УДК 004.89

Лаврентьев С.А.
Lavrentyev S.A.
студент магистратуры
graduate student
2 курс, факультет ИБ
2 course Faculty IB
МФ МГТУ им. Н. Э. Баумана
MF MSTU them. N.E. Bauman
Россия, г. Москва
Russia, Moscow
Научный руководитель: Коннова Н.С.
scientific advisor Konnova N.S.
доцент, кандидат технических наук
Associate Professor, Candidate of Engineering Sciences

СОЗДАНИЕ СИСТЕМЫ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ ПО ГЕОМЕТРИИ ЛИЦА НА ОСНОВЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

CREATING A BIOMETRIC IDENTIFICATION SYSTEM BASED ON FACIAL GEOMETRY BASED ON MACHINE LEARNING ALGORITHMS

Аннотация*: В этой статье описано создание системы биометрической идентификации по геометрии лица на основе алгоритмов машинного обучения. При этом был учтен ряд факторов, как язык программирования, использованные библиотеки. В итоге было проведено тестирование системы и отображены его результаты.*

Annotation*: This article describes the creation of a biometric identification system based on facial geometry based on machine learning*

algorithms. A number of factors were taken into account, such as the programming language and the libraries used. As a result, the system was tested and its results were displayed.

***Ключевые слова**: нейронная сеть, биометрия, идентификация по геметрии лица.*

***Key words**: neural network, biometrics, the identity using face geometry*.

В современном мире технологии оказывают огромное влияние на повседневную жизнь каждого человека. Они делают ее удобней, комфортней и легче. Уже не является чем то новым оплачивать покупки бесконтактным способом с помощью телефона, или заказать практически любой товар через сеть Интернет и оплатить его. Более того, в метро вводится бесконтактный пропуск, а в Америке запускаются магазины без наличия на выходе терминалов оплаты за покупки. Все это достигается за счет использовании систем биометрической идентификации.

Исходя требований к современным системам распознавания образов, необходимо создать систему, позволяющую распознавать образы в изображениях, при этом дополнительно решать ряд задач:

- использование обобщенной информации при обучении сверточной НС;
- обеспечение устойчивости сторонним шумам.

При этом система должна удовлетворять следующим требованиям:

- кроссплатформенность;
- интуитивность во время эксплуатации;
- работа с изображениями любого расширения;

приведение выходных данных к виду, удобному для дальнейшей обработки.

Таким образом, система должна включать в себя следующие компоненты:

- сверточная нейронная сеть;

- модуль для нормализации входных данных;
- модуль для связи с пользователем.

В современном мире много высокоринетированных языков программирования. Однако для написания программ могут быть использованы готовые модули (библиотеки или фрейворки), необходимые для упрощения процесса программирования. Чем популярнее язык программирования среди разработчиков ввиду к-либо особенностей языков программирования, тем больше готовых модулей и замечаний можно найти. Однако для создания системы распознавания образов целесообразно использовать Pyhon – язык программирования, не привязанный к конкретной системе или оборудованию.

Кроме того, наиболее важной задачей при создании комплекса является выбор библиотеки, поскольку это позволяет упростить или усложнить процесс создания программы.

На рисунке 1 представлены наиболее часто используемые библиотеки, подходящие для решения задачи распознавания образов по биометрии лица.

Library	Rank	Overall	Github	Stack Overflow	Google Results
tensorflow	1	10.87	4.25	4.37	2.24
keras	2	1.93	0.61	0.83	0.48
caffe	3	1.86	1.00	0.30	0.55
theano	4	0.76	-0.16	0.36	0.55
pytorch	5	0.48	-0.20	-0.30	0.98
sonnet	6	0.43	-0.33	-0.36	1.12
mxnet	7	0.10	0.12	-0.31	0.28
torch	8	0.01	-0.15	-0.01	0.17
cntk	9	-0.02	0.10	-0.28	0.17
dlib	10	-0.60	-0.40	-0.22	0.02
caffe2	11	-0.67	-0.27	-0.36	-0.04
chainer	12	-0.70	-0.40	-0.23	-0.07
paddlepaddle	13	-0.83	-0.27	-0.37	-0.20
deeplearning4j	14	-0.89	-0.06	-0.32	-0.51
lasagne	15	-1.11	-0.38	-0.29	-0.44
bigdl	16	-1.13	-0.46	-0.37	-0.30
dynet	17	-1.25	-0.47	-0.37	-0.42
apache singa	18	-1.34	-0.50	-0.37	-0.47
nvidia digits	19	-1.39	-0.41	-0.35	-0.64
matconvnet	20	-1.41	-0.49	-0.35	-0.58
tflearn	21	-1.45	-0.23	-0.28	-0.94
nervana neon	22	-1.65	-0.39	-0.37	-0.89
opennn	23	-1.97	-0.53	-0.37	-1.07

Рисунок 1 – Сводный анализ распространения библиотек [1]

Каждый модуль комплекса работает автономно, поэтому целесообразно все части создавать отдельно. Однако при использовании сервиса Google Drive они могут взаимодействовать друг с другом посредством передачи промежуточных результатов (рисунок 2).

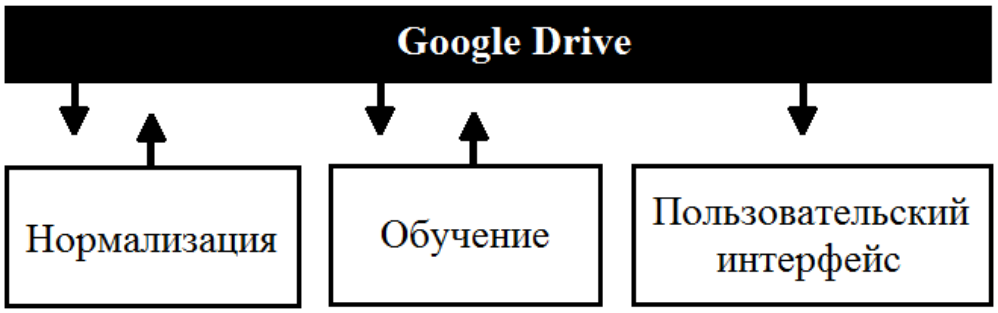

Рисунок 2 – Блок-схема алгоритма поиска распознавания образов на изображениях

Первый модуль комплекса отвечает за нормализацию входных данных, прежде чем они поступят на второй модуль. Поскольку предоставляемый Google Colab контейнер изначально пустой, то для работы первого модуля необходимо подключить сервис Google Drive, чтобы потом оттуда скачать выбранный набор данных. После этого необходимо подключить библиотеки для работы с изображениями и нормализации изображений. После этого обращаемся к каждому изображению и производится поиск лица. Если оно найдено, детектируется и отправляется на нормализацию. После того как изображение будет нормализовано (было повернуто в плоскости относительно главных осей, если образ был смещен, то проведены аффинные преобразования, обрезано от остального изображения и измен размер), тогда алгоритм переходит к обработке нового изображения.

Второй модуль комплекса получает данные из первого с помощью сервиса Google Drive. После этого входные данные поступают в генератор входных данных ImageDataGenerator, который преобразует значения всех изображений, чтобы они входили в интервал [0;1]. Затем выбираются каталоги наборов данных для метода flow_from_derectory, позволяющего к ним обратится. В этом методе настраиваются такие параметры, как: размер выборки (число изображений, которое сразу будет отправляться в НС), размер изображений и параметр, отвечающий за номер класса. Затем задаются количество эпох обучения и размер выборки и количество классов в каталоге.

После этого создается архитектура сети: два каскада идущих друг за другом сверточного и субкредитирующего слоя, два каскада сверточного слоя и еще один каскад из сверочного и подвыборочного слоя. Все они необходимы для выделения важных признаков на изображении. После этого следует полносвязанная часть, состоящая из слоя,

преобразующего двухмерный вектор в одномерный и два полносвязных слоя для конечной классификации.

После того, как начальные данные для настройки нейронной сети были определены, компилируется сеть с параметрами, среди которых функция потерь, значение которой в ходе работы должно уменьшаться. Так же эффективность созданной НС в процессе тестирования определяется по следующей формуле [2]:

$$acc = \frac{N_{true}}{N_{all}}. \quad (1)$$

Здесь acc – точность распознания образов, N_{true} – число правильно распознанных графических объектов, N_{all} – общее число изображений на этапе тестирования.

Затем необходимо выбрать метрики для оценивания эффективности модели.

После всех выбранных параметров создается генератор для запуска НС с заданными ранее параметрами: каталоги, в которых хранятся образы, количество эпох и число обращения генератора к классу.

После обучения сверточной НС, а также проверки результатов на достоверность сохраняются параметры сети и предаются на сервис Google Drive.

Сразу после запуска третьего модуля комплекса необходимо потребовать от пользователя загрузить необходимые для работы программы модули из сервиса Google Drive. После распаковки модели пользователь сможет загрузить свое изображение и сопоставить образ с существующими в модели.

Тестирование системы можно произвести в два этапа – отладочное и функциональное. При этом в первом случае для обучения и работы НС

можно использовать небольшой объем начальной выборки. При этом функциональное тестирование подразумевает работу системы на уровне пользователя.

Основная метрика оценки результатов тестирования – точность распознания образов, в зависимости от которого производится определение объекта одному из классов или же информирование, что того нет в базе. Соответственно, чем выше точность, тем меньше ошибка второго, но при этом растет ошибка первого рода.

В результате проведения тестирования (рисунок 3) были получены следующие результаты:

```
Train on  2500 samples
Epoch 1080/1084
25/25 [==============================] - 20s 53ms/step - loss: 0.3302 - accuracy: 0.8778
Epoch 1081/1084
25/25 [==============================] - 22s 52ms/step - loss: 0.4153 - accuracy: 0.8974
Epoch 1082/1084
25/25 [==============================] - 20s 52ms/step - loss: 0.5744 - accuracy: 0.8000
Epoch 1083/1084
25/25 [==============================] - 23s 53ms/step - loss: 0.4206 - accuracy: 0.8590
Epoch 1084/1084
25/25 [==============================] - 18s 55ms/step - loss: 0.2127 - accuracy: 0.9222
```

Рисунок 3 – Результаты тестирования системы биометрической идентификации

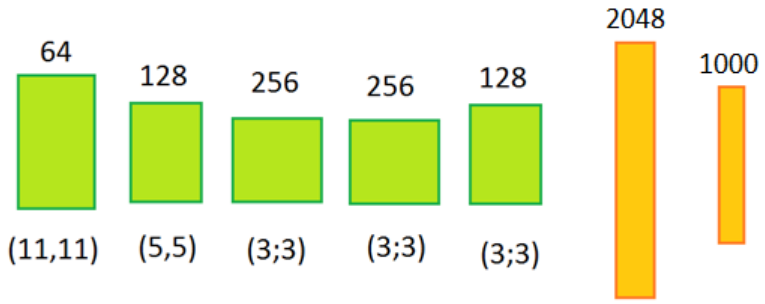

Рисунок 4 – сводная информация о структуре сети

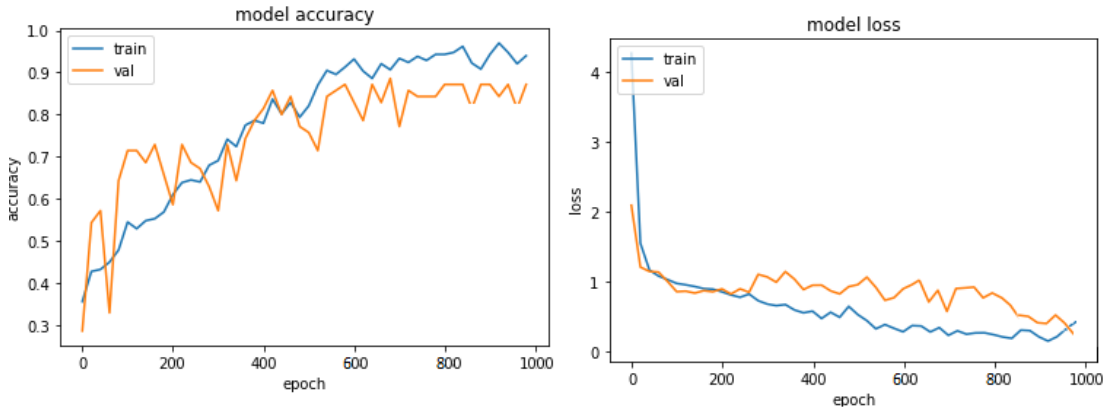

Рисунок 5 – графики роста точности на тренировочном и проверочном наборах и графики падения значений функции потерь

Проведя несколько запусков системы при разных обучающих выборках от 1000 до 8000 образов, проанализировав находящиеся в системе данные, а также сравнив их с теми, что поданы на вход системы, были получены следующие результаты. Из 87,34% человек из 100% поданных на вход образов распознались системой. При этом только 5,31% людей, не входящих в базу, определены в один из классов, когда вероятность ошибок второго рода составляет 6,12%.

При этом вероятность ошибок первого рода должна быть не больше 5%. Из этого следует, что система в дальнейшем требует доработки.

Использованные источники

1. Michael Li // Ranking Popular Deep Learning Libraries for Data Science [Электронный документ]. URL: https://blog.thedataincubator.com/2017/10/ranking-popular-deep-learning-libraries-for-data-science/ (Дата посещения 07.02.2020).
2. Сайт 100byte.ru // Бартеньев О. В. Параметры, влияющие на эффективность нейронной сети, созданной средствами Keras [Электронный документ]. URL: http://www.100byte.ru/python/factors/factors.html (Дата посещения 07.02.2020).