

Д. В. Бруенков
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **А. С. Руденков**, канд. техн. наук, доцент

РАЗРАБОТКА ПРИЛОЖЕНИЯ СРЕДСТВАМИ PYTHON ДЛЯ ОРГАНИЗАЦИИ ДОСТУПА СОТРУДНИКОВ ПУТЕМ АУТЕНТИФИКАЦИИ ПО ГЕОМЕТРИИ ЛИЦА

В условиях стремительного развития цифровых технологий и постоянного роста требований к безопасности современные системы аутентификации становятся настоящей визитной карточкой инноваций. Актуальность задачи распознавания лиц не вызывает сомнений: именно с помощью нейросетевых решений, реализованных на языке Python, удаётся достигать поразительной точности идентификации, что особенно важно при организации доступа сотрудников в учреждениях. Идентификация – это сравнение образцов по схеме «1:N». Чтобы идентифицировать личность система сравнивает обнаруженное лицо со всеми известными ей и даёт ответ на вопрос «Известен ли этот человек системе?» [1].

Современные алгоритмы компьютерного зрения, начиная от классических методов обработки изображений и заканчивая передовыми нейронными сетями, успешно преодолевают ограничения традиционных систем контроля доступа. Проект, опирающийся на принципы LBPH и использующий библиотеки OpenCV, TensorFlow и Keras, демонстрирует, как даже при неблагоприятных условиях (изменения освещённости, частичное закрытие лица) возможно добиться высокой точности распознавания. Эта методика не только повышает уровень безопасности, но и открывает новые горизонты в области удобства использования систем аутентификации.

Один из важных инструментов в нашей системе – метод каскадов Хаара (рисунок 1), который превращает обычную камеру в «острый взгляд» охранника. Эта технология появилась еще в 2000-х и до сих пор прижилась благодаря скорости и эффективности. Как это работает? Алгоритм смотрит на изображение через серию прямоугольных фильтров, выделяя контрастные области – например, тёмные глаза на светлом лице или линии носа. Каждый фильтр отбрасывает ненужные части, а оставшиеся участки проверяются более сложными фильтрами. В нашей разработке метод Хаара стал первым шагом в обработке: он быстро находит лица даже в толпе и экономит до 70 % вычислительных ресурсов. Это позволяет системе работать в реальном времени на обычных компьютерах, а совместная работа с нейросетями даёт возможность быстро проверять данные, сочетая скорость классического компьютерного зрения и возможности глубокого обучения.

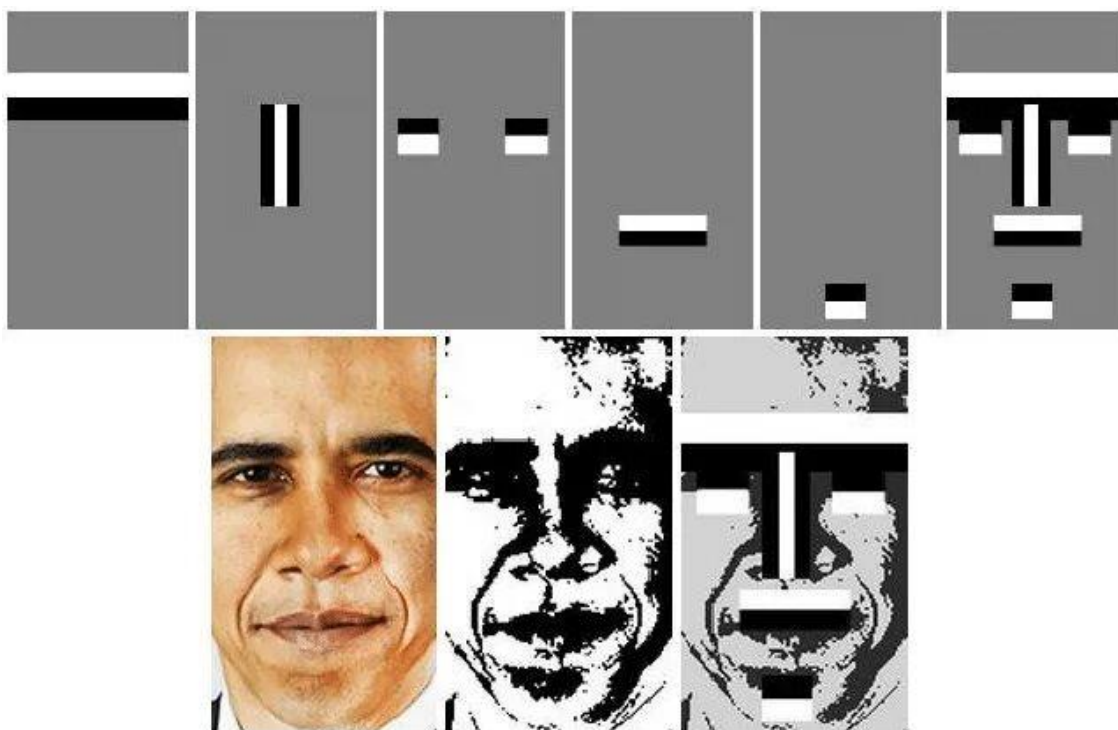


Рисунок 1 – Каскады Хаара, иллюстрирующие обнаружение лиц на фотографии с помощью прямоугольных областей

Наша система основана на библиотеке OpenCV (Open Source Computer Vision Library). OpenCV (от англ. Open Source Computer Vision Library) – это библиотека алгоритмов компьютерного зрения с открытым исходным кодом. Библиотека распространяется по лицензии BSD, следовательно, она может свободно использоваться в академических и коммерческих целях [2]. OpenCV предлагает уже готовые алгоритмы для работы с изображениями, распознавания объектов и видео. Это делает её отличным выбором для задач, связанных с распознаванием лиц. В нашем приложении OpenCV отвечает за захват видео с камеры, обработку кадров и определение лиц с помощью каскадов Хаара. Благодаря своей производительности и совместимости с Python, нам удалось создать быструю и надежную систему аутентификации, работающую даже на слабых устройствах.

Система распознавания лиц построена по модульному принципу и включает три взаимосвязанных компонента. Первый модуль отвечает за создание датасета – он автоматически захватывает изображения с камеры, находит лица и формирует базу для обучения, сохраняя каждое изображение с уникальным идентификатором. Второй модуль занимается обучением модели с использованием метода LBPH (Local Binary Patterns Histograms), который анализирует текстуру лица и создает уникальные цифровые отпечатки. Третий модуль – это система идентификации, которая в реальном времени сравнивает лица с камеры с базой данных, принимая решение о доступе. Программный код обученной модели изображён на рисунке 2. Такая трёхэтапная архитектура делает систему гибкой: можно добавлять новых пользователей в датасет, переобучать модель для повышения точности, а алгоритм идентификации легко интегрировать с уже существующими системами контроля доступа.

```

import cv2
import os
path = os.path.dirname(os.path.abspath(__file__))
recognizer = cv2.face.LBPHFaceRecognizer_create()
trainer_path = path + r'/trainer/trainer.yml'
print(f"Путь к файлу тренера: {trainer_path}")
recognizer.read(trainer_path)
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcas-
cade_frontalface_default.xml")
cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
while True:
    ret, im = cam.read()
    if not ret:
        print("Не удалось захватить видео")
        break
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, scaleFactor=1.2, minNeigh-
bors=5, minSize=(100, 100), flags=cv2.CASCADE_SCALE_IMAGE)
    for (x, y, w, h) in faces:
        nbr_predicted, confidence = recognizer.predict(gray[y:y+h, x:x+w])
        if confidence < 50:
            if nbr_predicted == 1:
                name = 'Denis Bruenkov'
                color = (0, 255, 0)
            else:
                name = 'Unknown'
                color = (0, 0, 255)
        else:
            name = 'NoName'
            color = (0, 0, 255)
        cv2.rectangle(im, (x, y), (x+w, y+h), color, 2)
        cv2.putText(im, name, (x, y-10), font, 0.9, color, 2)
    cv2.imshow('Face recognition', im)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cam.release()
cv2.destroyAllWindows()

```

Рисунок 2 – Программный код обученной модели для идентификации лиц с камеры

Таким образом, Python-библиотеки позволяют сравнительно быстро и эффективно создать надежное приложение для анализа видеопотока с использованием нейросетей с целью оптимизации систем контроля и управления для организации доступа сотрудников путем аутентификации по геометрии лица.

Литература

1. Губарев, А. С. Выпускная квалификационная работа (бакалаврская работа) на тему «Разработка интеллектуальной системы распознавания лиц» / А. С. Губарев, Тольятти, 2022. – 12 с.
2. Прохоренок, Н. OpenCV и Java: обработка изображений и компьютерное зрение. / Н. Прохоренок, Санкт-Петербург : БХВ-Петербург, 2018. – 7 с.