**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені Ігоря Сікорського»**

**Факультет прикладної математики**

**Кафедра прикладної математики**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Р. Чертов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

з напряму підготовки 6.040301 «Прикладна математика»

на тему: «Математичне та програмне забезпечення для обробки й класифікації архіву новинних відеорепортажів».

Виконав: студент IV курсу, групи КМ-71

|  |  |  |
| --- | --- | --- |
|  | Лисий Павло Олексійович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Керівник | Старший викладач  Бай Ю.П. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Консультант зі спеціальних питань | -  - | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Консультант із нормоконтролю | старший викладач Мальчиков В. В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Рецензент | - | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Засвідчую, що в цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.

Студент Лисий Павло Олексійович

Київ — 2021**Національний технічний університет україни**

**«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — перший (бакалаврський)

Напрям підготовки 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Р. Чертов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Лисому Павлу Олексійовичу

1. Тема роботи: «Математичне та програмне забезпечення для обробки й класифікації архіву новинних відеорепортажів»,

керівник роботи Бай Юлія Петрівна, страший викладач.

затверджені наказом по університету від «28» травня 2016 р. № 000-С.

2. Термін подання студентом роботи: «15» червня 2016 р.

3. Вихідні дані до роботи: розроблювана система повинна працювати з відеопотоком.

4. Зміст роботи: виконати аналіз існуючих методів розв’язання задачі, вибрати метод розпізнавання звуку за зображення з відеофайлу, спроектувати автоматизовану систему парсингу та класифікації новин, здійснити програмну реалізацію розробленої системи, провести тестування розробленої системи.

5. Перелік ілюстративного матеріалу: ілюстрації, блок-схеми розроблених алгоритмів, схема взаємодії модулів системи, знімки екранних форм, приклади роботи програми.

6. Дата видачі завдання: «15» квітня 2016 р.

Календарний план

| № з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів роботи | Примітка |
| --- | --- | --- | --- |
| 1 | Огляд та порівняння існуючих методів | 12.11.2015 |  |
| 2 | Проектування алгоритму | 14.12.2015 |  |
| 3 | Тестування алгоритму | 24.12.2015 |  |
| 4 | Підготовка розділу:  «ПОСТАНОВКА ЗАДАЧІ» | 01.02.2016 |  |
| 5 | Підготовка розділу:  «АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗВʼЯЗАННЯ ЗАДАЧІ» | 01.03.2016 |  |
| 6 | Підготовка розділу:  «ОПИС МАТЕМАТИЧНОЇ МОДЕЛІ» | 15.03.2016 |  |
| 7 | Підготовка розділу:  «ОПИС ПРОГРАМНИХ ЗАСОБІВ» | 05.04.2016 |  |
| 8 | Підготовка розділу: «ЕКСПЕРЕМЕНТАЛЬНІ РЕЗУЛЬТАТИ» | 15.04.2016 |  |
| 9 | Підготовка розділу: «ВИСНОВКИ» | 03.05.2016 |  |
| 10 | Оформлення пояснювальної записки | 01.06.2016 |  |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Лисий П.О. |
| Керівник роботи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Бай Ю.П. | |

Анотація

Дипломну роботу виконано на \_\_ аркушах, вона містить \_\_ додатки та перелік посилань на використані джерела з \_\_ найменувань. У роботі наведено \_\_ рисунки та \_\_ таблиці.

Метою даної дипломної роботи є створення математичного та програмного забезпечення для обробки й класифікації архіву новинних відеорепортажів. Протягом даної роботи розглядається процес оцифрування новинних сюжетів з використанням методів computer-vision(cv) та natural language processing(nlp). Під оцифруванням новинних сюжетів мається на увазі: визначення start-end time кожного окремо сюжету, хронометражу сюжету, виділення назви сюжету, виділення тексту сюжету, виділення типу (новини, анонс та реклама), визначення журналістів та гостей, визначення ведучих, визначення тегів та класифікація сюжета(позитивний/негативний).

У роботі проведено аналіз існуючих рішень указаної задачі — nlp та cv. Виконано їх порівняння з погляду точності отримуваних розв’язків, ефективності алгоритмів та пристосованості методів до використання нечітких даних.

Розроблено автоматизовану систему, що реалізує обраний метод. Виконано тестування розробленої системи.

Ключові слова: nlp, cv, медіа сюжети, новини.

ABSTRACT

Thesis is performed on \_\_ sheets, it contains \_\_ appendices and a list of references to the sources used with \_\_ names. The paper presents \_\_ figures and \_\_ tables.

The purpose of this thesis is to create mathematical and software for processing and classification of the archive of news videos. During this work, the process of digitization of news stories using computer-vision (cv) and natural language processing (nlp) methods is considered. Digitization of news stories means: algorithmic definition of start-end time of each plot, timing of the plot, selection of the title of the plot, selection of the text of the plot, selection of type (news, announcement and advertising), definition of journalists and guests, definition of presenters, definition of tags and plot classification (positive / negative).

The analysis of the existing solutions of the specified problem - nlp and cv is carried out in the work. They are compared in terms of accuracy of the obtained solutions, efficiency of algorithms and adaptability of methods to the use of fuzzy data.

An automated system that implements the selected method has been developed. The developed system has been tested.

Key words: nlp, cv, media stories, news.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів

NLP — natural language processing.

CV — computer vision.

CNN – convolution neural network

start-end time — час початку і кінця, починаючи відлік від початку сюжета, кожного окремого відеосюжету з новинного репортажу.

Fps - frames per second

Вступ

У наш час інформаційні технології досягли значного розвитку й продовжують розвиватися. Разом із тим, підвищується актуальність задач розпізнавання тексту та звуку.

Однією з таких задач є задача класифікації новинних репортажів на телеканалі «Україна». Новинні канали виконують класифікацію та оцифрування новиниих сюжетів вручну або за допомогою підрядників, проте цю задачу можна виконати завдяки штучному інтелекту.

Розроблена система могла б забезпечити економію ресурсів працівників та звільнення людей від рутинної роботи. Також, програмний код допомагає уникнути людських помилок.

На сьогоднішній день розроблено багато алгоритмів аналізу тексту, звуку на відеопотоку. Тому, використовуючи ці напрацювання, можна створити алгоритм та розробити систему для аналізу і класифікації новинних репортажів на телеканалі «Україна». Проблема є актуальною, адже такої системи немає у використанні ні однієї медіа компанії України.

У даній роботі проаналізовано доступні математичні методи та програмні рішення для розв’язання поставлену задачу.

# Постановка задачі

Метою даної дипломної роботи є створення математичного та програмного забезпечення для обробки й класифікації архіву новинних відеорепортажів.

При розробленні відповідного забезпечення потрібно розв’язати наступні завдання:

1. проведення порівняльного аналізу існуючих методів nlp та cv;

вибір та адаптація існуючого методу для вирішення задачі nlp та cv;

розробка програмного забезпечення на базі вибраного математичного методу;

тестування розробленої автоматизованої системи.

Реалізована система має задовольняти такі вимоги:

1. включати визначення start-end time кожного окремо сюжету, визначення хронометражу сюжета, виділення назви сюжета, виділення тексту сюжета, виділення типу (новини, анонс та реклама), визначення журналістів та гостей, визначення ведучих, визначення тегів та класифікація сюжета(позитивний/негативний).
2. мати високі показники ефективності розпізнавання start-end time сюжета(точність повинна бути до 1 секунди, >85% правильності);
3. Записувати таблицю оцифрованих данних для новинного випуску в файл формату Microsoft Excel;

# 2. Аналіз існуючих методів розв'язання задачі

## 2.1 Computer vision (cv)

### Tesseract ORC

Tesseract - є програмою, що розроблялися компанією Hewlett-Packard з середини 1980-х по середину 1990-х років. Потім програма близько 10 років «пролежала на полиці» і в серпні 2006 року її купила Google. Google відкрив вихідний код під ліцензією Apache 2.0 для продовження розробки. На сьогоднішній день бібліотека є найбільш крупним рішенням, якщо вам потрібно обробити дані з будь-якого фото. Оскільки у нас обробка відеофайлу - відеофайл представим як набір кадрів і будемо зчитувати текст з кожного кадра окремо. Основні переваги це висока точність та легкість впровадження у власний проект.

### OCRopus

OCRopus - OCR-система для розпізнавання текстів на базі tesseract. Використовує код для мови моделювання з проекту OpenFST.

В даний час OCRopus використовує тільки інтерфейс командного рядка, приймаючи на вхід зображення з текстом, і виводячи дані в форматі hOCR (відкритий формат на основі HTML). Опції командного рядка дозволяють виконувати окремо конкретні операції (наприклад, розпізнавання одного рядка).

### Основні переваги такі ж як і в Tesseract ORC, проте спосіб OCRopus важчий в впроваджені в код ніж Tesseract ORC, адже Tesseract ORC можна імпортувати як бібліотеку, а OCRopus потрібно вставляти напряму в код.

### CNN

CNN, або згорткові нейронні мережі, можуть бути використані для вирішення задачі розпізнавання тексту з картинки. Основна перевага – можна досягти більш високої точності. Основні недоліки – навчання нейронної мережі займає багато часу, потрібно збирати дуже багато даних, розмічювати їх, вибрати архітектуру та проводити навчання, тому швидше буде використати та підлаштувати напереднавченну модель під свої завдання.

## Voice to text system

Щоб працювати з текстом, для початку необхідно його отримати. Для цього необхідно розпізнати текст зі звуку.

### Google Speech Recognition

### Це бібліотека для розпізнавання мови з підтримкою API, бібліотека працює онлайн та офлайн. Загальна кількість мов, які можна розпізнати завдяки Google Speech Recognition = 119. Бібліотека Google Speech Recognition є найбільш точною та якісною при роботі зі звуком на українській мові. Також, бібліотека непогано справляється з звуковими доріжками, що перевантажені шумом. SpeechRecogntion має низький бар’єр для входу та сумісність із багатьма доступними API розпізнавання мов. (<https://pypi.org/project/SpeechRecognition/>).

### Wav2letter ++

### Wav2letter ++ це швидкий інструментарій для обробки мовлення з відкритим кодом від the Speech у Facebook AI Research, створений для полегшення досліджень у наскрізних моделях розпізнавання мови. Модель працює і зручна у використанні, проте SpeechRecognition краще оброблює шуми на аудіодоріжці. (https://github.com/flashlight/wav2letter).

### CMU Sphinx

CMU Sphinx на ринку понад 20 років досліджень CMU. Деяка перевага цієї бібліотеки: інструменти CMUSphinx розроблені спеціально для малоресурсних платформ, гнучкого дизайну та зосереджені на практичній розробці додатків, а не на дослідженні. Точність визначення тексту менша ніж з використанням SpeechRecognition. (https://cmusphinx.github.io/).

## Natural language processing (nlp)

### Визначення тегів

2.3.1.1 TF-IDM

TF-IDF це term frequency-inverse document frequency. Це простий і зручний спосіб оцінити важливість терміна для будь-якого документа щодо всіх інших документів. Принцип такий - якщо слово зустрічається в будь-якому документі часто, при цьому зустрічаючись рідко у всіх інших документах - це слово має велику значимість для того самого документа. Слова, неважливі для взагалі всіх документів, наприклад, прийменники - отримають дуже низьку вагу TF-IDF (бо часто зустрічаються у всіх документах), а важливі - високий.

2.3.1.2 NLTK

NLTK – бібліотека для python. Найвідоміша NLP бібліотека, створена дослідниками в цій галузі. Вона популярна в академічних колах і в основному використовується для навчання або створення різних методів обробки тексту. Так само вона досить повільна в силу того, що написана повністю на Python і працює з рядками.

2.3.1.3 Spacy

Spacy – бібліотека для python. Вона значно швидше ніж NLTK, так як вона написана на Cython і працює з об'єктами. SpaCy надає в основному кращі інструменти для вирішення конкретного завдання. В цілому, SpaCy з її напереднавченними моделями, швидкістю і зручним API набагато краще підходить для розробників, що створюють готові рішення. Для створення власних моделей ні NLTK, ні Spacy не підходить. Для цього існують Tensorflow, PyTorch та інші.

### Класифікація настрою сюжету

2.3.2.1 SentimentIntensityAnalyzer

Найкращим знайденим рішенням виявилась функція SentimentIntensityAnalyzer з бібліотеки NLTK, яку було визначено вище. (<https://www.nltk.org/howto/sentiment.html>)

(<https://www.nltk.org/api/nltk.sentiment.html>)

Бібліотека дозволяє визначити настрій сюжету по ключовим словам у тексті, результат роботи моделі – число, від 0 до 1, по кожному з наступних параметрів: positive, negative, neutral. Це число означає на скільки відсотків текст позитивний, негативний чи нейтральний.

## Висновки до розділу

## На основі порівняльного аналізу можна зробити висновок, що для задачі computer vision краще використовувати Tesseract ORC. Для задачі voice to text system

### краще використовувати Google Speech Recognition. Для задачі визначення тегів(nlp) можна об'єднати декілька рішень, а саме: TF-IDM, бібліотека на python NLTK та Spacy. Для задачі класифікація настрою сюжету найкращим рішенням є використовувати бібліотеку SentimentIntensityAnalyzer.

# 

# Математичне забезпечення

## Задача парсингу тексту з використанням Tesseract ORC

Початковий етап – це парсинг відео-потоку. Основна ідея – парсити текст та звук в новинному випуску. Почнемо з тексту. В тексті зосереджується ключова інформація, а саме: імена ведучих, назви сюжетів, імена кореспондетів та гостів. Приклад зони, з якої потрібно зчитувати текст зображено на рисунку 1.



Рис.1 – зона, з якої необхідно зчитувати текст.

Оскільки, відео це набір кадрів, тому нам необхідно зчитувати текст з вибраної області певну кількість разів за секунду. Для новинних репортажів fps = 30, тому необхідно отримувати текст 30 разів за секунду. Кожний кадр предсталяє собою 3х-мірний масив даних. Далі цей масив подається на вхід бібліотеки Tesseract ORC, яка виділяє український текст з вибраного кадра. Tesseract - це механізм розпізнавання тексту з відкритим кодом (OCR), доступний за ліцензією Apache 2.0. Поточний офіційний випуск - 4.1.1. Tesseract можна використовувати безпосередньо через командний рядок або за допомогою API для вилучення друкованого тексту із зображень. Він підтримує широкий спектр мов. Tesseract не має вбудованого графічного інтерфейсу, але є кілька доступних на сторінці 3rdParty. За допомогою Tesseract будемо парсити текст з виділеної області та зберігати його в масив, до якого також будем додавати номер кадру в відеофайлі.

## Розпізнавання тексту завдяки Google Speech Recognition

Розпізнавання мови розпочалось з досліджень, проведені в Bell Labs на початку 1950-х. Ранні системи обмежувались одним оратором і мали обмежений словниковий запас близько десятка слів. Сучасні системи розпізнавання мови пройшли довгий шлях від своїх давніх аналогів. Вони можуть розпізнавати мовлення кількох мовців і мають величезний словниковий запас на багатьох мовах. Перший компонент розпізнавання мови - це, звичайно, мова. Мова повинна бути перетворена з фізичного звуку в електричний сигнал за допомогою мікрофона, а потім у цифрові дані за допомогою аналого-цифрового перетворювача. Оцифрувавши кілька моделей, їх можна використовувати для транскрипції звуку в текст. Більшість сучасних систем розпізнавання мовлення покладаються на Hidden Markov Model (HMM). Цей підхід працює на припущенні, що мовленнєвий сигнал при розгляді на досить короткому часовому масштабі може бути розумно апроксимований як стаціонарний процес - тобто процес, в якому статистичні властивості не змінюються з часом. У типовому HMM мовний сигнал розділений на 10-мілісекундні фрагменти. Спектр потужності кожного фрагмента, який по суті є графіком потужності сигналу як функції від частоти, відображається на вектор дійсних чисел, відомий як цепстральні коефіцієнти. Розмір цього вектора, як правило, невеликий - іноді до 10, хоча більш точні системи можуть мати розмір 32 і більше. Кінцевий вихід HMM є послідовністю цих векторів. Для декодування мови в текст групи векторів узгоджуються з однією або кількома фонемами - фундаментальною одиницею мови. Цей розрахунок вимагає підготовки, оскільки звук фонеми різниться залежно від динаміка і навіть різниться від одного до іншого висловлювання того самого динаміка. Потім застосовується спеціальний алгоритм для визначення найбільш вірогідного слова (або слів), що виробляють задану послідовність фонем. У багатьох сучасних системах розпізнавання мови нейронні мережі використовуються для спрощення мовного сигналу з використанням методів перетворення ознак та зменшення розмірності перед розпізнаванням HMM. Детектори голосової активності (VAD) також використовуються для зменшення звукового сигналу лише до тих частин, які можуть містити мову. Це запобігає марнуванню часу декодера на аналіз непотрібних частин сигналу. Документація до Google Speech Recognition наведена у розділі [-----][…].

## Визначення тегів

### TF-IDM

### TF-IDM = TF \* IDF

TF – term frequency

IDF – inverse document frequency

Для розрахунку TF використовуємо наступну формулу:

Рис.2 – формула підрахунку TF

Де, – число входжень слова в документі, а – загальна кількість слів в документі.

Для розрахунку IDF використовуємо формулу, зображену на рисинку 3.

Рис.3 – формула підрахунку IDF

Де, – кількість документів в базі, а – кількість документів, в яких зустрічається слово (при ).

Вибір основи логарифму у формулі не має значення, адже зміна основи призведе до зміни ваги кожного слова на постійний множник, тобто вагове співвідношення залишиться незмінним.

Більшу вагу TF-IDF отримають слова з високою частотою появи в межах документа та низькою частотою вживання в інших документах бази.

### NLTK та Spacy

NLTK та Spacy є напереднатреновані 2 нейромережі, архітектурою яких є CNN.

Завдяки NLTK можна отримати токенезацію текстів, отримати теги, отримати іменовані сутності, відобразити дерево синтаксичного аналізу[1][…]. Більш детально про архітектуру та можливості NLTK можна прочитати в документації [2][…].

Особливості Spacy: Підтримка 64 + мов, 55 навчених пайплайнів для 17 мов, багатозадачне навчання за допомогою попередньо навчених трансформаторів, таких як BERT, швидкість роботи, токенізація з мовною мотивацією, компоненти для розпізнавання іменованих сутностей, позначення частини мови, розбору залежностей, сегментації речень, класифікації тексту , лематизації, морфологічний аналіз, зв’язування сутності тощо. Також Spacy легко розширюється за допомогою спеціальних компонентів(PyTorch, TensorFlow та інші фреймворки). Більш детально про архітектуру та можливості Spacy можна прочитати в документації [3][…].

## SentimentIntensityAnalyzer для визначення настрою сюжету

SentimentIntensityAnalyzer є функцією з бібліотеки NLTK, яка дозволяє визначити настрій сюжету по ключовим словам у тексті, результат роботи моделі – число, від 0 до 1, по кожному з наступних параметрів: positive, negative, neutral. Це число означає на скільки відсотків текст позитивний, негативний чи нейтральний.

Модель працює на напереднавченній нейронній мережі, яка аналізує токени в кожному реченні і визначає речення позитивне, негативне чи нейтральне. Для оцінки тексту береться сумма параметрів оцінки кожного речення і виводиться фінальний коефіцієнт для всього тексту. Більш детально про можливості SentimentIntensityAnalyzer можна прочитати в документації [4][…].

## Висновки до розділу

У цьому розділі розібрано математичне забезпечення для задачі «Математичне та програмне забезпечення для обробки й класифікації архіву новинних відеорепортажів». Розібрані основні методи, бібліотекти на алгоритми, з якими проводиться робота для виконання поставлених цілей. Наступним етапом є написання програмного коду. Розробка алгоритму буде відбуватись на мові програмуванні python 3.8.8. Середа розробки – jupyter notebook.

# Програмне забезпечення

## Структура програми

Програмне забезпечення для розв’язання поставленої задачі спроектовано з дотриманням парадигм об’єктно-орієнтованого програмування [16]. Використано шаблон проектування Модель-Подання-Пред’явник, відповідно до якого логіка роботи програми відділяється від візуального відображення даних у різні класи. На рисунку 4.1 представлено схему потоків даних між модулями використаного шаблону.

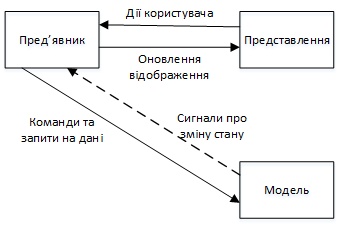


Рисунок 4.1 – Схема потоків даних між модулями MVP розробленого програмного забезпечення

Усю логіку роботи системи MANFIS зосереджено в модулі «Модель», у зв’язку з чим на нього покладаються такі функції:

* зчитування параметрів мережі з файлу;
* обчислення вихідних сигналів;
* здійснення однієї ітерації навчання;
* запис параметрів мережі у файл.

[…]

## Опис розроблених алгоритмів

До кожної підсистеми розробленої ANFIS застосовано гібридне навчання, описане в 3.2. Блок-схему алгоритму проведення такого навчання наведено на рисунку 4.2.

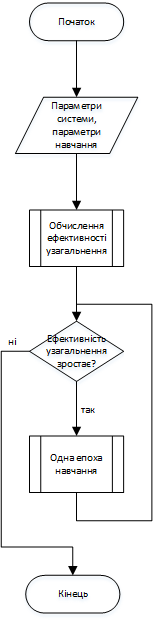


Рисунок 4.2 – Блок-схема алгоритму гібридного навчання ANFIS

[…]

## Формат вихідних даних

### Вихідні дані для ініціалізації системи

Вихідними даними для ініціалізації системи є файл із її параметрами weights.xml, що знаходиться в директорії sources.

Параметри у файлі відформатовано наступним чином:

1. у перших найглибших чотирьох тегах знаходяться передумовні параметри, у наступних найглибших тегах знаходяться наслідкові параметри;

передумовні параметри згруповано за елементами обличчя, ознаки яких вони описують;

кожний рядок передумовних параметрів містить параметри вузлової функції нейрона першого шару мережі, ,  та  відповідно;

наслідкові параметри згруповано за емоцією, яку вони описують:

1. здивування;
2. щастя;
3. сум;
4. відраза;
5. злість;

кожний рядок наслідкових параметрів складений із параметрів вузлової функції нейронів четвертого шару;

усі параметри — дійсні числа, ціла частина відділяється від дробової крапкою.

### Вихідні дані для навчання системи

[…]

## Формат результуючих даних

[…]

## Результати випробування

### Навчання ANFIS

[…]

### Випробування системи

[…]

## Висновки до розділу

У цьому розділі розроблено програмне забезпечення для автоматизованої системи розпізнання емоційного стану людини за статичним фронтальним зображенням її обличчя.

Реалізовану програмно систему навчено на вибірці зі 180 зображень.

У ході тестування обчислено показники ефективності навчання та тестування. Для навчання сукупно він склав 85%, для тестування сукупно — 73%.

Висновки

У роботі розглянуто основні підходи до розпізнавання емоційного стану людини за статичним фронтальним зображенням її обличчя: штучні нейронні мережі, приховані марківські моделі, порівняння з шаблоном та адаптивне нейронечітке виведення. У результаті проведеного порівняльного аналізу за наперед визначеними критеріями для вирішення поставленої задачі обрано модифікацію архітектури системи адаптивного нейронечіткого виведення — MANFIS.

Розглянуто й модифіковано методологію виділення вектора ознак із зображення обличчя. Для кожної з ознак уведено множину нечітких значень, яких вона може набувати. На множині цих нечітких значень побудовано правила продукції для ідентифікації кожної емоції.

Спроектоване математичне забезпечення реалізовано програмно.

Розроблену програмну систему навчено на вибірці зі 180 зображень гібридним алгоритмом із використанням перехресної перевірки для зупинки тренування.

У ході тестування виявлено показники ефективності: у середньому за всіма емоціями показник ефективності навчання склав 85%, а показник ефективності узагальнення 73%.

Перелік посилань

1. Птах, Стівен, Едвард Лопер та Юан Кляйн (2009), Обробка природної мови за допомогою Python . O'Reilly Media Inc.
2. <https://www.nltk.org/>
3. <https://spacy.io/>
4. <https://www.nltk.org/api/nltk.sentiment.html>
5. Лістинги програм

Ілюстративний матеріал

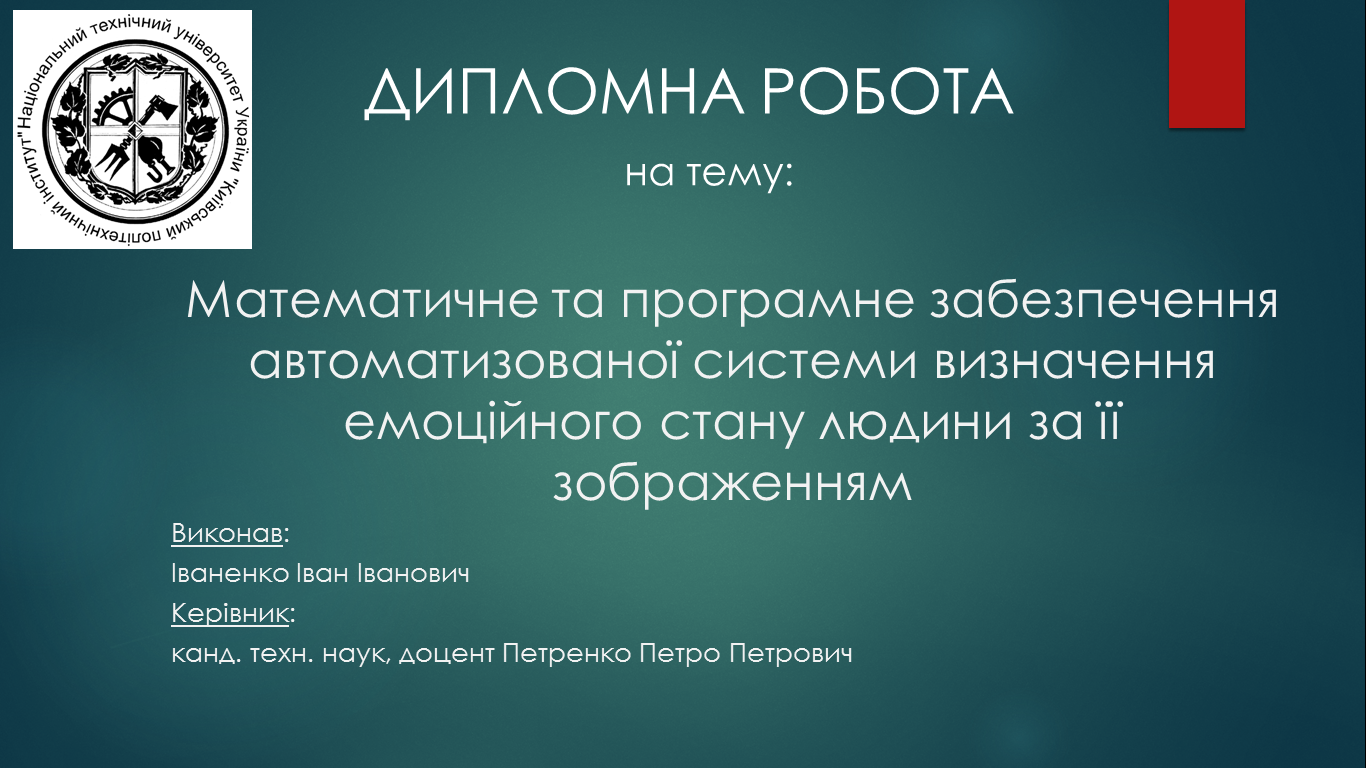


Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2