**ADR-readout**

Haidash Yevhen, Hantsiak Andrii, Duboviy Daniil,

Lieschuk Valentyn, Karpliuk Oleksii

Technická univerzita v Košiciach

Informatika, Inteligentné systémy, KKUI

Košice, 04011

evgengajdas@gmail.com

**ABSTRACT**

Ціль проекту полягає в створенні веб сторінки, на якій користувач зможе порівняти налаштування свого аналогового годинника зі світовим часом шляхом завантаження фотографії свого годинника на сайт. Основною проблемою для вирішення є зчитування точної години і хвилини з фотографії. Проблему зчитування розбито на дві частини. Перша частина полягає в отриманні спрощеного малюнка з оригінального зображення, на якому виділено область циферблата, годинна та хвилинна стрілка. Автори вирішують це завдання використовуючи OpenCV – бібліотеку в Python для обробки зображень. У другій частині згорткова нейронна мережа перетворює спрощену схему годинника на чисельні значення. Було перевірено різні топології моделі, яка навчалася і тестувалася на наборі даних з 50 тисяч картинок. У статті описується процес навчання моделі, та аналізуються різні спроби покращити точність моделі. Учням вдалося досягти точності зчитування часу у 86% випадках, з допустимою похибкою в три хвилини.

**Keywords**: CNN, OpenCV, clock readout, python, model training, website, pytorch

# Introduction

Проблема дослідження полягає в процесі обробки зображень та в побудові ефективної згорткової нейронної мережі для розпізнавання часу на аналоговому годиннику. Також розкриваються приховані труднощі при створенні веб-сайту.

Дослідження має на меті продемонструвати роботу команди над проектом з науково технічної сторони. Також ціль дослідження полягає в тому, щоб висвітлити читачеві проблематику та запропонувати можливі рішення і продемонструвати ефективні методи і підходи для оптимізації ефективності навчання моделі.

Розпізнавання зображень за допомогою CNN має величезну значущість у таких сферах як: Медична діагностика, автономні автомобілі, безпека та відеоспостереження, якість зображень. Усе це підкреслює важливість вирішення проблеми розпізнавання зображень за допомогою згорткових нейронних мереж у сучасному світі. У світі наукової спільноти це дослідження є корисним для тих, хто тільки починає знайомитися з основами комп’ютерного зору.

# LITERATURE REVIEW

Попередні дослідження в області розпізнавання зображень вказують на високу ефективність застосування CNN в різних сферах.

У своїй роботі учні опиралися на університетський підручник, відеоролики в YouTube та онлайн форуми, майже не використовуючи академічні джерела. Авторам не вдалося у повній мірі реалізувати стабільно працюючий сайт. Напрямки подальших досліджень можуть включати аналіз альтернативних архітектур CNN, як наприклад ResNet, Inception та DenseNet. Також додаткового дослідження потребує процес перетворення реальної фотографії у спрощену за допомогою OpenCV.

Хоч дослідження учнів має обмежений обсяг і нереалізовані функції, воно все ж відкриває шлях для подальших досліджень в області розпізнавання зображень за допомогою CNN.

# Загальна структура проєкту

За початковою задумкою, весь обсяг роботи в нашому проекті було поділено на три частини: Написання програми для спрощення реальної фотографії, навчання згорткової нейронної мережі на датасеті зі спрощеними зображеннями та реалізація цих двох програм на веб-сайті з легким для використання користувацьким інтерфейсом. Готовим продуктом нашої роботи, мав стати сайт, який дозволить користувачу завантажити зображення свого годинника, обрати часовий пояс, який його цікавить, і отримати на виході різницю між точним світовим часом і часом на його годиннику. Такий сайт є корисним для колекціонерів годинників, яким потрібно постійно налаштовувати свої годинники і крім цього міняти налаштування пружинок, щоб змінити швидкість стрілок.

Отже перед тим, як користувач отримає вказівки щодо налаштування, його фотографія повинна пройти через програму по спрощенню, потім ця спрощена схема передається до навченої моделі, яка в свою чергу дасть на виході час у числових значеннях, і тільки після цього процесу користувачу виведеться різниця у часі між його годинником і актуальним часом. В ідеалі веб-сайт повинен працювати на сервері, але в рамці нашого дослідження ми використовуємо тільки локальний сервер.

# Виділення ознак за допомогою OpenCV

Ця частина проекту була створена для трансформації зображення у вигляд придатний для використання вводом нейронної мережі. Його функціонал можна розділити на 6 етапів:

1) Завантаження зображення: зображення завантажується в програму як numpy array з розмірами 500 на 500 пікселів.

2)Перетворення у відтінки сірого: Перетворення зображення у відтінки сірого спрощує подальше опрацювання, оскільки воно видаляє колірну інформацію, залишаючи тільки яскраву складову. Це зазвичай використовується для роботи з областями контрастного зображення, що поліпшує результати виявлення об'єктів.

3)Зниження шуму: Медіанний фільтр застосований для зниження шуму на зображенні. Це важливо, тому що шум може впливати на результати обробки зображень, як-от виявлення об'єктів або меж.

4)Виявлення кіл на зображенні: Перетворення Хафа використовується для виявлення кіл на зображенні. Це допомагає знайти контур годинника і позбавитися заднього фону

5)Виявлення ліній на зображенні: Перетворення Хафа також було використане для виявлення прямих ліній на зображенні. За допомогою цього методу і зміні параметрів вдається виявляти хвилинну і годинникову стрілки. Алгоритм пошуку стрілок працює по такому принципу: спочатку алгоритм знайде найдовші лінії, потім запише ті лінії, які лежать ближче до центру. В кінці цього пошуку має залишитися 6 ліній де кожна пара відповідає за одну стрілку і між тими лініями є кут, за яким вираховується чи є стрілка секундною, тобто вираховується кут між лініями і оскільки секундна стрілка тонша за інші то лінії з таким кутом видаляються.

6)Збереження обробленого зображення: Після завершення всіх кроків обробки зображення зберігається як новий файл. Це дає змогу зберегти отримані дані для подальшого використання.

# Навчання моделі

У цій частині проекту за мету було поставлено створити і навчити модель для класифікації часу. Для цього у розпорядженні студентів був набір даних із онлайн платформи Kaggle. Цей датасет складався із 50 тисяч картинок розміром 300 на 300 пікселів і відповідних міток для кожної картинки, які включали в собі годину від 0 до 12 та хвилину від 0 до 59. Створювати модель було вирішено у python з використанням бібліотеки pytorch. Перед тим як використовувати зображення з датасету, їх потрібно було попередньо обробити, щоб зменшити розмір вступного простору. Тому кожна фотографія пройшла перетворення з розміру 300 на 300 у розмір 100 на 100. Крім цього із трьох колірних каналів було отримано один канал, який відображав зображення у градації сірого кольору, що зменшило вступний простір ще в три рази. Всі ці оброблені зображення було поділено на навчальний, валідаційний і тестувальні датасети, при цьому із цих трьох датасетів в подальшому було сформовано даталоудери з розміром бетчу в 32 зображення, що запобігає перевантаженню памяті пристрою. Були також написані цикли для тренування і тестування моделі.

Наступним кроком став підбір гіперпараметрів і різних архітектур моделі. Автори протестували різні варіації гіперпараметрів та топологій, для прикладу було проведено багато тестів над архітектурою з 60 вихідними нейронами для класів хвилин та 12 нейронами для годин відповідно. Зрештою автори зупинилися на топології, яка на виході мала 13 нейронів, з них 12 це класи для годин, і 1 нейрон виступав як значення хвилини від 0 до 60. Тобто було застосовано класифікацію для години і регресію для хвилини.

Після низки тестів розробники зупинилися на топології, яка складалася з чотирьох згорткових шарів, і функції dropout після яких ідуть дві паралельні повністю з’єднані лінійні мережі для класифікації години і регресії хвилини, кожна з цих мереж має по два приховані шари. Ефективне навчання також забезпечували оптимізатор Adam, та складена функція помилки. 

Як видно на малюнку(1), функція втрат складається з суми функцій помилок години і хвилини, які перед цим ще множаться на свої ваги, що дозволяє регулювати значущість тієї чи іншої помилки. Значення помилки hourLoss рахується за крос-ентропійною функцією, а значення minuteLoss в свою чергу рахується як середня квадратична помилки. Шляхом чисельних тестів було визначено, що найраща точність моделі досягається при відношенні ваг 20 : 1, де 20 значення ваги для години.

Навчену модель було збережено у .pth форматі і в подальшому використано у функції для передбачення. При всіх цих конфігураціях вдалося досягти наступних результатів: точність в 0.7323 при допустимому відхиленні хвилинної стрілки в 2 хвилини, 0.8638 при відхиленні в 3 хвилини і 0.9293 відхиленні в 5 хвилин.

**4. CONCLUSION**

Застосування бібліотеки OpenCV для обробки зображень дозволило створити спрощену схему годинника, але отримане зображення не вдалося обробляти нейронною мережею, адже отриманий результат не схожий на схеми годинників з тренувального набору даних. Проте вдалося створити, та натренувати згорткову нейронну мережу на основі навчального набору даних, який містить в собі 50 000 схематичних зображень годинників, що показують різний час. Результати дослідження показали, що обраний підхід забезпечує точність у 93% випадків з допустимою похибкою до трьох хвилин. Процес навчання моделі та оптимізації ефективності показав, що правильний вибір архітектури та налаштування моделі можуть суттєво вплинути на точність результатів. Детально описано методи, використані для вирішення проблеми обробки зображень, і вказано на можливі шляхи покращення точності та продуктивності системи. Це дослідження буде корисним для тих, хто тільки починає вивчати комп'ютерний зір і прагне зрозуміти практичні аспекти застосування згорткових нейронних мереж для розпізнавання зображень.

**REFERENCES**

[Format for References: Times New Roman, 11 pt; 0 pt before and after paragraph spacing; single line spacing; numbered bullets. Use APA citation style]

1. APA Citation Style. (2014). Retrieved on July 27 from <http://www.apastyle.org/>
2. Lau, Linda. (2014). This is a sample template for authors. *Journal of Digital Forensics, Security and Law, 9(2), 1-2.*

# Manuscript Templates for Conference Proceedings. (2014). Retrieved on July 27 from [Http://Www.Ieee.Org/Conferences\_Events/Conferences/Publishing/Templates.Html](http://www.ieee.org/conferences_events/conferences/publishing/templates.html)

1. Research and Documentation Online. (2014). Retrieved on July 27 from <http://bcs.bedfordstmartins.com/resdoc5e/RES5e_ch09_s1-0001.html>
2. University of New Haven. (2014). Retrieved on July 27 from <http://d-forensics.org/2014/show/home>
3. Williams College Libraries. (2014). Retrieved on July 27 from <http://library.williams.edu/citing/styles/apa/php>
4. Official OpenCV documentation (v 4.9.0) <https://docs.opencv.org/4.x/>