УДК 004.4

КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

С. А. Филиппов[0009-0007-9015-7982]

Институт информационных технологий и интеллектуальных систем Казанского федерального университета, ул. Кремлевская, 35, г. Казань, 420008

woppilif@icloud.com

Аннотация

Для классификации изображений в настоящее время можно применить множество различных инструментов, каждый из которых направлен на решение определенного спектра задач. В статье проведен краткий обзор библиотек и технологий для классификации изображений. Построена архитектура простой свёрточной нейронной сети для классификации изображений.

Были проведены эксперименты по распознаванию изображений с такими популярными нейронными сетями, как VGG16 и ResNet 50. Обе нейронные сети показали хорошие результаты. Однако ResNet 50 переобучилась из-за того, что в наборе данных присутствовали однотипные изображения для обучения, поскольку в данной нейронной сети больше слоев, позволяющих считывать признаки объектов на изображениях. С обученными моделями был проведен сравнительный анализ по распознаванию изображений, специально подготовленных для этого эксперимента.

***Ключевые слова**:* распознавание изображений, нейронная сеть, сверточная нейронная сеть, классификация изображений, машинное обучение

ВВЕДЕНИЕ

Машинное обучение — направление, которое в настоящее время актуально для решения различных задач. Такие задачи, как генерация изображений [1], генерация музыки [2], создание сцен для игр [3], озвучивание видеороликов в интернете [4], синхронный перевод [5] — лишь небольшой список того, каким потенциалом обладает машинное обучение.

Названная технология развивается большинством крупных компаний, разрабатывающих программное обеспечение. Такие компании, как показывает практика, могут как создавать свои собственные библиотеки для работы с машинным обучением, так и вносить вклад в разработки с открытым исходным кодом.

Так, например, компания Google разработала библиотеку под названием TensorFlow [6] — открытую библиотеку для машинного обучения, позволяющую решать большой спектр задач. На официальном сайте, посвященном этой библиотеке, представлены многочисленные примеры и варианты использования. В контексте этой статьи рассмотрен вопрос классификации изображений встроенными средствами этой библиотеки с использованием дополнительных функций библиотеки Keras [7].

Для классификации изображений применяют сверточные нейронные сети (Convolutional neural network, CNN) [8]. Во время обучения нейронной сети этого типа выделяют определенные признаки, на основе которых происходит обучение.

Библиотека TensorFlow позволяет спроектировать собственную нейронную сеть и установить специфические параметры обучения. В случае, когда в разработке собственной архитектуры сети нет необходимости, можно воспользоваться готовыми решениями. VGG16 [9] и ResNet 50 [10] — яркий пример спроектированных нейронных сетей, готовых для решения задач классификации изображений.

Далее описаны эксперименты по разработке собственной архитектуры нейронной сети, а также представлены результаты вычислительных экспериментов для сетей VGG16 и ResNet 50.

1. ПРОЕКТИРОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Для проектирования нейронной сети с целью распознавания изображений с помощью CNN необходимо: подготовить наборы данных для обучения и тестирования, определить параметры размеров изображений, выстроить слои нейронной сети в определённом порядке, настроить эти слои и при этом для повышения точности предсказаний с помощью таких средств, как оптимизация гиперпараметров [11], увеличение данных [12], методы регуляризации [13], провести более

тонкую настройку эмпирическим путем, а также интерпретировать результаты обучения и сделать выводы.

В качестве набора данных для обучения нейронной сети использовался дата-сет Fruits 360 [14]. Из него было выбрано десять классов изображений: 'Apple Golden 1', 'Apricot', 'Beetroot', 'Cherry 1', 'Corn', 'Guava', 'Lemon', 'Orange', 'Tomato 1', 'Watermelon'. Для каждого класса в данном наборе присутствует несколько изображений, показывающих объект на 360 градусов. Так, в наборе присутствует 5038 изображений для обучения и 1681 изображение для тестирования. Некоторые из них представлены на рисунке 1.

Рис. 1. Пример данных для обучения из набора данных.

В качестве языка программирования для разработки нейронной сети был выбран Python. Разработка осуществлялась в интегрированной среде разработки PyCharm [15] на компьютере под управлением MacOS [16] на чипе M1 [17].

Процесс обучения занял три часа без применения дискретной графической карты, так как стандартные средства библиотеки TensorFlow осуществляют работу с CPU для вычислений, а не с GPU. Для большей производительности была установлена дополнительная библиотека TensorFlow MacOS [18] для процессоров M1,

что позволило получить доступ к GPU и увеличить производительность в несколько раз.

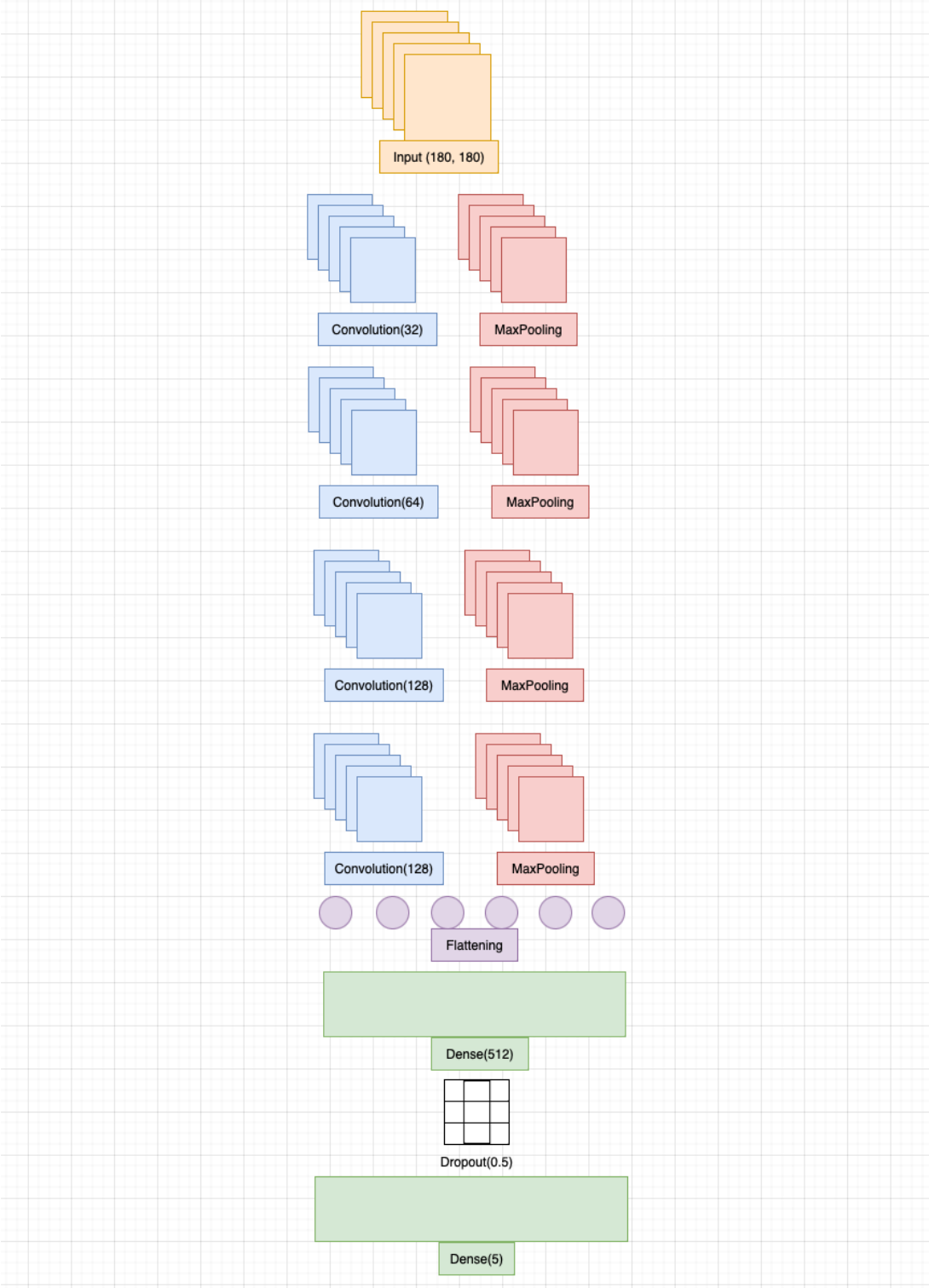


Рис. 2. Слои нейронной сети.

Далее были выстроены слои нейронной сети, которые можно увидеть на рисунке 2. Слои были выделены разными цветами, чтобы показать, как они упорядочены. Соответственно, нейронная сеть состоит из следующих слоев:

1. Слой Conv2D [19] с 32 фильтрами размером 3x3 и функцией активации ReLU, который обрабатывает входное изображение размером *m* x *n* x 3 (3 соответствует RGB-каналам изображения), где *m* — ширина изображения, а *n* — его высота. Этот слой выполняет первичное извлечение признаков из входного изображения.

2. Слой MaxPooling2D [20] с размером фильтра 2x2, который уменьшает размерность выходных данных в два раза. Это позволяет уменьшить количество параметров, которые необходимо обучить, и избежать переобучения модели.

3. Последовательность из двух слоев Conv2D и MaxPooling2D, аналогичных первым двум, но с большим количеством фильтров (64 и 128 соответственно). Эти слои выполняют более сложное извлечение признаков из входного изображения.

4. Еще один слой Conv2D с 128 фильтрами размером 3x3 и функцией активации ReLU, за которым следует слой MaxPooling2D с размером фильтра 2x2. Эти слои позволяют еще более точно извлечь признаки из изображения.

5. Слой Flatten [21], который преобразует выходные данные из последнего слоя в одномерный массив.

6. Полносвязный слой Dense [22] с 512 нейронами и функцией активации ReLU, который выполняет обработку извлеченных признаков и уменьшение их размерности до 512.

7. Слой Dropout [23], который помогает предотвратить переобучение модели, выбрасывая случайно выбранные нейроны во время обучения.

8. Полносвязный слой Dense с 5 нейронами и функцией активации softmax [24], который генерирует вероятности принадлежности изображения к каждому из 5 классов. Это позволяет модели выбрать наиболее подходящий класс для данного изображения.

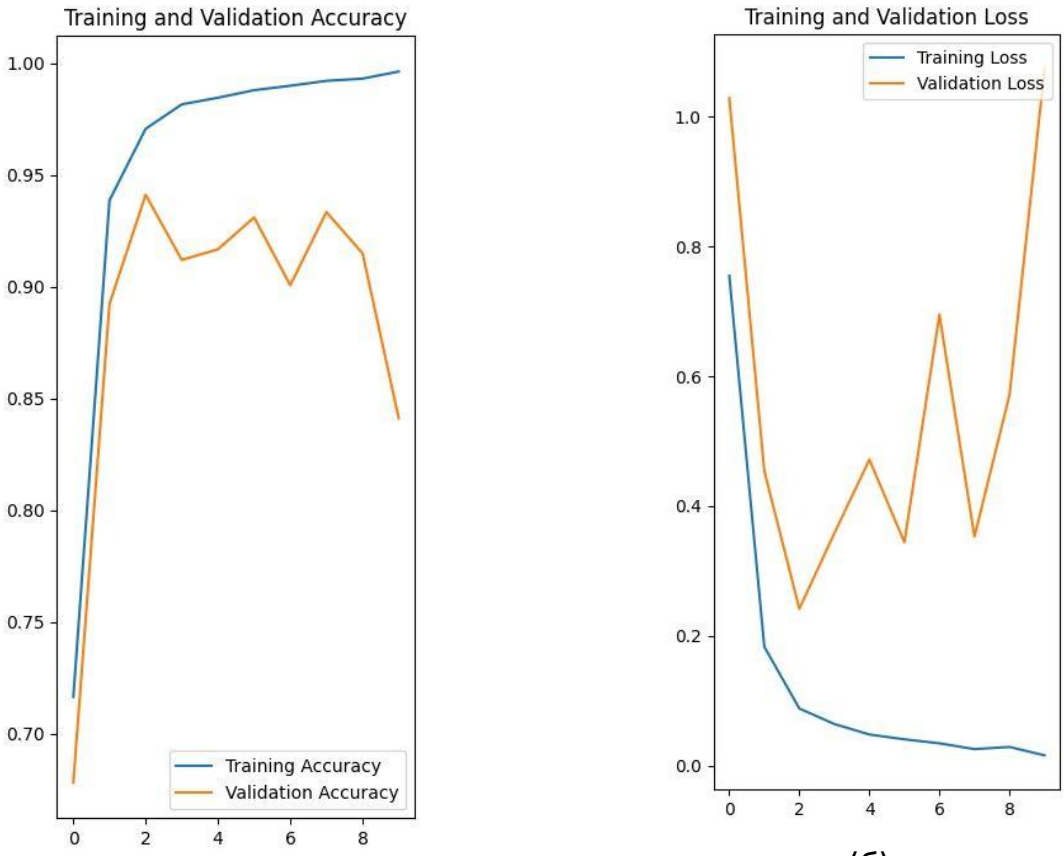
Далее было проведено обучение нейронной сети. Для ясности введем несколько терминов:

1. Точность — это показатель того, насколько хорошо работает модель классификации. Обычно она выражается в процентах и представляет собой количество правильных прогнозов из общего числа. Точность является бинарной, то есть она может быть истинной или ложной только для конкретной выборки. Это значение часто строится и отслеживается во время обучения, и иногда оно используется для оценки общей точности модели. Точность легче понять, чем потери, что является еще одним показателем производительности модели.

2. Функция потерь, также известная как функция затрат, используется для измерения точности прогнозов модели. Она учитывает вероятность или неопределенность прогноза в зависимости от того, насколько близок прогноз к истинному значению. Это позволяет нам получить более детальное представление о том, насколько хорошо работает модель.

2. РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ С РАЗРАБОТАННОЙ НЕЙРОННОЙ СЕТЬЮ

Ниже приведены результаты обучения нейронной сети, которое состояло из 10 эпох. После обучения были получены значения точности обучения и функции потерь, которые отображены на рисунке 3.



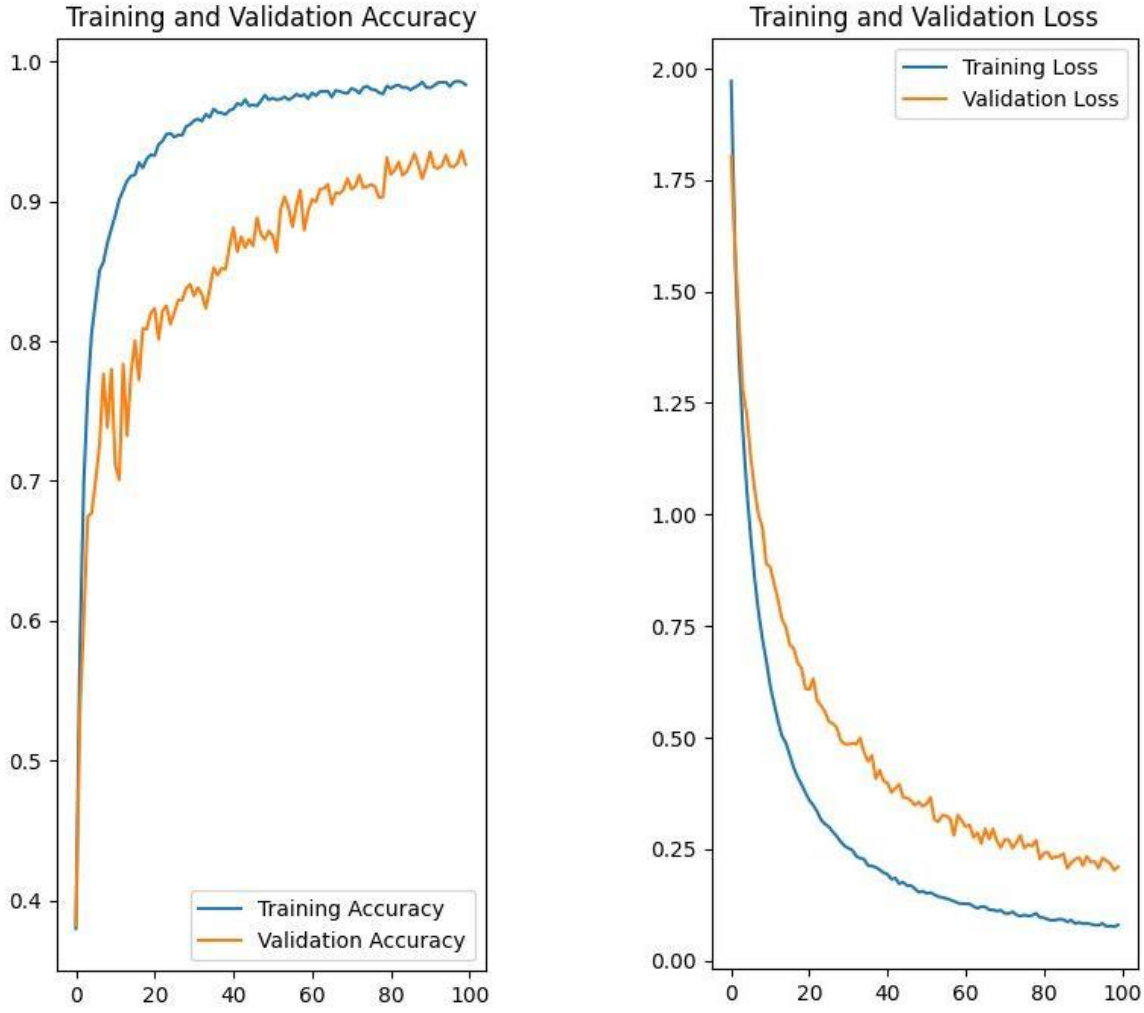
(а) (б)

Рис. 3. Графики точности и потерь для разработанной нейронной сети.

Из графика точности (рис. 3 а) видно, что нейронная сеть на первом этапе обучения показывает результаты, близкие к истинным значениям. Далее положительные результаты заметно снижаются, и на графике видно, как в процессе обучения нейронная сеть деградирует. На графике потерь (рис. 3 б) отражена аналогичная ситуация.

3. РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ С НЕЙРОННОЙ СЕТЬЮ VGG16

VGG16 по сравнению с разработанной моделью имеет большее количество слоев для выявления признаков изображений, что положительно сказывается на показателях точности модели, как показано на рис. 4 а.

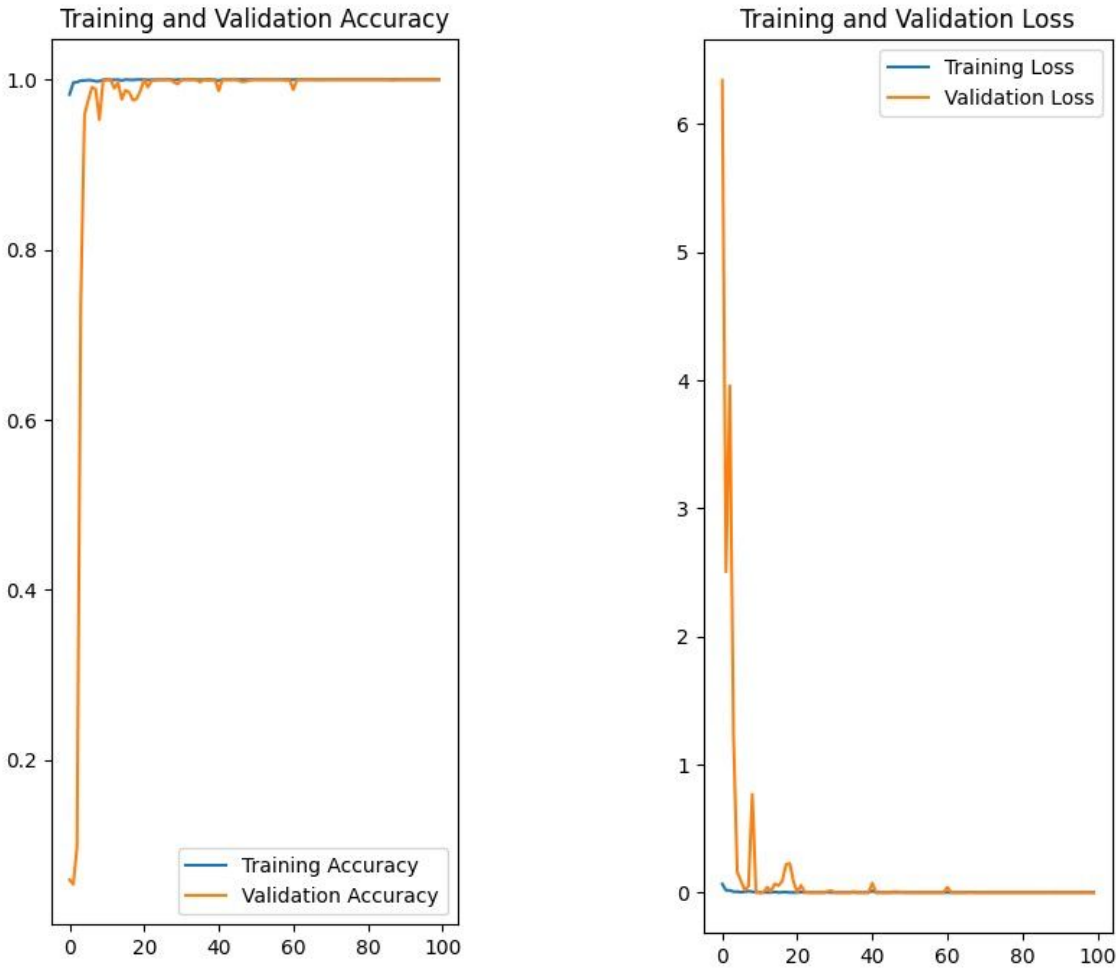


(а) (б)

Рис. 4. Графики точности и потерь для VGG16.

Постепенное обучение модели показано на первых этапах обучения – тогда, когда линии на графике точности пересекались, а после стали расходиться. Если бы графики все время пересекались, то это бы говорило о том, что обучаемая сеть переобучилась. Переобучение возникает тогда, когда обучение начинает происходить от различных шумов в данных или объем данных для обучения слишком мал. Далее, модель не может классифицировать тестовые данные верно, так как она обучена на слишком большом количестве ложных параметров.

4. РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ С НЕЙРОННОЙ СЕТЬЮ RESNET 50



(а) (б)

Рис. 5. Графики точности и потерь для ResNet 50.

ResNet 50 – альтернатива для VGG 16, с большим количеством слоев. Слои в этой нейронной сети следуют правилу: количество фильтров на каждом уровне равно количеству выходных данных модели, что влияет на ее производительность. Признаки с изображений считываются в большем объеме, что влияет на показатели обучения сети в положительную сторону. Такую модель можно применять для классификации изображений с большим количеством объектов на них. В случае, если изображения содержат мало объектов для считывания признаков о них, можно получить переобученную модель уже на первых трех десятках эпох ее обучения, как это показано на рисунках 5(а) и 5(б).

5. РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

После обучения нейронных сетей был выбран набор данных для тестирования функции распознавания изображений. Результаты представлены на рисунках 6 и 7.

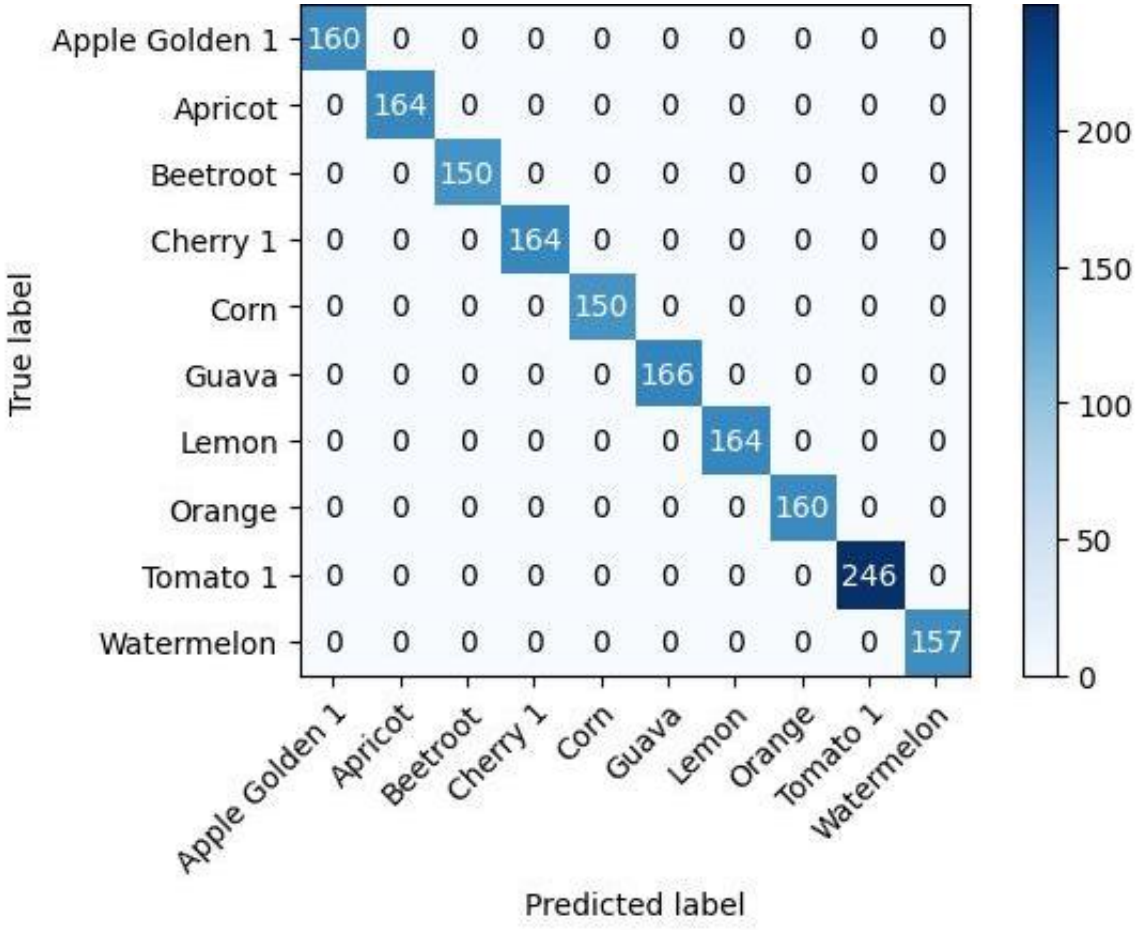


Рис. 6. Результаты эксперимента по классификации для модели ResNet 50.

Приведенная матрица на оси ординат показывает те значения, которые являются истинными, а на оси абсцисс – те, которые были предсказаны. Здесь наглядно видно результат переобучения. Нейронная сеть ни разу не ошиблась, что говорит о её переобучении.

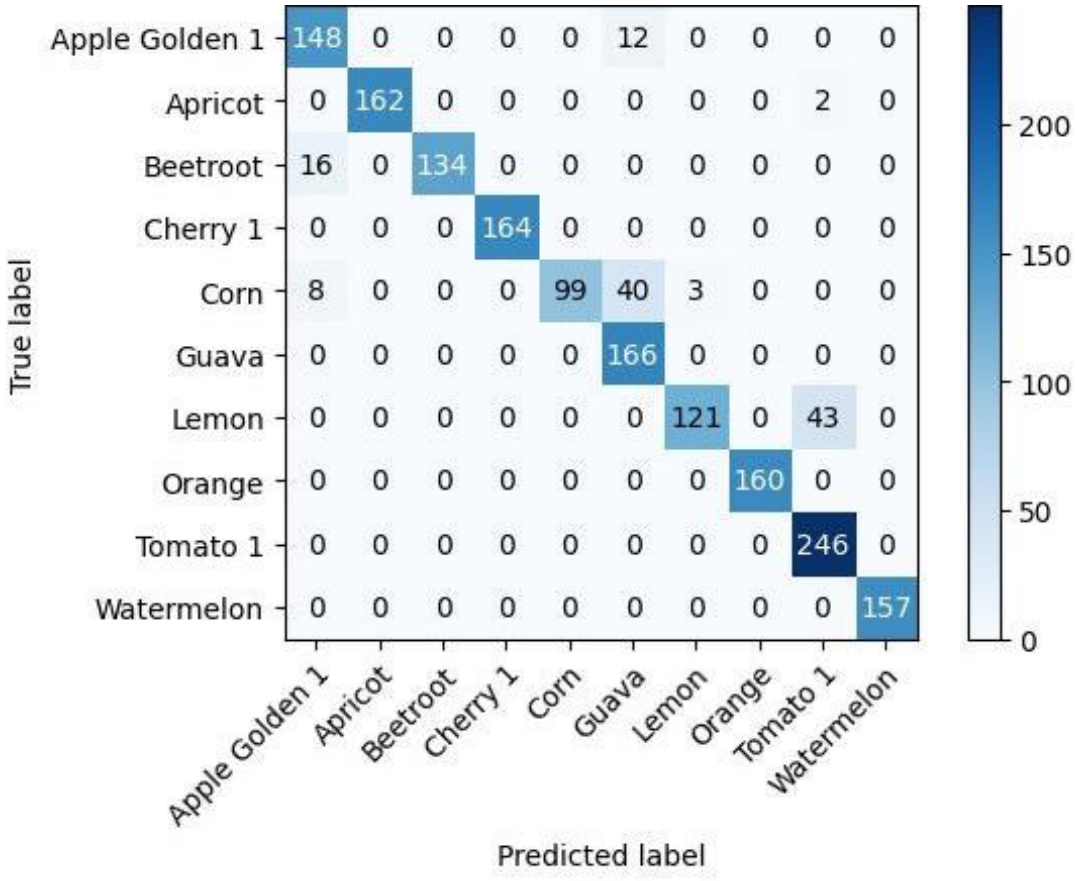


Рис. 7. Результаты эксперимента по классификации для модели VGG16.

ЗАКЛЮЧЕНИЕ

Машинное обучение позволяет решать задачи классификации изображений как с использованием самостоятельно разработанной модели нейронной сети, так и с использованием известных решений для этих задач. Вычислительные эксперименты показали, что с задачей лучше всего справилась модель VGG 16 – модель не переобучилась и показала результаты лучше своего аналога ResNet 50. В подготовленном наборе данных присутствовали изображения, в которых нет шумов и объектов, не относящихся к данным для обучения. Так, было определено, что для решения задач классификации изображений необязательно использовать такие большие модели, как ResNet 50, и был сделан вывод, что для эффективного обучения и получения результатов необходимо иметь хорошо подготовленный набор данных для решения специфичной задачи машинного обучения.

Благодарности

Выражаю благодарность доценту Института информационных технологий и интеллектуальных систем Казанского федерального университета М.О. Таланову за консультации по данной теме.

СПИСОК ЛИТЕРАТУРЫ

1. *Фисько Д.В.* Обзор методов условной генерации изображений нейросетевыми моделями // Актуальные вопросы современной науки и технологий. 2021. С. 57–62.

2. *Мосин Е.Д., Белов Ю.С.* Генерация музыки с использованием двунаправленной рекуррентной нейронной сети // Научное обозрение. Технические науки. 2023. № 1. С. 10–14. https://doi.org/10.17513/srts.1419

3. *Козар Б.А., Кугуракова В.В., Сахибгареева Г.Ф.* Структуризация сущностей естественного текста с использованием нейронных сетей для генерации трехмерных сцен // Программные продукты и системы. 2022. Т. 35. № 3. С. 329–339. https://doi.org/10.15827/0236-235X.139.329-339.

4. *Пантюхин Д.В.* Нейронные сети синтеза речи голосовых помощников и поющих автоматов // Речевые технологии/Speech Technologies. 2021. № 3-4. С. 3–16. https://doi.org/10.58633/2305-8129_2021_3-4_3

5. *Шамансуров Ш.* Влияние искусственного интеллекта на развитие области синхронного перевода // Oriental renaissance: Innovative, educational, natural and social sciences. 2023. Т. 3. № 23. С. 305–309.
https://doi.org/10.5281/zenodo.10365801

6. *Pang B., Nijkamp E., Wu Y.N.* Deep learning with tensorflow: a review // Journal of Educational and Behavioral Statistics. 2020. Vol. 45. No. 2. P. 227–248. https://doi.org/10.3102/1076998619872761

7. *Ketkar N., Ketkar N.* Introduction to Keras // Deep learning with python: a hands-on introduction. 2017. P. 97–111.
https://doi.org/10.1007/978-1-4842-2766-4_7

8. *Convolutional Neural Networks.*
URL: https://www.ibm.com/topics/convolutional-neural-networks

9. *Sapijaszko G., Mikhael W.B.* An overview of recent convolutional neural network algorithms for image recognition // 2018 IEEE 61st International midwest symposium on circuits and systems (MWSCAS). IEEE, 2018. P. 743–746. https://doi.org/10.1109/MWSCAS.2018.8623911

10. *He K. et al.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 770–778. https://doi.org/10.48550/arXiv.1512.03385

11. *Nguyen V.* Bayesian optimization for accelerating hyper-parameter tuning // 2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE). IEEE. 2019. P. 302–305. https://doi.org/10.1109/AIKE.2019.00060

12. *Poojary R., Raina R., Mondal A.K.* Effect of data-augmentation on fine-tuned CNN model performance // IAES International Journal of Artificial Intelligence. 2021. Vol. 10. No. 1. P. 84. https://doi.org/10.11591/ijai.v10.i1.pp84-92

13. *Tian Y., Zhang Y.* A comprehensive survey on regularization strategies in machine learning // Information Fusion. 2022. Vol. 80. P. 146–166. https://doi.org/10.1016/j.inffus.2021.11.005

14. Fruits 360. URL: https://www.kaggle.com/datasets/moltean/fruits

15. PyCharm, Quick start guide. URL: https://www.jetbrains.com/help/pycharm/quick-start-guide.html

16. MacOS. URL: https://www.apple.com/za/macos/what-is/

17. *Zhang Z.* Analysis of the Advantages of the M1 CPU and Its Impact on the Future Development of Apple // 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE). IEEE, 2021. P. 732–735. https://doi.org/10.1109/ICBASE53849.2021.00143

18. Tensorflow Macos. URL: https://developer.apple.com/metal/tensorflow-plugin/

19. Conv2D layer. URL: https://keras.io/api/layers/convolution_layers/convolution2d/

20. MaxPooling2D layer. URL: https://keras.io/api/layers/pooling_layers/max_pooling2d/

21. Flatten layer. URL: https://keras.io/api/layers/reshaping_layers/flatten/

22. Dense layer. URL: https://keras.io/api/layers/core_layers/dense/

23. Dropout layer.
URL: https://keras.io/api/layers/regularization_layers/dropout/
24. Layer activation functions, Softmax function.
URL: https://keras.io/api/layers/activations/#softmax-function

IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

S. A. Filippov[0009-0007-9015-7982]

Institute of Information Technologies and Intelligent Systems of Kazan Federal University, 35 Kremlevskaya str., Kazan, 420008

woppilif@icloud.com

Abstract

Nowadays, many different tools can be used to classify images, each of which is aimed at solving a certain range of tasks. This article provides a brief overview of libraries and technologies for image classification. The architecture of a simple convolutional neural network for image classification is built. Image recognition experiments have been conducted with popular neural networks such as VGG 16 and ResNet 50. Both neural networks have shown good results. However, ResNet 50 overfitted due to the fact that the dataset contained the same type of images for training, since this neural network has more layers that allow reading the attributes of objects in the images. A comparative analysis of image recognition specially prepared for this experiment was carried out with the trained models.

***Keywords**:* image recognition, neural network, convolutional neural network, image classification, machine learning

REFERENCES

1. *Fisko D.V.* Overview of methods for conditional image synthesis // Aktual'nye voprosy sovremennoj nauki i tehnologij. 2021. S. 57–62.
2. *Mosin E.D., Belov Yu.S.* Music generation using a bidirectional recurrent neural network // Nauchnoe obozrenie. Technical science. 2023. No. 1. P. 10–14. https://doi.org/10.17513/srts.1419

3. *Kozar B.A., Kugurakova V.V., Sakhibgareeva G.F.* Structuring natural text entities using neural networks for generating 3d-scenes // Software & Systems. 2022. V. 35. No. 3. S. 329–339. https://doi.org/10.15827/0236-235X.139.329-339.

4. *Pantiukhin D.V.* Neural networks for speech synthesis of voice assistants and singing machines // Rechevye tehnologii. 2021. No. 3-4. S. 3–16. https://doi.org/10.58633/2305-8129_2021_3-4_3

5. *Shamansurov Sh.* Vlijanie iskusstvennogo intellekta na razvitie oblasti sinhronnogo perevoda // Oriental renaissance: Innovative, educational, natural and social sciences. 2023. V. 3. No. 23. S. 305–309. https://doi.org/10.5281/zenodo.10365801

6. *Pang B., Nijkamp E., Wu Y.N.* Deep learning with tensorflow: A review // Journal of Educational and Behavioral Statistics. 2020. Vol. 45. No. 2. P. 227–248. https://doi.org/10.3102/1076998619872761

7. *Ketkar N., Ketkar N.* Introduction to Keras // Deep learning with python: a hands-on introduction. 2017. P. 97–111. https://doi.org/10.1007/978-1-4842-2766-4_7

8. *Convolutional Neural Networks.* URL: https://www.ibm.com/topics/convolutional-neural-networks

9. *Sapijaszko G., Mikhael W.B.* An overview of recent convolutional neural network algorithms for image recognition // 2018 IEEE 61st International midwest symposium on circuits and systems (MWSCAS). IEEE, 2018. P. 743–746. https://doi.org/10.1109/MWSCAS.2018.8623911

10. *He K. et al.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 770–778. https://doi.org/10.48550/arXiv.1512.03385

11. *Nguyen V.* Bayesian optimization for accelerating hyper-parameter tuning // 2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE). IEEE. 2019. P. 302–305. https://doi.org/10.1109/AIKE.2019.00060

12. *Poojary R., Raina R., Mondal A.K.* Effect of data-augmentation on fine-tuned CNN model performance // IAES International Journal of Artificial Intelligence. 2021. Vol. 10. No. 1. P. 84. https://doi.org/10.11591/ijai.v10.i1.pp84-92

13. *Tian Y., Zhang Y.* A comprehensive survey on regularization strategies in machine learning // Information Fusion. 2022. Vol. 80. P. 146–166. https://doi.org/10.1016/j.inffus.2021.11.005

14. Fruits 360. URL: https://www.kaggle.com/datasets/moltean/fruits

15. PyCharm, Quick start guide. URL: https://www.jetbrains.com/help/pycharm/quick-start-guide.html

16. MacOS. URL: https://www.apple.com/za/macos/what-is/

17. *Zhang Z.* Analysis of the Advantages of the M1 CPU and Its Impact on the Future Development of Apple // 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE). IEEE, 2021. P. 732–735. https://doi.org/10.1109/ICBASE53849.2021.00143

18. Tensorflow Macos. URL: https://developer.apple.com/metal/tensorflow-plugin/

19. Conv2D layer. URL: https://keras.io/api/layers/convolution_layers/convolution2d/

20. MaxPooling2D layer. URL: https://keras.io/api/layers/pooling_layers/max_pooling2d/

21. Flatten layer. URL: https://keras.io/api/layers/reshaping_layers/flatten/

22. Dense layer. URL: https://keras.io/api/layers/core_layers/dense/

23. Dropout layer. URL: https://keras.io/api/layers/regularization_layers/dropout/

24. Layer activation functions, Softmax function. URL: https://keras.io/api/layers/activations/#softmax-function

СВЕДЕНИЯ ОБ АВТОРЕ

ФИЛИППОВ Сергей Алексеевич – магистрант второго курса Института информационных технологий и интеллектуальных систем Казанского федерального университета.

Sergey Alekseevich FILIPPOV – second-year master's student at the Institute of Information Technologies and Intelligent Systems of the Kazan Federal University.

email: woppilif@icloud.com;

ORCID: 0009-0007-9015-7982

Материал поступил в редакцию 24 мая 2024 года