

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна математика

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____

(підпис)

“ ____ ” _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Пригорко Максиму Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Математичні моделі і методи розпізнавання
повітряних об'єктів в відеопотоці

затверджена наказом по університету від 05 листопада 2021 р. № 1641 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2021 р.

3. Вихідні дані до роботи відеопотоки в оптичному та інфрачервоному діапазоні
частот, які містять літальні об'єкти

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

1. Актуальність теми роботи _____

2. Постановка задачі _____

3. Аналіз предметної області _____

4. Метод чисельного аналізу _____

5. Результати обчислювального експерименту _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	8 – 14 листопада 2021 р.	виконано
2	Вибір та обґрунтування методу	15 – 21 листопада 2021 р.	виконано
3	Розробка алгоритму і програми	22 – 28 листопада 2021 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	29 листопада – 5 грудня 2021 р.	виконано
5	Робота над текстом пояснювальної записки	6 – 9 грудня 2021 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 грудня 2021 р.	виконано

Дата видачі завдання 8 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Тевяшев А.Д.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 44 с., 1 табл., 29 рис., 2 дод., 7 джерел.

НЕЙРОННА МЕРЕЖА, ПОВІТРЯНИЙ ОБ'ЄКТ, МЕТОД ТРАНСФЕРНОГО НАВЧАННЯ, INCEPTION V3, DENSENET, RESNEXT, MOBILENET.

Об'єкт дослідження – процес відеоспостереження за ЛО в оптичному та інфрачервоному діапазоні частот з використанням технологій трансферного навчання.

Мета роботи – підвищення ефективності розпізнавання, порівняння та оцінка ефективності початкових нейронних мереж для заміни та навчання останнього шару для вирішення задач розпізнавання типа ЛО у відеопотоці.

Методи дослідження – технології трансферного навчання, нейронні мережі.

У роботі був проведений аналіз ефективності нейронних мереж для розпізнавання літальних об'єктів в відеопотоці. Були побудовані та проаналізовані навчені моделі.

Було досліджено та проаналізовано декілька навчених моделей нейронних мереж розпізнавання та детектування літальних об'єктів в відеопотоці, порівняно їх ефективність та їх доцільність використання у даній роботі. Розроблено програму, яка у реальному часі показує ймовірність знаходження в конкретному кадрі відеопотока літального об'єкту. Результати навчання моделі подані у вигляді графіків загальної точності та помилок. Результати роботи програми приведені у вигляді скріншотів з показаною ймовірністю віднесення літального об'єкту у кадрі до конкретного класу.

ABSTRACT

Introductory note: 44 pages, 1 tables, 29 figures, 2 appendixes, 7 sources.

NEURAL NETWORK, FLYING OBJECT, TRANSFER TRAINING METHOD, INCEPTION V3, DENSENET, RESNEXT, MOBILENET.

Object of research – the process of video surveillance of FO in the optical and infrared frequency bands using transfer learning technologies.

Purpose of work – improving the efficiency of recognition, comparing and evaluating the effectiveness of initial neural networks to replace and train the last layer to solve problems of recognition of the type of FO in the video stream.

Methods of research – transfer learning technologies, neural networks.

The paper analyzes the effectiveness of neural networks for recognizing flying objects in the video stream. Trained models were built and analyzed.

Several studied models of neural networks for the recognition and detection of aircraft objects in the video stream were investigated and analyzed, compared their efficiency and their feasibility in this work. A program has been developed that shows in real time the probability of finding an aerial video stream in a particular frame. The learning results of the model are presented in the form of graphs of general accuracy and errors. The results of the program are presented in the form of screenshots showing the probability of assigning an aircraft object in the frame to a particular class.

ЗМІСТ

	С.
Вступ	7
1 Аналіз предметної області та постановка задач дослідження	8
1.1 Методи розпізнавання ПО з використанням глибоких нейронних мереж..	8
1.2 Попередня обробка зображення	13
1.3 Змістовна та формальна постановка задачі	19
1.4 Постановка задач дослідження	20
2 Вибір та обґрунтування методу розв’язання	22
2.1 Стадія попередньої обробки	22
2.2 Inception v3	23
2.3 ResNet	26
2.4 DenseNet	28
2.5 MobileNet	30
3 Програмна реалізація	32
3.1 Формування навчальної вибірки	32
3.2 Алгоритм розв’язання задачі класифікації літальних об’єктів	34
3.3 Опис програми	35
4 Результати обчислювального експерименту та їх аналіз	37
Висновки	40
Перелік джерел посилання	41
Додаток А Код навчання моделі	42
Додаток Б Код програми розпізнавання літальних об’єктів в відеопотоці	44

ВСТУП

Актуальність теми. Прогрес у розвитку повітряних об'єктів збільшує складність завдань їх виявлення та розпізнавання. Швидке та якісне вирішення цих завдань має велике значення для цивільного застосування в галузі управління повітряним рухом, моніторингу повітряної обстановки в аеропортах, а також для завдань військового призначення. Тема кваліфікаційної роботи є актуальною, тому що із розвитком у сфері стелс-технологій, з'явилася потреба у точному та швидкому візуальному розпізнаванні літальних об'єктів.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є розробка програмного забезпечення для детектування та класифікації літальних об'єктів у відеопотоці в інфрачервоному та оптичному діапазоні частот. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі «розпізнавання літальних об'єктів в відеопотоці»;
- провести навчання нейронної мережі;
- розробити програмне забезпечення для вирішення задачі класифікації літальних об'єктів в відеопотоці.

Об'єктом дослідження є процес відеоспостереження за ЛО в оптичному та інфрачервоному діапазоні частот з використанням технологій трансферного навчання.

Предметом дослідження є методи навчання нейронних мереж, методи попередньої обробки зображень

Методи дослідження. У кваліфікаційній роботі використовуються технології трансферного навчання, нейронні мережі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Методи розпізнавання ПО з використанням глибоких нейронних мереж

Недавня поява високоефективного глибокого навчання у вигляді інструментів, таких як Caffe, TensorFlow і ряду інших спеціалізованих бібліотек штучного інтелекту, стимулювали прогрес в дослідженнях глибокого навчання [8]. Так конкурс ImageNet дозволив отримати значний прогрес в згортальних архітектурах нейронних мереж, який робить цю технологію прийнятною для розпізнавання об'єктів ПО.

Поява ранніх згортальних нейронних мереж (Convolutional Neural Networks – CNN), заснованих на ідеях, що лежать в основі зорової системи ссавців, вивела рішення задач розпізнавання образів на новий рівень. Їх поява призвела до розвитку нейромережевих методів навчання, появи інфраструктури для їх реалізації у вигляді відносно недорогого доступу до високопродуктивних хмарних обчислень, великим набором навчальних даних.

Історія розробки нейромережевих методів сходить до середини минулого століття, коли їх архітектуру намагалися реалізувати на аналогових пристроях. У 1970-х роках роботи з штучними нейронними мережами проводилися на цифрових комп'ютерах, але в силу слабкості їх обчислювальних ресурсів, експерименти проводилися з архітектурою, яка містила один або два шари з невеликою кількістю нейронів. Ще однією перешкодою в їх використанні для задач розпізнавання цифрових зображень, як зрозуміло з висоти сьогодення розуміння, була нестача помічених даних до появи Інтернету.

Технологія навчання багат шарових мереж у вигляді зворотного поширення помилки з використанням градієнтного спуску з'явилася в середині 80-х років.

У 1998 році ЛеКун (LeCun) розробив одну з перших архітектур CNN з 5 шарами, як показано на рисунку 1.1.

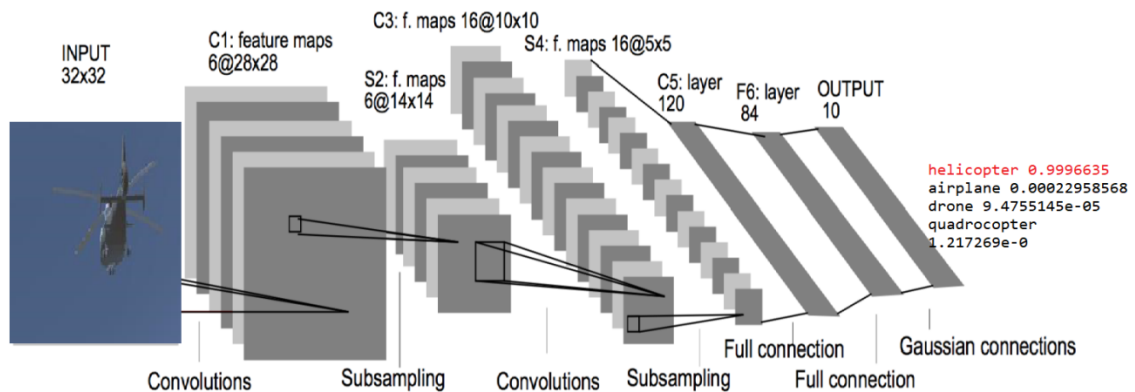


Рисунок 1.1 – Типова архітектура згортальної нейронної мережі типу LeNet-5

Це традиційні методи обробки зображень, такі як масштабно-інваріантна трансформація ознак (SIFT) і гістограмо-орієнтовані градієнти (HOG) були реалізовані на CNN для розпізнавання десятків або сотень класів зображень (Caltech 5, Caltech 101, MIT-CBCL) після тренування на тисячах прикладів зображень.

Сучасний етап розвитку класифікація зображень пов'язаний з проектом The PASCAL Visual Object Classes (VOC), був запущений в 2005 році і замінений проектом ImageNet Large Scale Visual Recognition Challenge (ILSVRC) з 2010 року. Станом на сьогоднішній день Imagenet містить 15 мільйонів помічених зображень приблизно в 22000 категорії.

Використовуючи графічні процесори (GP-GPU) і набір даних ILSVRC 2010 року, Хінтон і Крижевський побудував і випробував мережу AlexNet, яка вважається першою сучасною великомасштабною CNN. Мережа з великим відривом виграла конкурс з розпізнавання зображень ImageNet LSVRC-2012 у 2012 році (з кількістю помилок 15,3% проти 26,2% у другого місця).

AlexNet містить 5 згортальних шарів і 3 повнозв'язних шари. Згодом, 22-шаровий класифікатор GoogLeNet досяг 6,67% помилок класифікації.

На сьогоднішній день тенденція у розвитку CNN полягає в появі дедалі глибших мереж, які перевершують рівень розпізнавання образів середньої людини.

Застосування багат шарового перцептрона з традиційною структурою при вирішенні реальних задач розпізнавання і класифікації зображень викликає певні труднощі. По-перше, зображення, як правило, мають велику розмірність, внаслідок чого зростає число нейронів і синаптичних зв'язків у мережі. У свою чергу, це вимагає збільшення навчальної вибірки, внаслідок чого збільшується час і обчислювальна складність процесу навчання. По-друге, ігнорується топологія вхідних даних. Компоненти вхідного шару можуть бути представлені в будь-якому порядку, без урахування мети навчання. Однак зображення мають строгу двовимірну структуру, в якій існує залежність між просторово-сусідніми пікселями. Від цих недоліків вільні так звані згорткові нейронні мережі, які являють собою особливий клас багат шарових перцептронів, спеціально створених для розпізнавання двовимірних поверхонь з високим ступенем інваріантності до масштабування, зсуву, повороту, зміни ракурсу та інших просторових спотворень, і глибокі НМ побудовані на базі автоенкодерів, попереднє навчання яких здійснюється на базі обмежених машин Больцмана.

Згорткова нейронна мережа (ЗНМ) – тип багат шарової нейронної мережі, яка свою назву «згорткова мережа» отримала за назвою операції – згортка, вона часто використовується для обробки зображень і може бути описана наступною формулою:

$$(f \times g)[m, n] = \sum_{k, l} f[m - k, n - l] \cdot g[k, l], \quad (1.1)$$

де f – вихідна матриця зображення;

g – ядро (матриця) згортки.

Неформально цю операцію можна описати наступним чином – вікном розміру ядра g проходимо з заданим кроком (зазвичай 1) все зображення f на кожному кроці поелементно множимо вміст вікна на ядро g , результат сумується і записується в таблицю результату. Ідея згорткових нейронних мереж полягає в чергуванні згорткових шарів і субдискретизуючих шарів. Структура мережі – односпрямована (без зворотних зв'язків), багатошарова (рис. 1.2). Модель згорткової мережі складається з трьох типів шарів: згорткові шари, субдискретизуючі (підвибірка) верстви і прошарки «звичайної» нейронної мережі – персептрона. Архітектура згорткових нейронних мереж реалізує три ідеї, які забезпечують інваріантність мережі до невеликих зрушень, змін масштабу і спотворень:

- кожен нейрон отримує вхідний сигнал від локального рецептивного поля у попередньому шарі, що забезпечує локальну двовимірну зв'язність нейронів;

- кожен прихований шар мережі складається з безлічі карт ознак, на яких всі нейрони мають загальні ваги, що забезпечує інваріантність до зміщення і скорочення загальної кількості вагових коефіцієнтів мережі;

- за кожним шаром згортки слідує обчислювальний шар, який здійснює локальне усереднення та підвибірку, що забезпечує зменшення розширення для карт ознак.

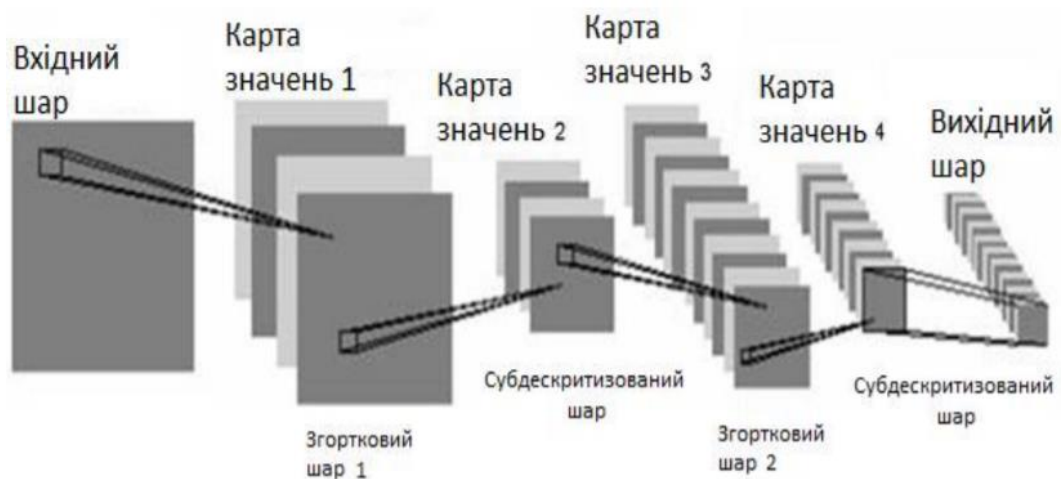


Рисунок 1.2 – Структура згорткової мережі

Робота згорткової нейронної мережі забезпечується двома основними елементами:

- фільтри (визначники ознак);
- карти ознак.

Фільтр – це невелика матриця, що представляє ознаку, яку необхідно знайти на вихідному зображенні. За допомогою верхнього фільтра визначаються частини вихідного зображення з вертикальними лініями, нижній фільтр служить для визначення частин зображення з горизонтальними лініями. Безпосередньо процес визначення заснований на операції згортки фільтром оригінального зображення. Результати згортки, які визначають місце розташування ознак вихідного зображення, називаються картами ознак. Мета процесу згортки – зменшити розмірність карти ознак до такої міри, щоб з повним набором ознак могла працювати мережа прямого поширення (в більшості випадків багатошаровий перцептрон). Згортковий шар реалізує ідею локальних рецептивних полів, тобто кожен вихідний нейрон з'єднаний тільки з певною (невеликою) областю вхідної матриці і таким чином моделює деякі особливості людського зору. Недоліками згорткових нейронних мереж (ЗНМ) є:

- висока складність архітектури;
- повнозв'язаність;
- фіксована площа вікна шару згортки. З метою підвищення ефективності роботи ЗНМ необхідно знайти оптимальні значення наступних параметрів: • кількість карт ознак;
- щільність зв'язків між картами ознак;
- розмір вікна;
- площа перекриття;
- початкова ініціалізація ваг.

1.2 Попередня обробка зображення

Зображення ПО, що рухається, можуть бути отримані з різним ступенем спотворення, пов'язаного з дальністю, і як наслідок –розмір зображення, його розмитістю в наслідок аберації світлових променів, що проходять через шари повітря з різною щільністю (рисунок 1.3).

Крім того, навіть при великомасштабному зображенні виникають візуальні перешкоди, пов'язані з освітленістю, туманом, хмарністю(рисунок 1.4).



Рисунок 1.3 – Зображення БПЛА MQ-9 Predator з хорошою роздільною здатністю, реально отримане дрібне зображення і його збільшений вигляд в піксельної сітці



Рисунок 1.4 – Зображення БПЛА MQ-9 Predator в ідеальному для розпізнавання вигляді і в умовах візуальних перешкод

Перетворення кольорового зображення до виду в відтінках сірого. Оскільки отримується для розпізнавання зображення, яке представлено у вигляді кольорового триканального RGB-зображення, його необхідно перетворити до «сі-

рого» (зображення у відтінках сірого). З огляду на неоднакову сприйнятливість людського зору до різних складових спектра, використовують наступну схему (рисунок 1.4).

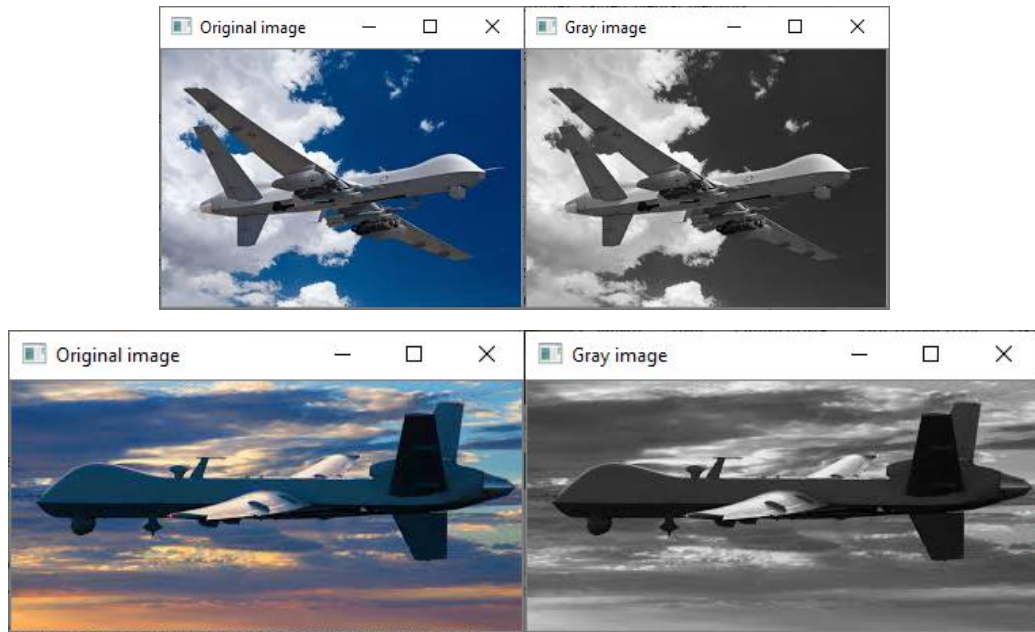


Рисунок 1.4 – Результат приведення зображення до виду в відтінках сірого

Поліпшення контрастності зображення шляхом корекції гістограм. Камери і детектори зображень повинні, як правило, займатися обробкою не лише контрасту, а й світла від датчиків зображення. У стандартній камері затвор і діфрагма об'єктива поперемінно подають на датчики або занадто багато, або занадто мало світла. Часто діапазон контрастів занадто великий для датчиків. Отже, існує компроміс між збором даних темних областей (наприклад, тіней), які вимагають збільшення часу експозиції, і світлих областей, які вимагають малого часу експозиції, щоб уникнути «пересвіту».

Після того, як знімок зроблений, можна спробувати розширити динамічний діапазон зображення. Найбільш часто використовуваним методом для цього є корекція гістограм. Будується графічне представлення розподілу інтенсивності зображення у вигляді гістограми, що визначає кількість пікселів для кожного розглянутого значення інтенсивності в діапазоні градацій сірого від 0 до 255 (рисунок 1.5).

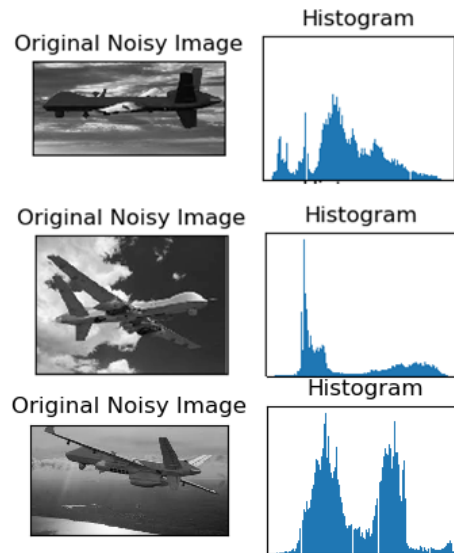


Рисунок 1.5 – Визначення гістограм інтенсивності
для зображень у відтінках сірого

Вирівнювання має на увазі відображення одного розподілу (даної гістограми) в інший розподіл (ширше і однорідний розподіл значень інтенсивності – в ідеалі рівномірний розподіл), щоб значення інтенсивності розподілялися по всьому діапазону. Алгоритм нормалізує яскравість і збільшує контраст зображення (рисунок 1.6).

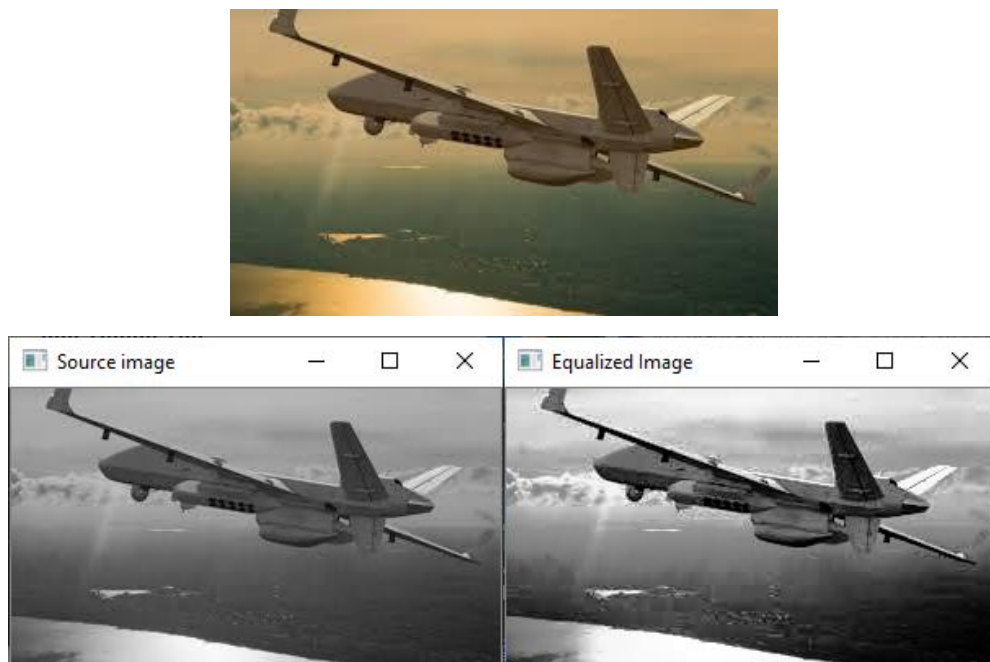


Рисунок 1.6 – Результат вирівнювання гістограми сірого зображення

Використання порогового перетворення. Хороший результат для визначення контуру може бути отриманий для зображень в бінарному (чорно-білому) поданні. У цьому випадку гранична точка однозначно знаходиться по зміні значення інтенсивності пікселя.

Таке перетворення бінаризації може бути визначено за допомогою призначення порогової величини. Початкове зображення вже має бути зображенням у відтінках сірого. Якщо значення пікселя менше порогового значення, воно встановлюється на 0, в іншому випадку встановлюється максимальне значення.

Можуть бути використані різні варіації алгоритму порогової бінаризації, представленої у таблиці 1.1 і на рисунку 1.7.

Таблиця 1.1 – Варіанти типу порогової бінаризації

THRESH_BINARY	$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$
THRESH_BINARY_INV	$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$
THRESH_TRUNC	$\text{dst}(x, y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$
THRESH_TOZERO	$\text{dst}(x, y) = \begin{cases} \text{src}(x, y) & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$
THRESH_TOZERO_INV	$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$

Природно, результат порогового перетворення залежить від величини значення порога. На рисунках 1.8 і 1.9 представлені результати бінаризації для різних значень порога.

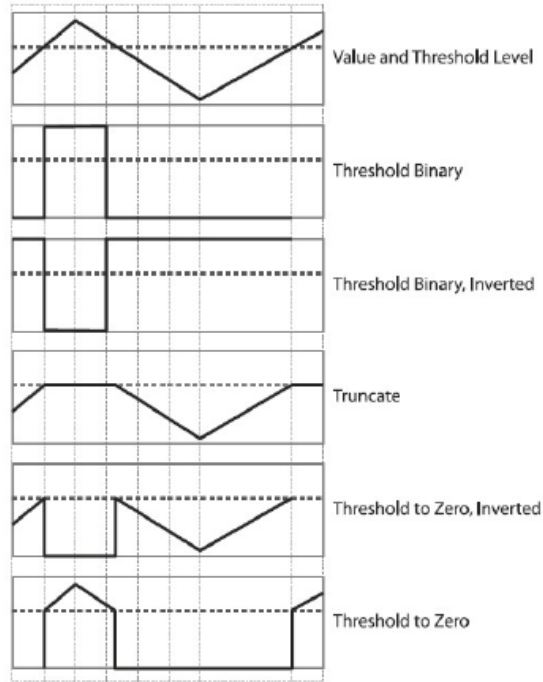


Рисунок 1.7 – Схеми різних типів порогового перетворення

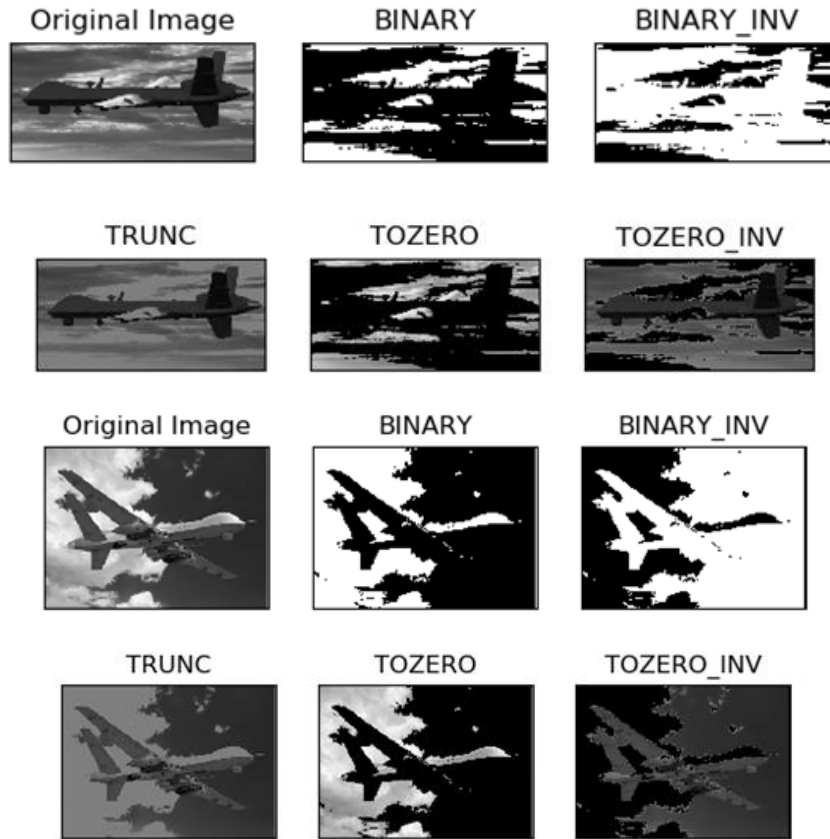


Рисунок 1.8 – Результати застосування різних типів порогового перетворення

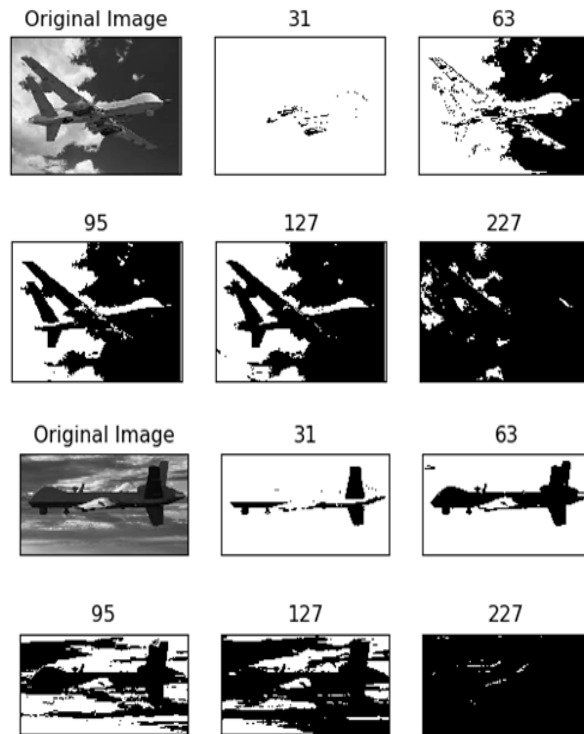


Рисунок 1.9 – Результати бінарizaції для різних значень порога

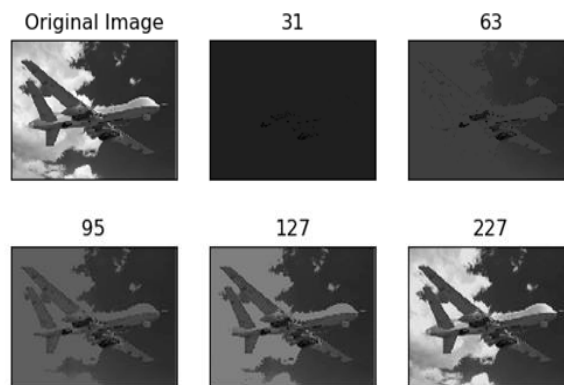


Рисунок 1.10 – Результати в режимі TRUNC для різних значень порога

Використання адаптивного порога. Використання фіксованого порога може бути не виправданим для різних умов освітлення в різних областях. У цьому випадку застосовують адаптивну установку порогових значень. Алгоритм визначає поріг для пікселя на основі невеликої області навколо нього. Таким чином, ми отримуємо різні порогові значення для різних областей одного і того ж зображення, що дає кращі результати для зображень з різним освітленням, як показано на рисунку 1.10 для різних режимів усереднення.

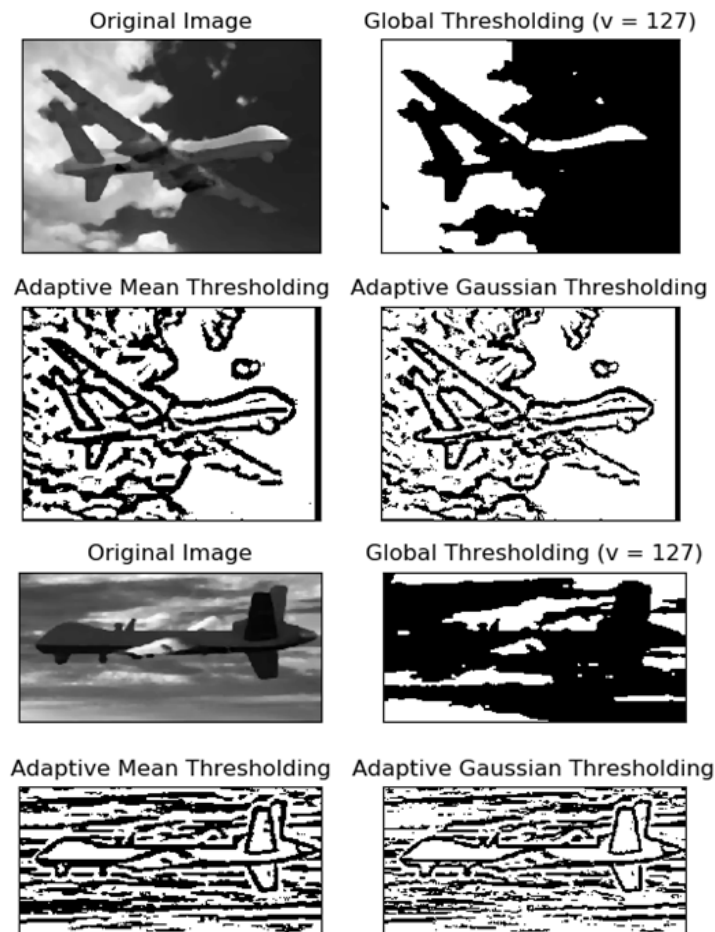


Рисунок 1.11 – Результати адаптивної бінаризації для різних режимів усереднення

1.3 Формальна та змістовна постановка задачі

Для розпізнавання літальних об'єктів різних класів представлених у відеопотоці, під різними кутами, з різним ступенем освітленості необхідно виконати:

- скласти репрезентативну вибірку зображень різних класів літальних об'єктів з різними кутами повороту, різним ступенем освітленості;
- провести перетворення зображень з вибірки у вигляд зображень у відтінках сірого;
- провести поліпшення контрастності зображень;

- провести нормалізацію даних з навчальної вибірки;
- здійснити навчання моделі для розпізнавання літальних об'єктів у відеопотоці в оптичному та інфрачервоному діапазоні частот;
- оцінити ефективність навчання нейронної мережі.

1.4 Постановка задач дослідження

Обробку відеозображення наземними засобами стеження за повітряною обстановкою можна розбити на ряд завдань. Серед них можна виділити наступні завдання: виявлення рухомого ПО; визначення характеристик об'єкта (дальність, розмір, швидкість, маневреність і т. п.); визначення типу ПО. Ці завдання можуть вирішуватися як поетапно, так і одночасно. Результатом рішення може бути виявлення всіх об'єктів на одному кадрі з визначенням їх типу і наступним їх супроводом в послідовності відеокадрів або виявлення рухомих об'єктів по відеоряду з подальшою ідентифікацією об'єктів на кожному кадрі. У такому підході ми припускаємо, що рухомий об'єкт або об'єкти спочатку будуть виявлені і локалізовані на відеозображенні, а потім буде вирішуватися завдання віднесення кожного об'єкта до певного класу.

В області розпізнавання образів на зображенні прийнята наступна термінологія для визначення завдань, представлена на рисунку 1.12.



Рисунок 1.12 – Різні завдання розпізнавання образів на зображенні

Детектування – визначення класу кожного об’єкта на зображенні. Класифікація – визначення класу одного об’єкта на зображенні. Класифікація з локалізацією – класифікація одного об’єкта і вказівкою його місця на зображенні.

Таким чином, в цій роботі досліджується та порівнюються навчання нейронної мережі з різними початковими моделями для трансферного навчання.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Стадія попередньої обробки

Попередня обробка зображень є важливою частиною системи і може сильно вплинути на максимальну точність, якої модель досягає під час навчання. Як мінімум, зображення потрібно декодувати та змінювати розмір, щоб вони відповідали моделі.

Однак простого декодування та зміни розміру буде недостатньо, щоб отримати хорошу точність. Набір навчальних даних ImageNet містить 1 281 167 зображень. Один прохід через набір навчальних зображень називають епохою. Під час навчання моделі знадобиться кілька проходів через навчальний набір даних, щоб покращити її можливості розпізнавання зображень.

Надзвичайно корисно постійно змінювати зображення перед подачею їх у модель і робити це таким чином, щоб конкретне зображення дещо відрізнялося в кожному епоху. З одного боку, добре розроблений етап попередньої обробки може значно підвищити можливості розпізнавання моделі. З іншого боку, занадто простий етап попередньої обробки може створити штучну стелю на максимальній точності, якої та сама модель може досягти під час навчання.

Нормалізація. Також називають перемасштабуванням даних, це процес проектування пікселів даних зображення (інтенсивності) до попередньо визначеного діапазону (зазвичай $(0,1)$ або $(-1, 1)$). Це зазвичай використовується для різних форматів даних, треба нормалізувати всі з них, щоб застосувати до них ті самі алгоритми.

Нормалізація зазвичай застосовується для перетворення значень пікселів зображення на типовий або більш звичний вигляд.

Його переваги включають:

– приведення зображень до одного виду. Наприклад, масштабування всіх зображень до однакового діапазону $[0,1]$ або $[-1,1]$ дозволяє всім зображенням вносити однаковий внесок у загальні втрати, а не тоді, коли інші зображення

мають високий і низький діапазон пікселів. сильні і слабкі втрати відповідно;

– забезпечує стандартну швидкість навчання. Оскільки зображення з високим рівнем пікселів вимагають низької швидкості навчання, а зображення з низькими пікселями – високою швидкістю навчання, повторне масштабування допомагає забезпечити стандартну швидкість навчання для всіх зображень.

Збільшення даних. Збільшення даних – це процес внесення незначних змін до наявних даних для збільшення їх різноманітності без збору нових даних. Це техніка, яка використовується для збільшення набору даних. Стандартні методи збільшення даних включають горизонтальне та вертикальне перегортання, обертання, обрізання, зріз тощо. Збільшення даних допомагає запобігти засвоєнню нейронною мережею невідповідних функцій. Це призводить до кращої продуктивності моделі.

2.2 Inception v3

Метою факторізації згорток є зменшення кількості з'єднань/параметрів без зниження ефективності мережі.

Дві згортки 3×3 замінюють одну згортку 5×5 наступним чином:

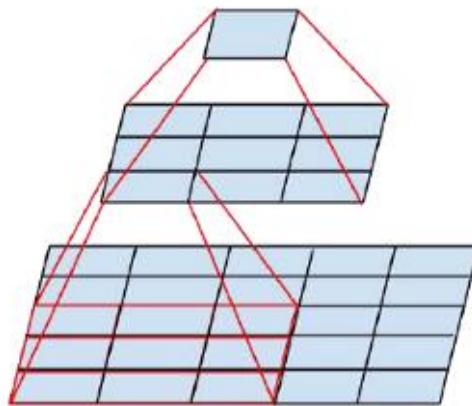


Рисунок 2.1 – Дві згортки 3×3 , які замінюють одну згортку 5×5

Дві згортки 3×3 замінюють одну згортку 5×5 .

При використанні 1 шару фільтра 5×5 кількість параметрів $= 5 \times 5 = 25$.

Використовуючи 2 шари фільтрів 3×3 , кількість параметрів $= 3 \times 3 + 3 \times 3 = 18$.

Кількість параметрів зменшено на 28%.

За допомогою цієї техніки одним із нових початкових модулів стає:

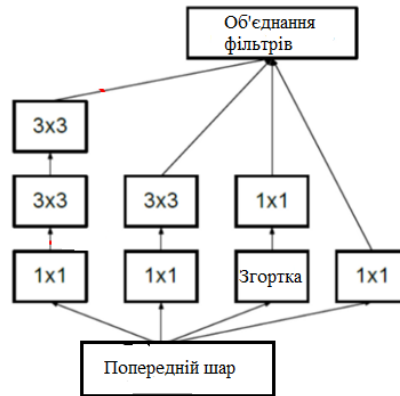


Рисунок 2.2 – Початковий модуль з використанням факторизації

Початковий модуль із використанням факторизації

Одна згортка 3×1 , а потім одна згортка 1×3 замінює одну згортку 3×3 наступним чином:

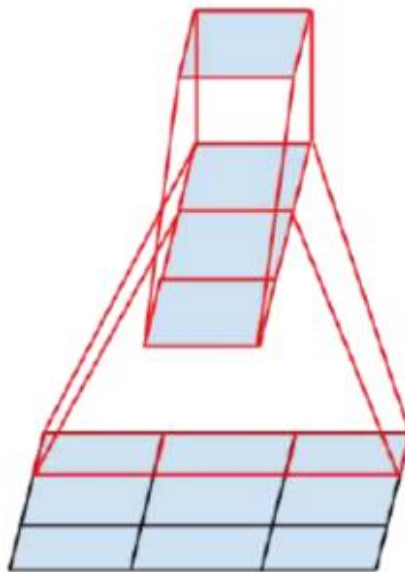


Рисунок 2.3 – Заміна згортки 3×3 згорткою 3×1

При використанні фільтра 3×3 кількість параметрів $= 3 \times 3 = 9$.

При використанні фільтрів 3×1 і 1×3 кількість параметрів $= 3 \times 1 + 1 \times 3 = 6$.

Кількість параметрів зменшено на 33%.

Якщо ми використовуємо два фільтри 2×2 , кількість параметрів $= 2 \times 2 \times 2 = 8$.

Кількість параметрів зменшено лише на 11%

За допомогою цієї техніки одним із нових початкових модулів стає (рисунок 2.4).

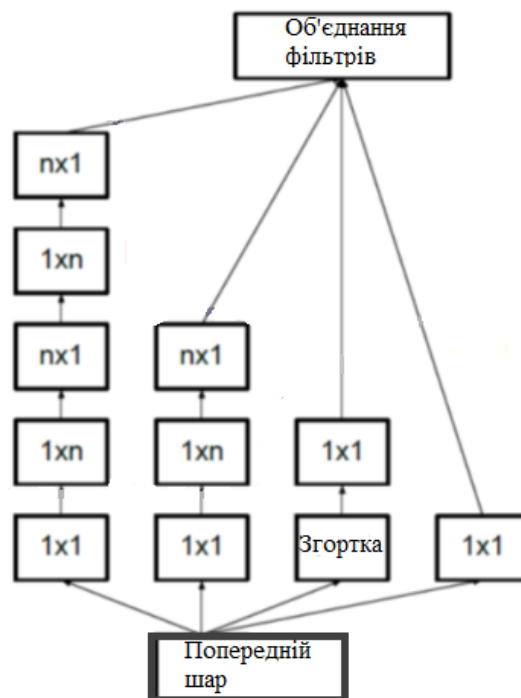


Рисунок 2.4 – Початковий модуль з використанням асиметричної факторизації

При факторизації кількість параметрів зменшується для всієї мережі, менша ймовірність перенавчання, а отже, мережа може піти глибше.

На верхній частині останнього шару 17×17 використовується лише один допоміжний класифікатор замість двох допоміжних класифікаторів.

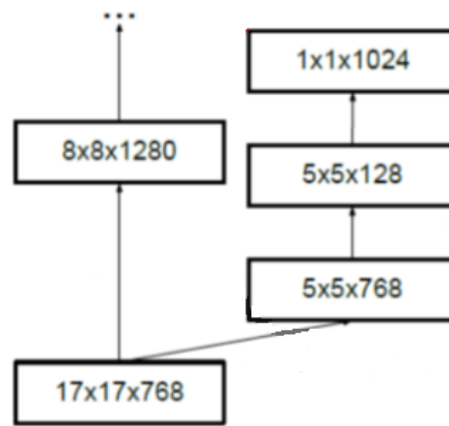


Рисунок 2.5 – Допоміжний класифікатор

Допоміжний класифікатор виступає як регуляризація. У Inception-v3 допоміжний класифікатор використовується як регуляризатор.

2.3 ResNet

ResNet дає змогу тренувати до сотень або навіть тисяч шарів і при цьому досягає вражаючої продуктивності.

Користуючись перевагами його потужної репрезентативної здатності, було збільшено продуктивність багатьох програм комп'ютерного зору, крім класифікації зображень, таких як виявлення об'єктів та розпізнавання обличчя.

Основна ідея ResNet полягає в тому, щоб запровадити так зване «підключення до ідентифікації», яке пропускає один або кілька шарів, як показано на рисунку 2.6.

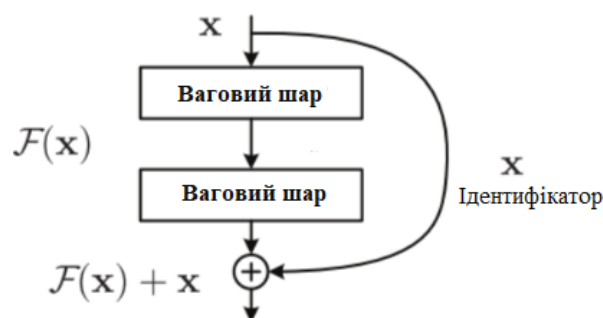


Рисунок 2.6 – Залишковий блок

Ідентифікаційні швидкі з'єднання можуть використовуватися безпосередньо, коли вхід та вихід мають однакові розмірності. Коли розмірності збільшуються, він розглядає два варіанти:

- швидке з'єднання виконує зіставлення ідентифікаторів з додатковими нулями, доданими збільшення розмірності. Ця опція не вводить додаткових параметрів;

- проекція швидкого з'єднання використовується для зіставлення розмірностей (виконано за допомогою 1×1 згорток).

Стекування шарів не повинно знижувати продуктивність мережі, тому що ми можемо просто скласти відображення ідентифікацій (рівень, який нічого не робить) у поточній мережі, і результуюча архітектура буде працювати так само. Це вказує на те, що більш глибока модель не повинна створювати похибку навчання, більшу, ніж її більш дрібні аналоги. Вони припускають, що дозволити складеним шарам відповідати залишковому відображенню легше, ніж дозволити їм безпосередньо відповідати бажаному базовому відображенню. І залишковий блок дозволяє йому робити саме це.

Автори удосконалили залишковий блок і запропонували варіант залишкового блоку до активації, в якому градієнти можуть безперешкодно перетікати через ярлики з будь-яким іншим попереднім шаром. Насправді, використовуючи вихідний залишковий блок у, навчання 1202-шарового ResNet призвело до гіршої продуктивності, ніж його 110-шаровий аналог.

Автори за допомогою експериментів продемонстрували, що тепер вони можуть тренувати 1001-шаровий глибокий ResNet, щоб перевершувати його більш дрібні аналоги. Завдяки своїм переконливим результатам ResNet швидко став однією з найпопулярніших архітектур у різних задачах комп'ютерного зору.

2.4 DenseNet

Щільно зв'язані згорткові мережі (DenseNets) — це наступний крок на шляху до постійного збільшення глибини глибоких згорткових мереж.

Проблеми виникають, коли вони заглиблюються. Це пояснюється тим, що шлях для інформації від вхідного шару до вихідного шару (і для градієнта в протилежному напрямку) стає настільки великим, що вони можуть зникнути, не досягнувши іншої сторони.

Автори вирішують задачу забезпечення максимального інформаційного (та градієнтного) потоку. Для цього вони просто з'єднують кожен шар безпосередньо один з одним.

Замість того, щоб черпати репрезентаційну силу з надзвичайно глибоких або широких архітектур, DenseNet використовує потенціал мережі шляхом повторного використання функцій.

Завдяки такому підключенню DenseNet вимагає менше параметрів, ніж еквівалентні традиційні згорткові нейронні мережі, оскільки немає необхідності вивчати зайві карти функцій.

Крім того, деякі варіанти ResNet довели, що багато шарів ледве вносять свій внесок і їх можна відкинути. Насправді кількість параметрів ResNet велике, тому що кожен шар має свої ваги, які потрібно вивчити. Натомість шари DenseNet дуже вузькі (наприклад, 12 фільтрів), і вони просто додають невеликий набір нових карт функцій.

Іншою проблемою з дуже глибокими мережами були проблеми з навчанням через згаданий потік інформації та градієнти. DenseNet вирішує цю проблему, оскільки кожен шар має прямий доступ до градієнтів із функції втрат і вхідного зображення.

Традиційні нейронні мережі з прямим зв'язком підключають вихід шару до наступного шару після застосування комбінації операцій.

Ми вже бачили, що зазвичай цей композит включає операцію згортки або об'єднання шарів, пакетну нормалізацію та функцію активації. Рівняння для

цього буде таким:

$$x_l = H_l(x_{l-1}). \quad (2.1)$$

ResNet розширив цю поведінку, включаючи пропуск з'єднання, переформулюючи це рівняння на:

$$x_l = H_l(x_{l-1}) + x_{l-1}. \quad (2.2)$$

Перша різниця між DenseNet та ResNet починається саме тут. DenseNet не підсумовує вихідні карти об'єктів шару з вхідними картами об'єктів, а об'єднує їх. Отже, рівняння знову перетворюється на:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]). \quad (2.3)$$

Групування карт об'єктів не може бути виконано, коли їх розміри різні. Незалежно від того, чи є групування додаванням чи конкатенацією. Тому так само, як використовується для ResNet, DenseNet поділяються на DenseBlocks, де розміри карт функцій залишаються незмінними в межах блоку, але кількість фільтрів змінюється між ними. Ці шари між ними називаються шарами переходу і забезпечують зниження дискретизації із застосуванням пакетної нормалізації, згортки 1 x 1 і шарів об'єднання 2 x 2.

Оскільки ми об'єднуємо карти об'єктів, розмір каналу збільшується на кожному шарі. Якщо ми змусимо H_l створювати k карт об'єктів щоразу, то ми можемо узагальнити для першого шару:

$$k_l = k_0 + k \cdot (l - 1). \quad (2.4)$$

Цей гіперпараметр k є швидкістю росту. Швидкість росту регулює, скі-

льки інформації додається в мережу на кожному рівні. Кожен шар має доступ до своїх попередніх карт об'єктів, а отже, і до сукупних знань. Кожен шар потім додає нову інформацію до цього колективного знання в конкретних k інформаційних картах ознак.

2.5 MobileNet

Згорткові нейронні мережі стали дуже популярними в комп'ютерному зорі. Однак, щоб досягти більш високого ступеня точності, сучасні згорткові нейронні мережі стають глибшими і все складнішими. Такі мережі складно використовувати в реальних програмах.

MobileNet – це ефективна і портативна архітектура згорткової нейронної мережі, яка використовується в реальних програмах. MobileNet в основному використовує згортки, які можна розділити по глибині, замість стандартних згорток, які використовувалися в попередніх архітектурах, для створення більш легких моделей. MobileNet вводить два нових глобальних гіперпараметри (множник ширини і множник роздільної здатності), які дозволяють розробникам моделей компромісувати затримку або точність на швидкість і низький розмір залежно від їх вимоги.

MobileNet побудовано на шарах згортки, що розділяється по глибині. Кожен шар згортки, що розділяє по глибині, складається з глибинної згортки та точкової згортки. Підраховуючи глибинні та точкові згортки як окремі шари, MobileNet має 28 шарів. Стандартний MobileNet має 4,2 мільйона параметрів, які можна далі зменшено шляхом відповідної настройки гіперпараметра множника ширини.

Помножувач ширини α — це глобальний гіперпараметр, який використовується для побудови менших і менш дорогих моделей. Його значення знаходиться в межах від 0 до 1. Для даного шару і значення α , кількість вхідних каналів M стає $\alpha \cdot M$ і кількість вихідних каналів N стає $\alpha \cdot N$, отже, знижують-

ся вартість обчислень і розмір моделі за рахунок продуктивності. Вартість обчислень і кількість параметрів зменшуються приблизно в α^2 .

Другий параметр, введений у MobileNet, називається коефіцієнтом роздільної здатності ρ . Цей гіперпараметр використовується для зменшення роздільної здатності вхідного зображення, і це згодом зменшує вхідний сигнал для кожного шару на той самий коефіцієнт.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Формування навчальної вибірки

Для формування навчальної вибірки були знайдені відеоматеріали різних типів ЛО та розбиті на кадри, що дало змогу сформувати значну базу для навчання моделі.

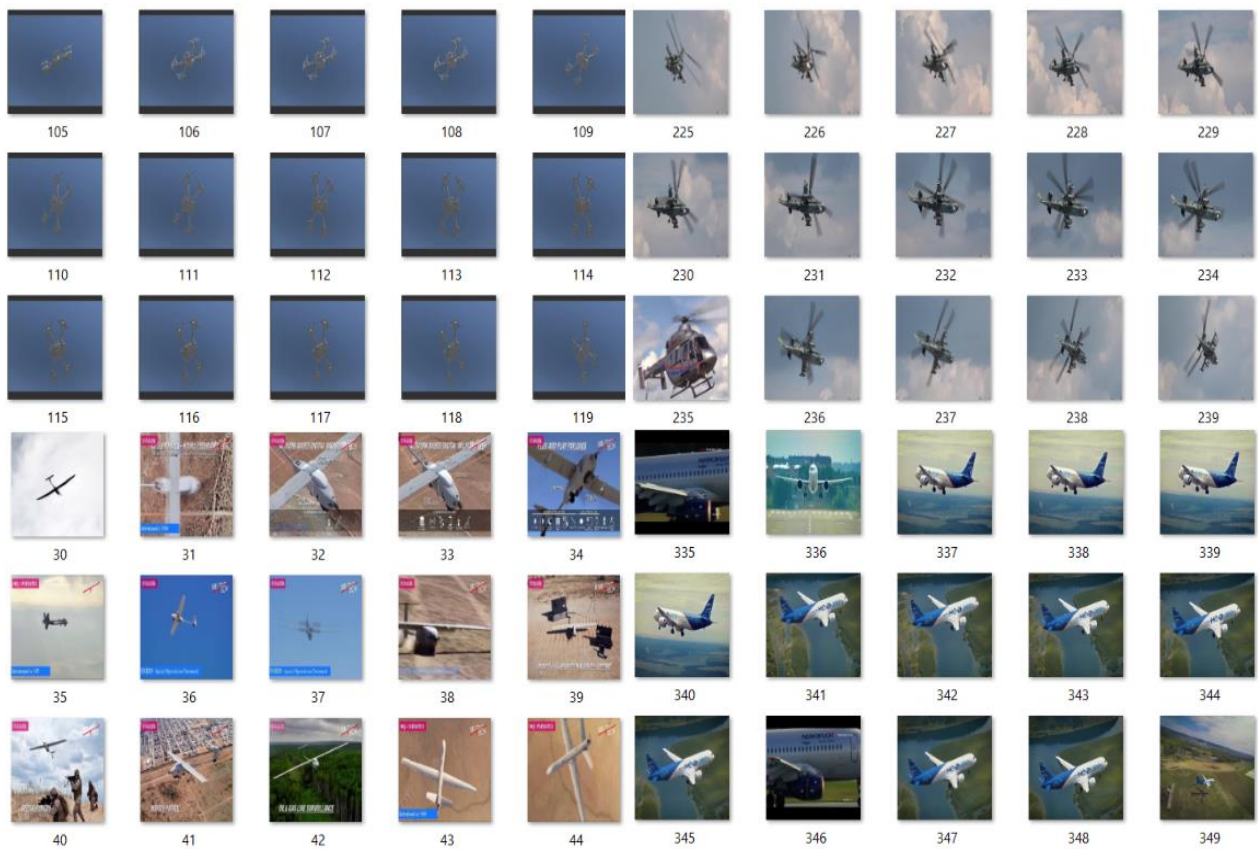


Рисунок 3.1 – Навчальна вибірка

Для кращої ефективності навчання, кольорові зображення з вибірки були перетворені до виду у відтінках сірого.

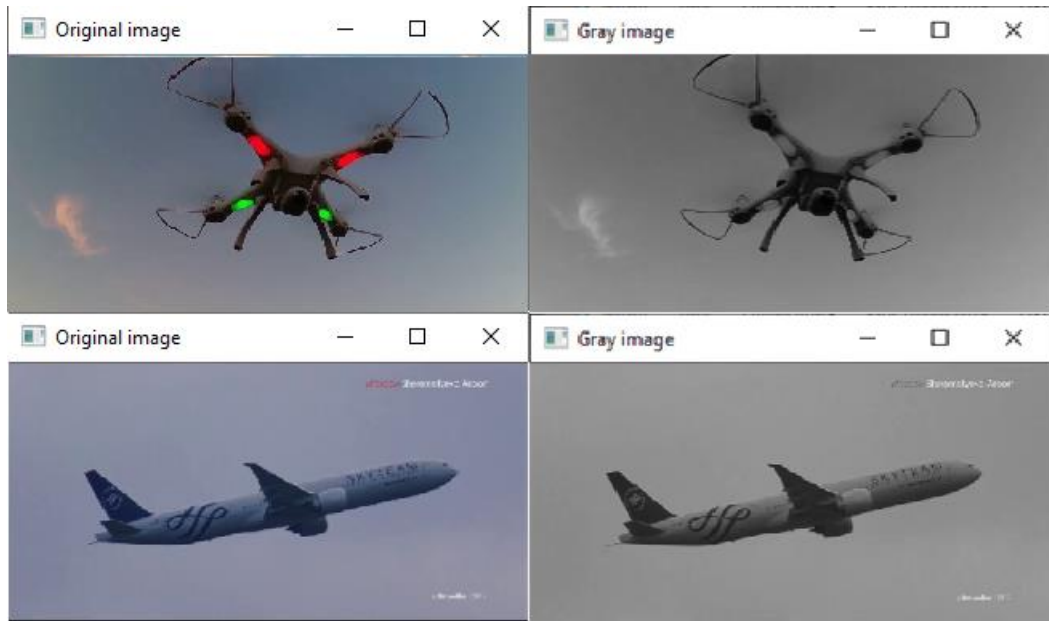


Рисунок 3.2 – Зображення квадрокоптера та літака у відтінках сірого

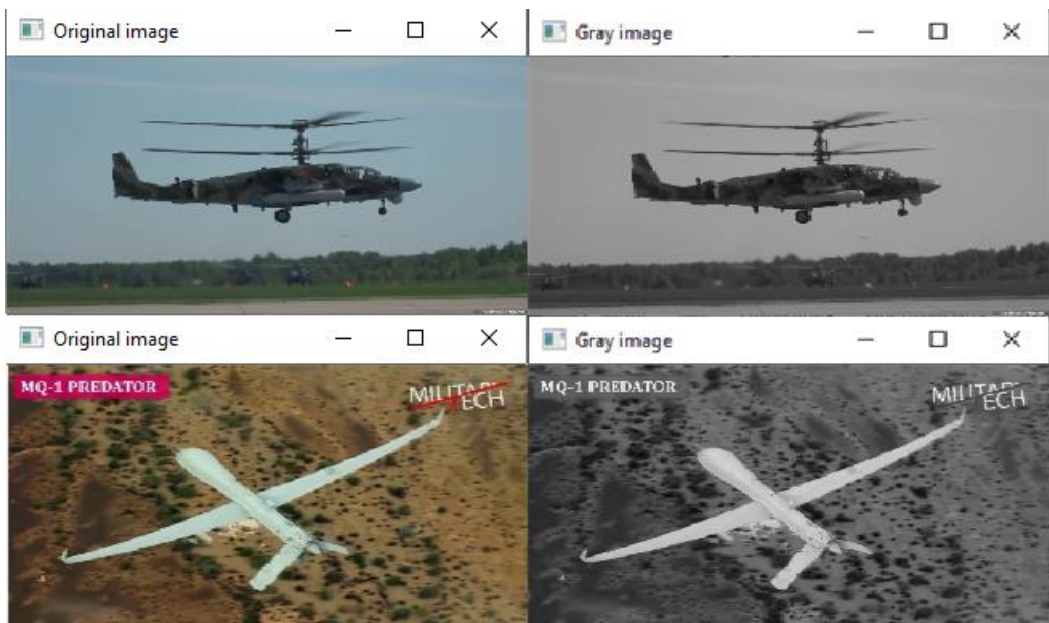


Рисунок 3.3 – Зображення гелікоптера та БПЛА у відтінках сірого

Після цього, для вирівнювання якості зображень, було зроблено поліпшення контрастності зображень. Це все робиться для прискорення роботи класифікатора.

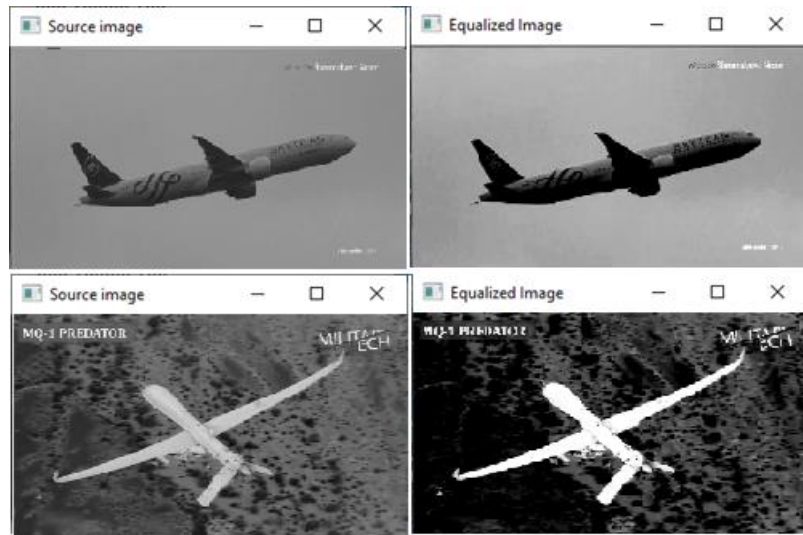


Рисунок 3.4 – Зображення літака та БПЛА з поліпшеною контрастністю



Рисунок 3.5 – Зображення квадрокоптера та гелікоптера з поліпшеною контрастністю

3.2 Алгоритм розв'язання задачі класифікації літальних об'єктів

Машинне навчання стає доступнішим, з'являється більше можливостей застосовувати цю технологію, використовуючи «готові компоненти». Наприклад, трансферне навчання дозволяє використовувати накопичений при вирішенні одного завдання досвід для вирішення іншої, аналогічної проблеми. Нейронна мережа спочатку навчається на великому обсязі даних, потім на цільово-

му наборі. Нейронні мережі, які використовуються для класифікації, зазвичай містять N вихідних нейронів в останньому шарі, де N - це кількість класів. Такий вихідний вектор сприймається як набір ймовірностей приналежності до класу. У нашій задачі розпізнавання ЛО кількість класів може відрізнятись від того, яке було у вихідному датасеті. У такому разі нам доведеться повністю викинути цей останній шар і поставити новий з необхідною кількістю вихідних нейронів. Найчастіше наприкінці класифікаційних мереж використовується повнозв'язний шар. Так як ми замінили цей шар, використовувати набуті ваги для нього вже не вдасться. Доведеться тренувати його з нуля, ініціалізувавши його ваги випадковими значеннями. Ваги для всіх інших шарів ми завантажуюмо з попередньої моделі.

Існують різні стратегії донавчання моделі. Ми скористаємося наступною: тренуватимемо всю мережу з кінця в кінець, а набуті ваги не фіксуватимемо, щоб дати їм трохи скоригуватися і підлаштуватися під наші дані. Такий процес називається тонким налаштуванням.

3.3 Опис програми

Python – це найпопулярніша високорівнева мова програмування з динамічною семантикою. Його використання знижує вартість розробки та обслуговування програм. Він легко взаємодіє з іншими мовами програмування.

Машинне навчання – це технологія, яка допомагає програмам на основі штучного інтелекту навчатися та видавати результати автоматично, без людського втручання. Одна з основних причин, чому Python використовується для машинного навчання, полягає в тому, що у нього є безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку.

Основною його перевагою є те, що ця мова не залежить від платформ, тому що підтримується багатьма з них, включаючи Linux, Windows та MacOS.

Код Python може використовуватися для створення програм для більшос-

ті операційних систем, а це означає, що це програмне забезпечення легко розповсюджувати та використовувати в цих системах без спеціальних інтерпретаторів.

Google Colab - це безкоштовне інтерактивне хмарне середовище для роботи з кодом від Google. Принцип у неї такий самий, як у решти онлайн-офісів компанії: вона дозволяє одночасно з колегами працювати з великою кількістю даних.

Головна його особливість — безкоштовні потужні графічні процесори GPU та TPU, завдяки яким можна займатися не лише базовою аналітикою даних, а й складнішими дослідженнями у галузі машинного навчання. З тим, що CPU обчислює годинами, GPU або TPU справляються за хвилини або секунди.

CPU – центральний процесор – мозок комп'ютера, який виконує операції з даними. Настільки універсальний, що може використовуватися майже всім завданням: від запису фотографій на флешку до моделювання фізичних процесів.

GPU – графічний процесор. Обробляє дані швидше, оскільки завдання виконує паралельно, а не послідовно, як CPU. Він заточений виключно під графіку, тому на ньому зручніше працювати із зображенням та відео, наприклад займатися 3D-моделюванням або монтажем.

TPU – тензорний процесор, розробка Google. Він призначений для тренування нейронних мереж. У цього процесора в рази вища продуктивність за більших обсягів обчислювальних завдань.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАНОВОГО ЕКСПЕРЕМЕНТУ ТА ЇХ АНАЛІЗ

Програми навчання мережі та її використання для класифікації написані на Python з використанням бібліотеки Keras.

Результати навчання нейронних мереж на основі різних початкових моделей приведені на рисунках:

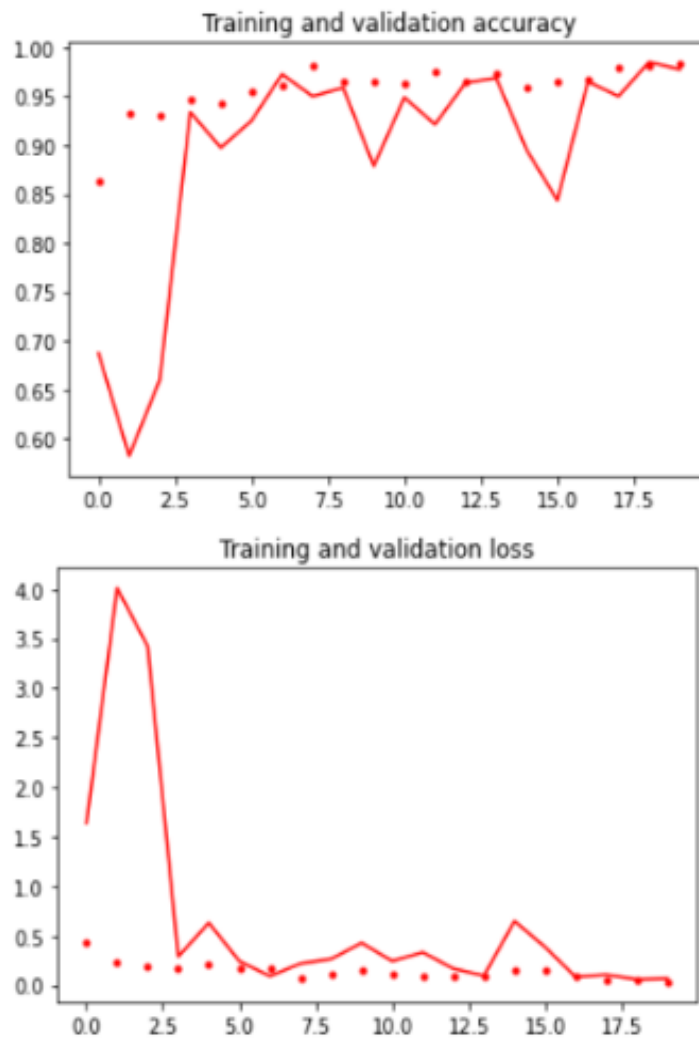


Рисунок 4.1 – Графік результату навчання нейронної мережі на основі MobileNet

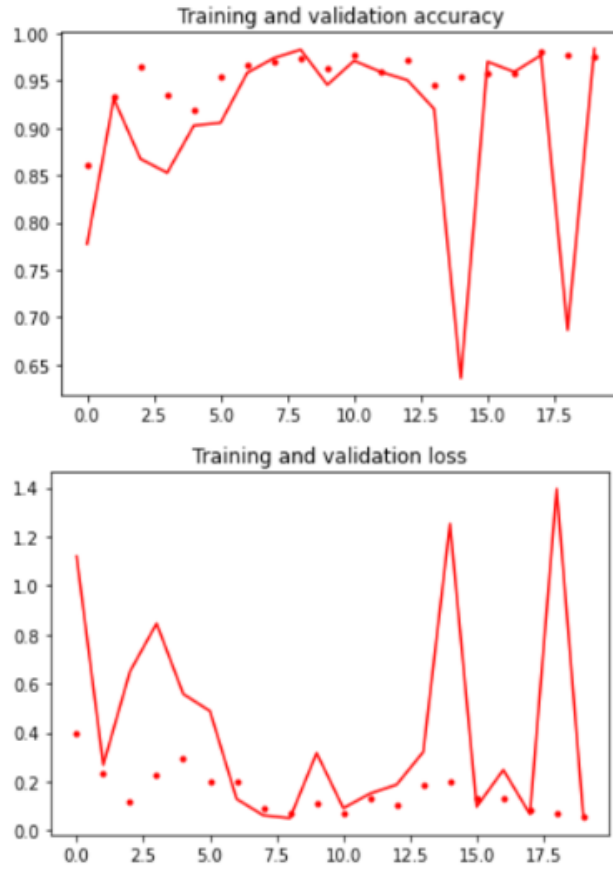


Рисунок 4.2 – Графік результату навчання нейронної мережі на основі ResNet

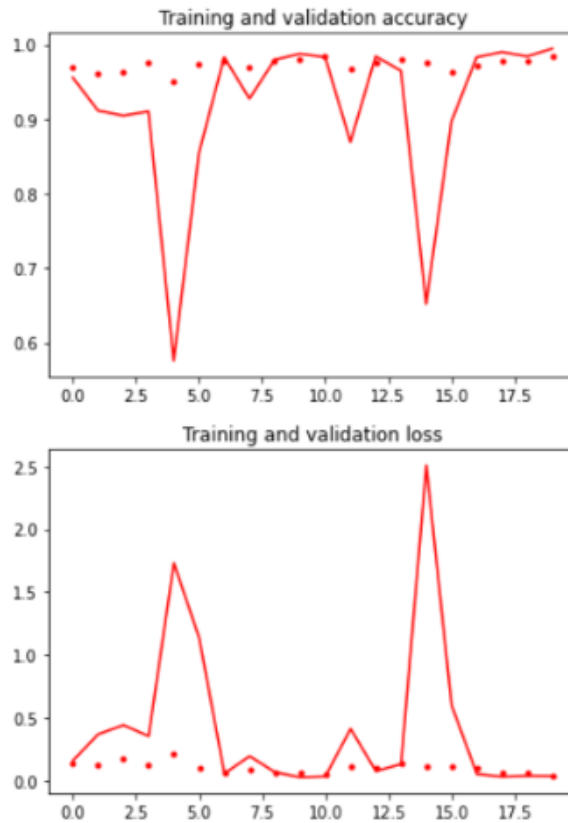


Рисунок 4.3 – Графік результату навчання нейронної мережі на основі DenseNet

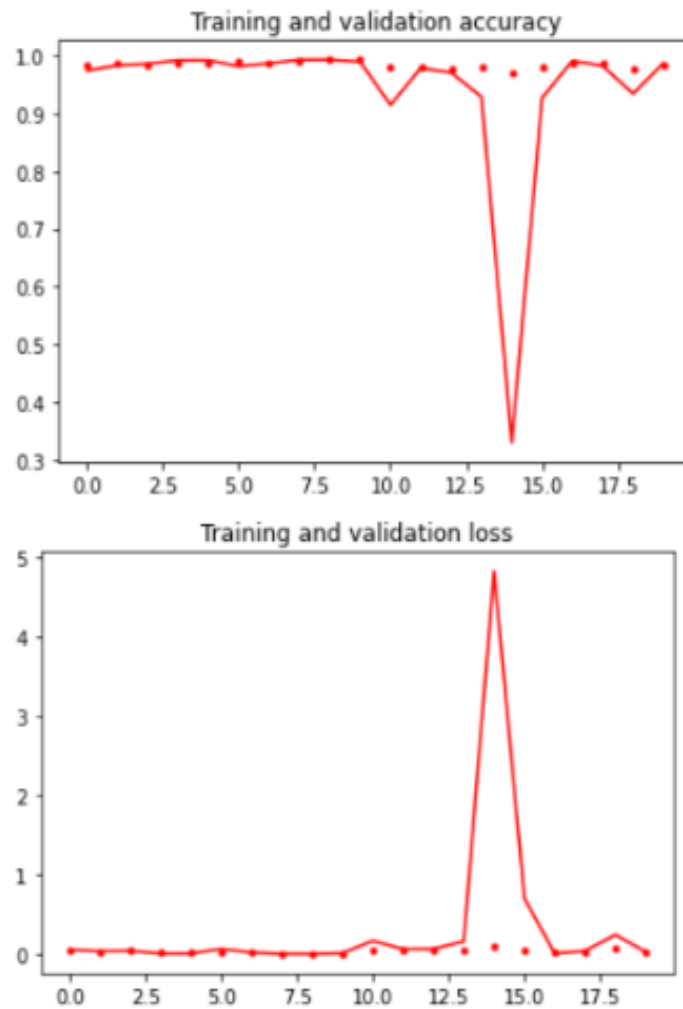


Рисунок 4.4 – Графік результату навчання нейронної мережі на основі InceptionV3

Для вирішення задачі по детектуванню та класифікації літальних об'єктів у відеопотоці в оптичному та інфрачервоному діапазоні частот, виявилась нейронна мережа, яка була навчена на основі моделі MobileNet. З графіків тренування ми бачимо, що точність класифікації була досягнута при навчанні нейронної мережі на основі MobileNet, також ми бачимо що вона показала найменшу величину похибки на всьому етапі навчання. До того ж великою перевагою цієї нейронної мережі є те, що вона не потребує великої обчислювальної потужності пристрою, на якій вона буде використовуватися.

ВИСНОВКИ

У роботі був проведений аналіз чотирьох моделей для навчання нейронних мереж, які були використані для задачі детектування та класифікації літальних об'єктів у відеопотоці в інфрачервоному та оптичному діапазоні частот. Такі як Inception V3, ResNet, DenseNet та MobileNet. Найкращих результатів досягла нейронна мережа на основі моделі MobileNet, вона досягла 98% точності на наборі даних, який не використовувався у навчальній вибірці.

Був проведений аналіз методів та застосування обробки зображень, такі як перетворення тестових зображень до виду у відтінках сірого, поліпшення контрастності, приведення до бінарного вигляду та використання адаптивного порогу.

Розроблено програмне забезпечення для класифікації літальних об'єктів у відеопотоці в інфрачервоному та оптичному діапазоні частот.

Результати роботи підтверджують, що запропонований метод побудови системи розпізнавання типів повітряних об'єктів призведе до спрощення вимог реалізації апаратних засобів зі збільшенням точності розв'язання завдання розпізнавання повітряної обстановки. Це, у свою чергу, створює передумови підвищення мобільності і розширення сфери застосування таких систем аж до індивідуальних пристроїв виявлення.

У роботі показано, що варіантом подолання великоваговості навчання глибоких мереж з нуля частково є використання трансферного навчання. Цей метод передбачає, що попередньо навчена універсальна мережа доучується на специфічних зображеннях повітряних об'єктів. Застосування трансферного навчання дозволяє уникнути обмежень, пов'язаних з часом навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mönning, N., Manandhar S. Evaluation of Complex-Valued Neural Networks on Real-Valued Classification Tasks. 2018. arXiv:1811.12351.
2. Toward aircraft recognition with convolutional neural networks / Mash R., Becherer N., Woolley B., Pecarina J. 2016. DOI: 10.1109/NAECON.2016.7856803.
3. Yesilevskyi V., Teviashev A., Koliadin A. Transfer learning in aircraft classification // IST-2019. The 8-th International Scientific and Technical Conference September 9 – 14, Kobleve-Kharkiv, Ukraine. 2019. P. 132–135.
4. Tomasi S., Tomasi C. Good Features to Track // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997. P. 593–600.
5. Cornelis K., Cornelis N., Leibe B., Van Gool L. Dynamic 3D scene analysis from a moving vehicle // Computer Vision and Pattern Recognition (CVPR'07), 2007. P. 1-8.
6. Понс Ж., Форсайт Д. Компьютерное зрение. Современный подход. Москва: издательство Вильямс, 2004. С. 400–470.
7. Rethinking the Inception Architecture for Computer Vision / C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens [та ін.] // IEEE Conference on Computer Vision and Pattern Recognition (CVPR) – Las Vegas, NV, USA. 2016. P. 63–78.