МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д.т.н., проф.,
_____Ю.П. Кондратенко
«____»_____2022 року

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НЕЙРОМЕРЕЖЕВА СИСТЕМА ДЛЯ РОЗПІЗНАВАННЯ ДОКУМЕНТІВ ТА ВВЕДЕННЯ ДАНИХ

Спеціальність 122 «Комп'ютерні наук»

122 – БКР – 401.21810109

Виконав студент 4-го курсу, групи 401
_____ ***В. Д. Жело***
« » червня 2022 р.

Керівник: канд. техн. наук, доцент
_____ ***О. В. Козлов***
« » червня 2022 р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____2022 р.

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Жело Владиславу Дмитровичу.

1. Тема кваліфікаційної роботи «Нейромережева система для розпізнавання документів та введення даних».

Керівник роботи Козлов Олексій Валерійович канд. тех. наук, доцент

Затв. наказом Ректора ЧНУ ім. Петра Могили від «07» грудня 2021 р. № 318

2. Строк представлення кваліфікаційної роботи студентом «___» _____ 20__р.

3. Вхідні (початкові) дані до роботи: документи, які зчитуватиме нейромережева система.

Очікуваний результат: успішне зчитування документів та виведення у вигляді тексту для подальшої роботи з інформацією.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– аналіз сучасного стану задачі автоматизованої обробки паперових документів;

– огляд існуючих аналогів та підприємств;

– експертне оцінювання технологій оптичного зчитування даних;

– порівняльний аналіз результатів декількох бібліотек та пакетів.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Охорона праці»

7. Консультанти розділів роботи

Консультант	Кафедра (організація)	Розділ	Підпис
старший викладач, Макарова О. В.	кафедри екології	Спеціальна частина з охорони праці	

Керівник роботи __канд. техн. наук, доцент Козлов О. В._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання ___Жело В. Д._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «_23_» _____листопада____2021 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Нейромережева система для розпізнавання документів та введення даних.

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Подання заяви на затвердження теми керівника БКР	19.11.2021	23.11.2021	виконано
2.	Отримання завдання на виконання БКР	23.11.2021	23.11.2021	виконано
3	Складання плану звіту	27.11.2021	02.12.2021	виконано
4	Складання календарного плану роботи на весь період виконання БКР	27.11.2021	08.12.2021	виконано
5	Дослідження літературних джерел, аналіз предметної області розпізнавання документів та введення даних	15.01.2022	04.03.2022	виконано
6	Аналіз існуючих аналогів, сервісів та веб-додатків – їх переваг та недоліків	25.03.2022	03.04.2022	виконано
7	Огляд існуючих бібліотек та пакетів Python	23.03.2022	26.03.2022	виконано
8	Вибір технічного та програмного забезпечення	02.04.2022	07.04.2022	виконано
9	Збір даних результатів роботи бібліотек та пакетів для зчитування даних	08.04.2022	15.04.2022	виконано
10	Аналіз отриманої статистики, подальша робота з кращою технологією	25.04.2022	01.05.2022	виконано
11	Виконання завдання БКР	06.05.2022	13.05.2022	виконано
12	Реалізація демоверсії веб-застосунку	15.05.2022	20.05.2022	виконано
13	Розробка спеціальної частини з охорони праці	21.05.2022	25.05.2022	виконано
14	Проходження переддипломної практики	26.05.2022	05.06.2022	виконано
15	Попередній захист БКР	30.05.2022	30.05.2022	виконано
16	Подання БКР рецензенту	20.06.2022	22.06.2022	виконано
17	Створення презентації для захисту та написання доповіді	20.06.2022	23.06.2022	виконано
18	Подання БКР та презентації для захисту	21.06.2022	24.06.2022	виконано
19	Захист БКР перед ЕК	27.06.2022	29.06.2022	виконано

Розробив студент Жело Владислав Дмитрович _____
(прізвище та ініціали) (*підпис*)

Керівник роботи канд. тех. наук, доц. Козлов Олексій Валерійович
(*ступень, звання, прізвище, ім'я, по батькові)* _____
(підпис)

«____» _____2021 р.

АНОТАЦІЯ
до бакалаврської роботи

Тема: «Нейромережева система для розпізнавання документів та введення даних»

Студент: Жело Владислав Дмитрович
Керівник: канд. тех. наук, доц. Козлов Олексій Валерійович

Бакалаврська кваліфікаційна робота присвячена розробці, програмній реалізації та впровадженню нейромережевої системи для розпізнавання документів та введення даних.

Об'єкт дослідження – процеси розпізнавання документів та введення даних.

Предмет дослідження – технології та програмні засоби для розпізнавання тексту та введення даних.

Мета – розробка нейромережевої системи для підвищення ефективності розпізнавання документів та введення даних.

Бакалаврська кваліфікаційна робота складається з фахової частини і спеціальної частини з охорони праці. Пояснювальна записка дипломної роботи складається зі вступу, чотирьох основних розділів, висновків та додатків.

У першому розділі виконано аналіз технологій та програмних засобів для розпізнавання тексту та введення даних. У другому розділі розглянуто моделі, методи та інформаційні технології для вирішення поставленої задачі. У третьому розділі проведено моделювання та проектування нейромережевої системи для підвищення ефективності розпізнавання документів та введення даних. У четвертому розділі виконано програмну реалізацію та розробку документації. У п'ятому розділі розглянуто питання охорони праці.

В результаті виконаної роботи зроблено висновки щодо можливості та вагомості розпізнавання документів та введення даних на основі розробленої нейромережевої системи.

Сторінок – 77, таблиць – 3, рисунків – 32, посилань – 21, додатків – 3.

Ключові слова: Нейронна мережа, розпізнавання тексту, штучний інтелект, Python, Tesseract.

ANNOTATION
to bachelor's work

Topic: «Neural network system for document recognition and data entry»

Student: Zhelo Vladislav Dmitrovich
Head: Candidate of Technical Sciences, Associate Professor
Kozlov Alexey Valerievich

The bachelor's thesis is devoted to the development, software implementation and implementation of a neural network system for document recognition and data entry.

The object of research – processes of document recognition and data entry.

The subject of research – technologies and software for text recognition and data entry.

The goal is to develop a neural network system to increase the efficiency of document recognition and data entry.

The bachelor's qualification work consists of a professional part and a special part on labour protection. The explanatory note of the thesis consists of an introduction, four main sections, conclusions and appendices.

The first section analyses the technologies and software for text recognition and data entry. The second section considers models, methods and information technologies to solve the problem. In the third section, modelling and design of the neural network system to improve the efficiency of document recognition and data entry. In the fourth section the software implementation and development of documentation is performed. The fifth section deals with occupational safety.

As a result of the performed work, conclusions were made about the possibility and importance of document recognition and data entry based on the developed neural network system.

Pages - 77, tables - 3, drawings - 32, posting - 21, supplements - 3.

Key words: Neuronal mesh, text recognition, piece intelligence, Python, Tesseract.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БКР – бакалаврська кваліфікаційна робота

ІС – інформаційна система

Лк – одиниця освітленності, Люкс

ПЕОМ – персональна електронно-обчислювальна машина

ПЗ – програмне забезпечення

СНД – Співдружність Незалежних Держав

ШІ – штучний інтелект

CPOE – (англ. «computerized physician order entry») – мережеві картки пацієнтів

Desktop – програма, що встановлюється на ПЕОМ

FPGA – Програмована користувачем вентильна матриця, ПКВМ (англ. «Field-Programmable Gate Array»)

OCR – (англ. «Optical Character Recognition») – оптичне розпізнавання символів

PATH – шлях файлу

Web – всесвітнє павутиння

Пояснювальна записка

до кваліфікаційної роботи

на тему:

НЕЙРОМЕРЕЖЕВА СИСТЕМА ДЛЯ РОЗПІЗНАВАННЯ ДОКУМЕНТІВ ТА ВВЕДЕННЯ ДАНИХ

Спеціальність 122 «Комп’ютерні науки»

122 – БКР – 401.21810109

Виконав студент 4-го курсу, групи 401
_____ ***В. Д. Жело***
«20» червня 2022 р.

Керівник: канд. техн. наук, доцент
_____***О. В. Козлов***
«20» червня 2022 р.

Миколаїв – 2022

ВСТУП

Починаючи з XX століття, ми все більше і більше починаємо впроваджувати сучасні технології в наше життя. Багато сфер та областей виробництва автоматизуються.

Але оскільки цей напрямок — штучний інтелект, автоматизація, робототехніка — відносно нові, вони регулярно покращуються та вдосконалюються.

Було знайдено дуже актуальну сферу державних соціальних систем, а саме: паспортні столи, опікунські ради, пенсійні фонди, центри зайнятості, податкові інспекції, поліцейські ділянки, соціальні фонди, оплата комунальних послуг тощо. У вищезгаданих галузях в наш час є один недолік — довгий і затягнутий процес рахунок переписувань імен, і навіть частин тексту.

Якщо в конторах України впровадяться зчитувальні обладнання, які переносять весь текст з паперу або фізичного документа у форматі тексту на комп'ютер — рутинні процеси прискорилися б у рази: досить просто використовувати дві команди — Ctrl+C та Ctrl+V.

Враховуючи всю актуальність цього питання, можна затвердити, що **об'єктом** дослідження є процеси розпізнавання документів та введення даних. Відповідно, **предметом** дослідження є технології та програмні засоби для розпізнавання тексту та введення даних, за рахунок OCR.

Мета: розробка нейромережевої системи для підвищення ефективності розпізнавання документів та введення даних.

Для досягнення мети використовуватимуться:

1) знання про OCR;
2) деякі бібліотеки, які використовують нейронні мережі пошуку та розпізнавання символів на зображеннях;
3) орієнтація у мові програмування Python;
4) допоміжні знання про Web-бібліотеки, як Django та ін.

1 АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗПІЗНАВАННЯ ТЕКСТУ ТА ВВЕДЕННЯ ДАНИХ

1.1 Аналіз якості роботи соціальних сфер

З моменту створення першої ЕОМ до сучасних суперкомп'ютерів головною метою комп'ютерів було і залишається полегшення життя людей. Але якщо раніше вони використовувалися для простих обчислень, як звичайні калькулятори, то зараз вони використовуються у всіх сферах життя: політика, освіта, оборона, фінанси, розваги та інше. Причиною все більшого поширення програмної автоматизації стала не тільки зручність, а також зростаюча обчислювальна потужність. Вона зростала за законом Мура, який стверджує, що кожні 1,5 роки кількість транзисторів на кристалі зростає на 30%.

Зі зростанням обчислювальної потужності зростає складність завдань, які поставлені комп'ютеру. Що людина буде обчислювати годинами, а можливо й днями, процесор може розрахувати за кілька секунд. Завдячуючи цій властивості, людина довіряє комп'ютеру навіть питання медицини. Починаючи з CPOE – мережевої картки з інструкціями для лікування конкретного клієнта, до DaVinci від Intuitive. - Система оперування на відстані.

Але зараз не про це. Протягом 20 останніх років, з 2000-х років, незважаючи на стрімкий розвиток технологій сучасності, інтернет-простору, веб-технологій; відносно швидке вивчення та розвиток штучного інтелекту, робототехніки, нейронних мереж; незважаючи на регулярне покращення та удосконалення технічних характеристик портативних комп'ютерів, ноутбуків, планшетів, телефонів, фотоапаратів; незважаючи на швидку автоматизацію побутових завдань, завдань виробничого характеру, немає прогресу в соціальній сфері, зокрема України.

Останні 20 років у більшості страхових компаній, пенсійних фондах, податкових інспекціях досі прижитий формат паперових документів у всіх питаннях. Це також присутній у охоронних та правоохоронних органах, таких

як швидка допомога, поліклініка, поліція, служба охорони тощо. Найголовніший атрибут на кожному робочому місці у цих галузях — величезні папки з паперами та документами. Питання стоїть зараз не про рукописні документи (від чого було б добре відмовлятися дедалі більше), а про друковані папери.

Оскільки друковані документи, протоколи, папери тощо. На сьогоднішній день активно використовуються паралельно з комп'ютерними системами, тобто має бути швидкому і продуктивному алгоритму перенесення інформації з друкованих документів (паспортних даних, свідоцтв, договорів) в електронний вигляд у текстовому форматі на комп'ютер. За цим стоїть майбутнє всього світу — відмова від паперових документів та перехід на електронні формати, електронні підписи, виписки протоколів та договорів на особисті електронні сховища чи загальні електронні документообіги тощо. Для ділової сфери це дуже актуально, оскільки немає ризику елементарно втратити фізичний документ.

Як мінімум, це хоча б частково вирішить глобальну проблему вирубування лісів, які використовують для виробництва паперу. Вирішення цього питання може торкнутися навіть екологічної сфери життя, а відповідно — і економічної, і фінансової, як мінімум.

Паралельно з поступовим переходом на електронні документи без потреби що-небудь друкувати в принципі (щодо швидкого майбутнього) надзвичайно важливо, щоб нинішні процеси оформлення не були витратними і тривалими за часом. Я зараз говорю про ті випадки, коли прийом, наприклад, до пенсійного фонду, триває мінімум годину просто через те, що величезна кількість часу витрачається на переписування паспортних даних та інших особистих документів. Хоча мета прийому може бути найпростішою.

Також як мінімум збір паспортних даних для перепису населення буде прискорено в кілька разів, тому що всі позиції вручну переписувати 5-6 хвилин, при цьому процес ускладняється ризиком помилки при переписуванні

інформації. Тим часом із запропонованим мною алгоритмом на основі оптичного зчитування символів прискорить цей процес у 10 разів.

Звісно, зараз дуже практикуються електронні черги у відповідні контори. Проте, як підказує практика та статистика, дуже великий відсоток людей просто не користуються такими послугами. Такою мірою, що все одно паралельно працює жива черга з паперовими списками, які не вселяють довіри. Як результат — електронні черги не зможуть прижитися здебільшого через відсоток населення України похилого віку, яким чужі нові технології. Тому альтернатива — прискорення процесу роботи кадрів, які обробляють кожен день дані багатьох і багатьох клієнтів, хоч би яка організація це була: чи то пенсійний фонд, нотаріус, адвокат, юриспруденція, стразова компанія тощо.

Немає сумнівів, що коли все більше і більше будуть запроваджені такі подібні пропозиції, а саме зчитування паперу на екран комп'ютера в текстовому форматі — тоді потреби в електронних чергах (принаймні на якийсь час) можуть піти на другий план, тому що їх природно не буде, або вони будуть вкрай малі і швидко зникаючі.

Безпосередньо про роботу паспортного столу. Одні з їх функцій та завдань:

1) набуття громадянства України;
2) видача паспорту громадянина України;
3) оформлення документів для виїзду за кордон;
4) реєстрація / зняття місця проживання;
5) документування іноземців;
6) набуття статусу біженця або додаткової захисту.

Без автоматизації введення та виведення особистих даних (паспорти, свідоцтва про народження, прописки тощо) працівникам дуже важко виконувати свої функції. Вони змушені робити все вручну. Їхня завантаженість настільки велика, що відгуки громадян України або негативні, або нейтрально-негативні. Тут не потрібні навіть збирання даних для

статистики. 100 відсотків відгуків є негативними. Люди висловлюють невдоволення навіть із такого простого приводу, що паспортний стіл ніколи не відповідає на телефонні дзвінки. Це все через ту саму завантаженість.

Паспортний стіл — лише одна з багатьох інших соціальних організацій, якій дуже актуально мати можливість, щоб комп'ютер рахував із фотографії текст без шансу на помилку чи друкарську помилку, як правило. Тому питання, яке торкнуться бакалаврської роботи — дуже важливе для майбутнього, не побоюся це слова, всього цивілізованого світу.

Дуже прикро та недозволено, що за часи існування незалежної України та інших країн СНД використовуються примітивні способи роботи з документами. Найголовнішим недоліком таких методів є елементарна втрата документа. Якщо немає хоча б копії документа, це спричинить багато зайвої роботи з відновлення. У гіршому випадку доведеться пожинати наслідки втрати і добре, якщо документ можна створити заново з самого початку.

Але незважаючи на все це все ж таки є зрушення саме в такому форматі, як це запропоновано в цій бакалаврській роботі. Є все ж таки прецеденти того, що паперові документи (право водія, паспорт, навіть свідоцтво про шлюб) можна перевести в електронний текстовий формат для подальшої роботи з отриманою від паперу інформацією.

Є надія, що вже в наступні 3-5 років вдасться почати з того, щоб полегшити роботу всім соціальним працівникам у різних службах та організаціях, щоб усі папери в друкованому вигляді можна було відсканувати та просто використовуючи команди "копіювати/вставити", - обробити інформацію зі знімка у тих чи інших потребах.

Отже, основні причини переходу на автоматизований формат введення даних:

1) екологічна причина - вирубування лісів;
2) черги до соціальних контор зменшуються/зникають зовсім;

3) відсутність зайвих помилок у листуванні інформації, т.к. без участі людини, тобто. немає природної похибки.

1.2 Огляд та аналіз наявних аналогів та підприємств

Як було зазначено вище, є деякі програмні забезпечення, які організовують виведення тексту з фотографії чи з камери напряму. Забезпечення ґрунтується на оптичному розпізнаванні тексту. Як мінімум, можна назвати кілька компаній та сервісів, які можуть надати подібні послуги для замовників. Ось одна з них.

1.2.1 Компанія Azoft

Компанія Azoft демонструє, як працює їхня система розпізнавання документів. Вони розповідають загалом, як працює їхній софт. У ході розробки компанія виділила 4 основні етапи роботи програми:

1) локалізація;
2) фільтрування;
3) вилучення рядків;
4) розпізнавання тексту за символами або рядками.

Локалізація — це знаходження документа на зображенні. Для локалізації ми пробували три основні підходи:

1) OpenCV та навчений класифікатор Хаара;
2) Повнозгорткові нейронні мережі;
3) Аналітичний підхід на основі пошуку зв'язкових компонентів.

У першому підході документ позначається прямокутником і обрізається до тих пір, поки в рамці не залишиться виділення з текстом; якщо документ на відео не знайдено, алгоритм продовжує пошук у відеопотоці.

У другому підході ми застосовуємо повнозгорткові нейронні мережі. На вход мережі подається кольорове зображення паспорта. На виході формується два канали: перший використовується для пошуку центру першої сторінки

паспорта, другий — для пошуку центру другої сторінки. При цьому мережа тренується пророкувати маску, в якій над центрами сторінок паспорта знаходяться гаусові піки (Рисунок 1.1).

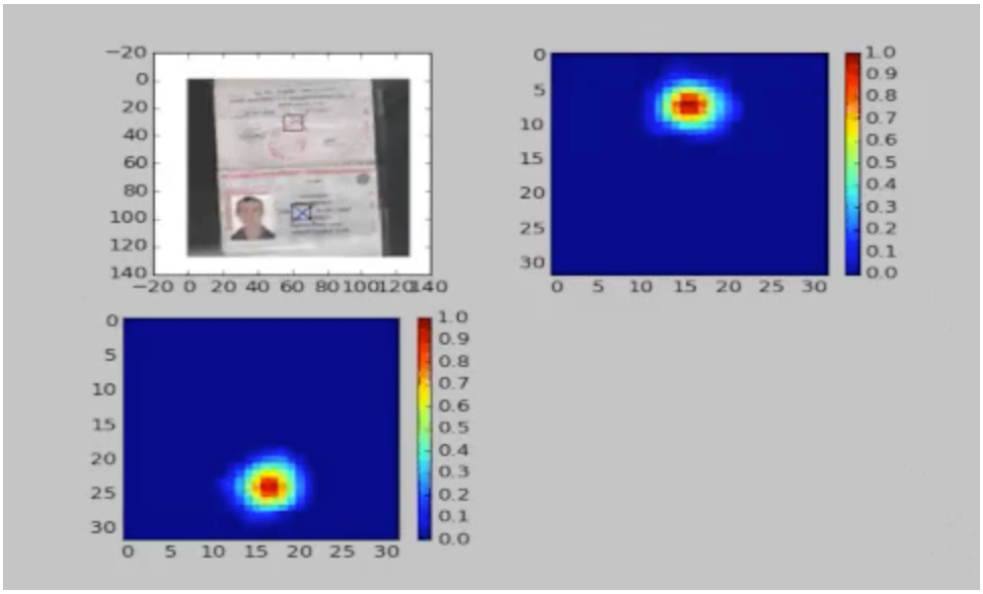

Рисунок 1.1 – гаусові пики

Третій підхід вони використали для розпізнавання паспорта. Використовували шаблон, де на розвороті паспорта знаходилися дві області із серією та номером документа. Якщо такий шаблон у кадрі, перед нами паспорт (Рисунок 1.2).

Рисунок 1.2 – шаблон

Фільтрація

Фон документів ламінований, з відблисками світла та водяними знаками. Для нейронної мережі все це шум, який заважає розпізнавати символи. Тому ми цей шум постаралися забрати.

Для цього використається:

1) алгоритм фільтрації шуму fastNlMeansDenoisingColored з вікном відповідного розміру для затирання ліній;

2) білатеральний фільтр для отримання однорідного фону;

3) метод адаптивної бінаризації.

Для водійських прав вони застосували OpenCV і детектували контури букв. Потім розбиваються рядки на букви, які потім подаються на вхід нейронної мережі для розпізнавання.

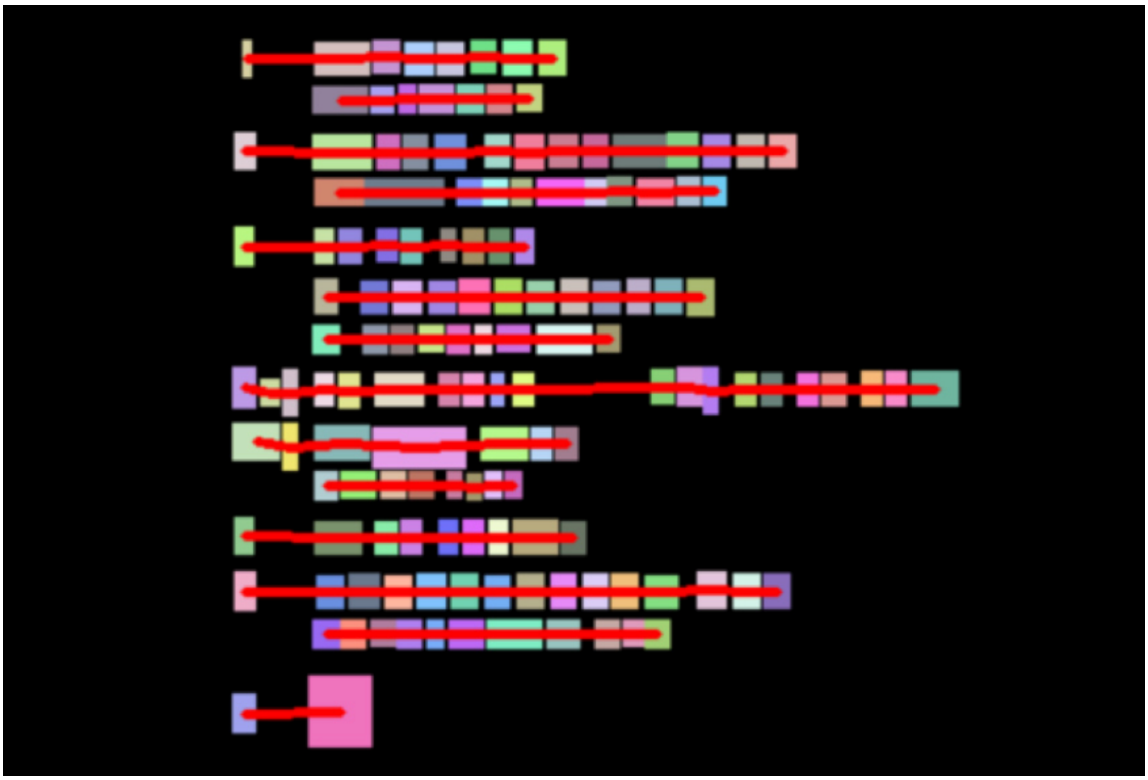

Рисунок 1.3 – як система зчитує рядки

Для отримання рядків паспорта застосовано повнозгорткову нейронну мережу.

Розпізнавання тексту

Для розпізнавання тексту використано нейронні мережі. Текст на правах розпізнано по літерах за допомогою згорткової нейронної мережі, навченої великому датасеті зображень букв. Для створення та навчання нейронної мережі використано фреймворк Torch.

Текст у паспорті також розпізнається за допомогою повнозгорткової нейронної мережі. Регістр літери при розпізнаванні не враховується. Для навчання мережі підготовлено спеціальний скрипт – генератор навчальної вибірки:

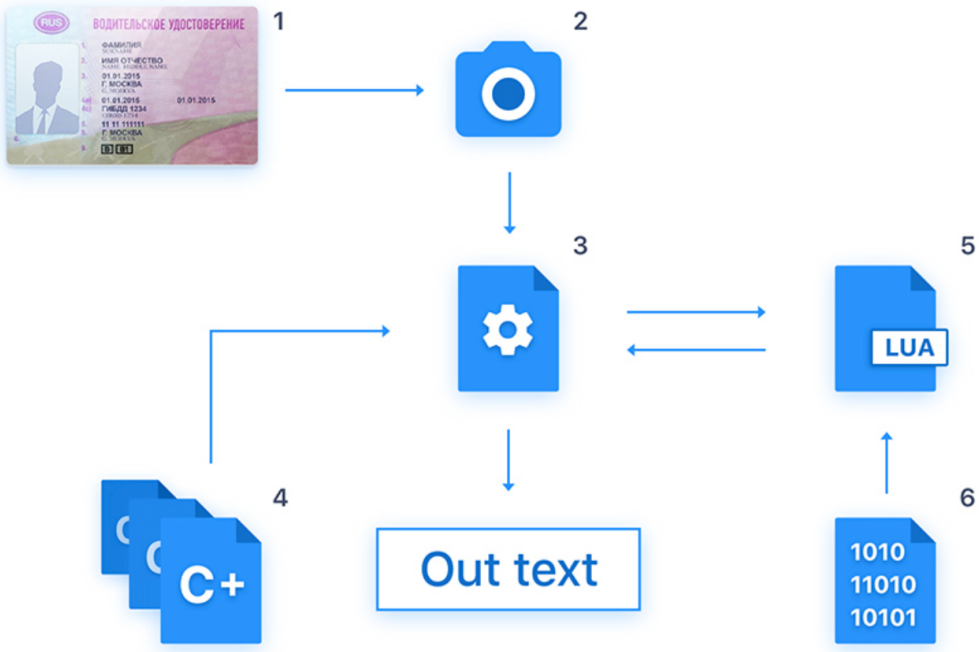

Рисунок 1.4 – структурна схема передачі та обробки даних

1) водійське посвідчення
2) камера
3) виконуваний бінарний файл

4) вихідні коди C++, у тому числі зібраний виконуваний файл Lua скрипт, який здійснює роботу з нейромережею. У ньому описано архітектуру мережі.

Бінарний файл з вагами мережі, що завантажується Lua скриптом. Вага отримана після навчання мережі.

1.2.2 Компанія Euresys та їхній продукт EasyOCR2

Euresys — провідна інноваційна високотехнологічна компанія, розробник та постачальник компонентів для збирання зображень та відео, пристроїв захоплення кадрів, IP-ядер FPGA та програмного забезпечення для обробки зображень. Euresys займається комп'ютерним зором, машинним зором, промисловою автоматизацією та медичною візуалізацією.

Досвід компанії в області збирання зображень охоплює отримання аналогового та цифрового відео, програмування FPGA, високочастотну електроніку, стиснення відео та управління камерою. Завдяки придбанню компанії Sensor to Image, її 30-річний ноу-хау в області обробки зображень включає GigE Vision, USB3 Vision, CoaXPress, Camera Link та GenICam.

З точки зору аналізу зображень, навички Euresys застосовні до виявлення плям, субпіксельного вимірювання, зіставлення образів, аналізу кольору, оптичного розпізнавання символів, зчитування та перевірки штрих-коду, 3D-перевірки та класифікації з використанням глибокого навчання.

EasyOCR2 знаходить символи на зображенні так:

1. EasyOCR2 сегментує зображення, знаходячи краплі, що становлять (частини) символів.

2. Краплі, які занадто великі або занадто малі, щоб вважатися частиною символу, відфільтровуються.

3. EasyOCR2 підганяє блоки символів до виявлених великих двійкових об'єктів відповідно до заданої топології та методу виявлення.

Топологія визначає структуру тексту на зображенні, визначаючи кількість рядків, кількість слів у рядку та кількість символів у слові.

4. EasyOCR2 витягує пікселі всередині кожного символьного поля зображення.

Отримані зображення можна використовувати для вивчення або розпізнавання символів.

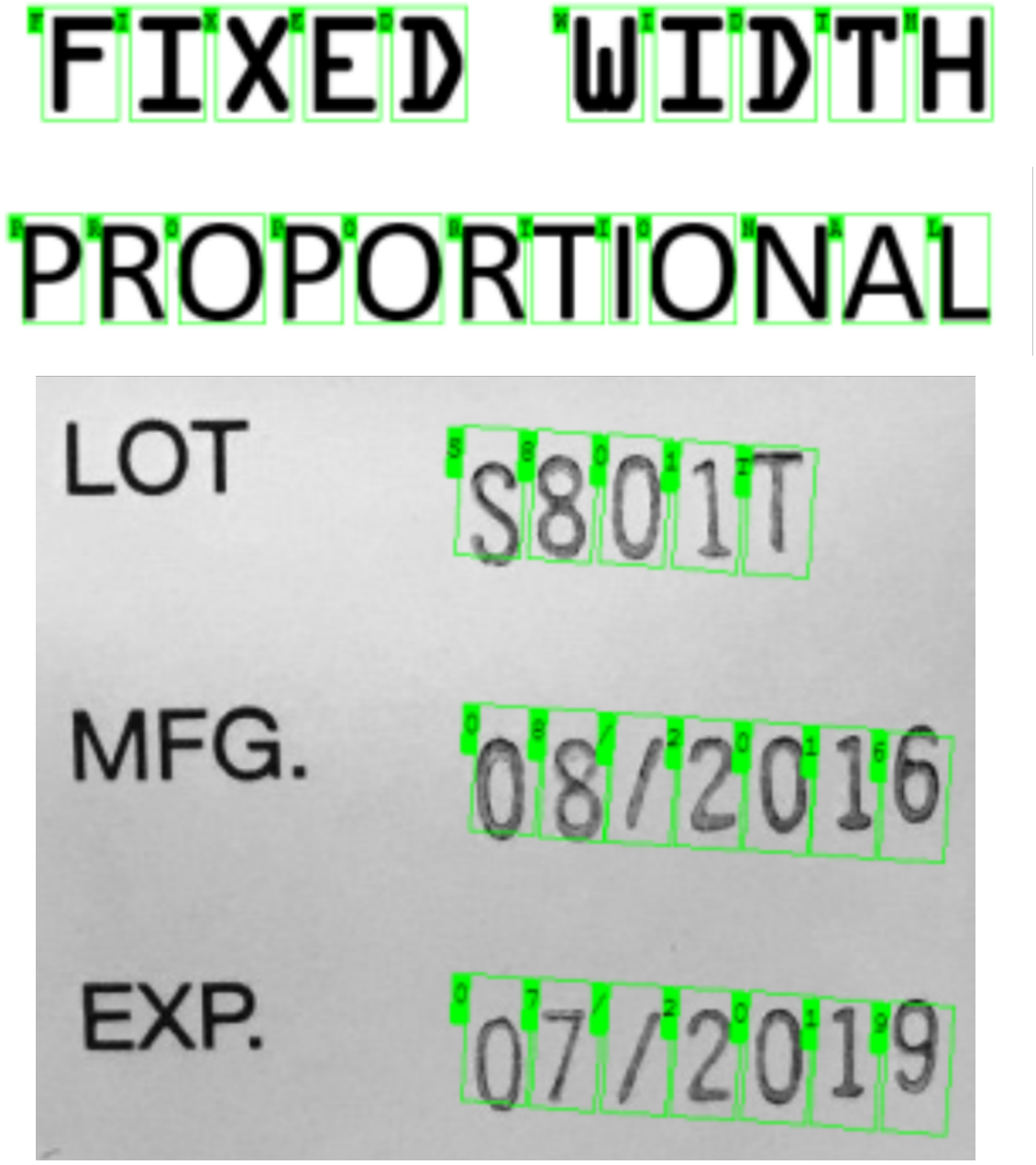

Рисунок 1.5 – демонстрація роботи платного та ексклюзивного продукту компанії Euresys

1.3 Постановка задачі

Враховуючи всю актуальність цього питання, можна затвердити, що об'єктом дослідження є нейронні мережі, як найпопулярніший напрямок у сфері розробки та дослідження штучного інтелекту. Відповідно, предметом дослідження є створення та налаштування роботи нейронної системи для розпізнавання тексту на основі OCR – оптичного розпізнавання символів.

Ціль: розпізнавання документів з подальшою можливістю використовувати розпізнаний текст.

Для досягнення мети використовуватимуться:

1) знання про хоча б одну OCR;

2) деякі бібліотеки, які використовують нейронні мережі пошуку та розпізнавання символів на зображеннях;

3) орієнтація у мові програмування Python.

Для виконання цього завдання повинні бути описані *вимоги проекту*.

Призначення — програма буде створена для зчитування тексту з фотознімка або будь-якого іншого фото з текстом.

Кордони проекту — зчитування фотографії та виведення тексту в рядковому типі змінних.

Класи та характеристики користувачів — користувачами можуть бути будь-які фізичні або юридичні особи.

Основний користувач — працівник соціальної сфери.

Вимоги до даних — дані мають бути у форматі фотографії (png, jpg). Якість фотографії хороша без розмиття, тому що зчитування може давати збої. Також полегшити роботу може скан того чи іншого документа.

Висновки до розділу 1

Виходячи з усього того, що було описано в першому розділі, можна підсумувати, що проблематика, яка була зачеплена, вкрай необхідна сьогодні

і також у наступних роках, коли починається епоха комп'ютеризації абсолютно всіх сфер життя та діяльності. Немає жодних сумнівів, що подібні продукти будуть розвиватися все більше і більше, ІІ буде все більш досконалим і мало чим відрізнятиметься від роботи та розумового процесу людини.

Щодо безпосередньої мети цієї БКР – необхідно полегшити роботу всіх соціальних служб, які щодня обслуговують величезну кількість людей, зокрема людей похилого віку.

Це полегшить роботу величезної кількості працівників соціальних служб та послуг.

2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Методи для вирішення задачі

OCR – це використання технології для ідентифікації та перетворення відсканованих рукописних або друкованих текстових символів на електронну форму, що легко розпізнається комп'ютерами та іншими програмами. Базовий процес розпізнування включає вивчення тексту та переклад символів код, який можна використовувати для обробки даних. OCR іноді також називають розпізнаванням тексту.

Історія оптичного розпізнавання символів

У 1974 році Рей Курцвейл заснував компанію Kurzweil Computer Products, Inc., чий продукт для багатошрифтового оптичного розпізнавання символів (OCR) міг розпізнавати текст, надрукований практично будь-яким шрифтом. Він вирішив, що найкращим застосуванням цієї технології буде пристрій для машинного навчання для сліпих, тому він створив машину для читання, яка могла б читати текст вголос у форматі синтезу мовлення. У 1980 році Курцвейл продав свою компанію компанії Xerox, яка була зацікавлена в подальшій комерціалізації перетворення тексту з паперу в комп’ютер.

Технологія OCR стала популярною на початку 1990-х років під час оцифрування історичних газет. З тих пір технологія зазнала ряд удосконалень. Сучасні рішення мають можливість забезпечити майже ідеальну точність OCR. Для автоматизації складних процесів обробки документів використовуються передові методи. До того, як технологія OCR була доступна, єдиним варіантом цифрового форматування документів було введення тексту вручну. Це не тільки забирало багато часу, але й викликало неминучі неточності та помилки друку. Сьогодні послуги OCR широко доступні для населення. Наприклад, Google Cloud Vision OCR

використовується для сканування та зберігання документів на вашому смартфоні.

Технологія складається з поєднання апаратного та програмного забезпечення, яке використовується з метою перетворення фізичних документів у машиночитань текст. Апаратне забезпечення, таке як оптичний сканер або спеціалізована монтажна плата, використовується для копіювання або читання тексту, тоді як ПЗ відповідає за розширену обробку. ПЗ може використовувати штучний інтелект для реалізації досконаліших методів інтелектуального розпізнавання (ICR), таких як ідентифікація мов або стилів рукописного введення.

OCR найчастіше використовується для перетворення друкованих юридичних або історичних документів на PDF-файли. Після цього отримані електронні копії користувачі можуть редагувати, форматувати за допомогою звичайних редакторів тексту.

Як працює OCR

Першим кроком процесу оптичного розпізнавання є використання сканера для обробки фізичної форми документа. Після копіювання всіх сторінок програма OCR перетворює документ на двоколірну або чорно-білу версію. Відскановане растрове зображення аналізується наявність світлих і темних областей. Темні області ідентифікуються як символи, які необхідно розпізнати, а світлі області – як тло.Після цього темні області обробляються для пошуку літер або цифр.

Існуючі програми розпізнавання можуть мати різні методи роботи, але, як правило, всі вони включають націлення на один символ, слово або блок тексту. Для ідентифікації символів використовуються два основних алгоритми.

Обробка розпізнається матеріалу відбувається на прикладах різних шрифтів і текстових форматів.

Розпізнавання ґрунтується на використанні правил виявлення ознак, що

стосуються особливостей конкретної літери або цифри (ICR). За допомогою функції виявлення ПЗ оцінює дані документа відповідно до правил, як формується буква або цифра. Наприклад, велика літера "A" може зберігатися як дві діагональні лінії, що перетинаються з горизонтальною лінією посередині.

Коли символ ідентифікований, він перетворюється на код ASCII, який можна використовувати комп'ютерними системами. Перед збереженням подальшого використання оброблені тексти необхідно перевірити зміст помилок на правильність складних макетів.

Варіанти використання:

1) сканування друкованих документів у версії, які можна редагувати за допомогою традиційних редакторів тексту;

2) індексування друкованого матеріалу для пошукових систем.

3) автоматизована обробка та введення даних;

4) розшифровує документи в текст, який може бути прочитаний вголос для користувачів із порушеннями зору;

5) архівування історичної інформації (газет, журналів), а також пошук за ними;

6) розміщення важливих підписаних юридичних документів в електронній базі даних. Розпізнавання номерних знаків за допомогою камери контролю швидкості та програмного забезпечення підсвічуванням;

7) сортування листів для доставки пошти;

8) переклад слів у зображенні на задану мову;

9) забезпечення пошуку сканованих книг.

До того, як з'явилася технологія OCR, єдиним методом оцифрування паперових носіїв був ручний повторний друк тексту. Цей процес займав багато годин, а також часто приводив до помилок під час друку. Використання OCR заощаджує годину, допомагає виключити помилки, мінімізувати зусилля. Крім цього, технологія дозволяє виконувати дії, які недоступні для фізичних

копій, наприклад, може використовувати стиснення у ZIP-файлі, виділяти ключові слова, розміщувати документи на веб-сайті, прикріплювати їх до електронної пошти.

Існує два основних методи розпізнавання.

Розглянемо перший випадок – розпізнавання за допомогою *метрики*.

Метрика — це деяке умовне значення функції, що визначає положення об’єкта в просторі. Таким чином, якщо два об’єкти розташовані близько один до одного, тобто вони схожі (наприклад, дві літери А, написані різними шрифтами), то показники для таких об’єктів будуть збігатися або бути надзвичайно схожими. Для розпізнавання в цьому режимі була обрана метрика Хеммінга.

Метрика Хеммінга — це показник, який показує, наскільки сильно об’єкти не схожі один на одного.

Цей показник часто використовується при кодуванні інформації та передачі даних. Наприклад, після сеансу передачі вихід має таку послідовність бітів (1001001), також відомо, що має прийти інша бітова послідовність (1000101). Треба обчислити метрику, порівнюючи частини послідовності з відповідними місцями в іншій послідовності. Таким чином, метрика Хеммінга в нашому випадку дорівнює 2. Оскільки об’єкти відрізняються двома положеннями. 2 – ступінь несхожості, чим більше, тим гірше в нашому випадку.

Тому, щоб визначити, яка буква зображена, потрібно знайти її метрику з усіма готовими шаблонами. І відповіддю буде шаблон, метрика якого найближча до 0.

Але, як показала практика, підрахунок лише однієї метрики не дає позитивного результату, тому багато букв схожі один на одного. напр. «j» «i», що призводить до неправильної ідентифікації.

Тоді було вирішено придумати нові метрики, які дозволяють виділити певний набір букв в окремий клас. Зокрема, були реалізовані метрики

(Відображення по горизонталі і вертикалі, поширеність ваги по горизонталі і вертикалі) .

Експеримент виявив, що такі літери, як «H» «I» «i» «O» «o» «X» «x» «l» мають суперсиметрію (повністю збігаються зі своїми відображеннями і значущі пікселі розподілені рівномірно по всьому зображенню), тому їх винесли в окремий клас, що зменшує перерахування всіх метрик приблизно в 6 разів. Подібні дії були проведені і для інших листів. В середньому скорочення переліку досягає приблизно в 3 рази.

Існує також унікальна літера, така як «J», яка є єдиною у своєму класі, і тому однозначно ідентифікована. Далі для кожного класу розраховується метрика Хеммінга, яка на цьому етапі дає кращу продуктивність, ніж при прямому застосуванні.

При створенні шаблонів використовувався шрифт «consolas», тому якщо розпізнаний текст написаний цим шрифтом, точність розпізнавання становить близько 99 відсотків. Коли ви змінюєте шрифт, точність падає до 70 відсотків.

Другий спосіб розпізнавання — за допомогою *нейронної мережі*.

Що таке *нейронна мережа*, як в біологічному сенсі, так і в математичному, дуже складна тема. Можна лише сказати, що в математичному сенсі нейронна мережа є лише моделлю біологічного визначення.

Також існує безліч варіацій цих моделей. Наприклад, одношарова мережа Кохонена. Принцип роботи нейронної мережі такий, що після викладання нового зображення на вхідний шар нейронів мережа реагує на імпульс того чи іншого нейрона. Оскільки всі нейрони називаються за значеннями букв, отже, реакційний нейрон несе відповідь розпізнавання. Заглиблюючись у термінологію мереж, можна сказати, що окрім виходу нейрон має ще й багато входів. Ці вхідні дані описують значення пікселя зображення. Тобто, якщо є зображення розміром 16x16, мережа повинна мати 256 входів.

Кожен вхід сприймається з певним коефіцієнтом, і в результаті в кінці розпізнавання на кожному нейроні накопичується певний заряд, заряд буде більшим за цей нейрон і видаватиме імпульс.

Але для того, щоб вхідні коефіцієнти були правильно налаштовані, спочатку потрібно навчити мережу. Цим займається окремий навчальний модуль. Цей модуль бере наступне зображення з навчального набору і передає його в мережі. Мережа аналізує всі положення чорних пікселів і вирівнює коефіцієнти, мінімізуючи помилку збігу за допомогою градієнтного методу, після чого це зображення порівнюється з певним нейроном.

2.2 Технології розробки системи

Це завдання може бути реалізовано декількома технологіями.

Природно, використовуватиметься мова програмування Python, оскільки завдання пов'язане зі штучним інтелектом та нейронними мережами.

Технології:

1) Tesseract OCR;
2) Easy OCR.

Tesseract OCR

Tesseract – це механізм розпізнавання тексту (OCR) з відкритим вихідним кодом, доступний за ліцензією Apache 2.0. Його можна використовувати безпосередньо або (для програмістів) за допомогою API для вилучення друкованого тексту із зображень. Він підтримує багато мов. Tesseract не має вбудованого графічного інтерфейсу, але деякі з них є на сторінці сторонніх розробників. Tesseract сумісний з багатьма мовами програмування та фреймворками через оболонки, які можна знайти тут. Його можна використовувати з наявним аналізом макета для розпізнавання тексту у великому документі або його можна використовувати у поєднанні із зовнішнім детектором тексту для розпізнавання тексту із зображення одного текстового рядка.

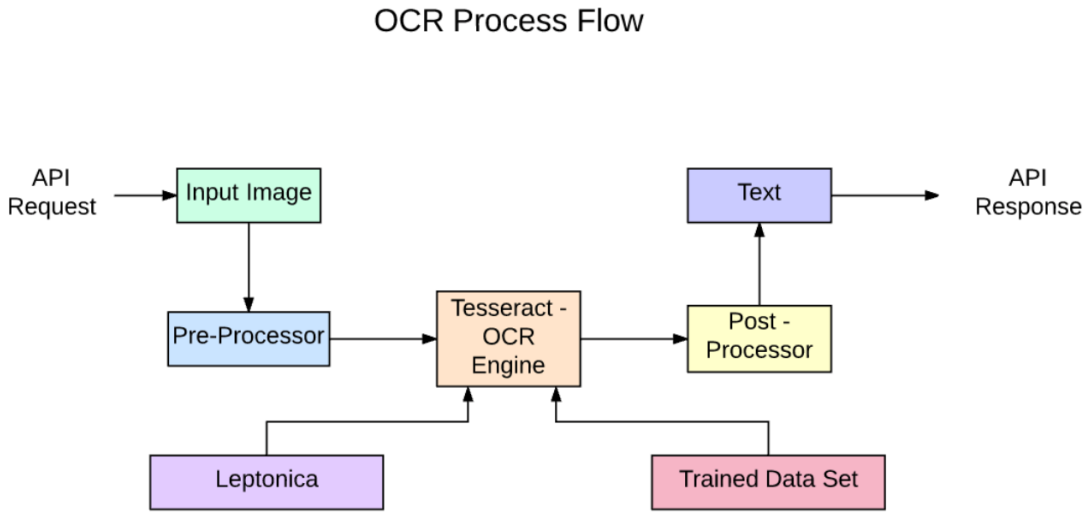

Рисунок 2.1 – Структура процесу Tesseract OCR

Tesseract 4.00 включає нову підсистему нейронної мережі, налаштовану як розпізнавач текстових рядків. Він бере свій початок у реалізації LSTM на основі Python OCRopus, але був перероблений для Tesseract у C++. Система нейронних мереж Tesseract передує TensorFlow, але сумісна з ним, оскільки існує мова опису мережі під назвою Variable Graph Specification Language (VGSL), яка також доступна для TensorFlow.

Щоб розпізнати зображення, що містить один символ, ми зазвичай використовуємо нейронну мережу (CNN). Текст довільної довжини є послідовністю символів, і такі проблеми вирішуються за допомогою RNN, а LSTM — популярна форма RNN.

Технологія - як це працює

LSTM відмінно підходять для вивчення послідовностей, але дуже сповільнюються, коли кількість станів дуже велика. Є емпіричні результати, які показують, що краще попросити LSTM вивчити довгу послідовність, ніж коротку послідовність багатьох класів. Tesseract розроблений на основі моделі OCRopus у Python, яка була відгалуженням LSMT на C++, що називається

CLSTM. CLSTM – це реалізація моделі рекурентної нейронної мережі LSTM на C++ з використанням бібліотеки Eigen для чисельних обчислень.

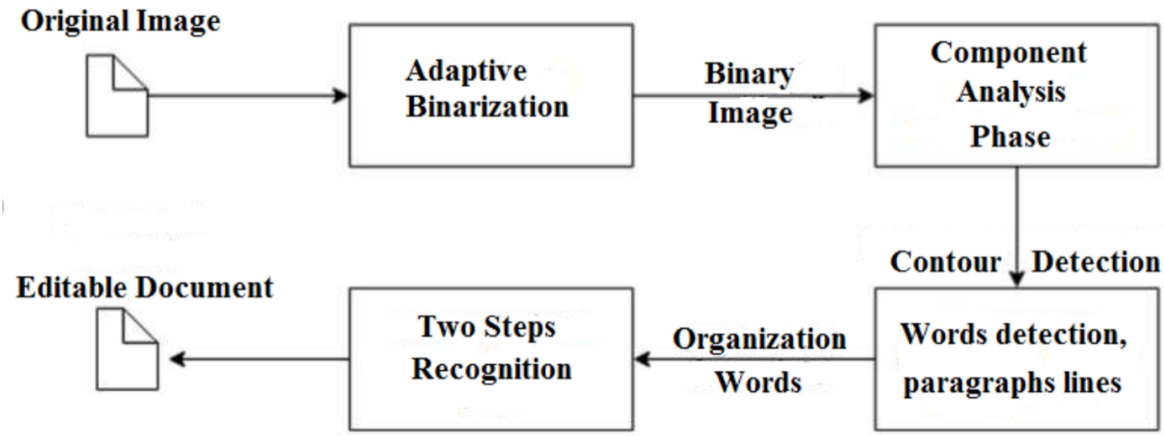

Рисунок 2.2 – Робота Tesseract з бумагою

Legacy Tesseract 3.x залежав від багатоетапного процесу, у якому ми можемо розрізняти кроки:

1) пошук слів;
2) пошук лінії;
3) класифікація персонажів.

Пошук слів виконувався шляхом організації текстових рядків у краплі, а рядки та області аналізувалися на наявність фіксованого кроку чи пропорційного тексту. Рядки тексту розбиваються на слова по-різному, залежно від міжсимвольного інтервалу. Потім розпізнавання протікає як двопрохідний процес. При першому проході робиться спроба розпізнати кожне слово по черзі. Кожне задовільне слово передається адаптивному класифікатору як навчальні дані. Потім адаптивний класифікатор отримує більш точно розпізнавати текст внизу сторінки.

Модернізація інструменту Tesseract була спробою очистити код та додати нову модель LSTM. Вхідне зображення обробляється прямокутниках (прямокутник) рядково, вводячи їх у модель LSTM і видаючи результат. На Рисунку 2.2 можна візуалізувати, як це працює.

EasyOCR

Пакет EasyOCR створено та підтримується компанією Jaided AI, яка спеціалізується на послугах оптичного розпізнавання символів.

EasyOCR реалізований за допомогою Python та бібліотеки PyTorch. Якщо у вас є графічний процесор із підтримкою CUDA, базова бібліотека глибокого навчання PyTorch може значно прискорити виявлення тексту та швидкість розпізнавання тексту.

На момент написання цієї БКР EasyOCR може розпізнавати текст 58 мовами, включаючи англійську, німецьку, хінді, російську та інші! Супроводжуючі EasyOCR планують додати додаткові мови у майбутньому. Ви можете знайти повний список мов, які підтримує EasyOCR на головній сторінці ,

В даний час EasyOCR підтримує лише розпізнавання друкованого тексту. Пізніше 2022 року вони також планують випустити модель розпізнавання рукописного введення.

Є також бібліотеки та технології у сфері OCR для зчитування тексту з фотографії. Не кажучи вже про онлайн-сервіси, які надають подібні послуги платно або безкоштовно. Але мінус у тому, що безкоштовні сервіси відповідної якості, що неприпустимо для цільових користувачів реалізації, описаної в даній бакалаврській роботі.

Комп'ютерний зір

Комп'ютерний зір – це процес, за допомогою якого ми можемо зрозуміти зображення та відео, як вони зберігаються і як ми можемо маніпулювати ними та вилучати з них дані. Комп'ютерний зір є основою або переважно використовується для штучного інтелекту. Computer-Vision відіграє важливу роль у безпілотних автомобілях, робототехніці, а також додатках для корекції фотографій.

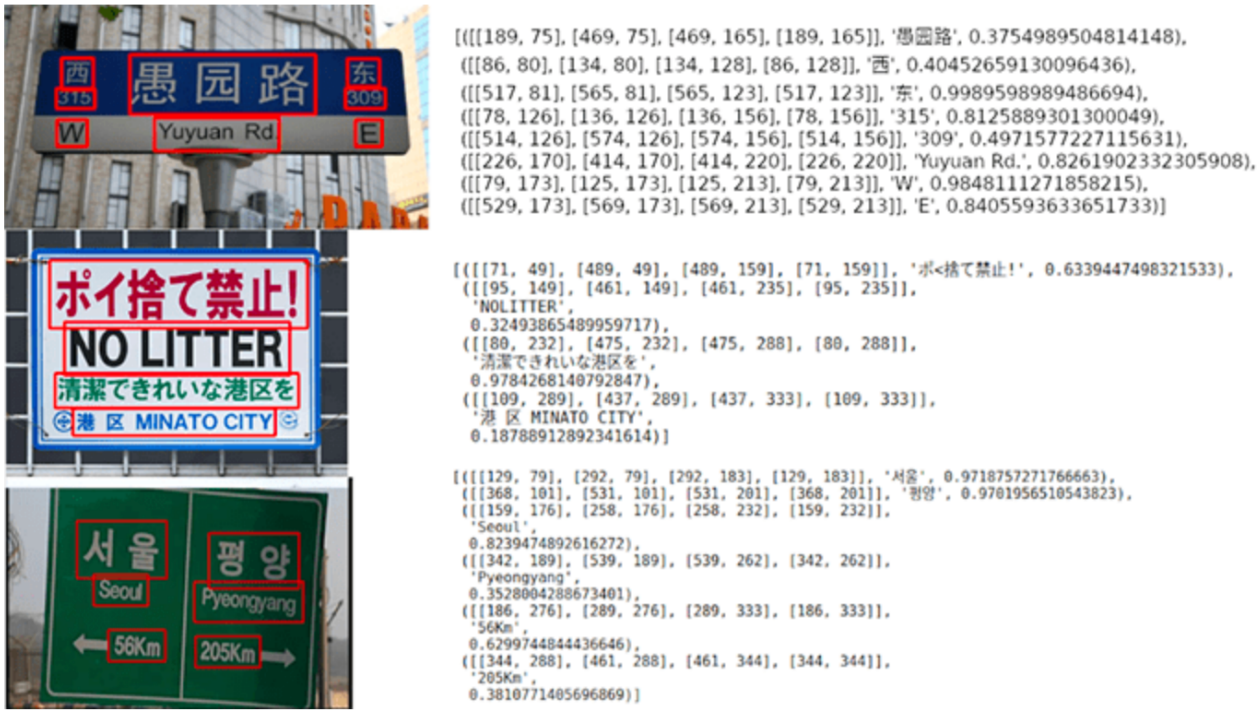

Рисунок 2.3 – можливості OCR технологій

OpenCV

OpenCV – це величезна бібліотека з відкритим вихідним кодом для комп'ютерного зору, машинного навчання та обробки зображень, і тепер вона відіграє важливу роль у роботі в реальному часі, що дуже важливо у сучасних системах. З його допомогою можна обробляти зображення та відео для ідентифікації об'єктів, облич або навіть почерку людини. Без неї неможливо буде вирішити питання, оскільки первинне завдання – просто змусити комп'ютер прочитати зображення, якщо йдеться про камери, які можуть читати документ без його фотографування, а просто піднесення документа до об'єктиву.

Він інтегрований з різними бібліотеками, такими як NumPy, python здатний обробляти структуру OpenCV для аналізу. Щоб визначити шаблон зображення та його різні функції, ми використовуємо векторний простір та виконуємо математичні операції з цими функціями.

Перша версія OpenCV була 1.0. OpenCV випускається під ліцензією BSD і, отже, безкоштовний як академічного, так комерційного використання. Він має інтерфейси C++, C, Python та Java і підтримує Windows, Linux, Mac OS, iOS та Android. При розробці OpenCV основна увага приділялася додаткам реального часу підвищення обчислювальної ефективності. Всі речі написані на оптимізованому C/C++, щоб використати переваги багатоядерної обробки.

Ось результат обробки на зображенні:

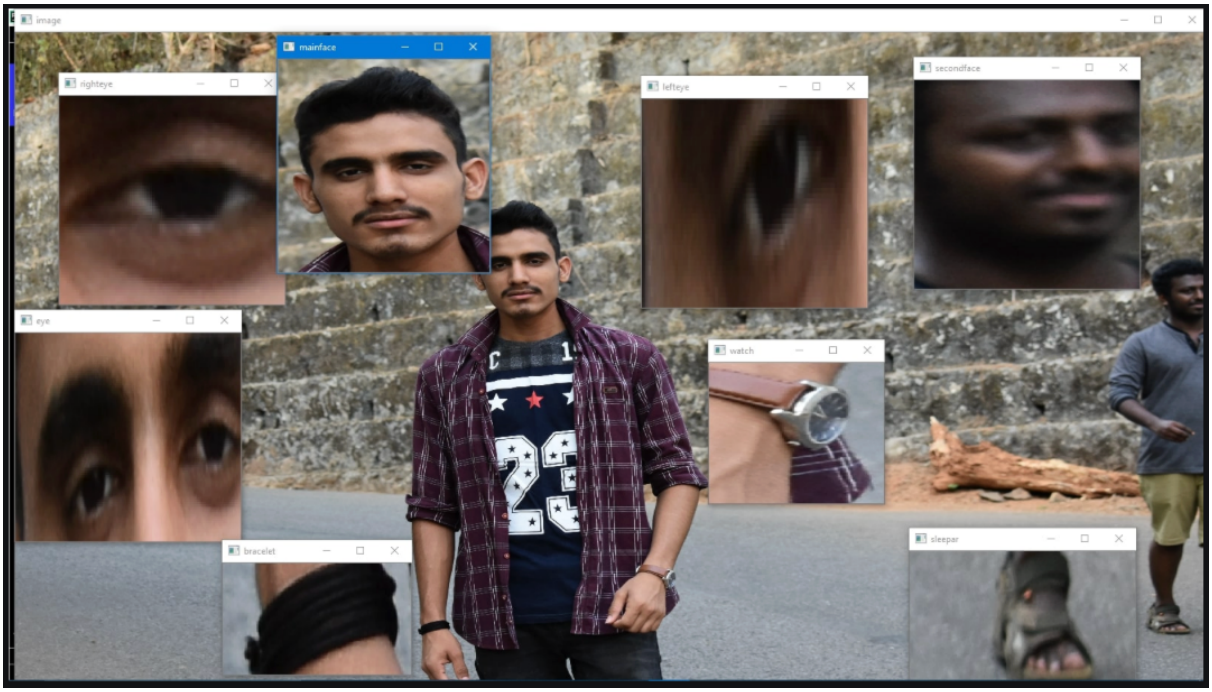

Рисунок 2.4 – фрагментація інформації за допомогою OpenCV

З наведеного вище вихідного зображення можна отримати множину фрагментів інформації, які є у вихідному зображенні. Як і на зображенні вище, доступні дві особи та людина (я) на зображеннях з браслетом, годинником тощо, тому за допомогою OpenCV ми можемо отримати всі ці типи інформації з вихідного зображення.

Програми OpenCV: є багато програм, які вирішуються за допомогою OpenCV, деякі з них перераховані нижче:

1) розпізнавання особи;
2) автоматизований контроль та спостереження;

3) кількість людей – підрахунок (прохідність у торговому центрі тощо);

4) підрахунок транспортних засобів на автомагістралях разом із їх швидкостями;

5) інтерактивні арт-інсталяції;

6) виявлення анамолії (дефектів) у виробничому процесі (непарні браковані вироби);

7) зшивання зображень перегляду вулиць;

8) пошук та вилучення відео/зображень;

9) роботизована навігація та керування автомобілями без водія;

10) розпізнавання об'єктів;

11) аналіз медичних зображень;

12) фільми - 3D-структура з руху;

13) позпізнавання реклами телеканалів;

14) функціональність OpenCV;

15) введення/виведення зображення/відео, обробка, відображення (ядро, imgproc, highgui);

16) виявлення об'єктів/функцій (objdetect, features2d, nonfree);

17) монокулярний або стерео-комп'ютерний зір на основі геометрії (calib3d, зшивка, відеостаб);

18) комп'ютерна фотографія (фото, відео, супердозвіл);

19) машинне навчання та кластеризація (ml, flann);

20) прискорення CUDA (графічний процесор).

Обробка зображень – це метод виконання деяких операцій із зображенням з метою отримання покращеного зображення та/або вилучення з нього деякої корисної інформації.

Якщо говорити про основне визначення обробки зображень, то «Обробка зображень – це аналіз та обробка оцифрованого зображення, особливо з метою покращення його якості».

Цифрове зображення

Зображення може бути визначено як двовимірна функція f(x, y), де x та y – просторові (площинні) координати, а амплітуда жирності будь-якої пари координат (x, y) називається інтенсивністю або рівнем сірого. зображення у цей момент.

Іншими словами, зображення є не що інше, як двомірну матрицю (тривимірну у разі кольорових зображень), яка визначається математичною функцією f(x, y) у будь-якій точці і дає значення пікселя в цій точці зображення. зображення, значення пікселя описує, наскільки яскравим є цей піксель і якого кольору він має бути.

Обробка зображень – це в основному обробка сигналів, при якій входом є зображення, а виходом – зображення або характеристики відповідно до вимог, пов'язаних із зображенням.

Обробка зображень в основному включає наступні три кроки:

1) імпорт зображення;

2) аналіз та обробка зображення;

3) висновок, у якому результат може бути змінено, зображення або звіт, заснований на аналізі зображення.

Висновки до розділу 2

З цього розділу випливає, що штучний інтелект і нейронні мережі можуть незабаром торкнутися багатьох сфер діяльності, починаючи з самоврядних транспортних засобів, закінчуючи такими побутовими речами, як розпізнавання музики, голоси і так далі. Також були описані технології, такі як EasyOCR і OpenCV, які доступні для будь-якого користувача. Завдяки цьому, якщо поставити собі за мету, можна самостійно реалізувати багато дивовижних речей, які ще не перебувають у ходу і лише починають впроваджуватися в житті кожної людини.

Йдеться також про такі речі як:

- бот для автоматизації повідомлень у месенджерах;
- голосові команди для комп'ютера, смартфона або навіть для RGB-світильника і так далі.

Цей список можна заповнювати до безкінечності. І це не якесь далеке майбутнє. Це те, що може зробити, за бажання, будь-яка людина, яка захоплюється програмуванням та вивченням нейронних мереж.

Без сумніву, технологія комп'ютерного зору та розпізнавання символів, предметів тощо – зовсім нова епоха комп'ютеризації, яка потребує численних покращень та удосконалень. Взяти, наприклад, штучний інтелект, який вигадав свою власну мову спілкування, або погрози від того самого інтелекту знищити людство.

Тому звернено увагу на те, про що говорять письменники-фантасти. Це все може здійснитися, якщо не бути пильним та обережним щодо автоматизації за рахунок навчання штучних інтелектів та нейронних мереж.

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Негнучкість використання бібліотеки Tesseract у моїх умовах

Обрано для виконання своєї БКР EasyOCR пакет. Незважаючи на те, що всі пакети та бібліотеки, в які входять у функціонал розпізнавання тексту, мають свої переваги та недоліки, все одно вигідно зупинитися саме на цьому пакеті.

Насамперед спочатку остаточний вибір був на Tesseract. Але його головний недолік у тому, що його неможливо сьогодні використовувати для операційних систем MacOS, як у даному випадку. Робочій ноутбук – єдиний девайс, операційна система якого саме така.

Пов'язано це з тим, що Tesseract необхідно завантажити в змінну шляху PATH, що найлегше зробити для ОС Windows – ця система в таких випадках є більш гнучкою. Система безпеки в MacOS не дозволяє завантажувати open-source ресурс з Інтернету в дорогу зі змінною PATH. Як результат, при імпортуванні бібліотеки Tesseract в цей проект, в консоль виводиться помилка про те, що Tesseract не завантажений або не перебуває в дорозі зі змінною PATH.

```
File "/home/appuser/venv/lib/python3.8/site-packages/streamlit/script_runner.py", line
    exec(code, module.__dict__)
File "/app/ocr_python/main.py", line 75, in <module>
    ocr.inicial()
File "/app/ocr_python/main.py", line 28, in inicial
    self.texto = self.extrair_texto(img)
File "/app/ocr_python/main.py", line 38, in extrair_texto
    texto = pytesseract.image_to_string(img, lang="por")
File "/home/appuser/venv/lib/python3.8/site-packages/pytesseract/pytesseract.py", line
    return {
File "/home/appuser/venv/lib/python3.8/site-packages/pytesseract/pytesseract.py", line
    Output.STRING: lambda: run_and_get_output(*args),
File "/home/appuser/venv/lib/python3.8/site-packages/pytesseract/pytesseract.py", line
    run_tesseract(**kwargs)
File "/home/appuser/venv/lib/python3.8/site-packages/pytesseract/pytesseract.py", line
    raise TesseractNotFoundError()
```

Рисунок 3.1 – скріншот помилки

Причина такої низки перешкод на шляху встановлення пакета на ОС MacOS може також полягати в тому, що на момент звернення до головного сайту Tesseract кореневе посилання на середовище розробників та програмістів GitHub було втрачено або перенесено з ініціативи розробника. Цей факт також доводить те, що в інструкції з установки Tesseract на MacOS використовуються некоректні та неактуальні команди та функції, що робить процес завантаження та встановлення неможливим за моїх умов.

Тому заради збереження продуктивності робочого ноутбука було вирішено використати саме бібліотеку EasyOCR для реалізації безпосередньо завдання та проблеми моєї БКР.

Якщо поставити рішення цього питання ґрунтовно, то можна знайти способи її вирішення. Але це не зовсім доцільно, тому що насамперед спосіб вирішення – завантаження бібліотеки Tesseract шляхом розпаковування .exe файлу, тим часом коли файл розширення .pkg, який підтримує операційну систему MacOS, – просто не передбачено.

Це виявилося навіть на краще. Невигідно використовувати пакет Tesseract, принаймні якщо питання стосується проблеми розпізнавання тексту з фотографії та відеокамери.

Підсумувавши, необхідно тезисами описати основний недолік Tesseract:

1) з моменту березня 2022 посилання GitHub на пакет перенесено або видалено;

2) відповідно, неможливо використати бібліотеку на MacOS;

3) якість роботи пакета, яке буде досліджено у наступному пункті.

У наступному пункті проведено порівняльний аналіз кількох бібліотек, включаючи Tesseract та EasyOCR.Там порівнено якість роботи цих бібліотек на практиці.

3.2 Порівняльний аналіз та дослідження кількох окремих пакетів

На ринку є такі платні сервіси для оптичного розпізнавання об'єктів і символів, як Amazon Textract, когнітивний сервіс Microsoft, хмарний зір Google і так далі. Одночасно з цим, ми в цьому пункті досліджуємо 3 хороші безкоштовні OCR-бібліотеки з відкритим вихідним кодом – Pytesseract, EasyOCR та Keras-OCR. Вони також дали хороші результати, аналогічні до інших платних API-сервісів. У цій статті ми побачимо, як їх настроювати, використовувати та як вони працюють у різних випадках використання.

Рисунок 3.2 – блоки зображень за замовчуванням

Keras-Ocr:

Дана бібліотека підтримує у своїй функції Pipeline() URL-посилання як тип вихідних даних, що підтримується, що набагато полегшує процес зчитування. У реалізації програми використовується папка з фото з Github у вигляді url-посилань одного закордонного програміста, які є свого роду тестом: чим більше зображень розпізнає та чи інша бібліотека, тим вища її якість. До речі, така можливість є і у пакетів EasyOCR та Tesseract.

За замовчуванням результати зчитування представляються нам у вигляді блоку зображень, які накладаються на вихідне зображення, що зчитується, як на Рисунку 3.2:

Код програми з поточної бібліотеки міститься в Додатку А.

Але використовуючи фрагмент коду нижче, можна вивести результат у текстовий документ, щоб працювати з отриманими даними, попередньо переконавшись, що нейронна мережа розпізнала і обробила вхідні дані коректно:

```
x_max = 0
temp_str = ""
myfile = open("my_file.txt", "a+")
for i in  prediction_groups[0]:
    x_max_local = i[1][: 0].max()
    if x_max_local > x_max:
        x_max = x_max_local
        temp_str = temp_str + " " + i[0].ljust(15)
    else:
        x_max = 0
        temp_str = temp_str + "\n"
        myfile.write(temp_str)
        print(temp_str)
        temp_str = ""
```

myfile.close()

Tesseract:

Слід почати з того, що на малюнку нижче буде фотографія, яка буде зчитувати Tesseract і EasyOCR:

Рисунок 3.3 – одне із зображень для тестування пакетів

Найпростіший код програми наведено нижче:

```
import pytesseract
import cv2

img1 = cv2.imread('8.jpg')
cv2.imshow('sample', img1)
cv2.waitkey(0)
cv2.destroyAllWindows()
text1 = pytesseract.image_to_string(img1)
print(text1)
```

```
eS
\ — =
= HR260K8337/m F
```

Рисунок 3.4 – скріншот результату роботи програми на основі пакету Tesseract

Немає жодних сумнівів, що велика перевага даного пакета в тому, що його функціонал вміщується в дуже невелику кількість функцій та рядків. Але, порівнюючи результат роботи та вихідне зображення, якість роботи бібліотеки іноді залишає бажати кращого.

EasyOCR:

Як випливає з назви, EasyOCR це проста легка бібліотека для порівняння з іншими. Вона підтримує багато мов. Крім того, шляхом налаштування різних гіперпараметрів можна підвищити продуктивність для конкретних випадків використання.

Код та результат роботи останньої досліджуваної бібліотеки наведено нижче:

```
import easyocr
import cv2
import PIL
from PIL import ImageDraw

reader = easyocr.Reader(["ru", "en"])
img = PIL.Image.open("8.jpg")
img

bound = reader.readtext("8.jpg")
bound
```

```
[([[72, 92], [214, 92], [214, 124], [72, 124]],
  'HRZ6DK8337',
  0.4664633224303828)]
```

Рисунок 3.5 – скріншот результату роботи програми на основі пакету EasyOCR

Висновок:

Прогноз OCR залежить не тільки від моделі, але і від багатьох інших факторів, таких як чіткість, відтінки сірого зображення, гіперпараметр, задана вага і т.д.

Tesseract добре працює із зображеннями високої роздільної здатності. Деякі морфологічні операції, такі як дилатація, ерозія, бінаризація OTSU, можуть допомогти збільшити продуктивність Tesseract.

EasyOCR – це полегшена модель, що забезпечує хорошу продуктивність при конвертації квитанцій або PDF. Це дає більш точні результати з організованими текстами, такими як PDF, квитанції, рахунки.

Keras-OCR є інструментом розпізнавання зображень для конкретних зображень. Якщо текст знаходиться всередині зображення, а їх шрифти та кольори не організовані, Keras-ocr дає добрі результати.

Хоча жорстких правил не існує, можна враховувати три вищезгадані моменти при виборі інструменту OCR.

З результатами роботи всіх трьох бібліотек можна ознайомитись, перейшовши до Додатку Б.

3.3 Дослідження бібліотеки EasyOCR

Бібліотека оптичного розпізнавання символів EasyOCR зчитує короткі тексти (наприклад, серійні номери, номери деталей та дати).

Він використовує файли шрифтів (попередньо визначені шрифти OCR-A, OCR-B і напівстандартні шрифти або інші шрифти) з алгоритмом зіставлення шаблонів, який може розпізнавати навіть погано надруковані, зламані або з'єднані символи будь-якого розміру.

Є 4 етапи для розпізнавання символів:

Рисунок 3.6 – 4 етапи на шляху до потрібного результату

У наведеній нижче схемі робочого процесу бібліотеки EasyOCR ілюструються поетапно кроки, які необхідно зробити пакету, щоб розпізнати текст:

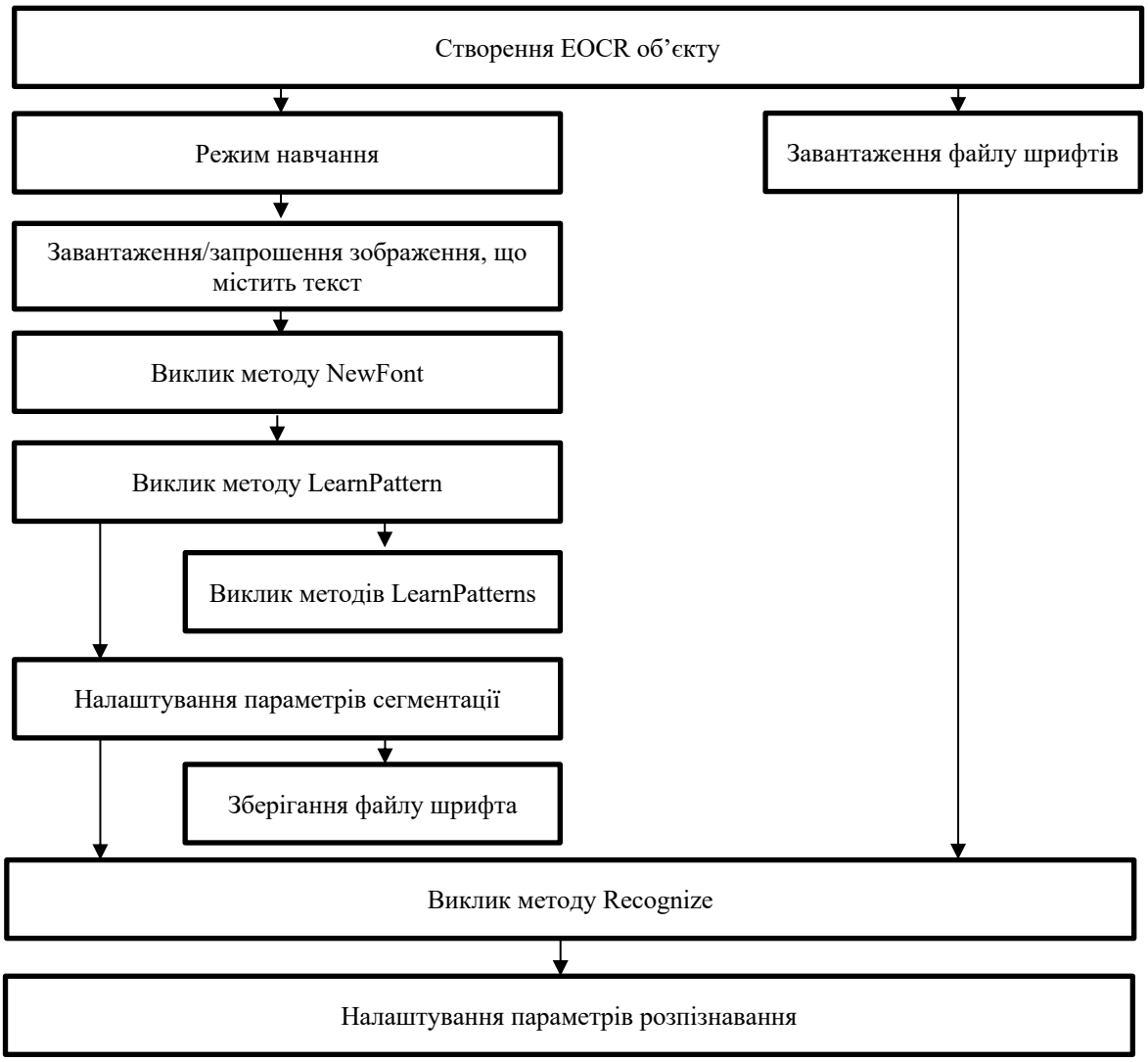

Рисунок 3.7 – робочий процес бібліотеки EasyOCR

Нижче наведена спроба певною мірою описати методи, функції та процеси, які виробляє бібліотека; спроба описати принцип її роботи, спираючись і ґрунтуючись на малюнок 3.7.

Процес вивчення

Якщо це необхідно, можна вивчити символи для створення файлу шрифту.

Символи є один за одним EasyOCR, який аналізує їх і створює базу даних символів, що називається шрифтом. Кожен символ має числовий код (зазвичай код ASCII) і належить до класу символів (який може використовуватися в процесі розпізнавання).

Файли шрифтів створюються так:

NewFont очищає поточний шрифт.

LearnPattern або LearnPatterns додає шаблони з вихідного зображення до шрифту.

Шаблони впорядковані за значенням їх індексу, призначеним процесом FindAllChars.

Шаблони у шрифті зберігаються у вигляді невеликого масиву пікселів, за замовчуванням 5 пікселів у ширину та 9 пікселів у висоту. Цей розмір можна змінити перед навчанням за допомогою параметрів PatternWidth та PatternHeight.

RemovePattern видаляє небажані шаблони (необов'язково).

Save записує вміст шрифту у файл на диску зі значеннями параметрів: NoiseArea, MaxCharWidth, MaxCharHeight, MinCharWidth, MinCharHeight, CharSpacing, TextColor.

Сегментація

EasyOCR аналізує краплі, щоб знайти символи та їх обмежувальну рамку, використовуючи один із двох режимів сегментації:

1) режим збереження об'єктів: одна, свого роду, крапля відповідає одному символу.

2) режим «переклеювання» об'єктів: кілька крапель групуються на символи номінального розміру. Це корисно, коли символи зламані або складаються з кількох частин. Якщо великий двійковий об'єкт занадто великий, щоб його можна було вважати одним символом, його можна розділити автоматично за допомогою CutLargeChars.

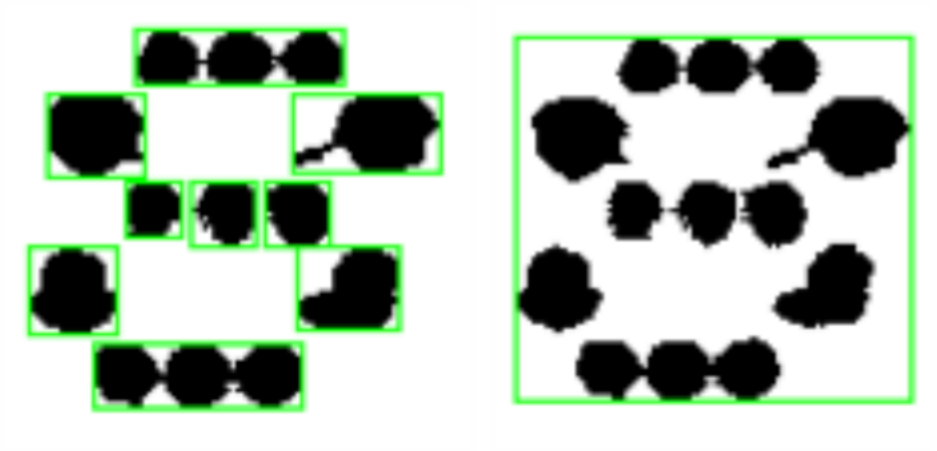

Рисунок 3.8 – Фільтри видаляють дуже великі та дуже маленькі небажані функції.

Потім EasyOCR обробляє зображення символу, нормалізуючи розмір до рамки, що обмежує, витягує відповідні функції і зберігає їх у файлі шрифту. Шаблони у шрифті зберігаються у вигляді масивів пікселів, визначених PatternWidth та PatternHeight (за замовчуванням 5 пікселів завширшки та 9 пікселів у висоту).

Параметри сегментації

Параметри сегментації повинні бути однаковими під час навчання та розпізнавання. Хороша сегментація покращує розпізнавання.

Параметр Threshold допомагає відокремити текст від тла.

При надто великому значенні чорні символи на білому тлі стають густішими, що може призвести до їх злиття, при надто маленькому значенні деталі зникають.

Якщо умови освітлення дуже мінливі, хорошим вибором буде автоматичне встановлення порога.

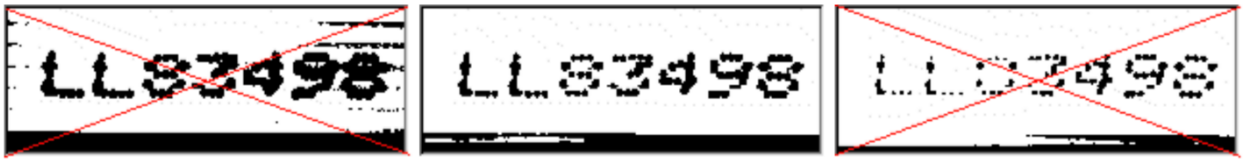

Рисунок 3.9 – Занадто високе граничне значення (ліворуч); регулювання порога (посередині); занадто низьке граничне значення (праворуч)

NoiseArea: області великих двійкових об'єктів, менші від цього значення, відкидаються. Переконайтеся, що дрібні деталі символів збережені (наприклад, точка над літерою «і»).

MaxCharWidth, MaxCharHeight: максимальна величина символу. Якщо крапля не міститься у прямокутник з цими розмірами, вона відкидається або розділяється на кілька частин за допомогою вертикальних ліній розрізу. Якщо кілька великих двійкових об'єктів розміщуються у прямокутнику з такими розмірами, вони групуються разом.

MinCharWidth, MinCharHeight: Мінімальний розмір символу. Якщо BLOB-об'єкт або група BLOB-об'єктів розміщуються у прямокутнику з такими розмірами, він відкидається.

CharSpacing: ширина найменшого проміжку між сусідніми літерами. Якщо він більше, ніж MaxCharWidth, він не діє.

Якщо проміжок між двома символами ширший від цього, вони розглядаються як різні символи. Це запобігає неправильному групуванню тонких символів.

RemoveBorder: не можна використовувати краплі поряд з краями зображення/області інтересу для розпізнавання символів. За замовчуванням вони відкидаються.

Розпізнавання

Символи порівнюються з набором шаблонів, які називаються шрифтом. Символ розпізнається шляхом знаходження найкращої відповідності між символом та малюнком у шрифті. Після того, як символ був знайдений, його розмір нормалізується (розтягується, щоб помістится в певний прямокутник) для порівняння. Нормалізований символ порівнюється з кожним нормалізованим шаблоном базі даних шрифтів, і повертаються найкращі збіги.

1) Load: зчитує попередньо записаний шрифт із файлу на диску.

2) BuildObjects: зображення сегментоване на об'єкти чи краплі (з'єднані компоненти), які допомагають знайти символи. Цей крок можна пропустити, якщо відоме точне положення символів. Якщо процес ізоляції символів упущено, необхідно вказати відомі розташування символів: AddChar і EmptyChars.

3) FindAllChars: вибирає об'єкти, що розглядаються як символи, і сортує їх зверху вниз, а потім зліва направо.

4) ReadText: виконує зіставлення та фільтрує символи, якщо структура маркування фіксована або було надано фільтр набору символів.

Розпізнавання символів: символи порівнюються із набором шаблонів, які називаються шрифтом.

Найкращий збіг розтягується, щоб помістится в заданий прямокутник і порівнюється з кожним нормалізованим шаблоном в базі даних шрифтів.

Фільтр набору символів може підвищити надійність розпізнавання та час виконання за рахунок обмеження діапазону порівнюваних символів. Наприклад, якщо маркування завжди складається з двох великих літер, за якими слідують п'ять цифр, остання з яких завжди парна, можна призначити кожному символу клас (максимум 32 класу), а потім налаштувати фільтр символів, щоб дозволити наступні класи під час розпізнавання: дві великі , чотири парні чи непарні цифри, одна парна цифра.

Кроки 2–4 можна повторювати за бажанням для обробки інших

зображень чи областей інтересу. Можна також використовувати метод Recognize.

Додаткову інформацію, таку як геометричне положення виявлених символів, можна отримати за допомогою CharGetOrgX, CharGetOrgY, CharGetWidth, CharGetHeight і так далі.

CompareAspectRatio робить порівняння символів та шрифтів чутливим до різниці між вузькими та широкими символами. Це покращує розпізнавання, коли символи схожі друг на друга після нормалізації розміру.

Параметри розпізнавання

MaxCharWidth, MaxCharHeight: якщо великий двійковий об'єкт не поміщається у прямокутник із цими розмірами, він не розглядається як можливий символ (надто великий) і відкидається. Більш того, якщо прямокутник з такими розмірами містить кілька BLOB-об'єктів, вони групуються разом, утворюючи єдиний символ. Розмір зовнішнього прямокутника слід вибирати таким, щоб він міг вмістити найбільший символ із шрифту, збільшений із невеликим запасом міцності.

MinCharWidth, MinCharHeight: якщо великий двійковий об'єкт або група двійкових об'єктів поміщаються у прямокутник із цими розмірами, він не розглядається як можливий символ (надто маленький) та відкидається.Розмір внутрішнього прямокутника слід вибирати таким, щоб він містив найменший символ шрифту, стиснутий із невеликим запасом міцності.

RemoveNarrowOrFlat: маленькі символи відкидаються, якщо вони вузькі або плоскі. За замовчуванням вони відкидаються, якщо вони одночасно вузькі та плоскі.

CharSpacing: якщо два BLOB-об'єкти розділені вертикальним проміжком, що перевищує це значення, вони вважаються різними символами. Ця функція корисна, щоб уникнути групування тонких символів, які помістилися у зовнішньому прямокутнику. Його значення має дорівнювати ширині найменшого проміжку між сусідніми літерами. Якщо для нього

встановлено велике значення (більше, ніж MaxCharWidth), це не діє

CutLargeChars: коли великий двійковий об'єкт або група великих двійкових об'єктів більше, ніж MaxCharWidth, вони відкидаються. Якщо цей параметр увімкнено, великий двійковий об'єкт розбивається на стільки частин, скільки необхідно для його розміщення, а кількість порожнього простору, що має бути вставлена між розділеними великими двійковими об'єктами, визначається параметром RelativeSpacing. Це спроба розділити зворушливі персонажі.

RelativeSpacing: коли увімкнено режим CutLargeChars, встановлення цього значення дозволяє вказати кількість пробілів, яка повинна бути вставлена між розділеними частинами великих двійкових об'єктів.

Рисунок 3.10 – Неправильні налаштування розпізнавання

Розширені налаштування

Ці опції розпізнавання можна налаштувати для оптимізації розпізнавання:

CompareAspectRatio: коли цей параметр увімкнено, EasyOCR менш терпимий до розміру та враховує виміряне співвідношення сторін. Використання цього режиму покращує розпізнавання, коли символи виглядають однаково після нормалізації розміру, оскільки він посилює різницю між вузькими та широкими символами.

Фільтрування символів (у методі ReadText) можна використовувати, якщо структура маркування фіксована.

Коли об'єкти більші, ніж властивість MaxCharWidth, їх можна розділити на потрібну кількість частин за допомогою вертикальних ліній розрізу.

ESegmentationMode, режим ізоляції символів, визначає спосіб ізоляції символів:

Зберігати режим об'єктів: персонаж – це крапля; не робиться жодних спроб групувати великі двійкові об'єкти, тому пошкоджені символи не можуть бути оброблені, а дрібні елементи, такі як акценти та точки, можуть бути відкинуті за критерієм мінімального розміру символу.

Режим повторної вставки об'єктів: краплі групуються в окремі символи, якщо вони відповідають максимальному розміру символів і не розділені вертикальним зазором, що дозволяє зберегти акценти та точки.

Висновки до розділу 3

У цьому розділі було досліджено бібліотеку EasuOCR, її функціонал, суть роботи тощо. Виходячи з дослідницької роботи в пункті 3.3 стало в загальних рисах ясно, який алгоритм роботи використовується для розробки даної бібліотеки та інших, які використовують оптичне розпізнавання даних.

Також стало ясно, що не всі бібліотеки у вільному доступі здатні коректно виконувати поставлене завдання. Тому було проведено безпосередньо порівняльний аналіз для того, щоб обрати найкращу бібліотеку для вирішення проблеми БКР.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ

4.1 Дослідження пакета EasyOCR

Імпортується бібліотека за допомогою команди:

import easyocr

Далі потрібно повідомити EasyOCR, якою мовою треба читати. EasyOCR може читати кілька мов одночасно, але вони мають бути сумісні один з одним. Англійська сумісна з усіма мовами. Мови, які розділяють більшість символів (наприклад, латинський алфавіт) один з одним, сумісні.

Найбільш неочікувана фотографія:

Рисунок 4.1 – Фотографія, яку буде протестовано

Отже, потрібно читати традиційною китайською та англійською мовами. Розміщено коди мов у форматі списку і передано його як перший аргумент для об'єкта Reader. У разі це ['ch_tra', 'en']:

reader = easyocr.Reader(['ch_tra', 'en'])

Потім EasyOCR перевірить, чи є у користувача потрібні файли моделі, і завантажить їх автоматично. Потім буде завантажено модель на згадку, що може зайняти кілька секунд залежно від вашого обладнання. Після того, як це

буде зроблено, можна прочитати стільки зображень скільки потрібно, не запускаючи цей рядок знову.

Крім списку коду мови, можна передати кілька необов'язкових аргументів. Першим є gpu, який за умовчанням має значення True, що означає, що EasyOCR спробує використовувати графічний процесор (GPU) у обчисленнях, якщо це можливо. Якщо не треба використовувати GPU, можна написати gpu=False. Для користувача з кількома графічними процесорами можна вказати тут, який з них є потреба використати, наприклад, gpu='cuda:0'. Ще один необов'язковий аргумент – model_storage_directory. Використовується для встановлення місця, де EasyOCR зберігає файли моделей. Якщо не вказано, це буде ~/.EasyOCR/model.

Щоб отримати текст із зображення, просто треба передати шлях до зображення у функцію читання тексту, як показано нижче:

```
result = reader.readtext('chinese_tra.jpg')
```

Для отримання тексту із зображення передано шлях до зображення у функцію читання тексту, як показано нижче:

Стандартний висновок має формат списку, кожен елемент є списками координат текстового поля [x, y], тексту та рівня достовірності моделі відповідно:

```
[([[448, 111], [917, 111], [917, 243], [448, 243]],'高鐵左營站',0.9247),
([[454, 214], [629, 214], [629, 290], [454, 290]], 'HSR', 0.9931),
([[664, 222], [925, 222], [925, 302], [664, 302]],'Station',0.3260),
([[312, 306], [937, 306], [937, 445], [312, 445]],'汽車臨停接送區',0.7417),
([[482, 418], [633, 418], [633, 494], [482, 494]],'Kiss',0.9577),
([[331, 421], [453, 421], [453, 487], [331, 487]], 'Car', 0.9630),
([[653, 429], [769, 429], [769, 495], [653, 495]], 'and', 0.9243),
([[797, 429], [939, 429], [939, 497], [797, 497]],'Ride',0.6400)]
```

Рисунок 4.2. Результат зчитування Рисунка 4.1

Результат упорядкований зверху донизу. Як видно, це може не відповідати природному людському читанню, але є можливість автоматично комбінувати ці слова в кінці цього уроку.

Замість шляху до файлу chinese_tra.jpg передано зображення у вигляді масиву numpy (з opencv),

```
img = cv2.imread('chinese_tra.jpg')
result = reader.readtext(img)
```

або URL-адреси зображення:

```
result = reader.readtext('https://www.somewebsite.com/chinese_tra.jpg')
```

Також нижче є приклад із використанням хінді. У цій фотографії, як описувалося раніше, дані можна виводити у форму шарів зображень поверх вихідного, що може бути дуже зручно для відео реєстраторів:

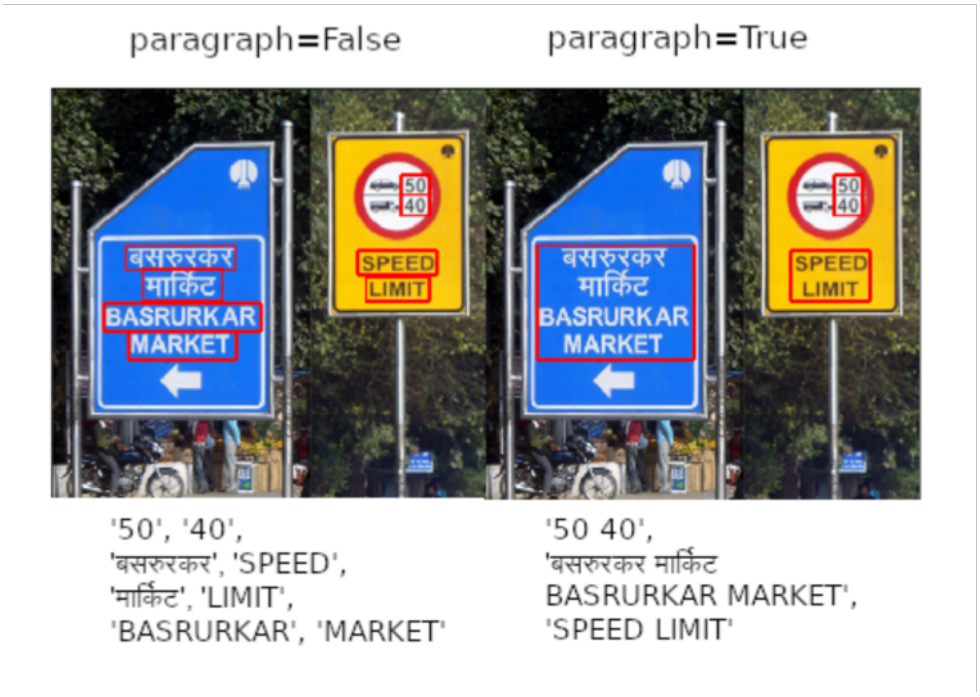

Рисунок 4.3 – Демонстрація здатності пакету до розпізнавання хінді та деванагарі

Ця бібліотека підтримує багато основних мов, також мов, які менш поширені, зокрема азіатські та східні. Навчання відбувається за рахунок баз даних мов, записаних у форматі .pth. Ці фоли розміром понад 200 мБ кожна. Тому ця бібліотека досить добре навчена.

Саме коли оголошено змінну як об'єкт з назвою Reader, цим самим навчається нейронна мережа. Вона отримує в параметрах посилання базу даних мови, залежно, який необхідний користувачеві.

Нижче наведено частину бази даних, які підтримує EasyOCR:

Supported Languages

Language	Code Name
Abaza	abq
Adyghe	ady
Afrikaans	af
Angika	ang
Arabic	ar
Assamese	as
Avar	ava
Azerbaijani	az
Belarusian	be

Рисунок 4.4. Які мови підтримує EasyOCR (серед мов є українська)

4.2 Опис програмної реалізації

У коді, з яким можна ознайомитися в Додатку, можна помітити, що він не настільки великий за обсягом, що робить його дуже зручним для експлуатації.

Дана програмна реалізація вирішує поставлене завдання в БКР - нейромережева система для розпізнавання документів та введення даних.

Нижче можна побачити робоче середовище розробки:

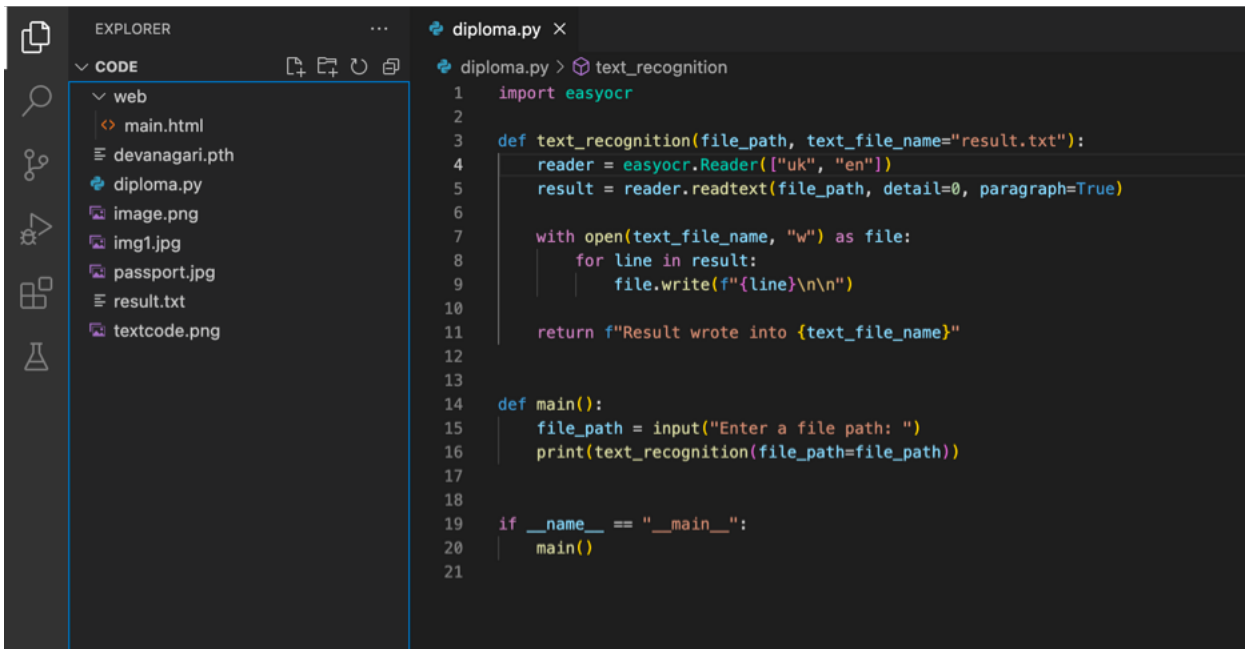

Рисунок 4.5 – Робоче середовище

Програма містить у собі тільки одну функцію. Її назва – text_recognition.

Фрагмент коду нижче:

```
with open(text_file_name, "w") as file:
    for line in result:
    file.write(f"{line}\n\n")
```

Тут за допомогою циклу реалізується виведення зчитаної інформації зі змінної result у текстовий файл result.txt, з яким потів можна працювати далі для робочих питань, залежачи від користувача.

Далі для прикладу фото паспорту автора БКР, яке програма повинна зчитати:

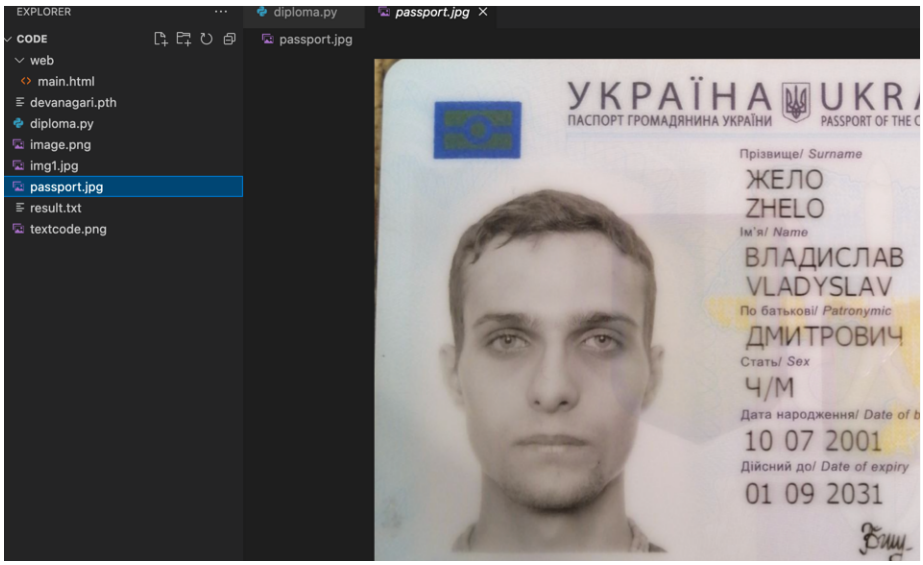

Рисунок 4.6 – Зображення паспорту у кореневій папці програми

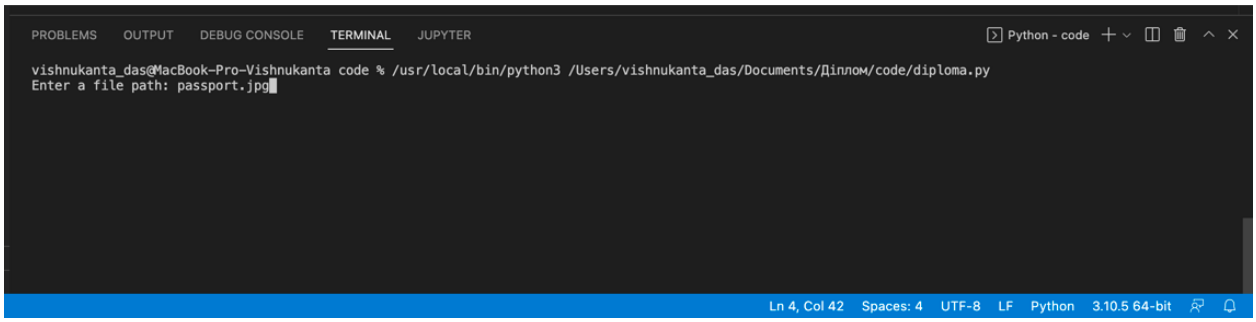

Рисунок 4.7 – Зображення паспорту у кореневій папці програми

За малюнком 4.7 можна зазначити, що потрібно ввести ім’я зображення. Після цього програма почне зчитування.

Закінчення роботи програми – це наступний текст у консолі:

CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU

Result wrote into result.txt

Відкриємо файл result.txt та оцінимо роботу програми:

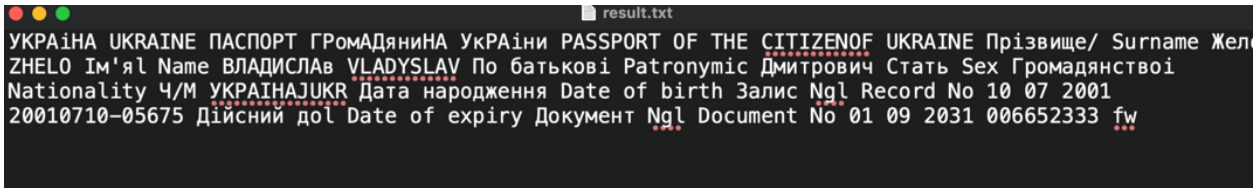

Рисунок 4.8 – Результат роботи у файлі result.txt

Можна оцінити роботу програми таким чином: незважаючи на неідеальну якість фотографії, яка була зроблена звичайним смартфоном, який не має високої якості камери, можна сказати, що результат близький до ідеального.

Єдині помилки та недоліки:

1) це некоректний регістр ліченого тексту
2) сприйняття знаків слеша "/" як англійську "l".

4.3 Керівництво користувача

Все дуже просто. Згідно Рисунка 4.9 треба завантажити через кнопку «Выбрать файл» потрібне зображення формату .png, .jpg, .tiff:

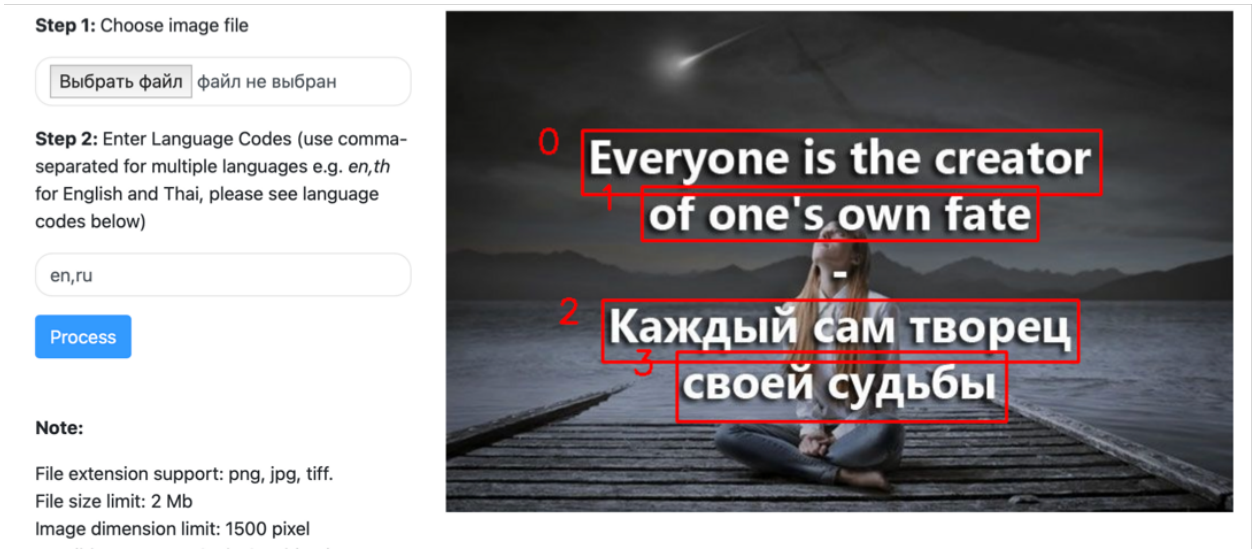

Рисунок 4.9 – Зовнішній вигляд web-додатку

Після чого користувач може приймати роботу у вигляді таблиці:

Result

No.	Text	Confident Score
0	Everyone is the creator	0.2118
1	of one's own fate	0.4273
2	Каждый сам творец	0.3934
3	своей судьбы	0.6172

Рисунок 4.10 – Результат роботи

Демоверсія, яка була продемонстрована вище – це задум, який був реалізований до 24.02.2022. Єдині дані, які залишилися – це наведені в приклад скріншоти.

Планувалося реалізувати повноцінний desktop-додаток, який міг би задовольнити абсолютно всі потреби потенційного користувача: не тільки зчитування з фотографії або зі скана, а також друга вкладка з можливістю використовувати відеопристрій, починаючи з веб-камери, закінчуючи спеціальними робочими камерами зчитування, призначені для запису документів.

Складність реалізації подібної ідеї полягає в тому, що фізично не вистачає часу та можливості зайнятися самостійним вивченням Django або інших технологій, які дозволяють код, написаний мовою Python, перенести до графічного інтерфейсу. Це займає величезний обсяг часу та праці, особливо якщо раніше з подібною технологією ніколи не перетинатися.

Є проблеми, які ще не вирішені. Один з них:

1) фільтрація та підпис тексту, наприклад: «серія, номер паспорта», «ПІБ» тощо. Але це залежить безпосередньо від замовника цього продукту;

2) необхідний зручніший і адаптивніший інтерфейс програми;

3) складність реалізації програми спільно з відеопристроями, які мають закуповуватися у тих чи інших компаніях;

4) більш якісне навчання нейронної мережі, адже є помилки в результатах роботи, які потрібно виправляти тільки вручну і періодично перевіряти ще раз.

Висновки до розділу 4

Можна зробити висновок, що дана бібліотека досить легка в реалізації: у кількох рядках коду можна навчити нейронну мережу, просто підключивши БД до змінної, залежно від того, якою мовою текст необхідно рахувати (у даному випадку – українська та англійська, якщо питання стосується паспортів). Також підтримуються рукописні тексти. Але це також залежить від читабельності того чи іншого почерку.

Незважаючи на погану якість зображення6 нейронна мережа успішно впоралася із завданням, лише трохи переплутала регістр символів і знак слеша «/».

Основне завдання, над яким необхідно працювати в майбутньому – розробка десктопного програмного забезпечення для того, щоб у користувача не виникало технічних питань під час експлуатації продукту.

Спеціальний розділ

ОХОРОНА ПРАЦІ

до кваліфікаційної роботи

на тему:

НЕЙРОМЕРЕЖЕВА СИСТЕМА ДЛЯ РОЗПІЗНАВАННЯ ДОКУМЕНТІВ ТА ВВЕДЕННЯ ДАНИХ

Спеціальність 122 «Комп’ютерні науки»

122 – БКР – 401.21810109

Виконав студент 4-го курсу, групи 401

_____*Жело В. Д.*

(підпис, ініціали та прізвище)

«___» _____ 2022 р.

Консультант, старший викладач

(наук. ступінь, вчене звання)

_____*Макарова О. В.*

(підпис, ініціали та прізвище)

«___» _____ 2022 р.

Миколаїв – 2022

5 ОХОРОНА ПРАЦІ

5.1 Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Основним законодавчим документом є Закон України «Про охорону праці», який визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні.

Тема дипломної роботи - нейромережева система для розпізнавання документів та введення даних. Виконується аналіз умов приміщення, в якому виконувалась робота та організації, в якій будуть використовуватися результати дипломної роботи. Це робиться для того, щоб уникнути неприємних ситуацій зі службами цивільної безпеки.

Основними шкідливими факторами на робочому місці користувача ПК є: електромагнітні випромінювання, ризик ураження електричним струмом; підвищений або знижений рівень освітлення; несприятливі параметри мікроклімату; напруга зору, уваги; фізичні перевантаження через тривале перебування у фіксованій позі.

Основний шкідливий фактор для організації, в якій будуть використовуватися результати дипломної роботи - це некоректне використовування обладнання та програми безпосередньо. Справа в тому, що при роботі з цією програмою використовується камера для зчитування документу. Ґрунтуючись на документах про права людини слід уникнути

порушення прав на конфіденційність, випадково без потреби фіксувати свої власні документи або власне обличчя або обличчя колег.

5.2 Характеристика приміщення

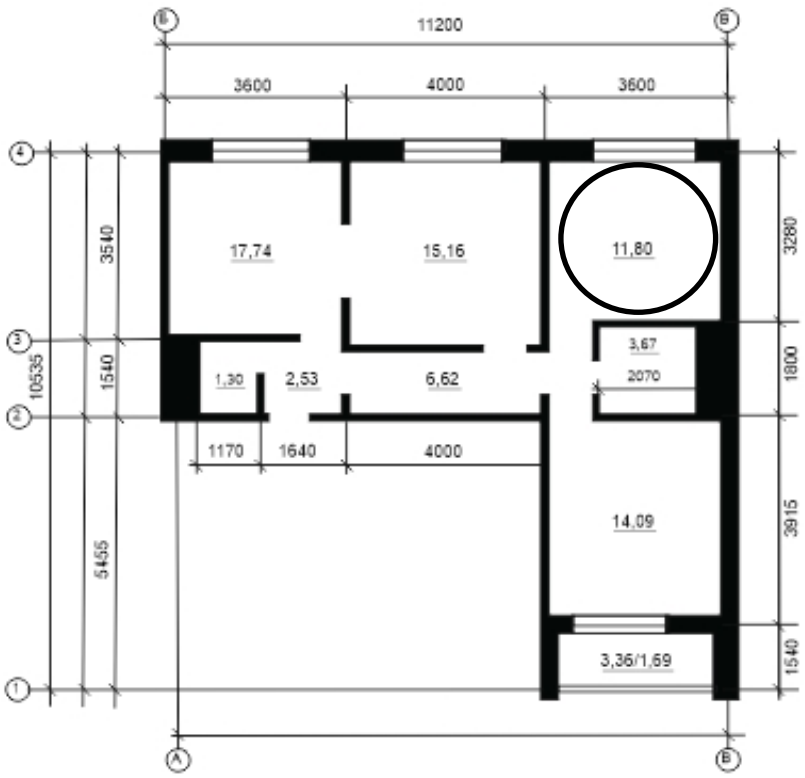

Рисунок 5.1 – План приміщення до 24.02.2022

Розробка програми велася у приміщенні, план якого приведено на рисунку 4.1.

Приміщення має одностороннє природне освітлення та загальне штучне освітлення.

Ширина приміщення 3,2 м, довжина – 3,6 м, висота стелі – 2,7 м. Кількість робочих місць – одне. Приміщення знаходиться на четвертому поверсі п’ятиповерхової будівлі. Площа – 11,80 м^2, об’єм – 31,86 м^3.

Виходячи з цього отримано дані наведені в таблиці 1. Нормативні значення згідно з Таблиці 5.1:

Таблиця 5.1 – Площа робочого приміщення

Параметр	Норма	Фактичні параметри
Площа, S	Не менше ніж 6 м^2	11,80 м^2
Об’єм, V	Не менше ніж 20 м^3	31,86 м^3

Показники відповідають вимогам нормативних документів.

5.3 Мікрокліматичні умови

Оптимальні та фактичні параметри мікроклімату наведені у таблиці 5.2 про нормативні значення мікроклімату.

Таблиця 5.2 – Нормативні значення мікроклімату

Період року	Параметр	Оптимальний	Фактичний
Теплий	Температура	23 – 25	28
	Вологість	60 – 40	50
	Швидкість повітря	< 0,1 м/с	
Холодний	Температура	22 – 24	22
	Вологість	60 – 40	40
	Швидкість повітря	< 0,1 м/с	

Значення відносної вологості повітря в холодний період року знаходиться на межі допустимих значень, доцільно використовувати в цей період зволожувачі повітря.

Температура повітря в теплий період року виходить за межі допустимих значень, для забезпечення вимог [17] необхідно встановлення кондиціонера

5.4 Освітлення

Норми освітленості регламентуються у «ДСТУ EN 12464-1:2016 Світло та освітлення. Освітлення робочих місць. Частина 1. Внутрішні робочі місця».

Згідно з робота відноситься до розряду IIIв. При загальному освітленні показник освітлення робочої поверхні має бути у межах 200 – 400 лк.

У даному приміщенні використовується природне, штучне та змішане освітлення.

Джерелом природного освітлення є вікно шириною 1,7 м і висотою 1,8 м.

Джерелом штучного освітлення є світильник з п’ятьма світлодіодними лампами потужністю 18 Вт, світловий потік 1800 лм.

Для визначення освітленості робочої зони скористаємось методом світлового потоку.

Формула світлового потоку має вигляд:

$$\Phi = \frac{EkSZ}{N\eta} \quad (1)$$

де Ф – світловий потік, Лм;

E – освітленість робочого місця, Лк;

k – коефіцієнт запасу, що враховує зменшення світлового потоку ламп у процесі експлуатації, $k = 1{,}2$;

Z – коефіцієнт нерівномірності, $Z = 1{,}1$;

S – площа приміщення, м^2;

N – кількість ламп;

η – коефіцієнт використання світлового потоку.

Звідси освітленість на робочому місці дорівнює:

$$E = \frac{\Phi N\eta}{kSZ} \quad (2)$$

Для визначення коефіцієнту використання світлового потоку η потрібно розрахувати індекс приміщення i за формулою 3:

$$i = \frac{S}{h\,(A + B)} \quad (3)$$

де S – площа приміщення, $S = 11{,}80$ м^2;

h – висота світильників над робочою поверхнею, м;

A - ширина приміщення, A = 2,7 м;

B - довжина приміщення, B = 3,6 м.

Висота світильників над робочою поверхнею знаходиться за формулою 4:

$$h = H - h_{\text{св}} - h_{\text{рп}} \quad (4)$$

де Н – висота приміщення, м;

$h_{\text{св}}$ – висота світильника, м;

$h_{\text{рп}}$ – висота робочої поверхні, м.

$$h = 2{,}7 - 0{,}3 - 0{,}8 = 1{,}6 \text{ м}$$

$$i = \frac{8{,}75}{1{,}6\ (2{,}7+3{,}6)} = 0{,}91$$

За індексом приміщення та коефіцієнтами світлового потоку від стелі – 70%, стін – 50%, підлоги – 30 % визначаємо значення коефіцієнту використання світлового потоку η = 0,5.

Підставимо всі значення у формулу 2 для визначення освітленості:

$$E = \frac{1800 \cdot 5 \cdot 0{,}51}{1{,}2 \cdot 8{,}75 \cdot 1{,}1} = 397 \text{ Лк}$$

Освітленість на робочому місці становить 397 лк, що відповідає вимогам для IIIв розряду зорових робіт.

5.5 Рівень шуму на робочому місці

Джерелом шуму в приміщенні є комп’ютер. Згідно технічній документації шум кулера у блоці живлення має рівень 10-15 дБ, кулера процесора – 10-15 дБ, загальний рівень шуму комп’ютера 25-30 дБ. Беручи до уваги незначний рівень шуму інших компонентів комп’ютера та незначний рівень фонового шуму іншого обладнання, сумарний рівень звукового забруднення у приміщені не перевищує 50 дБ.

Згідно з [103] допустимий рівень звуку на робочому місці має бути не вище ніж 50 дБ. Отже рівень звукового забруднення не перевищує норму.

Може бути перевищено рівень шуму з причини шумних сусідів, крику дітей надворі, шуму близьких поруч тощо. Тому робоче місце може бути змінено, але є ризик, що умови на тимчасовому робочому місці не будуть відповідати [17], [18], [19], [20].

5.6 Випромінювання

В приміщенні відсутні джерела інфрачервоного, ультрафіолетового та електромагнітного випромінювання. В моніторі комп'ютера використовується рідкокристалічна матриця з світлодіодною підсвіткою, що не створює значного електромагнітного випромінювання.

До того ж в роботі використовуються захисні окуляри проти шкідливого випромінювання. Ще періодичний відпочинок для очей може поліпшити стан очей.

5.7 Ергономіка робочого місця

Робоче місце працівника з ПЕОМ має відповідати вимогам [17].

Висота робочої поверхні стола має бути у межах 680-800 мм (фактичний розмір 750 мм), рекомендована ширина стола 600-1400 мм (фактичний розмір 1500 мм), рекомендована глибина 800-1000 мм (фактично 700 мм). Простір для ніг заввишки не менше ніж 600 мм (фактично 700 мм), завширшки не менше ніж 500 мм (фактично 700 мм), завглибшки не менше ніж 650 мм (фактично 700 мм). Крісло є підйомно-поворотним, має підлокітники та можливість регулювання за висотою, кутом нахилу сидіння та спинки.

Таким чином за ергономічними показниками робоче місце відповідає нормативним вимогам.

5.8 Ризики для виробничої організації

Будь яка організація чи послуга (паспортний стіл, громадська контора тощо), яка має справу з документами громадян України, може користуватися результатом БКР.

Тому слід зазначити, що за документом [19], не допускається збирання, зберігання, використання і поширення конфіденційної інформації про особу без її згоди, крім випадків, визначених законом, і лише в інтересах національної безпеки, економічного добробуту та прав людини.

З цього випливає, що фотографії, які можуть зберігатись в організації, не повинні використовуватися в особливих цілях. Також камери для фіксації документу на місці не повинні використовуватися в цілях, окремих від робочого процесу компанії.

Висновки до спеціального розділу «Охорона праці»

Аналіз умов праці в розглянутому робочому приміщенні показав, що умови праці з ПЕОМ відповідають вимогам [17, 18, 19, 20], оскільки площа та об’єм не менше нормативних значень, рівні шуму, вібрації і загазованості не перевищують нормативних значень.

Рівень освітлення робочого місця відповідає нормам. В приміщенні відсутні джерела інфрачервоного, ультрафіолетового та електромагнітного випромінювання. В моніторі комп’ютера використовується рідкокристалічна матриця з світлодіодною підсвіткою, що не створює значного електромагнітного випромінювання.

До того ж в роботі використовуються захисні окуляри проти шкідливого випромінювання. Ще періодичний відпочинок для очей може поліпшити стан очей.

Для підтримання параметрів мікроклімату в приміщенні встановлено радіатор водяної системи центрального опалення, що складається з 12 секцій.

Ергономіка робочого місця і режим зорової роботи задовольняють вимогам і сприяють зниженню втоми.

Рівень шуму не перевищує норму та не заважає спокійній мозкової праці.

Усі ризики враховані

Щодо умов роботи у надзвичайних обставинах, то, на жаль, не вдається виконувати всі вимоги охорони праці, а саме стан робочого місця. Регулярно умови робочого місця змінюватимуться і незважаючи на підтримку застосовного стану вологості, освітлення, немає можливості підтримувати психічний та моральний стан здоров'я під час виконання БКР.

ВИСНОВКИ

У ході виконання БКР було реалізовано декілька бібліотек, що підтримують оптичне розпізнавання даних. Було проведено порівняльний аналіз безкоштовних пакетів із відкритим вихідним кодом. Було затверджено, що бібліотека EasyOCR задовольняє бажання потенційних користувачів, які використовуватимуть продукт, який був продемонстрований у роботі. Незважаючи на погану якість фотографії, програма досить успішно провела зчитування тексту з фотографії паспорта.

Було запропоновано демоверсію веб-додатку, яка дозволяє інтуїтивно просто використовувати у своїх цілях можливості бібліотеки. Є цілі, більш досконало вивчивши середовище розробки Django та інші, вдосконалити веб-додаток, зробивши аналогічне ПЗ.

В першому розділі ясно, що проблематика, яка була зачеплена, вкрай необхідна сьогодні і також у наступних роках, коли починається епоха комп'ютеризації абсолютно всіх сфер життя та діяльності. Немає жодних сумнівів, що подібні продукти будуть розвиватися все більше і більше, ІІ буде все більш досконалим і мало чим відрізнятиметься від роботи та розумового процесу людини.

В другому розділі було описання сфери навчання нейронних мереж та ШІ. З цього розділу випливає, що ШІ і нейронні мережі можуть незабаром торкнутися багатьох сфер діяльності, починаючи з самоврядних транспортних засобів, закінчуючи такими побутовими речами, як розпізнавання музики, голоси і так далі. Також були описані технології, такі як EasyOCR і OpenCV, які доступні для будь-якого користувача. Завдяки цьому, якщо поставити собі за мету, можна самостійно реалізувати багато дивовижних речей, які ще не перебувають у ході і лише починають впроваджуватися в житті кожної людини.

У третьому розділі також стало зрозуміло, що не всі бібліотеки у вільному доступі здатні коректно виконувати поставлене завдання. Тому було проведено

безпосередньо порівняльний аналіз для того, щоб обрати найкращу бібліотеку для вирішення проблеми БКР.

У четвертому розділі було показано принцип роботи пакету EasyOCR. Також було показано продемонстровано реалізацію завдання БКР. Була освячена основна проблема, над якою необхідно працювати у майбутньому – адаптація інтерфейсу майбутньої desktop-програми під потреби того чи іншого користувача та замовника.

Під час виконання БКР було навчено працювати в екстремальних умовах, вирішувати питання та завдання, з якими раніше не доводилося стикатися. Окрім технічних навичок, які доводилося наскільки можна вдосконалювати під час виконання роботи, були розвинені такі якості, як відповідальність, терпіння та посидючість. Завдяки вимогам, що були описані в охороні праці, проблему посидючості та терпіння було вирішено успішно, оскільки умови, в яких виконувалася БКР, сприяла підтримці фізичного та ментального здоров'я.

Виходячи з цього, тема БКР була висвітлена досконало з багатьох сторін. Важливість вирішення проблеми цієї роботи також було доведено.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tony Yiu, Understanding Neural Networks [Електронний ресурс] / Towards Data Science — Режим доступу: https://towardsdatascience.com/understanding-neural-networks- 19020b758230. — Дата доступу: 20.02.2022.

2. D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/pdf/1409.0473.pdf. — Дата доступу: 20.02.2022.

3. Michael Nielsen, Neural Networks and Deep Learning [Електронний ресурс] / Neural Networks and Deep Learning — Режим доступу: http://neuralnetworksanddeeplearning.com/index.html. — Дата доступу: 12.02.2022.

4. S. Hochreiter and J. Schmidhuber, Long short-term memory. Neural computation [Електронний ресурс] / Bioinf At. — Режим доступу: https://www.bioinf.jku.at/publications/older/2604.pdf. — Дата доступу: 04.03.2022.

5. T. Tieleman and G. Hinton, Divide the gradient by a running average of its recent magnitude. Neural computation [Електронний ресурс] / Coursera. — Режим доступу: https://www.cs.toronto.edu/lecture_slides_lec6.pdf. — Дата доступу: 14.05.2022.

6. L. Gaunt, M. Brockschmidt, A probabilistic programming language for program induction [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/pdf/1612.00817.pdf. — Дата доступу: 14.05.2022.

7. Goodfellow, J. Pouget-Abadie, Generative adversarial nets [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/pdf/1406.2661.pdf. — Дата доступу: 14.05.2022.

8. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition [Електронний ресурс] / Arxiv Org — Режим доступу: https://arxiv.org/pdf/1409.1556.pdf. — Дата доступу: 24.05.2022.

9. Karpathy and L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions [Електронний ресурс] / Stanford Edu. — Режим доступу: https://cs.stanford.edu/people/karpathy/cvpr2015.pdf. — Дата доступу: 14.05.2022.

10. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/pdf/1409.1556.pdf. — Дата доступу: 28.05.2022.

11. W. Zaremba, I. Sutskever, Recurrent neural network regularization [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/pdf/1409.2329.pdf. — Дата доступу: 28.05.2022.

12. H. Zhang, T. Xu, Text to photo-realistic image synthesis with stacked generative adversarial networks [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/abs/1612.03242. — Дата доступу: 28.05.2022.

13. L. Gaunt, A probabilistic programming language for program induction [Електронний ресурс] / Arxiv Org. — Режим доступу: https://arxiv.org/abs/1608.04428. — Дата доступу: 28.05.2022.

14. Eisenman, B. Learning React [Text] / Bonnie Eisenman. — 2nd edition. — Sebastopol: O'Reilly Media, 2017. — 242 p.

15. Kirupa, Understanding WebViews [Електронний ресурс] / Kirupa. — Режим доступу: https://www.kirupa.com/apps/webview.htm. — Дата доступу: 28.05.2022.

16. Optical Character Recognition (OCR) — Image, Opencv, pytesseract and easyocr [Електронний ресурс] — Режим доступу: https://medium.com/@nan-dacoumar/optical-character-recognition-ocr-image-opencv-pytesseract-and-easyocr-62603ca4357 — Дата доступу: 11.06.20.

17. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. – К.: Постанова Головного державного санітарного лікаря України, 1998. - № 7.

18. Санітарні норми мікроклімату виробничих приміщень : ДСН 3.3.6.042-99. – К., 2000.- С.16

19. Природне і штучне освітлення : ДБН В.2.5-28:2006. – К. : Міністерство будівництва, архітектури та житлово-комунального господарства України, 2006. – 68 с. – (Національні стандарти України).

20. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. Державні санітарні норми. ДСН 3.3.6.037-99. – К., 1999.

21. Конституція України від 28.06.1996 р. [статті 32 (ч. 1, 2), 34].

ДОДАТОК А
Лістинг коду з використанням пакету Keras-Ocr

```
import keras_ocr
from matplotlib import pyplot as plt
pipeline = keras_ocr.pipeline.Pipeline()
images = [
        keras_ocr.tools.read(url) for url  in [
                'https://raw.githubusercontent.com/Thangasami/OCR-
/main/number/1.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
/main/number/2.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
        /main/number/3.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
        /main/number/4.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
        /main/number/5.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
        /main/number/6.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
        /main/number/7.jpg',
                'https://raw.githubusercontent.com/Thangasami/OCR-
        /main/number/8.jpg',
                ]]
prediction_groups = pipeline.recognize(images)
```

ДОДАТОК Б
Результати роботи трьох бібліотек

Таблиця 3.1 – результати роботи Tesseract, Keras-OCR та EasyOCR

Дані	Вихідні дані	Tesseract	Keras-OCR	EasyOCR
Номерний знак високої якості	HR26DK8337	HR260K8337	HR26DK8337	HRZ6DK8337
Номерний знак низької якості	MH14GN9239	Пусто	MHL4GN9239	9239
Рукопис низької якості	AMIT ASHISH	Пусто	ADIT ASHISH	AdIT ASHISH
Рукопис високої якості	LAKSHMINIVAS TOURIST HOME	LAKSHMINIVAS TOURIST HOME	LAKSHMINIVAS TOURIST HOME	LAKSHMINIVAS TOURIST HOME
Фото с текстом високої якості	Albert Einstein	Albert Einstein	Albert Einstein	Albert Einstein

Закінчення таблиці 3.1

Дані	Вихідні дані	Tesseract	Keras-OCR	EasyOCR
Фото с текстом високої якості	Kotak Mahindra Bank	Kotak Mahindra Bank	Kotak Mahindra Bank	Kotak Mahindra Bank
Рецепт доктора высокого качества	Order #19866	Order #19866	Order #119666	Order #19866
Рецепт доктора низької якості	Amoxicillin 500mg	Пусто	Amoxicillin 500mg	Amoxicillin 500mg

ДОДАТОК В
Лістинг коду з використанням пакету EasyOCR, що вирішує питання за завданням БКР

```
import easyocr

def text_recognition(file_path, text_file_name="result.txt"):
    reader = easyocr.Reader(["ru", "en"])
    result = reader.readtext(file_path, detail=0, paragraph=True)
    with open(text_file_name, "w") as file:
        for line in result:
            file.write(f"{line}\n\n")
    return f"Result wrote into {text_file_name}"
def main():
    file_path = input("Enter a file path: ")
    print(text_recognition(file_path=file_path))



if __name__ == "__main__":
    main()
```