

## ОСОБЛИВОСТІ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ

У статті розглядаються алгоритми машинного розпізнавання і класифікації об'єктів, які використовують для вирішення завдань комп'ютерного зору на мобільних пристроях. Описані їх недоліки і переваги. Встановлено, що багато алгоритмів виявлення є нестійкими до різних ознак або властивостей зображення, на якому відбувається пошук розпізнаваного об'єкта, який може бути повернений, спотворений або навіть деформований. У роботі запропоновано використання згорткової нейронної мережі для усунення всіх зазначених недоліків. Для реалізації поставлених завдань обрані сучасні інформаційні технології *TensorFlow*, *Keras*, *Core ML 2*. Розроблена система використовує середовище глибокого навчання, яка підтримує багато типів машинного навчання, націлених в першу чергу на вирішення задач класифікації і сегментації зображень в розробці мобільного.

**Ключові слова:** алгоритм; розпізнавання і класифікація даних; нейронна мережа; машинне навчання.

**Постановка проблеми.** Рішення задачі розпізнавання та класифікації об'єктів на зображенні є одним з найважливіших застосувань нейронних мереж. Метою роботи є розробка нової моделі, і відповідного їй методу прогнозування, що відноситься до класу нейромережових моделей. Нова модель, і відповідний їй метод, повинні мати високу швидкість обчислення прогнозних значень і порівнянну з іншими моделями точність розпізнавання та класифікації об'єктів на зображенні.

**Аналіз останніх досліджень і публікацій.** На даний момент даний напрямок розвивається тільки в лабораторіях зарубіжних університетів, таких як Корнельський університет (США), Федеральна політехнічна школа Лозанни (Швейцарія), Сінгапурський університет технології та дизайну і т. д. Після вивчення і аналізу літературних джерел, присвячених розпізнаванню та класифікації зображень, можна зробити висновок, що в більшості робіт не враховується метод регуляризації штучних нейронних мереж. Це можна віднести до недоліків відомих методів, оскільки при такому підході ускладнюється модель, відбувається некоректне перенавчання моделі і втрачається точність обчислень. Це призводить до надмірних витрат потенціалу нейромережі.

При застосуванні розпізнавання об'єктів на зображенні згорткові нейронні мережі складаються з декількох шарів невеликих збірників нейронів, які обробляють частини вхідного зображення, звані рецептивними полями у статті *Selective Sensitivity to Direction of Motion in Ganglion Cells of the Rabbit's Retina* [1]. Виходи цих збірників потім укладаються таким чином, щоб вони перекривалися, для отримання кращого подання первинного зображення це повторюється для кожного такого шару. Висновок з перекриттям дозволяє згортковій нейронній мережі може витримувати паралельні переноси вхідного зображення. У статті *Deep Residual Learning for Image Recognition* [2] згорткові мережі можуть включати шари локальної або глобальної підвибірки, які поєднують виходи кластерів нейронів. Вони також складаються з різних комбінацій згорткових і пов'язаних шарів, із застосуванням поточної нелінійності в кінці кожного шару. Для зниження числа вільних параметрів і поліпшення. Узагальнено вводиться операція згортки на малих областях входу. У статті

Режим доступу: <http://sap.pstu.edu>

*Dropout: A Simple Way to Prevent Neural Networks from Overfitting* [3] суть методу полягає в тому, що в процесі навчання вибирається шар, з якого випадковим чином викидається певна кількість нейронів, які виключаються з подальших обчислень. Такий прийом покращує ефективність навчання і якість результату. Більш навчені нейрони отримують в мережі більшу вагу. Одним з головних переваг згорткових мереж є використання спільної ваги в згорткових шарах, які описані в статті *One pixel attack for fooling deep neural networks* [4], що означає, що для кожного пікселя шару використовується один і той самий фільтр, це зменшує обсяг необхідної пам'яті, та покращує продуктивність. Дуже подібну до згорткових нейронних мереж архітектуру також застосовують і деякі нейронні мережі з тимчасовою затримкою, особливо призначені для задач розпізнавання та класифікації об'єктів на зображенні, оскільки укладення виходів нейронів може бути здійснене через певні проміжки часу, зручним для аналізу зображень.

У порівнянні з іншими алгоритмами класифікації зображень, згорткові нейронні мережі використовують відносно мало попередньої обробки. Це означає, що мережа несе відповідальність за навчання фільтрів, в традиційних алгоритмах розроблялися вручну.

**Метою даної роботи** є розробка нової моделі і відповідного їй методу прогнозування, що відноситься до класу нейромережевих моделей. Нова модель і відповідний їй метод повинні мати високу швидкість обчислення прогнозних значень і порівнянну з іншими моделями точність розпізнавання та класифікації об'єктів на зображенні. Нова нейромережева модель повинна розпізнати та класифікувати об'єкти на зображенні по класу, ця нейромережева модель впроваджується в інший проект написаний для мобільного пристрою, який за допомогою цієї нейромережевої моделі проводить аналіз вибраного об'єкта і показує ймовірність у відсотковому співвідношенні до якого класу належить об'єкт на обраному зображенні.

**Виклад основного матеріалу.** Для створення моделі прогнозів, що містить задачу розпізнавання та класифікації об'єктів на зображенні з використанням згорткової нейромережі, була поставлена задача реалізації алгоритмів прогнозування, як самостійної моделі, для подальшого опрацювання.

При виборі мови програмування приймалися в розрахунок наступні вимоги:

- наявність готових бібліотек, що містять реалізацію відомих математичних функцій, а також бібліотек, що реалізують;
- взаємодія програми з різними джерелами даних, без ліцензійних обмежень.

Для вирішення задачі розпізнавання та класифікації об'єктів на зображенні з використанням згорткової нейромережі було розроблено програмне забезпечення на базі спеціалізованого комплексу Visual Studio Code.

Visual Studio Code – це редактор вихідного коду, розроблений Microsoft для Windows, Linux і macOS.

Для розробки моделі розпізнавання та класифікації об'єктів на зображенні використовується система macOS. Visual Studio Code підтримує ряд мов програмування, підсвічування синтаксису, IntelliSense, рефакторинг, налагодження, навігацію по коду, підтримку Git та інші можливості. З 2018 року з'явилися розширення Python для Visual Studio Code з відкритим вихідним кодом. Воно надає розробникам широкі можливості для редагування, налагодження і тестування коду.

Для розробки моделі розпізнавання та класифікації об'єктів на зображенні був використаний мовний пакет Python версії 2.7.15 для Visual Studio Code.

Python – високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду.

Код моделі розпізнавання та класифікації об'єктів на зображенні написана на Python і збережений у файлі `classifire.py`. Після того як навчання моделі пройшло успішно, модель зберігається автоматично в файл `animals.mlmodel`.

Для розробки програми, в якій буде відбуватися завантаження зображення і її обробка працює під керівництвом операційної системи iOS. Для написання проекту використовувалася середовище розробки Xcode. Xcode є єдиною середовищем, для розробки iOS додатків.

Xcode – інтегроване середовище розробки програмного забезпечення під macOS і iOS, розроблена корпорацією Apple. Для розробки проекту в якому буде виконуватися завдання розпізнавання та класифікації об'єктів на зображенні був використаний мовний пакет Swift [3].

Swift – відкрита мультипарадигмальна компільована мова програмування загального призначення. Створено компанією Apple в першу чергу для розробників iOS і macOS. Swift працює з фреймворками Cocoa і Cocoa Touch і сумісний з основною кодовою базою Apple, написаної на Objective-C.

При вирішенні поставлених завдань в роботі використано метод зворотного поширення помилки - метод навчання багатозарового перцептрона. Це ітеративний градієнтний алгоритм, який використовується з метою мінімізації помилки роботи багатозарового перцептрона і отримання бажаного виходу. Основна ідея цього методу полягає в поширенні сигналів помилки від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи. Навчання алгоритмом зворотного поширення помилки передбачає два проходи по всім верствам мережі: прямого і зворотного. При прямому проході вхідний вектор подається на вхідний прошарок нейронної мережі, після чого поширюється по мережі від шару до шару [3]. В результаті генерується набір вихідних сигналів, який і є фактичною реакцією мережі на даний вхідний образ. Під час прямого проходження всі синаптичні ваги мережі фіксовані. Під час зворотного проходження всі синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного, в результаті чого формується сигнал помилки [1]. Цей сигнал згодом поширюється по мережі в напрямку, протилежному напрямку синаптичних зв'язків. Синаптичні ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного [4].

Метою розпізнавання та класифікації об'єктів на зображенні - є ідентифікація до якого класу належить об'єкт на зображенні. Для навчання моделі з розпізнавання та класифікації об'єктів на зображенні були підготовлені і відсортовані зображення у форматі `.jpg`. Зображення наближено до формату 300x200 пікселів. Було підготовлено 34 зображення кішки, 65 зображень дельфіна і 53 зображення страуса. Кожному об'єкту створена своя директорія з зображеннями даного об'єкта.

Після навчання моделі класифікатора, у вікні Turi Create Vizualization, ми можемо спостерігати, що всього зображень оброблено 152 штуки. У кожного зображення є свій порядковий номер, директорія, мініатюра зображення та класифікація об'єкта на зображенні (рис. 1).

Після навчання моделі класифікатора, ми можемо спостерігати в Debug Console (рис. 2) наступні значення:

1. Iteration – ітерація;
2. Elapsed Time – витрачений час;
3. Training accuracy – точність навчання;
4. Validation accuracy – точність перевірки.

	path	image	animals
30	dataset/cat/image_0031.jpg		cat
31	dataset/cat/image_0032.jpg		cat
32	dataset/cat/image_0033.jpg		cat
33	dataset/cat/image_0034.jpg		cat
34	dataset/dolphin/image_0001.jpg		dolphin
35	dataset/dolphin/image_0002.jpg		dolphin
36	dataset/dolphin/image_0003.jpg		dolphin
37	dataset/dolphin/image_0004.jpg		dolphin

Row # 152 rows

Рисунок 1 – Turi Create Vizualization

```

Automatically generating validation set from 5% of the data.
Beginning model training on processed features.
Calibrating solver; this may take some time.
+-----+-----+-----+-----+
| Iteration | Elapsed Time | Training-accuracy | Validation-accuracy |
+-----+-----+-----+-----+
| 1         | 0.756960     | 0.661379          | 0.648148            |
| 2         | 1.179773     | 0.776736          | 0.717593            |
| 3         | 1.392729     | 0.661379          | 0.648148            |
| 4         | 1.734310     | 0.858928          | 0.819444            |
| 5         | 1.960998     | 0.927421          | 0.879630            |
| 10        | 3.009042     | 0.952415          | 0.921296            |
+-----+-----+-----+-----+
Completed (Iteration limit reached).
    
```

Рисунок 2 – Результати навчання

Коли навчання закінчується, представляється відображення підготовки і (іноді) перевірка показників точності, описанні в Debug Console.

Отримано 95 % і 92 % точності навчання і валідації відповідно. Показники можуть варіюватися, оскільки перевірка встановлена випадковим чином, для кожного тренування зображення вибираються випадковим чином, 10 зображень. Немає способу дізнатися, які саме зображення обрані.

Тренування точності (Training accuracy) проста – підготовка включає в себе ймовірність, наскільки важкі показники, щоб дати кожній функції для обчислення відповіді. Оскільки зображення позначені «кішкою» або «дельфіном», алгоритм навчання може перевірити свої відповіді і обчислити, який відсоток він отримав правильно. Потім він подає правильну або неправильну інформацію в наступну ітерацію для уточнення показників.

Перевірка точності (Validation accuracy) те ж саме – перш, ніж почати підготовку, обраного випадковим чином 10 % з даних відбираються для перевірки даних. Об'єкти витягуються, а відповіді обчислюються з тими ж вагами, що й набір даних навчання. Але результати не використовуються безпосередньо для перерахунку ваги. Їх мета полягає в запобіганні моделі перенавчання - зациклення на неважливу функцію, який колір, фон або освітлення. Якщо точність валідації сильно відрізняється від точності тренування, то алгоритм сам вносить корективи. Так, що вибір перевірка зображень впливає, як на точність перевірки, так і на точність навчання. Turi Create дозволяє надати фіксований набір перевірюваних даних, якщо створено набір з аналогічними характеристиками.

Після закінчення навчання моделі класифікатора автоматично відбувається збереження моделі `animals.mlmodel`, який підтримується програмним продуктом Xcode.

Наступним кроком має бути впровадження моделі класифікатора в проект Xcode і порівняння результатів з моделями-аналогами. Було вибрано дві моделі класифікатора, ResNet 50 та Inceptionv3. Обидві моделі позиціонують себе такими, що виявляють домінуючі об'єкти, присутні в зображенні з набору тисячі категорій, таких як дерева, тварини, продукти харчування, транспортні засоби, люди і багато іншого.

Моделі були впроваджені у даний проект Xcode по черзі, в якому було проведено 30 експериментів з зображенням об'єкта дельфін. Результати усіх експериментів представлені на рис. 3.

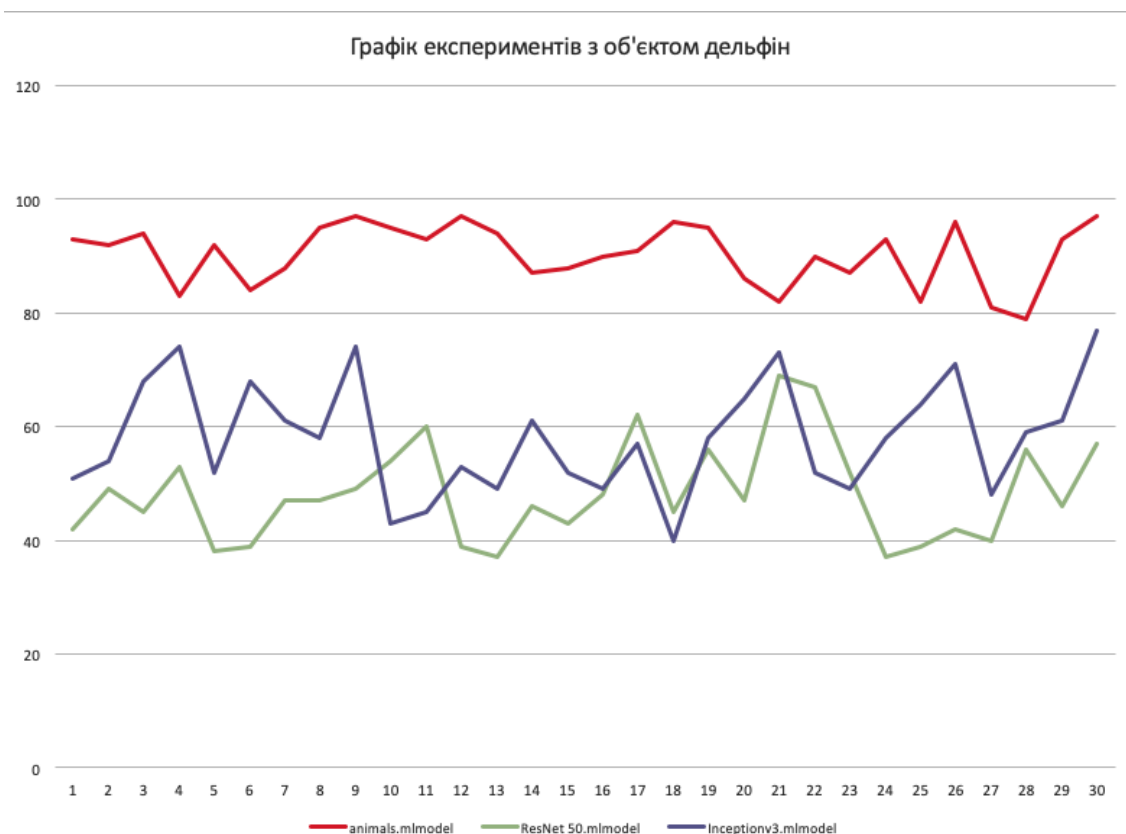


Рисунок 3 – Результат обробки зображення з дельфіном

Після обчислення ймовірності по моделі `animals.mlmodel`, отримуємо результати, що даний об'єкт на фото дельфін з 81 – 97 % ймовірністю, що показує досить високий відсоток схожості.

Після обчислення ймовірності по моделі ResNet 50, отримуємо результати, що даний об'єкт на фото дельфін з 37 – 69 % ймовірністю, що показує досить низький відсоток схожості.

Після обчислення ймовірності по моделі Inceptionv3, отримуємо результати, що даний об'єкт на фото дельфін з 40 – 77 % ймовірністю, що показує досить низький відсоток схожості.

Реалізація запропонованої моделі розпізнавання і класифікації за допомогою TuriCreate показала високу точність обчислення ймовірності та показала порівнянну зі спеціалізованими моделями точність обчислення ймовірності.

Виконавши усі необхідні роботи по модулюванню обчислення ймовірності розпізнавання і класифікації об'єктів на зображенні та проведення експериментів та аналізу, можна зробити висновок, отримана нова нейронна мережа повнозв'язна. Створена нейронна мережа навчається дуже швидко: від 60 до 120 секунд в залежності від потужності комп'ютера.

## ВИСНОВКИ

Для того, щоб побудувати якісний класифікатор, необхідно мати якісні дані. Жоден з методів побудови класифікаторів, заснований на нейронних мережах або статистичний, ніколи не дасть класифікатор потрібної якості, якщо наявний набір прикладів не буде досить повним і представницьким для такого завдання, з яким доведеться працювати системі.

Обраний метод обчислення ймовірності з використанням моделей нейронних мереж, дозволить отримувати більш точні і своєчасні дані за короткий проміжок часу. Також було визначено основні завдання, які необхідно вирішити для досягнення поставленої мети. Коротко був описаний об'єкт дослідження і розроблена математична постановка задачі. Проведено аналіз використання моделей та виявлено переваги і недоліки розглянутих моделей.

Розроблено нову модель розпізнавання і класифікації об'єктів на зображенні, проведено навчання нейронної мережі за методом зворотне поширення помилки. Нова модель відноситься до нейромережевого класу моделей і має єдиний параметр, що спрощує завдання ідентифікації моделі, усуваючи основний недолік моделей даного класу.

### *Список використаних джерел:*

1. Barlow, H. B. Selective sensitivity to *direction* of movement in ganglion cells of the rabbit retina / H. B. Barlow, R. M. Hill // Science. – 1963. – Vol. 139. – P. 412–414.
2. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) DC, USA: IEEE Computer Society. – Washington, 2016. – P. 2342–2354.
3. Dropout: A Simple Way to Prevent Neural Networks from Overfittin / N. Srivastava [et al.] // Journal of Machine Learning Research. – 2014. – Vol. 15. – P. 1929–1958.
4. Jiawei Su One pixel attack for fooling deep neural networks / Jiawei Su, D. V. Vargas, Kouichi Sakurai // IEEE Transactions on Evolutionary Computation. – 2019. – 04 January

Левицкая Т. А., Петин В. Д.

## ОСОБЕННОСТИ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ЗАДАЧА КЛАССИФИКАЦИИ

*В статье рассматриваются алгоритмы машинного распознавания и классификации объектов, используемые для решения задач компьютерного зрения на мобильных устройствах. Описаны их недостатки и преимущества. Установлено, что многие алгоритмы обнаружения являются неустойчивыми к различным признакам или свойствам изображения, на котором происходит поиск распознаваемого объекта, который может быть повернут, искажен или даже деформирован. В работе предложено использование сверточной нейронной сети для устранения всех указанных недостатков. Для реализации поставленных задач выбраны современные информационные технологии TensorFlow, Keras, Core ML 2. Разработанная система использует среду глубокого обучения, которая поддерживает много типов машинного обучения, нацеленных в первую очередь на решение задач классификации и сегментации изображений в мобильной разработке.*

**Ключевые слова:** алгоритм; распознавание и классификация данных; нейронная сеть; машинное обучение.

Levitskaya T., Petin V.

## FEATURES OF APPLICATION OF NEURAL NETWORKS FOR CLASSIFICATION TASKS

*This article is devoted to the creation of a classifier model, which is implemented in a project developed for a mobile application on the iOS platform, which calculates the percentage probability of matching the recognized object in the selected image with the class of the trained model.*

*The relevance of this article is due to the fact that the algorithms used for recognition and classification of objects have a high accuracy in the percentage probability of matching the recognized object in the selected image with the class of the trained model. The practical value of the developed system lies in the application of algorithms for recognition and classification of objects on the selected image in a mobile application on the iOS platform. The specific implementation of the classifier model and the mobile application on the iOS platform, based on this classifier model, developed within the framework of the thesis, is quite accurate and stable, and the mobile application is visual. Technical solutions look reasonable, do not raise questions and correspond to the task. Based on the results of the study, the advantages are the high accuracy of calculating the correspondence of the recognized object in the selected image with the class of the trained model of the classifier. The system on the iOS platform is implemented as an application containing the classifier model and the project itself to implement the calculation of the percentage probability of matching the object in the image using this classifier model. The neural network has reached 97 % accuracy on training data and 81 % on test Data. By results of comparison of the received model with already existing models its adequacy and novelty is shown. The prospects for the development of this direction were justified.*

**Keywords:** neural networks, machine learning, iOS, mobile app, object recognition and classification.

Рекомендована к публикации: доц., канд. техн. наук, Мироненко Д. С.

Статья принята: 12.01.2019 г.

Режим доступа: <http://sap.pstu.edu>