 0% адам суретін түсіруге қабілетті адамдарға бір деңгейде фотосуреттерді тануға немесе тексеруге қабілетті DeepFace, бағдарламалық жасақтама алгоритмін жасады.

2015 - Amazon өзінің жеке компьютерлік оқыту платформасын іске қосады.

2015 - Microsoft бірнеше компьютерлерде компьютерлік оқыту проблемаларын тиімді бөлу мүмкіндігін беретін Distributed Machine Learning Toolkit құрады.

2015 - Stephen Hawking, Elon Musk және Стив Возняк (басқалар арасында) мақұлдаған 3000-нан астам АИ және Робототехника зерттеушілері адамның араласуынсыз мақсатты таңдайтын және тартатын дербес қарудың қауіптілігі туралы ашық хатпен қол қояды.

2016 - Google жасанды интеллект алгоритмі әлемдегі ең күрделі ойын-сауық ойыны болып саналатын және шахматқа қарағанда бірнеше есе қиын болып есептелетін қытайлық Go ойындарында кәсіби ойыншы болып табылады. Google DeepMind компаниясы әзірлеген АльфаГо алгоритмі «Go» жарысында бесеуден бес ойын жеңіп алды.[2]

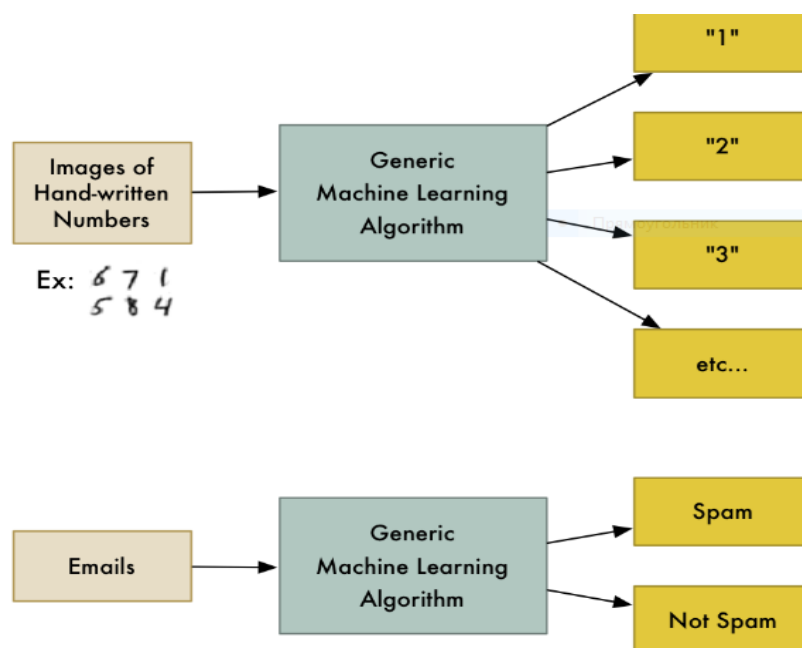
1.3 Машиналық оқыту әдістерін қолдану

Машиналық оқыту – ол әр мәселеге арнайы код жазуды қажет етпейтін, деректер жиынтығына қарап сізге қызықты ақпаратты қорытып бере алатын жалпы алгоритм идеясы. Код жазудың орнына сіз сол жалпы алгоритмге мәліметтерді бересіз, ал ол болса сол деректерге сүйене отырып өзінің жұмыс істеу логикасын құрады.

Осы алгоритмнің бір түрі ретінде классификациялау алгоритмін келтіруге болады. Ол мәліметтерді әр түрлі топқа бөле алады.

Бірінші суретте классификациялау алгоритмі кодты өзгертусіз-ақ, қолмен жазылған санды тануға, әрі email хаттарын спам/спам емес етіп бөлуге

қолданыла алады. Екеуі де бір алгоритм, бірақ оқыту деректері әртүрлі болғандықтан, әртүрлі классификациялау логикасымен жұмыс жасайды.



Сурет 1.1 - классификациялау алгоритмі

Бұл машиналық оқыту алгоритмі қара қорап тәріздес. Әр түрлі классификациялау мәселесін шешуге қайта қолданыла алады.

Машиналық оқыту осындай көптеген жалпы алгоритмдерді біріктіретін термин.

Машиналық оқыту алгоритмдерін екі негізгі топқа бөлуге болады. Олар: бақыланатын оқыту және бақылаусыз оқыту. Екеуінің айырмашылығы қарапайым болғанымен, өте маңызды.

Бақыланатын оқыту.

Сіз жылжымайтын мүлік сататын агент деп есептейік. Сіздің бизнесіңіз кеңейгендіктен сіз өзіңізге көмектеседі деген оймен қосымша көмекшілерді жалдадыңыз. Бірақ мынадай мәселе туындады. Сіздің тәжірибеңіз мол болғандықтан, үйдің түріне қарап-ақ оның нарықтағы бағасын шамамен айта аласыз. Ал сіздің көмекшілеріңіз тәжірибенің жоқтығынан олай істей алмайды.

Сондықтан сіз оларға көмектесетін кішігірім бағдарлама жазуды ұйғардыңыз. Бірінші кестеде үйдің көлемі, мекен-жайы, көрші үйлердің бағасы сияқты т.б. көптеген мәліметтерді ескере отырып, керек үйдің бағасын есептеп беру мақсатына ие. Ол үшін сіз соңғы үш айда сатылған үйлердің мынадай сипаттамаларын: бөлмелер саны, аумағы, қабат саны сияқты көптеген мәліметтерді тіркей бастадыңыз. Ең бастысы бұл үйлер қаншаға сатылғандығын жазып отырдыңыз. Мәліметтер мына түрде жазылған:

Bedrooms	Sq. feet	Neighborhood	Sale price
3	2000	Normaltown	\$250,000
2	800	Hipsterton	\$300,000
2	850	Normaltown	\$150,000
1	550	Normaltown	\$78,000
4	2000	Skid Row	\$150,000

Кесте 1.1 – үйдің параметрлері

Бұл біздің оқыту деректеріміз.

Екінші кестеде оқыту деректерін қолданып, біз мысалға мына үйдің бағасын есептеп бере алатын бағдарламаны жасаймыз.

Bedrooms	Sq. feet	Neighborhood	Sale price
3	2000	Hipsterton	???

Кесте 1.2 - Үйдің бағасын есептеу

Біз оқыту деректерін қолдану арқылы керек үйдің бағасын жоспарлағымыз келеді.

Осының бәрі Бақыланатын оқыту деп аталады. Сіз әр үй қаншаға сатылғандығы жайлы білесіз, басқа сөзбен айтқанда сіз мәселенің шешімін білгендіктен, мәселені шешуге қандай логика керек екендігін түсіну үшін қолдана аласыз.

Өзіңіздің бағдарламаңызды құру үшін сіз дайын оқыту деректерін машиналық оқыту алгоритіміне бересіз. Алгоритм сол мәліметтерді қолданып, сандар сәйкес келуі үшін қандай математика қолдану керектігін түсінуге тырысады.

Екінші суретте арифметикалық символдары өшірілген теңдеуге ұқсайды.

Math Quiz #1 - Teacher's Answer Key			
1) 2	4	5	= 3
2) 5	2	8	= 2
3) 2	2	1	= 3
4) 4	2	2	= 6
5) 6	2	2	= 10
6) 3	1	1	= 2
7) 5	3	4	= 11
8) 1	8	1	= 7

Сурет 1.2 - Арифметикалық символдары өшірілген теңдеу

Сіз сол жақтағы сандармен бірнәрсе істеу арқылы он жақтағы санға шығуға болатынын білесіз.

Бақыланатын оқытуда сіз компьютерге сол теңдеуді табуға мүмкіндік бересіз. Дәл осы мәселені шешуге қажет математиканы білгеннен кейін, сіз сол алгоритмді осыған ұқсас мәселелердің шешімін табуға қолдана аласыз.

Бақылаусыз оқыту

Жылжымайтын мүлік агентінің мысалына қайта оралайық. Егер біз үйлер қай бағамен сатылғандығын білмесек ше? Үшінші таблицада үйдің көлемін, орналасуын және т.б. деректерді ғана белгілі, осы мәліметтер арқылы сіз керемет заттар жасай аласыз. Осы бақыланбайтын оқыту деп аталады.

Bedrooms	Sq. feet	Neighborhood
3	2000	Normaltown
2	800	Hipsterton
2	850	Normaltown
1	550	Normaltown
4	2000	Skid Row

Кесте 1.3 – үйдің параметрлері

Сіз белгісіз санды (мысалға үйдің бағасын) болжауға тырыспасаңыз да, сіз бәрібір машиналық оқыту арқылы қызықты заттар жасай аласыз.

Бұл біреу сізге кестеге толтырылған сандарды беріп: “Мына сандар не білдіретінін білмеймін, бірақ сен мүмкін солардың арасындағы бір байланысты байқап қаларсың – іске сәт!” дегенмен пара-пар.

Сонда бұл деректермен не істеуге болады? Біріншіден, сіз осы деректерге қарап, әртүрлі нарық сегменттерін анықтай алатын алгоритм жасау аласыз. Сіз жергілікті колледждің қасындағы үй сатып алушылардың көлемі шағын, бірақ бөлмелері көп үйлерді, ал қаланың сыртындағылары 3 бөлмелі, бірақ көлемі үлкен үйді ұнататынын байқауыңыз мүмкін. Тұтынушылардың осындай айырмашылықтарын білу, маркетингтік стратегияның дұрыс таңдалуына көмектесетіні белгілі.

Екіншіден, сіз басқа үйлерден ерекшеленетін үйлерді табуыңыз мүмкін. Мысалы, ол аумағы өте үлкен үй болуы мүмкін. Осы үйден түсетін комиссия көп болғандықтан, сіз мықты деген агенттеріңізді сол үйді сатуға жіберуіңізге болады.

Біз осы постыда бақыланатын оқытуды көбірек қарастырамыз. Бірақ осыны байқап, бақылаусыз оқытудың пайдасы жоқ екен деп ойлап

қалмаңыздар. Шын мәнінде бақылаусыз оқыту соңғы жылдары үлкен трендке айналып жатыр. Ал оның басты себебі – деректерді дұрыс жауаппен беруді қажет етпеуінде.[4]

1.4 Жасанды нейрондық желінің негізгі ұғымдары

Нейрондық желілер - қарапайым процессорлар (жасанды нейрондар) жүйесі және бір-бірімен байланысы болып табылады. Мұндай процессорлар әдетте өте қарапайым (әсіресе дербес компьютерлерде қолданылатын процессорлармен салыстырғанда). Осындай желінің әрбір процессоры мезгілінде қабылдайтын сигналдармен ғана жұмыс істейді және ол басқа өңдеушілерге мерзімді түрде жіберетін сигналдармен жұмыс істейді. Дегенмен, бақыланатын өзара әрекеттестігі бар өте үлкен желіге қосылғандықтан, бұл процессорлар өте күрделі тапсырмаларды орындауға қабілетті, өйткені жүйеде нейрондық желілер дайындалады.[5]

Нейрондық желілер кең ауқымды деректерді өңдеу және талдау міндеттерін шеше алады - үлгіні тану және жіктеу, болжау, бақылау және т.б.

Ең жиі пайдаланылатын бағдарламалар нейроимитаторлар болып табылады - олар қарапайым компьютерлерде жұмыс істейді, және қазіргі заманғы нейроалгоритмдер үлкен көлемдегі ақпаратты жылдам өңдеуге мүмкіндік береді. Күрделі нақты проблемаларды шешу үшін нейрондық желілерді пайдаланылады. Бағдарламалар нейрондық желілерді құру, үйрету және манипуляциялау, бастапқы деректер, нейрондық желілердің қасиеттері және нейрондық шешімдер, сондай-ақ ең қарапайым әрекеттер тізбектерін орындаудың автоматтандырылған процедуралары, мысалы, оңтайлы желі параметрлерін және оқу алгоритмін анықтауға арналған негізгі әрекеттер жиынтығын қамтиды.

1.5 Терең оқыту (Deep learning) жұмыс бағыттары

Терең оқыту (deep learning) - деректерді тереңірек түсінуді үйренетін машина алгоритмдерінің классы.

Болжаудың мәселесі нысандар болашақ үшін болжам жасау қажет болған сәтте аяқталатын уақыттық сериялардың сегменттері болып табылады. Болжаудың проблемаларын шешу үшін регрессиялық немесе жіктеу әдістерін бейімдеу жиі мүмкін болады, ал екінші жағдайда шешім қабылдау міндеттері туралы көп айтылады.

Бақыланбаған оқыту. Бұл жағдайда жауаптар орнатылмаған және сіз нысандар арасындағы тәуелділіктерді іздегіңіз келеді.

Класстерлеу міндеті - объектілерді буындардың ұқсастығы туралы деректерді пайдаланып класстерлерге топтау. Сапа функционалдығын әртүрлі түрде анықтауға болады, мысалы, орташа қапсырмалардың және іштей қашықтықтардың арақатынасы.

Қауымдастық ережесін табу міндеті (қауымдастықтың ережелерін үйрену). Бастапқы деректер сипаттамалар түрінде берілген. Нысандар сипаттамасында жиі кездесетін белгілердің осындай белгілерін және осындай белгілерді табу қажет.

Сенімділік доменін құру міндеті (кванттық бағалау) - берілген үлгілік фракциясы бар өте тегіс шекарасы бар ең аз көлемді аймақ.

Өлшемділікті төмендету мақсаты - кейбір конверсиялау функцияларының көмегімен, үлгі нысандар туралы маңызды ақпаратты жоғалтпастан бастапқы ерекшеліктерге негізделген жаңа мүмкіндіктердің ең аз санына өту. Сызықтық қайта құрулар классында ең танымал мысал негізгі компоненттік әдіс болып табылады.

Жетіспейтін мәндерді толтыру міндеті матрицасында жоғалған мәндерді ауыстыру болып табылады - олардың болжамды мәндері бойынша белгілер.

Жартылай оқыту (жартылай бақыланатын оқыту) мұғаліммен немесе мұғалімсіз оқытудың аралық орнын алады. Әрбір пайдалану жағдайы – «жауап» жұбы, бірақ жауаптар пайдалану жағдайларының бөліктері ғана белгілі. Қолданбалы міндеттің мысалы - олардың кейбіреуі қазірдің өзінде кейбір рубрикаларға тағайындалған жағдайда, көптеген мәтіндерді автоматты түрде айқындау.

Динамикалық оқыту (онлайн-оқыту) мұғаліммен бірге және мұғалімсіз оқыту сияқты болуы мүмкін. Ерекшелену - жағдайлардың ағынын пайдалану. Әрбір прецедент туралы дереу шешім қабылдау және сол уақытта жаңа прецеденттерді ескере отырып, тәуелділікті модельді аяқтау қажет. Міндеттерді болжау кезінде, уақыт факторы маңызды рөл атқарады.

2 ӨМІР СҮРУ САЛАЛАРЫНДАҒЫ МАШИНАЛЫҚ ОҚЫТУ

2.1 Өмір сүру салаларындағы машиналық оқыту бағыттары

Image Recognition - ең маңызды машина оқыту бағдарламаларының бірі. Негізінде, бұл сандық кескіндегі мүмкіндікті немесе объектіні анықтау мен анықтамауға арналған тәсіл. Сонымен қатар, бұл әдіс бетті табу, бетті тану, оптикалық таңбаларды тану және тағы басқалар сияқты әрі қарай талдау үшін пайдаланылуы мүмкін. Бірнеше әдіс бар болса да, кескінді тану үшін машинада оқыту әдісін қолдану қажет. Кескінді тану үшін машинада оқыту әдісі кескіннің негізгі ерекшеліктерін алуды қарастырады және осы ерекшеліктерді машиналық оқыту моделіне енгізеді.

Sentiment analysis - нақты уақыт режимінде машиналық оқыту әдістерін қолдану. Ол сондай-ақ пікір білдіруге, көңіл-күйдің жіктелуіне және т.б. жатады. Бұл баяндамашы мен жазушының көзқарасын немесе пікірін анықтауға арналған процесі. Басқаша айтқанда, мәтіннен эмоцияны анықтау процесі. Сезімдерді талдаудың басты мәселесі - «басқа адамдар ойлайды?». Мысалы, біреу «фильм жақсы емес дейді» деп ойлап көріңіз. Мәтіннен нақты ой мен пікірді білу (жақсы немесе жаман) - бұл көңіл-күйді талдау міндеті. Бұл пікірді сараптауға арналған өтінім қосымша қарауға да қолданылады, мысалы, шолу веб-сайтында, шешімдерді қабылдау туралы өтініш.

News classification - машиналық оқытудың басқа көзқарас қосымшасы. Шындығында, ақпараттың көлемі интернетте айтарлықтай өсті. Алайда, әрбір адамның жеке мүддесі немесе таңдауы бар. Осылайша, тиісті ақпараттың бір бөлігін жинау немесе жинау осы веб-сайттың мұхиттағы пайдаланушыларына қиындық тудырады. Мақсатты оқырмандарға қызықты жаңалықтар топтамасын жаңалықтар сайттарының қол жетімділігін сөзсіз арттыратындығын қамтамасыз ету. Сонымен қатар, оқырмандар немесе пайдаланушылар нақты жаңалықты тиімді және тиімді іздеуге болады.

Сөйлеуді тану - ауызша сөздерді мәтінге айналдыру процесі. Бұл қосымша автоматты сөйлеуді тану, компьютерде сөйлеуді тану немесе мәтіндегі сөйлеуді тану деп аталады. Бұл өрістің машинада оқыту тәсілін және үлкен деректерді алға жылжытудан пайдасы бар. Қазіргі уақытта барлық коммерциялық мақсатта сөйлеуді тану жүйесі сөйлеуді тану үшін компьютерлік оқыту әдісін қолданады. Неліктен? Компьютерді оқыту әдісін қолданатын сөйлеуді тану жүйесі дәстүрлі әдісті қолдана отырып, сөйлеуді тану жүйесіне қарағанда әлдеқайда жақсы. Себебі, машинада оқыту әдісінде жүйе валидацияға барар алдында үйренеді. Негізінде, сөйлеуді тануды бағдарламалық қамтамасыз етудің бағдарламалық жасақтамасы екі оқу сатысын қамтиды: 1. Бағдарламалық жасақтаманы сатып алудан бұрын (бағдарламалық жасақтаманы тәуелсіз динамикалық доменде оқыту) 2. Пайдаланушы бағдарламалық жасақтаманы сатып алғаннан кейін (бағдарламаны динамикаға тәуелді доменде оқыту).

Әлеуметтік медиа компьютерді оқыту тәсілін тартымды және керемет ерекшеліктерді, яғни сіз білуі мүмкін, ұсыныс, пайдаланушыларға жауап беру мүмкіндігін жасау үшін пайдаланады. Бұл мүмкіндіктер машина оқыту техникасының нәтижесі ғана. Әлеуметтік есепте сізді тарту үшін компьютерлік оқыту әдісін қалай қолданатындығын ойлайсыз ба? Мысалы, Facebook үнемі сіздің сөйлескен адамдарыңыз, сіздің ұнатқандарыңыз, жұмыс орныңыз, оқу орны сияқты әрекеттеріңізді үнемі байқайды. Машиналық оқыту әрдайым тәжірибеге сүйенеді. Осылайша, Facebook сізге өз қызметіңізге негізделген ұсыныс береді.

Машиналарды оқыту әдістері, құралдар медициналық мәселелер бойынша кеңінен қолданылады. Ауруды, терапияны жоспарлауды, медициналық зерттеулерді, ауруды болжауды анықтауға мысал ретінде. Медициналық ғылымда компьютерлік оқыту негізінде бағдарламалық қамтамасыз етуді пайдалану біздің медициналық ғылымымызда серпіліс әкеледі.

2.2 Медицина саласында машиналық оқыту әдістерін қолдану

Machine learning (ML) дисциплинасы үлкен дерекқордан жаңа білімді қалыптастыруға көмектесетін есептеу әдістері мен оқыту механизмдерін қамтамасыз етеді. ML қосымшалары жіктеу, болжау, тану үлгілері мен білім алудың мәселелерін шешуге арналған тәсілдерді жасау үшін пайдалы және мұнда деректердің мысалдар жиынтығы болып табылады және осының арқасында нәтиже жаңа мысалдарды болжаудың нысанын алады. Бұл тұрғыдан алғанда, ML медицина облыстарында диагностикалық және болжамдық мәселелерді шешуге көмектесетін әдістер мен құралдарды қамтамасыз ете алады, мұнда енгізу деректер субъектілердің сипаттамалары бар деректер жиынтығы болып табылады және шығу - бұл белгілі бір аурудың диагнозы немесе болжауы болады. Дегенмен диагноз және болжау салыстырмалы түрде тікелей ML проблемалары болып табылады, ML қолдану арқылы клиникалық шешім қабылдау медицина қоғамдастығында кеңінен қолданылмайды, өйткені мұндай күрделі міндет тек қана дәлдікке ғана емес, сонымен қатар дәрігерлердің функционалдық медицина саласында ML тәсілдерін қолдану тәсілдеріне байланысты.[6]

Клиникалық шешімдерге қатысты мәселелерде ML қосымшасын ойдағыдай іске асыру үшін кейбір нақты талаптарды қарастыру қажет. Мысалы, аурудың дамуын болжау, әдетте, белгілі бір қауіп факторларын эволюциялаумен байланысты; кейбір созылмалы аурулар (мысалы, рак, жүрек-тамыр аурулары және қант диабеті) жағдайында тәуекел факторлары жас немесе жыныс сияқты өзгермейтін сипаттамаларды қамтиды. Аурудың пайда болуын алдын-ала болжау үшін мұндай өзгермейтін қасиеттерді пайдалану аурудың эволюциясын болдырмау үшін пайдалы болмауы мүмкін, өйткені қазіргі уақытта осы биологиялық сипаттамаларды өзгертуге медицинаның күші

жеткіліксіз. Осылайша, ML қосымшалары әдетте өзгертін қасиеттерге назар аударады, бұл болжамды тапсырманы қиын әрі күрделі етеді.

Қарастырылатын тағы бір маңызды аспекті медицина қызметкерлеріне осы проблема туралы пайдалы ақпаратпен қамтамасыз ету үшін түсіндіруге болатын жақындауларды алу қажеттілігі болып табылады. Бұл, әдетте, шешімдерді оңай түсінікті түрде түсіндіруге мүмкіндік беретін символикалық оқыту әдістері (мысалы, шешім ағаштары мен ережелер жүйелері) арқылы қол жеткізіледі. Алайда, түсінікті модельді алу үшін символикалық оқыту алгоритмін қолдану көбінесе болжаудағы дәлдікті құрбан етеді. Медициналық деректер жиынтығын талдауда жалпы алғанда жоғары көрсеткіштерге жиі кедергі келтіретін тағы бір проблема, әдетте, сау адамдардың көпшілігінің немесе теріс сыныбының (қалыпты деректер) және азғана немесе ауыр науқастардың оң класын қамтитын теңдестірілмеген сыныптық үлестірім (маңызды класс), қате классификацияның құны жоғары болады. Соңғы кезде әдеттегі қателіктердің жоғары деңгейі бар, өйткені стандартты ML алгоритмдерінің өнімділігі азшылық үлгілерінің үлгілерін елемей, нәтижелерді қолайлы дәлдікпен және спецификациямен (сау субъектілер дұрыс диагноз қойылған) қолдана отырып, көпшіліктің сыныбына толтыруға бейім, алайда төмен сезімталдық (науқастар дұрыс диагноз қойылған) әлі де бар.

ML әдістерін дамытудан басқа, жақсы көрсеткіштерге қол жеткізуге және медицина қызметкерлеріне болжауға ақпаратпен қамтамасыз етуге, шешімдерді қолдай білуге және сенімді болжамдарға арналған медициналық сынақтардың санын азайтуға мүмкіндік береді. Диагностика немесе болжамды сенімділік шарасы да маңызды, себебі бұл медицина қызметкерлеріне жаңа тәсілдерді тәжірибеге енгізу үшін жеткілікті сенімділік береді. Екінші жағынан, пациент туралы ақпараттың аз мөлшеріне негізделген сенімді болжауды қамтамасыз ететін тәсіл болуы керек, себебі бұл ақпарат жинау жиі қымбат, мүмкін, құпиялылық мәселелеріне, уақытты тұтынуға және, мүмкін, зиянды науқасқа.

Осы зерттеу жаһандық ауырлық ауыртпалығындағы алғашқы ондыққа жататын жүрек-қан тамырлары ауруларының алдын-ала болжамында медициналық шешімдерді қолдау бойынша ML әдісін енгізу туралы болды. Мақсат - амбулаториялық қан қысымын (АБП) мониторингтік зерттеуден алынған деректерді жинау және алдын ала өңдеуді енгізу, диагностика мен болжамды медициналық диагнозға нақты қолдану арқылы қолданыстағы ML алгоритмін енгізу және жүрек-қан тамырлары өлімінің қауіп факторлары үшін жаңа болжамдық табалдырықтарды анықтау.

2.3 Кеуде рентгендік суреттердің ақпараттық қамтамасын құру

Туберкулезге арналған сандық суреттердің стандартты деректер базасы Шэньчжэнь қаласындағы № 3 халық ауруханасында, Гуандун медициналық колледжінде, Шэньчжэнь қ., Қытай Ұлттық медициналық кітапханасыда, АҚШ-тың Мэриленд қаласында құрылған. Кеуде қуысының рентген диагностикасы

амбулаториялық негізде жүзеге асырылады және Philips DR сандық диагностикалық жүйелерінің көмегімен күнделікті өмірдің бөлігі ретінде қабылданады.[7]

Рентген сәулелерінің саны:

- 336 туберкулез оқиғасы және
- 326 қалыпты жағдай.

Сурет параметрлері:

- Форматы: PNG
- Әрбір рентгенге арналған сурет өлшемі әр түрлі болады. Бұл шамамен 3000 x 3000.

Сурет файлының атауы CHNCXR _ ##### _ 0 / 1.png ретінде кодталады, мұндағы «0» қалыпты және «1» - қалыпты емес өкпе.

Барлық рентген суреттерінің клиникалық белгілері бірдей форматта мәтіндік файлдар ретінде сақталады: CHNCXR _ ##### _ 0 / 1.txt. Әрбір мәтіндік файл пациенттің жасын, жынысын және өкпе бұзылысын қамтиды.(өкпе туберкулезі).

IRB Details: Деректер жиынтығы деректер провайдерлері тарапынан анықталмаған және олардың мекемелерінде IRB бақылауынан босатылды. Деректер жиынтығын және жариялауды пайдалану NIH Human Research Research Office бағдарламасы жүргізген IRB сауалнамасынан (№5357) алынып тасталды.

2.4 Суреттің алдын-ала өңдеу әдістері

Автоматтандырылған табу жүйесі әдетте алдын ала өңдеу қадамдары үшін кіріс кескіндерін қабылдайды. Алдын-ала өңдеудің негізгі мақсаты суреттердің сапасын арттыру және ROI (қызығушылық аймағы) айқынырақ болу болып табылады. Осылайша, алдын ала өңдеу сапасы кейінгі процедуралардың орындалуына үлкен ықпал етеді. Кәдімгі рентген сәулелеріндегі нақты қолдану үшін импульсті жақсарту, суретті сегменттеу және сүйектің сөнуі сияқты типтік алдын ала өңдеу әдістері бар. Бұл бөлім осы әдістерді қысқаша сипаттайды.

Enhancement(Жақсарту).

Контраст, шеткі белгілері және бейнелердегі шуды жіктеу мен зақымдануды анықтауға үлкен ықпал етеді. Кеуде қуысының рентген суреттерінің жасырын және төмен қарама-қайшы аудандарында егжей-тегжейлі мәліметтер алу үшін құрылымдық ақпаратты ерекшелеп, шуды басу үшін кеуде радиографиясын күшейту керек. Кеуде қуысының рентгендік сәулеленуін жақсарту контрастты күшейтуді, шуды басуды, жиектік қайрауды және сүзгіні қамтиды. Контрастты жақсарту - суреттегі жарықтық мәндерінің ауқымын созу процесі, бұл кескіннің жалпы немесе жергілікті контрастын жақсартады және кескінді анық етеді. Кескінді қайрау кескіннің контурын толтырады, кескіннің шетін және сұр шәкілдің секірісін жақсартады, яғни суреттің толық мәліметтерін жақсартады.

Шуды басу - кескіннің егжей-тегжейін мүмкіндігінше сақтай отырып, суретті түсіру процесі. Кескінді күшейту процесінде нақты доменде немесе жиілік ауқымында орындалатын сүзу әрекеті пайдаланылуы мүмкін. Сүзу - бұл пиксельдің соңғы шығу мәнін анықтау үшін берілген пиксельдің айналасындағы пикселдердің мәнін пайдаланатын көршілес оператор. Жалпы алғанда, алдын-ала өңдеу қадамы ретінде, кескінді жақсарту, кескін мәліметтерін жоғалтпастан, шамадан тыс шуды енгізіп, бөлшектердің бұрмалануына әкеліп соқтырмай, қате диагноз қою жылдамдығын төмендетуге көмектеседі.

Сегментация

Кеуде радиографиясында, әдетте, ROI алу үшін анатомияны бөлу қажет. Кеуде радиографиясында тапсырмалардың әртүрлі мақсаттарға (өкпе түгіндерін, кардиомегалияны және аномальды асимметрияны анықтау және т.б.) байланысты сегменттеу бойынша көптеген зерттеулер бар. Кейбір зерттеулерде өкпе далалары сегменті ; басқалары өкпенің өрістерін немесе қабырғаның сұлбасын анықтайды; ал кейбіреулер диафрагманы немесе касофениялық бұрышты тікелей анықтауға тырысады. Өткір тастың сегменттеуі маңызды болып табылады, өйткені өкпенің өрістерінің ROI-лерін анықтайды, мұнда өкпе опастерлері, қуыстар, шоғырлану және түйіндер сияқты нақты радиологиялық белгілерді іздеуге болады. Сегментация әдістерін кескіндеу әдістеріне және машинада оқытуға негізделген әдістерге бөлуге болады. 1-кестеде пайдаланылған деректермен, бағалау шараларымен және тиісті сегменттеу нәтижелерімен бағаланатын сегменттеу әдістері келтірілген.

Машиналық оқытуға негізделген әдістер. Санатты пиксельге негізделген әдістер деп те атауға болады. Кеуде радиографиялары үшін әр пиксель өкпенің, жүректің, ортастинаның, диафрагманың және т.б. сияқты тиісті анатомиялық құрылымға тағайындалады. Классификатор пиксельдің сұр мәні, кеңістіктегі орналасу туралы ақпарат және статистикалық ақпарат сияқты түрлі мүмкіндіктерді пайдалана алады. Кейбір классификаторларда, мысалы, k-жақын көршісінің (KNN) классификаторы, векторлық машинаны қолдау (SVM), марковтық кездейсоқ өріс (MRF) немесе нейрондық желі (NN) классификаторды оқыту үшін мүмкіндіктер енгізіледі. Әдіс баяу машинада оқыту әдістеріне және терең оқытуға негізделген әдістерге бөлінеді.

Ұсақ механикаға негізделген әдістерде ерекшеліктерді шығару процесі интуитивті болып табылады және негізгі қиындық - бұл функциялардың тиісті санаттарын анықтау және оларды сенімді түрде алу. Mcnittgray және т.б. алдымен ерекшеліктерін пайдаланып, өкпе талшығының сегменттеу әдісін ұсынды. Қолданылатын ерекшеліктер - сұр түсті, жергілікті айырмашылықтың өлшемі және жергілікті текстураның өлшемі. KNN, желілік дискриминант талдауларын (LDA) және кері пропагандалық нейрондық желіні (NN) пайдалану арқылы әдіс CXR-дің әрбір пикселін бірнеше анатомиялық санаттардың біріне (жүрек, ішкі диафрагма, жоғарғы ортастинь, өкпе, қолғап және фон) біріктіреді. Сәйкес пайыздар тиісінше 70%, 70% және 76% жіктеушінің тиісінше. Әдебиеттерге ұқсас, Vittitoe et al. CXP-ді өкпеге немесе

жеңіл емес өкпеге бөлетін кеңістіктік және құрылымдық ақпаратты пайдаланды. Markov Random Field пайдаланды, жоғары сезімталдық, ерекше және дәлдікке ие модель құру үшін. Ши және басқалар. СРР-ларда өкпенің аймағын сегменттеу үшін бақылаусыз тәсіл қолданды. Гаусс ядроларының және кеңістіктегі шектеулердің негізінде анықталған С-құралдарымен (FCM) кластерлер арқылы өкпенің өрістерін бөлді. Бұл әдіс JSRT деректер жинағының 52 CXR-де сынақтан өткізілді және дәлдігі 0.978 ± 0.0213 болды.[9]

2.5 Ауруды анықтаудың өзіндік ерекшелігі

Кеуде қуысының рентгендерінде негізгі тыныс алу органдарының және қан айналымы органдарының бар екені анықталды. Жыл сайын миллиондаған адамдар көкірек ауруынан зардап шегеді. Туберкулез, интерстициальды өкпе ауруы (LID), пневмония, өкпе рагы және басқа аурулар әлемдегі ең көп кездесетін аурулар болып табылады. Кеуде радиографиясында аномалияның үш негізгі түрі бар: текстураның бұзылуы, аймақтың келбеті мен құрылымында, мысалы, интерстициальды зақымданулармен сипатталады; өкпелік түйіндер сияқты тығыздықтағы оқшауланған өзгерістер ретінде көрінетін фокальды ауытқулар; және аурудың қалыпты анатомиясын, мысалы, кардиомеальды құрылымын өзгертетін қалыпсыз пішінді. Кейде кеуде құрылымы мен формасы туберкулез сияқты белгілі бір аурумен бірге өзгереді.

Өкпе ісігінің табылуы

Дүниежүзілік онкологиялық диспансердің хабарлауынша, өкпенің қатерлі ісігі ерлердегі ең көп кездесетін қатерлі ісік және әйелдердің ішінде ең көп таралған үшінші түрі болып табылады. Бұл 5 жастағы 10-15% тірі қалуымен адам агрессивті адам ауруының біреуі болып табылады. Өкпенің түйіндері - өкпенің қатерлі ісігінің ерте көрінісі; Осылайша, өкпе рагына ерте диагностикалау және емдеу үшін өкпе түйіндерін ерте анықтау және диагностикалау өте маңызды. Нудультар жиі кеуде радиографиясына өкпе аймағындағы кіші дөңгелек немесе сопақ төмен контрасттық мата массасы ретінде пайда болады. Олар бірнеше ерекшеліктермен ерекшеленеді, мысалы, мөлшерде үлкен өзгерістер, тығыздықтағы үлкен өзгерістер, өкпе аймағындағы орналасудың белгісіздігі және т.б. Өкпе түтікшелерін жасау автоматты түрде анықтау алгоритмі медицина кескінінің АЖЖ саласында әрдайым күрделі, бірақ маңызды аспект болып табылады.

КАА клиникалық тұрғыдан рентгенге кеуде радиографиясына өкпе тіндерін анықтауда пайдалы екенін дәлелдеу үшін, Кобаяши және басқалар. байқаушы орындаушылық зерттеулер жүргізді. Бұл зерттеуде өкпе рентгенографиясы бар 60 науқас, өкпелік түйіндер және 60 кеуде тінсіз кеуде радиографиясының оқиғалары қолданылды. Сынауға қатысқан 16 рентгенолог компьютерлік көмексіз және компьютерлік интервенциясыз кеуде радиографиясын түсіндірді. Радиоаналлер өкпенің түтікшелерін операциялық сипаттамалы қисықтарды (ROC) пайдалану арқылы ажыратуға арналған.

Нәтижелер ААЖ жүйелері рентгенологтың өкпе түйіндерін анықтаудың 0.894-ден 0.940-ға дейінгі дәлдігін арттыратынын көрсетті.

Дәстүрлі өкпе түтікшелері САД-жүйелері бейнені алдын-ала өңдеуді (кеңейту және өкпенің сегменттеуі), кандидатты анықтауға және жалған позитивтерді азайту үшін мүмкіндіктерді алуды қамтиды. Кескінді алдын-ала өңдеудің мақсаты - түйіндерді жақсарту, өкпе тіндерін бөлу, басқа мата аймақтарын алып тастау және деректер шуын азайту. Нодульдерді анықтау кандидаты суреттегі көптеген түйіндерді анықтау үшін түрлі алгоритмдерді пайдаланады. Номерге алгоритм сезімталдығын арттыру үшін бұл қадам қате дабылдың жылдамдығын қатаң талап етпейді. Күдікті концентрациядағы жалған позитивті төмендеуі нодулдерді алып тастайды және жүйенің жалған оң жалған дабылын азайтады. Қазіргі кезде көптеген алгоритмдер анықталатын нәтижелердегі жалған позитивтерді азайта отырып, түйіндерді анықтау жылдамдығын жақсарту жолын қарастырады.

Туберкулезді анықтау.

Туберкулез (ТБ) бүкіл әлемде өлім-жітімнің тоғызыншы жетекші себебі болып табылады және АҚТҚ / ЖҚТБ-ның жоғары рейтингісінде бір жұқпалы агентінің жетекші себебі болып табылады. Жаһандық деңгейде 2016 жылы туберкулездің дамыған және аурудан қайтыс болған адамдардың үлесі 16% құрады, бұл 10,4 млн адам (90% ересектер, 65% ер, 10% АҚТҚ) туберкулезбен ауырады [78]. ЦРР-ларда туберкулезді анықтау қиын міндет болып табылады, өйткені ол түрлі көріністерге ие. Туберкулездің қалыптан тыс көріністері көбінесе жұқа милиарлы үлгілерден айқын эффузияларға дейін, концентрациялар, инфильтраттар және кавитация сияқты СХР-тарда өкпе құрылымы мен геометриясына әсер етеді.

Кейбір зерттеулер туберкулезді анықтады, оның формасы, құрылымы және өкпенің жергілікті ерекшеліктері, жалпы нәтижеге назар аударылды. Клиникалық рентген суреттерінің құрылымдық ерекшеліктерін визуалды диагностикалау және диагностикалау үшін радиологтарды имитациялау үшін, Rohmah et al. туберкулез немесе туберкулез емес тіндер ретінде суреттеу үшін дескриптор ретінде текстураның мүмкіндіктерін пайдаланды. Нәтижелер туберкулездің суреттің гистограммадағы статистикалық ерекшеліктеріне негізделі алатынын көрсетті.сегменттелген өкпе аймақтық құрылымдық ерекшеліктеріне негізделген туберкулез индексі (TI) ұсынды және шешім шырынының көмегімен қалыпты және аномальды ЦХР-ды жіктеді және 94,9% дәлдік жылдамдығын алды. Noor және басқалар. СХР суреттерінде туберкулезді анықтау үшін статистикалық интерпретация техникасын ұсынды. Олар алдымен CWR кескініне Wavelet түрлендірісін қолданды, Wavelet коэффициенттерінен 12 текстуралық шараны есептеді, PCA өлшемдерін азайтты және эллипсоид пен кемсітушілік функцияларын пайдалана отырып, дұрыс емес жіктеу ықтималдығын бағалады. Текстуралық ерекшеліктерді алудан басқа кейбір зерттеулер жіктеудің тиімділігін арттыру үшін кеуде радиографиясын алдын-ала тексеру үшін сүйек басуын қолданады.[8]

3 ПРОГРАММАНЫҢ СИПАТТАМАСЫ

3.1 Жалпы мәліметтер

Берілген дипломдық жұмыста Үлкен деректермен жұмыс кезінде Кеуде рентгендік суреттердің пайдаланып ақпараттық жүйе жасалды. Программада Кеуде рентгендік суреттері арқылы науқастың туберкулез ауруымен ауыратындығы немесе ауырмайтындығы орындалады. Программа Python ортасында жасалған.

Бұл программа үлкен мәліметтермен нақты және ыңғайлы жұмыс процесіне айналды. Бұл жұмысты жеңілдетеді және жұмыс орнын ұйымдастыруға көмектеседі.

3.2 Функционалдық тағайындалуы

Ағымдық функционалда Кеуде рентгендік суреттері арқылы науқастың туберкулез ауруымен ауыратындығы немесе ауырмайтындығы ақпараттық жүйесі автоматты түрде жұмыс жасайды және Рентген сәулелерінің саны:

- 336 туберкулез оқиғасы және
- 326 қалыпты жағдай.

Сурет параметрлері:

- Форматы: PNG
- Әрбір рентгенге арналған сурет өлшемі әр түрлі болады. Бұл шамамен 3000 x 3000.

Сурет файлының атауы CHNCXR _ ##### _ 0 / 1.png ретінде кодталады, мұндағы «0» қалыпты және «1» - қалыпты емес өкпе.

Барлық рентген суреттерінің клиникалық белгілері бірдей форматта мәтіндік файлдар ретінде сақталады: CHNCXR _ ##### _ 0 / 1.txt. Әрбір мәтіндік файл пациенттің жасын, жынысын және өкпе бұзылысын қамтиды.(өкпе туберкулезі).Программа интерфейсі жақсы дайындалған, қолданушылар саны шектелмеген үлкен мәліметтерге қорғау қойылған.

3.3 Логикалық құрылымның баяндалуы

Осы программада ең бірінші кесте құрылады, одан кейін Деректерді жүктеу , оқыту және тестілеу жүреді . Осыдан кейін жұмыс туралы толық ақпарат тіркеледі.

Осы программада ең бірінші керек суреттерді бір бірімен байланыстырамыз. Бізде 704 маска бар, бірақ 800 сурет. Сондықтан біз Маскадан 1-1 суретке математика түрінде байланыс жасаймыз. Осыдан кейін жұмыс туралы толық ақпарат тіркеледі.

3.4 Қолданылатын техникалық құрал-жабдықтар

Мәліметтер базасына кіру үшін Asus in search of incredible 500 GB ноутбугы қолданылады. Оны басып шығару үшін HP LeaserJet 1018 принтері қолданылды.

3.5 Шақыру және жүктеу

Программаны шақыру үшін Anaconda Prompt-ді жүктеу қажет. Файл менюінен Jupyter notebook ашылады.

```
Anaconda Prompt - jupyter notebook

(base) C:\Users\nazna>activate 3point6
(3point6) C:\Users\nazna>jupyter notebook
[I 22:07:33.855 NotebookApp] Serving notebooks from local directory: C:\Users\nazna
[I 22:07:33.855 NotebookApp] The Jupyter Notebook is running at:
[I 22:07:33.859 NotebookApp] http://localhost:8888/?token=3d9ca6a09f40d9158960a79b5f
[I 22:07:33.860 NotebookApp] Use Control-C to stop this server and shut down all ker
[C 22:07:33.868 NotebookApp]
```

Сурет 3.1 - Программаны шақыру және жүктеу

3.6 Кіріс мәліметтер

Кіріс мәліметтері болып келесілер табылады: Lung Segmentation, train, masks және test.

Lung Segmentation			Поиск: L
Имя	Дата изменения	Тип	
.ipynb_checkpoints	15.04.2019 21:09	Папка с файлами	
ClinicalReadings	03.02.2019 13:54	Папка с файлами	
masks	03.02.2019 13:59	Папка с файлами	
test	03.02.2019 14:00	Папка с файлами	
train	03.02.2019 13:59	Папка с файлами	
cxr_reg_weights.best.hdf5	11.05.2019 14:33	Файл "HDF5"	
Lung Segmentation	12.05.2019 10:00	Файл "IPYNB"	
Lung Segmentation_v1	26.04.2019 11:26	Файл	
NLM-ChinaCXSet-ReadMe	25.01.2019 12:37	Документ Micros...	
NLM-MontgomeryCXSet-ReadMe	25.01.2019 12:23	Chrome HTML Do...	

Сурет 3.2 - Кіріс мәліметтері

3.7 Шығыс мәліметтер

Өртүрлі процедуралар көмегімен қараулар мен өзгерістер қалыптасады. Қалыптасқан өзгерістердің барлық нұсқалары Қосымшада көрсетілген.

4 ПРАКТИКАЛЫҚ БӨЛІМ

4.1 PYTHON бағдарламалау тілінің сипаттамасы

Python - жалпыға бірдей жоғары деңгейлі бағдарламалау тілі, ол деректер ғылымында және машина алгоритмдерін жобалауда жиі қолданылады. Әрбір тапсырманы орындаудың бірнеше жолын ұсынатын көптеген модульдер мен кітапханалар бар. Бұл Python және оның кітапханаларына numpy, scipy, pandas, matplotlib сияқты жылдам кіріспе береді және нақты әлемдік проблемаларды шешетін машина алгоритмдерін әзірлеу үшін қалай қолдануға болатынын түсіндіреді. Машиналық оқытуда жұмысты бастау үшін Python бағдарламасын пайдаланудың ең жақсы жолы - бұл жобаны аяқтау.

Python бағдарламалау тілінде жаңадан жұмысты бастағандар кішігірім жобадан бастауы тиіс. Ол үшін арнайы кітаптар оқып және курстардан өткен тиімді болып саналады. Бұл курстармен мен кітаптар сізге көптеген мәліметтер мен үзінділер береді, бірақ олардың бәрі бір-біріне қаншалықты сәйкес келетінін көре алмайсыз.

Сіз өзіңіздің дербес деректеріңізді машиналық оқытуда қолданған кезде, жоба басталады. Машиналық оқытуда жобаны желілік болмауы мүмкін, бірақ ол белгілі бірнеше қадамдарға ие болады :

- Мәселені анықтау.

- Деректерді дайындау.

- Алгоритмдерді бағалау.

- Нәтижелерді жақсарту.

- Нәтижелер.

Жұмысты бастау үшін ең алдымен Python бағдарламалау тіліннің 3,6 нұсқасын жүктейміз және керекті кітапханаларды жүктейміз. Машиналық оқытуда қажетті Python тілінің қажетті кітапханаларын орнықтырамыз :

- NumPy: Python көмегімен сандық өңдеу үшін.

- PIL: қарапайым суреттерді өңдеу кітапханасы.

- Scikit-learn: бүгінгі күнді қамтитын компьютерлік оқыту алгоритмі бар (бізде 0.20+ нұсқасы қажет болады, сол себептен - төмендегіні жаңартып алу керек).

- Keras және TensorFlow: терең білім алу үшін. TensorFlow бағдарламасының CPU нұсқасы бүгінгі күннің үлгісіне сай келеді.

- OpenCV: Біз осы блог жазбасы үшін OpenCV қолданбаймыз, imutils оған (келесі оқ) байланысты. Осының арқасында сіз OpenCV-ны орнату үшін пәрменді пайдалануыңызға болады, тек OpenCV-ді толық орнатпаңыз және оны баптай алмайтыныңызды есте сақтаңыз. Суретте орнатылған Python бағдарламалау тіліннің және кітапханалдың версиясын тексереміз.

```

1 # Check the versions of libraries
2
3 # Python version
4 import sys
5 print('Python: {}'.format(sys.version))
6 # scipy
7 import scipy
8 print('scipy: {}'.format(scipy.__version__))
9 # numpy
10 import numpy
11 print('numpy: {}'.format(numpy.__version__))
12 # matplotlib
13 import matplotlib
14 print('matplotlib: {}'.format(matplotlib.__version__))
15 # pandas
16 import pandas
17 print('pandas: {}'.format(pandas.__version__))
18 # scikit-learn
19 import sklearn
20 print('sklearn: {}'.format(sklearn.__version__))

```

Here is the output I get on my OS X workstation:

```

1 Python: 3.6.8 (default, Dec 30 2018, 13:01:55)
2 [GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)]
3 scipy: 1.1.0
4 numpy: 1.15.4
5 matplotlib: 3.0.2
6 pandas: 0.23.4
7 sklearn: 0.20.2

```

Сурет 4.1 - Python бағдарламалау тіліннің кітапханалары

Машиналық оқытуда жұмысты бастау үшін қолданылатын барлық модульдерді, функцияларды және кітапханаларды импорттаймыз. Ол үшін төмендегі суреттегідей арнайы кодты тереміз

```

1 # Load libraries
2 import pandas
3 from pandas.plotting import scatter_matrix
4 import matplotlib.pyplot as plt
5 from sklearn import model_selection
6 from sklearn.metrics import classification_report
7 from sklearn.metrics import confusion_matrix
8 from sklearn.metrics import accuracy_score
9 from sklearn.linear_model import LogisticRegression
10 from sklearn.tree import DecisionTreeClassifier
11 from sklearn.neighbors import KNeighborsClassifier
12 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
13 from sklearn.naive_bayes import GaussianNB
14 from sklearn.svm import SVC

```

Сурет 4.2 - Python бағдарламалау тіліннің кітапханаларын импорттау

4.2 Anaconda Python бағдарламалау тілінің сипаттамасы

2011 жылдан бастап, Python рір-да жазылған бағдарламалық пакеттерді орнатуға және басқаруға арналған пакетті басқару жүйесін енгізді. Дегенмен, сандық есептеулер үшін Python-да жазылған бірнеше тәуелділік болды және қиындық туғызды, сондықтан рір-тің бастапқы шығарылымында мәселені өзі шеше алмады. Бұл проблеманы айналып өту үшін, Continuum Analytics Anaconda-ды шығарды, Anaconda қолданылатын ғылыми қосымшаларға және

Conda пакетті және қоршаған ортаны басқару жүйесіне бағытталған Python бағдарламалау тілінің дистрибутиві. Pip-тің соңғы нұсқалары дөңгелектерді пайдалану арқылы сыртқы тәуелділіктерді өңдеуі мүмкін екендігін ескере отырып, бірақ Anaconda арқылы деректер ғылымына арналған маңызды кітапханаларды біртіндеп орнатуға мүмкіндік берді.

Conda Anaconda Python Distribution-ке тығыз байланысты болса да, екеуі әртүрлі мақсаттарға ие және әр түрлі жобалар болып саналады :

Anaconda - бұл PyData экожүйесінде, соның ішінде Python-дің бағдарламалық жасақтамасын толыққанды тарату үшін қолданылады, сондай-ақ, үшінші тараптың ашық бастапқы коды бар бірнеше жобалар үшін екіліктері бар. Anaconda басқа Miniconda да бар, ол ең төменгі Python таратылымы болып табылады, соның ішінде негізінен Conda және оның тәуелділіктері, сізге тек қана пакеттерін нөлден бастап орнатуға болады.

Conda - Anaconda немесе Miniconda үлестірілмей орнатылуы болуы мүмкін пакет, тәуелділік және қоршаған ортаны басқару жүйесі. Ол Windows, macOS және Linux жүйесінде жұмыс істейді және Python бағдарламалары үшін жасалған, бірақ ол кез келген тілге арналған бағдарламалық жасақтаманы жинақтап, таратуы мүмкін. Негізгі мақсат - сыртқы тәуелділіктерді шешудің бағдарламалық жасақтаманың алдын ала жасалынған нұсқаларын жүктеу арқылы оңай шешу.

Бұл мағынада, пакеттерді тіл-агностикалық түрде табуға және орнатуға көмектесетін APT немесе YUM сияқты жалпы мақсаттағы пакет менеджерінің кросс-платформалық нұсқасына ұқсайды. Сондай-ақ, Conda қоршаған ортаны басқарушы болып табылады, сондықтан сізге Python басқа нұсқасын қажет ететін буманы қажет болса, Conda көмегімен Python-тың әдеттегі нұсқасын сақтай отырып, Python-тың мүлдем басқа нұсқасымен жеке ортаны орнатуға болады әдепкі ортаңыз.

Ең бірінші қадам болып платформаға арналған Anaconda Python пакетін жүктеп аламыз. Anaconda - бұл ғылыми Python үшін еркін және ыңғайлы орта.

1.conda install Anaconda (Python 3.6 нұсқасы) жүктейміз

Келесі қадамда біз Anaconda Python бағдарламалық жасақтамасын орнатамыз.Орнатуды жүктегеннен кейін орнату өте қарапайым және жылдам жүреді. Орнатуды ашып, шебер нұсқауларын орындаңыз.

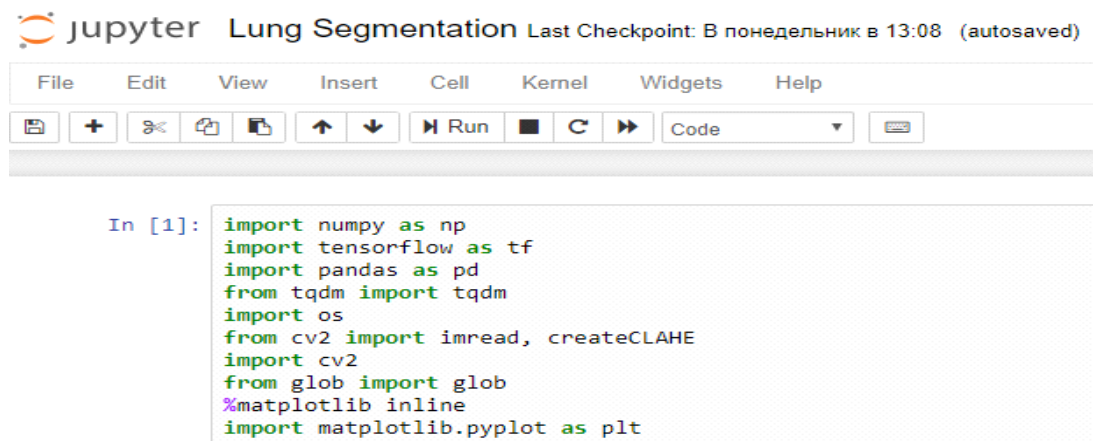
Ескерту: Ол автоматты түрде Python және кейбір негізгі кітапханаларды орнатады. Жүйеңізге сәйкес 5-тен 10 минут немесе бірнеше уақыт кетуі мүмкін.Келесі кезекте керекті қосымша кітапханаларды орнатамыз. Келесі командаларды теру үшін Anaconda prompt терезесін ашамыз conda create -n tensorflow pip python=3.6 кодын тереміз. Орнатылғандығын тексері үшін келесі төмендегі суретте көрсетілгендей командаларды еңгіземіз

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

Сурет 4.3 – Кітапханалардың орнатылуын тексеру

4.3 Программалық қамтаманың құрылымы

Бірінші кезекте төмендегі суретте көрсетілгендей қажетті кітапханалар топтамаларын импорттаймыз.



Сурет 4.4 – Кітапханаларды импорттау

Бізде 704 маска бар, бірақ 800 сурет. Сондықтан төмендегі суретте көрсетілгендей Маскадан 1-1 суретке математика түрінде байланыс жасаймыз.

```
In [6]: testing_files = set(os.listdir(image_path)) & set(os.listdir(mask_path))
training_files = check

def getData(X_shape, flag = "test"):
    im_array = []
    mask_array = []

    if flag == "test":
        for i in tqdm(testing_files):
            im = cv2.resize(cv2.imread(os.path.join(image_path,i)),(X_shape,X_shape))[:, :,0]
            mask = cv2.resize(cv2.imread(os.path.join(mask_path,i)),(X_shape,X_shape))[:, :,0]

            im_array.append(im)
            mask_array.append(mask)

        return im_array,mask_array

    if flag == "train":
        for i in tqdm(training_files):
            im = cv2.resize(cv2.imread(os.path.join(image_path,i.split("_mask")[0]+".png")),(X_shape,X_shape))[:, :,0]
            mask = cv2.resize(cv2.imread(os.path.join(mask_path,i+".png")),(X_shape,X_shape))[:, :,0]

            im_array.append(im)
            mask_array.append(mask)

        return im_array,mask_array
```

Сурет 4.5 - Маскадан 1-1 суретке математика түрінде байланыс

Бұрын Мен Montgomery деректер жиынтығын болжау кезінде Шэньчжэнь деректер жиынтығында жаттығуға жоспарлаған едім. Дегенмен, деректердің сипаты екі жиынтықта да әртүрлі болды. Шэньчжэнь деректер жиынтығынан алынған суреттер Montgomery деректер жиынтығымен салыстырғанда суреттерге азырақ түсті. Осылайша, мен екі деректер жиынтығын бөлек жүктеймін және кездейсоқтық түрінде біріктірім.

```
In [7]: # выполнить проверку работоспособности
def plotMask(X,y):
    sample = []

    for i in range(6):
        left = X[i]
        right = y[i]
        combined = np.hstack((left,right))
        sample.append(combined)

    for i in range(0,6,3):
        plt.figure(figsize=(25,10))

        plt.subplot(2,3,1+i)
        plt.imshow(sample[i])

        plt.subplot(2,3,2+i)
        plt.imshow(sample[i+1])

        plt.subplot(2,3,3+i)
        plt.imshow(sample[i+2])

    plt.show()
```

Сурет 4.6 - Деректер жиынтығын бөлек жүктеу және кездейсоқтық түрінде біріктіру.

Келесі қадам болып деректерді жүктеу , оқыту және тестілеу болып келеді. Keras-та Lung Segmentation деректер жиынымен жұмыс істеу үшін кірістірілген құралдарды пайдалана отырып, csv форматында деректерді жүктей алмайсыз. Сондықтан csv файлындағы деректерді numpy кітапханасынан, loadtxt функциясынан оқып шығамыз:

```
In [8]: # Загрузка данных обучения и тестирования
dim = 256*2
X_train,y_train = getData(dim,flag="train")
X_test, y_test = getData(dim)
```

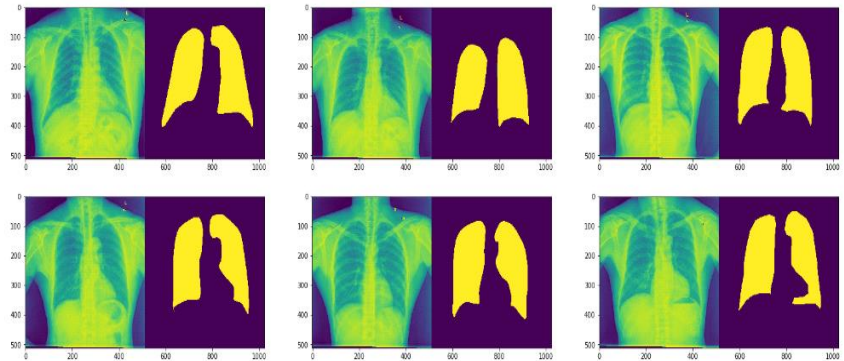
100%	<div></div>	566/566 [03:21<00:00, 2.20it/s]
100%	<div></div>	138/138 [01:46<00:00, 2.75it/s]

Сурет 4.7 - Деректерді жүктеу , оқыту және тестілеу

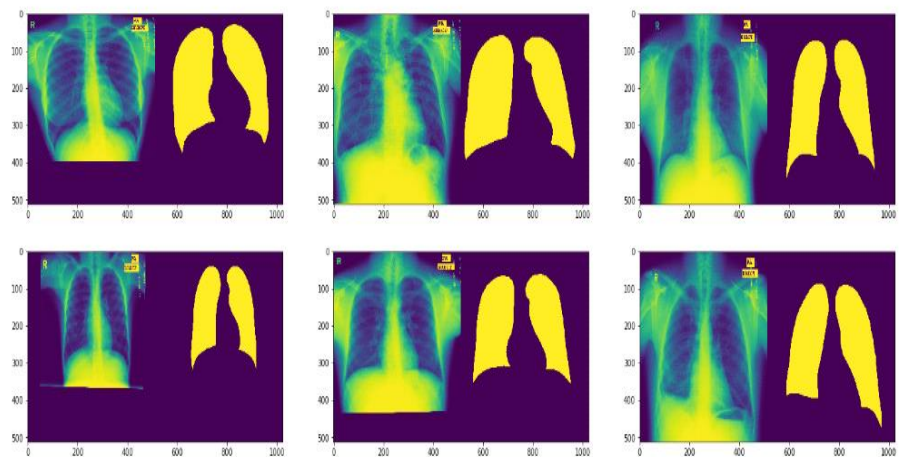
Келесі қадамда жұмысты тексеріп көруіміз қажет. Деректер жиынтығынан суреттер туралы деректерді таңдап, тегіс векторлық 784 пикселден пішімді 28x28 екі өлшемді матрицаға дейін өзгертеміз:


```
In [9]: print("training set")
        plotMask(X_train,y_train)
        print("testing set")
        plotMask(X_test,y_test)
```

training set



testing set



Сурет 4.8 - Жұмысты тексеріп көру

Келесі қадам ретінде мен U-Nets сегменттік желісін қолданамын. Оларды оқуға болады. Жинақтардың екеуі де дұрыс көрінеді. Оларды біріктіріп, әрі қарай оларды бірыңғай деректер жиынтығы ретінде қолдандым. Ең бірінші қосымша кітапханалар жиынтығын импорттап аламыз.

```
In [11]: from keras.models import *
        from keras.layers import *
        from keras.optimizers import *
        from keras import backend as keras
        from keras.preprocessing.image import ImageDataGenerator
        from keras.callbacks import ModelCheckpoint, LearningRateScheduler

        Using TensorFlow backend.
```

Сурет 4.9 - кітапханаларды импорттау

Kaggle үшін Кеуде рентгендік суреттерін тану үшін Keras-Асс 0.997-ден (ең жоғарғы 8%) Kaggle конвультивті нейрондық желісін қолданамыз. Мынадай белгіленеді:

```
In [12]: def dice_coef(y_true, y_pred):
y_true_f = keras.flatten(y_true)
y_pred_f = keras.flatten(y_pred)
intersection = keras.sum(y_true_f * y_pred_f)
return (2. * intersection + 1) / (keras.sum(y_true_f) + keras.sum(y_pred_f) + 1)

def dice_coef_loss(y_true, y_pred):
return -dice_coef(y_true, y_pred)

def unet(input_size=(256,256,1)):
inputs = Input(input_size)

conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv1)
pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(pool1)
conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv2)
pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool2)
conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv3)
pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)

conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(pool3)
conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv4)
pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)

conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(pool4)
conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv5)
```

Сурет 4.10 - U-Nets сегменттік желісін қолдану

Келесі қадамда Unet үлгісін құрастырып және оқытымыз.

```
model = unet(input_size=(512,512,1))
model.compile(optimizer=Adam(lr=1e-5), loss=dice_coef_loss,
              metrics=[dice_coef, 'binary_accuracy'])
model.summary()
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 512, 512, 1)	0	
conv2d_1 (Conv2D)	(None, 512, 512, 32)	320	input_1[0][0]
conv2d_2 (Conv2D)	(None, 512, 512, 32)	9248	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 256, 256, 32)	0	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 256, 256, 64)	18496	max_pooling2d_1[0][0]
conv2d_4 (Conv2D)	(None, 256, 256, 64)	36928	conv2d_3[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 128, 128, 64)	0	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 128, 128, 128)	73856	max_pooling2d_2[0][0]
conv2d_6 (Conv2D)	(None, 128, 128, 128)	147584	conv2d_5[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 64, 64, 128)	0	conv2d_6[0][0]
conv2d_7 (Conv2D)	(None, 64, 64, 256)	295168	max_pooling2d_3[0][0]
conv2d_8 (Conv2D)	(None, 64, 64, 256)	590080	conv2d_7[0][0]
Total params: 7,759,521			
Trainable params: 7,759,521			
Non-trainable params: 0			

Сурет 4.11 - Unet үлгісін құрастырып және оқыту

Ең маңызды қадам ретінде келесі қадамды атап айтқан дұрыс болады. Үлгіні оқыту. Бастапқыда 60-40 жаттығу сынағын пайдаланып, -0.97 жоғалттым. Бізде жіктеу міндеті бар, сондықтан біз соңғы желі деңгейіндегі белсендіру функциясы ретінде softmax және категориялы кросс-энтропия қателік функциясы ретінде пайдаланамыз.

Оқу барысында біз екі кері шақыру функцияларын қолданамыз: ModelCheckpoint және ReduceLROnPlateau. Бірінші функция оқыту процесінде нейрондық желіні сақтау үшін қолданылады және келесідей анықталады. Нейрондық желі mnist-cnn.hdf5 деп аталатын файлға сақталған болса, тест

жинағында дұрыс жауап үлесінің ағымдағы мәні (monitor = 'val_acc') алдыңғы кезеңдерге қарағанда жақсы болса (set_best_only = True параметрімен орнатылған). Verbose = 1 параметрі функция оның әрекеттерінің журналын басып шығаратынын айтады. ReduceLROnPlateau функциясы оқыту үдерісі желі сапасын (плато) жақсартпаса, оқу жылдамдығын төмендету үшін қолданылады.

Сынақ деректеріне дұрыс жауаптардың үлес салмағын қадағалап отырамыз (monitor = 'val_acc') және егер бұл көрсеткіш үш дәуірден (төзімділік = 3) артық болмаса, оқу жылдамдығының параметрі 0,5 (фактор = 0,5) көбейтіледі. Бұл оқу жылдамдығының параметрі 0.1 (min_lr = 0.1) жеткенше жалғасады. Нейрондық желіні үйрету үшін, алдыңғы бөлімде сұраған детанген генераторы пайдаланылады:

```
from IPython.display import clear_output
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split

model.compile(optimizer=Adam(lr=2e-4),
              loss=[dice_coef_loss],
              metrics = [dice_coef, 'binary_accuracy'])

train_vol, validation_vol, train_seg, validation_seg = train_test_split((images-127.0)/127.0,
                                                                    (mask>127).astype(np.float32),
                                                                    test_size = 0.1,random_state = 2018)

train_vol, test_vol, train_seg, test_seg = train_test_split(train_vol,train_seg,
                                                            test_size = 0.1,
                                                            random_state = 2018)
```

Сурет 4.12 - Үлгіні оқыту

Үлгіні оқыту 20 дәуірі жалғасуда, бұл қажет емес. Сондықтан, соңғы дәуірде қайта даярлау жүреді. Оқу үрдісінде берілген журналға сәйкес қайта даярлауды анықтай аласыз:

```
loss_history = model.fit(x = train_vol,
                        y = train_seg,
                        batch_size = 16,
                        epochs = 20,
                        validation_data =(test_vol,test_seg) ,
                        callbacks=callbacks_list)

clear_output()
model.save("model_v7.h5")
```

Train on 569 samples, validate on 64 samples
Epoch 1/20
569/569 [=====] - ETA: 2:07:40 - loss: -0.3406 - dice_coef: 0.3406 - binary_accuracy: 0.259 - ETA: 2:06:36 - loss: -0.3260 - dice_coef: 0.3260 - binary_accuracy: 0.244 - ETA: 2:03:24 - loss: -0.3354 - dice_coef: 0.3354 - binary_accuracy: 0.287 - ETA: 1:59:48 - loss: -0.3376 - dice_coef: 0.3376 - binary_accuracy: 0.360 - ETA: 1:56:23 - loss: -0.3394 - dic

Сурет 4.13 – Үлгіні оқыту

19-шы кезеңде сынақ жиынтығына дұрыс жауаптардың үлесі 0,99595-ден 0,99619 дейін жақсарды, сондықтан желі сақталды. Бірақ 20-шы кезеңде бұл көрсеткіш 0,9955-ге дейін төмендеді. Осылайша, оқудың 20 кезеңде аяқталғаннан кейін, желі үздік жұмыс сапасын қамтамасыз етпейді. Сондықтан тест-жинақтың қолжазбалы нөмірлерін танымас бұрын, ең жақсы желілік опцияны жүктеп алу керек.

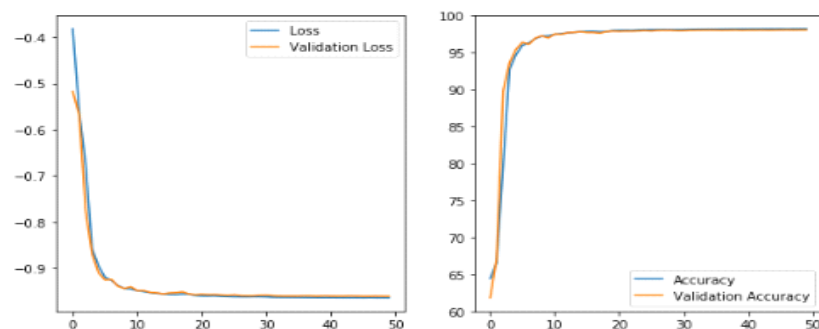
Келесі қадамда көрсеткішті бақылау және бағалау. Бинарлық классификаторларды бағалау binary ассигасу төлсипатты тағайындаудың екі әдісін салыстырады, олардың біреуі әдеттегі әдіс, ал екіншісі зерттеледі. Жіктеушінің немесе болжаудың өнімділігін өлшеу үшін пайдаланылатын көптеген метрикалар бар; әртүрлі өрістер әртүрлі мақсаттарға байланысты нақты метрикалар үшін түрлі артықшылықтарға ие. Мысалы, медицинада сезімталдық пен спецификация жиі пайдаланылады, ал информатикада дәлдік пен еске салу артықшылық береді. Маңызды айырмашылық пропорцияға тәуелсіз метрикалар арасында (халықтың әрқайсысының қаншалықты жиі кездесетінін) және таралуына тәуелді көрсеткіштерді - екі түрді де пайдалы, бірақ оларда әр түрлі сипаттар бар.

```
In [13]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (10, 5))
ax1.plot(loss_history.history['loss'], '-', label = 'Loss')
ax1.plot(loss_history.history['val_loss'], '-', label = 'Validation Loss')
ax1.legend()

ax2.plot(100*np.array(loss_history.history['binary_accuracy']), '-', label = 'Accuracy')
ax2.plot(100*np.array(loss_history.history['val_binary_accuracy']), '-', label = 'Validation Accuracy')
ax2.legend()
```

Out[13]:

<matplotlib.legend.Legend at 0x7f5ce83ba978>



Сурет 4.14 - Көрсеткішті бақылау және бағалау

```
def recognition(img):
    resized_img = cv2.resize(img, (224, 224))
    print(resized_img.shape)
    img = resized_img.reshape(1,224,224,3)
    img = img.astype('float32')
    img /= 255
    preds = model.predict(img)
    labels = [0,1]
    ind = preds[0].argmax()
    print("Отбет : ",labels[ind], ", точность :", round(preds[0][ind]*100,2),"%")
    acc = round(preds[0][ind]*100,2)
    if acc > 75:
        res = ""+str(labels[ind])+", "+str(acc) + "%"
    else:
        res = "0, 0%"
    return res
```

```
(224, 224, 3)
[[0.9960907 0.00390928]]
ответ : 0 , точность : 99.61 %
```

Сурет 4.15 - Модельдің жауабын шығару

```
img = cv2.imread('test/image_1_1999.png')
#cv2.imshow("img", img)
#cv2.waitKey(0) & 0xFF
resized_img = cv2.resize(img, (224, 224))
print(resized_img.shape)
img = resized_img.reshape(1,224,224,3)
img = img.astype('float32')
img /= 255
preds = model.predict(img)
labels = [0, 1]
ind = preds[0].argmax()
print(preds)
import numpy as np

if round(preds[0][ind]*100,2)>75:
    print("ответ : ",labels[ind], ", точность :", round(preds[0][ind]*100,2),"%")
else:
    print("объект не распознан! Преблизительно - ", labels[ind], "точность :", round(preds[0][ind]*100,2),"%")

(224, 224, 3)
[[0.2814194 0.7185806]]
объект не распознан! Преблизительно - 1 точность : 71.86 %
```

Сурет 4.16 - Модельдің жауабын шығару

Төменде туберкулезбен ауырмайтын және туберкулезбен ауыратын бір бейне бейнеленген.

```
#Ниже показано одно изображение с положительным туберкулезом и одно изображение с отрицательным туб
from IPython.display import Image
pos_image = np.random.choice(tb_positive,1)
neg_image = np.random.choice(tb_negative,1)
print("Image %s is positive on tuberculosis." % pos_image[0])
display(Image(os.path.join("C:/Users/nazna/Desktop/1/CXR_png",pos_image[0]),width=256,height=256))
print("Image %s is negative on tuberculosis." % neg_image[0])
display(Image(os.path.join("C:/Users/nazna/Desktop/1/CXR_png",neg_image[0]),width=256,height=256))

Image CHNCXR_0453_1.png is positive on tuberculosis.
```



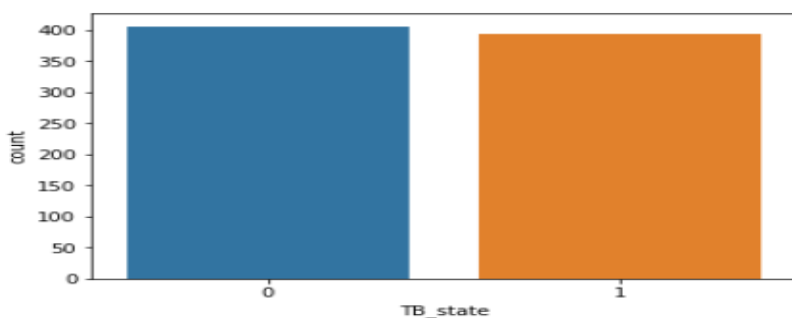
Image CHNCXR_0079_0.png is negative on tuberculosis.



Сурет 4.17 – Оң және теріс нәтижелер

Бұрын айтылғандай, туберкулезбен сырқаттанғандардың үлесі барлық рентгендердің 50% -ын құрайды.

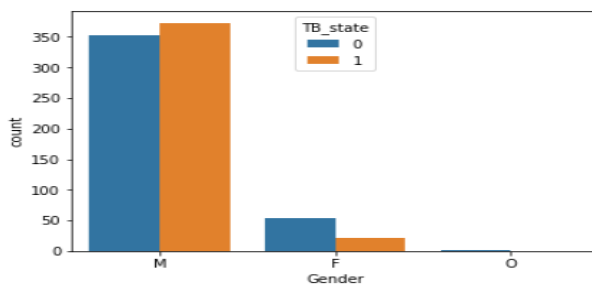
```
#Как упоминалось ранее, доля туберкулезных больных составляет  
sns.countplot(x='TB_state', data=img_df)|  
<matplotlib.axes._subplots.AxesSubplot at 0x13a5794a828>
```



Сурет 4.18 – Пайыздық көрсеткіші

Бізде ерлердің рентгендік суретті көп, сондай-ақ пациенттердің оң туберкулездің көрсеткіші ерлерде (51,4%) әйелдерге қарағанда (28,4%) жоғары.

```
#У нас больше пациентов мужского пола, и показатель положительного тубер.  
sns.countplot(x='Gender', hue='TB_state', data=img_df)|  
<matplotlib.axes._subplots.AxesSubplot at 0x13a579d4ef0>
```



Сурет 4.19 – Пайыздық көрсеткіші

ҚОРЫТЫНДЫ

Бүгінгі күні машиналық оқыту алгоритмдері компьютерлермен адамдармен қарым-қатынас жасауға, автомобильдерді дербес басқаруға, спорттық матч туралы есептерді жазуға, жариялауға және террорист күдіктілерді табуға мүмкіндік береді. Сондықтан қазіргі таңда машиналық оқыту көптеген өндірістерге және олардағы жұмысқа қатты әсер ететініне сенімдімін, сондықтан әрбір менеджер кем дегенде қандай машинада оқытуды және оны қалай дамитындығын білуі керек.

Осы дипломдық жұмыстың барысында келесі нәтижелерге қол жеткізілді:

- Машина жасаудың қарастырылған және қолданылатын әдістері (К-құралдары, CNN).
- Машина жасау алгоритмдері туралы әдебиет және мақалалар зерттелді.
- Алгоритмдердің дәлдігін салыстыру.
- Терең оқу алгоритмдері үшін түрлі дайын кітапханалар мен шешімдер сыналды.
- Жоғары тану дәлдігіне қол жеткіздім (95%) .

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Polyak B.T., Tsybakov. Optimal Orders of Accuracy for Search Algorithms of Stochastic Optimization. Probl. Peredachi Inf., 1990.
- 2 Robbins H., Monro S. Stochastic Approximation Method. — Ann. Math. Statist, 1951.
- 3 Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. // Journal of Machine Learning Research. — Vol. 12. — P. 2825–2830, 2011.
- 4 Я.З. Цыпкин. Адаптация и обучение в автоматических системах. — Москва: Наука, 1968.
- 5 Маркова С.В., Жигалов К.Ю. применение нейронной сети для создания системы распознавания изображений // Фундаментальные исследования. – 2017. – № 8-1. – С. 60-64.
- 6 Айвазян С.А., Бухстабер В.М., Энюков И.С., Мешалкин Л.Д. Қолданбалы статистика: Өлшемді жіктеу және азайту. - М.: Қаржы және статистика, 1989.
- 7 А. Шалимов, М.А. Бессонов. Аудио хабарламалардың тілін анықтау үшін технологияларды дамытудың жай-күйін және перспективаларын талдау
- 8 <https://www.kaggle.com/pavansanagapati/urban-sound-classification/kernels>
- 9 <http://www.fundamental-research.ru/ru/article/view?id=41621>

Қосымша А

```
import numpy as np
import tensorflow as tf
import pandas as pd
from tqdm import tqdm
import os
from cv2 import imread, createCLAHE
import cv2
from glob import glob
%matplotlib inline
import matplotlib.pyplot as plt

image_path = os.path.join("C:/Users/nazna/Desktop/Lung
Segmentation/CXR_png")
mask_path = os.path.join("C:/Users/nazna/Desktop/Lung
Segmentation/", "masks/")

# we have 704 masks but 800 images. Hence we are going to
# make a 1-1 correspondance from mask to images, not the usual other way.
images = os.listdir(image_path)
mask = os.listdir(mask_path)
mask = [fName.split(".png")[0] for fName in mask]
image_file_name = [fName.split("_mask")[0] for fName in mask]
check = [i for i in mask if "mask" in i]
print("Total mask that has modified name:", len(check))
testing_files = set(os.listdir(image_path)) & set(os.listdir(mask_path))
training_files = check

def getData(X_shape, flag = "test"):
    im_array = []
    mask_array = []

    if flag == "test":
        for i in tqdm(testing_files):
            im = cv2.resize(cv2.imread(os.path.join(image_path, i)), (X_shape, X_shape))[:, :, 0]
            mask = cv2.resize(cv2.imread(os.path.join(mask_path, i)), (X_shape, X_shape))[:, :, 0]

            im_array.append(im)
            mask_array.append(mask)

    return im_array, mask_array
```

```

if flag == "train":
    for i in tqdm(training_files):
        im =
cv2.resize(cv2.imread(os.path.join(image_path,i.split("_mask")[0]+".png")), (X_shape, X_shape))[:, :, 0]
        mask =
cv2.resize(cv2.imread(os.path.join(mask_path,i+".png")), (X_shape, X_shape))[:, :, 0]

        im_array.append(im)
        mask_array.append(mask)

    return im_array, mask_array
#perform sanity check

def plotMask(X,y):
    sample = []

    for i in range(6):
        left = X[i]
        right = y[i]
        combined = np.hstack((left,right))
        sample.append(combined)

    for i in range(0,6,3):

        plt.figure(figsize=(25,10))

        plt.subplot(2,3,1+i)
        plt.imshow(sample[i])

        plt.subplot(2,3,2+i)
        plt.imshow(sample[i+1])

        plt.subplot(2,3,3+i)
        plt.imshow(sample[i+2])

    plt.show()
# Load training and testing data

```

А қосымшасының жалғасы

```
dim = 256*2
X_train,y_train = getData(dim,flag="train")
X_test, y_test = getData(dim)
print("training set")
plotMask(X_train,y_train)
print("testing set")
plotMask(X_test,y_test)
X_train = np.array(X_train).reshape(len(X_train),dim,dim,1)
y_train = np.array(y_train).reshape(len(y_train),dim,dim,1)
X_test = np.array(X_test).reshape(len(X_test),dim,dim,1)
y_test = np.array(y_test).reshape(len(y_test),dim,dim,1)
assert X_train.shape == y_train.shape
assert X_test.shape == y_test.shape
images = np.concatenate((X_train,X_test),axis=0)
mask = np.concatenate((y_train,y_test),axis=0)
from keras.models import *
from keras.layers import *
from keras.optimizers import *
from keras import backend as keras
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, LearningRateScheduler
def dice_coef(y_true, y_pred):
    y_true_f = keras.flatten(y_true)
    y_pred_f = keras.flatten(y_pred)
    intersection = keras.sum(y_true_f * y_pred_f)
    return (2. * intersection + 1) / (keras.sum(y_true_f) + keras.sum(y_pred_f) + 1)

def dice_coef_loss(y_true, y_pred):
    return -dice_coef(y_true, y_pred)

def unet(input_size=(256,256,1)):
    inputs = Input(input_size)

    conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
    conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(pool1)
    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv2)
```

```

pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool2)
conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv3)
pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)

conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(pool3)
conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv4)
pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)

conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(pool4)
conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv5)

up6 = concatenate([Conv2DTranspose(256, (2, 2), strides=(2, 2),
padding='same')(conv5), conv4], axis=3)
conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(up6)
conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv6)

up7 = concatenate([Conv2DTranspose(128, (2, 2), strides=(2, 2),
padding='same')(conv6), conv3], axis=3)
conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(up7)
conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv7)

up8 = concatenate([Conv2DTranspose(64, (2, 2), strides=(2, 2),
padding='same')(conv7), conv2], axis=3)
conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(up8)
conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv8)

up9 = concatenate([Conv2DTranspose(32, (2, 2), strides=(2, 2),
padding='same')(conv8), conv1], axis=3)
conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(up9)
conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv9)

conv10 = Conv2D(1, (1, 1), activation='sigmoid')(conv9)

return Model(inputs=[inputs], outputs=[conv10])
model = unet(input_size=(512,512,1))
model.compile(optimizer=Adam(lr=1e-5), loss=dice_coef_loss,
              metrics=[dice_coef, 'binary_accuracy'])
model.summary()
А қосымшасының жалғасы

from keras.callbacks import ModelCheckpoint, LearningRateScheduler,

```

```

EarlyStopping, ReduceLROnPlateau
weight_path="{ }_weights.best.hdf5".format('cxr_reg')

checkpoint = ModelCheckpoint(weight_path, monitor='val_loss', verbose=1,
                             save_best_only=True, mode='min', save_weights_only = True)

reduceLROnPlat = ReduceLROnPlateau(monitor='val_loss', factor=0.5,
                                   patience=3,
                                   verbose=1, mode='min', epsilon=0.0001, cooldown=2,
min_lr=1e-6)
early = EarlyStopping(monitor="val_loss",
                      mode="min",
                      patience=15) # probably needs to be more patient, but kaggle time is
limited
callbacks_list = [checkpoint, early, reduceLROnPlat]
from IPython.display import clear_output
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split

model.compile(optimizer=Adam(lr=2e-4),
              loss=[dice_coef_loss],
              metrics = [dice_coef, 'binary_accuracy'])

train_vol, validation_vol, train_seg, validation_seg = train_test_split((images-
127.0)/127.0,
                                (mask>127).astype(np.float32),
                                test_size = 0.1,random_state = 2018)

train_vol, test_vol, train_seg, test_seg = train_test_split(train_vol,train_seg,
                                                            test_size = 0.1,
                                                            random_state = 2018)

loss_history = model.fit(x = train_vol,
                        y = train_seg,
                        batch_size = 16,
                        epochs = 20,
                        validation_data =(test_vol,test_seg) ,
                        callbacks=callbacks_list)

А қосымшасының жалғасы

model.save("Lung Segmentation_v1")

```

```

clear_output()
fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (10, 5))
ax1.plot(loss_history.history['loss'], '-', label = 'Loss')
ax1.plot(loss_history.history['val_loss'], '-', label = 'Validation Loss')
ax1.legend()

ax2.plot(100*np.array(loss_history.history['binary_accuracy']), '-',
        label = 'Accuracy')
ax2.plot(100*np.array(loss_history.history['val_binary_accuracy']), '-',
        label = 'Validation Accuracy')
ax2.legend()
pred_candidates = np.random.randint(1, validation_vol.shape[0], 10)
preds = model.predict(validation_vol)

plt.figure(figsize=(20,10))

for i in range(0,9,3):
    plt.subplot(3,3,i+1)

    plt.imshow(np.squeeze(validation_vol[pred_candidates[i]]))
    plt.xlabel("Base Image")

    plt.subplot(3,3,i+2)
    plt.imshow(np.squeeze(validation_seg[pred_candidates[i]]))
    plt.xlabel("Mask")

    plt.subplot(3,3,i+3)
    plt.imshow(np.squeeze(preds[pred_candidates[i]]))
    plt.xlabel("Prediction")

```