

## Результаты разработки программы распознавания изображений на языке Python

В. С. Ростовцев<sup>1</sup>, О. Н. Черемисинова<sup>2</sup>

<sup>1</sup>кандидат технических наук, доцент кафедры электронных вычислительных машин, Вятский государственный университет. Россия, г. Киров. E-mail: rostov\_kirov@mail.ru

<sup>2</sup>студент 4 курса кафедры электронных вычислительных машин, Вятский государственный университет. Россия, г. Киров. E-mail: cheremisinova8@gmail.com

**Аннотация.** В статье рассматриваются принципы и результаты разработки программы распознавания образов на примере идентификации породы кошек, используя изображения, представленные в формате \*.jpeg. В настоящее время не существует точных методов распознавания образов, применение которых не сопровождалось бы ошибками распознавания, даже с учетом значительных временных затрат. Анализ современных литературных источников показывает, что наиболее перспективным подходом для решения данной задачи является подход с использованием сверточных нейронных сетей.

Результаты разработки программы распознавания образов позволяют оценить затраты на ее создание, экспериментальное моделирование и показывают качество распознавания при использовании библиотеки Keras языка Python. Важным моментом является использование режима аугментации, который позволяет автоматически генерировать расширенную обучающую выборку путем случайного изменения и последовательного добавления искажений к изображению. В качестве таких операций могут выступать поворот, растягивание, сдвиг и некоторые другие.

**Ключевые слова:** распознавание изображений, машинное обучение, сверточные нейронные сети, аугментация.

**Введение.** Вопросам распознавания образов посвящено много публикаций. Однако на сегодняшний день не существует общепринятой методики оценки эффективности применения нейронных сетей для задач распознавания образов. Программа распознавания изображений, разработанная на языке Python, демонстрирует затраты и эффективность применения готовой библиотеки Keras [2] для решения поставленной задачи [5]. Отличительной особенностью разработанной программы является отсутствие затрат на приобретение готовых программных продуктов, так как язык Python и библиотека Keras являются свободно распространяемыми.

**Цель исследования.** Сверточная нейронная сеть (англ. convolutional neural network, CNN), нацеленная на решение этой проблемы, впервые была предложена Яном Лекуном в 1988 году. Данная модель учитывает особенности работы зрительной коры головного мозга и основывается на переходе от выявленных простых признаков (линий и кривых) к более сложным абстракциям. Достоинство сверточных нейронных сетей – меньшее количество весов, которые нужно вычислить. Качество распознавания значительно выше, чем у полносвязной нейронной сети за счет выделения отличительных черт объекта, а не попиксельного анализа изображения. К недостаткам можно отнести большое количество варьируемых параметров сети.

**Методы исследования.** В качестве тестовой задачи в данной работе используется распознавание породы кошки по ее фотографии. Большинство существующих аналогов не способны автоматически определить породу кошки по фотографии.

Для разработки программы выбран язык Python [2] с применением библиотеки Keras, представляющей собой надстройку над фреймворками TensorFlow, Theano и др. Для бэкэнда будет использоваться TensorFlow, с которой Keras имеет максимальную совместимость [2].

Входными данными для сверточной нейронной сети являются изображения одинакового размера, представленные в виде матрицы чисел в диапазоне от 0 до 255, определяющие интенсивность цвета пикселя. Матрица имеет размер  $N \times N \times 3$ , где  $N$  – ширина и высота изображения, а 3 – количество цветовых каналов. Матрица нормализуется, при этом числа преобразуются в диапазон от 0 до 1. Нормализация существенно увеличивает точность распознавания за счет выравнивания вклада отдельных признаков, исключая доминирование одних полей перед другими.

Для качественного обучения требуется большое число разнообразных примеров обучающей выборки. Существует способ их искусственной генерации – аугментация [1], который заключается в последовательном добавлении искажений к изображению, таких как поворот, растягивание, сдвиг и др., за счет чего получаются новые изображения. В библиотеке Keras содержатся готовые сред-

ства для аугментации изображения, позволяющие производить сдвиг, растягивание, поворот и другие операции на случайную величину из заданного интервала. Выбор преобразований должен основываться на наиболее часто встречающихся искажениях в исходных данных. Разработчик экспериментально подбирает параметры аугментации таким образом, чтобы качество распознавания не упало и нейронная сеть научилась распознавать зашумленные данные. На рис. 1 приведен фрагмент исходного кода задания параметров аугментации.

```
datagen = ImageDataGenerator(  
    rotation_range=20,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True)
```

Рис. 1. Задание параметров аугментации

Все изображения делятся на два набора: тренировочные и валидационные. Тренировочные данные используются для настройки весов нейронной сети с помощью минимизации функции ошибки в процессе обучения. Объем валидационных данных гораздо меньше тренировочного и обычно составляет до 10% от тренировочного. На валидационных данных нейросеть не обучается, поэтому они позволяют следить за появлением эффекта переобучения нейронной сети. Валидационные данные, так же как и тренировочные, должны включать в себя изображения всех классов. Разделим изображения каждого класса на две группы в соотношении 4/1, где первая часть будет являться тренировочными данными, а вторая – валидационными.

Основным блоком сверточной нейронной сети является слой свертки, содержащий в себе фильтр с коллекцией ядер (матриц) свертки. Суть операции свертки заключается в том, что участок изображения поэлементно умножается на матрицу свертки. После этого результаты суммируются и записываются в соответствующую ячейку выходной матрицы [6]. Цель данного действия – найти определенные признаки на изображении.

К полученным элементам матрицы применяется функция активации, в качестве которой чаще всего выступает ReLU. Она вычисляется как  $f(x)=\max(0,x)$ . Данная функция существенно ускоряет процесс обучения и упрощает вычисления по сравнению с другими функциями активации.

Следующий слой – слой пулинга – делает карту признаков менее подробной, сокращая группу пикселей (обычно размером 2x2) до одного пикселя. При этом используется нелинейное преобразование, чаще всего – функция максимума.

Данные три слоя, повторяясь некоторое количество раз с разными параметрами, образуют первую часть сверточной нейронной сети.

Вторая часть – полносвязная нейронная сеть, которая выводит K-мерный вектор, где K – число распознаваемых классов. В ней слои не обладают пространственной структурой. Задачей данной части является определение класса объекта на основе выявленных признаков.

Архитектура выбранной нейронной сети разработана на основе проведенного анализа наиболее успешных решений, изложенных в специальной литературе. Ядро свертки сверточных слоев имеет размер 3x3, каждый последующий сверточный слой имеет количество карт признаков в два раза большее, чем у предыдущего слоя. В слое пулинга используется фильтр 2x2. Полносвязная сеть имеет 4 слоя: 128 нейронов во входном слое, по 256 нейронов в скрытых слоях, 3 нейрона в выходном слое (равно количеству возможных классов изображения). На рис. 2 представлена архитектура выбранной нейронной сети.

Обучение данной нейронной сети осуществлялось с использованием метода обратного распространения ошибки [4].

Количество тренировочных изображений в каждом классе выбрано следующее: персидская кошка – 1049, сиамская кошка – 1234, сфинкс – 1001. Это достаточно небольшой обучающий набор, из-за чего есть вероятность переобучения сети. Чтобы избежать этого, используется аугментация и функция Dropout с большим коэффициентом вероятности. Аугментация позволяет увеличить обучающую выборку, добавив в нее изображения с искажениями. Dropout – это функция регуляризации, случайно выключающая на каждом шаге несколько нейронов с заданной вероятностью. Такой подход исключает взаимную адаптацию нейронов на этапе обучения. На этапе предсказания функция Dropout не используется.

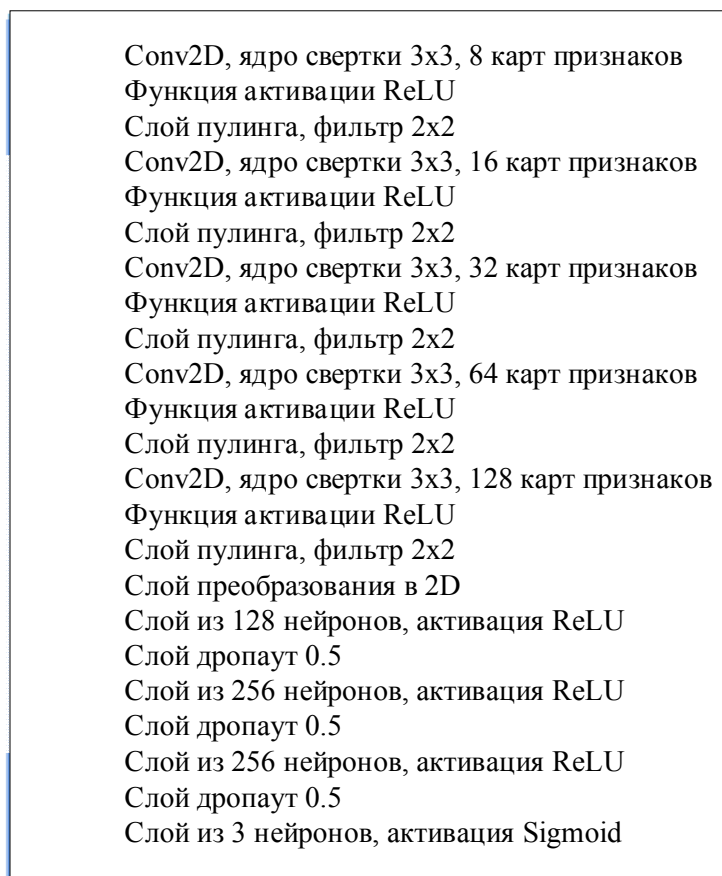


Рис. 2. Архитектура нейронной сети

*Результаты исследования.* В ходе разработки программы и проведения ряда экспериментов установлена оптимальная длительность обучения – 100 эпох по критерию минимизации ошибки обучения. Время обучения сверточной нейронной сети на процессоре CPU Intel Core i5 и GPU NVIDIA GeForce 840M составило около полутора часов.

Есть вероятность, что на распознаваемом изображении не окажется кошки и результат работы программы будет неверным. Для предотвращения подобных ошибок используется готовое решение TensorFlow Object Detection API. Это платформа с открытым исходным кодом, созданная поверх TensorFlow, с помощью которой можно найти объекты на изображении и области, в которых они расположены. Классы найденных объектов и их области записываются в массивы. Если в данном массиве не содержится класса, соответствующего кошке, определение породы не будет осуществляться.

Результаты экспериментальной апробации работы сверточной нейросети представлены в табл. 1.

Таблица 1

### Количество распознаваемых изображений

Порода кошки на изображении	Предсказанная порода			
	общее количество изображений	персидский кот	сиамский кот	сфинкс
Персидский кот	30	28	1	1
Сиамский кот	30	0	28	2
Сфинкс	30	1	6	23

На разработку программы было затрачено около полутора месяцев. В результате написана программа объемом 500 строк на языке Python. Процентное соотношение верно и неверно распознанных изображений представлено в табл. 2.

*Выводы.* Из представленных выше таблиц видно, что точность распознавания породы кошки достаточно высокая – 88%. Для рассмотренной задачи в условиях ограниченного объема данных такую вероятность можно считать вполне приемлемой. Следовательно, разработанный программный продукт можно использовать на практике для решения подобного класса задач.

Таблица 2

## Результаты распознавания (в процентах)

	Результат		
	общее количество изображений	верно распознано, (%)	неверно распознано, (%)
Персидский кот	30	93	7
Сиамский кот	30	93	7
Сфинкс	30	77	23
Всего	90	88	12

Формирование тестовых и валидационных данных заняло 25% от указанного времени. Экспериментальное моделирование и выбор параметров аугментации – примерно 35% времени.

Основными трудностями разработки программы являлись выбор оптимальной архитектуры сети и подбор параметров сверточной нейронной сети, а также выявление эффекта переобучения сети.

## Список литературы

1. Аугментация (augmentation, «раздутие») данных для обучения нейронной сети на примере печатных символов. URL: <https://habr.com/company/smartengines/blog/264677> (дата обращения: 19.12.2018).
2. Официальная документация по библиотеке глубокого обучения Keras. URL: <https://keras.io> (дата обращения: 01.11.2018).
3. Официальная документация по языку Python. URL: <https://www.python.org> (дата обращения: 01.11.2018).
4. Сверточная нейронная сеть, часть 2: обучение алгоритмом обратного распространения ошибки. URL: <https://habr.com/post/348028> (дата обращения: 05.10.2018).
5. Сверточная нейронная сеть. URL: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1> (дата обращения: 01.10.2018).
6. Что такое сверточная нейронная сеть. URL: <https://habr.com/post/309508> (дата обращения: 01.10.2018).

## Results of the development of image recognition software on Python computer language

V. S. Rostovcev<sup>1</sup>, O. N. Cheremisinova<sup>2</sup>

<sup>1</sup>PhD of technical sciences, associate professor of the Department of electronic computing machines, Vyatka State University, Russia, Kirov. E-mail: [rostov\\_kirov@mail.ru](mailto:rostov_kirov@mail.ru)

<sup>2</sup>student of the 4th year of the Department of electronic computing machines, Vyatka State University, Russia, Kirov. E-mail: [cheremisinova8@gmail.com](mailto:cheremisinova8@gmail.com)

**Abstract.** The article deals with the principles and results of the development of pattern recognition program on the example of cat breed identification, using the images presented in the \*.jpeg format. Currently, there are no precise methods of pattern recognition, the use of which would not be accompanied by errors of recognition, even taking into account the significant time costs. Analysis of modern literature shows that the most promising approach to this problem is the approach using convolutional neural networks.

The results of the development of the pattern recognition program allow us to estimate the cost of its creation, experimental modeling and show the quality of recognition when using the Keras library of Python. An important point is the use of augmentation mode, which allows you to automatically generate an extended training sample by randomly changing and sequentially adding distortion to the image. As such operations can act as rotation, stretching, shift and some others.

**Keywords:** image recognition, machine learning, convolutional neural networks, augmentation.

## References

1. *Augmentaciya (augmentation, "razdutie") dannyh dlya obucheniya nejronnoj seti na primere pechatnyh simbolov* – Augmentation ("bloat") of data for neural network training on the example of printed characters. Available at: <https://habr.com/company/smartengines/blog/264677> (date accessed: 19.12.2018).
2. *Oficial'naya dokumentaciya po biblioteke glubokogo obucheniya Keras* – Official documentation of the Keras deep learning library. Available at: <https://keras.io> (date accessed: 01.11.2018).
3. *Oficial'naya dokumentaciya po yazyku Python* – Official Python documentation. Available at: <https://www.python.org> (date accessed: 01.11.2018).
4. *Svertochnaya nejronnaya set', chast' 2: obuchenie algoritmom obratnogo rasprostraneniya oshibki* – Convolutional neural network, part 2: learning by the error back propagation algorithm. Available at: <https://habr.com/post/348028> (date accessed: 05.10.2018).
5. *Svertochnaya nejronnaya set'* – Convolutional neural network. Available at: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1> (date accessed: 01.10.2018).
6. *Chto takoe svertochnaya nejronnaya set* – What is convolutional neural network. Available at: <https://habr.com/post/309508> (date accessed: 01.10.2018).