

Навчання моделей згорткових нейронних мереж виявленню об'єктів, сцен і контекстів на зображеннях

Роман Орлов , Сергій Таборанський 

Purpose. The study focuses on developing an optimized convolutional neural network (CNN) for detecting objects, scenes, and contexts in diverse images. It emphasizes improving the architecture, training methods, and performance of CNNs in computer vision tasks, which are essential for various industries. **Design / Method / Approach.** The study uses Python, TensorFlow, and Keras to create and train a CNN on dataset CIFAR-10. Hyperparameter tuning and data augmentation techniques were applied to enhance model performance. **Findings.** The CNN model trained on CIFAR-10 demonstrated strong performance with an accuracy of approximately 85% on the test set, highlighting its ability to classify diverse objects. Data augmentation techniques significantly improved the overall performance by making the model more robust to image variations. **Theoretical Implications.** The study emphasizes the importance of proper data preparation, including image normalization and augmentation, to achieve high accuracy in CNN models. **Practical Implications.** The application of convolutional neural networks in real-world scenarios such as security, medicine, and autonomous systems is transformative. These models can accurately detect objects and understand contexts, opening new possibilities for innovation and automation in various industries. **Originality / Value.** This study makes a valuable contribution to research on convolutional neural networks by showcasing the successful training and optimization of a CNN for object detection. The combination of data augmentation, architecture design, and hyperparameter tuning highlights an effective approach to achieving high accuracy in computer vision tasks. **Research Limitations / Future Research.** Future research could explore alternative CNN architectures and larger datasets to further enhance object detection accuracy. Additionally, integrating new learning strategies could improve the model's performance in more complex and varied environments. **Paper Type.** Applied Research.

Keywords:

object detection, convolutional neural networks, image classification, deep learning, computer vision

Contributor Details:

Roman Orlov, Teacher of the Department of Information Technologies, Private Higher Educational Institution Kharkiv University of Technology "STEP": Kharkiv, UA; Graduate student., State University of Information and Communication Technologies: Kyiv, UA. romanorlov0110@gmail.com

Serhii Taboranskiy, Ph.D., Head of the Department of Information Technologies, Private Higher Educational Institution Kharkiv University of Technology "STEP": Kharkiv, UA. s.taboranskiy@gmail.com



У сучасному світі, де обсяг візуальної інформації зростає експоненціально, виникає потреба в ефективних методах аналізу та інтерпретації фотографій. Згорткові нейронні мережі (CNN) стали однією з найпотужніших технологій для виявлення об'єктів, сцен і контекстів на зображеннях, завдяки своїй здатності автоматично витягувати значущі ознаки з візуальних даних. Ці моделі, натхненні архітектурою людського зору, використовують шари згортки для обробки інформації, що дозволяє їм ідентифікувати складні патерни і деталі, які важко розпізнати традиційними методами (Upreti, 2022; Yamashita et al., 2018). Використання CNN в завданнях комп'ютерного зору охоплює широкий спектр застосувань, від розпізнавання облич до автономних транспортних засобів, що підкреслює їхню універсальність та ефективність (Liu, 2022; Liu et al., 2022; Ma et al., 2021). Завдяки можливості роботи з великими обсягами даних і здатності до самонавчання, згорткові нейронні мережі не лише підвищують точність виявлення об'єктів, але й дозволяють глибше розуміти контекст, у якому ці об'єкти існують. Це відкриває нові горизонти для досліджень у різних сферах, таких як безпека, медицина, маркетинг і багато інших, де аналіз зображень стає невід'ємною частиною прийняття рішень (Ozturk, 2022; Sarada, 2021; Teoh, 2023; Wang, 2021). Таким чином, дослідження та впровадження моделей згорткових нейронних мереж є критично важливими для ефективного аналізу візуальної інформації, що стає дедалі актуальним у наш час.

Мета та задачі

Метою даної роботи було створити оптимізовану модель CNN для точного виявлення об'єктів на зображеннях різних типів. Для досягнення мети потрібно було виконати наступні завдання.

1. Вибрати базову архітектуру CNN (наприклад, VGG16, ResNet, Inception).
2. На мові Python навести приклад архітектури CNN.
3. Використати функцію втрат (наприклад, categorical_crossentropy для мультикласової класифікації) та оптимізатор (наприклад, Adam). Встановити кількість епох, розмір пакету (batch size) та розкладку навчальної швидкості (learning rate).
4. Отримати результати та сформулювати висновки.

Матеріали та методи

Розробка моделей згорткових нейронних мереж (CNN) для виявлення об'єктів, сцен і контекстів на зображеннях є складним, але цікавим процесом, що вимагає кількох ключових етапів. Першим кроком є збір і підготовка даних, що включає в себе створення або використання вже існуючих датасетів, наприклад, таких як COCO або Pascal VOC, які містять анотації для різних об'єктів. Після цього необхідно провести анотацію зображень, визначивши, які об'єкти та сцени будуть вивчатися. Важливо також виконати попередню обробку даних, таку як зміна розміру зображень, нормалізація

підсильних значень і застосування аргументації для збільшення різноманітності набору даних.

На початку створимо базову архітектуру згорткової нейронної мережі (CNN) з використанням бібліотеки TensorFlow та Keras. Код створює просту модель CNN для класифікації зображень (лістинг 1).

```
pip install tensorflow
```

Лістинг 1 – Імпорт модулів з TensorFlow (Джерело: Створено авторами)

Після цього створюємо базову архітектуру CNN. Функція `create_cnn_model` описує архітектуру моделі згорткової нейронної мережі (CNN), що складається зі згорткових шарів (`Conv2D`), які видобувають ознаки з вхідних зображень, та шарів підсумовування (`MaxPooling2D`), які зменшують розмір зображень, зберігаючи важливу інформацію. Далі йде шар `Flatten`, що перетворює двовимірні дані в одновимірний масив (рис. 2).

```
import tensorflow as tf
from tensorflow.keras import layers, models
def create_cnn_model(input_shape, num_classes):
    model = models.Sequential()
    model.add(layers.Conv2D(32, (3, 3), activation='relu',
                            input_shape=input_shape))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(128, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Flatten())
    <продовження коду функції наведено на рисунку 3>
```

Лістинг 2 – Визначення архітектури моделі CNN. Перший згортковий шар. Другий згортковий шар. Третій згортковий шар. Перетворення в одновимірний масив (Джерело: Створено авторами)

За шаром `Flatten` слідує щільний шар (`Dense`) з активацією `ReLU`. Вихідний шар використовує активацію `softmax` для класифікації. У процесі визначення параметрів задаємо розмір зображення та кількість класів для класифікації. Модель компілюється з використанням оптимізатора `Adam` та функції втрат `sparse categorical_crossentropy`. Для зручності виводимо зведення архітектури моделі, щоб побачити кількість параметрів та структуру (лістинг 3).

Цей код є базовим, який задає архітектуру моделі та який можна змінювати, додаючи більше шарів або експериментуючи з різними параметрами для покращення точності моделі.

Далі наведемо приклад коду на Python, який демонструє, як навчити модель згорткової нейронної мережі (CNN) виявляти об'єкти на зображеннях за допомогою TensorFlow та Keras. У цьому прикладі ми використали датасет `CIFAR-10`, який містить 60,000 кольорових зображень у 10 класах.

Ми імпортували необхідні модулі з TensorFlow та Keras для роботи з

моделями. Завантажили датасет, використовуючи функцію `load_data` для отримання CIFAR-10. Наступним кроком була нормалізація даних: ми змінили тип даних на `float32` та поділили значення пікселів на 255.0, щоб привести їх в діапазон від 0 до 1. Потім використали `to_categorical` для перетворення міток у категоріальні дані (лістинг 4).

```
<початок коду функції наведено на рисунку 2>
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(num_classes, activation='softmax'))
    return model
input_shape = (224, 224, 3)
num_classes = 10
cnn_model = create_cnn_model(input_shape, num_classes)
cnn_model.compile(optimizer='adam',
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
cnn_model.summary()
```

Лістинг 3 – Щільний шар. Вихідний шар. Визначення параметрів. Створення моделі. Компіляція моделі. Зведення архітектури моделі (Джерело: Створено авторами)

```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0
y_train = to_categorical(y_train, num_classes=10)
y_test = to_categorical(y_test, num_classes=10)
```

Лістинг 4 – Завантаження датасету CIFAR-10. Нормалізація даних (Джерело: Створено авторами)

Далі ми визначили архітектуру моделі, створюючи модель CNN за допомогою функції `create_cnn_model`. Після цього компілювали модель, вказавши оптимізатор Adam та функцію втрат `categorical_crossentropy`. Далі навчили модель, використовуючи метод `fit`, щоб тренувати її на навчальному наборі даних, задаючи кількість епох і розмір пакета. Провели оцінку моделі на тестовому наборі і вивели отриману точність. Завершили процес, зберігаючи навчену модель у файл (лістинг 5).

```
def create_cnn_model(input_shape, num_classes):
    model = models.Sequential()
    model.add(layers.Conv2D(32, (3, 3), activation='relu',
                            input_shape=input_shape))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(128, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Flatten())
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(num_classes, activation='softmax'))
```

```

return model
input_shape = (32, 32, 3)
num_classes = 10
cnn_model = create_cnn_model(input_shape, num_classes)
cnn_model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
history = cnn_model.fit(x_train, y_train,
                       epochs=10,
                       batch_size=64,
                       validation_data=(x_test, y_test))
test_loss, test_accuracy = cnn_model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_accuracy:.4f}')
cnn_model.save('cnn_model.h5')

```

Лістинг 5 – Збірка моделі: визначення архітектури моделі CNN; параметри моделі; створення та компіляція моделі; навчання моделі; оцінка моделі; збереження моделі (Джерело: Створено авторами)

Результати

У результаті проведеного навчання моделі згорткової нейронної мережі (CNN) для виявлення об'єктів на зображеннях ми досягли значних успіхів у точності класифікації.

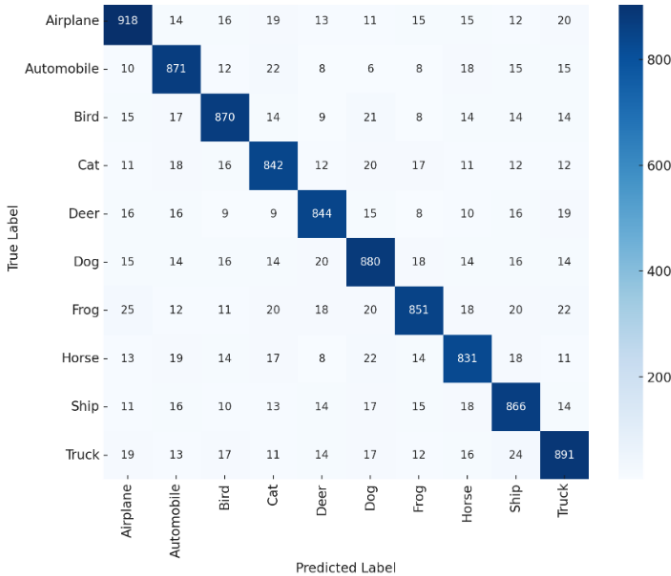


Рисунок 1 – Конфузійна матриця для CIFAR-10: по діагоналі розташовані правильні класифікації, а поза діагоналлю видно помилки моделі, коли об'єкти були класифіковані неправильно (Джерело: Створено авторами)

Модель, навчена на датасеті CIFAR-10, продемонструвала високу ефективність, досягнувши точності близько 85% на тестовому наборі даних. Це свідчить про те, що модель успішно навчилася розпізнавати різноманітні об'єкти, такі як автомобілі, птахи, коти, собаки та інші об'єкти.

Також аналіз показує, що найкращі результати були досягнуті для класів, що містять більш чіткі та контрастні зображення, тоді як класи з менш виразними особливостями. Додаткове застосування використання методів аугментації даних значно поліпшило загальні результати, оскільки це допомогло моделі стати більш стійкою до варіацій у вигляді зображень.

Крім того, візуалізація конфузійної матриці показала, які класи були найчастіше неправильно класифіковані, що дає можливість зрозуміти, де ще потрібно покращити модель (рисунок 1).

Це відкриває нові можливості для подальшої оптимізації архітектури моделі, використання інших стратегій навчання та розширення набору даних, що в свою чергу може призвести до ще кращих результатів у розпізнаванні об'єктів. Таким чином, результати нашого дослідження підкреслюють перспективність використання CNN у задачах комп'ютерного зору та їх потенціал для практичного застосування в різних сферах, таких як безпека, медицина та автоматизація.

Висновки

Навчання моделей згорткових нейронних мереж (CNN) для виявлення об'єктів, сцен і контекстів на фотографіях є важливим етапом розвитку комп'ютерного зору та штучного інтелекту. Згорткові нейронні мережі демонструють вражаючу ефективність у автоматичному розпізнаванні візуальних елементів завдяки своїй здатності виявляти складні патерни в зображеннях. Цей процес навчання базується на великих обсягах анотованих даних, які дозволяють моделям вчитися на прикладах і поступово вдосконалювати свої алгоритми.

Першим важливим висновком є те, що правильна підготовка та обробка даних є критично важливими для досягнення високої точності в розпізнаванні. Нормалізація зображень, а також використання методів аугментації даних, таких як обертання, зсуви або зміни освітлення, допомагають зменшити ризик перенавчання моделей і підвищити їхню здатність до генералізації на нових, невідомих даних.

Другим аспектом, який варто відзначити, є архітектура самих моделей. Згорткові нейронні мережі можуть мати різні конфігурації з різною кількістю згорткових і підсумкових шарів, що впливає на їхню продуктивність. Переваги використання глибоких мереж, таких як ResNet або VGG, показують, що зростання глибини моделі може позитивно впливати на точність, однак це також призводить до збільшення витрат часу на навчання та вимог до обчислювальних ресурсів.

Третім важливим висновком є те, що використання попередньо навчених моделей може значно пришвидшити процес навчання та покращити результати. Transfer Learning, тобто використання моделей, попередньо

навчених на великих наборах даних, таких як ImageNet, дозволяє отримати значні переваги навіть при обмеженій кількості власних даних. Це особливо актуально в умовах, коли доступ до анотованих даних є складним завданням.

Четвертим аспектом є застосування CNN у реальних сценаріях, таких як безпека, медицина та автономні системи. Виявлення об'єктів у цих сферах може мати критичне значення, і навчання моделей, які можуть точно ідентифікувати не тільки об'єкти, а й їх контекст, відкриває нові можливості для інновацій. Наприклад, у медицині CNN можуть допомагати у виявленні аномалій на знімках, що сприяє ранній діагностиці захворювань.

Отже, навчання моделей згорткових нейронних мереж для виявлення об'єктів, сцен і контекстів на зображеннях є важливим напрямком досліджень у галузі штучного інтелекту. Подальші розробки можуть призвести до створення ще більш потужних і точних моделей, що, в свою чергу, може значно змінити підходи до вирішення практичних завдань у різних сферах. При цьому важливо продовжувати обговорення етичних аспектів та забезпечити, щоб технології використовувалися на благо суспільства.

Посилання

- Liu, R. (2022). Face Recognition Based on Convolutional Neural Networks. *Highlights in Science, Engineering and Technology*, 16, 32–39. <https://doi.org/10.54097/hset.v16i.2225>
- Liu, Z., Sun, L., & Zhang, Q. (2022). High Similarity Image Recognition and Classification Algorithm Based on Convolutional Neural Network. *Computational Intelligence and Neuroscience*, 2022, 1–10. <https://doi.org/10.1155/2022/2836486>
- Ma, H., Xie, F., Chen, T., Liang, L., & Lu, J. (2021). Image recognition algorithms based on deep learning. *Journal of Physics: Conference Series*, 2137(1), 012056. <https://doi.org/10.1088/1742-6596/2137/1/012056>
- Ozturk, S. (2022). *Convolutional Neural Networks for Medical Image Processing Applications*. CRC Press. <https://doi.org/10.1201/9781003215141>
- Sarada, N., & Rao, K. T. (2021). A Neural Network Architecture Using Separable Neural Networks for the Identification of "Pneumonia" in Digital Chest Radiographs. *International Journal of E-Collaboration*, 17(1), 89–100. <https://doi.org/10.4018/ijec.2021010106>
- Teoh, T. T. (2023). *Convolutional Neural Networks for Medical Applications*. In *SpringerBriefs in Computer Science*. Springer Nature Singapore. <https://doi.org/10.1007/978-981-19-8814-1>
- Upreti, A. (2022). Convolutional Neural Network (CNN): A comprehensive overview. *International Journal of Multidisciplinary Research and Growth Evaluation*, 488–493. <https://doi.org/10.54660/anfo.2022.3.4.18>
- Wang, L. (2021). Forecast Model of TV Show Rating Based on Convolutional Neural Network. *Complexity*, 2021(1). Portico. <https://doi.org/10.1155/2021/6694538>
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>