

## ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.89

<https://doi.org/10.35546/kntu2078-4481.2022.3.5>

Н. І. БОЙКО

Національний університет «Львівська політехніка»

ORCID: 0000-0002-6962-9363

Р. Т. ОМЕЛЮХ

Національний університет «Львівська політехніка»

ORCID: 0000-0001-6859-2792

АЛГОРИТМ ПАРАЛЕЛІЗАЦІЇ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ЗА НАПРЯМКОМ  
“SMART HOUSE SECURITY”

Було проаналізовано велику кількість літературних джерел щодо доцільності вибору саме моделі CNN. В них описувались всі сильні сторони даної структури нейронної мережі. Розглядається задача класифікації за допомогою нейронної моделі CNN та OpenCV. За допомогою OpenCV можна обробляти відеопотоки у реальному часі, а вже за допомогою тренованої моделі класифікувати розкадровані зображення.

Таким чином ціллю дослідження є моделювання архітектури моделі таким чином, щоб мінімізувати час класифікації та збільшити її точність. Також покращити швидкість класифікації можна за допомогою багатопоточності.

Для повного аналізу дослідження слід розглянути принцип роботи CNN. Це клас глибоких нейронних згорткових мереж, які можуть розпізнавати і класифікувати певні ознаки зображень і широко використовуються для аналізу візуальних образів. У ході проведення експериментів використовується модель класифікації CNN. Для порівняння результатів підбрані різні параметри. Такі як оптимайзери та кількості epoch, щоб отримати інформацію про найоптимальніші варіанти.

Під час проведення експериментів відбувались тренування з різними розмірами epoch та оптимайзерами. Для покращення швидкості було підбрано 2 найбільш результативні, а саме “Adam” та “RMSProp”, тому що при виборі інших – точність різко падала до значень менше 40%. Для визначення ефективності одного над іншим, ми виконували тренування та класифікацію зображень з однаковими параметрами. При збільшенні кількості epoch можна помітити позитивну динаміку по точності і негативну динаміку по втратах на обох моделях при збільшенні кількості epoch. Це підтверджує результативність CNN для свого прямого призначення, а саме класифікації зображень.

Також у роботі були проведені експерименти над розміром зображення. В результаті даного експерименту час тренування зменшився ще на 8 с на одній епосі, на основі того, що розмір вхідної матриці пікселів зображення став меншим.

Також проведені експерименти довели, що при оптимальному підборі розміру зображення можна також збільшити швидкість обробки. Після підбору усіх оптимальних вхідних параметрів нейронної мережі був проаналізований процес розпаралелення класифікації. В роботі пропонуються способи оптимізації моделі.

**Ключові слова:** нейронна мережа, оптимайзер, модель, конволюційний шар, багатопоточність, згорткова нейронна мережа, параметр.

N. I. BOYKO

Lviv Polytechnic National University

ORCID: 0000-0002-6962-9363

R. T. OMELIUKH

Lviv Polytechnic National University

ORCID: 0000-0001-6859-2792

PARALLELIZATION ALGORITHM OF IMAGE CLASSIFICATION ACCORDING  
TO “SMART HOUSE SECURITY”

Many literature sources were analyzed regarding the practicality of choosing the CNN model. They described all the strengths of this neural network structure. The problem of classification using the neural model CNN and OpenCV is considered. With the help of OpenCV, you can process video streams in real-time, and already with the help of a trained model, you can classify interlaced images.

Thus, the study aims to model the architecture of the model in such a way as to minimize the classification time and increase its accuracy. You can also improve classification speed using multithreading.

For a complete analysis of the study, the working principle of CNN should be considered. It is a class of deep neural convolutional networks that can recognize and classify certain image features and is widely used for visual image analysis.

*During the experiments, the CNN classification model is used. To get information about the most optimal options, various parameters are selected to compare the results, such as optimizers and the number of epochs.*

*During the experiments, training took place with different sizes of epochs and optimizers. To improve the speed, the two most effective ones were selected, namely “Adam” and “RMSProp”, because when choosing the others, the accuracy dropped sharply to values less than 40%. To determine the effectiveness of one over the other, we performed training and classification of images with the same parameters. When the number of epochs increases, it is possible to notice positive dynamics in terms of accuracy and negative dynamics in terms of losses on both models when the number of epochs increases. This confirms the effectiveness of CNN for its direct purpose, namely image classification.*

*Experiments on image size were also carried out in work. As a result of this experiment, the training time was reduced by another 8 s per epoch because the input image pixel matrix size became smaller.*

*Also, the conducted experiments proved that with the optimal image size selection, the processing speed could also be increased. After selecting all the optimal input parameters of the neural network, the parallelization process of the classification was analyzed. The work offers ways to optimize the model.*

**Key words:** neural network, optimizer, model, convolutional layer, multithreading, convolutional neural network, parameter.

### Постановка проблеми

Кожного дня мільйони людей працюють над покращенням суспільного життя. Впроваджуються нові технології та рішення, які допомагають спрощувати ті чи інші процеси. Ще декілька років тому не можливо було уявити, що за допомогою декількох кліків та автоматика людина зможе контролювати всі процеси у власному будинку, навіть без фізичної присутності у ньому. Це і дало поштовх розвитку сфері “Smart House”. Хоч і багато чого досягнуто у цьому напрямку, але в певних аспектах є ще куди розвиватись. Особливу увагу у цьому напрямку зосереджено на безпеці та комфорті людини, адже не виключається небезпечний побутовий фактор – пожежа в будинку.

Проблематика раннього виявлення пожежі дуже гостро постає в сучасному світі, оскільки вони становлять постійну загрозу для людських життів. Шкода завдана в результаті пожеж настільки суттєва, що ідея раннього попередження набуває нових масштабів. Найбільш дієвим способом виявлення пожежі на ранній стадії та для того, щоб уникнути втрат є встановлення системи пожежної сигналізації. Система пожежної сигналізації оснащена різними взаємозалежними датчиками, які працюють разом для виявлення пожежі та оповіщення [1]. Загально прийняті методи протипожежної безпеки не є дуже ефективними, тому що в результаті дії різних факторів їх точність та час спрацювання знижується, а також існує ймовірність фіктивного спрацювання. Основна проблема фізичних датчиків, те що вони можуть розпізнати пожежу, коли вони встановлені поблизу місця займання, що в результаті зменшує їхню результативність у великих приміщеннях та на відкритому просторі. Ще одним недоліком стає неінформативність, тому що з них ми не можемо отримати повної інформації про розташування де почалось займання та величину площі що охоплена вогнем. Також датчики в тричі дорожчі, ніж звичайні камери спостереження і спеціалізуються лише на вузькому спектрі завдань.

З розвитком технологій з’являються нові методи вирішення цієї проблеми. В більшості випадків вони базуються на обробці набору фото/відео з камер спостереження та використання методів глибокого машинного навчання. Існує декілька ефективних алгоритмів для обробки інформації, але вони мають ряд недоліків [2], оскільки вхідний датасет містить зображення з різним кольором текстурою та ракурсом, що ускладнює досягнення високої точності даними алгоритмами.

Одним з найбільш ефективних алгоритмів вважається [3], алгоритм на основі конволюційних нейронних мереж (CNN). Він дозволяє найкраще виявити ключові характеристики кольору із зображення, необхідні для точніших результатів у візуальному розпізнаванні.

### Аналіз останніх досліджень і публікацій

Даному напрямку дослідження присвячено небагато статей, адже даний процес постійно удосконалюється. Проте є деякі джерела, досліджуючи які дозволило полегшити обробку та аналіз даних. У статті [1] подається обґрунтування методу виявлення пожежі за допомогою обробки зображень. Він ґрунтується на визначенні кольору з інфрачервоних зображень. Методика використовує характеристики яскравості та інтенсивності полум’я на інфрачервоних зображеннях для підвищення надійності та точності виявлення пожежі. Вона дозволяє відрізнити пожежу від фону, за допомогою сегментації на основі гістограми та аналізу оптичного потоку. Дані дослідження будуть корисні у майбутній роботі на даною тему.

Наступна робота [6] дозволяє отримати ґрунтовне розуміння про те як влаштована CNN. Також надаються відповіді на різні запитання про шари мережі та їх принцип роботи між собою, та за яким принципом відбувається зміна розміру вхідних зображень. В даній статті автори проводять якісні дослідження стосовно ефективності часу виконання з моделлю CNN та вказують на усі переваги та недоліки даної моделі.

У статті [10] автори присвятили своє дослідження моделі в CNN та покращення її ефективності на малому наборі даних. Суть запропонованого методу полягає в тому, щоб передавати раніше вивчені параметри мережі до нової моделі для нового набору даних.

У публікації [9] автором порівнюється час виконання та точність результатів різних моделей тренування, в залежності від вибраного тренувального датасету та кількості епох. CNN порівнюється з AlexNet та іншими відомими мережами ResNet та DenseNet, які досягли відмінних результатів у випробуванні. Також автори показали, що дуже висока ефективність класифікації може бути досягнута за допомогою глибоких CNN, навіть при обмежених даних. Для вирішення цієї проблеми, слід збільшити датасет для навчання, використовуючи різні методи збільшення.

Стаття [4] описує роботу CNN. В ній розкрито структуру моделі, принцип роботи кожного шару з візуалізаціями процесів на кожному етапі обробки. Також висвітлені поради про вибір кращої функції активації для досягнення точніших результатів в залежності поставленої задачі. В підсумку публікація дозволяє глибше усвідомити всі процеси на простих прикладах.

В публікації [3] описуються всі можливі способи для виявлення та запобігання різних ситуацій таких як пожежа, затоплення та інші ситуації в яких може бути нанесено матеріальну шкоду майну. На основі даної статті ми дізнались які покращення можливо зробити.

В роботі автора [7] описано про всі аспекти що використовуються у напрямку “Smart house security”. Також автор наголошує на більшій концентрації розробки програмного забезпечення на основі AI, яке дає можливість розширити спектр задач для вирішення питань безпеки та здоров’я жителів такого будинку.

#### Формулювання мети дослідження

**Метою дослідження** є розробка моделі, яка складається з CNN та OpenCV для класифікації зображень за напрямком “Smart house security”.

До основних завдань дослідження, які повністю розкривають суть мети дослідження є:

- Огляд прикладів задач класифікації з використанням CNN та OpenCV.
- Порівняння швидкості результату виконання та їхніх точностей в залежності від параметрів та способів обробки вхідних даних.
- Паралелізація класифікації зображень за допомогою багатопоточності.

Викладання основного матеріалу дослідження

В межах даної роботи розглядається задача класифікації за допомогою нейронної моделі CNN та OpenCV. За допомогою OpenCV можна обробляти відеопотоки у реальному часі, а вже за допомогою тренуваної моделі класифікувати розкадровані зображення. Таким чином ціллю дослідження є моделювання архітектури моделі таким чином, щоб мінімізувати час класифікації та збільшити її точність. Також покращити швидкість класифікації можна за допомогою багатопоточності.

Для повного аналізу дослідження слід розглянути принцип роботи CNN [5]. Це клас глибоких нейронних згорткових мереж, які можуть розпізнавати і класифікувати певні ознаки зображень і широко використовуються для аналізу візуальних образів. Їх застосовують для різних задач класифікації, до прикладу: розпізнавання зображень і відео, класифікації зображень, аналізу медичних зображень обробки природної мови. Через ефективність у зменшенні кількості параметрів без втрати якості моделей. Зображення мають високу розмірність (оскільки кожен піксель розглядається як характеристика) [6].

На основі досліджень інформації про дану модель було прийнято рішення використовувати саме її для виконання поставлених задач.

1. Конволюційний шар (згортка) [9] використовується для отримання різних характеристик з вхідних зображень. У цьому шарі виконується операція згортки між вхідним зображенням та фільтром певного розміру. На виході отримуємо карту ознак, яка надає інформацію про зображення, а саме про кути та краї. Наступним кроком ця карта ознак передається на інші шари для вивчення нових ознак [10]. Математична модель карти ознак  $n$  згорткового шару  $l$  наведена у Формулі 1:

$$X_j^l = f \left[ \sum_{i \in V_j^l} x_i^{l-1} * k_{ij}^l + b_j^l \right], \quad (1)$$

де  $L$  – номер згорткового шару;

$b_j^l$  – значення що приєднується до карти ознак на  $l$  шарі;

$k_{i,j}^l$  – згортка що застосовується до карти ознак  $j$ ;

$V_j^l$  – список усіх рівнів слою  $(l-1)$ , що з'єднуються з картою ознак  $j$  шару  $l$ ;

2. Pooling Layer [10] – на цьому шарі зменшується розмір згорнутої карти ознак для зниження обчислювальних витрат за рахунок зменшення зв'язків між шарами. Залежно від методу існує кілька типів операцій пулінга [2]: Max Pooling із карти ознак може бути застосований для зниження дискретизації згорткових вихідних нашого варіанту розміром  $2*2$  і для кожного такого субрегіону виводить максимум [3].

3. FC Layer – шар, який називається шаром FC, що перетворює матрицю у вектор і подається в повністю пов'язаний шар, як нейронну мережу (рис. 1). Ці шари зазвичай утворюють кілька останніх шарів архітектури CNN [5].

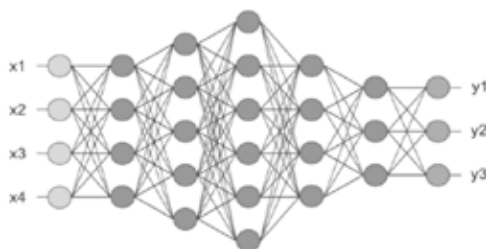


Рис. 1. Схема перетворення матриці ознак на вектор

На наведеній вище схемі (Рис.1) матриця карти ознак буде перетворено на вектор  $(x_1, x_2, x_3, \dots)$ . За допомогою пов'язаних шарів ми об'єднали ці ознаки разом, щоб створити модель (Формула 2) [6].

$$y = g(Wx + b), \quad (2)$$

де  $x$  – вхідний вектор із розмірністю  $[p_1, 1]$ ;

$W$  – матриця ваг з розмірами  $[p_1, n_1]$  де  $p_1$  – кількість нейронів у попередньому шарі, а  $n_1$  – кількість нейронів у поточному шарі;

$b$  – вектор зміщення з розмірністю  $[p_1, 1]$ ;

$g$  – функція активації ReLU .

4. Функції активації являється одним із найважливіших параметрів моделі CNN. Вона додає нелінійність до мережі. Існує кілька функцій активації, що широко використовуються, таких як ReLU, Softmax, tanH і сигмоїдна функція. Кожна із цих функцій має специфічне застосування. Для бінарної класифікації CNN-моделі переважно використовують функції sigmoid та SoftMax, а для багатокласової класифікації зазвичай використовується SoftMax.

ReLU реалізує функцію  $y = \max(x, 0)$ , тому розміри входу та виходу цього шару однакові. Нелінійність необхідна для створення нелінійних меж прийняття рішень. Якби нелінійна функція активації була відсутня, глибокі архітектури CNN перетворилися б на один еквівалентний конволюційний шар, який би працював не так добре. В більшості випадків використовується вона для активації, тому що з нею мережа навчається швидше, у порівнянні з іншими нелінійними функціями, що використовуються в CNN (наприклад, гіперболічний тангенс, абсолютна величина гіперболічного тангенсу та сигмоїд) (рис. 2а) [8].

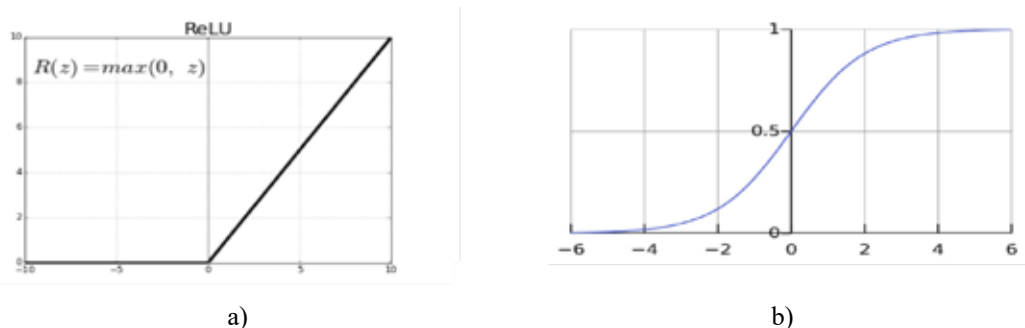


Рис. 2. Графік функцій: а) ReLU; б) SoftMax

Функція SoftMax може використовуватись у класифікаторі лише тоді, коли класи є взаємовиключними. Багатошарові нейронні мережі закінчуються передостаннім шаром, який видає реальні оцінки, які незручно масштабувати і з якими можуть виникнути труднощі у роботі. SoftMax вирішує цю проблему, оскільки він перетворює оцінки в нормалізований розподіл ймовірності, який може бути показаний користувачеві або використаний як вхідні дані для інших систем. З цієї причини зазвичай функція SoftMax додається як останній шар нейронної мережі (рис. 2б).

Переваги CNN:

- Автоматизований вибір відповідних ознак класифікації, що пришвидшує обробку даних.
- Висока точність класифікацій.
- Зменшує кількість параметрів без втрати якості моделей.

Недоліки CNN:

- Великий час тренування на більших датасетах.
- Дуже чутлива до ракурсу та освітлення, тобто те саме фото під іншим ракурсом та іншим освітленням може мати на виході інший результат класифікації (рішенням цієї проблеми є збільшення датасету).

- При малому датасеті через велику кількість параметрів може відбутись перенавчання моделі.
- Можливість фіктивного спрацювання, тобто коли кольорова палітра дуже схожа до тої на якій тренувалась модель. Наприклад: фото з вогнем і фото з заходом сонця кольору схожого до червоного, може на виході отримати однаковий результат класифікації такий як “Fire”.

Розглянемо детально етапи виконання поставленої мети. На рис. 3 представлено Опис моделі у вигляді UML діаграми.

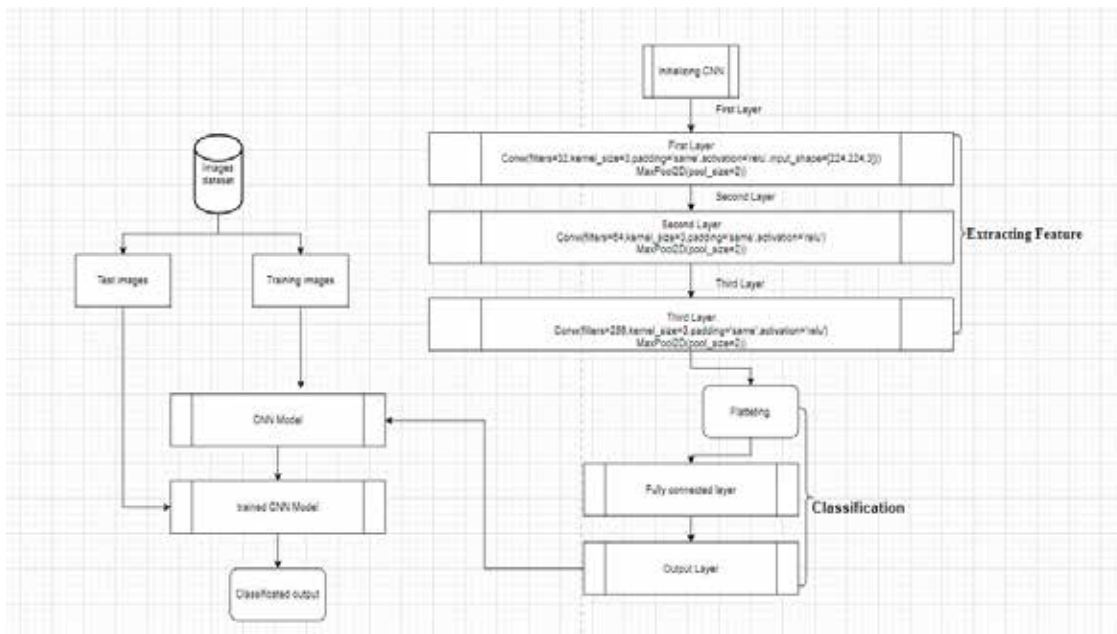


Рис. 3. UML діаграма алгоритму класифікації

Наступним етапом є представлення алгоритму виконання задачі, основними кроками якого є:

1. Побудова CNN.
2. У першому шарі побудованої моделі передаються на вхід зображення. Далі відбувається операція «згортка» з нашим фільтром. На виході з CL отримуємо карту ознак що має такий самий розмір як і фільтр (рис. 4).

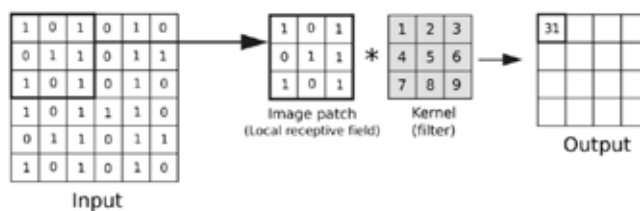


Рис. 4. Згортковий шар CNN

3. Наступним кроком – передаємо нашу карту ознак у PL для її ущільнення та утворення нової карти з сильнішими ознаками. PL застосовується для уникнення перенавчання. В 2 і 3 шар ми вже передаємо не зображення, а карти ознак і виконуємо всі ті самі кроки як і у першому шарі нашої моделі, але вже з іншою кількістю фільтрів (рис. 5a).

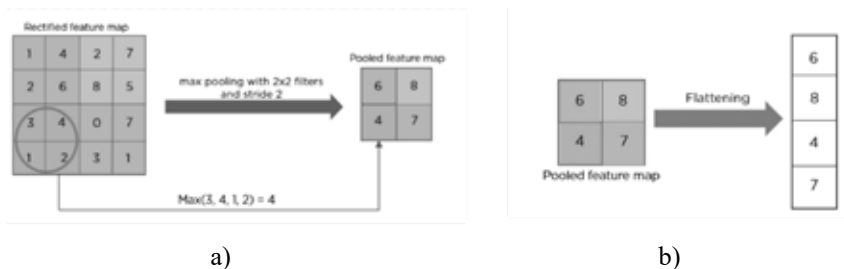


Рис. 5. a) Max Pooling; b) Flattening

4. За допомогою Flatten перетворюємо нашу карту ознак в одновимірний вектор для майбутньої побудови нейронної мережі (рис. 5b).

5. Наступним кроком передаємо наш результуючий вектор у FCL, де наші вектори з'єднуються з кількома повністю пов'язаними шарами, які є такими ж, як штучні нейронні мережі, і виконують ті ж математичні операції.

6. Після проходження через FC останній шар використовується функція активації SoftMax (замість ReLU) для отримання ймовірності того, що вхідні дані знаходяться в певному класі (класифікація) (рис. 6).

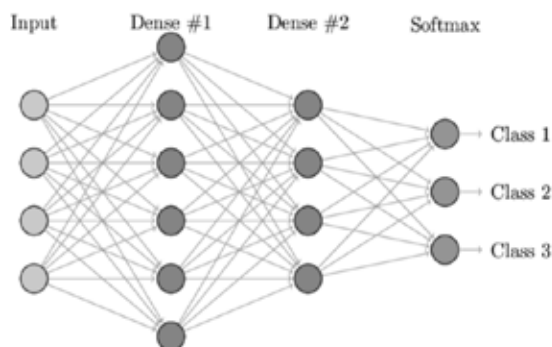


Рис. 6. FCL+OL

7. Після побудови нашої моделі навчаємо класифікатор на тренувальному датасеті і отримуємо претреновану модель, яка і буде класифікувати тестові зображення.

У ході проведення експериментів використовується модель класифікації CNN. Для порівняння результатів підібрані різні параметри. Такі як оптимайзери та кількості епох, щоб отримати інформацію про найоптимальніші варіанти.

Параметри для експериментів:

- Оптимайзери (“Adam”, “RMSprop”, “SGD”).
- Розміри зображень (56x56, 122x122, 224x224).
- BatchSize (4, 16, 64).

Для перевірки роботи класифікації за допомогою даної моделі буде використано датасет, що складається з зображень, на яких є полум'я та просто нейтральні зображення кімнат дому. Загалом було використано в сумі 2704 зображення для тренування моделі. Для класифікації зображень слід помістити їх у різні папки для умовного поділу на класи (рис. 7). Ці дані було розподілені на тренувальні та тестові у відношенні 80% тренувальні та 20% тестові.

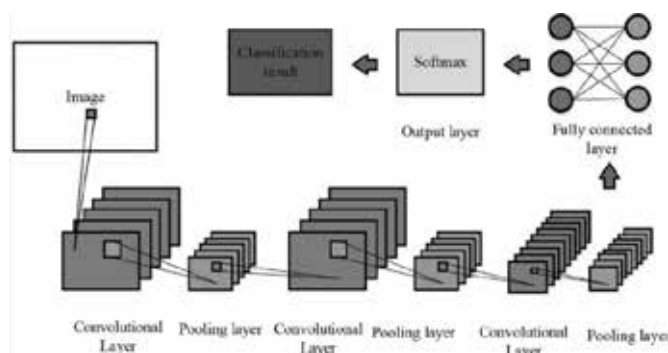


Рис. 7. Архітектура побудованої CNN моделі

Наша CNN модель складається з 3 шарів, доданої функції maxPool до кожного шару для запобігання перетренування та оптимізатора adam. Наступним ключовим параметром є batchsize [8], що впливає на кількість даних опрацьованих за один раз. Чим він більший тим точність більша, але все залежить від характеристик процесора на якому іде виконання. Оскільки це може впливати на час виконання навчання нашої моделі. Рекомендується вибрати величину batchsize=2^n. Для того, щоб почати класифікацію нам необхідно зробити зображення всі одного розміру в нашому дослідженні це 244\*244. Умова передавання зображення в нашу нейронну мережу, це є перетворення зображення в матрицю пікселів. А далі вже відбуватиметься процедура визначення класифікаційних ознак зображення та класифікація як це описано в алгоритмі роботи нашої задачі. На рис. 8a, 8b представлені тренувальні датасети для класу з пожежею та з кімнатами.



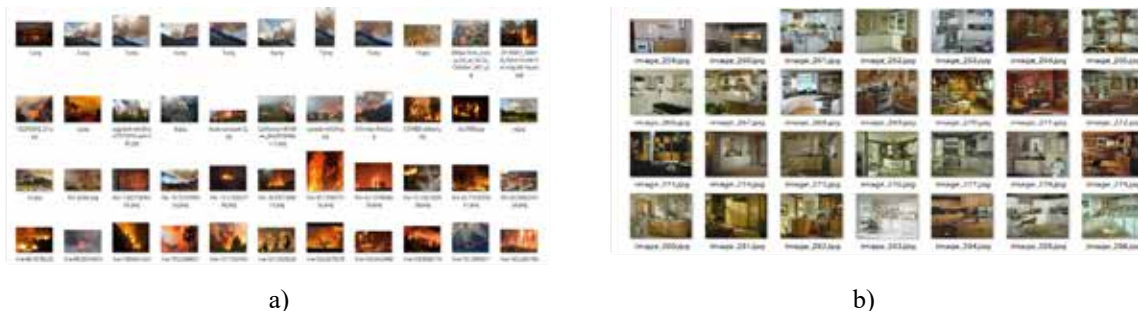


Рис. 8. Тренувальні датасети: а) для класу “Fire”; б) для класу “Netural”

Після закінчення тренування нашої моделі з оптимізатором Adam можна побачити що було досягнуто досить високої точності. Загальний час тренування на одну епоху 95 секунд.

Для кращої візуалізації побудуємо графіки точностей та втрат, що наочно показують ефективність тренування (рис. 9а, 9б).

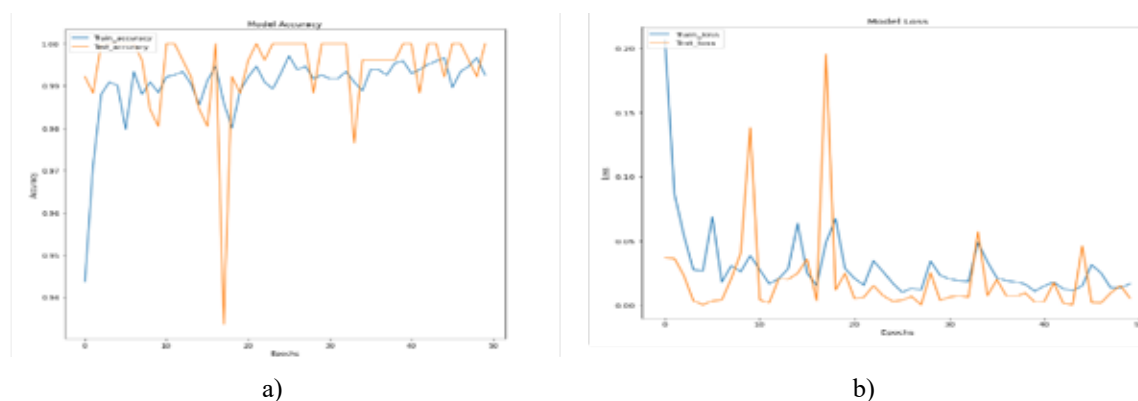


Рис. 9. Графік залежності: а) точності при 50 тренувальних епохах; б) втрат від кількості епох

Як можемо побачити з Рис. 9а, наша модель досягла чудового результату. Точність класифікації в межах [0.92;0.99]. Якщо розглядати цей графік з точки зору більшого масштабу то лінії Train\_ accuracy і Test\_ accuracy будуть майже накладатись, а це означає що наших тренувальних даних виявилось достатньо для ефективного навчання.

На Рис. 9б представлено графік, який також підтверджує що наша модель досягла чудового результату, оскільки функція Loss показує на скільки було неправильне передбачення (чим менше значення тим краще передбачення). На 9 та 19 епосі Рис. 9б відбувся стрибок даного значення. Оскільки CNN модель чутлива до ракурсів та освітлення. Тому можна зробити висновок, що це і є основною причиною стрибку значення test\_loss.

Наступним кроком – завантажуюмо зображення для класифікації (рис. 10а, 10б):



Рис. 10. Тестоване зображення: а) з набору “Fire”; б) з набору “Netural”

Після закінчення тренування нашої моделі з оптимізатором RMSprop можна побачити, що було досягнуто хорошої точності, але менш ефективно порівняно з попереднім оптимайзером. Загальний час тренування на одну епоху 102 секунди, що теж більше ніж у попередньому тренуванні. На рис. 11а, 11б зображені графіки точностей та втрат.

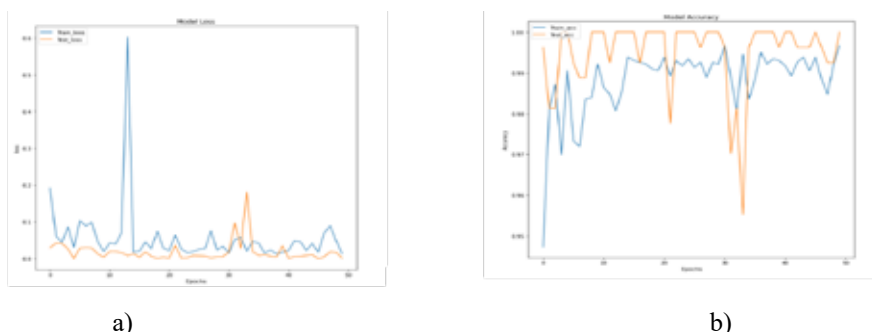


Рис. 11. Графік залежності: а) точності при 50 епохах; б) втрат при 50 епохах

Рис. 11а також підтверджує що наша модель досягла чудового результату, оскільки функція Loss показує на скільки було погане передбачення і чим менше значення, то тим краще передбачення було зроблено. Також можна помітити що в більшості випадків значення Loos прямує до 0, що є ще однією ознакою хорошої класифікації з високою точністю.

Як можемо побачити На рис. 11б, наша модель досягла чудового результату – точність класифікації в межах [0.95;0.99].

Прийдемо, до іншої нашої поставленої задачі, а саме розпаралелення класифікації зображень. Оскільки вхідні дані це відео, що будуть розбиватись покадрово і передаватись з оптимальним часовим інтервалом. Було вирішено розпаралелювати саме класифікацію, а не тренування, оскільки модель тренується разово на великій кількості даних після чого претренована модель класифікуватиме вхідний датасет зображень.

Було проведено експерименти з визначенням часу виконання програми. Приведені таблиці результатів експериментів, на яких позначено найоптимальніший розподіл роботи програми по потоках, графіки часу виконання, та діаграми залежності ефективності від кількості вхідних зображень.

Формули 3-4, за якими було вираховано прискорення та паралельну ефективність:

Отож, оцінка паралельного прискорення розраховувалась за Формулою 3:

$$S_n = \frac{T_1}{T_n}, \tag{3}$$

де  $T_1$  – час виконання послідовного алгоритму;

$T_n$  – час виконання паралельного алгоритму;

$n$  – кількість процесів (потоків).

Оцінка ефективності буде обчислюватись за формулою 4:

$$P_n = \frac{S_n}{n}, \tag{4}$$

Таблиця 1

**Розрахунок прискорення**

Кількість картинок	1 потік	Паралельний алгоритм					
		2 потоки		4 потоки		8 потоків	
		T	S	T	S	T	S
100	8.04475	4.09019	1.96683	2.434093	3.305029	2.01975	3.98303
200	15.8410	8.1130	1.95252	4.58647	3.45384	3.54599	4.46728
400	31.64040	16.55203	1.91157	9.36195	3.37967	7.54332	4.19449
800	64.27154	34.0405	1.88808	20.61760	3.11731	16.8859	3.80621

Таблиця 2

**Розрахунок ефективності**

Кількість картинок	1 ядро	2-ядерний	4-ядерний
100	0,983415	0,82625725	0,49787875
200	0,97626	0,86346	0,55841
400	0,955785	0,8449175	0,52431125
800	0,94404	0,7793275	0,47577625



На рис. 12а та 12б зображена ефективність та прискорення паралельного обчислення.

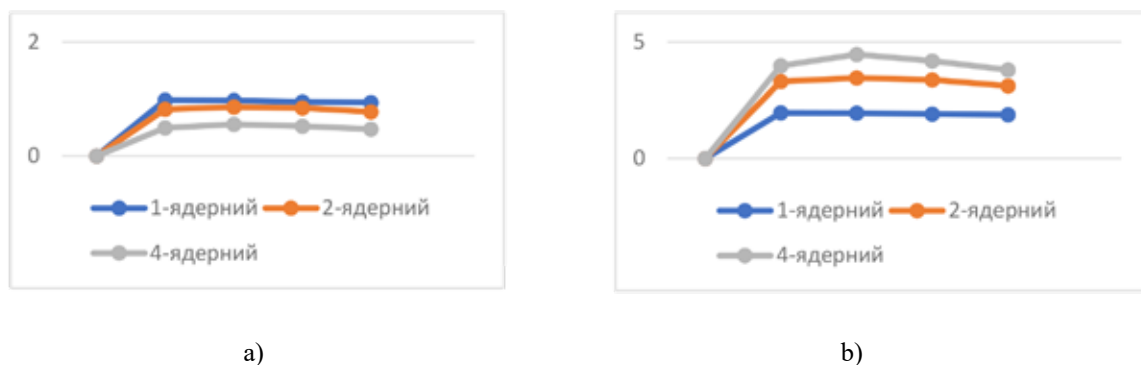


Рис. 12. а) ефективність паралельного обчислення; б) прискорення паралельного обчислення

### Висновки

Було проаналізовано велику кількість літературних джерел щодо доцільності вибору саме моделі CNN. В них описувались всі сильні сторони даної структури нейронної мережі. Під час проведення експериментів відбувались тренування з різними розмірами епох та оптимайзерами. Для покращення швидкості було підібрано 2 найбільш результативні, а саме “Adam” та “RMSProp”, тому що при виборі інших – точність різко падала до значень менше 40%. Для визначення ефективності одного над іншим, ми виконували тренування та класифікацію зображень з однаковими параметрами.

CNN з “Adam” оптимайзером по часу тренування, що припадає на одну епоху, виконував тренування швидше на 8с. Тому при великій кількості епох він дає вигреш у часі. Точність класифікації моделі на основі даного оптимайзера становить 96%.

CNN з “RMSProp” оптимайзером по часу тренування, що припадає на одну епоху виконував тренування повільніше, але точність класифікації моделі на основі даного оптимайзера становить 97%.

При збільшенні кількості епох ми могли помітити позитивну динаміку по точності і негативну динаміку по втратах на обох моделях при збільшенні кількості епох. Це підтверджує результативність CNN для свого прямого призначення, а саме класифікації зображень.

Також у роботі були проведені експерименти над розміром зображення. В результаті даного експерименту час тренування зменшився ще на 8 с на одній епосі, на основі того, що розмір вхідної матриці пікселів зображення став меншим.

Отож, можна зробити висновок, що при оптимальному підборі розміру зображення можна також збільшити швидкість обробки. Після підбору усіх оптимальних вхідних параметрів нашої нейронної мережі ми приступили до розпаралелення класифікації, нам вдалось суттєво прискорити її класифікацію, що представлено в таблицях 1–2.

Проте нами пропонуються способи оптимізації моделі:

1. Використання в ролі оптимізатора “adam”.
2.  $\text{BatchSize}=2^n$ .
3. Підбір оптимального розміру вхідного зображення. У представленому експерименті це дало вигреш у 6с на одну епоху, що в результаті оптимізує швидкість на 3 хв на кожних 50 епохах без втрати точності.
4. Для збільшення точності необхідно використовувати велику кількість даних. Це необхідно для будь яких мереж глибокого навчання. Перевага CNN в тому, що вона чутлива до ракурсів освітлення та багато інших параметрів властивих для зображень. Тому за допомогою зближення, повороту, та інших дій над фото ми можемо штучно збільшувати наш датасет.
5. Для збільшення швидкості можна тренувати модель за допомогою GPU. В проведеному експерименті це дозволило зменшити час з 108 с до 28 с, в порівнянні з тренуванням за допомогою CPU.
6. Використання більшої кількості даних при розпаралеленні. Тому що при малих даних не завжди раціонально запускати багатопотокове виконання програми.

### Список використаної літератури

1. Celia T., Özkaraman H. Fire And Smoke Detection Without Sensors: Image Processing Based Approach. Demirel 15th European SignalProcessing Conference (EUSIPCO 2007), Poznan, Poland, copyright by EURASIP. Electrical and Electronic Engineering, Eastern MediterraneanUniversity, TURKEY, September. 2007. pp. 1-5.
2. Divya R., Mageshwari D. A Novel Fire Detection System using Image Processing and Artificial Intelligence Techniques. Proceedings published by International Journal of Computer Applications (IJCA), Chennai. 2012.
3. Patel P., Tiwari S. Flame Detection using Image Processing Techniques, Journal of Computer Applications. Vol. 58. №18. 2012. pp. 1-4.

4. Xiong Z., Caballero R., Wang H., Finn A. M., Lelic M. A., and Peng P.Y. Video-based Smoke Detection: Possibilities, Techniques, and Challenges. United Technologies Research Centre, East Hartford.
5. Stauffer C. and Grimson W.E.L. Adaptive Background Mixture Models for Real- Time Tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition. 1999.
6. Shams R., Hossain S., Priyom S., Fatema N. Automated Fire Extinguishing System With Gsm Alarm. Thesis in the Department of Electrical and Electronic Engineering Of BRAC University.
7. Töreyn B.U., Dedeoğlu Y., Çetin A.E. Flame detection in video using hidden Markov models. Proc. IEEE Internet Conf. On Image Processing. 2005. pp. 1230-1233.
8. Arrue B. C., Ollero A., and Martinez de Dios J. R. An intelligent system for false alarm reduction in infrared forest-fire detection. IEEE Intelligent Systems. Vol. 15. № 3. 2000. pp. 64–73.
9. Śrutek M., Andrysiak T. Flame Detection Based On Infrared Images. University of Technology and Life Sciences, Bydgoszcz, Poland. 2013. pp. 1-10.
10. Yuan C. Automatic Fire Detection Using Computer Vision Techniques for UAV-based Forest Fire Surveillance. A Thesis in the Department of Mechanical, Industrial and Aerospace Engineering. 2017. pp. 72-87.

#### References

1. Celia T., Özkaraman H. Fire And Smoke Detection Without Sensors: Image Processing Based Approach. Demirel 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, copyright by EURASIP. Electrical and Electronic Engineering, Eastern Mediterranean University, TURKEY, September. 2007. pp. 1-5.
2. Divya R., Mageshwari D. A Novel Fire Detection System using Image Processing and Artificial Intelligence Techniques. Proceedings published by International Journal of Computer Applications (IJCA), Chennai. 2012.
3. Patel P., Tiwari S. Flame Detection using Image Processing Techniques, Journal of Computer Applications. Vol. 58. №18. 2012. pp. 1-4.
4. Xiong Z., Caballero R., Wang H., Finn A. M., Lelic M. A., and Peng P.Y. Video-based Smoke Detection: Possibilities, Techniques, and Challenges. United Technologies Research Centre, East Hartford.
5. Stauffer C. and Grimson W.E.L. Adaptive Background Mixture Models for Real- Time Tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition. 1999.
6. Shams R., Hossain S., Priyom S., Fatema N. Automated Fire Extinguishing System With Gsm Alarm. Thesis in the Department of Electrical and Electronic Engineering Of BRAC University.
7. Töreyn B.U., Dedeoğlu Y., Çetin A.E. Flame detection in video using hidden Markov models. Proc. IEEE Internet Conf. On Image Processing. 2005. pp. 1230-1233.
8. Arrue B. C., Ollero A., and Martinez de Dios J. R. An intelligent system for false alarm reduction in infrared forest-fire detection. IEEE Intelligent Systems. Vol. 15. №3. 2000. pp. 64-73.
9. Śrutek M., Andrysiak T. Flame Detection Based On Infrared Images. University of Technology and Life Sciences, Bydgoszcz, Poland. 2013. pp. 1-10.
10. Yuan C. Automatic Fire Detection Using Computer Vision Techniques for UAV-based Forest Fire Surveillance. A Thesis in the Department of Mechanical, Industrial and Aerospace Engineering. 2017. pp. 72-87.