

# ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЛЮДИННО-МАШИННОГО ІНТЕРФЕЙСУ ДЛЯ АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ ЦИФР НА ФОТОГРАФІЯХ АВТОМОБІЛЬНИХ НОМЕРІВ

Вінницький національний технічний університет;

## *Анотація*

*У тезах запропоновано результати проектування людинно-машинного інтерфейсу для автоматизованого розпізнавання цифр на фотографіях автомобільних номерів. Наведено результати тестування створеної системи.*

**Ключові слова:** людинно-машинний інтерфейс для автоматизованого розпізнавання цифр на фотографіях автомобільних номерів, результати експериментів.

## *Abstract*

*The thesis proposes the results of the design of the human-machine interface for the automatic recognition of numbers in photos of car numbers. The results of testing of the created system are given.*

**Keywords:** human-machine-human interface for automated number recognition on car license plate photos, experimental results.

## Вступ

Враховуючи, що більшість інформації людина сприймає у вигляді зображень, їх автоматизований аналіз є природнім і дуже інформативним способом опису навколишнього середовища. Так як кількість джерел отримання візуальної інформації стрімко зростає, задача автоматизованого аналізу зображень стає вкрай актуальною. Існує багато методів розпізнавання зображень, наприклад, методи основані на дослідженні контуру об'єкта, методи перебору вигляду об'єкта під різними кутами, зсувами і тому подібне [1].

Найбільш ефективним методами у розпізнаванні зображень є штучні нейронні мережі, зокрема, згорткові нейронні мережі глибокого навчання. Глибоке навчання (deep learning) – це галузь машинного навчання, яке ґрунтується на підборі алгоритмів, що здійснюються за допомогою нейронних мереж із спеціалізованою архітектури, про що говоритиметься далі [1].

Метою роботи є створення програмного забезпечення для розпізнавання цифр номерного знаку.

## Результати дослідження

Програмне забезпечення для поставленої задачі розпізнавання було розроблене у пакеті прикладних програм Matlab R2016b із використанням Deep Learning toolbox. Класифікатор розроблений на основі згорткової нейронної мережі навчався на зображеннях з Каліфорнійського технологічного університету [2]. Цей університет є одним з найбільш цитованих на використовуваних наборів зображень, зібраних з відомих марок та трендів. Дані завантажуються з допомогою спеціальної функції зображеної на рисунку 1.

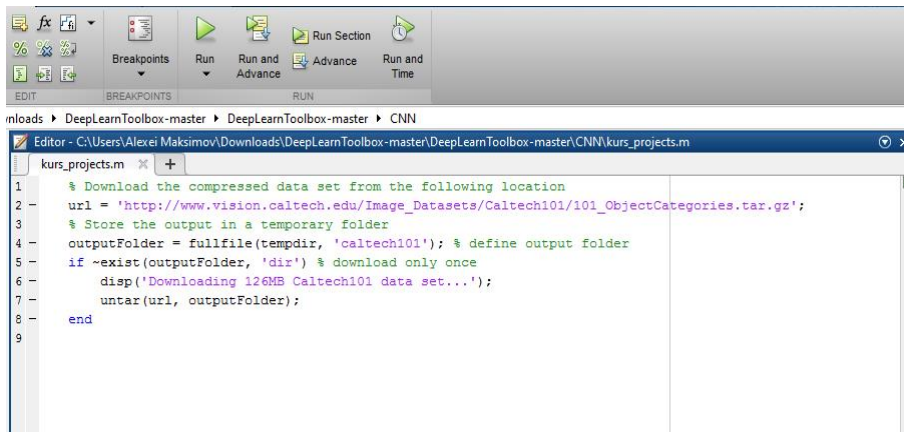


Рисунок 1 – Функція завантаження зображень

Після завантаження зображень, необхідно вибрати декілька категорій, з якими буде продовжуватись робота. Далі потрібно підключити функцію, яка завантажує зображення в пам'ять до початку роботи з ними, тим самим полегшуючи роботу з великим обсягом даних. Оскільки у категоріях зображень не однакова кількість даних, їх потрібно налаштувати таким чином, щоб кількість зображень в навчальному наборі врівноважилась. На рисунку 2 зображено категорії зображень до та після врівноваження.

tbl =		ans =	
3x2 table array		3x2 table array	
Label	Count	Label	Count
airplanes	800	airplanes	67
ferry	67	ferry	67
laptop	81	laptop	67

Рисунок 2 – Категорії зображень до та після їх врівноваження

Як видно з категорій, серед зображень, для навчання, обрано 3 категорії, а саме: літаки, пароплави а також ноутбуки. У кожній з категорій буде зберігатись по 67 зображень. Такі категорії обрані для того, щоб форми об'єктів суттєво різнились, для більш точного навчання нейронної мережі. Нижче, на рисунку 3, можна побачити отримані зображення з трьох різних категорій.

```

% Find the first instance of an image for each category
airplanes = find(imds.Labels == 'airplanes', 1);
ferry = find(imds.Labels == 'ferry', 1);
laptop = find(imds.Labels == 'laptop', 1);

```



Рисунок 3 – Отримані зображення з різних категорій

Тепер, коли зображення завантажені, потрібно підключити модуль згорткової нейронної мережі (CNN). Існують вже створені безкоштовні попередньо навчені модулі, які вже встигли завоювати популярність та якими користуються багато компаній. Багато з них проходили навчання на колекціях зображень з Каліфорнійського технологічного інституту. Для навчання була обрана одна з таких мереж AlexNet [3]. Оскільки модель має специфічний формат (MatConvNet), її потрібно конвертувати за допомогою вбудованої функції Matlab (convent). Отримана мережа має 23 шари та зображена на рисунку 4. На даному рисунку простежуються загальна структура нейронної мережі.

```

hw to MATLAB? See resources for Getting Started.
Layers: [23x1 nnet.cnn.layer.Layer]

ans =

23x1 Layer array with layers:

 1 'input'           Image Input           227x227x3 images with 'zerocenter' normalization
 2 'conv1'          Convolution           96 11x11x3 convolutions with stride [4 4] and padding [0 0]
 3 'relu1'          ReLU                  ReLU
 4 'norm1'          Cross Channel Normalization cross channel normalization with 5 channels per element
 5 'pool1'          Max Pooling           3x3 max pooling with stride [2 2] and padding [0 0]
 6 'conv2'          Convolution           256 5x5x48 convolutions with stride [1 1] and padding [2 2]
 7 'relu2'          ReLU                  ReLU
 8 'norm2'          Cross Channel Normalization cross channel normalization with 5 channels per element
 9 'pool2'          Max Pooling           3x3 max pooling with stride [2 2] and padding [0 0]
10 'conv3'          Convolution           384 3x3x256 convolutions with stride [1 1] and padding [1 1]
11 'relu3'          ReLU                  ReLU
12 'conv4'          Convolution           384 3x3x192 convolutions with stride [1 1] and padding [1 1]
13 'relu4'          ReLU                  ReLU
14 'conv5'          Convolution           256 3x3x192 convolutions with stride [1 1] and padding [1 1]
15 'relu5'          ReLU                  ReLU
16 'pool5'          Max Pooling           3x3 max pooling with stride [2 2] and padding [0 0]
17 'fc6'            Fully Connected      4096 fully connected layer
18 'relu6'          ReLU                  ReLU
19 'fc7'            Fully Connected      4096 fully connected layer
20 'relu7'          ReLU                  ReLU
21 'fc8'            Fully Connected      1000 fully connected layer
22 'prob'           Softmax              softmax
23 'classificationLayer' Classification Output cross-entropy with 'n01440764', 'n01443537', and 998 other classes

```

Рисунок 4 – Згорткова мережа, яка складається з 23 шарів

Кожен рівень CNN відповідає або застосовує функцію активації до вхідного зображення. Проте є, декілька шарів, які підходять до вилучення характеристик зображення. Шари на початку мережі фіксують основні атрибути зображення, наприклад край зображення [1,2]. Побачити це можна на першому згортковому рівні мережі, який зображено на рисунку 5. Ці шари далі обробляються більш глибокими мережевими рівнями, які об'єднують попередні функції для створення більш якісного зображення.

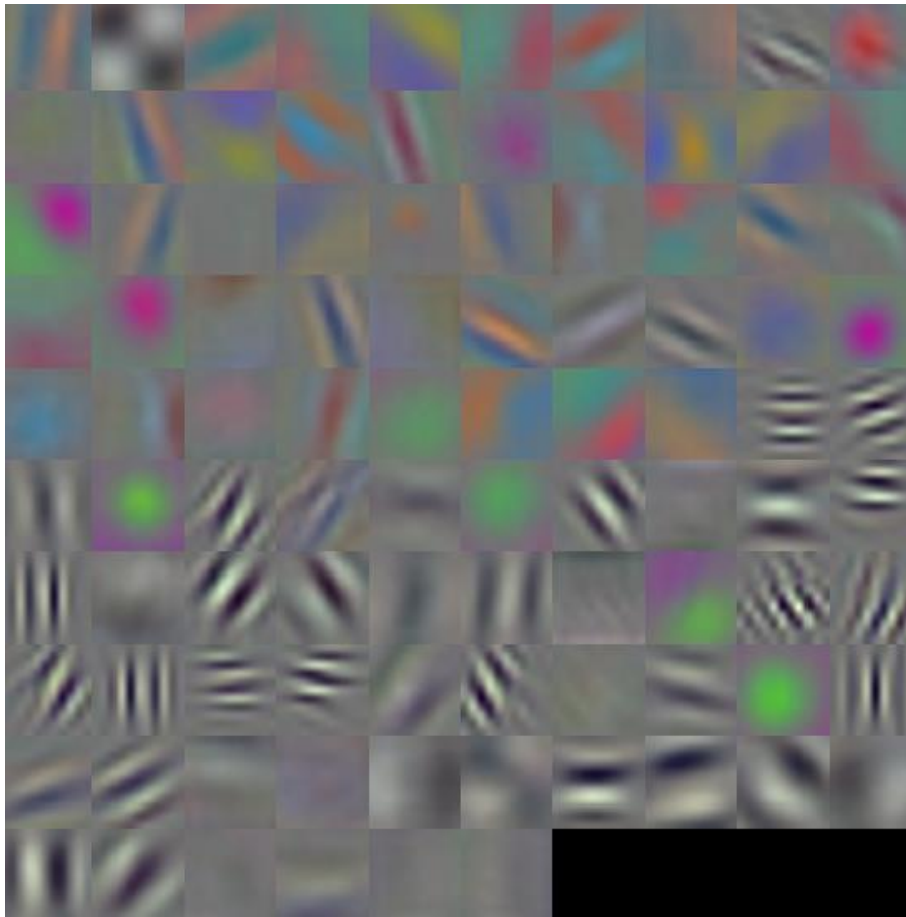


Рисунок 5 – Перший згортковий рівень CNN

Далі зображення розбиваються на два класи: навчальні (30%) та тестові (70%). Це потрібно для валідації. Також потрібно рандомізувати дані класи, щоб уникнути зміщення результатів. Тренувальні та тестові класи будуть оброблятися моделлю CNN. При обробці використовуються функції зображення CNN для підготовки мультикласового класифікатора SVM. На наступному етапі для навчання використовується Gradient Descent solver, який змінює параметр функції fitsecos (“Learners” на “Linear”).

Тепер класифікатор згорткової нейронної мережі готовий до категоризації та розпізнавання нових зображень, а саме, згідно тематики даної роботи, автомобільних номерних знаків. Для цього було проведено переналаштування CNN за принципами описаними вище. Відмінність полягала лише у кількостях категорій, які розпізнає мережа. Оскільки класифікатор має розпізнавати автомобілі на пропускному пункті, то має бути дві категорії: автомобілі, які можуть проїхати на заборонену територію, та, відповідно, – не можуть. На рисунках 6 зображено результати тестування даного класифікатора.

```

newImage = fullfile(rootFolder, 'airplanes', 'image_0690.jpg');

% Pre-process the images as required for the CNN
img = readAndPreprocessImage(newImage);

% Extract image features using the CNN
imageFeatures = activations(convnet, img, featureLayer);

% Make a prediction using the classifier
label = predict(classifier, imageFeatures)

label =
    airplanes

```

Рисунок 6 – Тестування класифікатора на навчальних зображеннях

Слід зауважити, що система може навчатися на різних апаратних засобах, використовуючи ресурси процесора (CPU) або ресурси відеоадаптера (GPU), при чому відеоадаптер може бути виключно виробника NVidia. Отже, доцільним є виявлення залежності між тривалістю навчання згорткової нейронної мережі та обраною апаратною платформою (CPU/GPU). Тривалість навчання згорткової нейронної мережі на різних платформах зображено на рисунку 7.



Рисунок 7 – Залежність тривалості навчання CNN від апаратного забезпечення

Згідно наведеного вище рисунку було визначено, що навчання мережі із застосуванням GPU є у 4 рази меншою порівняно із навчанням мережі на CPU при застосуванні алгоритмів з Matlab Deep learning toolbox.

### Висновки

Для досягнення поставленої мети було розроблено класифікатор згорткової нейронної мережі на основі зображень Каліфорнійського університету. Класифікатор містив три класи зображень: літаки, пароплави та ноутбуки, і показав високу якість розпізнавання зображень. Отримана згорткова нейронна мережа складається з 23 шарів, яких виявилось цілком достатньо для високої якості розпізнавання образів. Було виявлено, що згорткову нейронну мережу доцільно навчати на

відеоадаптері виробника NVidia. Швидкість такого навчання в чотири рази швидше ніж на процесорі комп'ютера.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kovtun V.V. The automated speaker recognition system of critical use. / Mykola M. Bykov, Viacheslav V. Kovtun, Igor D. Ivasyuk, Andrzej Kotyra, Aisha Mussabekova // Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2018, 108082V (1 October 2018); doi: 10.1117/12.2501688.

2. Ковтун В.В. Концепція впровадження автоматизованої системи розпізнавання мовця у процес автентифікації для доступу до критичної системи / В.В. Ковтун // Вісник Вінницького політехнічного інституту, Вінниця. – 2018. - №5. – 41-52 с. DOI 10.31649/1997-9266-2018-140-5-41-52.

3. Ковтун В.В. Дослідження ефективності ознак розпізнавання мовців при використанні згорткових неймереж / М.М. Биков, В.В. Ковтун // Оптико-електронні інформаційно-енергетичні технології, Вінниця. – 2016. - №2(32). – 22-28 с.

**Олійник Олександр Юрійович** — студент групи КІВ-166, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, Вінниця

**Соболев Олександр Володимирович** — студент групи КІВ-166, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, Вінниця

**Ковтун В'ячеслав Васильович** — канд. техн. наук, доцент кафедри т комп'ютерних систем управління, Вінницький національний технічний університет, Вінниця, e-mail: kovtun\_v\_v@vntu.edu.ua

Науковий керівник: **Ковтун В'ячеслав Васильович** — канд. техн. наук, доцент кафедри т комп'ютерних систем управління, Вінницький національний технічний університет, Вінниця

**Sobolev Alexander Vladimirovich** — Student of the Group KIB-166, Faculty for Computer Systems and Automatic, Vinnytsia National Technical University, Vinnytsia

**Olejnik Oleksandr Yurievich** — Student of the Group KIB-166, Faculty for Computer Systems and Automatic, Vinnytsia National Technical University, Vinnytsia

**Kovtun Vjatcheslav Vasilievich** — Cand. Sc. (Eng), Assistant Professor for the Computer Control Systems Department, Vinnytsia National Technical University, Vinnytsia, e-mail: kovtun\_v\_v@vntu.edu.ua

Supervisor: **Kovtun Vjatcheslav Vasilievich** — Cand. Sc. (Eng), Assistant Professor for the Computer Control Systems Department, Vinnytsia National Technical University, Vinnytsia