МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
НАУКОВО-НАВЧАЛЬНИЙ ІНСТИТУТ
КОМП'ЮТЕРНОЇ ФІЗИКИ та ЕНЕРГЕТИКИ
Кафедра комп'ютерної фізики

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи
за освітньо-кваліфікаційним рівнем магістр

«Використання машинного навчання для визначення кардіо-торакального індексу»

Виконав: студент 2 курсу, групи НФ-64
Конюхов В.Д.
спеціальність:
105 Прикладна фізика та наноматеріали
освітня програма: Комп'ютерна фізика
Керівник:
доктор фізико-математичних наук:
Нємченко К.Е.
Рецензент:

Харків – 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут комп'ютерної фізики та енергетики
Кафедра комп'ютерної фізики
Рівень вищої освіти (освітньо-кваліфікаційний рівень) магістр
Спеціальність: 105 Прикладна фізика та наноматеріали
Освітня програма: Комп'ютерна фізика

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Костянтин НЄМЧЕНКО
підпис

"_____" _____ 2020 року

ЗАВДАННЯ
на дипломну роботу магістра

Конюхов Владислав Дмитрович

1.Тема роботи «Використання машинного навчання для визначення кардіо-торакального індексу»

керівник роботи Нємченко К.Е.

затверджені наказом по університету від 06 грудня 2020 № 0210 – 05/2267
2. Строк подання студентом роботи 12 травня 2020 р

3. Перелік питань, які потрібно розробити:

Огляд та вивчення принципів основних алгоритмів сегментації зображень та використання методів штучного інтелекту та машинного навчання для обробки та розпізнавання зображень. Огляд принципів нейронних мереж та їх порівняння. Вивчення основних алгоритмів сегментації зображень та принципів обробки комп'ютерного зображення. Створення програми для розпізнавання об'єктів на зображеннях. Розробка алгоритму та програми для визначення кардіо-торакального індексу за рентгенографічними зображеннями.

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір та затвердження теми
2	Робота над літературою та іншими джерелами
3	Написання вступу до роботи та першого розділу
4	Написання другого розділу
5	Написання третього розділу
6	Завершення написання дипломної роботи(висновки и література)
7	Подача науковому керівнику, доробка зауважень
8	Відгук, рецензія, підготовка до захисту
9	Попередній захист на кафедрі
10	Захист

5. Дата видачі завдання

Студент _____ Владислав КОНЮХОВ
підпис

Керівник роботи _____ Костянтин НЄМЧЕНКО
підпис

ПЕРЕЛІК СКОРОЧЕНЬ ТА АБРЕВІАТУР

КТІ – кардіо-торакальний індекс;

ШІ – штучний інтеллект

МН – машинне навчання

КЗ – комп’ютерний зір;

КМГ – кардіомегалія;

ШНМ– штучна нейронна мережа;

РНМ – рекурентна нейронна мережа;

ЗНМ – згорткова нейронна мережа;

ДКЧП – довга короткочасна пам’ять;

ССЗ – серцево судинні захворювання;

РЕФЕРАТ

Конюхов Владислав Дмитрович. Використання машинного навчання для визначення кардіо-торакального індексу. Кваліфікаційна робота магістра з прикладної фізики. –Х.: ХНУ імені В. Н. Каразіна, 2020. – 66 с.

Основною метою роботи є визначення кардіо-торакального індексу за графічними зображеннями грудної клітини людини. Для розв’язання цієї задачі буде пройдено головні етапи сегментації зображень за допомогою нейронної мережі. Будуть розглянуті та використані принципи роботи машинного навчання, також будуть розглянуті методи сегментації зображень. Буде створена вибірка зображень, за допомогою якої буде навчено нейронну мережу. В роботі досліджуються нейронні мережі та основні принципи застосування штучного інтелекту для обробки зображень. Досліджуються загальні алгоритми розпізнавання об'єктів. Розроблено алгоритми, які застосовані в програмі, що застосовується для визначення кардіо-торакального індексу на реальних рентгенівських зображеннях.

КЛЮЧОВІ СЛОВА: КОМП’ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, КАРДІО–ТОРАКАЛЬНИЙ ІНДЕКС, НЕЙРОННА МЕРЕЖА

РЕФЕРАТ

Конюхов Владислав Дмитриевич. Использование машинного обучения для определения кардио-торакального индекса. Квалификационная работа магистра прикладной физики. – Х.: ХНУ имени В.Н. Каразина, 2020 –66с.

Основной целью работы является определение кардио-торакального индекса за графическими изображениями грудной клетки человека. Для решения этой задачи будет пройден главные этапы сегментации изображений с помощью нейронной сети. Будут рассмотрены и использованы принципы работы машинного обучения, также будут рассмотрены методы сегментации изображений. Будет создана выборка изображений, с помощью которой будет обучено нейронную сеть. В работе исследуются нейронные сети и основные принципы применения искусственного интеллекта для обработки изображений. Исследуются общие алгоритмы распознавания объектов. Разработаны алгоритмы, применённые в оригинальной программе, которая используется для определения кардио-торакального индекса на реальных рентгеновских изображениях.

КЛЮЧЕВЫЕ СЛОВА: КОМПЬЮТЕРНОЕ ЗРЕНИЕ, МАШИННОЕ ОБУЧЕНИЕ, СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ .КАРДИО–ТОРАКАЛЬНИЙ ИНДЕКС, НЕЙРОННЫЕ СЕТИ.

ABSTRACT

Koniukhov Vladyslav. Using machine learning to determine cardiothoracic index. Master's qualification work in applied physics. -Kh .: V. N. Karazin KhNU, 2020 –66p.

The main goal of the work is to determine the cardio-thoracic index behind graphic images of the human chest. To solve this problem, we will go through the main stages of image segmentation using a neural network. The principles of machine learning will be reviewed and used, and image segmentation techniques will also be considered. A sample of images will be created, with the help of which a neural network will be trained. The work explores neural networks and the basic principles of the use of artificial intelligence for image processing. The general algorithms for object recognition are investigated. Algorithms are developed that are used in the original program, which is used to determine the cardio-thoracic index in real x-ray images.

KEYWORDS: COMPUTER VISION, MACHINE LEARNING, SEGMENTATION OF IMAGES. CARDIO-TORACAL INDEX, NEURAL NETWORKS.

ЗМІСТ

Вступ…….…..9
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ……...12
1.1 Загальна інформація про КТІ…….……...…..12
1.2 Вступ в теорію машинного навчання…….…...…….....16
1.3 Вступ в теорію комп’ютерного зору……..28
РОЗДІЛ 2. ТЕОРЕТИЧНИЙ АНАЛІЗ ІСНУЮЧИХ ПРОБЛЕМ……30
1.4 Огляд нейронних мереж……..30
1.5 Огляд алгоритмів сегментації…….....40
1.6 Аугментація зображень…….....46
РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІВ ТА ЇХ
ПРАКТИЧНЕ ЗАСТОСУВАННЯ ……..50
1.7 Постановка задачі в медицині…….……...50
1.8 Створення навчальної вибірки…….…..…..53
1.9 Навчання нейронної мережі……..... 58
1.10 Визначення КТІ……...63
1.11 Створення програмного забезпечення……66
ВИСНОВКИ …….……71
СПИСОК ЛІТЕРАТУРИ ……..…….……...72

ВСТУП

У наш час машинне навчання та комп’ютерний зір знаходять своє застосування у багатьох сферах діяльності сучасної людини. В цій роботі буде наглядно продемонстровано створення алгоритму визначення кардіо-торакального індекса за допомогою машинного навчання та комп’ютерного зору.

Сучасну медицину дуже важко уявити без застосування персональних комп'ютерів. Комп’ютери мають широке поширення у багатьох гілках медицини. Розпочинаючи з CPOE (computerized physician order entry) – комп’ютеризована система приписів для лікаря, закінчуючи роботами-інтернами, які допомагають хірургам у проведенні операції. Також велике значення комп’ютери мають при роботі клініки в цілому, допомагаючи планувати та виконувати різні адміністративні завдання, відслідковувати фінанси та інше.

Рентгенівські знімки відіграють велику роль в діагностиці та лікуванні пацієнтів. Завдяки цьому лікарі можуть діагностувати велику кількість захворювань та патологій. В еру комп’ютерних технологій ми дуже легко можемо автоматизувати медичні процеси розпізнавання медичних зображень. Ми можемо знаходити відхилення від норми без втручання лікаря, тим самим прискорюючи виконання роботи на унеможливлюючи людський фактор в прийманні рішення.

Чому саме так важливо визначати кардіо-торакальний індекс? Серцево-судинні захворювання (ССЗ) є основною причиною смерті в усьому світі: ні з якоїсь іншої причини щорічно не вмирає стільки людей, скільки від ССЗ. За оцінками, в 2016 році від ССЗ померло 17,9 мільйона чоловік, що склало 31% всіх випадків смерті в світі. 85% цих смертей сталося в результаті серцевого нападу та інсульту[1].

Ми можемо побачити що кількість смертей від серцевих захворювання є дуже великою, тому дуже важливо вчасно та точно реагувати на зміни розмірів та форми серця. А за допомогою машинного навчання ми можемо легко збільшувати базу рентгенівських зображень та покращити точність сегментації зображень.

Актуальність теми. Автоматична діагностика хвороб і патологій по рентгенівським зображенням грудної клітини придбала важливе значення під час пандемії вірусу COVID-19. У численних дослідницьких центрах почали створюватися нові системи розпізнавання ознак легеневих захворювань, зокрема пневмонії, з використанням методів штучного інтелекту. У цих дослідженнях на першому етапі завжди ставиться завдання визначення контурів основних об'єктів на знімках грудної клітини – легких, серця тощо. Розроблені в цій дипломній роботі оригінальні алгоритми розпізнавання об'єктів на рентгенівських зображеннях грудної клітини дозволяють не тільки вирішувати головне завдання, поставлене в дипломі - визначення кардіо-торакального індексу, а й забезпечувати реалізацію першого етапу виконання завдання з діагностування легеневих захворювань.

Для досягнення поставленої **мети** роботи – визначення кардіо-торакального індексу за графічними зображеннями грудної клітини людини – було визначено такі **задачі**:

– розробити алгоритм сегментації зображень за допомогою нейронної мережі;

– розглянути та використати принципи роботи машинного навчання;

– створити вибірку зображень, за допомогою якої буде навчено нейронну мережу;

– розробити алгоритми, що застосовуються для визначення кардіо-торакального індексу на реальних рентгенівських зображеннях.

Об’єкт дослідження – машинне навчання та нейронні мережі в задачах розпізнавання зображень.

Предмет дослідження – нові методи визначення контурів об’єктів на рентгенівських зображеннях грудної клітини з метою визначення кардіо-торакального індексу для профілактики певних захворювань.

Методи дослідження. Теоретичні положення роботи базуються на математичних методах науки про данні, методів використання нейронних мереж, з використанням алгоритмів манного навчання та штучного інтелекту.

Наукова новизна одержаних результатів.

– розроблено новий алгоритм визначення форми та розмірів легенів та серця на зображеннях грудної клітини для визначення кардіо-торакального індексу із застосуванням машинного навчання.

– розроблено та використано програмний продукт для реалізації даного алгоритму.

Практичне значення отриманих результатів:

- обґрунтовано принципову можливість визначення кардіо-торакального індексу з високою ступеню точності.
- значно підвищено ефективність методу, зокрема, в порівнянні с існуючими методиками, що використовуються в медичних цілях.

Структура і обсяг роботи. Магістерська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 41 сторінку, із них основного тексту 34 сторінки, 16 рисунків за текстом, список використаних джерел із 29 найменувань на 3 сторінках.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Загальна інформація про КТІ.

Кардіо-торакальний індекс дорівнює відношенню поперечного розміру серця (ПР+ЛР) за винятком жирової подушки верхівки серця до внутрішнього розміру грудної клітки (ВРГК): КТ індекс = (ПР+ЛР): ВРГК, де ПР — відстань від серединної лінії до правої границі силуету серця, а ЛР — відстань від серединної лінії до лівої границі силуету серця.

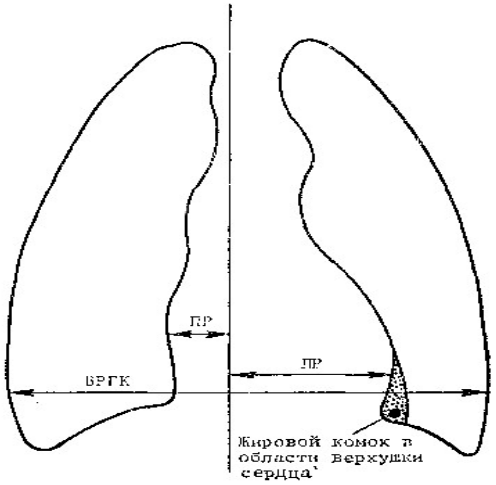


Рис 1.1 Визначення кардіоторакального індексу.

Кардіомегалія – це значне збільшення розмірів серця за рахунок його гіпертрофії та дилатації або нагромадження продуктів порушеного обміну речовин, або розвитку неопластических процесів. Гіпертрофія міокарда є компенсаторною реакцією, що дозволяє серцю підтримувати нормальний кровообіг при наявності того чи іншого патологічного стану. Гіпертрофія ніколи не приводить до значного збільшення розмірів серця й супроводжується лише помірним розширенням його меж[2]. КМГ виникає, як правило, при розвитку міогенної дилатації серця й характеризується різними симптомами серцевої недостатності та порушенням ритму. В залежності від причин, котрі викликають збільшення розмірів серця, маємо спочатку

можливий розвиток парціальної КМГ (значне збільшення окремої серцевої камери), внаслідок чого розвивається тотальна КМГ. Дифузні ураження міокарда відразу приводять до тотальної КМГ. Найчастіше ступінь КМГ залежить від тривалості патологічного процесу, що викликає збільшення розмірів серця, і його виразності.

Деякі фактори можуть виявляти вплив на оцінку розмірів серця при рентгенологічному дослідженні. У дітей з худою конституцією та астенічної конституції з подовженою грудною кліткою внаслідок низького стояння діафрагми й вертикального розташування серця можливість діагностики КМГ менша. І навпаки, при гіпертонічній конституції високе стояння діафрагми та горизонтальне положення серця можуть приводити до вигаданої кардіомегалії. Подібним чином при неадекватному вдиху під час рентгенографії або підвищенні рівня стояння діафрагми, наприклад, при збільшенні живота, серце розташовується горизонтальніше, у результаті чого його розміри видадуться збільшеними. З іншого боку, кардіомегалія може бути не видна при перероздутій грудній клітці (хронічні захворювання легенів). На інтерпретацію розмірів серця також впливає положення пацієнта під час дослідження. Якщо рентгенографія в передній прямій проекції виконана правильно, рукоятка грудини на знімку розташована по центру тіл грудних хребців. Навіть при незначному повороті тіла хворого конфігурація тіні серця і його розміри змінюються. Накладення тіні перикардіальної або апікальної жирової подушки може привести до переоцінки розмірів серця. Серце може також зміщатися в результаті ателектазу, пневмотораксу й евентрації органів черевної порожнини через грижовий отвір у діафрагмі, які приводять до зміни конфігурації й видимих розмірів серця. Нарешті, до збільшення розмірів серця можуть приводити зміни техніки виконання рентгенографії (неприпустимі відстані, використання портативної апаратури, знімок у задній проекції). Найкращим показником наявної кардіомегалії є серійні рентгенограми,

виконані за стандартною методикою, на яких в динаміці відзначається прогресуюче збільшення серця.

Найкраще при рентгенологічному дослідженні виявляються ознаки кардіомегалії в результаті збільшення порожнин серця. Значна гіпертрофія шлуночків може ховатись за тінню серця, що має на рентгенограмі нормальні розміри. По стандартній рентгенограмі, виконаній в передній прямій і бічній проекціях, буває важко судити про збільшення окремих камер серця. У нормі на передній прямій рентгенограмі серця границю правих відділів серця утворюють верхня порожниста вена, праве передсердя й іноді нижня порожниста вена, а ліву границю - дута аорти, основний стовбур і ліва гілка легеневої артерії, ліве передсердя й лівий шлуночок. Однак при збільшенні серця взаєморозташування цих структур може зазнати різних змін. Гадане розширення лівої границі серця насправді може бути викликане збільшенням правих відділів, що приводять до зсуву лівих відділів серця назад і навпаки.

Традиційна рентгенографія органів грудної клітки з контрастуванням стравоходу барієм може дати більш точну інформацію про збільшення окремих камер серця, чим стандартна рентгеноскопія, але це дослідження використовується менш широко, оскільки ехокардіографія дає більш точну інформацію про наявність гіпертрофії міокарда, порушеннях руху стінки шлуночка й наявності рідини в порожнині перикарда.

За даними середня величина КТІ максимальна у новонароджених (близько 57%) і поступово знижується до 45% до віку 6 років. Вище 6-річного віку середня величина КТІ не перевищує 50%. Більшість рентгенологів вважають норму верхньої границі КТІ величину 50%. В той же час в ході обстеження здорових чоловіків європейської, азійської та афро-американської раси встановлено, що КТІ перевищує 50% у європейців в 2% випадків, у азіатов - в 4% випадків, а у афро-американців в 9% випадків. З цих даних слідують два висновки:

1. Для вибірки здорових людей існує ймовірність перевищення КТІ кордону 50% .

2. Частка здорових людей з КТИ вище 50% різна для людей різних національностей[3].

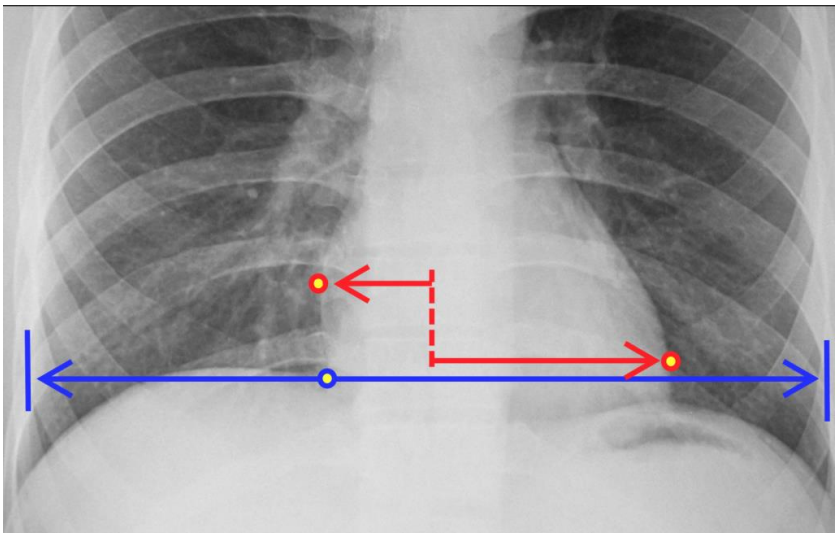


Рис 1.2 Визначення кардіо-торакального індексу на рентген знімку.

1.2. Вступ в теорію машинного навчання.

Перш ніж починати розгляд машинного навчання, спочатку потрібно розібратися з тим що таке штучна нейронна мережа.

Штучна нейронна мережа - це мережа простих елементів, званих *нейронами*, які отримують вхід, змінюють свій внутрішній стан (збудження) відповідно до цього входу, і виробляють вихід, залежний від входу та збудження. Мережа утворюється з'єднанням виходів певних нейронів зі входами інших нейронів з утворенням орієнтованого зваженого графу. Ваги, як і функції, що обчислюють збудження, можуть змінюватися процесом, званим навчанням, який керується правилом навчання. Ця система навчається для виконання задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу. Наприклад, у процесі розпізнаванні зображень мережі можуть навчатися ідентифікувати зображення, які містять котів, аналізуючи приклади зображень, мічені як «кіт» і «не кіт», та використовуючи отримані результати для ідентифікування котів в інших зображеннях. Вони роблять це без жодного апріорного знання про котів, наприклад, що вони мають хутро, хвости, вуса. Натомість, вони розвивають свій власний набір доречних характеристик з навчального матеріалу, який вони можуть обробити.

ШНМ ґрунтується на сукупності з'єднаних вузлів, що називають штучними нейронами (це аналогія до біологічних нейронів у головному мозку тварин). Кожне з'єднання (аналогічне синапсові) між штучними нейронами може передавати сигнал від одного до іншого. Штучний нейрон, що отримує сигнал, може обробляти його, й потім сигналізувати штучним нейронам, приєднаним до нього.

Зазвичай в більш поширених реалізаціях ШНМ сигнал на з'єднанні між штучними нейронами є дійсним числом, а вихід кожного штучного нейрону обчислюється нелінійною функцією суми його входів. Штучні нейрони та з'єднання зазвичай мають вагу, яка підлаштовується в перебігу навчання. Вага

збільшує або зменшує силу сигналу на з'єднанні. Штучні нейрони можуть мати такий поріг, що сигнал надсилається лише якщо сукупний сигнал перетинає цей поріг. Штучні нейрони зазвичай організовано в шари. Різні шари можуть виконувати різні види перетворень своїх входів. Сигнали проходять від першого (входового) до останнього (виходового) шару, можливо, після проходження шарами декілька разів.

Основною метою підходу ШНМ було вирішення задач таким же способом, як це робив би людський мозок. З часом увага зосередилася на відповідності певним розумовим здібностям, ведучи до відхилень від біології. ШНМ використовували в ряді різноманітних задач, включно з комп'ютерним баченням, мовленнєве розпізнавання, машинним перекладом, соціально-мережевим фільтруванням, грою в настільні та відеоігри, медичним діагностуванням[4].

1.2.1. Складові штучної нейронної мережі

Нейрони

Нейрон з міткою j , що отримує вхід $p_j(t)$ від нейронів-попередників, складається з наступних складових:

- збудження
- можливо, порогу
- функції збудження
- та функції виходу

Нейрон входу не може мати попередників, а слугує інтерфейсом входу для всієї нейронної мережі. Аналогічно, нейрон не може мати наступників, і відтак слугує інтерфейсом виходу для всієї мережі.

З'єднання та ваги

Мережа складається зі з'єднань, кожне з яких передає вихід нейрону і до входу нейрону j. В цьому сенсі і є попередником j , а j є наступником і. Кожному з'єднанню призначено вагу w_{ij}.

Правило навчання

Правило навчання - це алгоритм або правило, який змінює параметри нейронної мережі, щоб заданий вхід до мережі видавав придатний вихід. Цей процес навчання зазвичай полягає в зміні ваг та порогів змінних мережі[5].

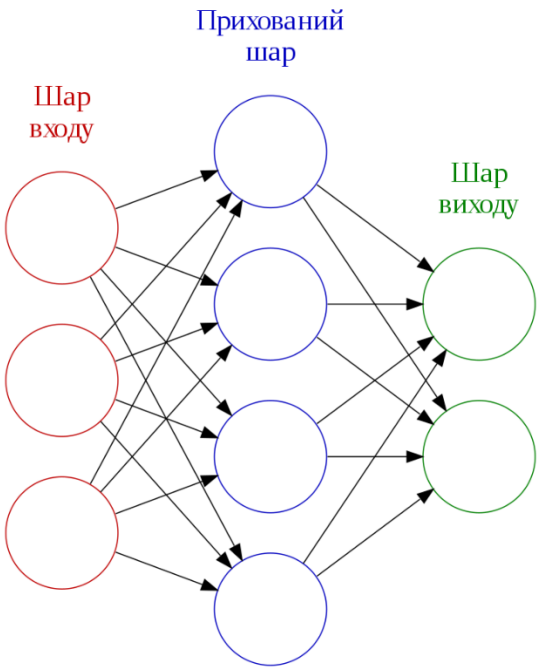


Рис 1.3 Штучна нейронна мережа.

1.2.2.Варіанти

Метод групового урахування аргументів

Цей метод групового урахування аргументів може демонструвати повністю автоматичну структурну та параметричну оптимізацію моделей. Поліноми Колмогорова - Габора є функціями збудження вузлів, що дозволяють множення та додавання. Використовування глибинний

багатошаровий перцептрон прямого поширення з вісьмома шарами. Він є мережею керованого навчання, що росте шар за шаром, де кожен з шарів тренується згідно з регресійним аналізом. Елементи котрі являється непотрібними виявляються застосуванням затверджувального набору та обрізаються шляхом регуляризації. Розмір та глибина отримуваної в результаті мережі залежить від задачі.

Згорткові нейронні мережі

Згорткова нейронна мережа - це клас глибинних мереж прямого поширення, які складаються з одного чи більше згорткових шарів, із повноз'єднаними шарами (які в свою чергу відповідають шарам звичайних ШНМ) на верхівці. Він використовує зв'язані ваги та шари агрегування. Зокрема, за згортковою архітектурою Фукусіми часто зорганізовують максимізаційне агрегування. Ця архітектура дозволяє згортковим нейронним мережам отримувати користь від двовимірної структури вхідних даних.

ЗНМ є зручними для обробки візуальних та інших двовимірних даних. Вони показують чудові результати при застосуваннях як для зображень, так і для мовлення. Їх може бути треновано стандартним зворотним поширенням. ЗНМ є простішими для тренування за інші звичайні глибинні нейронні мережі прямого поширення, та мають набагато менше параметрів, які треба оцінювати.

Довга короткочасна пам'ять

Мережі довгої короткочасної пам'яті - це мережі, які уникають проблеми зникання градієнту. Довга короткочасна пам'ять зазвичай доповнювана рекурентними вентилями, які називають забувальними. Мережі ДКЧП запобігають зниканню та вибуханню зворотно поширюваних похибок. Натомість, отриманні похибки також можуть плинути в зворотному напрямку необмеженим числом віртуальних шарів розгорнутої в просторі. Таким чином, ДКЧП може вчитися для виконання задач „дуже глибокого навчання“, що потребують спогадів про події, які сталися тисячі або навіть мільйони

дискретних кроків часу тому. Можливо виводити проблемно-орієнтовані подібні архітектури. Довга короткочасна пам'ять може мати справу з сигналами та тривалими затримками, які містять суміш низько - та високочастотних складових.

Глибинне резервуарне обчислення

Глибинне резервуарне обчислення та глибинні мережі з відлунням стану можуть забезпечувати систему для ефективного тренування моделей для використання в ієрархічній обробці часових даних, в той же час уможливлюючи дослідження властивої ролі шаруватого компонування.

Глибинні мережі переконань

Глибинна мережа переконань - це ймовірнісна породжувальна модель, яка складається з декількох шарів прихованих вузлів. Її також можна розглядати як композицію простих модулів навчання, що складають кожен з шарів.

Нейронні мережі зберігання та вибірки великої пам'яті

Нейронні мережі зберігання та вибірки великої пам'яті є швидкими нейронними мережами глибинного навчання з n кількістю шарів, які в свою чергу можуть використовувати дуже багато фільтрів одночасно. Ці фільтри можуть бути стохастичними , нелінійними, , логічними, не стаціонарними та навіть не аналітичними. Вони є біологічно натхненними, і навчаються безперервно.

Глибинні складальні мережі

Глибинна складальна мережа ґрунтується на ієрархії блоків спрощених нейромережевих модулів. Її було представлено 2011 року Доном та Деном. Вона формулює навчання як задачу опуклої оптимізації з розв'язком замкненого вигляду, підкреслюючи важливість подібності цього механізму до складеного узагальнення. Кожен блок глибинної складальної мережі є простим модулем, який легко тренувати сам по собі керованим чином без зворотного поширення для всіх блоків.

Тензорні глибинні складальні мережі

Ця архітектура є розширенням глибинних складальних мереж . Вона пропонує два важливі поліпшення: вона може використовувати інформацію вищого порядку отриманих коваріаційних статистик, і перетворює неопуклу задачу нижчого рівня на опуклу підзадачу вищого рівня. Таким чином тензорні глибинні складальні мережі використовують коваріаційні статистики у білінійному відображенні з кожного з двох окремих наборів прихованих вузлів одного й того ж рівня на передбачення, через отриманний тензор третього порядку.

Піково-пластинові обмежені машини Больцмана

Потреба в глибинному навчанні з дійснозначними входами, як у ґаусових обмежених машинах Больцмана, привела до *піково-пластинових* обмежених машин Больцмана, які моделюють безперервнозначні входи строго двійковими латентними змінними. Подібно до базових обмеженних машин Больцмана та її варіантів, піково-пластинова обмеженні машини Больцмана є двочастковим графом, але видимі вузли (входи) є дійснозначними. Відмінність є в прихованому шарі, де кожен прихований вузол має змінну двійкового та змінну дійснозначної пластини. Пік є дискретною масою ймовірності на нулі, тоді як пластина є густиною ймовірності над безперервною областю визначення, їхня суміш формує апріорне.

Розширення обмеженних машин Больцмана, що називається μ-обмеженних машин Больцмана, забезпечує додаткові моделювальні потужності, використовуючи додаткові члени в енергетичній функції. Один із цих членів дає моделі можливість формувати умовний розподіл пікових змінних знеособленням пластинових змінних за заданого спостереження.

Мережі з окремими структурами пам'яті

Поєднання зовнішньої пам'яті з шаровою нейронною мережею бере свій початок у проведенних ранніх дослідженнях котрі на розподілених представлень та самоорганізаційних відображень Кохонена. Наприклад, у

розрідженій розподіленій пам'яті та ієрархічній часовій пам'яті зразки, закодовані нейронними мережами, можуть використовуватись як адреси для асоціативної пам'яті, з «нейронами», які по суті слугують дешифраторами та шифраторами адреси. Проте, ранні контролери таких типів пам'яті не були диференційовними.

Нейронні машини Тюрінга

Нейронні машини Тюрінга зв’язують мережі із зовнішніми ресурсами пам'яті, з якими вони також можуть взаємодіяти за допомогою процесів уваги. Ця подібна та зв'язана система є аналогічною до машини Тюрінга, але є диференційованою з краю в край, що може в свій час дозволяти їй дієво навчатися за градієнтним спуском. Попередньо отриманні результати показують, що нейронні машини Тюрінга можуть виводити з прикладів входу та виходу прості алгоритми, такі як копіювання, впорядкування та асоціативне пригадування.

Семантичне гешування

Підходи, які представляють попередній безпосередній досвід, і також використовують схожий для них досвід для формування локальної моделі, часто називають методами найближчого сусіда або *k*-найближчих сусідів. В семантичному гешуванні є корисним глибинне навчання, де з великого набору документів отримується глубоко глибинна графова модель векторів уточнюючих кількостей слів. Документи відображуються на комірки пам'яті таким чином, що семантично схожі документи розташовуються за близькими адресами. Потім ці документи, схожі на документ запиту, є можливість знаходити шляхом простого доступу до всіх адрес, котрі в свою чергу відрізняються від адреси документа запиту лише кількома бітами. На відміну від розрідженої розподіленої пам'яті, що оперує 1000-бітними адресами, семантичне гешування працює на 32- або 64-бітних адресах, що зустрічаються в традиційній комп'ютерній архітектурі.

Мережі з пам'яттю

Мережі з пам'яттю є іншим розширенням нейронних мереж, що обов’язково включають довготривалу пам'ять. Довготривала пам'ять може записуватися або читатися з наступною метою використання її для передбачення. Ці моделі застосовувалися в контексті питально-відповідальних систем, де довготривала пам'ять ефективно діє як (динамічна) база знань, а вихід є текстовою відповіддю.

Вказівникові мережі

Глибинні мережі можуть бути потенційно поліпшено поглибленням та скороченням параметрів, за збереження здатності до навчання. В той час як тренування надзвичайно глибоких нейронних мереж може бути непрактичним, в свою чергу ЦП-подібні архітектури, такі як вказівникові мережі та нейронні машини які отримують довільний доступ , долають це обмеження в першу чергу завдяки застосуванню зовнішньої пам'яті з довільним доступом та інших складових, котрі зазвичай належать до комп'ютерної архітектури, таких як регістри, вказівники. Такі системи працюють на векторах розподілів імовірностей, що зберігаються в комірках пам'яті та регістрах. Таким чином, ця модель є повністю диференційованою, й тренується з краю в край. Ключовою характеристикою цих моделей є те, що їхня глибина, розмір їхньої короткочасної пам'яті та число параметрів можливо змінювати незалежно.

Кодувально-декодувальні мережі

Кодувально-декодувальні системи ґрунтуються на нейронних мережах, що відображують високоструктурований вхід на високоструктурований вихід. Цей підхід міг виникти у контексті машинного перекладу, де вхід та вихід є писаними реченнями двома природними мовами. В тій праці РНМ або ЗНМ ДКЧП використовувалася як кодувальник для отримання зведення про вхідне речення, і це зведення декодувалося умовною РНМ-моделлю мови для

продукування перекладу. Для цих систем є спільними будівельні блоки: вентильні РНМ та ЗНМ, і треновані механізми уваги.

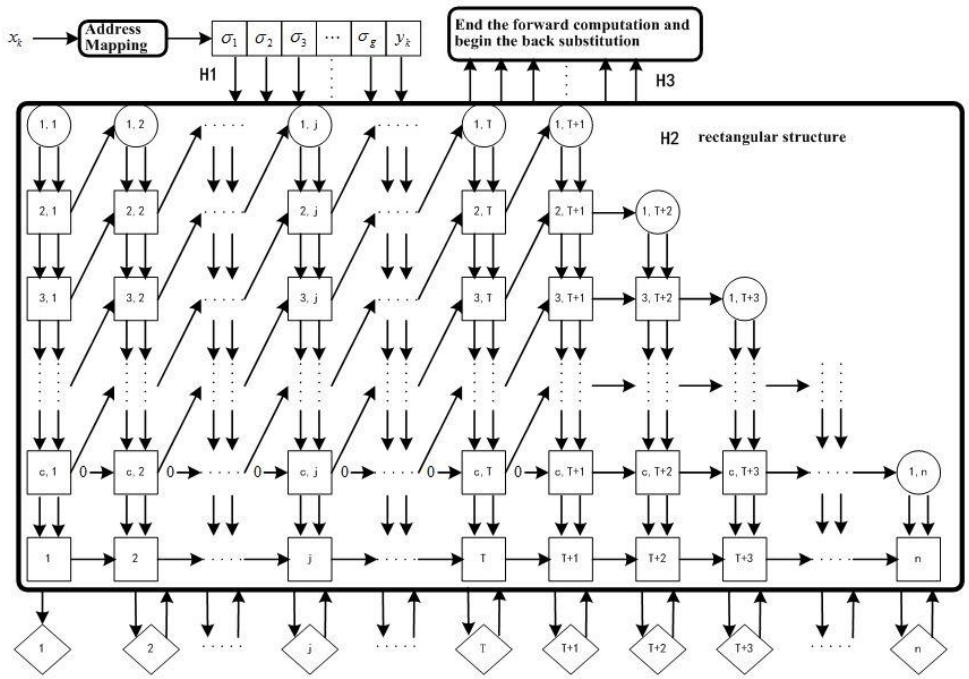


Рис 1.4 Паралельно-конвеєрна структура нейронної мережі АКММ.

Машинне навчання – це великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися. Машинне навчання активно застосовують як в режимі офлайн, так і в режимі онлайн . У машинному навчанні з учителем програміст, котрий навчає цю систему, може розмічати дані, та приводити машині певні приклади та спостерігає за її прогресом. Завданнями, які вирішуються за допомогою навчання з учителем є, наприклад, класифікація і регресія. Машинне навчання без учителя включає в себе наступні типи: кластеризація**,** узагальнення**,** пошук правил[6].

1.2.3. Методи машинного навчання

У загальному випадку ми можемо розрізняти два типи машинного навчання: навчання по прецедентах, або індуктивне навчання, дедуктивне навчання. Оскільки дедуктивне навчання прийнято відносити до області експертних систем, то терміни «машинне навчання» і «навчання по прецедентах» можна вважати синонімами. Вказаний метод навчання, як зараз прийнято говорити, дуже популярний, а ось експертні системи наразі переживають кризу. В їх основі лежать бази знань, котрі важко узгоджувати з реляційною моделлю даних, тому промислові СУБД не мають можливості ефективно використовуватися для наповнення баз знань експертних систем.

В свою чергу, навчання по прецедентах, , поділяють на три основних типи: контрольоване навчання(або навчання з учителем), неконтрольоване навчання(або навчання без учителя), і навчання з підкріпленням[7].

Окрім названих, також мають місце і інші методи навчання: активне, багатозадачне, різноманітне, трансферне і інші. Особливо успішно розвивається в останні роки «глибоке навчання», при використанні якого можуть успішно поєднуватися алгоритми навчання з вчителем і без вчителя.

Контрольоване навчання

Метод контрольного навчання застосовується в тих випадках, коли є дуже великі обсяги даних, на приклад - тисячі фотографій з маркерами (мітками, ярликами): це ягуар, а це кіт. Потрібно створити деякий алгоритм, за допомогою котрого машина могла б по фотографії, яку вона ще «не бачила» раніше, визначити, хто на ній зображений: ягуар або кіт. У ролі «вчителя» в даному випадку виступає людина, яка заздалегідь повинна була поставити маркери. Машина сама вибирає ознаки, за якими вона відрізняє ягуарів від котів. Тому в подальшому знайдений нею алгоритм може бути швидко переналаштований на рішення іншої задачі, наприклад, на розпізнавання курей і качок. Машина сама виконає складну і кропітку роботу яка включає в себе виділення ознак, за якими буде розрізняти цих птахів. А нейромережа,

яку навчили розпізнавати кішок, можна швидко навчити обробляти результати комп'ютерної томографії.

Неконтрольоване навчання

На маркованих, розмічених даних накопичилося вже досить багато, даних без маркерів все ж набагато більше. Це зображення без підписів, тексти без анотацій, аудіозаписи без коментарів,. Машина виконує завдання при неконтрольованому навчанні - знайти зв'язку між окремими даними, виявити закономірності, порівняти шаблони, можливо упорядкувати дані або описати їх структуру, а також виконати класифікацію даних.

Навчання з підкріпленням

Таке навчання може бути окремим випадком контрольованого навчання, але при цьому вчителем в даному випадку виступає «середовище». Машина (її в цій ситуації часто називають «агент») не має попередньої інформацією про середовище, але має можливість здійснювати в ній будь-які дії. Середовище реагує на дії машини і тим самим надає агенту дані, які дозволяють йому реагувати на них і вчитися. Фактично агент і середовище утворюють систему зі зворотним зв'язком.

Підкріплене навчання фактично використовується для вирішення більш складних завдань, ніж навчання без вчителя та з вчителем. Воно використовується, наприклад, в системах навігації для роботів, які навчаються уникати зіткнень з перешкодами шляхом набуття досвіду, отримуючи зворотний зв'язок при кожному зіткненні.

Застосування машинного навчання

Класифікація завдань машинного навчання виникає при розгляді бажаного виходу системи з машинним навчанням:

- В даній класифікації входи поділяються на декілька або більше класів, і система-учень мусить породити модель, яка відносить небачені входи до одного або більше з цих класів. Це, як правило, намагаються розв'язувати керованим чином. Прикладом цієї

класифікації є фільтри спаму, в котрих входами є повідомлення електронної пошти , а класами є «спам» та «не спам».

- У регресії, також керованій задачі, виходи є безперервними, а не дискретними.
- У кластеруванні набір входів повинен бути поділен на групи. На відміну від класифікації, групи не відомі заздалегідь, що зазвичай робить це завданням для спонтанного навчання.
- Оцінка густини знаходить розподіл входів у деякому просторі.
- Значне зниження розмірності може спрощувати входи шляхом відображення їх на простір меншої розмірності. Пов'язаною задачею є тематичне моделювання, в якому програмі надають перелік документів людською мовою, і ставлять задачу з'ясувати, які документи охоплюють подібні теми.

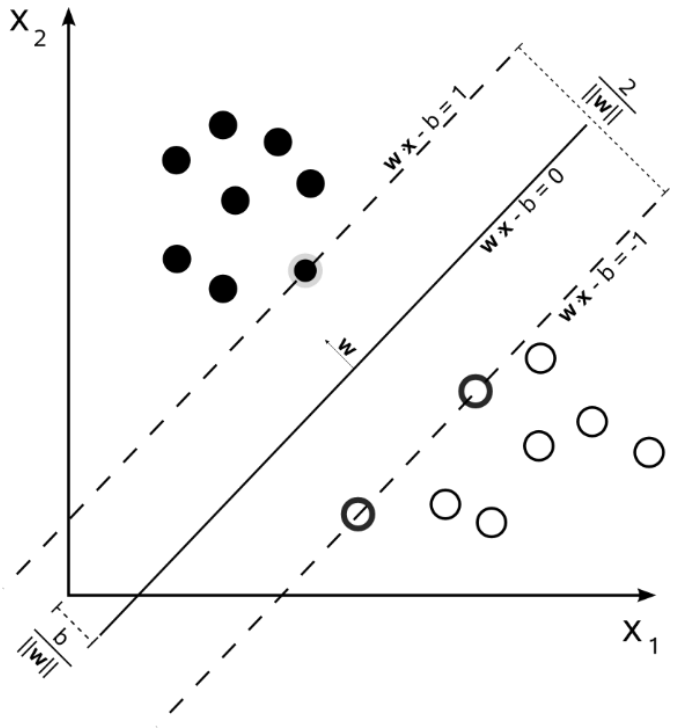


Рис 1.5 Метод опорних векторів є класифікатором, який поділяє свій вхідний простір на дві області, розділені лінійною межею.

1.3. Вступ в теорію комп’ютерного зору

Комп'ютерний зір - теорія та технологія створення машин, які можуть проводити виявлення, стеження та визначення об'єктів.

Як наукова дисципліна, комп'ютерний зір належить до теорії та технології створення штучних систем, які отримують інформацію у вигляді зображень.

Комп'ютерний зір також може бути описаний як доповнення к біологічному зору. У предметі біології вивчається зорове сприйняття людини і різноманітних тварин, в підсумку створюються моделі роботи таких систем в термінах фізіологічних процесів. Комп'ютерний зір, з іншого боку, вивчає і описує системи комп'ютерного зору, які виконано апаратно або програмно.

Типові завдання комп'ютерного зору

Розпізнавання

Класичне завдання в комп'ютерному зорі, обробці зображень - це визначення того, чи містить зображення деякий характерний об'єкт, особливість чи активність. Це завдання може бути легко вирішено людиною, але досі не вирішено задовільно в комп'ютерному зорі в загальному випадку: випадкові об'єкти у випадкових ситуаціях.

Наявні способи вирішення цього завдання придатні тільки для окремих об'єктів, таких як прості геометричні об'єкти, людські обличчя, друковані чи рукописні символи, автомобілі і лише у визначених умовах, зазвичай це певне освітлення, тло і положення об'єкта відносно камери.

Загалом існує різноманіття проблем розпізнавання:

Розпізнавання: один чи декілька заздалегідь заданих чи вивчених класів об'єктів або об'єктів можуть бути розпізнані, зазвичай разом з їх двовимірним положенням на зображенні чи тривимірним положенням в сцені.

Ідентифікація: розпізнається унікальний екземпляр об'єкта. Приклади: ідентифікація визначеного людського обличчя або відбитка пальців чи автомобільного номера.

Виявлення: зображення перевіряються на наявність визначеної умови. Наприклад, виявлення можливих неправильних або патологічних тканин чи клітин в медичних зображеннях. Виявлення, що ґрунтується на відносно простих і швидких обчисленнях, іноді використовується для знаходження невеликих ділянок в зображенні, що в свою чергу аналізується, а потім досліджуються за допомогою заходів, які потребують більше ресурсів, для отримання правильної інтерпретації.

Також існує кілька спеціалізованих завдань, що базуються на розпізнаванні, наприклад:

Пошук деяких зображень за вмістом: знаходження всіх зображень або деяких серед великого набору зображень. Вміст може бути визначено різними шляхами, наприклад в термінах схожості з певним зображенням або в термінах високорівневих критеріїв пошуку, котрі вводяться як текстові дані.

Оцінка положення: визначається положення чи орієнтація визначеного об'єкта відносно фото-камери чи відео-камери. Прикладом застосування цієї техніки може бути спрямування руки робота при вилученні об'єктів з конвеєра на лінії складання.

Оптичне розпізнавання символів: таке розпізнавання символів на зображеннях друкованого чи рукописного тексту, насамперед для перекладу в текстовий формат, найбільш зручний для редагування чи індексації[8].

РОЗДІЛ 2. ТЕОРЕТИЧНИЙ АНАЛІЗ ІСНУЮЧИХ ПРОБЛЕМ

2.1. Огляд нейронних мереж

В наш час існує велика кількість зарекомендувавших себе нейронних мереж. В цьому розділі будуть розглянуті найбільш популярні нейронні мережі для взаємодії з зображеннями.

ResNet: Це модель, запропонована Microsoft, яка отримала 96,4% точності в конкурсі ImageNet 2016. ResNet використовується як попередньо підготовлена модель для декількох застосувань. ResNet має велику кількість шарів разом із залишковими з'єднаннями, що робить його навчання можливим. Коли глибша мережу починає згортатися, виникає проблема: зі збільшенням глибини мережі точність спочатку збільшується, а потім швидко погіршується. Зниження точності навчання показує, що не всі мережі легко оптимізувати[9].

Щоб подолати цю проблему, компанія Microsoft ввела глибоку «залишкову» структуру навчання. Замість того, щоб сподіватися на те, що кожні кілька укладених шарів безпосередньо відповідають бажаному основному поданням, вони явно дозволяють цим верствам відповідати «залишковим». Формулювання F (x) + x може бути реалізована за допомогою нейронних мереж з сполуками для швидкого доступу.

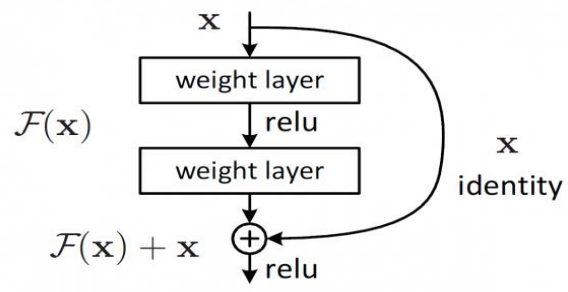


Рис 2.1 ResNet F (x) + x .

З'єднання швидкого доступу пропускають один або кілька шарів і виконують зіставлення унікальних ідентифікаторів. Їх виходи додаються до

виходів укладених шарів. Використовуючи ResNet, можна вирішити безліч проблем наприклад, в першу чергу для вирішення проблем таких як:

ResNet відносно легко оптимізувати: «прості» мережі (які просто складають шари) показують велику помилку навчання, коли глибина збільшується.

ResNet дозволяє відносно легко збільшити точність завдяки збільшенню глибини, чого з іншими мережами домогтися складніше.

Проста мережа: Прості базові лінії в основному натхненні філософією мереж VGG. Згорткові шари в основному мають фільтри 3 × 3 і слідують двох простих правил:

- Для однієї і тієї ж вихідний карти об'єктів шари мають однакову кількість фільтрів;
- Якщо розмір карти об'єктів зменшується вдвічі, число фільтрів подвоюється, щоб зберегти тимчасову складність кожного шару.

Насамперед варто відзначити, що модель ResNet має менше фільтрів і складність менше, ніж мережі VGG.

ResNet: на основі описаної вище простої мережі додано швидке з'єднання, яке перетворює мережу в її залишкову версію. Ідентифікаційні швидкі з'єднання F (x {W} + x) можуть використовуватися безпосередньо, коли вхід і вихід мають однакові розмірності. Коли розмірності збільшуються, він розглядає два варіанти:

- Швидке з'єднання виконує зіставлення ідентифікаторів з додатковими нулями, доданими для збільшення розмірності. Ця опція не вводить ніяких додаткових параметрів.
- Проекція швидкого з'єднання в F (x {W} + x) використовується для зіставлення розмірності (виконано за допомогою 1 × 1 згорток).

Для будь-якої з опцій, якщо швидкі з'єднання йдуть по картах об'єктів двох розмірностей, вони виконуються з кроком 2.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Рис 2.2 Кожен блок ResNet має два рівня глибини (використовується в невеликих мережах, таких як ResNet 18, 34) або 3 рівня (ResNet 50, 101, 152).

50-шарова ResNet: кожен 3-шаровий блок замінюється в 34-шарової мережі цим 3-шаровим вузьким місцем, в результаті виходить 50-шарова ResNet. Вони використовують варіант 2 для збільшення розмірності. Ця модель має 3,8 мільярда FLOPs.

ResNet з 101 і 152 шарами: вони створюють ResNet з 101 і 152 шарами, використовуючи більше 3-шарових блоків. Навіть після збільшення глибини 152-шарова ResNet (11,3 мільярда FLOP) має меншу складність, ніж мережі VGG-16/19 (15,3 / 19,6 мільярда FLOPs).

VGG-16: це модель, запропонована компанією Oxford, яка отримала 92,7% точності в конкурсі ImageNet 2013. У порівнянні з Resnet він має менші шари, отже, завдяки цьому можна набагато більше тренуватися. Для більшості існуючих орієнтирів сегментації VGG не є настільки хорошим, як ResNet з точки зору точності. До приходу ResNet VGG була стандартною попередньо підготовленою моделлю для великої кількості застосувань[10].

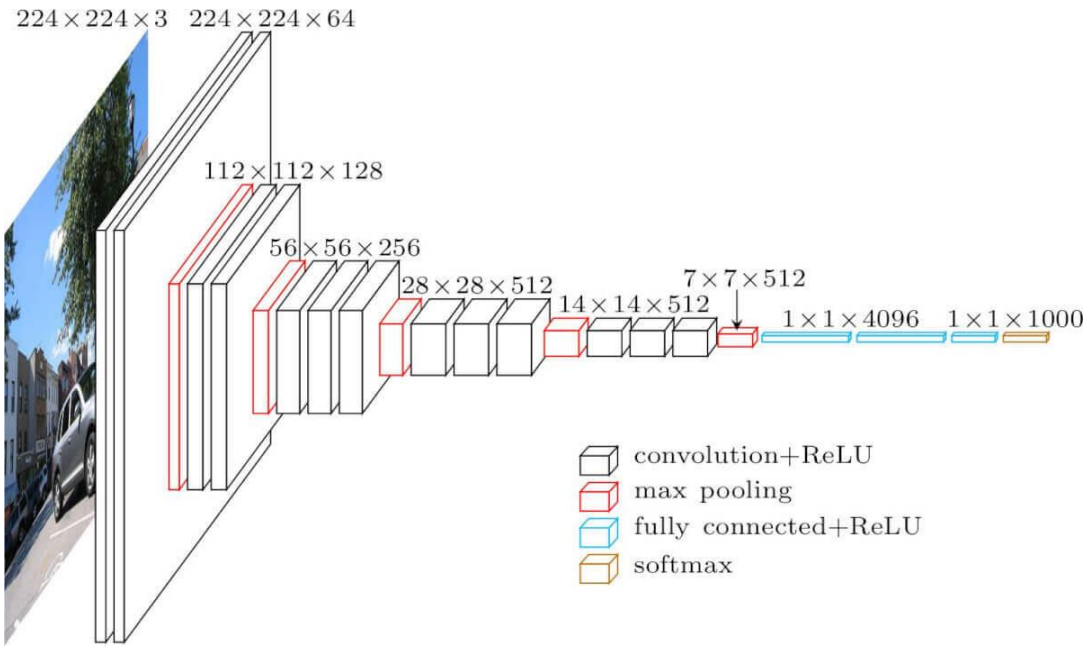


Рис 2.3 Архітектура VGG-16

На вхід шару conv1 подається RGB зображення яке має розмір 224x224 пікселів. Далі зображення проходять через стек згортальних шарів, в яких використовуються фільтри з дуже маленьким рецептивних полем розміру 3x3. В одній з описаних конфігурацій використовується згортковий фільтр котрий має розмір 1x1, який може бути представлений як лінійна трансформація вхідних каналів. Згортковий крок фіксується на значенні 1 піксель. Просторове доповнення входу згорткового шару вибирається таким чином, щоб просторову роздільну здатність можна було зберігати після згортки, тобто доповнення дорівнює 1 для 3x3 згортальних шарів. Просторовий пулінг здійснюється за допомогою п'яти шарів, які слідують за одним з згортальних шарів . Операція max-pooling виконується на вікні розміру 2x2 пікселів з кроком 2.

Після стека згортальних шарів (який має різну глибину в різних архітектурах) йдуть три повно зв’язних шару: перші два мають по 4096 каналів, третій - 1000 каналів (так як в змаганні ILSVRC потрібно класифікувати об'єкти по 1000 категоріям; отже, класу відповідає один канал).

Останнім йде soft-max шар. Конфігурація повнозв'язних верств одна і та ж у всіх нейромережах.

Всі приховані шари забезпечені ReLU. Відзначимо також, що мережі (за винятком однієї) не містять шару нормалізації (Local Response Normalisation), так як нормалізація не покращує результату на датасета ILSVRC, а веде до збільшення споживання пам'яті та часу виконання коду.

На жаль, мережа VGG має два серйозні недоліки:

- Дуже повільна швидкість навчання.
- Сама архітектура мережі важить надто багато (з'являються проблеми з диском і пропускною спроможністю)

Через велику глибину і кількості повнозв'язних вузлів, VGG16 важить більше 533 МБ. Це робить процес розгортання VGG дуже стомлюючим завданням. Хоча VGG16 і використовується для вирішення багатьох проблем класифікації за допомогою нейронних мереж, менші архітектури більш кращі. Але незважаючи на недоліки, дана архітектура є відмінним будівельним блоком для навчання, так як її легко реалізувати.

MobileNet: ця модель запропонована компанією Google, яка оптимізована для малого розміру моделі та більш швидкого виводу. Це ідеально для роботи на мобільних телефонах та обмежених ресурсах пристроях. Через невеликі розміри може бути невелике враження в точності моделі[11].

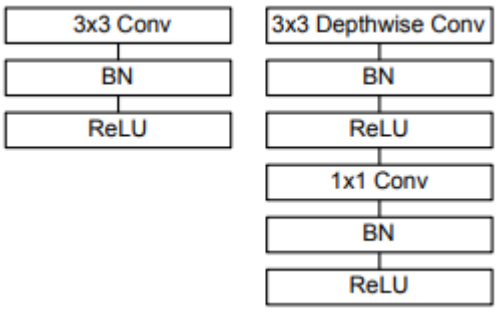


Рис 2.4 Архітектура MobileNet

Згорткова частина цікавої для нас мережі виявляється з одного звичайного згорткового шару з 3x3 згоркою в початку та тринадцяті блоковів, освіченій справі на малюнку, з поступово збільшуючим числом фільтрів і зменшуючись простою величиною тензора.

Окремістю даної архітектури є відсутнє максимум пулу шарів. Замість них для зменшення просторових розмірів використовується згортка із швидким кроком, дорівнюючим 2.

FCN: FCN є однією з перших запропонованих моделей для семантичної сегментації від кінця до кінця. Тут стандартні моделі класифікації зображень, такі як VGG і AlexNet, перетворюються на повністю згорнуті, роблячи шари FC 1x1 згортки. У FCN транспоновані згортки використовуються для надання можливості перегляду вибірки, на відміну від інших підходів, де використовуються математичні інтерполяції. Три варіанти - FCN8, FCN16 і FCN32. У FCN8 та FCN16 використовуються пропускні з'єднання[12].

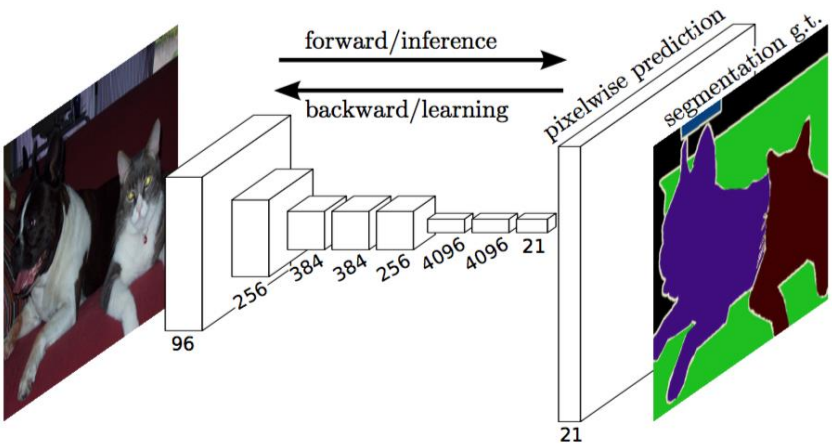


Рис 2.5 Архітектура FCN32

Основний шлях зменшення вибірки використовується для вилучення та інтерпретації контексту , що в той час як шлях до сканування використовується для точної локалізації. Крім того, для повного відновлення дрібнозернистої просторової інформації, втраченої в об'єднанні або зменшенні розмірів шарів, мережа часто використовує пропускні з'єднання.

Пропускне з'єднання - це з'єднання, яке обходить щонайменше один шар. Тут він часто використовується для передачі локальної інформації шляхом об'єднання або підсумовування карт функцій зі шляху зменшення кемпінгу із картками функцій із шляху до сканування. Об'єднання особливостей різних рівнів роздільної здатності допомагає поєднувати інформацію про контекст із просторовою інформацією.

Існують варіанти архітектури FCN, які в основному відрізняються просторовою точністю їх виведення. Наприклад, на малюнках нижче представлені варіанти FCN-32, FCN-16 та FCN-8. На малюнках згорткові шари представлені у вигляді вертикальних ліній між об'єднаними шарами, які чітко показують відносний розмір карт.

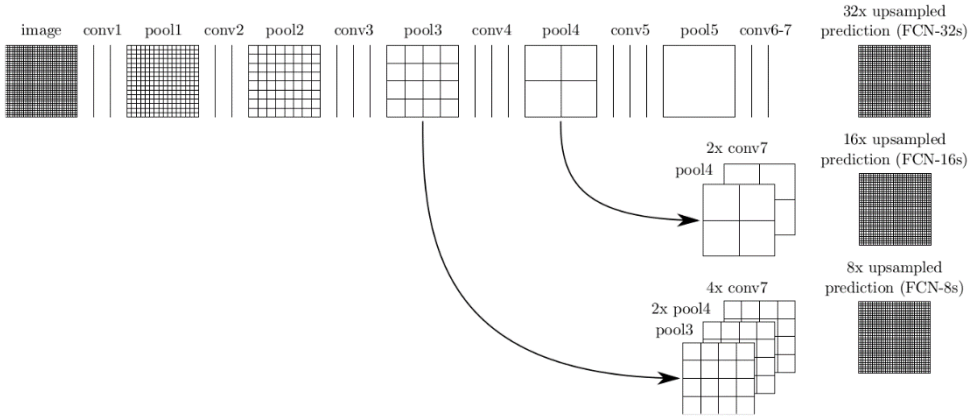


Рис 2.6 Архітектура FCN

Можна виділити головну різницю між трьома варіантами FCN.

Як показано нижче, ці 3 різні архітектури відрізняються кроком останньої згортки та пропускними з'єднаннями, які використовуються для отримання вихідних карт сегментації. Ми будемо використовувати термін пониження тиску шляху для позначення мережі до рівня conv7, і будемо використовувати термін збільшення тиску для позначення мережі, що складається з усіх шарів після conv7. Слушно зауважити, що 3 архітектури FCN мають однаковий шлях зменшення розміщення, але відрізняються за відповідними шляхами збірки.

FCN-32: безпосередньо створює карту сегментації з conv7, використовуючи транспонований шар згортки з кроком 32.

FCN-16: підсумовує 2-кратне пробне прогнозування з conv7 (використовуючи транспоновану згортку з кроком 2), а потім створює карту сегментації, використовуючи перенесений шар згортки з кроком 16 поверх цього.

FCN-8: підсумовує 2x збільшений зразок conv7, переглядає їх за допомогою перенесеної згортки кроком 2 і підсумовує їх, і застосовує перенесений шар згортання з кроком 8 на отримані карти ознак для отримання карта сегментації.

SegNet: Архітектура SegNet приймає рамки кодер-декодер. Шари кодера і декодера симетричні один одному. Операція оптимізації моделей шарів декодера використовує індекси максимального об'єднання відповідних шарів кодера. SegNet не має жодних пропускних з'єднань. На відміну від FCN, ніякі навчальні параметри не використовуються для підвищення вибору[13].

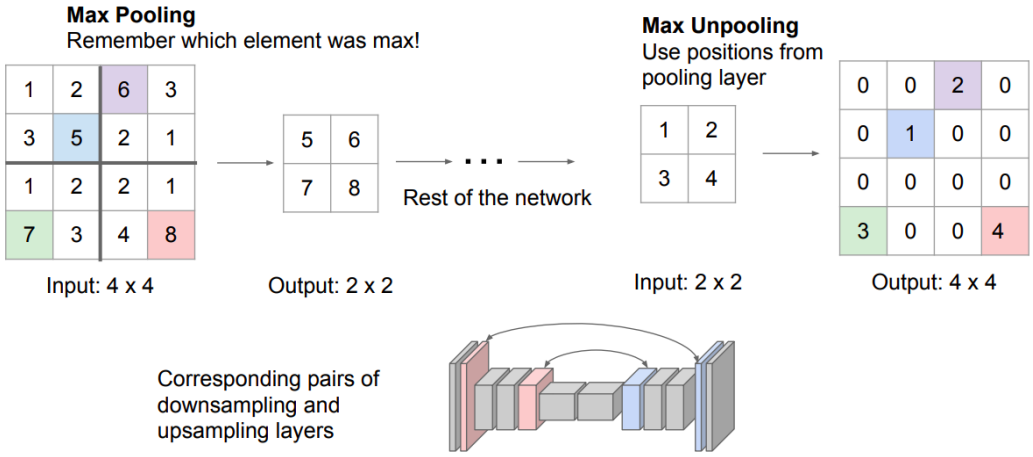


Рис 2.7 Максимальне об’єднання SegNet

UNet: Архітектура UNet приймає рамки кодер-декодер із пропущеними з'єднаннями. Як і у SegNet, шари кодера та декодера симетричні один одному.

Мережа містить стискаємий шлях (зліва) і розширює шлях (праворуч), тому архітектура схожа на букву U, що і відображено в назві. На кожному

кроці ми можемо наглядати подвоєння значної кількості каналів ознак[14]. Стискає шлях схожий на типову згорткову мережу, він містить два поспіль згорткових шара 3x3, після яких йде шар ReLU і пулінг з функцією максимуму 2 × 2 з кроком 2.

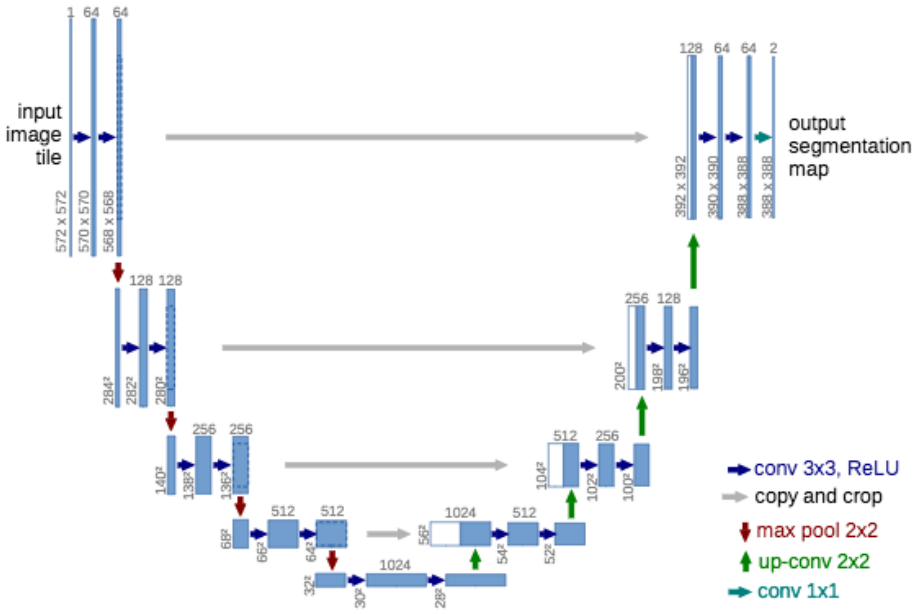


Рис 2.8 Архітектура UNet

PSPNet: ця мережа розбору сцени піраміди дуже гарно оптимізована для кращого глобального представлення контексту сцени. Спочатку зображення передається в базову мережу, щоб отримати карту функцій. Карта особливостей представлена в різних масштабах. Згортання застосовується до об'єднаних карт функцій. Після цього всі карти функцій підбираються до загального масштабу і об'єднуються разом. Нарешті, інший шар згортки використовується для отримання кінцевих результатів сегментації. Тут менші об'єкти добре фіксуються за допомогою функцій, об'єднаних до високої роздільної здатності, тоді як великі об'єкти фіксуються функціями, об'єднаними на менший розмір[15].

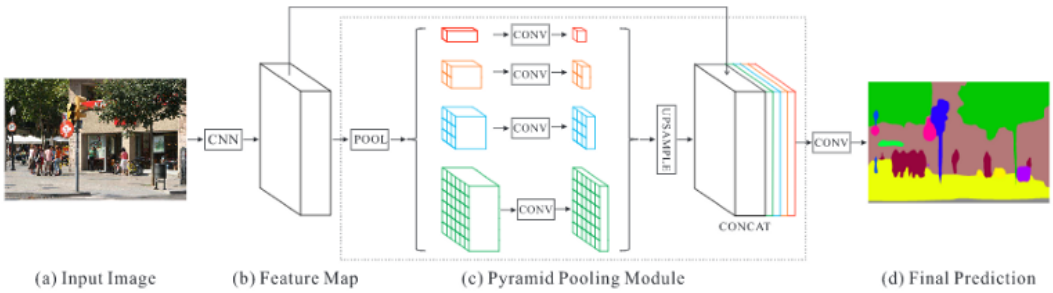


Рис 2.9 Архітектура PSPNet

Для зображень, що містять внутрішні та зовнішні сцени, кращим є PSPNet, оскільки об'єкти часто присутні у різних розмірах. Тут розмір вводу моделі повинен бути досить великим, приблизно 500x500.

Для зображень у галузі медицини популярним вибором є UNet. Завдяки пропущеним з'єднанням, UNet не пропускає крихітні деталі. UNet також може бути корисним для внутрішніх та зовнішніх сцен з предметами невеликого розміру.

Для простих наборів даних, з великим розміром і невеликою кількістю об'єктів, UNet і PSPNet можуть бути надмірними. Тут простих моделей, таких як FCN або Segnet, може бути достатньо.

2.2 Огляд алгоритмів сегментації

Сегментація - це такий процес розділення цифрового зображення на декілька сегментів. Мета сегментації полягає у спрощенні і/або зміні представлення зображення для полегшення його аналізу або передачі каналами зв'язку. Зазвичай ми використовуємо сегментацію зображень для виділення об'єктів та меж (лінії, криві) на зображеннях. Якщо виділити процес точніше то, сегментація зображень - це процес присвоєння таких міток кожному пікселю зображення, що пікселі з однаковими мітками мають спільні візуальні характеристики.

Результатом сегментації зображення є множина сегментів, які разом покривають все зображення, або множина контурів, виділених з зображення. Всі пікселі в сегменті схожі за унікальною характеристикою або за визначеною властивістю, наприклад колір, яскравість або текстура. Сусідні сегменти істотно відрізняються за цими характеристиками[16].

Бінаризація

Бінаризацією називається такий процес перетворює напівтонове зображення в зображення, яскравість пікселів якого може мати тільки два значення - 0 або 1.Таким зображення називається бінарним. Як можна побачити з визначення, зображення повинно бути напівтоновим, для того щоб його можна було бінаризувати. Для перекладу зображення з кольорового в півтонове існують такі методи[17]:

- I = 0,5 * (max (R, G, B) + min (R, G, B)), де I - яскравість пікселя котрий досліджуємо. Також слід зазначити, що при конвертації в півтонове одним з цих методів якість зображення погіршується.
- I = (R + G + B) / 3, де I - яскравість пікселя котрий досліджуємо;
- Найбільш ефективним є конвертація за формулою: Y = 0.299 R + 0.587 G + 0.114B.

Після проведення конвертації зображення можна бінаризувати вибране зображення.

Залежно від виду зображення виділяємо основні методи бінаризації:

- Метод середньої яскравості (являється локальним методом):

1. Виконуємо обчислення середньої яскравості
2. Визначаємо граничне значення яскравості за формулою: ГЗ = 255 - СЯ, де ГЗ- граничне значення, СЯ- середня яскравість.
3. Піксель стає чорним, якщо серед його сусідів є хоча б один піксель яскравість якого задовольняє умові: Яскравість ≤ ГЗ, або яскравість аналізованого пікселя ≤ яскравості зв'язкового з ним + Step, Step = 1 ... 2. При цьому важливо щоб сполучний піксель був уже включений в групу пікселів, які будуть відзначені як чорні.

- Метод «Мод»:

Потрібно виконати побудову гістограми яскравості. Для побудованої гістограми вибирається значення яскравості відповідне западині або мінімуму між двома максимальними вершинами гістограми. Недоліком даного методу є залежність результату від гістограми.

- Метод 120:

1. Потрібно побудувати гістограму
2. Потрібно знайти максимальний рівень сірого в діапазоні t = [0, 120]
3. Чорний колір встановлюється у пікселів, яскравість яких менше або дорівнює пороговому значенню плюс 12 ... 15, в інших випадках колір білий.

- Метод 40%:

1. Потрібно побудувати гістограму

2. Знаходиться кількість пікселів, що відповідає заданому відсотку
3. Потрібно відрахувати кількість пікселів по градаціях яскравості, обов'язково починаючи з 0 і до вказаного моменту, коли ця отримана кількість буде перевищувати кількість пікселів, заданих в постановці завдання бінаризації, і градація, на якій сталася постановка стає порогом бінаризації.

Метод Оцу

Цей метод за своїм призначенням використовує гістограму розподілу значень яскравості пікселів вибраного растрового зображення. Потім будується гістограма за значеннями $pi = \frac{ni}{N}$, де N - це загальна кількість пікселів на зображенні, ni - це кількість пікселів з рівнем яскравості i. Діапазон яскравості ділиться на два класи за допомогою порогового значення рівня яскравості k, k - ціле значення від 0 до L. Кожному класу відповідають відносні частоти $w_0 w_1$:

$$\omega_0(k) = \sum_{i=1}^{k} p_i \qquad \omega_1(k) = \sum_{i=k+1}^{L} p_i = 1 - \omega_0(k)$$

$$\mu_0(k) = \sum_{i=1}^{k} \frac{ip_i}{\omega_0} \qquad \mu_1(k) = \sum_{i=k+1}^{L} \frac{ip_i}{\omega_1}$$

Рис 2.10 Метод Оцу



Рис 2.11 Ліворуч-оригінальне зображення, праворуч-використання методу Оцу

Метод Бернсена

1. Ми повинні отримати звичайну квадратну апертуру з непарним числом пікселів , яка пробігає в циклі по всім пікселям вибранного зображення. На кожному кроці знаходиться Min і Max.
2. Ми знаходимо середнє значення Avg = (Min + Max) / 2.
3. Якщо поточний піксель більше Avg <E - він стає білим, інакше - чорним. E - унікальна константа задана користувачем.
4. Якщо середнє менше порога контрасту - то поточний піксель стає того кольору, який задавався параметром «колір сумнівного пікселя».

Має невеликий ряд недоліків: після обробки монотонних областей яскравості формуються сильні паразитні перешкоди, в деяких випадках призводить до появи помилкових чорних плям ,які в свою чергу створюють дефекти на зображенні.

Метод Ейквіла

В методах сегментації зображень можна виділити один з найбільш продуктивних методів котрим являється метод Еййквіла. Цей метод дуже часто застосовують для обробки контрастних і чітких зображень. В основі даного методу лежить обробка відповідного зображення за допомогою двох

концентричних вікон: маленького - S, і великого L. Зазвичай вікна отримують квадратну форму. Ці два вікна послідовно крок за кроком зліва направо зверху вниз накладаються на зображення з кроком рівним стороні маленького вікна S. Для вікна L розраховується поріг B так, щоб поділити пікселі на два кластери. Якщо математичні очікування рівня яскравості в двох кластерах мають різницю, що перевищує деякий заданий користувачем рівень $u_1 - u_2 \geq 1$, то всі пікселі всередині вікна S бінарізуються відповідно до порогом T. В іншому випадку, яскравість пікселів з вікна S замінюється деяким близьким значенням.

Метод Ніблека

В ідею даного методу було закладено основні положення зміни порогу яскравості. В бінаризації від точки до точки на підставі локального значення стандартного відхилення. Поріг яскравості в заданій точці (x, y) розраховується так:

$$B(x, y) = \mu(x, y) + k * s(x, y)$$

де μ (x, y) - середнє і s (x, y) - середньоквадратичне відхилення вибірки для деякої околиці точки. Розмір околиці обов’язково повинен бути мінімальним, але таким, щоб зберегти локальні деталі зображення. У цей же час розмір повинен бути досить великим, щоб знизити вплив шуму на результат. Значення k визначає, яку частину кордону об'єкта взяти в якості самого об'єкта. Значення k = -0.2 задає досить дуже гарний поділ об'єктів, якщо вони представлені чорним кольором, а значення k = + 0.2, - якщо об'єкти представлені білим кольором.

Метод Яновица і Брукштейна

В якості порогової поверхні даного метода бінаризації використовується поверхня потенціалів, котра будується на основі локальної максимізації градієнта яскравості. Значення градієнта яскравості дуже часто розраховується за допомогою контурного оператора Собеля або Кенні. Оператор Собеля та

Кенні дуже добре зарекомендували себе , тому в даному методі є загальна фільтрація з отримання контурних ліній. Зображення фільтрується з метою отримання контурних ліній товщини в 1 піксель, а потім ми усереднюємо це фільтром 3 × 3. Потенційну поверхню, тепер, будуємо за ітераційною інтерполюючою схемою. Розрахунок поверхні йде в порядку, починаючи від контурних пікселів. Для кожного не контурного пікселя розраховується інтерполяційний залишок R (x, y) і нове значення пікселя P (x, y) на n + 1-му кроці має розраховуватися відповідно до формулами:

$$\frac{P_{n+1}(x,y) = P_n(x,y) + \beta * R_n(x,y)}{4}$$

$$R_n(x,y) = P_n(x-1,y) + P_n(x+1,y) + P_n(x,y-1) + P_n(x,y+1) - 4 * P_n(x,y)$$

2.3. Аугментація зображень

Аугментація даних - це методика створення додаткових навчальних даних з наявних даних. Для досягнення хороших результатів глибокі нейронні мережі повинні навчатися на дуже великому обсязі даних.

Нейронні мережі потребують данні для навчання. Створення навчальних та тестових баз зображень адекватного об’єму для них є доволі таки не тривіальною задачкою. Насамперед при навчанні нейронної мережі потрібно розуміти про доцільність значної кількості навчальних зображень, тому що склад навчальної бази може впливати на якість отриманої системи розпізнавання більше ніж усі інші фактори[18].

Звідси витікає наступне – чим точніше навчальна вибірка апроксімірує генеральну сукупність зображень, котрі будуть поступати на вхід до вибраної системи, тим більше буде гранично досяжна якість результату. Тож виходить, що чітко зібрана навчальна вибірка – це є гранично конкретизоване технічне завдання.

Для початку розглянемо доступні способи підготовки вибірки зображень для створення системи розпізнавання з природних зображень.

Можна виділити такі основні етапи:

1. Збір графічних даних
2. Фільтрація
3. Підготовка інструментів для розмітки
4. Сама розмітка

Ці операції потребують доволі великої кількості часу і відповідно цей метод створення вибірки дуже затратний в плані часу[19].

А ось створення навчальних прикладів із штучних зображень має свої плюси. Цей підхід має на меті – штучну генерацію. Це означає, що ми маємо оригінальну навчальну базу зображень і хочемо її збільшити за допомогою штучної генерації зображень. За допомогою різних перетворень можна

створити необхідну кількість зображень для навчання нейронної мережі. Можна використати наступні перетворення:

1. Афінні перетворення.
2. Перспективні перетворення
3. Зміни контрасту
4. Гауссовий шум
5. Випадання регіонів
6. Зміни відтінку
7. Зміни насичення
8. Обрізання
9. Розмивання

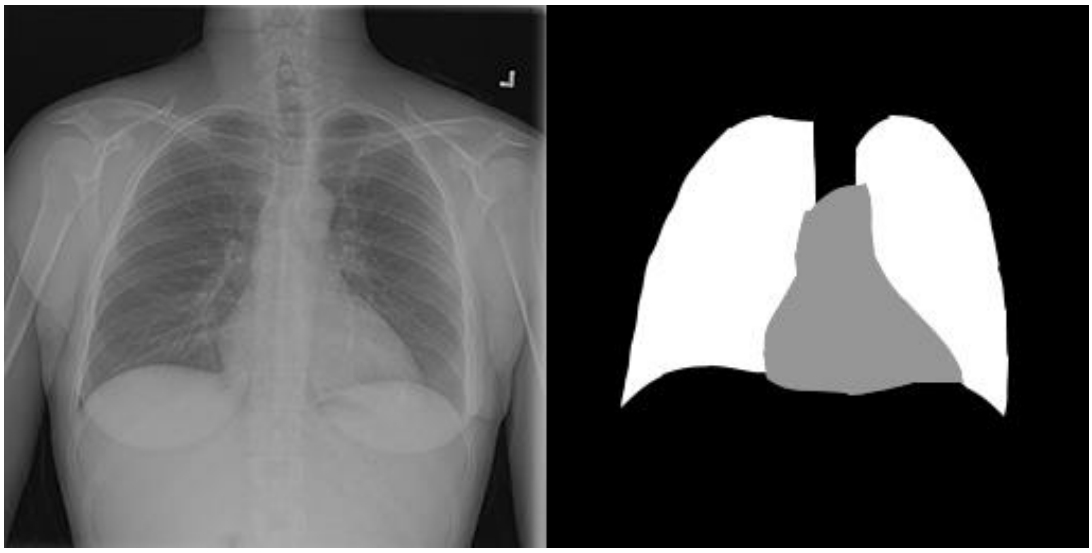


Рис 2.12 Приклад навчального зображення та його маски

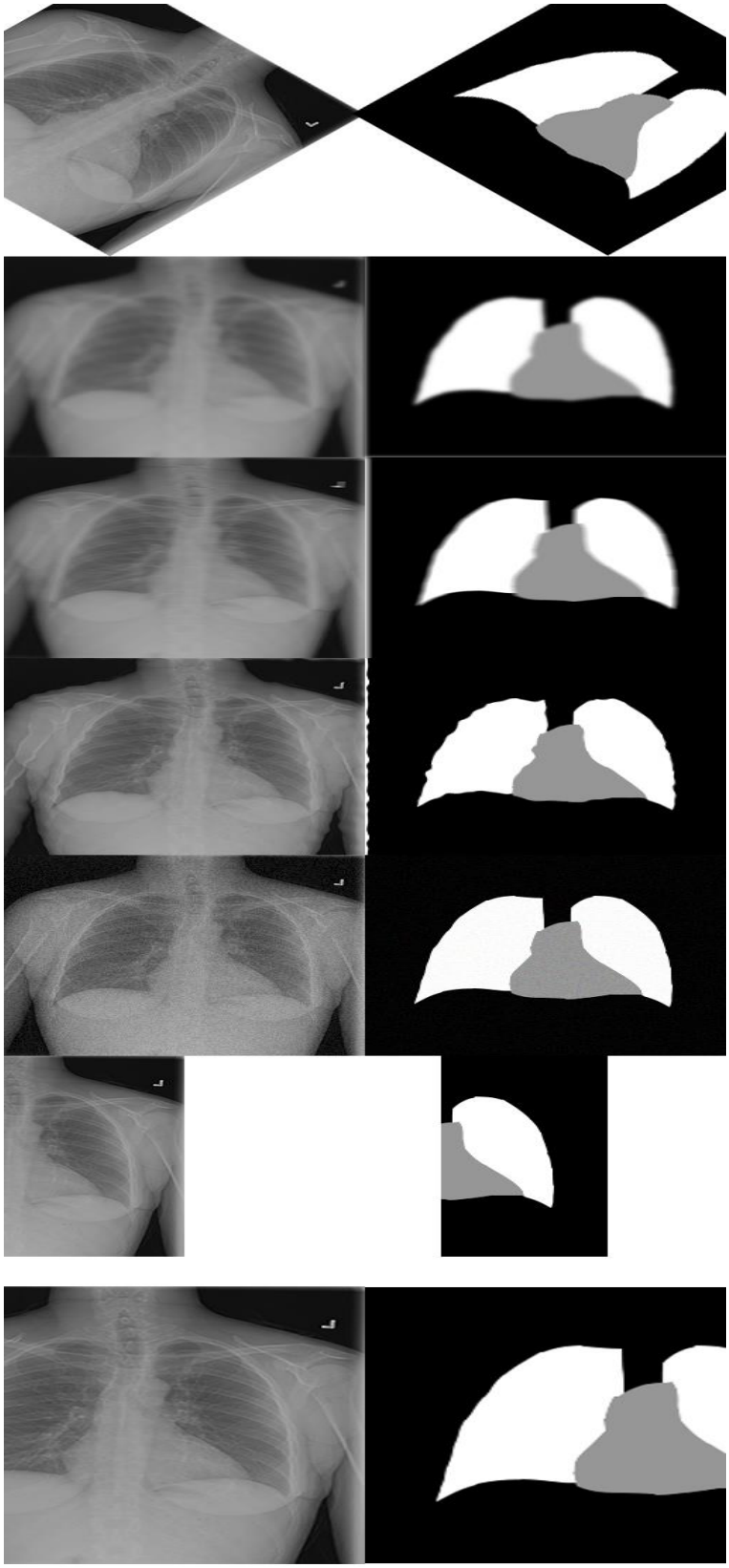


Рис 2.13 Приклад аугментації зображень

Проводити генерацію зображень можна за допомогою бібліотек, спеціальних програм, які дозволяють створювати цілі штучні документи або об’єкти, а також можна реалізувати самому ці методи перетворення[20].

Такий підхід не вимагає великої кількості людських ресурсів і в порівнянні з деякими методами дешевий, оскільки не вимагає певної розмітки і збору даних – абсолютно весь процес створення бази зображень визначається вибором алгоритму і параметрів. Але у метода є і головний мінус - це слабкий зв'язок якості роботи системи на згенерованих даних з якістю роботи в реальних умовах. Крім того, метод вимагає великих обчислювальних потужностей для створення необхідного числа прикладів. Вибір перетворень, які використовуються при створення бази для конкретного завдання, також становить певну складність.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІВ ТА ЇХ ПРАКТИЧНЕ ЗАСТОСУВАННЯ

3.1. Постановка задачі в медицині

В наш час сучасні системи штучного інтелекту вже допомагають медичним працівникам лікувати пацієнтів. Ось наприклад, деякі компанії в своїх працях використовуючи знімки КТ, комп’ютерне моделювання потоків крові та алгоритми глибокого навчання, вміють створювати 3D мапу серця. Ця невід’ємна допомога дає медичним спеціалістам можливість точніше та швидше діагностувати сердечні захворювання, значною кількістю зменшує кількість необхідних інвазійних процедур на 80%.

Але штучний інтелект може знаходити собі застосування и в інших випадках, насамперед не зв’язані з лікуванням хворого, але все одно впливаючи на якість медичного обслуговування. Вже зараз штучний інтелект дуже гарно зарекомендував себе у діагностуванні раку шкіри. Це була концепція в діагностуванні за допомогою програмного забезпечення, відповідно до якої штучний інтелект допомагає медичним фахівцям і навіть створює їм конкуренцію. Провідні експерти заявляють , що медичні зображення, включаючи фотографії, рентгенівські знімки та МРТ можуть ідеально поєднуватись з можливостями машинного навчання.

Повертаючись до діагностування раку шкіри, поступово програмне забезпечення перевершило можливості 21 дерматолога в розпізнаванні родинок, які могли бути потенціально раковими.



Рис 3.1 Приклад ШІ, знаходження пневмонії

Згідно роз'ясненням дослідників в медичному журналі Нової Англії, у рамках машинного навчання вже зараз проходить “підбір алгоритмів шляхом підстановки величезної кількості змінних в пошуках комбінацій для точного прогнозу”. Здатність штучного інтелекту обробляти великі масиви даних маючи на меті підготовку прогнозів – це вже існуючий прогрес.

Дуже важко сперечатися з тим що машинне навчання здатна покращити точність медичної візуалізації та діагностики патологій. Добре відомо, те що, проведення оцінки двома експертами буде більш точним у порівнянні з висновками одного спеціаліста. А за допомогою функції машинного навчання комп’ютер може виконувати контроль якості та виявляти найбільш часто пропускаємі особливості з ціллю значного покращення якості і результатів. Саме зараз багато починаючих компаній зробили акцент саме на цьому.

Машинне навчання може також застосовуватися для визначення скритих симптомів захворювання пацієнта та підбору відповідних методів лікування, що особливо важно для лікування захворювань, котрі вкрай потрібно діагностувати на ранніх стадіях.

Однак якими б не були видатними не сказалися б опит застосування машинного навчання, воно не зможе певною мірою замінити медичних працівників. За допомогою машинного навчання можливо підтвердити діагноз та навіть спрогнозувати результат хвороби, але не можливо вияснити причину

виникнення захворювання. Для цього як і раніше потрібен кваліфікаційний лікар.

Також експерти попереджають що активне впровадження машинного навчання не повинно випереджати дослідження його ефективності.

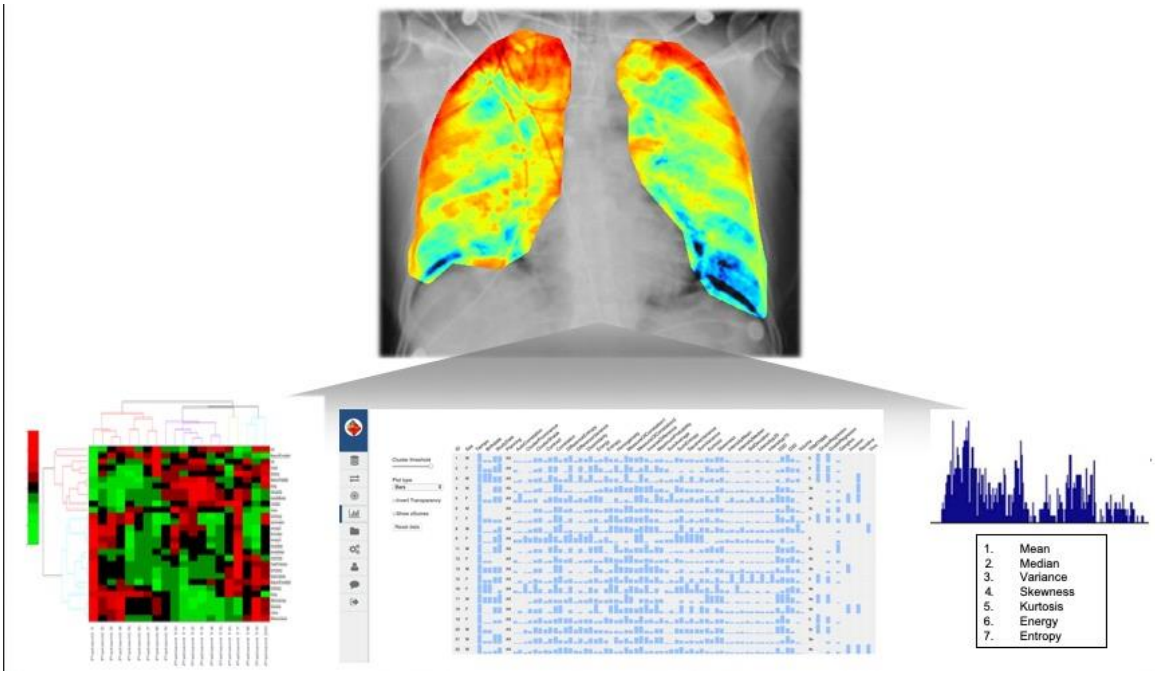


Рис 3.2 Визначення COVID-19 за допомогою МН

3.2. Створення навчальної вибірки

Перш ніж розпочинати роботу з нейронною мережею потрібно підготувати навчальну та тестову вибірку. Кількість та якість зображень відіграють майже першочергову функцію в якості та точності розпізнавання нейронною мережею зображень. Тож потрібно створити вхідні данні у RGB форматі та відповідні маски для зображень. Далі буде приведений алгоритм створення навчальної бази. При створенні програмного забезпечення були використанні існуючі наукові бази даних рентгенівських зображень грудної клітини.

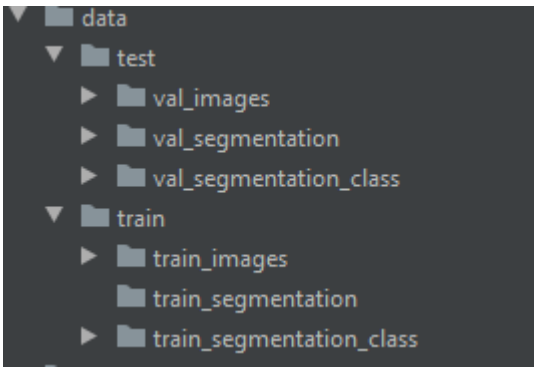


Рис 3.3 Структура бази даних зображень

До папки data складаються всі навчальні та тестові зображення. Відповідно до назв в папку test будуть складені зображення для перевірки роботи нейронної мережі. До папки train будуть складені зображення для навчання. Кожна з цих папок має по 3 вкладені папки, котрі включають у себе оригінальні зображення рентгену грудної клітини, зображення маски, там зображення з класами.

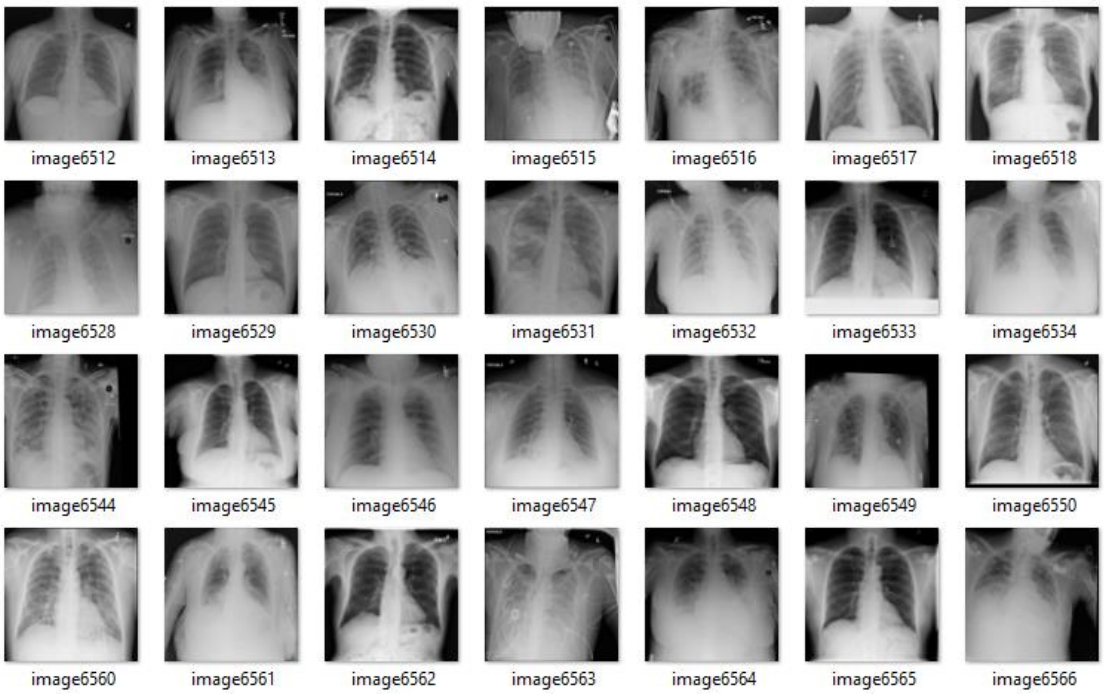


Рис 3.4 Приклад рентгенівських зображень з відкритої бази

Алгоритм створення бази зображень включає в себе такі наступні етапи:

- Перейменування всіх файлів у зручне іменування
- Конвертування зображень до формату PNG
- Якщо зображення мають градацію сірого, то обов’язково потрібно здійснити конвертацію у RGB
- Змінити розмір зображення до 256x256px, для прискорення та економії часу навчання
- Виділення регіонів які нас цікавлять, та таким образом створення маски зображення
- Створення зображення з класами відповідними до об’єктів котрі потрібно сегментувати

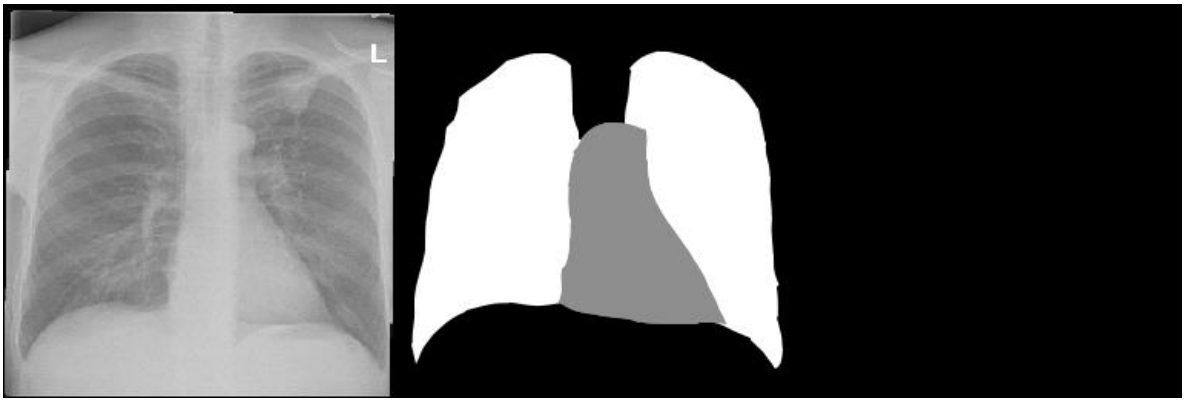


Рис 3.5 Зліва на право: оригінальне зображення, маска, зображення з класами.

Для подальшого використання нам потрібні зображення в форматі PNG та RGB фактором розміром 256x256 пікселів. Після отримання цих зображень можна приступати до ручного виділення області інтересів. Нас цікавлять легені та серце на зображенні. Для виділення елементів на зображенні буде використана програма Adobe Photoshop. Також для створення бази зображень можна використовувати інші додатки, такі як labelme або GIMP. Вибір на користь Photoshop пав за рахунок його компетентності.

Для маркування об’єктів на зображенні використовувались три кольори:

- #000000 для маркування фону
- #ffffff для маркування легень
- #8e8e8e для маркування серця

Для виділення об’єкту використовується інструмент “ магнітне ласо”, за допомогою якого ми виділяємо потрібний об’єкт по точкам. Виділивши об’єкт заливаємо його відповідним кольором. Після успішного виконання цих кроків переходимо до іншого об’єкта та повторюємо кроки. Виділивши усі бажані об’єкти зберігаємо зображення в форматі PNG.

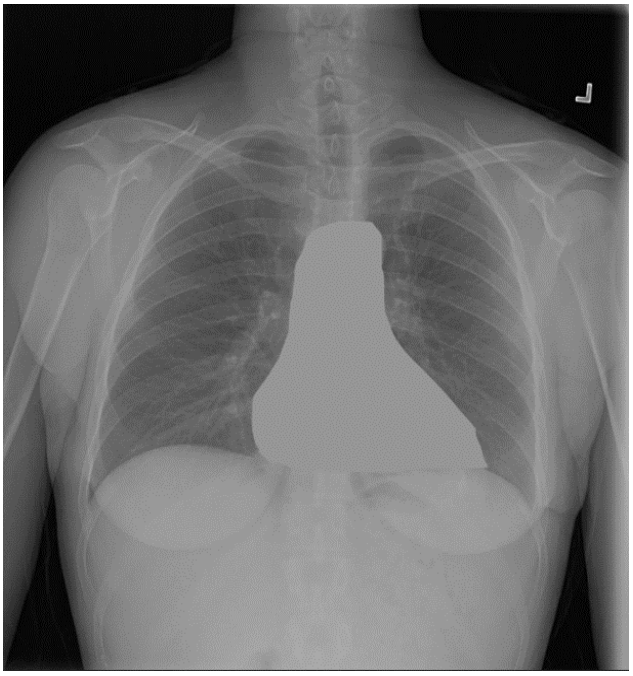


Рис 3.6 Виділення серця.

Створивши власну базу зображень, потрібно створити зображення з маркування де замість значень пікселів будуть стояти значення класів. Наприклад, ми хочемо позначити фон цифрою 0, легені цифрою 1, а серце цифрою 2. Таким чином кожен піксель буде зберігати не значення яскравості, а значення вибраного класу. Для конвертування зображення-маски в зображення з класами був використаний власноруч написаний скрипт, який брав кожен піксель зображення і перезаписував його у номер класу, котрий відповідав заданому об’єкту. Для цих потреб була використана мова Python у зв’язку з бібліотеками numpy та PIL. Після використання всіх етапів створення власної бази зображень було отримано 3022 зображення. 3002 зображень було розташовано в навчальній папці, а 20 зображень було використано для тестування мережі. Також було створено 100000 зображень за допомогою аугментації. Навчальні зображення розташовуються у папці train, а тестові зображення відповідно у папці test. Таким чином з одного зображення ми отримуємо три: оригінальне зображення, зображення-маску, та зображення з класами.

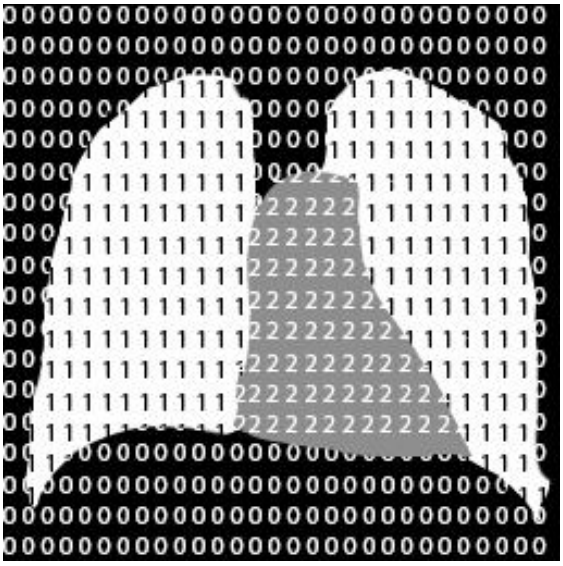


Рис 3.7 Візуальний приклад маркування об’єктів.

3.3. Навчання нейронної мережі

Після того як була створена база зображень можна приступати до створення штучної нейронної мережі, яка буде розпізнавати зображення та знаходити потрібні для нас об’єкти. Спочатку був обраний стек технологій відповідно до заданої задачі та області її використання. Мовою програмування був обраний Python. Python – це дуже популярна високорівнева мова програмування з динамічною семантикою. Він досить простий для читання та написання коду. Одна з основних причин, чому Python полягає в тому, що у нього є безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку. Також використовується бібліотека TensorFlow, розроблена для будування та тренування нейронних мереж. Також використовується бібліотека **Keras** котра спроектована для уможливлення швидких експериментів з мережами глибинного навчання. Для вирішення посталеної задачі було прийнято рішення використовувати нейронну мережу FCN-32.

Існують різні моделі для семантичної сегментації. Архітектура моделі повинна бути обрана належним чином залежно від випадку використання. Є кілька речей, які слід враховувати:

1. Кількість зображень для тренування
2. Розмір зображення
3. Домен зображень

Зазвичай моделі сегментації на основі глибокого навчання будуються на базі мережі згорткової нейронної мережі. Деякі початкові шари базової мережі використовуються в кодері, а решта сегментаційної мережі будується поверх цього. Для більшості моделей сегментації можна використовувати будь-яку базову мережу.

Першочергове завдання полягає в тому щоб для вибору моделі сегментації вибрати відповідну базову мережу. FCN є однією з перших запропонованих моделей для семантичної сегментації кінець-в-кінець.

Обравши модель ми реалізовуємо функцію яка створить вказану модель. Створюється функція з такими параметрами:

1. Кількість класів
2. Енкодер
3. Ширина зображення
4. Висота зображення

В першу чергу створюємо структуру файлів та папок. Створюємо наступні папки та файли:

- Папку Checkpoint, для зберігання чекпоінтів нейронної мережі та файлу конфігурації
- Папку Outputs, для зберігання передбачених зображень
- Файл train.py, в якій будемо проводити навчання нейронної мережі.

У файлі train.py самим першим ділом ми ініціалізуємо нашу модель наступним чином -

```
model = vgg_unet(n_classes=51 ,  input_height=256,
input_width=256)
```

Крім вибору моделі, дуже важливим є також вибір розміру вводу моделі. Якщо на зображені є велика кількість об’єктів, розмір вводу повинен бути більшим. У деяких випадках, якщо розмір вводу великий, модель повинна мати більше шарів для компенсаціях.

Стандартний розмір вводу - десь від 200x200 до 600x600. Модель з великим розміром входу вимагає більше пам’яті GPU, а також потребує більше часу для тренування.

Після підготовки набору даних та побудови моделі ми повинні тренувати модель. Навчання – це етап машинного навчання, коли модель поступово оптимізується або модель засвоює набір даних. Головна мета полягає в тому, щоб дізнатися достатньо про структуру набору навчальних даних, щоб зробити прогнози щодо небачених даних. Якщо ви дізнаєтесь занадто багато про набір даних, про тренінг то прогнози будуть працювати лише для даних які він бачив і не будуть узагальнюючими. Ця проблема називається перевиконанням – це як запам'ятовування відповідей, а не розуміння способів вирішення проблеми. Для тренування моделі створюється функція train в яку через параметри передаються: тренувальні зображення, зображення з анотаціями, шлях до папки checkpoint(в котрій зберігаємо ваги моделі) та кількість епох. На етапах навчання та оцінювання необхідно обчислити втрату моделі. Це вимірює, наскільки передбачення моделі відрізняється від бажаної мітки, іншими словами, наскільки погано працює модель. Ми хочемо мінімізувати або оптимізувати це значення. В нашому випадку ми використовуємо *categorical_crossentropy*[21]. Перехресна ентропія між двома розподілами ймовірності ρ та q над спільним простором подій вимірює середню кількість біт, необхідних для впізнання подій з простору подій, якщо схема кодування, яка використовується базується на розподілі ймовірностей q, замість істинного розподілу ρ. Перехресна ентропія двох розподілів ρ та q на тому самому ймовірнісному просторі визначається наступним чином:

$$H(p, q) = E_P[-\log q]$$

Також нам потрібен оптимізатор, котрий буде застосовувати обчислені градієнти до змінних моделі, щоб мінімізувати функцію втрат. Градієнти вказують на напрямок самого крутого підйому. Ітераційно підраховуючи втрати та градієнт для кожної партії, ми будемо коригувати модель під час тренувань. Поступово модель знайде найкраще поєднання ваг і ухилів, щоб

мінімізувати втрати. І чим менші збитки, тим краще прогнози моделі. В нашому випадку ми використовуємо оптимізатор *adadelta.*

Ададельта - це ще один варіант AdaGrad. Основна відмінність полягає в тому, що вона зменшує суму, на яку швидкість навчання адаптується до координат. Більше того, традиційно це називалося таким, що не має рівня навчання, оскільки він використовує кількість змін як калібрування для майбутніх змін. Алгоритм був запропонований в 2012. Це досить просто, враховуючи обговорення попередніх алгоритмів[22].

У двох словах, Ададельта використовує дві змінні стану, st, щоб зберігати середнє значення протікання другого моменту градієнта і Δxt, щоб зберігати середнє протікання другого моменту зміни параметрів у самій моделі. Зауважте, що ми використовуємо оригінальні позначення та найменування авторів для сумісності з іншими публікаціями та реалізаціями. Параметр du jour дорівнює ρ. Ми отримуємо такі нещільні оновлення:

$$s_t = ps_{t-1} + (1-p)g_t^2$$
$$g_t^{'} = \sqrt{\frac{\sim x_{t-1} + e}{s_t + e}} * g_t$$
$$x_t = x_{t-1} - g_t^{'}$$
$$\Delta x_t = p\Delta x_{t-1} + (1-p)x_t^2$$

Зробивши все, що було вказано, модель готова до тренувань! Навчальний цикл подає приклади наборів даних у модель, щоб допомогти зробити більш якісні прогнози. Наступний блок коду встановлює ці кроки навчання:

1. Повторіть кожну епоху. Епоха - це один прохід через набір даних.
2. У епоху потрібно переглянути кожен приклад навчального набору даних, схопивши його характеристики (x) та позначку (y).

3. Потрібно використати функції прикладу, зробіть прогноз і порівняйте його з міткою. Виміряйте неточність прогнозу та використовуйте це для обчислення втрат та градієнтів моделі.
4. Використовуємо оптимізатор для оновлення змінних моделі.
5. Слідкуємо за статистикою щодо візуалізації.
6. Повторюємо кожну епоху.

Кількість епох - це кількість разів для переходу циклу над колекцією набору даних. Навпроти інтуїтивно, тренування моделі довше не гарантує кращої моделі. Кількість епох - це параметр, який можна настроїти[23]. Вибір правильного числа зазвичай вимагає як досвіду, так і експериментування. В нашому випадку було обрано 125 епох. Після успішного навчання нам потрібно виконати передбачення, завдяки цьому мережа бере оригінальні тестові зображення і створює на його основі зображення-маску, зберігаючи його в указану папку **outputs.** Передбачення робляться завдяки тестовому набору зображень та збереженим вагам нейронної мережі. Ми використовуємо збереженні ваги для того щоб відновити нейронну мережу і за допомогою її розпізнати зображення і отримати зображення-маски. Також ми отримуємо точність 97.42%.

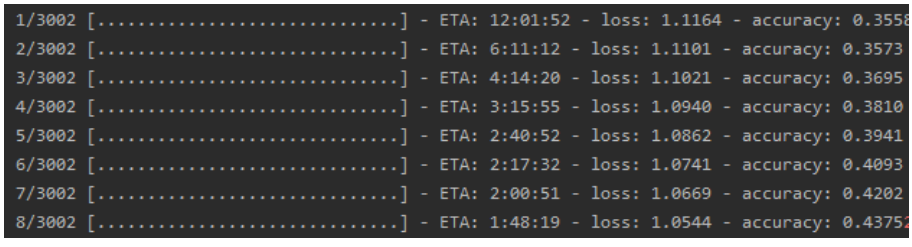


Рис 3.8 Перша епоха навчання.



Рис 3.9 Приклад зображення-маски отриманого передбаченням.

3.4. Визначення КТІ

Навчивши нейронну мережу розпізнавати зображення потрібно тепер визначати КТІ. Для цього нам потрібно знайти поперечний розмір серця та поперечний розмір грудної клітини на рівні діафрагми. Для роботи з зображення будемо використовувати бібліотеку OpenCV.

Алгоритм визначення кардіо-торакального індексу буде наступний:

1. Знайти контур легень
2. Знайти контур серця
3. Знайти крайні точки вказаних об’єктів
4. Розрахувати відстань між точками

Перш ніж починати шукати контури потрібно зрозуміти, що це таке. Контури можна пояснити просто кривою, що з'єднує всі безперервні точки (уздовж кордону), що мають однаковий колір або інтенсивність. Контури - корисний інструмент для аналізу фігур, виявлення та розпізнавання об'єктів[24].

Для кращої точності потрібно використовувати бінарні зображення. Тому перед тим, як знайти контури, застосуємо виявлення порогу чи ребра[25].

У OpenCV пошук контурів схожий на пошук білого об’єкта на чорному тлі. Тому потрібно пам’ятати, що предмет, який потрібно знайти, повинен бути білим, а фон повинен бути чорним. Для покращення результатів будемо використовувати конвертування зображення з формату RGB у градацію сірого. Для цього використаємо вбудовану функцію cvtColor, та передамо їй константу COLOR_BGR2GRAY, яка зазначає що нам потрібне зображення в градації сірого[26]. Ця функція перетворює зображення з одного кольорового простору в інший. Важливо пам’ятати головну особливість цієї бібліотеки, у разі перетворення кольорового простору в RGB, порядок каналів повинен бути визначений чітко (RGB або BGR). Зауважте, що кольоровий формат за замовчуванням у OpenCV часто називають RGB, але насправді це BGR (байти зворотні). Тож перший байт у стандартному (24-бітному) кольоровому зображенні буде 8-бітним компонент Blue, другий байт - зеленим, а третій байт - червоним. Четвертий, п'ятий та шостий байти будуть тоді другим пікселем (Синій, потім Зелений, потім Червоний) тощо.

У випадку лінійних перетворень діапазон значення не має. Але у випадку нелінійного перетворення вхідне RGB-зображення слід нормалізувати до відповідного діапазону значень, щоб отримати правильні результати. Після конвертації зображення у градацію сірого нам потрібно застосувати бінаризацію по порогу. Для цього використаємо функцію threshold, ця функція застосовує поріг фіксованого рівня до кожного елемента масиву[27].

Функція застосовує поріг фіксованого рівня до багатоканального масиву. Ця функція зазвичай використовується для отримання бінарного зображення із зображення в градації або для усунення шуму, тобто фільтрації пікселів із занадто малими або занадто великими значеннями. Існує кілька типів порогів, які підтримуються функцією. Вони визначаються за типом параметра. В даній функції використовуємо алгоритм Оцу. І нарешті використовуємо функцію findContours для пошуку контуру. Функція отримує контури з бінарного зображення за допомогою алгоритму Satoshi Suzuki та

інших. Після отримання контурів об’єктів потрібно для кожного контура знайти його крайні точки, для цього використовуються відповідні функції min та max. Отримавши точки ми можемо знайти кардіо-торакальний індекс, за допомогою евклідової відстані ми отримуємо відстань між точками, а отримавши відстань між точками ми можемо визначити кардіо-торакальний індекс. Отже спочатку ми знайшли контури легень та серця, потім знайшли крайні точки цих об’єктів, за допомогою евклідової відстані ми визначили ширину серця та легень і завдяки цьому отримали бажаний індекс.

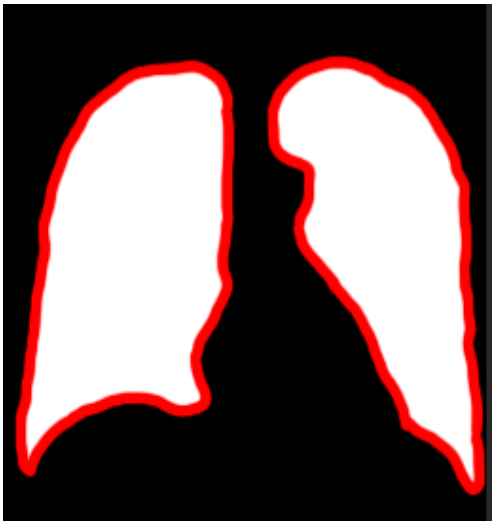


Рис 3.10 Візуалізація контурів легень.

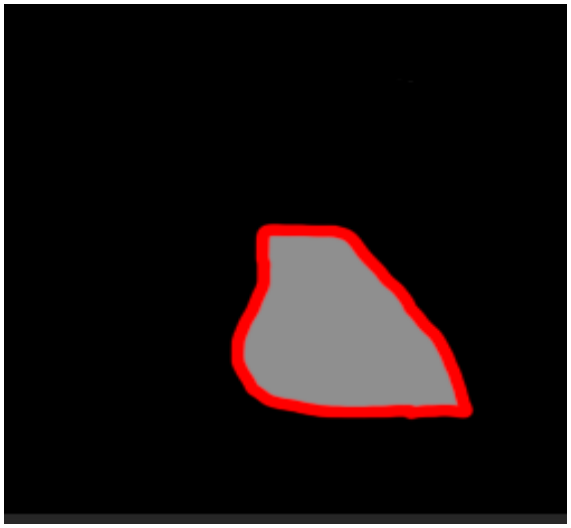


Рис 3.11 Візуалізація контурів серця.

3.5. Створення програмного забезпечення

Після того як ми навчили нейронну мережу на розпізнавання об’єктів та знаходження кардіо-торакального індексу за допомогою OpenCV можна створити графічний інтерфейс для взаємодії з програмою. Будемо використовувати бібліотеку Tkinter для створення GUI.

Для початку потрібно сформулювати задачі нашої програми. Далі приведено список основних її задач:

1. Відкриття рентгенівського зображення
2. Виведення цього зображення по центру програми
3. Розрахунок КТІ та його виведення у текстовий блок

В першу чергу створюємо меню, яке буде складатися з кнопок Файл та Довідка. За допомогою функції `add_command` створюємо вказані розділи. У меню Файл буде вибір загрузки зображення, а у Довідка буде розрахунок кардіо-торакального індексу. У центрі програми створюємо imageBox для показу зображення. На зображені нижче показано архітектуру програми. Також додаємо текстовий блок для виводу кардіо-торакального індексу. Після цього нам потрібно зробити фільтрацію файлів, щоб користувач міг загрузити тільки зображення, а не щось інше[28]. Основною функцією користувача буде буде працювати по наступному алгоритму:

1. Користувач обирає рентген зображення, воно відображається на віджеті зображення.
2. Користувач обирає розрахунок кардіо-торакальний та зображення маска відображається на віджеті зображення, а кардіо-таракальний відображається у текстовому блоці.

Алгоритм розрахунку кардіо-торакального індексу буде наступний:

1. Загружається модель по заздалегідь навчена, використовуючи ваги моделі.

2. Обране користувачем зображення передається мережі, котра в свою чергу робить прогнозування та повертає зображення-маску.
3. Потім скрипт розпізнає на зображенні-масці кардіо-торакальний індекс та повідомляє користувачу це значення.

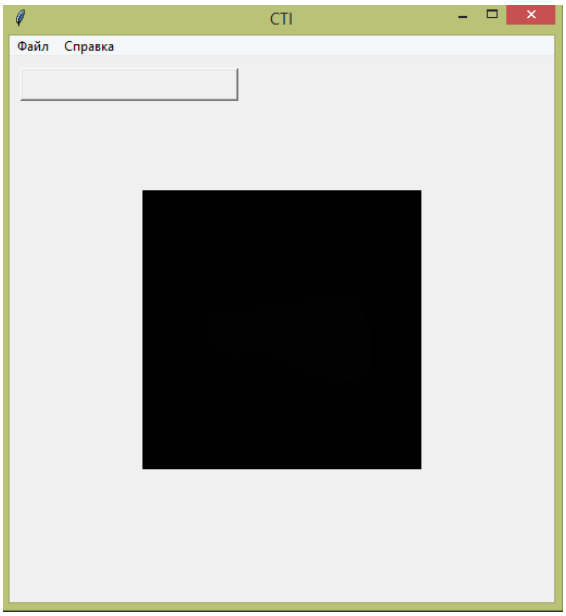


Рис 3.12 Програма розрахунку КТІ.

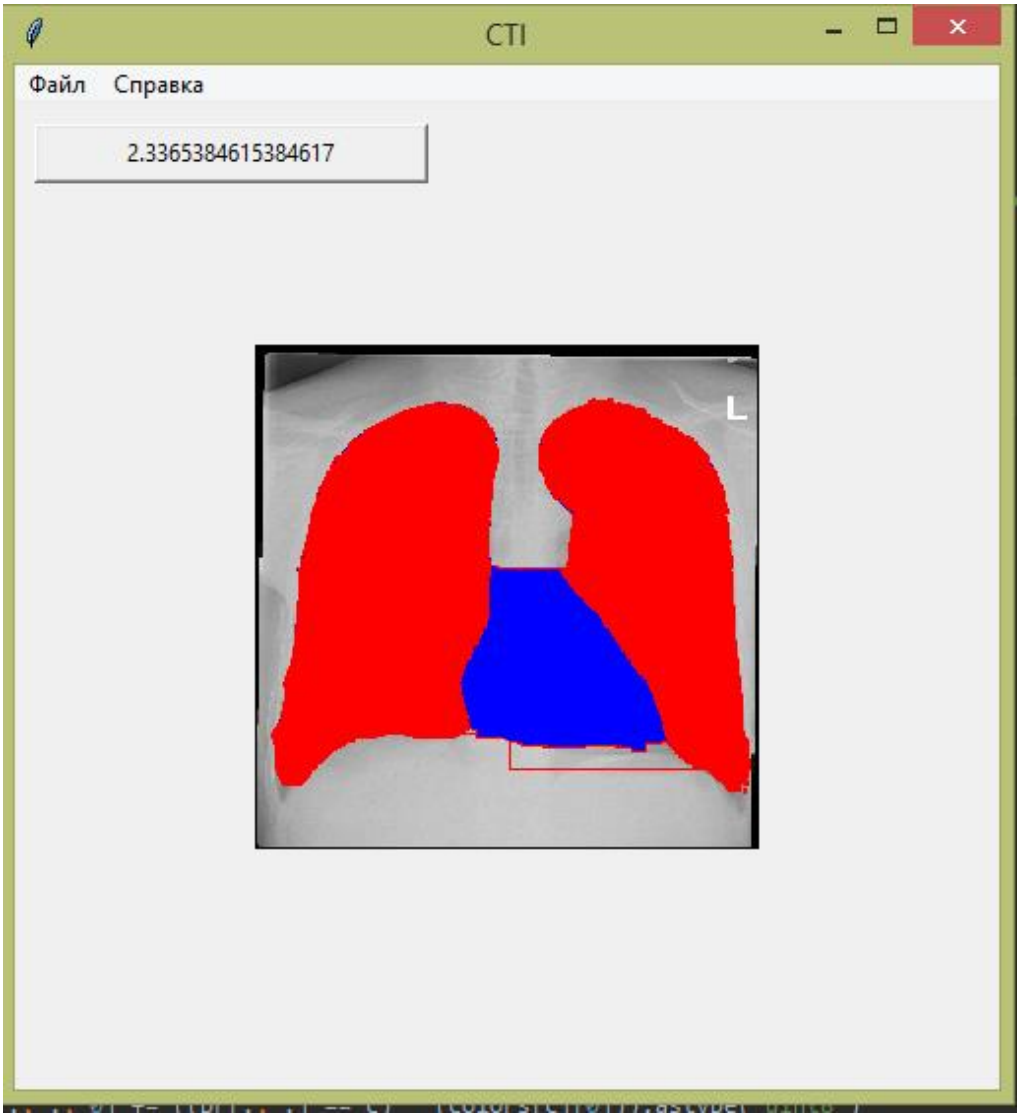


Рис 3.13 Програма виконала розрахунки, виділила об’єкти.

Створена в цій роботі програма може використовуватись на комп’ютерах з встановленим середовищем Python та встановленими необхідними бібліотеками. Перелік необхідних бібліотек:

- TensorFlow
- NumPy
- Pillow
- Keras
- OpenCV

Створена програма виконує всі вказані потреби:

- Має функцію відкриття/збереження файлу
- Має функцію визначення КТІ
- Має функцію графічного відображення легенів та серця

Після того як була створена програма потрібно провести тести на отримання точності комп'ютерного вимірювання КТІ та визначити похибку отримання результату. Для цього буде використано 10 зображень котрі не брали участь у навчанні. Використовуючи ці зображення буде визначений КТІ вручну та машинно за допомогою написаної програми. Відсоткова похибка буде визначена за формулою

$$e = \frac{a - q}{q} * 100$$

де, a – оцінкове значення, q – точне значення.

Таблиця 3.1

Порівняння ручного вимірювання КТІ та комп'ютерного

№	Ручне вимірювання	Комп'ютерне вимірювання
1	2.64	2.71
2	2.63	2.67
3	2.49	2.58
4	2.1	2.14
5	2.53	2.61
6	2.76	2.88
7	2.08	2.14
8	2.76	2.85
9	2.49	2.54
10	2.35	2.41

Скориставшись вищевказаною формулою отримаємо відсоток похибки 2.82%. Згідно цього точність комп’ютерного визначення КТІ становить 97.12%. Отримавши результат ми виконали головну мету цієї дипломної роботи – це порівняння ручного визначення та комп’ютерного визначення кардіо-торакального індексу. Таким чином використавши дану програму ми отримуємо текстовий результат визначення КТІ та графічний результат відображення органів котрі були сегментовані за допомогою машинного навчання.

ВИСНОВКИ

В процесі написання даної дипломної роботи була виконана робота по визначенню та принципам роботи комп’ютерного зору та машинного навчання. Були розглянуті основні нейронні мережі та підкреслена їх різниця. Були розглянуті основні алгоритми оптимізації з їх плюсами та мінусами. Були розглянуті на теорії та практиці всі етапи аугментації даних та їх використання. Була наглядно створена навчальна вибірка зображень для використання в згортковій нейронній мережі. Була навчена нейронна мережа для семантичної сегментації. Було написано програмне забезпечення на мові Python з графічним інтерфейсом користувача, програма отримує на вхід зображення рентгенівського знімку та визначає кардіо-торакальний індекс та виводить його. Було проведено випробування для визначення ручного та комп’ютерного визначення кардіо-торакального індексу. Було визначено похибку комп’ютерного визначення, завдяки чому отримали результат роботи алгоритму 97.12%.

В.М. Кораблев (2016) у своїй роботі “Обчислення кардіо-торакального індексу на флюрографах серії Проскан” [29], зазначив, що алгоритм ідентифікації діафрагми, контурів грудної клітини і серця, та подальшого визначення кардіо-торакального індексу, дає точність, яка не перевищує 84%, що менше результату який отримано в цієї роботі.

СПИСОК ЛІТЕРАТУРИ

1. ВОЗ[ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://WWW.WHO.INT/RU/NEWS-ROOM/FACT-SHEETS/DETAIL/CARDIOVASCULAR-DISEASES-(CVDS)
2. КАРДИОМЕГАЛИЯ [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://LMED.IN/BOLEZNI/BOLEZNI-SERDECHNO-SOSUDISTOY-SISTEMY/KARDIOMEGALIYA-3.HTML
3. НОРМАЛЬНАЯ РЕНТГЕНО-ЭХОКАРДИОГРАФИЧЕСКАЯ АНАТОМИЯ СЕРДЦА У ДЕТЕЙ РАННЕГО ВОЗРАСТА[ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTP://MEDICAL-DISS.COM/MEDICINA/NORMALNAYA-RENTGENO-EHOKARDIOGRAFICHESKAYA-ANATOMIYA-SERDTSA-U-DETEY-RANNEGO-VOZRASTA
4. НЕЙРОННАЯ СЕТЬ [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://EN.WIKIPEDIA.ORG/WIKI/ARTIFICIAL_NEURAL_NETWORK
5. МАШИННОЕ ОБУЧЕНИЕ [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://EN.WIKIPEDIA.ORG/WIKI/MACHINE_LEARNING
6. МЮЛЛЕР А. ВВЕДЕНИЕ В МАШИННОЕ ОБУЧЕНИЕ С ПОМОЩЬЮ PYTHON. РУКОВОДСТВО ДЛЯ СПЕЦИАЛИСТОВ ПО РАБОТЕ С ДАННЫМИ 2019, 480СТР.
7. ШОЛЛЕ Ф. ГЛУБОКОЕ ОБУЧЕНИЕ НА PYTHON 2010, 400СТР.
8. ДЭВИД А. Ф, ДЖИН П. КОМПЬЮТЕРНОЕ ЗРЕНИЕ. СОВРЕМЕННЫЙ ПОДХОД 2004, 928СТР.
9. RESNET [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://NEUROHIVE.IO/RU/VIDY-NEJROSETEJ/RESNET-34-50-101/
10. VGG-16 [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://NEUROHIVE.IO/RU/VIDY-NEJROSETEJ/VGG16-MODEL//
11. MOBILENETS [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://ARXIV.ORG/ABS/1704.04861/

12. FCN [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://LEONARDOARAUJOSANTOS.GITBOOKS.IO/ARTIFICIAL-INTELIGENCE/CONTENT/IMAGE_SEGMENTATION.HTML/
13. SEGNET [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://MI.ENG.CAM.AC.UK/PROJECTS/SEGNET//
14. UNET [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://ARXIV.ORG/ABS/1505.04597
15. PSPNET [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://ARXIV.ORG/ABS/1612.01105
16. А.ФЕДОРОВ БИНАРИЗАЦИЯ ЧЕРНО-БЕЛЫХ ИЗОБРАЖЕНИЙ [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTP://IT-CLAIM.RU/LIBRARY/BOOKS/ITS/WWWBOOK/IST4B/ITS4/FYODOROV.HTM
17. Х.С.ИСРАФИЛОВ ИСЛЕДОВАНИЕ МЕТОДОВ БИНАРИЗАЦИИ ИЗОБРАЖЕНИЙ [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTP://SCIENTIFICJOURNAL.RU/IMAGES/PDF/2017/VNO-30/ISSLEDOVANIE-METODOV-BINARIZATSII.PDF
18. DATA AUGMENTATION FOR DEEP LEARNING [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://TOWARDSDATASCIENCE.COM/DATA-AUGMENTATION-FOR-DEEP-LEARNING-4FE21D1A4EB9
19. DATA AUGMENTATION TENSORFLOW [ЭЛЕКТРОННЫЙ РЕСУРС] URL:HTTPS://WWW.TENSORFLOW.ORG/TUTORIALS/IMAGES/DATA_AUGMENTATION
20. DATA AUGMENTATION INCREASES ACCURACY OF YOUR MODEL [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://MEDIUM.COM/SECURE-AND-PRIVATE-AI-WRITING-CHALLENGE/DATA-AUGMENTATION-INCREASES-ACCURACY-OF-YOUR-MODEL-BUT-HOW-AA1913468722
21. ПЕРЕКРЁСТНАЯ ЭНТРОПИЯ [ЭЛЕКТРОННЫЙ РЕСУРС] URL:HTTPS://EN.WIKIPEDIA.ORG/WIKI/CROSS_ENTROPY.

22. ADADELTA [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://ARXIV.ORG/ABS/1212.5701
23. TRAIN AND EVALUATE WITH KERAS [ЭЛЕКТРОННЫЙ РЕСУРС] URL:HTTPS://WWW.TENSORFLOW.ORG/GUIDE/KERAS/TRAIN_AND_EVALUATE
24. A NEW ITERATIVE ALGORITHM FOR IMAGE RESTORATION [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://WWW.SCIENCEDIRECT.COM/SCIENCE/ARTICLE/ABS/PII/0734189X85900179.
25. CONTOURS : GETTING STARTED [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://DOCS.OPENCV.ORG/3.4/D4/D73/TUTORIAL_PY_CONTOURS_BEGIN.HTML.
26. CVTCOLOR [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://DOCS.OPENCV.ORG/3.4/D8/D01/GROUP__IMGPROC__COLOR__CONVERSIONS.HTML#GA397AE87E1288A81D2363B61574EB8CAB.
27. THRESHOLD [ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTPS://DOCS.OPENCV.ORG/3.4/D7/D1B/GROUP__IMGPROC__MISC.HTML#GAE8A4A146D1CA78C626A53577199E9C57.
28. LOVE D. TKINTER GUI PROGRAMMING BY EXAMPLE 2018, 340СТР.
29. В.М. КОРАБЛЕВ ИЗМЕРЕНИЕ КАРДИОТОРАКАЛЬНОГО ИНДЕКСА НА ФЛЮРОГРАФАХ СЕРИИ ПРОСКАН[ЭЛЕКТРОННЫЙ РЕСУРС] URL: HTTP://WWW.AMICO.RU/IMAGES/CMS/DATA/NEWSPIC/PUBLICATIY/2016/IZMERENIE_KARDIOTORAKALNOGO_INDEKSA_NA_PROSKAN.PDF