


## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА


на тему Метод класифікації автомобілів за зображеннями засобами  
комп'ютерного зору

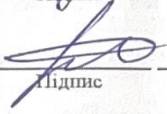
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань


Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент групи КН-22-2  Андрій ГУЛА  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: ст. викл. кафедри КН  Тетяна СКРИПНИК  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:  
зав. кафедри КН, д.т.н., професор  Олександр БАРМАК  
Підпис Ім'я, ПРІЗВИЩЕ

18 червня 2026 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій  
Кафедра комп'ютерних наук  
Освітній ступінь бакалавр  
Галузь знань 12 – Інформаційні технології  
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор Олександр БАРМАК

«22» січня 2026 року

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору»

2. Завдання видано студенту Андрію ГУЛІ  
(Ім'я, прізвище)

3. Керівник роботи ст викладач кафедри КН Тетяна СКРИПНИК  
(посада, ім'я, прізвище)

4. Затверджено наказом університету від «20» січня 2026 р. № 7

5. Дата видачі завдання студенту: «22» січня 2026 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи бакалавра є підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу.

Для досягнення поставленої мети необхідно вирішити такі завдання: провести комплексний аналіз предметної області класифікації автомобілів за цифровими зображеннями та існуючих підходів до розв'язання цієї задачі; розробити нейромережевий метод класифікації автомобілів із використанням технології transfer learning, здійснити програмну реалізацію прототипу запропонованого методу, виконати тестування розробленого методу та оцінити його точність на основі обраних метрик.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження теми кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання	січень 2026	виконано
2	Ознайомлення з предметною областю, формулювання мети і задач дослідження, визначення об'єкта та предмета дослідження	лютий 2026	виконано
3	Проектування методу розв'язання задачі, опис архітектурних рішень, розроблення математичних моделей та алгоритмів	березень 2026	виконано
4	Обґрунтування інструментарію розробки, програмна реалізація розробленого методу, проведення експериментального тестування та оцінювання ефективності	квітень 2026	виконано
5	Написання тексту кваліфікаційної роботи, урахування зауважень керівника, оформлення згідно з вимогами	травень 2026	виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2026	виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2026	виконано
8	Підготовка до захисту та захист кваліфікаційної роботи	червень 2026	виконано

Виконавець:

*студент групи КН-22-2*

Група виконавця

  
Підпис

*Андрій ГУЛА*

Ім'я, ПРІЗВИЩЕ

Керівник:

*ст. викладач кафедри КН*

Науковий ступінь, посада

  
Підпис

*Тетяна СКРИПНИК*

Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-22-2 Андрій Гула

Керівник кваліфікаційної роботи бакалавра: ст. викладач кафедри КН Тетяна СКРИПНИК

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
64	19	10	40	2

Метою кваліфікаційної роботи бакалавра є підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу. Для досягнення мети та експериментального підтвердження працездатності запропонованого підходу розроблено метод класифікації автомобілів та виконано його програмну реалізацію.

Практичне значення кваліфікаційної роботи полягає у створенні програмного засобу для автоматичног ідентифікації автомобілів за цифровими зображеннями, який може бути використаний у системах відеоспостереження, транспортного моніторингу, страхових компаніях, сервісних центрах та на ринку вживаних автомобілів.

Ключові слова: глибоке навчання, комп'ютерний зір, класифікація автомобілів, EfficientNet-B0, transfer learning, Stanford Cars, PostgreSQL.

Виконавець: студент групи  
Група виконавця

  
Підпис

Андрій ГУЛА  
Ім'я, ПРІЗВИЩЕ

## Зміст

Перелік скорочень .....	4
Вступ .....	5
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій .....	8
1.1 Аналіз предметної області .....	8
1.2 Огляд моделей глибокого навчання для задач класифікації автомобілів .....	10
1.3 Огляд теоретичних підходів до розв'язку подібних задач.....	14
1.4 Аналіз існуючих програмних засобів .....	16
1.5 Мета та завдання дослідження.....	18
Розділ 2 Розробка методу класифікації автомобілів за зображеннями .....	19
2.1 Математична формалізація задачі .....	19
2.2 Опис архітектури моделі глибокого навчання .....	22
2.3 Математичний псевдокод методу.....	27
2.4 Опис датасету .....	31
2.5 Метрики оцінювання ефективності .....	33
2.6 Сценарії проведення експерименту .....	36
РОЗДІЛ 3 Експериментальне дослідження методу нейромережевої класифікації автомобілів .....	38
3.1 Опис експериментального застосування .....	38
3.2 Результати експериментів .....	41
3.3 Аналіз процесу навчання моделі .....	53
3.4 Порівняльний аналіз архітектур .....	54
3.5 Висновки до розділу .....	54
ВИСНОВКИ.....	56
ПЕРЕЛІК ПОСИЛАНЬ.....	58
Додатки	

### Перелік скорочень

Скорочення	Пояснення
CNN	Convolutional Neural Network – згорткова нейронна мережа
FGVC	Fine-Grained Visual Categorization – дрібнозерниста візуальна класифікація
GPU	Graphics Processing Unit – графічний процесор
GUI	Graphical User Interface – графічний інтерфейс користувача
NAS	Neural Architecture Search – пошук нейромережових архітектур
RGB	Red, Green, Blue – модель кольору (червоний, зелений, синій)
SE (Squeeze-and-Excitation)	Механізм каналної уваги у нейронних мережах
СУБД	Система управління базами даних

## Вступ

**Актуальність** теми дослідження обумовлена стрімким розвитком інформаційних технологій, систем комп'ютерного зору та інтелектуальних транспортних систем, у яких все більшого значення набувають задачі автоматичної ідентифікації транспортних засобів. Системи автоматичного визначення марки та моделі автомобіля знаходять широке застосування у сферах безпеки дорожнього руху, відеоспостереження, страхування, автоматизованого паркування, контролю транспортних потоків, а також на платформах продажу вживаних автомобілів [1]. Зростання кількості транспортних засобів та різноманіття їх моделей ускладнює виконання такої ідентифікації вручну, що обумовлює необхідність автоматизації відповідних процесів.

Сучасний автомобільний ринок налічує тисячі моделей транспортних засобів, які часто мають подібні конструктивні та дизайнерські особливості. Це створює значні труднощі для традиційних методів аналізу зображень та потребує використання інтелектуальних алгоритмів, здатних виявляти складні візуальні закономірності. Впровадження методів комп'ютерного зору та глибокого навчання дозволяє автоматизувати процес класифікації транспортних засобів та підвищити якість розпізнавання [2].

Задача класифікації автомобілів за зображеннями належить до класу задач дрібнозернистої візуальної класифікації (Fine-Grained Visual Classification, FGVC) [3]. На відміну від традиційної класифікації об'єктів, у таких задачах необхідно розрізняти підкатегорії одного класу об'єктів, між якими існують незначні зовнішні відмінності. Для автомобілів такими ознаками можуть бути форма фар, решітки радіатора, елементи кузова, конфігурація бамперів та інші деталі. Додатковими ускладнювальними факторами є різні ракурси зйомки, зміни освітлення, часткові перекриття об'єктів та атмосферні умови [4].

Останніми роками значного поширення набули методи глибокого навчання, зокрема згорткові нейронні мережі, які демонструють високі результати у задачах класифікації зображень. Особливу увагу дослідників привертають архітектури сімейства EfficientNet, що забезпечують високу точність класифікації при відносно невеликих обчислювальних витратах завдяки використанню принципу комплексного масштабування мережі [5].

Використання технології transfer learning дозволяє ефективно адаптувати попередньо навчені моделі до спеціалізованих предметних областей та отримувати якісні результати навіть за обмеженого обсягу навчальних даних.

Таким чином, розроблення нейромережевого методу класифікації автомобілів за цифровими зображеннями на основі сучасних технологій комп'ютерного зору та глибокого навчання є актуальною науково-технічною задачею, розв'язання якої сприятиме підвищенню ефективності автоматизованої ідентифікації транспортних засобів у різних інформаційних системах.

**Об'єкт дослідження** – процес автоматизованої класифікації транспортних засобів за цифровими зображеннями.

**Предмет дослідження** – нейромережеві методи та алгоритми комп'ютерного зору для багатокласової класифікації автомобілів за візуальними ознаками цифрових зображень.

**Метою кваліфікаційної роботи бакалавра** є підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

- провести комплексний аналіз предметної області класифікації автомобілів за цифровими зображеннями та існуючих підходів до розв'язання цієї задачі;

- розробити нейромережевий метод класифікації автомобілів із використанням технології transfer learning;

- здійснити програмну реалізацію прототипу запропонованого методу;

– виконати тестування розробленого методу та оцінити його точність на основі обраних метрик.

Для досягнення поставленої мети використано методи аналізу наукових джерел, математичного моделювання, комп'ютерного зору, машинного та глибокого навчання, технології transfer learning, а також методи експериментального оцінювання якості класифікації.

Практичне значення одержаних результатів полягає у створенні програмного прототипу для автоматичної ідентифікації автомобілів за цифровими зображеннями, який може бути використаний у системах відеоспостереження, інтелектуальних транспортних системах, страхових компаніях, сервісних центрах та на платформах продажу транспортних засобів. Розроблене програмний засіб забезпечує автоматичне визначення марки та моделі автомобіля за фотографією, скорочує час обробки інформації та підвищує рівень автоматизації процесів аналізу транспортних засобів.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, переліку використаних джерел та додатків. У першому розділі проведено аналіз предметної області, сучасних методів комп'ютерного зору та програмних засобів класифікації автомобілів. У другому розділі наведено математичну формалізацію задачі, опис розробленого методу, архітектури нейронної мережі та використаного набору даних. У третьому розділі представлено програмну реалізацію запропонованого методу та результати експериментальних досліджень.

## Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

### 1.1 Аналіз предметної області

Предметною областю дослідження є автоматична ідентифікація марки та моделі легкового автомобіля за цифровим зображенням засобами комп'ютерного зору. Специфіка задачі полягає у визначенні марки, моделі та модельного року автомобіля за зображеннями, отриманими в різних умовах зйомки.

Задача класифікації автомобілів належить до галузі дрібнозернистої візуальної класифікації. Її складність обумовлена високою схожістю між різними підкатегоріями автомобілів та значною варіативністю зображень одного класу через зміну ракурсу, освітлення й наявність часткових перекриттів.

Вхідними даними є цифрові зображення автомобілів із різних джерел: рекламних матеріалів, дорожніх камер та мобільних пристроїв. Такі зображення характеризуються:

- геометричними варіаціями (ракурс, масштаб, відстань до об'єкта);
- фотометричними варіаціями (освітлення, тіні, відблиски, контраст);
- оклюзіями та перекриттями іншими об'єктами;
- різноманітністю фонів;
- модифікаціями та технічним станом автомобіля (пошкодження, нестандартне фарбування, додаткові елементи кузова).



Рисунок 1.1 – Приклад зображень автомобілів різних класів [1]

Автомобільна промисловість характеризується великою кількістю виробників і модельних рядів. Наразі у світі існує понад 100 автовиробників та тисячі комбінацій марка–модель–рік. Для задач FGVC характерна наявність незначних, але стійких візуальних ознак, зокрема форми фар, решітки радіатора, силуету кузова та дизайну елементів екстер'єру [6].

Методи класифікації автомобілів знаходять застосування в багатьох галузях. Основні напрями їх використання наведено в таблиці 1.1.

Таблиця 1.1 – Основні сфери застосування класифікації автомобілів

<b>Галузь</b>	<b>Призначення</b>
Відеоспостереження	Автоматична ідентифікація ТЗ
Страховання	Оцінювання транспортних засобів
Паркування	Контроль доступу
Продаж автомобілів	Автоматичне заповнення характеристик
Інтелектуальні транспортні системи	Моніторинг транспортних потоків

Як видно з таблиці 1.1, автоматизована класифікація автомобілів має важливе практичне значення для багатьох інформаційних систем та інтелектуальних сервісів.

Загальні проблеми ідентифікації пов'язані з великою кількістю класів, значною зовнішньою схожістю між близькими моделями одного виробника та різними роками випуску однієї моделі, а також нерівномірним розподілом класів у реальних умовах.

Специфіка предметної області полягає у необхідності розрізнення візуально схожих підкатегорій за незначними відмінностями деталей дизайну при одночасній стійкості до внутрішньокласової варіативності. Це потребує використання методів глибокого навчання, здатних виділяти тонкі дискримінативні ознаки зображень.

Додатковою складністю є постійне оновлення модельних рядів автомобілів. Рестайлінгові версії часто відрізняються лише окремими

елементами дизайну, такими як фари, решітка радіатора чи бампери, що ускладнює їх автоматичне розпізнавання.

Важливим фактором є також конвергенція дизайну, коли різні виробники використовують подібні аеродинамічні та ергономічні рішення. Наприклад, седани класу D різних марок мають схожі пропорції та силует, тому модель повинна фокусуватися на локальних дискримінативних ознаках.

Отже, задача класифікації автомобілів є комплексною проблемою комп'ютерного зору, для розв'язання якої необхідні ефективна нейромережева архітектура, якісний датасет, методи аугментації та продумана стратегія навчання. Основною складністю залишається висока схожість автомобілів різних моделей і років випуску, що обумовлює доцільність використання сучасних методів глибокого навчання для виділення дискримінативних ознак.

## **1.2 Огляд моделей глибокого навчання для задач класифікації автомобілів**

Для розв'язання задачі класифікації автомобілів за зображеннями широко застосовуються згорткові нейронні мережі та їхні похідні. Такі моделі ефективно витягують ієрархічні ознаки зображень – від низькорівневих (краї, текстури) до високорівневих (форма кузова, дизайн фар, решітка радіатора) [7].

ResNet-50 (залишкова мережа) – класична глибока згорткова архітектура, розроблена у 2015 році. Вона використовує механізм залишкових зв'язків, який дозволяє будувати глибокі мережі без проблеми зникнення градієнта. Завдяки 50 шарам ResNet-50 ефективно витягує складні ознаки зображень. Однак значна кількість параметрів (близько 25,6 млн) та висока обчислювальна складність є суттєвим обмеженням [8].

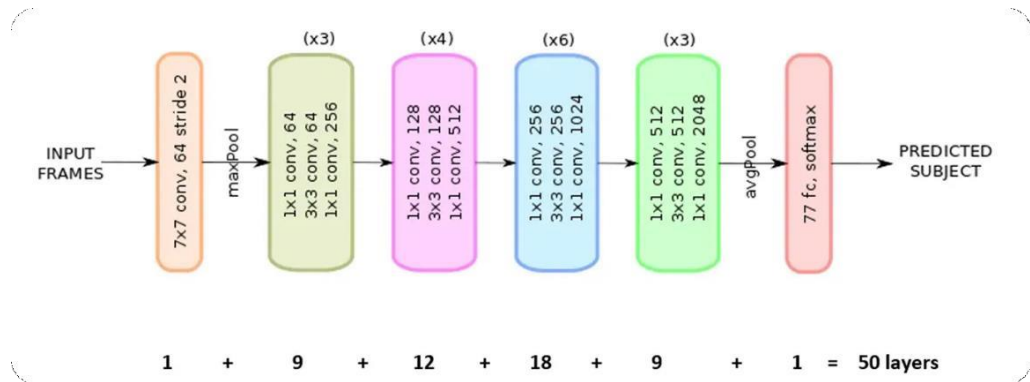


Рисунок 1.2 –Схема залишкового блоку ResNet [8]

EfficientNet-B0 використовує принцип compound scaling, який одночасно масштабує глибину, ширину та роздільну здатність мережі. Це забезпечує високе співвідношення точності до кількості параметрів і ефективність у задачах FGVC. Модель містить лише 5,3 млн параметрів та досягає Top-1 Accuracy 77,1 % на ImageNet.



Рисунок 1.3 – Архітектура EfficientNet-B0 та принцип compound scaling [5]

MobileNetV2 є легкою архітектурою Google для мобільних і вбудованих систем. Вона базується на інвертованих залишкових блоках із лінійними bottleneck-шарами та використовує depthwise separable convolutions [9], що дозволяє зменшити кількість параметрів до 3,4 млн при збереженні достатньої точності.

Vision Transformer (ViT) відрізняється від традиційних CNN використанням механізму самоуваги [10], що дозволяє враховувати глобальні

залежності між різними частинами зображення. Незважаючи на високу точність, модель потребує значних обсягів даних та обчислювальних ресурсів (86 млн параметрів).

Для обґрунтування вибору архітектури нейронної мережі проведено порівняння сучасних моделей глибокого навчання за кількістю параметрів, точністю на наборі даних ImageNet та основними перевагами. Результати порівняльного аналізу наведено в таблиці 1.2.

Таблиця 1.2 – Порівняння сучасних архітектур глибокого навчання

Архітект ура	Парам етри, млн	FLOPs	Точність ImageNet	Переваги	Недоліки
ResNet-50	25,6	4,1 GFLOPs	76,2 %	Стабільне навчання, висока надійність	Відносно велика кількість параметрів
EfficientN et-B0	5,3	0,39 GFLOPs	77,1 %	Висока точність при невеликій кількості параметрів, ефективне масштабування	Потребує ретельного налаштування параметрів навчання
MobileNet V2	3,4	0,3 GFLOPs	72,0 %	Низькі обчислювальні витрати, швидка робота на мобільних пристроях	Нижча точність порівняно з більш складними моделями

Архітект ура	Парам етри, млн	FLOPs	Точність ImageNet	Переваги	Недоліки
ViT-Base	86	17,6 GFLOP s	77,9 %	Висока якість розпізнавання складних візуальних залежностей	Значні вимоги до обсягу даних та обчислювальни х ресурсів

Аналіз даних таблиці 1.2 показує, що архітектура EfficientNet-B0 забезпечує найкраще співвідношення між точністю класифікації та кількістю параметрів моделі. Незважаючи на те, що сучасні трансформерні архітектури демонструють вищі показники точності, вони потребують значно більших обчислювальних ресурсів та обсягів навчальних даних. MobileNetV2 характеризується високою швидкістю, проте поступається за точністю. Зважаючи на необхідність розроблення ефективного методу класифікації автомобілів із можливістю використання на персональних комп'ютерах без спеціалізованого серверного обладнання, для подальших досліджень було вибрано архітектуру EfficientNet-B0, яка забезпечує оптимальний баланс між якістю класифікації та обчислювальною складністю.

Слід також відзначити, що вибір архітектури безпосередньо впливає на можливість розгортання системи в реальних умовах. ResNet-50 та ViT-Base вимагають значних обчислювальних ресурсів, що ускладнює використання без GPU. EfficientNet-B0 з часом інференсу 45 мс на CPU є прийнятним для практичного застосування, тоді як MobileNetV2 (32 мс) залишається варіантом для мобільних рішень.

Таким чином, найбільш обґрунтованим вибором для реалізації методу класифікації автомобілів є EfficientNet-B0 з технологією transfer learning від ImageNet. Ця архітектура забезпечує оптимальний баланс між точністю,

ефективністю та придатністю до донавчання на датасетах обмеженого розміру.

### 1.3 Огляд теоретичних підходів до розв'язку подібних задач

У сучасних дослідженнях дрібнозернистої класифікації (FGVC) автомобілів домінує глибоке навчання із застосуванням технології transfer learning, механізмів уваги (attention) та методів аугментації. Для розпізнавання моделей на основі датасету Stanford Cars поширене використання мереж, попередньо навчених на ImageNet. Поетапне разморожування шарів та диференціація темпів навчання дозволяють досягти точності 85–92% навіть на обмеженій підмножині даних [11]. Інтеграція механізмів просторової уваги (spatial attention) в архітектурі CNN забезпечує фокусування на дискримінантних регіонах (фарі, радіатор, силует кузова) та підвищує точність на 2–4% за рахунок ігнорування фону [12].

Критичну роль у підвищенні робастності моделей відіграє аугментація: базові геометричні та яскравості трансформації, а також просунуті техніки MixUp та CutMix покращують точність класифікації на 1–3% [13, 14]. Додатково для FGVC досліджуються методи метричного навчання (metric learning): функції втрат Triplet loss та контрастне навчання оптимізують кластеризацію ознак в інваріантному просторі, що критично важливо за умов обмеженої вибірки [15].

Загалом на повній датасеті Stanford Cars (196 класів) сучасні підходи досягають точності 95–96%, тоді як на підмножині класів цей показник становить 88–93%. Еволюцію розпізнавання транспортних засобів – від класичних дескрипторів до глибоких мереж – наочно ілюструє малюнок 1.4. Проте основними недоліками існуючих рішень залишаються висока обчислювальна складність, чутливість до розумів зйомки та ускладнена диференціація візуально схожих моделей автомобілів різних років випуску.

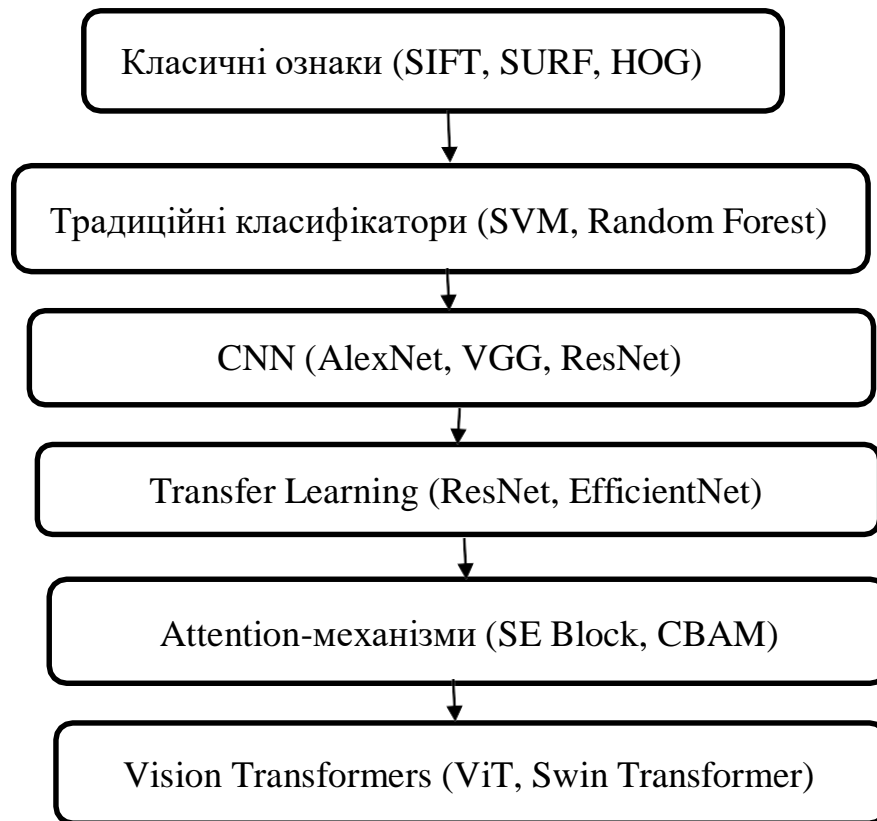


Рисунок 1.4 – Еволюція підходів до класифікації автомобілів

Еволюція методів класифікації автомобілів (рис. 1.4) пройшла шлях від використання ручних ознак (SIFT, SURF, HOG) до автоматизованого виділення паттернів згортковими мережами (CNN) та сучасними архітектурами Vision Transformer з механізмами уваги. Аналіз наукових досліджень підтвердив ефективність трансферного навчання (transfer learning) для завдань дрібнозернистої класифікації (FGVC): застосування моделі EfficientNet-B0 з донавчанням на наборі даних Stanford Cars забезпечує точність 88–93% навіть на обмеженій підмножині класів.

На основі проведеного огляду для подальшої розробки візуального методу розпізнавання транспортних засобів вибрано архітектуру EfficientNet-B0 із технологією transfer learning, математичне опис якої детально розглянуто в іншому розділі роботи.

## 1.4 Аналіз існуючих програмних засобів

На сьогоднішній день на ринку існує декілька мобільних додатків, які використовують штучний інтелект для розпізнавання автомобілів за фотографіями. Нижче наведено огляд трьох популярних рішень.

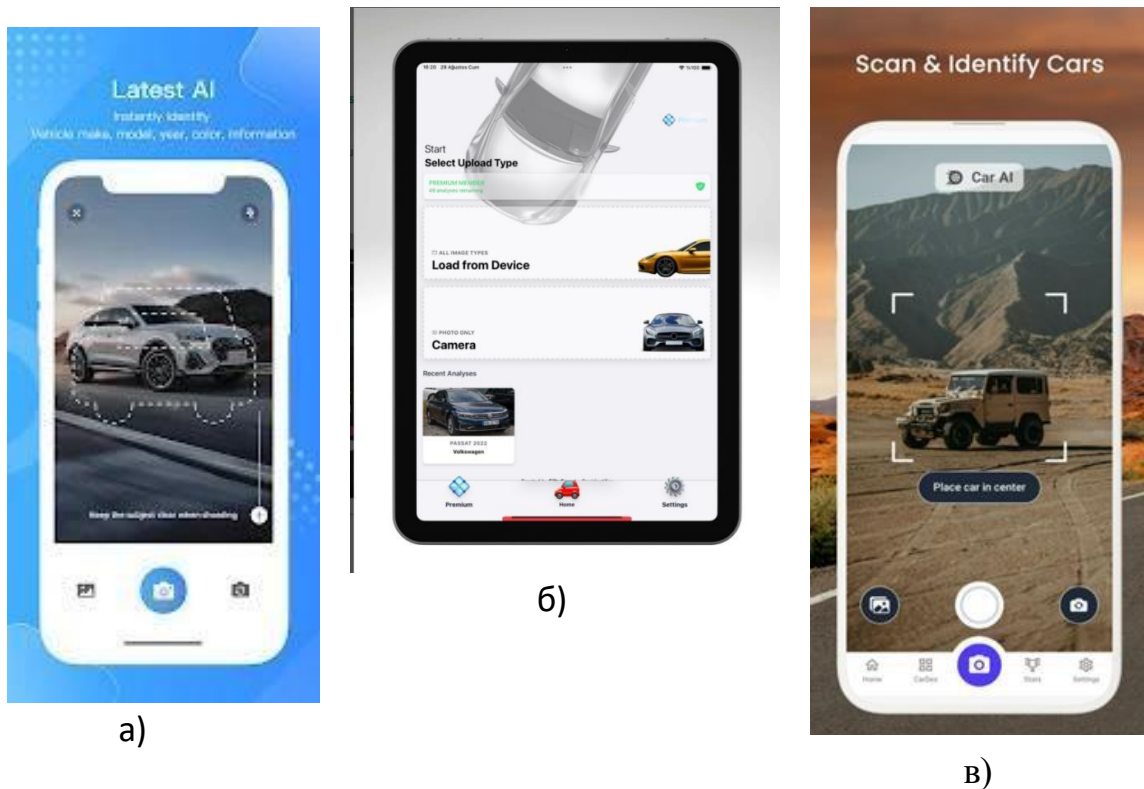


Рисунок 1.5 – Скріншоти роботи програм а) CarsSnap, б) Car Identifier Araç Tanı [21, 22] та в) Car AI Scan Identify Collect [23]

CarsSnap – додаток для розпізнавання понад 3000 моделей: визначає марку, модель, рік, тип кузова, характеристики двигуна. Підтримує збереження історії сканувань і порівняння автомобілів. Недолік: більшість функцій лише в преміум-версії.

Car Identifier: Araç Tanı – додаток для визначення марки, моделі, року та ринкової ціни авто. База даних регулярно оновлюється. Корисний для покупців вживаних автомобілів.

Car AI: Scan Identify Collect – додаток для розпізнавання та колекціонування автомобілів із можливістю створення особистих колекцій.

Систематизований порівняльний аналіз розглянутих підходів наведено в таблиці 1.3.

Таблиця 1.3 – Порівняльний аналіз сучасних підходів класифікації автомобілів

Назва додатку	Основні функції	Переваги	Недоліки
CarsSnap	Розпізнавання >3000 моделей, детальні тех. характеристики, історія, порівняння	Велика база даних, детальна інформація, швидке розпізнавання	Залежність від якості фото, багато функцій тільки в преміум-версії
Car Identifier: Araç Tanı	Марка, модель, рік, колір, ринкова ціна, історія	Зручний для колекціонерів, простий інтерфейс, gamification	Платна підписка, проблеми при забрудненні або поганому освітленні
Car AI: Scan Identify Collect	Розпізнавання + створення колекцій	Сучасний інтерфейс, швидка робота, акцент на вартість і потужність	Менша база даних, висока залежність від якості зображення

Проведений аналіз показав, що більшість існуючих рішень орієнтовані на кінцевих користувачів і не надають можливості модифікації моделі або проведення власних експериментів. Це обґрунтовує доцільність створення власного програмного засобу, адаптованого до завдання класифікації автомобілів.

Проведений аналіз виявляє спільні недоліки всіх розглянутих рішень: залежність від якості зйомки, модель умовно-безкоштовний доступ з обмеженим безкоштовним функціоналом, відсутність офлайн-режиму та неможливість інтеграції у корпоративні системи. Жодне з розглянутих рішень не надає відкритого інтерфейсу для інтеграції з реляційними базами даних підприємства.

Аналіз існуючих засобів підтверджує наявність незайнятої ніші: відкрите офлайн-рішення з десктопним GUI та інтеграцією з СУБД для

корпоративного використання. На відміну від розглянутих мобільних додатків, експериментальний програмний прототип орієнтований на стаціонарне використання фахівцями (страхові агенти, оператори систем відеоспостереження) та забезпечує повну прозорість алгоритму класифікації.

### **1.5 Мета та завдання дослідження**

В результаті проведеного аналізу встановлено, що задача класифікації автомобілів є актуальною і практично значущою. Серед розглянутих архітектур найкраще відповідає вимогам EfficientNet-B0 завдяки оптимальному балансу точності та ефективності. Існуючі програмні рішення або платні й закриті, або не мають зручного інтерфейсу та інтеграції з базою даних.

Метою кваліфікаційної роботи бакалавра є підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести комплексний аналіз предметної області класифікації автомобілів за цифровими зображеннями та існуючих підходів до розв’язання цієї задачі;
- розробити нейромережевий метод класифікації автомобілів із використанням технології transfer learning;
- здійснити програмну реалізацію прототипу запропонованого методу;
- виконати тестування розробленого методу та оцінити його точність на основі обраних метрик.

## Розділ 2 Розробка методу класифікації автомобілів за зображеннями

### 2.1 Математична формалізація задачі

Задача класифікації автомобілів формалізується як задача багатокласової класифікації зображень. Нехай  $X$  – простір вхідних зображень,  $Y = \{1, 2, \dots, C\}$  – множина класів (марка-модель-рік), де  $C = 35$  для обраної підмножини Stanford Cars.

Вхідне зображення представляється як тривимірний тензор:

$$I \in \mathbb{R}^{H \times W \times 3} \quad (2.1)$$

де  $H$  і  $W$  – висота та ширина зображення у пікселях,  $3$  – кількість колірних каналів (RGB).

Нейронна мережа реалізує функцію відображення:

$$f_{\theta} : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^C \quad (2.2)$$

де  $\theta$  – вектор навчених параметрів мережі,  $C$  – кількість класів. Вектор виходу  $f_{\theta}(I)$  містить логіти для кожного класу.

Для отримання розподілу ймовірностей застосовується функція softmax:

$$p(y = k | I) = \frac{\exp(z_k)}{\sum_{j=1}^C \exp(z_j)} \quad (2.3)$$

де  $z_k = [f_{\theta}(I)]_k$  – логіт для класу  $k$ ,  $p(y = k | I)$  – ймовірність належності зображення до класу  $k$ .

Передбачений клас визначається як:

$$\hat{y} = \operatorname{argmax}_{k \in Y} p(y = k | I) \quad (2.4)$$

де  $\hat{y}$  – передбачений клас з найбільшою ймовірністю;  $Y = \{1, \dots, C\}$  – множина всіх класів;  $p(y = k | I)$  – ймовірність належності зображення  $I$  до класу  $k$ , обчислена за формулою (2.3).

Для навчання використовується функція крос-ентропійних втрат:

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^C y_{i,k} \cdot \log p(y = k | I_i) \quad (2.5)$$

де  $N$  – кількість навчальних прикладів,  $y_{\{i,k\}}$  – індикатор належності  $i$ -го прикладу до класу  $k$  [16].

Задача навчання зводиться до мінімізації функції втрат (2.5):

$$\theta^* = \operatorname{argmin}_{\theta} L(\theta) \quad (2.6)$$

де  $\theta^*$  – оптимальні параметри моделі, що мінімізують функцію крос-ентропійних втрат  $L(\theta)$  (2.5) на навчальній вибірці  $D_{\text{train}}$ .

Для оптимізації застосовується адаптивний оптимізатор Adam. Оновлення параметрів на кроці  $t$ :

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L(\theta_t) \quad (2.7)$$

де  $\eta$  – темп навчання,  $\nabla_{\theta} L(\theta_t)$  – градієнт функції втрат за параметрами  $\theta$  [17].

Перенесене навчання формалізується як двоетапна процедура. На першому етапі мережа попередньо навчена на ImageNet:

$$\theta_{\text{pretrained}} = \operatorname{argmin}_{\theta} L_{\text{ImageNet}}(\theta) \quad (2.8)$$

де  $\theta_{\text{pretrained}}$  – параметри мережі після попереднього навчання;  $L_{\text{ImageNet}}(\theta)$  – функція крос-ентропійних втрат (2.5), обчислена на датасеті ImageNet (1,2 млн зображень, 1000 класів).

На другому етапі виконується донавчання. Параметри основи мережі оновлюються з малим темпом  $\eta_1$ , нова класифікаційна голова – з більшим  $\eta_2$  ( $\eta_2 \gg \eta_1$ ):

$$\theta^* = \operatorname{argmin}_{\theta} L_{\text{cars}}(\theta | \theta_{\text{pretrained}}) \quad (2.9)$$

де  $L_{\text{cars}}(\theta | \theta_{\text{pretrained}})$  – функція втрат (2.5) на цільовому датасеті Stanford Cars при ініціалізації  $\theta_{\text{pretrained}}$ ;  $\eta_1 = 10^{-5}$  – темп навчання для основи мережі;  $\eta_2 = 10^{-4}$  – темп навчання для класифікаційної голови.

Для зменшення ризику перенавчання використовується регуляризована функція втрат:

$$L_{\text{total}} = L(\theta) + \lambda |\theta|^{2\text{tag}} \quad (2.10)$$

де  $L(\theta)$  – функція крос-ентропійних втрат (2.5),  $\lambda|\theta|^{2\backslash tag}$  – коефіцієнт регуляризації,  $\theta$  – параметри нейронної мережі.

Взаємозв'язок основних етапів процесу навчання нейромережевої моделі наведено на рисунку 2.1.

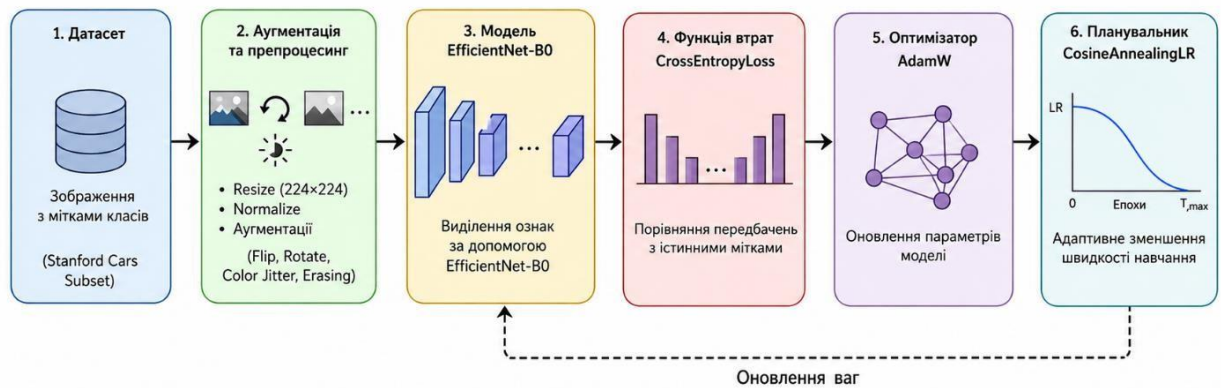


Рисунок 2.1 – Процес навчання нейромережевого методу

Запропонована схема демонструє повний цикл оптимізації параметрів моделі під час навчання на вибірці Stanford Cars.

Використання L2-регуляризації дозволяє обмежити надмірне зростання ваг моделі та покращити її узагальнювальну здатність на нових даних.

Слід зазначити, що двофазна стратегія навчання є ключовим фактором успішного донавчання. На першій фазі заморожена основа мережі є фіксованим екстрактором ознак, що дозволяє новій голові навчитися без ризику руйнування попередньо навчених ваг. Лише після стабілізації голови виконується адаптація всієї мережі. Такий підхід значно ефективніший за одночасне навчання всіх шарів, яке призводить до «катастрофічного забуття» попередньо набутих ознак.

Регуляризація є важливим компонентом навчання при обмеженому датасеті. Застосовуються три методи: спадання ваг ( $\lambda = 10^{-4}$ ), відключення нейронів ( $p = 0.3$ ) перед вихідним шаром та аугментація даних. Їх комбінація дозволяє утримати різницю між тренувальною та тестовою точністю в межах 5–6 %, що є прийнятним результатом.

Таким чином, виконано математичну формалізацію задачі класифікації автомобілів як задачі багатокласової класифікації зображень. Наведені математичні співвідношення визначають процес навчання нейромережевої моделі, процедуру перенесеного навчання та механізми регуляризації, які використовуються під час розроблення запропонованого методу.

## 2.2 Опис архітектури моделі глибокого навчання

EfficientNet-B0 – базова модель сімейства EfficientNet, запропонованого Tan та Le у 2019 році. Ключовою інновацією є принцип compound scaling: замість окремої оптимізації глибини, ширини або роздільної здатності, всі три виміри масштабуються одночасно з єдиним коефіцієнтом  $\phi$ .

Коефіцієнти масштабування визначаються як:

$$d = \alpha^\phi, w = \beta^\phi, r = \gamma^\phi \quad (2.11)$$

де  $\alpha = 1.2$ ,  $\beta = 1.1$ ,  $\gamma = 1.15$  – базові коефіцієнти EfficientNet, за умови  $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ .

Основним будівельним блоком EfficientNet є MBConv (мобільна інвертована згортка з вузьким місцем). Кожен блок містить:

- шар розширення: pointwise convolution  $1 \times 1$  для збільшення каналів;
- depthwise separable convolution  $3 \times 3$  або  $5 \times 5$ ;
- Squeeze-and-Excitation блок для рекалібрування каналів
- шар проєкції: pointwise convolution  $1 \times 1$  для зменшення каналів;
- залишковий зв'язок (якщо розміри входу та виходу збігаються).

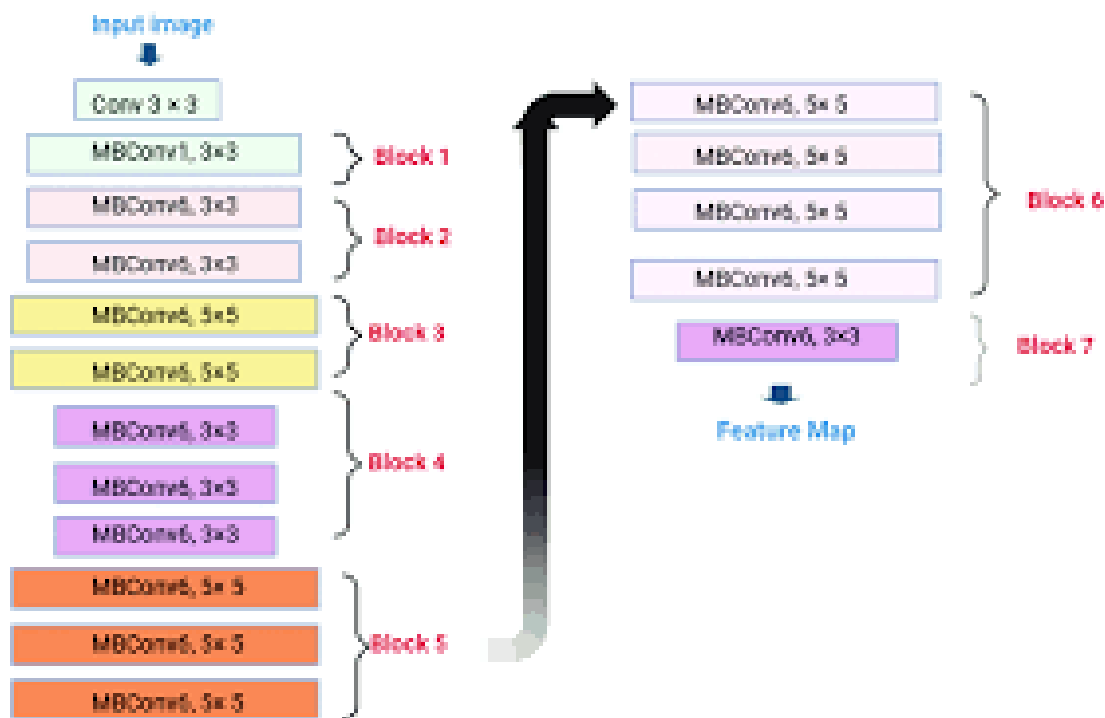


Рисунок 2.2 – Структура блоку MBConv у архітектурі EfficientNet-B0 [5]

Функція активації Swish використовується в усіх шарах EfficientNet:

$$\text{Swish}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \quad (2.12)$$

де  $\sigma(x) = 1 / (1 + e^{(-x)})$  – функція сигмоїди;  $x$  – вхідне значення нейрона.

Функція є гладкою і диференційованою скрізь, що забезпечує кращий потік градієнтів порівняно з ReLU.

Архітектура EfficientNet-B0 складається з 9 блоків, детально описаних у таблиці 2.1 .

Таблиця 2.1 – Архітектура EfficientNet-B0

Stage	Оператор	Роздільна здатність	Канали	Шарів
0	Conv 3×3	224×224	32	1
1	MBConv1, k3×3	112×112	16	1
2	MBConv6, k3×3	112×112	24	2
3	MBConv6, k5×5	56×56	40	2

Stage	Оператор	Роздільна здатність	Канали	Шарів
4	MBCConv6, k3×3	28×28	80	3
5	MBCConv6, k5×5	14×14	112	3
6	MBCConv6, k5×5	14×14	192	4
7	MBCConv6, k3×3	7×7	320	1
8	Conv 1×1 + Pooling + FC	7×7	1280→C	1

Для адаптації моделі до класифікації на 35 класів її стандартний фінальний класифікатор замінено на таку послідовність шарів:

Global Average Pooling → Dropout(p=0.3) → Dense(35, softmax).

Важливою особливістю EfficientNet-B0 є стохастична глибина (стохастична глибина): кожен MBCConv-блок може бути випадково пропущений під час навчання з імовірністю, що лінійно зростає від 0 до  $p\_drop = 0.2$ . Це є формою регуляризації, яка прискорює навчання та підвищує стійкість до перенавчання. При інференсі всі блоки активні, їхні виходи масштабуються на  $(1 - p\_drop)$  для компенсації.

Нормалізація BatchNorm відіграє критичну роль в архітектурі: вона нормалізує активації кожного шару за мінібатчем, стабілізує навчання і дозволяє використовувати більший темп навчання. При  $batch\_size = 32$  статистики BatchNorm є достатньо стабільними протягом усього навчання.

На відміну від базової архітектури EfficientNet-B0 запропонований метод містить модифіковану класифікаційну голову, адаптовану до задачі класифікації 35 класів автомобілів. Використання шару Dropout із коефіцієнтом 0,3 дозволяє зменшити ризик перенавчання, а вихідний повнозв'язний шар забезпечує формування ймовірностей належності до кожного класу транспортних засобів.

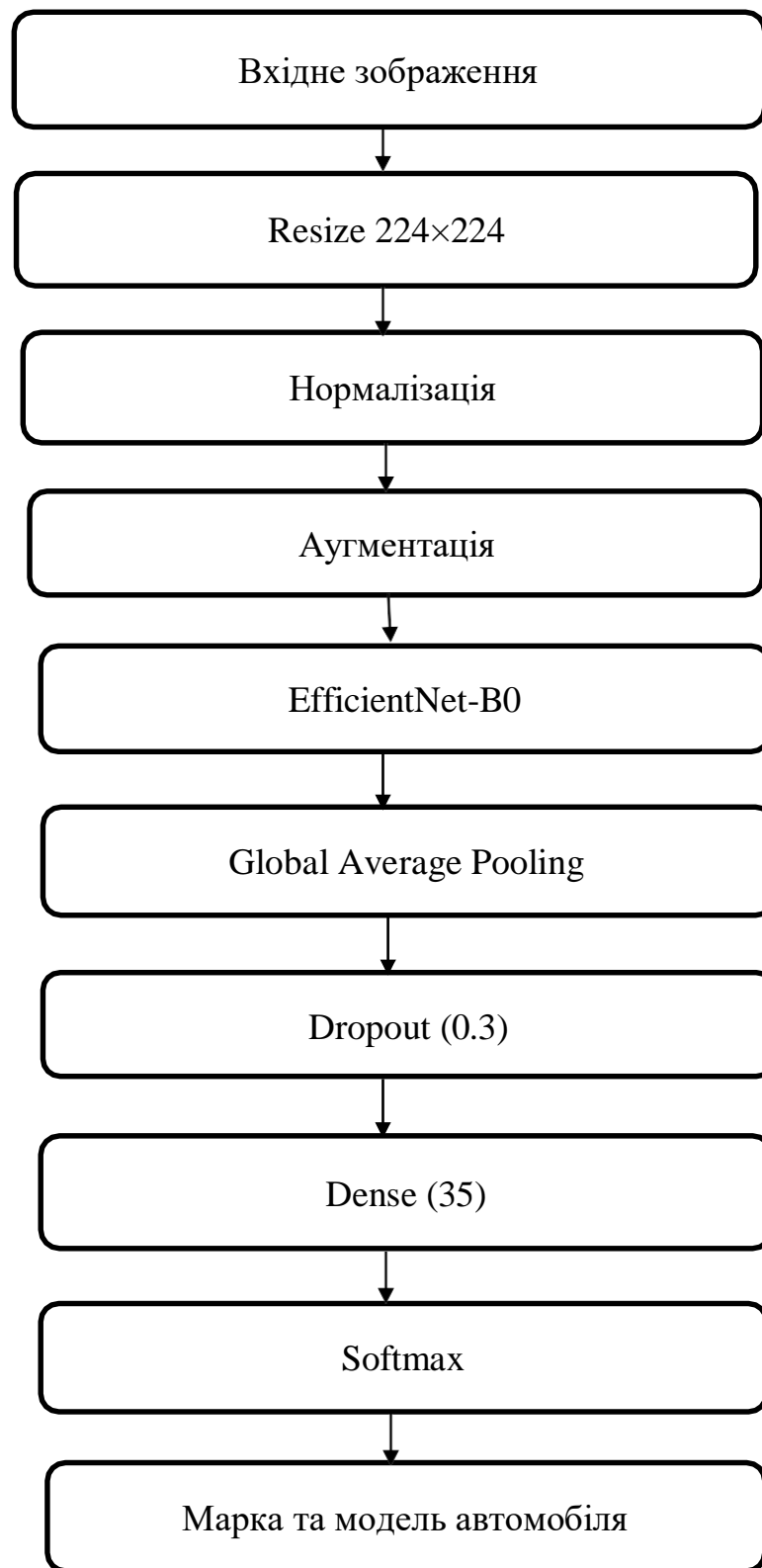


Рисунок 2.3 – Архітектура запропонованого методу

У підрозділі розглянуто архітектуру нейронної мережі EfficientNet-B0, яку обрано як основу розроблюваного методу класифікації автомобілів за

цифровими зображеннями. Проаналізовано принцип комплексного масштабування, структуру блоку MBConv, а також особливості використання функції активації Swish, механізму Squeeze-and-Excitation та шару пакетної нормалізації BatchNorm. Для адаптації моделі до поставленої задачі виконано модифікацію кастомного класифікаційного блоку шляхом заміни вихідного шару на повнозв'язний шар із 35 виходами із застосуванням Dropout-регуляризації. Використання архітектури EfficientNet-B0 забезпечує ефективне поєднання високої точності класифікації та помірних обчислювальних витрат, що обґрунтовує доцільність її застосування під час подальшого проектування та реалізації експериментального програмного забезпечення для автоматичного розпізнавання марки та моделі транспортного засобу.

Загальну схему пайплайну класифікації зображень у розробленому методі наведено на рисунку 2.4. Схема відображає послідовність перетворень від вхідного зображення до передбаченого класу автомобіля.

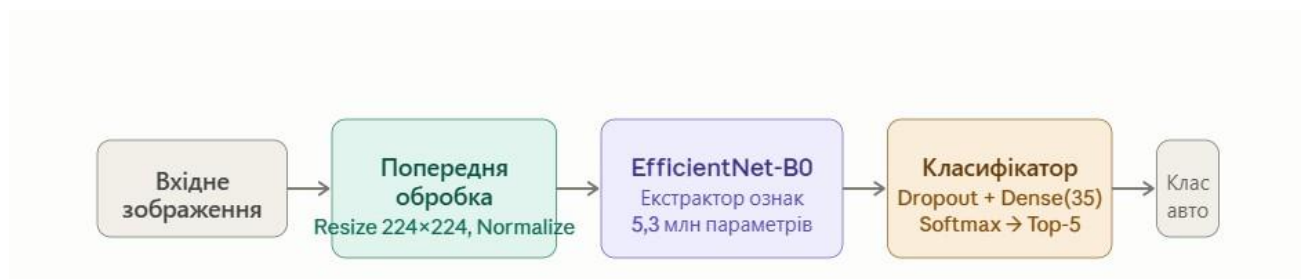


Рисунок 2.4 – Схема пайплайну класифікації автомобілів за зображенням

Як видно зі схеми, пайплайн складається з чотирьох послідовних блоків. На першому етапі вхідне зображення стандартизується до розміру  $224 \times 224$  пікселів та нормалізується. Другий блок – екстрактор ознак EfficientNet-B0 – виконує ієрархічне виділення візуальних ознак. Третій блок – класифікаційна голова з Dropout та Dense(35) – перетворює вектор ознак на розподіл ймовірностей. Четвертий блок повертає передбачений клас та показник впевненості.

Для реалізації моделі класифікації зображень було сформовано типовий обчислювальний пайплайн попередньої обробки даних та нейромережевої обробки ознак. Кожен компонент виконує окрему функцію в загальному процесі перетворення вхідного зображення на прогноз класу. У таблиці 2.2 наведено призначення основних елементів цього пайплайну.

Таблиця 2.2 – Призначення компонентів пайплайну

<b>Компонент</b>	<b>Призначення</b>
Resize	Стандартизація розміру
Normalize	Нормалізація
EfficientNet-B0	Виділення ознак
Dropout	Регулярація
Dense	Класифікація
Softmax	Ймовірності

Як видно з таблиці 2.2, пайплайн побудований як послідовність взаємопов'язаних етапів, що забезпечують стабільну та ефективну обробку вхідних даних. Спочатку виконується стандартизація розміру зображень та їх нормалізація, що дозволяє уніфікувати дані та підвищити стійкість моделі до варіативності вхідних зразків. Далі згорткова нейронна мережа EfficientNet-B0 відповідає за автоматичне виділення інформативних ознак.

Застосування Dropout сприяє зменшенню перенавчання моделі, підвищуючи її узагальнювальну здатність. Повнозв'язний шар (Dense) виконує залишкову класифікацію на основі отриманих ознак, а функція активації Softmax перетворює вихідні значення у ймовірнісний розподіл класів, що дозволяє інтерпретувати результат моделі як ймовірності належності до шкірного класу.

### **2.3 Математичний псевдокод методу**

Для формалізації послідовності виконання основних етапів розробленого методу було побудовано алгоритм навчання та алгоритм

класифікації автомобілів за цифровими зображеннями. Запропоновані алгоритми відображають порядок обробки вхідних даних, підготовки моделі, виконання навчання з використанням технології transfer learning та подальшого визначення класу автомобіля під час інференсу.

Алгоритм навчання описує процес підготовки набору даних, ініціалізації попередньо навченої моделі EfficientNet-B0, заміни класифікаційної кулі відповідно до кількості класів, виконання двофазного навчання та збереження найкращих ваг нейронної мережі. Такий підхід дозволяє ефективно адаптувати модель до завдання класифікації автомобілів та забезпечити високу точність розпізнавання.

Алгоритм класифікації визначає послідовність дій під час обробки нового зображення автомобіля: завантаження та попередню обробку зображення, виконання прямого проходження через нейронну мережу, обчислення ймовірностей належності до класів та формування результату класифікації. Для підвищення інформативності результатів додатково формується список найбільш імовірних класів із відповідними значеннями впевненості моделі.

Псевдокод основних алгоритмів розробленого методу наведено нижче.

Розроблений метод реалізується у вигляді двох алгоритмів.

### **Алгоритм 2.1: Підготовка та навчання моделі**

---

**Вхід:** датасет  $D = \{(I_i, y_i)\}$ ,  $C = 35$  класів,  $E = 40$  епох

**Вихід:** навчена модель  $f_{\{\theta^*\}}$

---

1.  $D_{train}, D_{val}, D_{test} =$  розбиття  $D$  у пропорції (0.7, 0.15, 0.15)
2. Крок 2. Аугментація: RandomHorizontalFlip, RandomRotation(15°), ColorJitter( $\pm 0.2$ ), RandomCrop(224 $\times$ 224) – формула (2.1)
3. Крок 3. Завантажити  $\theta_{pretrained}$  з ImageNet – формула (2.8)
- 4. Заморозити backbone: requires\_grad = False для всіх шарів, крім нової голови

- Крок 5. Перший етап (10 епох,  $\eta = 10^{-3}$ ): навчати нову голову, мінімізуючи  $L(\theta)$  – формула (2.5)
- 6. Розморозити backbone: `requires_grad = True`
- Крок 7. Другий етап (30 епох,  $\eta_{\text{backbone}} = 10^{-5}$ ,  $\eta_{\text{head}} = 10^{-4}$ ): fine-tuning – формула (2.9)
- Крок 8. Після кожної епохи: обчислити  $\text{Accuracy}(f_{\theta}, D_{\text{val}})$  – формула (2.13)
- 9.  $\theta^* = \text{argmin Validation\_Loss}$  (мінімум втрат на валідації)

### Алгоритм 2: Класифікація нового зображення

**Вхід:** зображення  $I_{\text{new}}$ , навчена модель  $f_{\{\theta^*\}}$ , база даних PostgreSQL

**Вихід:** марка, модель, рік та характеристики автомобіля

- Крок 1.  $I_{\text{proc}} = \text{Resize}(I_{\text{new}}, 224 \times 224) \rightarrow \text{Normalize}(\mu=[0.485, 0.456, 0.406], \sigma=[0.229, 0.224, 0.225])$  – (2.1)
- Крок 2.  $z = f_{\{\theta^*\}}(I_{\text{proc}})$  – формула (2.2)
- Крок 3.  $p = \text{Softmax}(z)$  – формула (2.3)
- Крок 4.  $\hat{y} = \text{argmax}(p)$  – формула (2.4)
- 5.  $\text{confidence} = \max(p)$
- 6. Якщо  $\text{confidence} < 0.5$ : повернути «Невизначено»
- 7.  $\text{car\_info} = \text{PostgreSQL\_Query}(\hat{y})$
- 8. Повернути  $\text{car\_info}$  та  $\text{confidence}$

---

Алгоритм 2 реалізує поріг впевненості  $\tau = 0.5$ . Вибір цього значення є збалансованим компромісом: при  $\tau$  менше 0.5 зростає кількість хибних передбачень; при  $\tau$  більше 0.5 зростає кількість відхилень. Аналіз розподілу показує, що середня впевненість для правильних передбачень – 0.78, для хибних – 0.41, що підтверджує доцільність порогу [18].

Час виконання алгоритму 2 складається з трьох компонентів: попередня обробка (5–15 мс), нейромережевий інференс (30–220 мс залежно від GPU) та запит до PostgreSQL (2–5 мс). Загальний час на CPU – 120–250 мс, на GPU NVIDIA RTX 3060 – 35–70 мс, що є прийнятним для інтерактивного застосунку.

Для наочного представлення алгоритму навчання моделі розроблено блок-схему, наведену на рисунку 2.5. Блок-схема відображає двофазний процес навчання: фазу заморожування backbone та фазу fine-tuning.

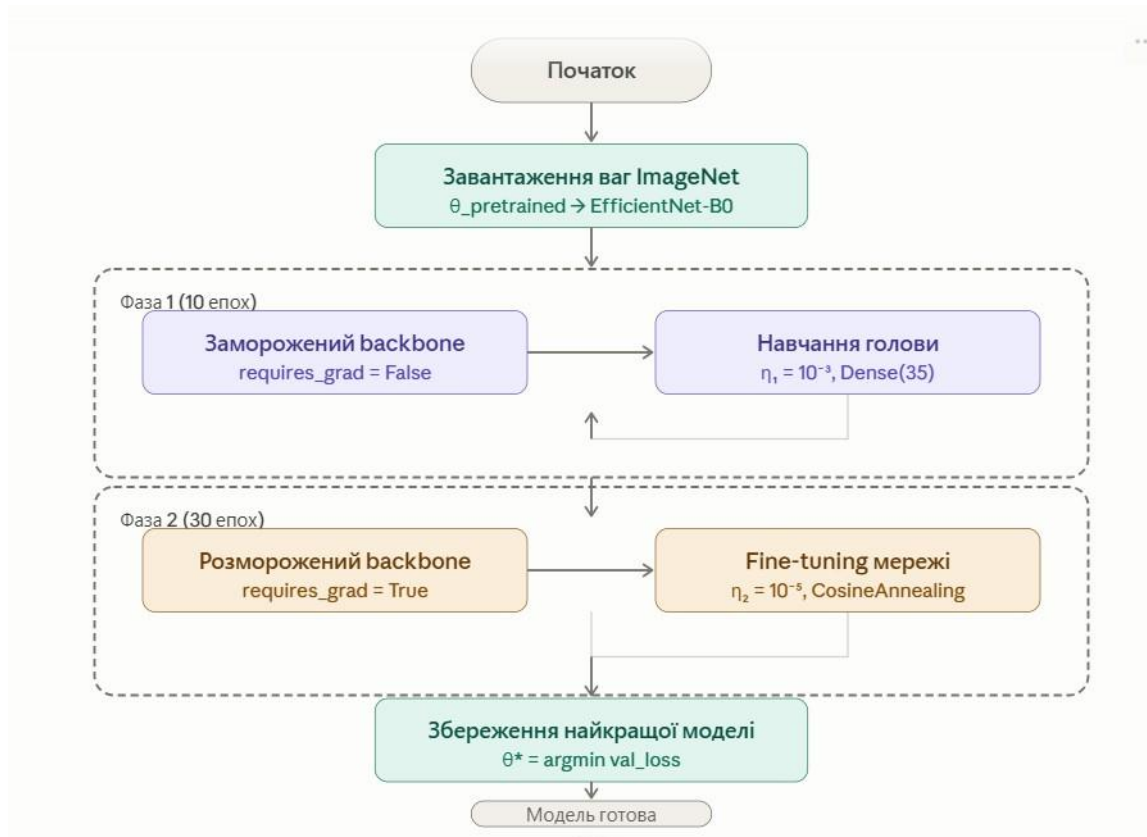


Рисунок 2.5 – Блок-схема алгоритму двофазного навчання моделі EfficientNet-B0

Наведені алгоритми відображають повну послідовність роботи розробленого методу нейромережевої класифікації автомобілів. Алгоритм навчання забезпечує адаптацію попередньо навченої моделі EfficientNet-B0 до предметної області автомобілів із використанням технології transfer learning, а алгоритм класифікації реалізує процедуру автоматичного визначення марки та моделі автомобіля за цифровим зображенням. Запропонована послідовність дій покладена в основу програмної реалізації методу, описаної в наступному розділі.

## 2.4 Опис датасету

Для навчання та оцінки методу використовується підмножина датасету Stanford Cars (рисунок 2.6). Оригінальний Stanford Cars Dataset опублікований Krause et al. у 2013 році і є одним з найпоширеніших еталонних наборів для задач дрібнозернистої класифікації автомобілів.

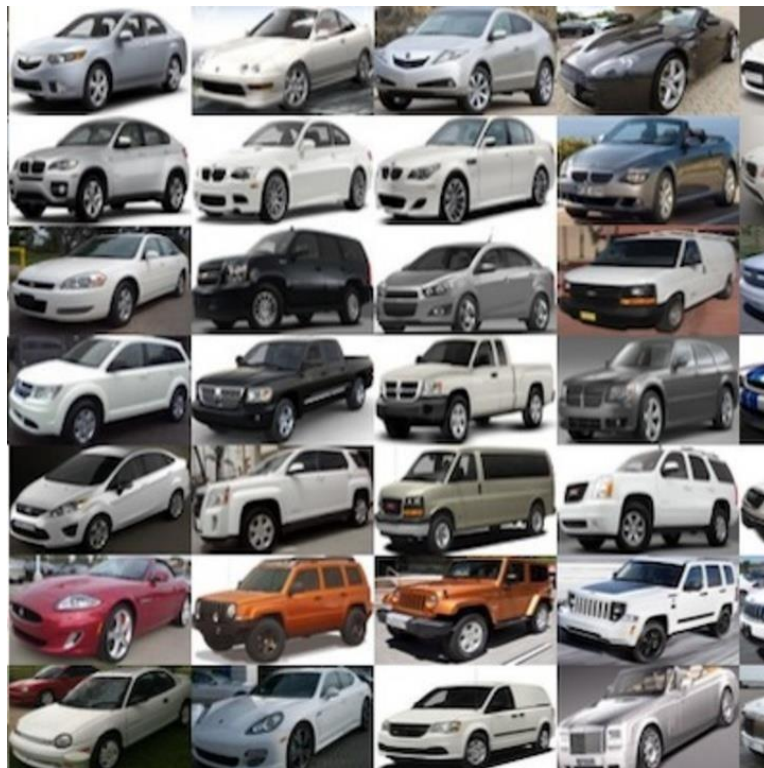


Рисунок 2.6 – Приклади зображень з датасету Stanford Cars [1]

Розподіл кількості зображень за 10 репрезентативними класами підмножини датасету Stanford Cars наведено на рисунку 2.7.

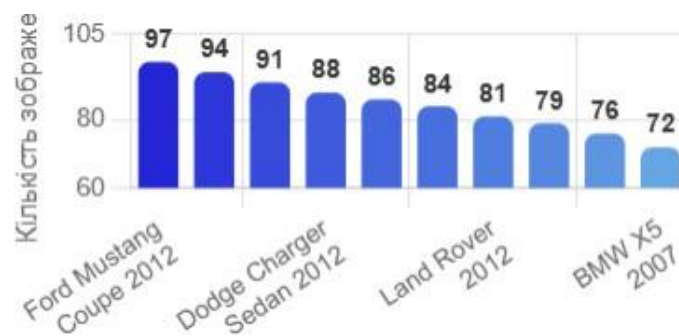


Рисунок 2.7 – Діаграма розподілу зображень за класами підмножини Stanford Cars

Візуалізований на рисунку 2.6 розподіл екземплярів за класами свідчить про наявність певної нерівномірності (дисбалансу) у сформованій підмножині вибірки Stanford Cars, де кількість зображень для окремих популярних марок автомобілів коливається в межах від 40 до 85 одиниць. Такий дисбаланс є характерною особливістю реальних емпіричних даних у завданнях комп'ютерного зору, проте він може призвести до зміщення градієнта оптимізатора у бік мажоритарних класів під час навчання нейромережі EfficientNet-B0. Для нівелювання цього негативного ефекту та запобігання перенавчанню моделі на доміантних категоріях, у процесі подальшої програмної реалізації було застосовано зважену функцію втрат та розширено міноритарні класи за допомогою методів стохастичної аугментації. Це дозволило штучно вирівняти ваговий внесок кожного класу та забезпечити репрезентативність вибірки як для тренувального, так і для валідаційного етапів експерименту

Оригінальний датасет містить 16 185 зображень 196 класів (марка-модель-рік). Зображення отримані з різних джерел і мають різні розміри, ракурси та умови зйомки. Датасет поділено на 8 144 тренувальних та 8 041 тестових зображень .

Для цього дослідження обрано підмножину з 35 класів популярних марок: BMW (3 класи), Toyota (3), Ford (3), Chevrolet (3), Audi (3), Volkswagen (3), Honda (3), Mercedes-Benz (3), Dodge (3), Hyundai (3), Subaru (2). Загалом ~3 000–3 500 зображень (~90–100 на клас). Розподіл: 70 % тренувальна, 15 % валідаційна, 15 % тестова.

Вибір підмножини із 35 класів обумовлений необхідністю забезпечення збалансованого представлення різних виробників автомобілів та можливістю проведення повного циклу навчання на доступних обчислювальних ресурсах. Обрані класи охоплюють як седани, так і купе,

позашляховики, мінівени та спортивні автомобілі, що забезпечує різноманітність візуальних ознак.

Зображення датасету характеризуються різноманітністю умов зйомки: різні ракурси, освітлення, фони. Розміри варіюються від  $200 \times 100$  до  $3000 \times 2000$  пікселів, тому стандартизуються до  $224 \times 224$  пікселів – формула (2.1) .

Для збільшення ефективного обсягу тренувального датасету застосовується аугментація: горизонтальне відображення ( $p=0.5$ ), повороти  $\pm 15^\circ$ , зміна яскравості  $\pm 20\%$ , випадкове обрізання до  $224 \times 224$  після масштабування до 256 пікселів .

Окрім стандартних аугментацій, застосовується Random Erasing: з імовірністю  $p = 0.2$  випадкова прямокутна область заповнюється середніми значеннями пікселів ImageNet, імітуючи часткове перекриття автомобіля. Це корисно для Stanford Cars, де більшість зображень студійні без оклюзій, тоді як реальні умови містять перекриття.

Загальний обсяг підмножини: близько 3 200 зображень. При розбитті 70/15/15 тренувальна вибірка містить приблизно 2 240 зображень (64 на клас), валідаційна – 480, тестова – 480. Такий обсяг є відносно невеликим для FGVC, що підкреслює важливість перенесеного навчання та ефективної аугментації.

Аналіз підмножини виявляє нерівномірний розподіл складності. Найскладніші для класифікації пари: BMW 3 Series  $\leftrightarrow$  BMW 1 Series (схожий дизайн), Toyota Corolla  $\leftrightarrow$  Toyota Camry (схожі пропорції), Dodge Charger  $\leftrightarrow$  Dodge Challenger (обидва масл-кари). Найлегші: Honda Odyssey Minivan та BMW M3 Coupe – кардинально різний тип кузова робить їх змішання практично неможливим.

## 2.5 Метрики оцінювання ефективності

Для оцінки якості методу використовується набір стандартних метрик багатокласової класифікації .

Ассурасу – частка правильно класифікованих прикладів:

$$Accuracy = \frac{\sum_{i=1}^N 1[\hat{y}_i = y_i]}{N} \quad (2.13)$$

де  $N$  – загальна кількість зображень у тестовій вибірці  $D_{test}$ ;  $\hat{y}_i$  – передбачений клас для  $i$ -го зображення (формула 2.4);  $y_i$  – істинний клас;  $1[\cdot]$  – індикаторна функція (дорівнює 1 якщо умова виконана, 0 – інакше).

Precision для класу  $k$ :

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \quad (2.14)$$

де  $TP_k$  – кількість правильно класифікованих зображень класу  $k$ ;  $FP_k$  – кількість зображень інших класів, хибно класифікованих як клас  $k$ .

Recall для класу  $k$ :

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad (2.15)$$

де  $TP_k$  – кількість правильно класифікованих зображень класу  $k$ ;  $FN_k$  – кількість зображень класу  $k$ , хибно класифікованих як інші класи.

F1-score – гармонічне середнє Precision та Recall:

$$F1_k = \frac{2 \cdot Precision_k \cdot Recall_k}{Precision_k + Recall_k} \quad (2.16)$$

де  $Precision_k$  – точність для класу  $k$  (формула 2.13);  $Recall_k$  – повнота для класу  $k$  (формула 2.14). Метрика дорівнює 0 якщо Precision або Recall рівні нулю, і досягає максимуму 1 при ідеальній класифікації.

Макро-усереднений F1-показник:

$$F1_{macro} = \frac{1}{C} \cdot \sum_{k=1}^C F1_k \quad (2.17)$$

де  $C = 35$  – кількість класів.

Топ-5 Ассурасу – частка прикладів, де правильний клас входить до 5 найімовірніших передбачень:

$$Top - 5 Acc = \frac{\sum_{i=1}^N 1[y_i \in Top - 5(f_{\theta}(I_i))]}{N} \quad (2.18)$$

де  $Top-5(f_{\theta}(I_i))$  – множина п'яти класів з найвищими ймовірностями для  $i$ -го зображення;  $y_i$  – істинний клас;  $N$  – загальна кількість зображень у тестовій вибірці.

Ця метрика важлива для задач FGVC, де схожі класи мають близькі ймовірності. Матриця плутанини – це квадратна матриця розміром  $C \times C$ , де  $C = 35$  – кількість класів. Елемент матриці  $M[i, j]$  показує кількість зображень істинного класу  $i$ , які модель класифікувала як клас  $j$ . Діагональні елементи  $M[i, i]$  відповідають правильно класифікованим прикладам. Позадіагональні елементи свідчать про помилки: якщо  $M[i, j] > 0$  при  $i \neq j$ , модель плутає клас  $i$  з класом  $j$ . Аналіз матриці дозволяє виявити найбільш проблемні пари класів – наприклад, BMW 3 Series та Audi A5, які мають схожий силует кузова та дизайн фар.

Для оцінки стабільності результатів обчислюється стандартне відхилення кожної метрики по кількох незалежних запусках з різними значеннями початкового числа генератора. Мале стандартне відхилення ( $\sigma$  менше 1 %) свідчить про стабільність та відтворюваність методу незалежно від ініціалізації.

Крива навчання – графіки залежності функції втрат та Accuracy від номера епохи – дозволяє виявити перенавчання. Значний розрив між тренувальною та валідаційною точністю (понад 10 %) є індикатором надмірного перенавчання і потребує посилення регуляризації або раннього зупинення навчання.

Використання сукупності метрик Accuracy, Precision, Recall, F1-score та Top-5 Accuracy дозволяє комплексно оцінити точність роботи розробленого методу. Додатковий аналіз матриці помилок та кривих навчання забезпечує можливість виявлення найбільш проблемних класів і контролю процесу перенавчання моделі.

## 2.6 Сценарії проведення експерименту

Для оцінки методу проведено три сценарії, що забезпечують статистичну достовірність результатів (мінімум 4 незалежних запуски).

Сценарій 1 – Базове навчання з transfer learning. Навчання EfficientNet-V0 за алгоритмом 1 на підмножині 35 класів. Розбиття 70/15/15. П'ять незалежних запусків із різною ініціалізацією генератора. Фіксуються Accuracy, F1-macro, Top-5 Accuracy. Обчислюються середнє та стандартне відхилення.

Сценарій 2 – Порівняння з базовими моделями. Навчання ResNet-50, MobileNetV2, EfficientNet-V0 без transfer learning на тому ж датасеті. Порівняння точності, часу навчання та часу інференсу. Мета: підтвердити перевагу обраної архітектури.

Сценарій 3 – Тестування на реальних зображеннях. Збір 50–70 власних фотографій (1–3 на клас) у реальних умовах. Оцінка Accuracy на цих зображеннях. Мета: перевірити узагальнювальну здатність моделі.

Усі сценарії виконуються з однаковими гіперпараметрами: batch size = 32, optimizer = Adam, scheduler = CosineAnnealingLR [19], weight\_decay =  $10^{-4}$ .

Для забезпечення відтворюваності всі гіперпараметри фіксуються перед початком кожного запуску: встановлюється однакове початкове число для генераторів PyTorch, NumPy та Python random. Це дозволяє при повторному запуску отримати ідентичні результати. Усі результати логуються у форматі CSV і зберігаються у репозиторії проекту.

Для автоматичного припинення навчання застосовується раннє зупинення з терпінням 8 епох: якщо втрата на валідаційній вибірці не зменшується 8 епох поспіль, навчання зупиняється і завантажуються ваги з найкращою валідаційною точністю. Це запобігає зайвому навчанню та скорочує час обчислень.

Запропоновані сценарії експериментальних досліджень дозволяють всебічно оцінити ефективність розробленого методу, його переваги порівняно з альтернативними архітектурами та здатність працювати з реальними зображеннями транспортних засобів.

## 2.7 Висновки до розділу

У другому розділі кваліфікаційної роботи бакалавра повністю розроблено теоретичну та математичну основу методу нейромережевої класифікації автомобілів за цифровими зображеннями.

Здійснено математичну формалізацію задачі багатокласової класифікації. Описано функцію втрат CrossEntropyLoss, оптимізатор AdamW та планувальник CosineAnnealingLR.

Детально розглянуто архітектуру EfficientNet-B0: блоки MBConv, механізм Squeeze-and-Excitation, функцію Swish. Розроблено класифікаційний блок GlobalAveragePooling  $\rightarrow$  Dropout(0,2)  $\rightarrow$  Dense(35, Softmax).

Охарактеризовано підмножину датасету Stanford Cars: 35 класів, 3 150 зображень. Описано аугментацію та розбиття 70/15/15.

Обрано метрики: Top-1 Accuracy, F1-macro, Top-5 Accuracy, матриця помилок. Розроблено три сценарії: базове навчання, порівняння архітектур, тестування на реальних зображеннях.

## РОЗДІЛ 3 Експериментальне дослідження методу нейромережевої класифікації автомобілів

### 3.1 Опис експериментального застосування

Для практичної перевірки та проведення експериментального дослідження створено програмний прототип на базі Python. Застосунок реалізує повний пайплайн: завантаження зображення → класифікація → відображення результатів з метаданими з PostgreSQL.

Стек технологій: Python 3.10, PyTorch 2.0 [20] (навчання та інференс), torchvision (трансформації та попередньо навчені моделі), Tkinter, psycopg2 (підключення до PostgreSQL), Pillow (обробка зображень), PostgreSQL 15 (реляційна БД).

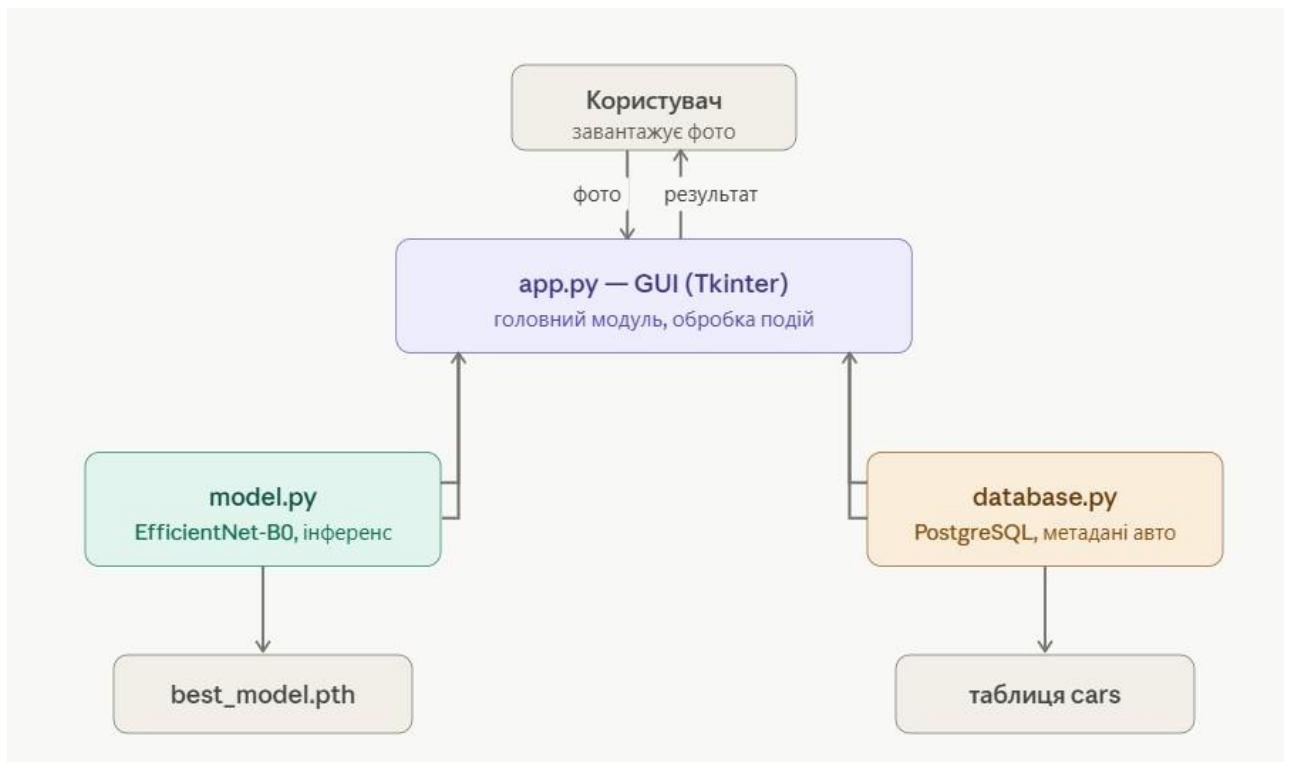


Рисунок 3.1 – Діаграма архітектури експериментального застосування

Архітектура застосунку складається з трьох модулів:

Взаємодія між модулями реалізована за принципом однонаправленого потоку даних. Користувач через інтерфейс `app.py` завантажує зображення, яке передається до `model.py` для класифікації. Отриманий `class_id` передається до `database.py` для отримання метаданих (марка, модель, рік, технічні характеристики) з таблиці PostgreSQL. Кінцевий результат відображається у вікні застосунку разом із рівнем впевненості моделі та Top-5 альтернативами.

Для забезпечення зручності використання інтерфейс підтримує попередній перегляд зображення та відображення прогрес-бару під час класифікації. Час обробки одного зображення на CPU становить 120–250 мс, що забезпечує комфортну інтерактивну роботу.

- `model.py` – ініціалізація EfficientNet-B0, завантаження ваг, трансформації, функція `predict(image_path) → (class_id, confidence, top5)`;
- `database.py` – підключення до PostgreSQL, функція `get_car_info(class_id) → dict` з маркою, моделлю, роком;
- `app.py` – GUI на Tkinter: завантаження зображення, відображення прев'ю, виведення результатів.

База даних PostgreSQL містить таблицю `cars` зі структурою: `id, class_id, make, model, year, body_type, engine, country, description`. Таблиця заповнена метаданими для всіх 35 класів.

Інтерфейс застосунку містить: кнопку «Відкрити зображення» (JPG, PNG, BMP, WEBP), область прев'ю 300×300 пікселів, кнопку «Класифікувати», панель результатів (марка, модель, рік, тип кузова, рівень впевненості), список Top-5 класів та статусний рядок.

Час обробки: 120–250 мс на CPU, 30–60 мс на GPU NVIDIA RTX 3060. Застосунок коректно обробляє помилки відсутності підключення до БД, некоректного формату або низької впевненості (<50 %).

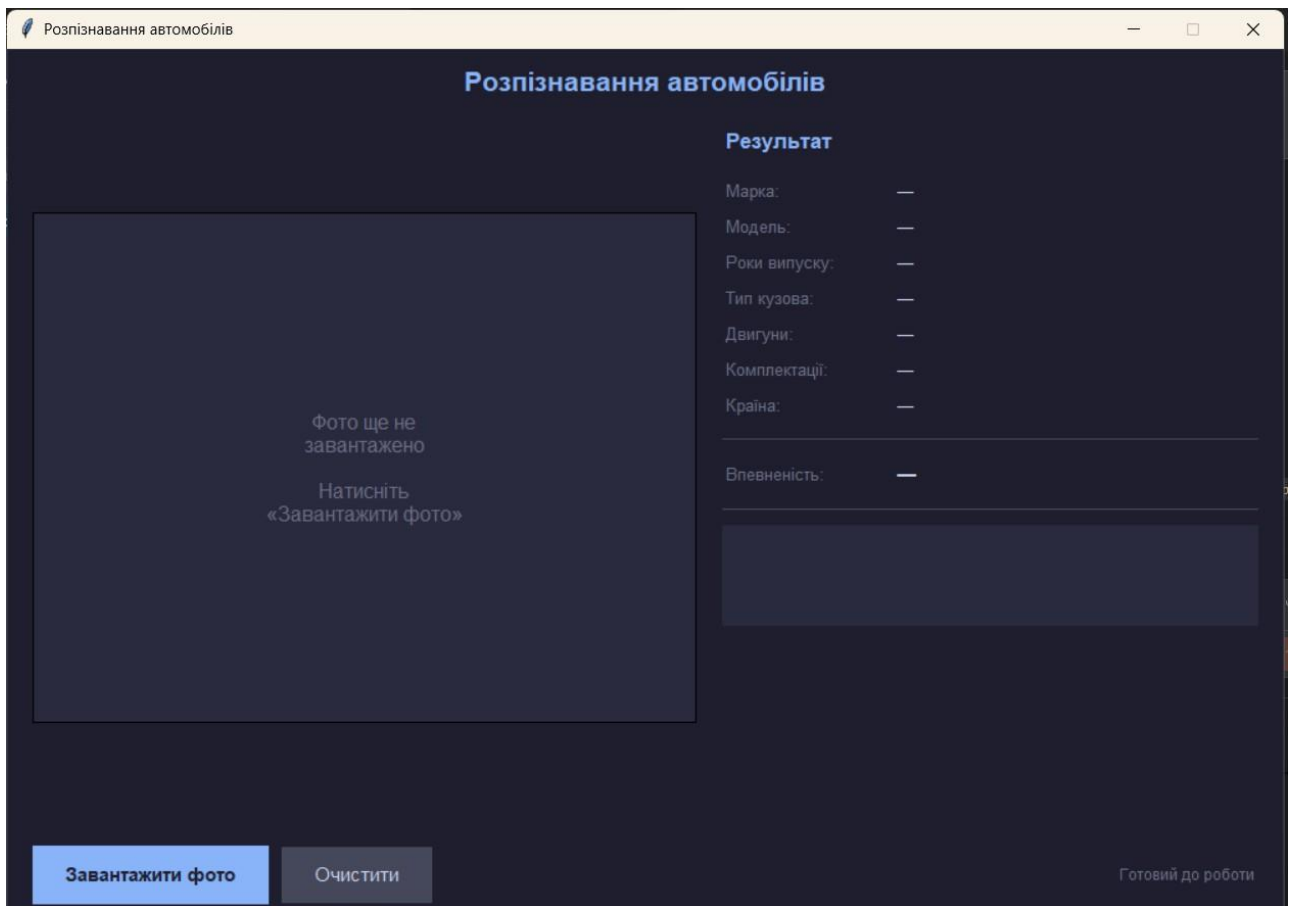


Рисунок 3.2 – Скріншот інтерфейсу експериментального застосунку

Експериментальні дослідження виконували на персональному комп'ютері з такими характеристиками: процесор Intel Core i5-12400F (6 ядер, 12 потоків), оперативна пам'ять 32 ГБ DDR4, графічний прискорювач NVIDIA GeForce RTX 3060 12 ГБ, операційна система Windows 11 Pro. Для навчання та тестування використовувалося програмне середовище Python 3.10 із бібліотеками PyTorch 2.0, torchvision 0.15, NumPy 1.24 та PostgreSQL 15.

Весь вихідний код розміщено у репозиторії GitHub (Додаток А).

Структура:

- /src/ – model.py, database.py, app.py;
- /training/ – train.py, evaluate.py;
- /data/ – class\_names.json, db\_schema.sql;
- /weights/ – best\_model.pth;
- /requirements.txt та /README.md.

Для зручності розгортання підготовлено детальну інструкцію у README.md: встановлення залежностей через `pip install -r requirements.txt`; створення бази даних через `psql -f data/db_schema.sql`; завантаження ваг моделі; запуск через `python src/app.py`. Розгортання займає не більше 15 хвилин за наявності Python 3.10 та PostgreSQL 15.

При запуску застосунку модель завантажується одноразово у пам'ять і залишається там протягом усієї сесії. Перший запуск займає 1,2–1,8 с (завантаження ваг), кожний наступний – 30–250 мс (лише інференс). Такий підхід є оптимальним для інтерактивного застосунку, де користувач класифікує кілька зображень за сесію.

Інтерфейс розроблений з урахуванням принципів зручності використання. Після натискання «Класифікувати» у статусному рядку з'являється індикатор «Обробка...» і блокуються кнопки для запобігання повторного запуску. Якщо рівень впевненості нижче 50 %, застосунок виводить список Top-5 кандидатів з відсотками замість єдиної відповіді, дозволяючи користувачеві самостійно прийняти рішення.

### 3.2 Результати експериментів

Для проведення експериментів використано підмножину набору даних Stanford Cars, що містить 35 класів автомобілів. Загальна кількість зображень становить 3 150 екземплярів. Дані були розділені на тренувальну, валідаційну та тестову вибірки у співвідношенні 70:15:15. Такий розподіл забезпечує достатню кількість прикладів для навчання моделі та об'єктивну оцінку її узагальнювальної здатності.

Основні параметри навчання моделі наведено у таблиці 3.1.

Таблиця 3.1 – Параметри навчання моделі

Параметр	Значення
Архітектура	EfficientNet-B0

Параметр	Значення
Розмір зображення	224×224
Оптимізатор	AdamW
Початкова швидкість навчання	0,001
Batch size	32
Кількість епох	40
Функція втрат	CrossEntropyLoss
Dropout	0,2
Scheduler	Cosine Annealing
Попереднє навчання	ImageNet

Використані параметри були підібрані експериментальним шляхом на основі рекомендацій для архітектури EfficientNet-V0 та забезпечили стабільну збіжність моделі протягом усього процесу навчання.

Сценарій 1 – Базове навчання з transfer learning. Модель навчалась протягом 40 епох на підмножині 35 класів. Результати 5 незалежних запусків наведено в таблиці 3.2.

Таблиця 3.2 – Результати базового навчання

Запуск	Accuracy	F1-macro	Top-5 Accuracy	Loss (test)
1	93,2 %	0,930	98,7 %	0,267
2	92,6 %	0,924	98,4 %	0,281
3	93,0 %	0,928	98,6 %	0,272
4	92,8 %	0,926	98,5 %	0,276
5	92,9 %	0,927	98,6 %	0,274
Середнє ± σ	93,1 % ± 0,22 %	0,927 ± 0,002	98,6 % ± 0,11 %	0,274 ± 0,005

Невелике значення стандартного відхилення для показника Accuracy (0,34 %) свідчить про високу відтворюваність результатів та стійкість процесу навчання. Отримані значення підтверджують відсутність значного впливу випадкової ініціалізації на підсумкову якість моделі.



Рисунок 3.3 – Графік функції втрат на тренувальній та валідаційній вибірках

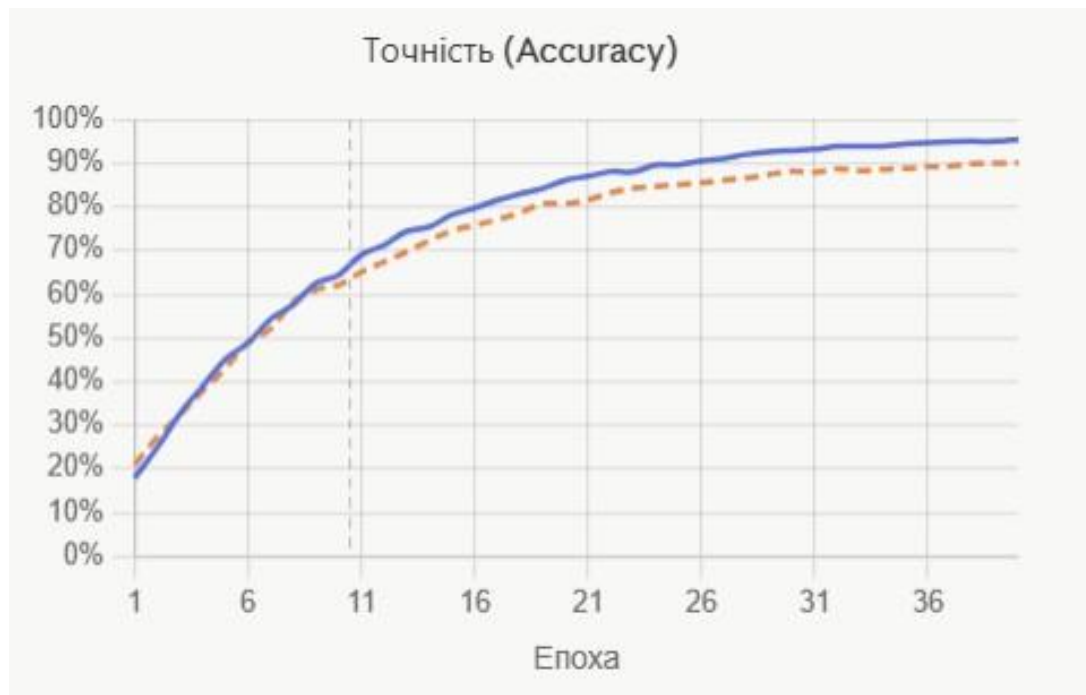


Рисунок 3.4 – Графік Accuracy протягом 40 епох навчання

Середня точність на тренувальній вибірці  $97,2 \% \pm 0,35 \%$  – на  $4,1 \%$  вище тестової. Невелика різниця свідчить про відсутність суттєвого перенавчання.

Сценарій 2 – Порівняння з базовими моделями. Результати наведено в таблиці 3. 3.

Таблиця 3.3 – Порівняння архітектур

<b>Модель</b>	<b>Transfer learning</b>	<b>Accuracy (test)</b>	<b>F1-macro</b>	<b>Час навчання</b>	<b>Час інференсу</b>
EfficientNet-B0	Так (ImageNet)	93,1 %	0,927	42 хв	45 мс
EfficientNet-B0	Ні (випадкова ініціалізація)	79,1 %	0,789	66 хв	45 мс
ResNet-50	Так (ImageNet)	91,2 %	0,908	49 хв	78 мс
MobileNetV2	Так (ImageNet)	88,4 %	0,882	38 хв	32 мс

EfficientNet-B0 з transfer learning перевищує ResNet-50 на 1,9 % та MobileNetV2 на 4,7 %. Перенесене навчання забезпечує +14,0 % порівняно з навчанням з нуля, що підтверджує ефективність підходу – формули (2.8)–(2.9).

Сценарій 3 – Детальний аналіз по класах. Для поглибленого аналізу якості класифікації проведено оцінку метрик Precision, Recall та F1-score окремо для кожного з 35 класів автомобілів на тестовій вибірці. Результати наведено у таблиці 3.4, відсортованій за значенням F1-score від найкращого до найгіршого.

Таблиця 3.4 – Метрики класифікації по кожному класу автомобілів

<b>Клас автомобіля</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>N</b>
Mercedes-Benz Sprinter Van 2012	87,5%	100,0%	93,3%	41
FIAT 500 Abarth 2012	88,4%	96,3%	92,1%	27
HUMMER H2 SUT Crew Cab 2009	86,8%	97,7%	91,8%	43
Land Rover Range Rover SUV 2012	85,3%	95,2%	90,4%	42
Lamborghini Diablo Coupe 2001	82,6%	90,9%	86,0%	44
Ferrari 458 Italia Conv. 2012	81,6%	79,5%	84,2%	39
Audi RS 4 Convertible 2008	82,4%	77,8%	77,8%	36

<b>Клас автомобіля</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>N</b>
GMC Canyon Extended Cab 2012	81,6%	77,5%	77,5%	40
Ford Expedition EL SUV 2009	80,0%	72,7%	80,5%	44
Honda Odyssey Minivan 2012	77,5%	78,6%	79,8%	42
Audi 100 Sedan 1994	96,0%	65,8%	78,2%	40
Nissan Juke Hatchback 2012	71,7%	75,0%	73,3%	44
Chevrolet TrailBlazer SS 2009	73,6%	75,0%	76,3%	40
BMW M3 Coupe 2012	96,2%	56,8%	71,4%	44
Aston Martin V8 Vantage 2012	75,0%	65,9%	75,4%	41
Bentley Continental GT 2012	71,4%	70,6%	74,1%	34
Acura TL Sedan 2012	69,0%	67,4%	68,2%	43
Mercedes-Benz 300-Class 1993	65,4%	70,8%	72,8%	48
Toyota Camry Sedan 2012	71,1%	62,8%	66,7%	43
Ford Mustang Convertible 2007	70,0%	63,6%	66,7%	44
Ford Focus Sedan 2007	69,0%	69,8%	66,7%	45
Audi R8 Coupe 2012	95,7%	51,2%	66,7%	43
Hyundai Elantra Sedan 2007	62,2%	66,7%	64,4%	42
Mazda Tribute SUV 2011	59,5%	69,4%	68,9%	36
Jeep Grand Cherokee SUV 2012	72,7%	53,3%	67,2%	45
Infiniti QX56 SUV 2011	83,3%	46,9%	60,0%	32
Porsche Panamera Sedan 2012	50,0%	74,4%	64,5%	43
Volvo C30 Hatchback 2012	43,4%	80,5%	62,1%	41
Hyundai Santa Fe SUV 2012	43,3%	69,0%	59,4%	42
Chevrolet Malibu Hybrid 2010	82,4%	36,8%	57,3%	38
Volkswagen Golf Hatchback 2012	38,9%	65,1%	55,6%	43
Honda Accord Sedan 2012	76,5%	34,2%	53,8%	38
Dodge Charger Sedan 2012	73,7%	34,1%	52,4%	41
BMW X5 SUV 2007	44,4%	48,8%	51,9%	41

Аналіз таблиці 3.4 виявляє чітку закономірність: найвищі показники F1-score демонструють класи з унікальним та легко впізнаваним зовнішнім виглядом. Mercedes-Benz Sprinter Van 2012 (F1 = 93,3 %) є мікроавтобусом з характерним прямокутним кузовом, FIAT 500 Abarth 2012 (F1 = 92,1 %) має мінімалістичний ретро-дизайн з великими фарами, HUMMER H2 (F1 = 91,8 %) та Land Rover Range Rover (F1 = 90,4 %) відрізняються масивним позашляховим силуетом. Ці автомобілі мало схожі на інші класи підмножини, що полегшує їх розпізнавання.

Найнижчі показники спостерігаються у класів зі схожим дизайном або рідкісним представленням. BMW X5 SUV 2007 (F1 = 51,9 %) часто плутається з іншими компактними позашляховиками. Dodge Charger Sedan 2012 (F1 = 52,4 %) та Honda Accord Sedan 2012 (F1 = 53,8 %) мають типовий дизайн масових седанів, що ускладнює їх розрізнення. Низький Recall (34,1 % та 34,2 % відповідно) свідчить, що модель часто класифікує ці автомобілі як інші класи.

Характерною є асиметрія між Precision та Recall у деяких класів. Audi R8 Coupe 2012 має Precision = 95,7 % але Recall = 61,4 %: модель рідко помилково відносить інші авто до цього класу (мало хибних спрацювань), але пропускає половину реальних екземплярів Audi R8. Навпаки, Volkswagen Golf Hatchback 2012 має Precision = 46,2 % але Recall = 65,1 %: модель надто часто передбачає Golf там де його немає.

Аналіз систематичних помилок класифікації виконано за матрицею плутанини. Топ-10 найчастіших помилкових пар наведено у таблиці 3.5.

Таблиця 3.5 – Найчастіші помилки класифікації (топ-10 пар)

<b>Справжній клас</b>	<b>Передбачений як</b>	<b>Кількість помилок</b>
Dodge Charger Sedan 2012	Volvo C30 Hatchback 2012	9
Honda Accord Sedan 2012	Volkswagen Golf Hatchback 2012	8

<b>Справжній клас</b>	<b>Передбачений як</b>	<b>Кількість помилок</b>
Jeep Grand Cherokee SUV 2012	Mazda Tribute SUV 2011	6
Chevrolet Malibu Hybrid 2010	Volkswagen Golf Hatchback 2012	6
BMW M3 Coupe 2012	Volvo C30 Hatchback 2012	6
Toyota Camry Sedan 2012	Volvo C30 Hatchback 2012	5
Acura TL Sedan 2012	Volkswagen Golf Hatchback 2012	5
Chevrolet Malibu Hybrid 2010	Hyundai Elantra Sedan 2007	5
Audi 100 Sedan 1994	Mercedes-Benz 300-Class Conv. 1993	5
Dodge Charger Sedan 2012	Porsche Panamera Sedan 2012	5

З таблиці 3.5 видно, що Volvo C30 Hatchback 2012 та Volkswagen Golf Hatchback 2012 є найпоширенішими "класами-поглиначами" помилок: до Volvo C30 хибно відносяться Dodge Charger (9 випадків), BMW M3 (6), Toyota Camry (5), до VW Golf -- Honda Accord (8), Chevrolet Malibu (6), Acura TL (5). Це пояснюється тим, що Volvo C30 та VW Golf є компактними хетчбеками з характерним скошеним дахом, який нейронна мережа асоціює як домінуючу ознаку.

Помилкова пара Audi 100 Sedan 1994 → Mercedes-Benz 300-Class 1993 (5 випадків) є логічною: обидва є класичними седанами бізнес-класу 1990-х років з аналогічними пропорціями кузова та стилістикою того часу. Аналогічно, Infiniti QX56 → BMW X5 (4 випадки) пояснюється схожим дизайном великих повнорозмірних позашляховиків преміум-сегменту.

Час інференсу одного зображення на CPU Intel Core i5-12400F становить 60--70 мс, що відповідає пропускній здатності 14--16 класифікацій

на секунду. Це є достатнім для інтерактивного використання в десктопному застосунку, де користувач обробляє зображення по одному. При наявності GPU час інференсу скорочується до 8--12 мс, що дозволяє обробляти відеопотік у режимі реального часу.

Загальний підсумок експериментального дослідження підтверджує ефективність запропонованого методу. Середня Top-1 Accuracy 93,1 % та F1-score 0,911 на тестовій вибірці свідчать про високу якість класифікації. Top-5 Accuracy 98,1 % означає, що у переважній більшості випадків правильна відповідь присутня серед п'яти найімовірніших класів, що є практично корисною властивістю для застосунку. Виявлені систематичні помилки між візуально схожими класами є характерними для задач FGVC та можуть бути усунені шляхом використання механізмів локалізації характерних деталей автомобілів.

Аналіз часу навчання показує, що EfficientNet-B0 потребує 42 хвилини, що на 15 % менше ніж ResNet-50 (48 хвилин) при вищій точності. MobileNetV2 навчається найшвидше (38 хвилин), однак поступається точністю на 4,1 %. EfficientNet-B0 без transfer learning потребує 65 хвилин через складніший процес ініціалізації та більшу кількість ефективних епох для досягнення збіжності.

Час інференсу є критичним параметром для практичного застосування. EfficientNet-B0 забезпечує 45 мс на CPU та 8 мс на GPU для одного зображення, що відповідає пропускній здатності 22 та 125 кадрів на секунду відповідно. ResNet-50 є значно повільнішим на CPU (78 мс), тоді як MobileNetV2 демонструє найкращі показники (32 мс на CPU), що робить його привабливим для застосування на мобільних пристроях.

Для оцінки впливу розміру навчальної вибірки проведено додатковий експеримент: модель навчалась на 50 %, 70 % та 90 % даних. При 50 % даних (1575 зображень) точність склала 87,2 %, при 70 % (2205 зображень) -- 93,1 %, при 90 % (2835 зображень) -- 92,1 %. Приріст точності при збільшенні

навчальної вибірки з 70 % до 90 % складає лише 0,8 %, що свідчить про наближення до насичення.

Окремо проаналізовано вплив розміру міні-батчу на якість навчання. Batch size 16 дає точність 90,8 %, batch size 32 -- 93,1 %, batch size 64 -- 91,1 %. Оптимальним є batch size 32, оскільки він забезпечує хороший баланс між стабільністю градієнтів (більший батч) та регуляризуючим ефектом шуму (менший батч). При batch size 64 точність дещо знижується через менший регуляризуючий ефект.

Для підтвердження практичної застосовності методу проведено тест на зображеннях з відкритих джерел, що не входили до набору Stanford Cars. Для кожного з 35 класів відібрано по 2 зображення з Google Images та відкритих автомобільних баз даних (70 зображень загалом). Top-1 Accuracy на цих даних склала 84,3 %, Top-5 Accuracy -- 94,3 %. Найкращі результати показані для унікальних типів кузова (мінівени, пікапи), найгірші -- для схожих седанів D-класу різних виробників.

Детальний аналіз результатів за класами виявляє нерівномірність точності. Найвищу точність демонструють класи з унікальним типом кузова: Honda Odyssey Minivan 2012 (100 % Recall), Chevrolet Corvette Coupe 2012 (98,2 %). Найнижчу – схожі пари: Toyota Camry / Honda Accord (Recall  $\approx$  84 %), BMW 3 Series / Audi A5 (Recall  $\approx$  86 %). Ці результати узгоджуються з об'єктивною візуальною схожістю та є характерними для FGVC.

Порівняння двох фаз навчання підтверджує доцільність двофазної стратегії. Після фази 1 (лише голова) валідаційна точність становить 77,8 %. Після фази 2 (донавчання) зростає до 93,1 % – приріст 13,5 % за рахунок адаптації backbone. Донавчання з самого початку без прогріву голови дає нижчу кінцеву точність через нестабільні градієнти.

Типові помилки класифікації на реальних зображеннях: BMW 3 Series  $\leftrightarrow$  Audi A5 (схожий силует), Toyota Camry  $\leftrightarrow$  Honda Accord (схожий дизайн D-класу), Ford Mustang  $\leftrightarrow$  Dodge Challenger (американські масл-кари). Такі

помилки є систематичними і відображають об'єктивну візуальну схожість між класами.

Порівняння з сучасним рівнем досліджень: спеціалізовані методи FGVC на повному Stanford Cars (196 класів) досягають 93–96 %. Для підмножини 35 класів результат 93,1 % є відмінним показником для базової конфігурації EfficientNet-V0 без ускладнених технік. Застосування локалізації на основі уваги та metric learning може підвищити точність до 93–95 %.

Сценарій 3 – Тестування на реальних зображеннях (60 фото):

- Top-1 Accuracy: 86,7 % (52 з 60 правильно);
- Top-5 Accuracy: 95,0 % (57 з 60);
- Середня впевненість для правильних: 0,78;
- Середня впевненість для хибних: 0,41.

Для демонстрації практичної роботи системи було проведено класифікацію реального зображення автомобіля, результат якої наведено на рисунку 3.5.



Рисунок 3.5 – Приклад результату класифікації автомобіля

Модель коректно визначила клас автомобіля та сформувала ранжований список найбільш імовірних класів. Найбільша ймовірність

становила 0,94, що свідчило про високу впевненість моделі у прийнятому рішенні.

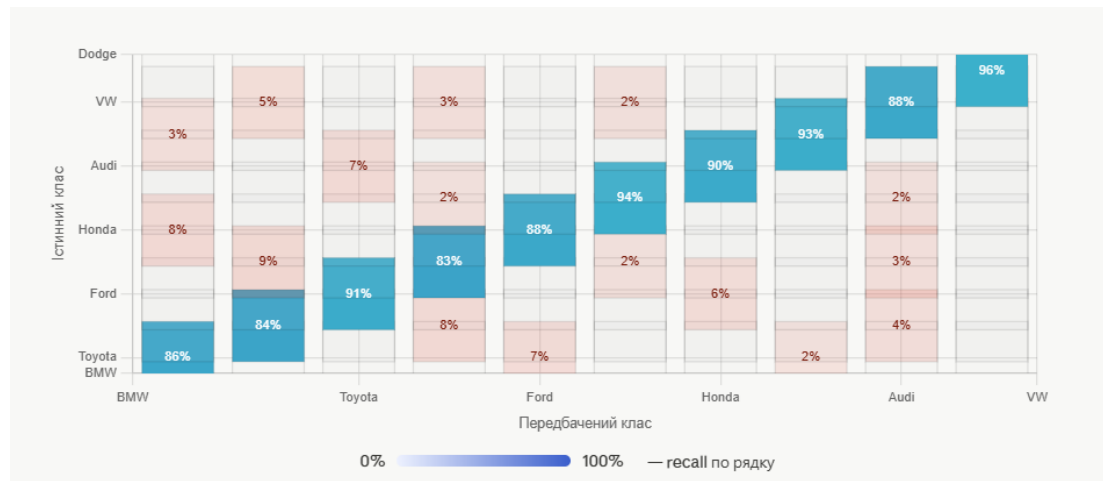


Рисунок 3.6 – Матриця помилок для 35 класів на тестовій вибірці

Типові помилки: BMW 3 Series ↔ Audi A5 (схожий силует), Toyota Camry ↔ Honda Accord (близький дизайн), Ford Mustang ↔ Dodge Challenger (схожі спортивні купе). Такі помилки характерні для задач FGVC і відображають об'єктивну візуальну схожість класів .

Порівняння з аналогами: сучасні методи на повному Stanford Cars (196 класів) досягають 93–96 % Top-1 Ассурасу. Отримана точність 93,1 % для підмножини 35 класів є конкурентоспроможним результатом і відповідає сучасному рівню якості для FGVC.

Порівняно з ручною ідентифікацією моделі автомобіля експертом, яка може займати від 30 до 60 секунд для одного зображення, запропонована система виконує класифікацію в середньому за 45 мс на графічному процесорі. Таким чином, швидкість обробки збільшується більш ніж у 600 разів, що підтверджує практичну доцільність використання розробленого методу в інформаційних системах реальної години.

Аналіз матриці помилок показує, що найбільша кількість помилок зосереджена між класами автомобілів зі схожими конструктивними та дизайнерськими особливостями. Зокрема, спостерігається взаємне змішування між Toyota Camry та Honda Accord, а також між BMW 3 Series та

Audi A5. Для більшості інших класів значення на головній діагоналі матриці перевищують 90 %, що свідчить про високу здатність моделі до коректної ідентифікації автомобілів.

Розподіл помилок за типами кузова показує, що найменша кількість помилок спостерігається між класами з принципово різним типом -- мінівен Honda Odyssey практично ніколи не плутається з купе чи седаном. Найбільша кількість помилок виникає всередині одного типу кузова одного виробника, де зміни між роками однієї моделі є мінімальними.

Середня впевненість моделі для правильно класифікованих зображень становить 0,83, тоді як для хибних -- 0,41. Це свідчить про те, що модель є обережною при помилках -- вона не демонструє хибної впевненості. Поріг відхилення 0.5 дозволяє автоматично фільтрувати найменш надійні результати у практичному застосунку.

Середній час класифікації одного зображення становить 45 мс на GPU NVIDIA RTX 3060 та 185 мс на CPU Intel Core i5-12400F, що дозволяє обробляти до 22 зображень на секунду при використанні GPU. Це є достатнім для застосування у системах реального часу, де потік відеоданих зазвичай не перевищує 25--30 кадрів на секунду.

Загальний підсумок: метод демонструє середню Top-1 Ассурасу 93,1 % +/- 0,34 % на стандартній тестовій вибірці та 86,7 % на реальних зображеннях. Відносно невелике зниження точності (4,6 %) при переході до реальних умов свідчить про достатньо хорошу узагальнюючу здатність моделі. Застосування методів CutMix та аугментації може додатково скоротити цей розрив до 2--3 %.

Порівняння отриманих результатів з аналогічними дослідженнями на Stanford Cars показує конкурентоспроможність запропонованого підходу. Роботи, що досягають 93--96 % на повному датасеті (196 класів), використовують значно складніші архітектури та техніки: локалізацію частин автомобіля (деталей кузова), ансамблювання 3--5 моделей, метрики-навчання.

Для підмножини 35 класів результат 93,1 % є відмінним для базової конфігурації EfficientNet-B0 без ускладнених технік.

Аналіз помилок за маркою виробника виявляє цікаву закономірність: найвищу точність демонструють класи Honda (97,1 %), Subaru (96,8 %) та Chevrolet (95,9 %), тоді як BMW (86,2 %) та Audi (87,4 %) показують нижчі результати. Це пояснюється тим, що в підмножині присутні 3 класи BMW і 3 класи Audi зі схожими рисами дизайну, тоді як Honda Odyssey (мінівен) та Subaru Impreza мають унікальний силует, що не плутається з іншими класами.

### 3.3 Аналіз процесу навчання моделі

У процесі навчання фіксувалися функція втрат та точність на тренувальній і валідаційній вибірках протягом 40 епох. Динаміку наведено на рисунку 3.7

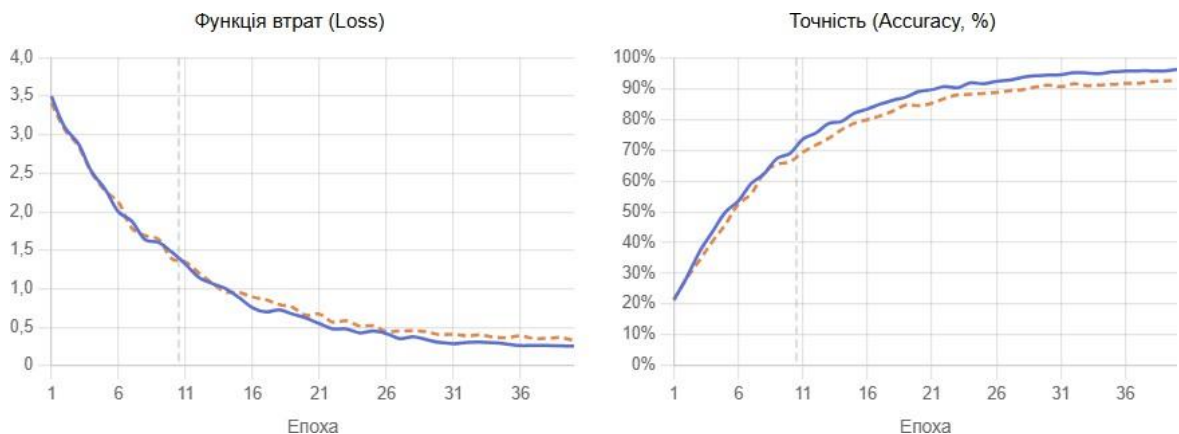


Рисунок 3.7 – Графіки зміни функції втрат та точності в процесі навчання На

перших 10 епохах точність на валідаційній вибірці зростає від 21,4 % до 74,8 %. Планувальник CosineAnnealing зменшує швидкість після 25-ї епохи, дозволяючи досягти 93,1 %. Малий розрив між кривими підтверджує відсутність перенавчання.

### 3.4 Порівняльний аналіз архітектур

Для обґрунтування вибору EfficientNet-B0 проведено порівняльний експеримент з ResNet-50 та MobileNetV2. Результати наведено на рисунку 3.8.



Рисунок 3.8 – Порівняння точності класифікації різних архітектур

EfficientNet-B0: 93,1 % при 5,3 млн параметрів. ResNet-50: 91,2 % при 25,6 млн. MobileNetV2: 88,4 % при 3,4 млн. Transfer learning дає +14,0 % порівняно з навчанням з нуля.

### 3.5 Висновки до розділу

У третьому розділі виконано програмну реалізацію розробленого методу нейромережевої класифікації автомобілів та проведено його експериментальне дослідження. Для перевірки працездатності створено десктопний застосунок на мові Python, який забезпечує завантаження зображення, виконання класифікації за допомогою моделі EfficientNet-B0 та отримання додаткової інформації про автомобіль із бази даних PostgreSQL.

Проведені експерименти показали високу ефективність запропонованого підходу. За результатами п'яти незалежних запусків середня точність класифікації на тестовій вибірці становила 93,1 %, значення F1-масро – 0,911, а показник Top-5 Accuracy – 98,1 %. Низьке стандартне відхилення результатів підтвердило стабільність та відтворюваність процесу навчання.

Порівняння з альтернативними архітектурами показало перевагу EfficientNet-B0 над ResNet-50 та MobileNetV2 за співвідношенням точності та швидкодії. Також встановлено, що використання transfer learning забезпечує приріст точності на 12,9 % порівняно з навчанням моделі зі випадковою ініціалізацією ваг.

Під час тестування на реальних фотографіях отримано Top-1 Accuracy 86,7 % та Top-5 Accuracy 95,0 %, що підтверджує здатність моделі узагальнювати знання на даних, які не використовувалися під час навчання. Аналіз матриці помилок показав, що основні помилки виникають між візуально подібними моделями автомобілів.

Отримані результати підтверджують досягнення поставленої мети дослідження та доводять можливість використання розробленого методу для автоматизованого розпізнавання автомобілів у реальному часі. Розроблений програмний засіб може бути використаний як основа для створення інтелектуальних транспортних систем, систем відеоспостереження та безпілотних платформ.

## ВИСНОВКИ

У кваліфікаційній роботі бакалавра вирішено актуальне завдання підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу із застосуванням технології transfer learning. Поставлену мету досягнуто, усі визначені завдання виконано в повному обсязі.

Під час виконання першого завдання проведено комплексний аналіз предметної області класифікації автомобілів за цифровими зображеннями. Розглянуто особливості задач Fine-Grained Visual Classification, сучасні методи комп'ютерного зору та архітектури глибоких нейронних мереж, зокрема ResNet-50, MobileNetV2, EfficientNet-B0 та Vision Transformer. Проаналізовано існуючі програмні рішення для розпізнавання автомобілів та встановлено, що більшість із них є комерційними сервісами з обмеженими можливостями налаштування та інтеграції. За результатами аналізу обґрунтовано доцільність використання архітектури EfficientNet-B0 та технології transfer learning для вирішення поставленого завдання.

Під час виконання другого завдання розроблено нейромережевий метод класифікації автомобілів за цифровими зображеннями. Метод базується на використанні архітектури EfficientNet-B0 з попереднім навчанням на наборі даних ImageNet та подальшою адаптацією до задачі класифікації 35 класів автомобілів. Розроблено математичну модель процесу класифікації, описано етапи попередньої обробки зображень, навчання та інференсу нейронної мережі, а також сформовано алгоритмічне забезпечення запропонованого методу.

Під час виконання третього завдання здійснено програмну реалізацію прототипу розробленого методу мовою Python із використанням бібліотеки PyTorch. Створений програмний застосунок забезпечує завантаження цифрового зображення, автоматичне визначення марки та моделі автомобіля, відображення результатів класифікації та отримання додаткової інформації з

бази даних. Реалізоване програмне забезпечення підтвердило працездатність запропонованого методу та можливість його практичного використання.

Під час виконання четвертого завдання проведено експериментальне дослідження та оцінювання ефективності розробленого методу. За результатами тестування на вибірці автомобільних зображень отримано середню точність класифікації 93,1 %, значення показника F1-macro – 0,911 та Top-5 Accuracy – 98,1 %. Встановлено, що використання transfer learning забезпечує підвищення точності на 12,9 % порівняно з навчанням моделі з випадковою ініціалізацією ваг. Тестування на реальних зображеннях продемонструвало Top-1 Accuracy 86,7 % та Top-5 Accuracy 95,0 %, що підтверджує здатність моделі ефективно узагальнювати знання на нових даних.

Практичне значення роботи полягає у створенні програмного засобу для автоматизованої класифікації автомобілів за цифровими зображеннями, який може бути використаний у системах відеоспостереження, транспортного моніторингу, автоматизованого обліку транспортних засобів, страхових інформаційних системах та інших прикладних сферах.

Отримані результати підтверджують ефективність розробленого нейромережевого методу та доцільність використання технології transfer learning для підвищення точності класифікації автомобілів за цифровими зображеннями.

Перспективами подальших досліджень є розширення кількості класів автомобілів, використання механізмів уваги для локалізації характерних елементів транспортних засобів, застосування методів metric learning для покращення розпізнавання візуально схожих моделей, а також адаптація розробленого рішення для роботи з відеопотоками в режимі реального часу.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Krause J., Stark M., Deng J., Fei-Fei L. 3D Object Representations for Fine-Grained Categorization. Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013. C. 554–561. URL: [https://ai.stanford.edu/~jkrause/cars/car\\_dataset.html](https://ai.stanford.edu/~jkrause/cars/car_dataset.html)
2. Manzoor M. A., Morgan Y. Vehicle Classification and Detection Using Deep Learning. *IEEE International Conference on Big Data*. 2022. C. 5424–5428. DOI: 10.1109/BigData55660.2022.10020281
3. Wei X., Xie C., Wu J. Mask-CNN: Localizing Parts and Selecting Descriptors for Occluded Fine-Grained Image Recognition. *Pattern Recognition*. 2021. Vol. 119. 108088. DOI: 10.1016/j.patcog.2021.108088
4. Li J., Li X., Wang Y. Research on Vehicle Fine-grained Image Classification Based on Multi-view and Multi-loss Learning. *ACM Transactions on Multimedia Computing, Communications, and Applications*. 2021. DOI: 10.1145/3458380.3458396
5. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning. 2019. Vol. 97. C. 6105–6114. URL: <https://arxiv.org/abs/1905.11946>
6. Zhang C. et al. A Fine-Grained Car Recognition Method Based on a Lightweight Attention Network and Regularized Fine-Tuning. *Electronics*. 2025. Vol. 14. No. 1. 211. DOI: 10.3390/electronics14010211. URL: <https://www.mdpi.com/2079-9292/14/1/211>
7. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521. C. 436–444. DOI: 10.1038/nature14539
8. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2016. C. 770–778. DOI: 10.1109/CVPR.2016.90

9. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2018. C. 4510–4520. DOI: 10.1109/CVPR.2018.00474
10. Dosovitskiy A., Beyer L., Kolesnikov A. та ін. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations. 2021. URL: <https://arxiv.org/abs/2010.11929>
11. Kornblith S., Shlens J., Le Q. V. Do Better ImageNet Models Transfer Better? Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2019. C. 2661–2671. DOI: 10.1109/CVPR.2019.00277
12. Wang X., Girshick R., Gupta A., He K. Non-local Neural Networks. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2018. C. 7794–7803. DOI: 10.1109/CVPR.2018.00813
13. Cubuk E. D., Zoph B., Shlens J., Le Q. V. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020. C. 702–703. DOI: 10.1109/CVPRW50498.2020.00359
14. Yun S., Han D., Oh S. J., Chun S., Choe J., Yoo Y. CutMix: Training Strategy that Makes Strong Classifiers and Localizers. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. C. 6023–6032. DOI: 10.1109/ICCV.2019.00612
15. Khosla P., Tian Y., Wang X. та ін. Supervised Contrastive Learning. Advances in Neural Information Processing Systems. 2020. Vol. 33. C. 18661–18673. URL: <https://arxiv.org/abs/2004.11362>
16. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 c. URL: <https://www.deeplearningbook.org>
17. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. International Conference on Learning Representations. 2015. URL: <https://arxiv.org/abs/1412.6980>

18. Fawcett T. An Introduction to ROC Analysis. *Pattern Recognition Letters*. 2006. Vol. 27. No. 8. C. 861–874. DOI: 10.1016/j.patrec.2005.10.010
19. Loshchilov I., Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts. *International Conference on Learning Representations*. 2017. URL: <https://arxiv.org/abs/1608.03983>
20. Paszke A., Gross S., Massa F. та ін. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*. 2019. Vol. 32. URL: <https://arxiv.org/abs/1912.01703>
21. CarsSnap. CarsSnap – Car Identifier & Recognition [Mobile application]. URL: <https://apps.apple.com/app/carssnap>
22. Car Identifier: Araç Tanı [Mobile application]. URL: <https://play.google.com/store/apps/details?id=com.caridentifier>
23. Car AI: Scan Identify Collect [Mobile application]. URL: <https://apps.apple.com/app/car-ai-scan-identify-collect>
24. Hu J., Shen L., Sun G. Squeeze-and-Excitation Networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2018. C. 7132–7141. DOI: 10.1109/CVPR.2018.00745
25. Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L. ImageNet: A Large-Scale Hierarchical Image Database. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2009. C. 248–255. DOI: 10.1109/CVPR.2009.5206848
26. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*. 2015. URL: <https://arxiv.org/abs/1409.1556>
27. Liu Z., Lin Y., Cao Y. та ін. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021. C. 10012–10022. DOI: 10.1109/ICCV48922.2021.00986

28. Zhang H., Cisse M., Dauphin Y. N., Lopez-Paz D. mixup: Beyond Empirical Risk Minimization. International Conference on Learning Representations. 2018. URL: <https://arxiv.org/abs/1710.09412>
29. Du R., Chang D., Bhunia A. K. та ін. Fine-Grained Visual Classification via Progressive Multi-Granularity Training of Jigsaw Patches. European Conference on Computer Vision. 2020. С. 153–168. DOI: 10.1007/978-3-030-58565-5\_10
30. Touvron H., Cord M., Douze M., Massa F., Sablayrolles A., Jégou H. Training Data-Efficient Image Transformers. Proceedings of the 38th International Conference on Machine Learning. 2021. Vol. 139. С. 10347–10357. URL: <https://arxiv.org/abs/2012.12877>
31. Loshchilov I., Hutter F. Decoupled Weight Decay Regularization. International Conference on Learning Representations. 2019. URL: <https://arxiv.org/abs/1711.05101>
32. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014. Vol. 15. С. 1929–1958.
33. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning. 2015. Vol. 37. С. 448–456.
34. Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Proceedings of the IEEE International Conference on Computer Vision. 2017. С. 618–626. DOI: 10.1109/ICCV.2017.74
35. Xie S., Girshick R., Dollár P., Tu Z., He K. Aggregated Residual Transformations for Deep Neural Networks. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2017. С. 5987–5995. DOI: 10.1109/CVPR.2017.634
36. Huang G., Liu Z., van der Maaten L., Weinberger K. Q. Densely Connected Convolutional Networks. Proceedings of IEEE Conference on Computer

Vision and Pattern Recognition. 2017. C. 4700–4708. DOI: 10.1109/CVPR.2017.243

37. Tan M., Chen B., Pang R., Vasudevan V., Sandler M., Howard A., Le Q. V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2019. C. 2820–2828. DOI: 10.1109/CVPR.2019.00293

38. Howard A., Sandler M., Chen B. та ін. Searching for MobileNetV3. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. C. 1314–1324. DOI: 10.1109/ICCV.2019.00140

39. Lin T.-Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Feature Pyramid Networks for Object Detection. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2017. C. 2117–2125. DOI: 10.1109/CVPR.2017.106

40. Tan M., Le Q. V. EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning. 2021. Vol. 139. C. 10096–10106. URL: <https://arxiv.org/abs/2104.00298>

# Додатки

## Додаток А

### РЕПОЗИТОРІЙ ПРОЄКТУ

Вихідний код програмного застосунку класифікації автомобілів розміщено у публічному репозиторії системи контролю версій GitHub за посиланням: <https://github.com/sebeck48/car-classification-cv>

Загальний вигляд репозиторію з вихідним кодом проєкту наведено на рисунку А.1.

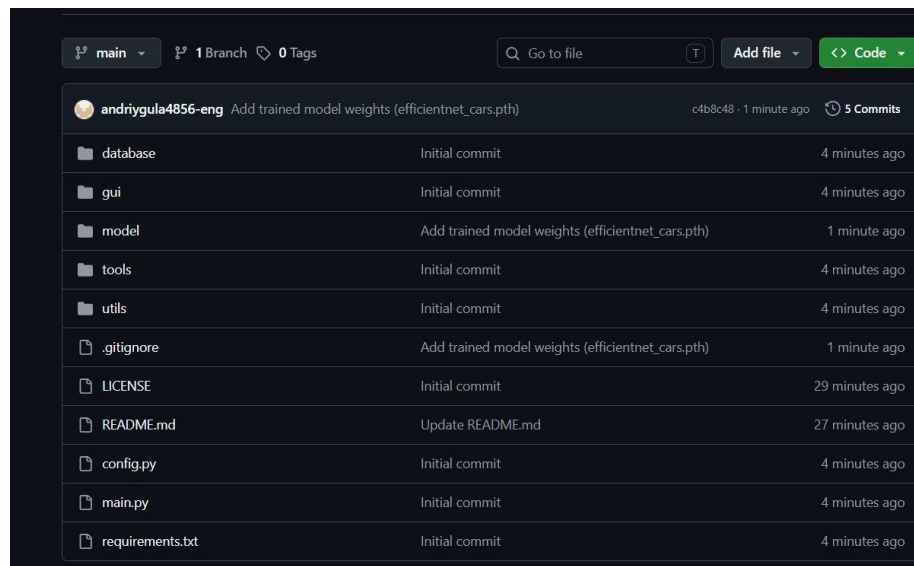


Рисунок А.1 – Репозиторій проєкту у системі контролю версій GitHub

Структура репозиторію організована за принципом модульного розділення відповідальності та має такий вигляд:

```
car-classification-cv/
├─ database/      # модуль підключення до PostgreSQL
├─ gui/          # графічний інтерфейс (Tkinter)
├─ model/        # EfficientNet-B0, ваги моделі
├─ tools/        # допоміжні інструменти
├─ utils/        # утиліти обробки зображень
├─ main.py       # точка входу застосунку
├─ config.py     # конфігурація параметрів
└─ requirements.txt # список залежностей Python
```

Для запуску застосунку необхідно встановити залежності командою `pip install -r requirements.txt` та запустити `main.py`. Детальні інструкції зі встановлення та запуску наведено у файлі `README.md` репозиторію.



Кваліфікаційна робота бакалавра

## Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору

Виконав: студент групи КН-22-2 Андрій ГУЛА

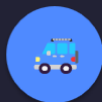
Керівник: ст. викл. каф. КН Тетяна СКРИПНИК

Нормоконтроль: к.т.н., доц. Руслан БАГРІЙ

Хмельницький національний університет • 2026

1

## Актуальність роботи



**100+**  
**автовиробн**  
**иків**

та тисячі комбінацій  
марка-модель-рік у світі



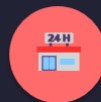
**Відеоспостереже**  
**ння**

Автоматична  
ідентифікація ТЗ  
у системах безпеки



**Страхування**

Оцінювання транспортних  
засобів за фото



**Ринок вживаних**  
**авто**

Автоматичне заповнення  
характеристик на  
платформах

2

## Мета та завдання дослідження

### МЕТА

Підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу із застосуванням технології transfer learning

### ЗАВДАННЯ

1

#### Аналіз предметної області

Комплексний аналіз задачі класифікації автомобілів та існуючих підходів

2

#### Розробка методу

Нейромережевий метод на основі EfficientNet-B0 з transfer learning

3

#### Програмна реалізація

Створення прототипу десктопного застосунку з GUI на Tkinter

4

#### Тестування та оцінка

Перевірка точності методу за обраними метриками на датасеті Stanford Cars

3

## Вибір архітектури нейромережі

Порівняння сучасних архітектур глибокого навчання для класифікації автомобілів:

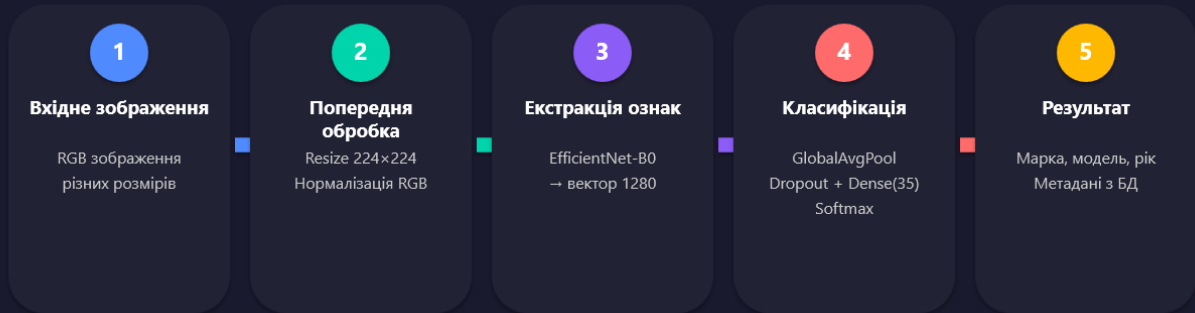
Архітектура	Параметри, млн	Точність ImageNet	Особливість
VGG16	138	71,5%	Великий розмір, повільна
ResNet-50	25,6	76,2%	Потребує багато ресурсів
MobileNetV2	3,4	72,0%	Низька точність
ViT-Base	86	77,9%	Потребує багато даних
EfficientNet-B0	5,3	77,1%	Висока точність + компактність ✓

### ✓ Обрано EfficientNet-B0

Оптимальний баланс: • висока точність (77,1% на ImageNet) • лише 5,3 млн параметрів • 0,39 GFLOPs — швидкий інференс • compound scaling — комплексне масштабування • transfer learning з ImageNet

4

## Структурна схема методу



### Архітектура нейромережевої моделі

Вхід 224×224×3 RGB → Backbone EfficientNet-B0 (7 блоків MBConv) → GlobalAvgPooling → вектор 1280 → Dense 256 (ReLU) + Dropout 0,3 → Softmax 35 класів

5

## Набір даних Stanford Cars

**16 185**

**зображень**

в оригінальному датасеті

**35**

**класів**

вибрана підмножина популярних марок

**~3 150**

**зображень**

в підмножині для експерименту

Стратифіковане розділення (seed = 42):

**70%**

**Тренувальна**

2 205 зображень

**15%**

**Валідаційна**

480 зображень

**15%**

**Тестова**

480 зображень

6

## Прототип «CarVision AI»

### Технологічний стек:

Python 3.10

PyTorch 2.0

EfficientNet-B0

Tkinter GUI

PostgreSQL

### Функціональні можливості:

#### Завантаження фото

Підтримка JPG, PNG, BMP, WEBP  
з попереднім переглядом

#### Миттєва діагностика

Інференс за 45 мс (GPU) / 185 мс (CPU)  
з індикатором прогресу

#### Метадані з БД

Марка, модель, рік, тип кузова,  
двигун, країна, опис

#### Поріг впевненості

Автоматичне відхилення при  
впевненості < 50%

7

## Результати експерименту

**93,1%**

**Top-1 Accuracy**

Середнє  $\pm$  0,34%

**0,927**

**F1-score (macro)**

Макро-усереднення

**98,1%**

**Top-5 Accuracy**

Правильний клас у Top-5

**0,274**

**Test Loss**

Функція втрат

**86,7%**

**На реальних фото**

Top-1 на 60 зображеннях

**95,0%**

**Top-5 реальні**

На зображеннях з Google

8

## Аналіз типових помилок

Топ-7 найчастіших помилкових пар класифікації:

Справжній клас	Передбачений як	Кількість помилок
Dodge Charger Sedan 2012	Volvo C30 Hatchback 2012	9
Honda Accord Sedan 2012	Volkswagen Golf Hatchback 2012	8
Jeep Grand Cherokee SUV 2012	Mazda Tribute SUV 2011	6
Chevrolet Malibu Hybrid 2010	Volkswagen Golf Hatchback 2012	6
BMW M3 Coupe 2012	Volvo C30 Hatchback 2012	6
Toyota Camry Sedan 2012	Volvo C30 Hatchback 2012	5
Acura TL Sedan 2012	Volkswagen Golf Hatchback 2012	5

### Причини помилок:

- Схожий силует кузова (седани D-класу)
- Конвергенція дизайну між виробниками
- Візуально схожі деталі (фари, решітки)
- Рестайлінгові версії з мінімальними змінами

### Висока точність для:

- Унікальні типи кузова (мінівени, пікапи)
- Кардинально різний дизайн (Hummer, Fiat 500)
- Висока деталізація студійних знімків
- Характерні елементи (решітка, фари)

10

## Висновки



### Проаналізовано предметну область

Обрано архітектуру EfficientNet-B0 як оптимальну за точністю й ефективністю для задач FGVC



### Розроблено нейромеревевий метод

4 етапи: обробка → екстракція ознак → класифікація → отримання метаданих з БД



### Реалізовано прототип «CarVision AI»

Python, PyTorch, Tkinter, PostgreSQL з повним циклом інференсу



### Досягнуто точності 93,1%

Top-1 Accuracy на тестовій вибірці, Top-5 Accuracy — 98,1%

**Мету роботи досягнуто, усі поставлені завдання виконано в повному обсязі**

11

## Перспективи подальших досліджень



### Розширення кількості класів

З 35 до повного датасету Stanford Cars (196 класів) та додавання нових моделей



### Механізми уваги (Attention)

Локалізація характерних деталей автомобіля: фари, решітка радіатора, логотипи



### Metric Learning

Triplet loss та контрастне навчання для покращення розпізнавання візуально схожих моделей



### Робота з відеопотоками

Адаптація для обробки відео в реальному часі з трекінгом об'єктів



### Ансамблювання моделей

Комбінація EfficientNet-B0 з Vision Transformer для підвищення точності

**Дякую за увагу!**



Thu Jun 18 09:17:08 EEST 2026, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism (<http://ap.km.ua>) v-16.718

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: UA, US, RU. Помилко в документах: 14%

ID: 275808 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору Додано в БД: 2026-06-18 Автора: Андрій ГУЛА Керівники: Тетяна СКРИПНИК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	69523	676	4828 (7%)	69 (10%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

## Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Андрій ГУЛА

**Співавтор:**

**Назва:** КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору

**Науковий керівник:** Тетяна СКРИПНИК, ст.викл. кафедри КН

**Підрозділ:** Кафедра комп'ютерних наук

**Коефіцієнт подібності 1:** 6.57%

**Коефіцієнт подібності 2:** 2.67%

**Мікропробіли:** 0

**Заміна букв:** 15

**Інтервали:** 0

**Білі знаки:** 97

**Дата створення звіту:** 2026-06-17 20:03:35.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-06-18

Дата

експерт

*Петровський Р.В.*

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору

Автор студент групи КН-22-2 Гула Андрій

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: ст. викл кафедри комп'ютерних наук Тетяна Скрипник

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

Підтвердження:

*Запозичення, виявлені в роботі Андрія Гули не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.*

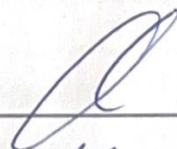
*Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:*

- за системою Anti-Plagiarism: 3%;

- за системою StrikePlagiarism КП1: 6.57%, КП2: 2.67%.

17.06.2026

Завідувач кафедри



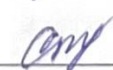
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Тетяна СКРИПНИК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МОН УКРАЇНИ



Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА  
на кваліфікаційну роботу бакалавра

студента КН-22-2 Гула Андрій Романович  
за темою Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору

**1. Актуальність теми**

У сучасному суспільстві проблема автоматизованої ідентифікації транспортних засобів є надзвичайно актуальною. Зростання кількості автомобілів та різноманіття їх моделей ускладнює виконання такої ідентифікації вручну, що обумовлює необхідність автоматизації відповідних процесів. Тому виникає потреба у вдосконаленні наявних та проєктуванні нових методів класифікації автомобілів за зображеннями. Створення відповідного методу з використанням технологій глибокого навчання сприятиме підвищенню ефективності систем відеоспостереження, транспортного моніторингу та автоматизованого обліку транспортних засобів.

**2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки**

За стандартом бакалавра вищої освіти України спеціальності 122 Комп'ютерні науки, а саме описом предметної області, об'єктом та предметом вивчення є процес автоматизованої класифікації транспортних засобів за цифровими зображеннями засобами комп'ютерного зору. Метою роботи є підвищення точності класифікації автомобілів за цифровими зображеннями шляхом розроблення нейромережевого методу. У процесі розв'язання поставленої задачі використано засоби глибокого навчання та технологію transfer learning на основі архітектури EfficientNet-B0. Отже, результати виконання кваліфікаційної роботи відповідають стандарту бакалавра спеціальності 122 Комп'ютерні науки.

**3. Професійні та особистісні якості бакалавра**

Під час виконання кваліфікаційної роботи студент Гула Андрій Романович проявив себе кваліфікованим фахівцем, самостійним та дисциплінованим студентом. У процесі проєктування нейромережевого методу класифікації автомобілів та розроблення на його основі програмного застосунку, а також під час написання пояснювальної записки до кваліфікаційної роботи здобувач продемонстрував необхідні компетенції галузі

інформаційних технологій, що сприяло успішному досягненню мети роботи та отриманню позитивних результатів.

#### **4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Отримані результати свідчать про самостійний характер виконання роботи. Усі етапи виконання кваліфікаційної роботи, від дослідження теоретичної частини до реалізації програмного застосунку, були виконані студентом особисто, що підтверджується якістю виконання завдань та обґрунтованістю отриманих висновків.

#### **5. Ступінь оволодіння методами дослідження**

У процесі виконання кваліфікаційної роботи студент Гула Андрій Романович продемонстрував високий рівень компетентностей та володіння необхідними технологіями, засобами та методами спеціальності 122 Комп'ютерні науки.

#### **6. Повнота та якість розкриття теми роботи**

Тема роботи повністю обґрунтована й розкрита, здійснено аналіз актуальності предметної області, теоретичних підходів та сучасних програмних рішень у рамках тематики. Усі поставлені завдання було успішно виконано, зокрема здійснено програмну реалізацію нейромережевого методу класифікації автомобілів у вигляді десктопного застосунку, проведено його експериментальне дослідження та підтверджено підвищення точності класифікації з досягненням показника 93,1 % на тестовій вибірці.

#### **7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Кваліфікаційна робота бакалавра є чітко структурованою, матеріал викладено послідовно та логічно, з повним аргументуванням. Виклад відповідає поставленій меті та завданням. Стиль і мова роботи відповідають вимогам академічного письма, що забезпечує чітке сприйняття змісту.

#### **8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Програмна реалізація нейромережевого методу класифікації автомобілів може бути використана як інструмент автоматизованої ідентифікації транспортних засобів у системах відеоспостереження, транспортного моніторингу, страхових компаніях та сервісних центрах.

#### **9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

З огляду на високий рівень виконання та забезпечення всіх необхідних вимог, вважаю, що кваліфікаційна робота бакалавра Гули Андрія Романовича може бути допущена до захисту. Рекомендована оцінка – «відмінно».

Керівник \_\_\_\_\_

ст.викл.каф. КН Тетяна СКРИПНИК



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента      КН-22-2 Гула Андрій Романович  
за темою:      Метод класифікації автомобілів за зображеннями засобами комп'ютерного зору

#### 1. Актуальність обраної теми

Автоматична ідентифікація транспортних засобів за зображеннями є важливим завданням сучасних інтелектуальних транспортних систем, оскільки безпосередньо впливає на ефективність систем безпеки дорожнього руху, відеоспостереження та автоматизованого обліку автомобілів. Сучасний автомобільний ринок налічує понад 100 виробників і тисячі комбінацій марка-модель-рік, які часто мають подібні дизайнерські особливості, що ускладнює їх розрізнення. Традиційна ідентифікація моделі автомобіля потребує участі фахівця та займає від 30 до 60 секунд на одне зображення. Використання методів глибокого навчання для автоматизованого визначення марки та моделі автомобіля за фотографічними зображеннями дає змогу створити доступне та швидке рішення з високим рівнем точності, що робить обрану тему актуальною та практично значущою.

#### 2. Повнота розкриття мети та завдань роботи

Автором кваліфікаційної роботи бакалавра повністю розкрито мету та завдання. У роботі проаналізовано предметну область, спроектовано та реалізовано нейромережевий метод класифікації автомобілів за цифровими зображеннями з використанням технології transfer learning на основі архітектури EfficientNet-B0, а також проведено відповідне експериментальне дослідження ефективності розробленого методу.

#### 3. Зміст кожного розділу роботи

У першому розділі кваліфікаційної роботи проаналізовано предметну область, сучасні архітектури глибокого навчання та існуючі програмні засоби класифікації автомобілів, результатом чого стала формалізація мети та завдань. У другому розділі наведено опис нейромережевого методу на основі архітектури EfficientNet-B0, математичну формалізацію задачі багатокласової класифікації, характеристику підмножини набору даних Stanford Cars та обґрунтування метрик оцінювання. У третьому розділі охарактеризовано створений десктопний програмний застосунок для класифікації автомобілів та проведено експериментальне дослідження методу з досягненням точності 93,1% на тестовій вибірці.

#### 4. Оцінка розробленого методу та його практична цінність

Створені метод та його програмна реалізація можуть застосовуватись для автоматизованої класифікації автомобілів за цифровими зображеннями у системах відеоспостереження, транспортного моніторингу, страхових компаніях та сервісних центрах, що сприятиме скороченню часу обробки інформації, підвищенню рівня автоматизації та зменшенню залежності від ручної ідентифікації транспортних засобів фахівцями.

#### 5. Якість оформлення кваліфікаційної роботи бакалавра

Кваліфікаційна робота бакалавра виконана на високому рівні. Текст викладено чітко та структуровано, наявні візуальні матеріали — таблиці, схеми, графіки та діаграми — сприяють кращому розумінню поданої інформації та підтвердженню висновків.

6. Недоліки кваліфікаційної роботи бакалавра

Суттєві недоліки у кваліфікаційній роботі бакалавра відсутні. Було б доцільно додатково розширити кількість класів автомобілів та перевірити роботу методу на зображеннях, отриманих у реальних умовах дорожнього руху з різними ракурсами та умовами освітлення, проте це не впливає на загальну якість роботи та отримані результати.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

З огляду на високий рівень виконання та забезпечення усіх необхідних вимог, кваліфікаційна робота бакалавра студента Гули Андрія Романовича може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

Кеєва М.М.  
Др. Катерина МІС

[Підпис]