

Научная статья

УДК 004.032.26; DOI: 10.61260/2218-13X-2024-1-68-78

ГЛУБОКАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ОБНАРУЖЕНИЯ ТОЧЕК НА ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ KERAS

Ситников Александр Иванович;

Никитенко Виталий Алексеевич;

Мельников Никита Алексеевич.

Воронежский институт МВД России, г. Воронеж, Россия.

✉ **Гладких Татьяна Васильевна.**

Воронежский государственный университет инженерных технологий, г. Воронеж, Россия

✉ **gtv1311@mail.ru**

Аннотация. Сверточные нейронные сети являются мощным инструментом для обработки и анализа изображений. Они позволяют автоматически извлекать признаки из входных данных и применять их для классификации, детектирования объектов и других задач компьютерного зрения. В статье представлена архитектура сверточных нейронных сетей, разработанная с использованием библиотеки Keras. Проведен анализ архитектуры сверточной нейронной сети, который состоит из нескольких последовательных слоев. Построена структура модели, которая будет обучаться с использованием оптимизатора Adam и мониторить метрики recall и precision в процессе обучения. Представлены результаты экспериментов, которые показали, что обученная модель успешно обнаруживает точки на изображениях, достигая высокой точности и полноты. Предложенная модель может быть использована в различных областях, где требуется обнаружение лицевых точек на фотографиях.

Ключевые слова: нейронная сеть, компьютерное зрение, анализ изображений, библиотека Keras, точки на изображениях

Для цитирования: Ситников А.И., Никитенко В.А., Мельников Н.А., Гладких Т.В. Глубокая нейронная сеть для обнаружения точек на изображениях с использованием Keras // Науч.-аналит. журн. «Вестник С.-Петербур. ун-та ГПС МЧС России». 2024. № 1. С. 68–78. DOI: 10.61260/2218-13X-2024-1-68-78.

Scientific article

DEEP NEURAL NETWORK FOR DETECTING POINTS IN IMAGES USING KERAS

Sitnikov Alexander I.;

Nikitenko Vitaly A.;

Melnikov Nikita A.

Voronezh institute of the Ministry of Internal Affairs of Russia, Voronezh, Russia.

✉ **Gladkikh Tatyana V.**

Voronezh state university of engineering technologies, Voronezh, Russia

✉ **gtv1311@mail.ru**

Abstract. Convolutional neural networks are a powerful tool for image processing and analysis. They allow you to automatically extract features from the input data and apply them to classification, object detection and other computer vision tasks. This article presents the convolutional neural networks architecture developed using the Keras library. The architecture of a convolutional neural network, which consists of several successive layers, is analyzed. The structure of the model has been built, which will be trained using the Adam optimizer and monitor the recall and precision metrics during the training process. The results of experiments are presented, which showed that the trained model successfully detects points in images, achieving

© Санкт-Петербургский университет ГПС МЧС России, 2024

high accuracy and completeness. The proposed model can be used in various areas where the detection of facial points in photographs is required.

Keywords: neural network, computer vision, image analysis, Keras library, dots on images

For citation: Sitnikov A.I., Nikitenko V.A., Melnikov N.A., Gladkikh T.V. Deep neural network for detecting points in images using Keras // Scientific and analytical journal «Vestnik Saint-Petersburg university of State fire service of EMERCOM of Russia». 2024. № 1. P. 68–78. DOI: 10.61260/2218-13X-2024-1-68-78.

Введение

В последние десятилетия нейронные сети приобрели исключительное значение в сфере компьютерного зрения и анализа изображений. Одной из важнейших задач в данной области стала точная локализация ключевых точек на изображениях. Эта технология находит широкое применение в разнообразных областях, таких как медицинская диагностика, автоматизированный контроль качества, антропометрия и ряд других сфер, где точное определение точек на изображениях имеет важное значение [1, 2].

В настоящее время существует множество подходов для анализа фотоизображений, одним из которых является глубокое машинное обучение [3]. Развитие машинного обучения в последние годы дало возможность большому количеству разработчиков начать двигаться в этом направлении, что дало большой толчок для развития данного ответвления. На данный момент существует множество предобученных моделей нейронных сетей, которые способны выполнять различные классификации. Одной из таких сетей является YOLO. YOLO – архитектура нейронной сети, написанная на языке программирования C++, целью которой является обнаружение объектов на фото- или видеопотоке. Преимуществами данной модели являются: возможность множественной классификации; точность определения и простота использования. Минусами – закрытый программный код; использование заранее идеализированных типов данных для дообучения модели, так как большинство бесплатных программ не могут обеспечить сохранение набора в данном формате.

Таким образом, было принято решение о создании собственной архитектуры модели нейронной сети, способной перекрывать все минусы модели YOLO. Преимущественно выбор падает на свёрточные нейронные сети (Convolutional Neural Networks, CNN), так как они лучше всего справляются с задачами определения ключевых точек на фотоизображениях лиц. Принцип работы свёрточной нейронной сети достаточно прост:

1. На вход модели поступает матрица изображения.
2. Изображение сворачивается посредством свёрточных слоев, представленных ядром свертки различной размерности.
3. Выравнивание данных внутри модели.
4. Анализ и вывод результата из модели.

В статье представлена глубокая нейронная сеть, разработанная с использованием библиотеки Keras, для обнаружения точек на изображениях. Модель обучается на собственном наборе данных и затем применяется для обнаружения точек на изображениях, представленных в формате XML.

Методы исследования

CNN являются мощным инструментом для обработки и анализа изображений [4]. Они позволяют автоматически извлекать признаки из входных данных и применять их для классификации, детектирования объектов и других задач компьютерного зрения. В статье представлена архитектура CNN, разработанная с использованием библиотеки Keras.

Архитектура CNN состоит из нескольких последовательных слоев. Входные данные представляются в виде трехмерного тензора размерности (512, 512, 3), где 512x512 – размер изображения, а 3 – количество цветовых каналов (RGB). Первый слой сети – свёрточный слой

с 32 фильтрами размером (3, 3), функцией активации ReLU (Rectified Linear Unit) и без использования смещения. Затем следует слой нормализации пакета и слой пулинга с размером пула (2, 2).

Далее архитектура сети повторяет эту последовательность сверточных слоев, слоев нормализации пакета и слоев пулинга, увеличивая количество фильтров с каждым слоем. В конце сети добавлены полносвязные слои для классификации. Первый полносвязный слой имеет 1024 нейрона и функцию активации ReLU, а второй – 936 нейронов и используется для изменения формы выходных данных.

Данная нейронная сеть представляет собой последовательную модель, состоящую из различных слоев.

Архитектура модели включает следующие слои:

1. Sequential Model: модель построена с использованием классической последовательной архитектуры, где слои добавляются один за другим в последовательном порядке.

2. Input Layer: на входе модели есть слой Convolution2D с 32 фильтрами, размером ядра (3, 3), активацией ReLU и без использования смещения (`use_bias=False`). Этот слой предназначен для обработки изображений размером 512x512x3 (цветное изображение).

3. BatchNormalization: после каждого слоя Convolution2D следует слой BatchNormalization, который помогает стабилизировать и ускорить обучение сети.

4. Max Pooling Layer: после каждого слоя BatchNormalization есть слой MaxPool2D для уменьшения размера данных. Слой MaxPooling уменьшает размерность данных и извлекает важные признаки из изображения.

5. LeakyReLUActivation: во втором слое Convolution2D применяется активация LeakyReLU с параметром $\alpha=0,01$. Эта функция активации поддерживает нелинейность в сети.

6. Дополнительные слои: архитектура включает в себя несколько пар слоев Convolution2D и BatchNormalization, где каждая пара состоит из слоя Convolution2D и его соответствующего слоя BatchNormalization. Эти пары помогают извлекать более абстрактные признаки из изображения.

7. Полносвязные слои: после последней пары Convolution2D и BatchNormalization следует Flatten Layer, который преобразует данные в одномерный вектор. Затем следует два полносвязных слоя (DenseLayers) с 1024 нейронами и активацией ReLU. Между ними есть слой Dropout с коэффициентом 0,1, который помогает в борьбе с переобучением.

8. Выходной слой: представляет собой Dense Layer с 936 нейронами и активацией по умолчанию. Размерность этого слоя подразумевает, что данная модель, вероятно, решает задачу регрессии или классификации с 936 классами.

9. Компиляция модели: модель компилируется с оптимизатором 'adam', функцией потерь 'mean_squared_error' и метриками recall и precision. Это указывает на то, что модель предназначена для решения задачи, связанной с оценкой точности и полноты (recall и precision).

10. Модельный чекпоинт: создается объект ModelCheckpoint, который будет сохранять веса модели в файл 'new_model.h5' при улучшении результатов в процессе обучения.

Данная архитектура представляет собой глубокую CNN с несколькими слоями, предназначенную для извлечения признаков из изображений. Модель будет обучаться с использованием оптимизатора Adam и мониторить метрики recall и precision в процессе обучения.

Первым достаточно важным параметром является оптимизатор. Оптимизатор – методы, которые используются для изменения или настройки атрибутов нейронной сети, таких как веса слоев, скорость обучения и др.

Разберем самые основные и часто используемые оптимизаторы модели:

1. Стохастический градиентный спуск (SGD).

2. Оптимизатор импульса (Momentum).

3. Адаптивная оценка момента (Adam).

4. Среднеквадратичное распространение ошибки (RMSProp).

Подробнее рассмотрим каждый из них.

Стохастический градиентный спуск – итерационный метод для оптимизации целевой функции с подходящими свойствами гладкости (дифференцируемости). Одним из больших плюсов данного оптимизатора является возможность мини-пакетной подачи данных.

Оптимизатор импульса. Плюсом данного оптимизатора является то, что он учитывает прошлые градиенты, чтобы сгладить обновления. Он помогает уменьшить колебания в сторону конвергенции оптимизатором.

Оптимизатор Adam – один из самых эффективных оптимизаторов для обучения нейронных сетей. В большей мере он сочетает в себе RMSProp и оптимизатор импульса. Вместо того чтобы адаптировать скорость обучения параметров на основе среднего первого момента (среднего значения), как в RMSProp, Adam также использует среднее значение вторых моментов градиентов. В частности, алгоритм вычисляет экспоненциальное скользящее среднее градиента и квадратичный градиент, а параметры β_1 и β_2 управляют скоростью затухания этих скользящих средних.

Преимущества оптимизатора Adam:

- простота реализации;
- вычислительная эффективность;
- небольшие требования к оперативной памяти;
- инвариант к диагональному масштабированию градиентов;
- подходит для нестационарных целей;
- гиперпараметры имеют наглядную интерпретацию и обычно требуют небольшой настройки.

Основным свойством RMSProp является отсутствие ограничений по сумме прошлых градиентов, но имеется большая зависимость от градиентов последних шагов. За счет этого происходит уменьшение вертикального смещения значений в сторону нуля, на основе этого функция RMSProp принимает среднее значение градиентов.

На основе описанных выше оптимизаторов, их плюсов и недостатков можно сделать вывод, что оптимизатор Adam лучше всего подходит для решения данной задачи.

В виде аргумента для расчета ошибок, создаваемых в процессе работы нейронных сетей, лучше всего подходит среднеквадратичная ошибка (MSE).

Для обучения модели была использована функция потерь «meansquarederror» (MSE), которая измеряет разницу между предсказанными и истинными координатами точек. Кроме того, в модели используются метрики recall и precision для оценки качества обнаружения точек.

Среднеквадратичная ошибка – среднеквадратичная разница между значением, наблюдаемым в статическом исследовании, и значениями, предсказанными моделью.

При сравнении наблюдений с прогнозируемыми значениями необходимо возвести различия в квадрат, поскольку некоторые значения данных будут больше прогноза, а другие – меньше. Учитывая, что наблюдения с такой же вероятностью будут больше прогнозируемых значений, как и меньше, различия будут равны нулю. Возведение этих различий в квадрат устраняет эту ситуацию:

$$precision = \frac{t_p}{t_p + f_p};$$

$$recall = \frac{t_p}{t_p + f_n},$$

где t_p – число верно определенных значений; f_p – число ошибочно определенных значений; f_n – число правильно нераспознанных значений.

Для обучения модели используется генератор данных, который позволяет эффективно обрабатывать большие объемы данных. Обучающая выборка разделена на обучающую и валидационную в соотношении 70:30. Генератор данных загружает изображения

и соответствующие им аннотации из XML-файлов. Изображения предварительно обрабатываются, изменяются размеры до 512×512 и нормализуются. Аннотации представляются в виде координат точек на изображении.

Концепция нейрона в нейронных сетях представляет собой основной строительный блок для создания сложных вычислительных моделей [5–7].

Математически нейрон можно описать как функцию, которая принимает входные вектора значений x , их веса w и смещение b . Выходом функции является скалярное произведение входных значений вида:

$$z = \langle w, x \rangle = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n,$$

где z – выход нейрона; w_1, w_2, \dots, w_n – веса, которые соответствуют входным значениям; x_1, x_2, \dots, x_n – входные значения, которые представляют собой активации (результаты) нейронов из предыдущего слоя; b – смещение, которое представляет собой некоторое смещение относительно суммы входов и весов.

Что происходит в нейроне:

1. Каждое входное значение x_n умножается на соответствующий вес w_n .
2. Произведения $w_n \cdot x_n$ суммируются.
3. К этой сумме добавляется смещение b .

После получения линейной комбинации результат передается через активационную функцию. Активационная функция определяет выход нейрона после применения нелинейности. Популярными активационными функциями являются: сигмоида, гиперболический тангенс (\tanh), ReLU и др. Рассмотрим некоторые из представленных функций подробнее:

1. Линейная функция $z_n = w_n \times x_n$.
2. Пороговая функция $z = \begin{cases} 1, & x > 0; \\ -1, & x \leq 0. \end{cases}$
3. Сигмоидная функция $z = \frac{1}{1 + e^{-wx}}$.
4. $\tanh z = \tanh\left(\frac{wx}{2}\right)$.
5. ReLU $z = \max(0, x)$.

Структура модели нейронной сети для распознавания точек лица на фотоизображениях имеет следующую последовательность действий [8]:

1. Считывание фотоизображений и приведение их к матричному виду:

$$F = (f_{ij})_{i,j=1,512}.$$

2. Нормализация данных заключается в делении каждого элемента матрицы на 255, что приводит ее к виду:

$$G = \left(\left[\frac{f_{ij}}{255} \right] \right)_{i,j=1,512},$$

где $[\bullet]$ – округление до целого (для краткости обозначим аргумент функции жирной точкой).

3. Процесс свертки заключается в выборе размерности ядра свертки и свертке данных с помощью этого ядра. Формула для поиска размерности ядра свертки имеет вид:

$$(w, h) = (mw - kw + 1, mh - kh + 1),$$

где (w, h) – вычисляемый размер конечного фото после прохождения через один сверточный слой; mw – предыдущая ширина фото; mh – предыдущая высота фото; kw – ширина ядра свертки; kh – высота ядра свертки.

4. Активацию данного слоя можно реализовать при помощи функции ReLU, математически описываемой в виде формулы:

$$\text{ReLU}(G) = \max(0, g),$$

где g – сигнал, подаваемый в нейроне.

То есть, если сигнал, подаваемый на вход, больше нуля, то функция оставит его без изменения, а если сигнал меньше или равен нулю, то функция примет значение 0.

Таким образом, зависимость будет иметь следующий вид:

$$\begin{aligned} g > 0 &\Rightarrow \text{ReLU}(g) = g \\ g \leq 0 &\Rightarrow \text{ReLU}(g) = 0 \end{aligned}$$

В дальнейшем в зависимости от полученных результатов внутри нейрона поступают откорректированные с помощью функции $\text{ReLU}(G)$ данные.

5. Полученное ядро скользит по изображению, сжимая его и уменьшая его размер. Значение каждого элемента матрицы можно рассчитать по формуле:

$$W = (w_{kl})_{k,l=1,512}, w_{kl} = \sum_{i=1}^3 \sum_{j=1}^3 H_{i+k-1, j+l-1} \cdot s_{i,j},$$

где f – исходная матрица изображения; s – ядро свертки.

6. После сверточного слоя идет слой BatchNormalization (пакетная нормализация), где в качестве аргумента передается вектор:

$$\begin{aligned} \bar{x} &= (w_1, w_2, \dots, w_i); \\ i &= \text{const} \end{aligned}$$

где \bar{x} – матрица, получаемая на выходе со сверточного слоя; x_i – i -й элемент матрицы.

7. Формула нормализации имеет вид:

$$\begin{aligned} \hat{y}_i &= \frac{w_i - \mu}{\sqrt{\sigma^2 + \theta}}; \\ \mu &= \frac{1}{m} \sum_{i=1}^m w_i; \\ \sigma^2 &= \frac{1}{m} \sum_{i=1}^m (w_i - \mu)^2, \end{aligned}$$

где μ – математическое ожидание; x_i – i -й элемент матрицы; σ^2 – дисперсия пакета.

8. Нормализация \hat{y}_i может изменить представление данных внутри слоя, во избежание данной проблемы следует ввести два дополнительных параметра:

- параметр сжатия γ ;
- параметр сдвига β .

Общая формула с применениями данных параметров коррекции будет иметь вид:

$$a_i = \gamma \hat{y}_i + \beta.$$

Если $\beta = \mu$ и $\gamma = \sqrt{\sigma^2 + \theta}$, то $a_i = w_i$, следовательно, они будут тождественны.

9. Далее идет слой MaxPooling2D, который на вход принимает матрицу A , в качестве параметра слоя прописывается значение фильтра:

$$A = (a_{i,j})_{i,j}.$$

MaxPooling2D слой начинает работать с первого элемента матрицы:

$$b_{i,j} = \begin{pmatrix} a_{2i-1,2j-1} & a_{2i-1,2j} \\ a_{2i,2j-1} & a_{2i,2j} \end{pmatrix}_{i,j=1,\log_2 n};$$

$$l_{i,j} = \max(b_{i,j});$$

$$L = (l_{i,j})_{i,j}.$$

10. После неоднократного прохождения матрицы изображения через Convolution2D, BatchNormalization и MaxPooling слоев данные попадают на следующую немаловажную часть архитектуры нейронной сети – слой Flatten.

Слой Flatten используется для преобразования многомерных данных матрицы изображения в одномерный вектор. Данные действия помогают подготовить данные к передаче их на выходной слой Dense. Параметрами данного слоя будет `batch_size` – размер пакета данных для обучения, `height` – высота входной матрицы, `width` – ширина входной матрицы, `channels` – число цветовой канальности матрицы фото.

Математическое представление данного слоя имеет следующий обобщенный вид:

$$\bar{M}(L) = \text{reshape}(l, (b, f_size)), f_size = height \cdot width \cdot channels,$$

где L – входная матрица фото; *reshape* – формула изменения размерности:

$$\text{reshape}(l, (b, f_size)) = [l_{1,1}, l_{1,2}, \dots, l_{1,n}, l_{2,1}, l_{2,2}, \dots, l_{2,n}, \dots, l_{n,1}, l_{n,2}, \dots, l_{n,n}].$$

11. Завершающим слоем архитектуры нейронной сети является слой Dense, который представляет из себя полносвязный слой, в котором нейроны попарно связаны с предыдущим слоем:

$$O = k\bar{M} + \bar{B},$$

где O – выход слоя; $k = \langle n, m \rangle$ – матрица весов слоя (n – количество входных признаков; m – число нейронов в слое); \bar{M} – подаваемый на слой вектор; \bar{B} – смещение размерности m .

Результаты исследования и их обсуждение

Для обучения и оценки модели был использован предразмеченный набор данных лиц, содержащий изображения лиц с размеченными на них точками [9, 10]. Обучение модели проводилось на графическом процессоре с использованием библиотеки TensorFlow (рис. 1).

После обучения модели она была протестирована на тестовом наборе изображений, представленных в формате XML. На этапе предобработки изображения были изменены размеры до 512x512 пикселей для соответствия архитектуре модели.

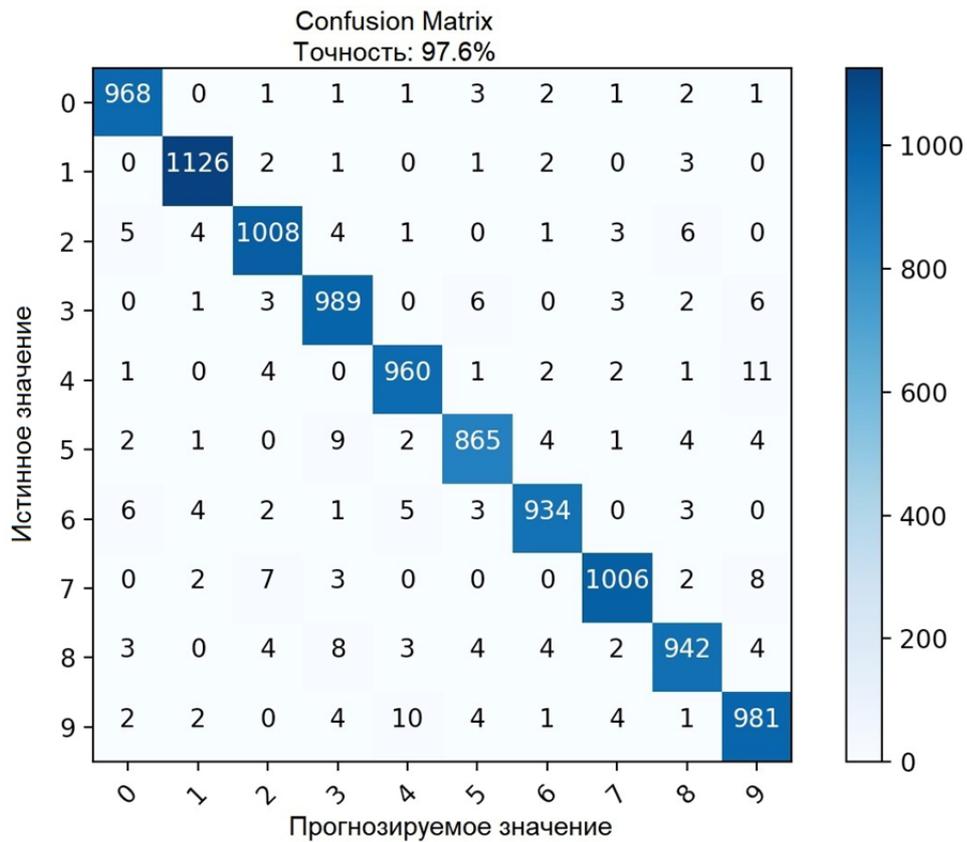


Рис. 1. Графический процессор

Результаты экспериментов показали, что обученная модель успешно обнаруживает точки на изображениях, достигая высокой точности и полноты. Кроме того, модель демонстрирует хорошую устойчивость к разнообразным условиям освещения и изменениям масштаба изображений.

Предложенная модель может быть использована в различных областях, где требуется обнаружение лицевых точек на фотографиях (рис. 2). Возможно применение данной модели в системах обнаружения, классификации преступников по фотоизображениям, в системах распознавания лиц с повышенной точностью, использование модели с разными программами, объединяя их в единый комплекс программного обеспечения.

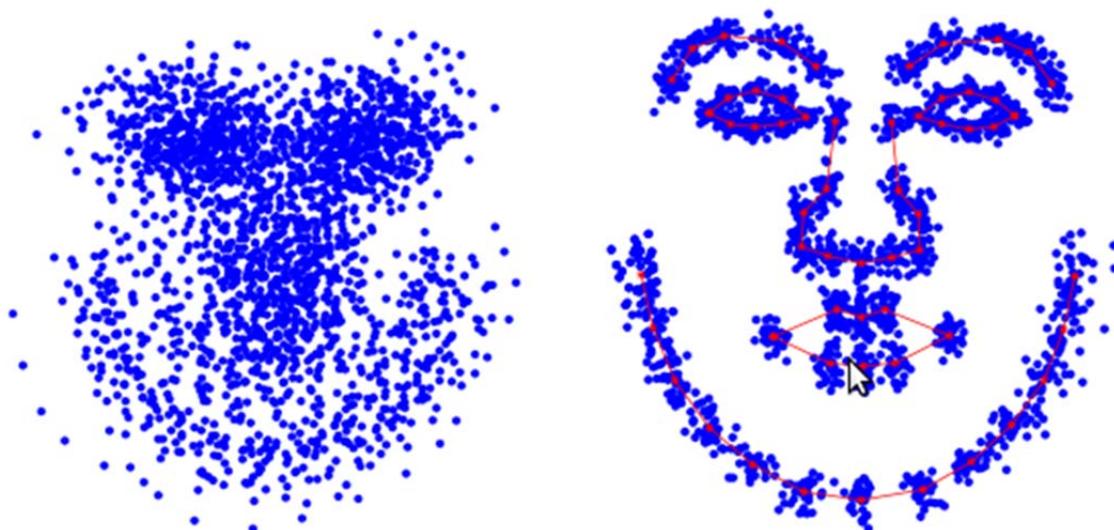


Рис. 2. Лицевые точки на фотографии

Заключение

В статье была представлена глубокая нейронная сеть, разработанная с использованием библиотеки Keras, для обнаружения точек на изображениях [11]. Модель успешно обучалась на заранее размеченном наборе данных лиц и успешно применялась к реальным изображениям в формате .jpg, .png, .jpeg. Полученные результаты подтверждают высокую эффективность предложенной модели в задаче обнаружения точек на изображениях, что делает её перспективным инструментом в различных областях, требующих точного обнаружения объектов на изображениях. Дальнейшие исследования могут включать в себя расширение модели и увеличение числа распознаваемых ею точек.

Список источников

1. Тынченко С.В., Путилин В.О., Овсянникова А.К. Сравнение алгоритмов обнаружения и локализации лица на изображении // Молодой ученый. Красноярск, 2018. С. 105–108.
2. Матричный метод как способ определения антропометрических точек лица / А.И. Ситников [и др.] // Актуальные проблемы деятельности подразделений УИС: материалы Всерос. науч.-практ. конф. Воронеж, 2022. С. 193–194.
3. Мельников Н.А. Некоторые особенности определения антропологических характеристик лица на основе глубокого машинного обучения // Актуальные вопросы эксплуатации систем охраны и защищенных телекоммуникационных систем: материалы Всерос. науч.-практ. конф. Воронеж, 2022. С. 278–280.
4. Толстых А.А., Голубинский А.Н. Распознавание объектов на телевизионных изображениях с использованием аппарата сверточных нейронных сетей // Вестник Воронежского института МВД России: сб. науч. трудов. Воронеж, 2017. № 1. С. 71–81.
5. Гафаров Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие. Казань: Изд-во Казанского ун-та, 2018. 121 с.
6. Толстых А.А., Голубинский А.Н. Выбор архитектуры искусственной нейронной сети на основе сравнения эффективности методов распознавания изображений // Вестник Воронежского института МВД России: сб. науч. трудов. Воронеж, 2018. № 1. С. 27–37.
7. Бутырский Е.Ю., Матвеев А.В. Математическое моделирование систем и процессов. СПб.: Информ. изд. учеб.-науч. центр «Стратегия будущего», 2022. 733 с. DOI: 10.37468/book_011222. EDN CCRIRT.
8. Толстых А.А., Голубинский А.Н. Сравнение эффективности методов изменения скорости обучения искусственных нейронных сетей в различных задачах классификации // Международный научно-исследовательский журнал: сб. науч. трудов. 2022. № 7–1 (121). С. 102–106.
9. Редькин В.С., Ситников А.И. Использование методов распознавания образов при видео фиксации для повышения безопасности объектов // Актуальные проблемы деятельности подразделений УИС: материалы Всерос. науч.-практ. конф. Воронеж, 2021. С. 233–234.
10. Ярышев С.Н., Рыжова В.А. Технологии глубокого обучения и нейронных сетей в задачах видеоанализа. СПб.: Ун-т ИТМО, 2022. 82 с.
11. Мельников Н.А., Ситников В.А., Никитенко В.А. Строение сверточной нейронной сети для определения контрольных точек лица человека // Технологии информационного общества: сб. трудов XVII Междунар. отрасл. науч.-техн. конф. М., 2023. С. 127–128.

References

1. Tynchenko S.V., Putilin V.O., Ovsyannikova A.K. Sravnenie algoritmov obnaruzheniya i lokalizacii lica na izobrazhenii // Molodoj uchenyj. Krasnoyarsk, 2018. S. 105–108.
2. Matrichnyj metod kak sposob opredeleniya antropometricheskikh toчек lica / A.I. Sitnikov [i dr.] // Aktual'nye problemy deyatel'nosti podrazdelenij UIS: materialy Vseros. nauch.-prakt. konf. Voronezh, 2022. S. 193–194.
3. Mel'nikov N.A. Nekotorye osobennosti opredeleniya antropologicheskikh harakteristik lica na osnove glubokogo mashinnogo obucheniya // Aktual'nye voprosy ekspluatacii sistem ohrany i zashchishchennyh telekommunikacionnyh sistem: materialy Vseros. nauch.-prakt. konf. Voronezh, 2022. S. 278–280.
4. Tolstyh A.A., Golubinskij A.N. Raspoznavanie ob"ektov na televizionnyh izobrazheniyah s ispol'zovaniem apparata svertochnyh nejronnyh setej // Vestnik Voronezhskogo instituta MVD Rossii: sb. nauch. trudov. Voronezh, 2017. № 1. С. 71–81.
5. Gafarov F.M., Galimyanov A.F. Iskusstvennye nejronnye seti i prilozheniya: ucheb. posobie. Kazan': Izd-vo Kazanskogo un-ta, 2018 121 s.
6. Tolstyh A.A., Golubinskij A.N. Vybor arhitektury iskusstvennoj nejronnoj seti na osnove sravneniya effektivnosti metodov raspoznavaniya izobrazhenij // Vestnik Voronezhskogo instituta MVD Rossii: sb. nauch. trudov. Voronezh, 2018. № 1. С. 27–37.
7. Butyrskij E.Yu., Matveev A.V. Matematicheskoe modelirovanie sistem i processov. SPb.: Inform. izd. ucheb.-nauch. centr «Strategiya budushchego», 2022. 733 s. DOI: 10.37468/book_011222. EDN CCRIRT.
8. Tolstyh A.A., Golubinskij A.N. Sravnenie effektivnosti metodov izmeneniya skorosti obucheniya iskusstvennyh nejronnyh setej v razlichnyh zadachah klassifikacii // Mezhdunarodnyj nauchno-issledovatel'skij zhurnal: sb. nauch. trudov. 2022. № 7–1 (121). С. 102–106.
9. Red'kin V.S., Sitnikov A.I. Ispol'zovanie metodov raspoznavaniya obrazov pri video fiksacii dlya povysheniya bezopasnosti ob"ektov // Aktual'nye problemy deyatel'nosti podrazdelenij UIS: materialy Vseros. nauch.-prakt. konf. Voronezh, 2021. S. 233–234.
10. Yaryshev S.N., Ryzhova V.A. Tekhnologii glubokogo obucheniya i nejronnyh setej v zadachah videoanaliza. SPb.: Un-t ITMO, 2022. 82 s.
11. Mel'nikov N.A., Sitnikov V.A., Nikitenko V.A. Stroenie svertochnoj nejronnoj seti dlya opredeleniya kontrol'nyh toчек lica cheloveka // Tekhnologii informacionnogo obshchestva: sb. trudov XVII Mezhdunar. otrasl. nauch.-tekhn. konf. M., 2023. S. 127–128.

Информация о статье:

Статья поступила в редакцию: 10.01.2024; одобрена после рецензирования: 13.02.2024;
принята к публикации: 16.02.2024

Information about the article:

The article was submitted to the editorial office: 10.01.2024; approved after review: 13.02.2024;
accepted for publication: 16.02.2024

Сведения об авторах:

Ситников Александр Иванович, доцент кафедры физики и радиоэлектроники Воронежского института МВД России (394065, г. Воронеж, пр. Патриотов, д. 53), кандидат технических наук, e-mail: sitnikov_74@list.ru, <https://orcid.org/0000-0003-3989-3144>, SPIN-код: 7609-7132

Никитенко Виталий Алексеевич, преподаватель кафедры математики и моделирования систем Воронежского института МВД России (394065, г. Воронеж, пр. Патриотов, д. 53), e-mail: vitalijnikitenko82043@gmail.com, <https://orcid.org/0009-0006-1948-3817>, SPIN-код: 7425-2863

Мельников Никита Алексеевич, курсант радиотехнического факультета Воронежского института МВД России (394065, г. Воронеж, пр. Патриотов, д. 53), e-mail: nick12541010@gmail.com, <https://orcid.org/0009-0000-2105-9245>

Гладких Татьяна Васильевна, доцент кафедры информационных технологий, моделирования и управления Воронежского государственного университета инженерных технологий (394036, г. Воронеж, пр. Революции, д. 19), кандидат технических наук, e-mail: gtv1311@mail.ru, <https://orcid.org/0000-0002-1935-8753>, SPIN-код: 4891-2710

Information about authors:

Sitnikov Alexander I., associate professor of the department of physics and radioelectronics of the Voronezh institute of the Ministry of internal affairs of Russia (394065, Voronezh, Patriotov ave., 53), candidate of technical sciences, e-mail: sitnikov_74@list.ru, <https://orcid.org/0000-0003-3989-3144>, SPIN: 7609-7132

Nikitenko Vitaly A., lecturer at the department of mathematics and system modeling at the Voronezh institute of the Ministry of internal affairs of Russia (394065, Voronezh, Patriotov ave., 53), e-mail: vitalijnikitenko82043@gmail.com, <https://orcid.org/0009-0006-1948-3817>, SPIN: 7425-2863

Melnikov Nikita A., cadet of the radio engineering faculty of the Voronezh institute of the Ministry of internal affairs of Russia (394065, Voronezh, Patriotov ave., 53), e-mail: nick12541010@gmail.com, <https://orcid.org/0009-0000-2105-9245>

Gladkikh Tatyana V., associate professor of the department of information technology, modeling and management of the Voronezh state university of engineering technologies (394036, Voronezh, Revolyutsii ave., 19), candidate of technical sciences, e-mail: gtv1311@mail.ru, <https://orcid.org/0000-0002-1935-8753>, SPIN: 4891-2710