

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по производственной практике**  
**Тема: Распознавание объектов на фотографиях с помощью**  
**библиотеки OpenCV**

Студент гр. 7381

Тарасенко Е.А.

Руководитель

Якубов Е.Н.

Санкт-Петербург

2020

**ЗАДАНИЕ**  
**НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ**

Студент: Тарасенко Е.А.

Группа: 7381

Тема практики: Распознавание объектов на фотографиях с помощью библиотеки OpenCV

Задание на практику:

Разработка приложения на языке Python с использованием библиотеки OpenCV для определения отсутствующих элементов на плате по ее фотографии.

Сроки прохождения практики: 29.06.2020 – 19.07.2020

Дата сдачи отчета: 20.07.2020

Дата защиты отчета: 20.07.2020

Студент

Тарасенко Е.А.

Руководитель

Якубов Е.Н.

## **АННОТАЦИЯ**

В ходе выполнения задания была разработана программа на языке Python 3.6 с использованием библиотеки OpenCV для определения мест на платах, в которых отсутствуют некоторые элементы. Полученное приложение при помощи алгоритмов компьютерного зрения определяет и отмечает на фотографии платы участок, где отсутствует необходимая деталь (конденсатор, входной или выходной разъем).

## **SUMMARY**

In the course of the assignment, a program in Python 3.6 was developed using the OpenCV library to determine places on boards that are missing some elements. The resulting application, using computer vision algorithms, identifies and marks on the photo of the board the area where the necessary part is missing (capacitor, input or output connector).

## СОДЕРЖАНИЕ

Введение	5
1. Постановка задачи	6
1.1. Компьютерное зрение и базовые возможности библиотеки OpenCV	6
1.2. Распознавание объектов на изображениях и каскад Хаара	7
2. Создание приложения	8
2.1. Подготовка датасетов и обучение каскадов	8
2.2. Возможности полученного приложения и оценка результатов	9
Заключение	11
Список использованных источников	12
Приложение А. Отзыв руководителя предприятия о прохождении практики	13

## **ВВЕДЕНИЕ**

В ходе прохождения производственной практики были приобретены некоторые навыки по работе с библиотекой OpenCV, содержащей функции для реализации и использования алгоритмов «компьютерного зрения». Перед выполнением работы был составлен план решения задачи распознавания объектов на фотографиях с помощью инструментов этой библиотеки. По итогам прохождения практики было создано приложение (на языке Python 3.6, с использованием среды разработки PyCharm), способное определять участки платы, на которых пропущен (не установлен) некоторый элемент.

# 1. ПОСТАНОВКА ЗАДАЧИ

## 1.1. Компьютерное зрение и базовые возможности библиотеки OpenCV

**Машинное зрение** — это научное направление в области искусственного интеллекта, в частности робототехники, и связанные с ним технологии получения изображений объектов реального мира, их обработки, и использования полученных данных для решения разного рода прикладных задач без участия (полного или частичного) человека.

**OpenCV** — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD.

OpenCV предоставляет (в числе прочих) необходимые инструменты для загрузки, обработки (кадрирования, поворота, отражения и изменения цветовой модели) и сохранения изображений.

Таким образом, сначала необходимо считать изображение платы, которую программа должна будет исследовать. Изображение представляется в виде матрицы пикселей, каждым элементом которой, в свою очередь, будет являться последовательность из чисел — кодировка цвета определенного пикселя. В данном случае существует особенность — по умолчанию OpenCV считывает изображение в цветовой модели «BGR» (а не в виде привычного «RGB»), поэтому, при необходимости, с помощью инструментов библиотеки можно перевести модель из BGR-формата в любой другой (RGB, CMYK или черно-белый «GRAY»). Итоговое изображение можно сохранить опять-таки с помощью стандартной функции OpenCV. Пример считывания, перевода в черно-белый формат и сохранения пользовательского изображения:

```
pic = cv2.imread('./image.jpg')
```

```
pic = cv2.cvtColor(pic, cv2.COLOR_BGR2GRAY)
cv2.imwrite('./exported_image.jpg', pic)
```

## **1.2. Распознавание объектов на изображениях и каскад Хаара**

**Признаки Хаара** — признаки цифрового изображения, используемые в распознавании образов. Признаки Хаара использовались в первом детекторе лиц, работающем в реальном времени. Для описания объекта с достаточной точностью признаки Хаара организовываются в каскадный классификатор.

Для более-менее работающего каскада необходимо иметь минимум 400-500 отрицательных изображений (в которых не присутствует интересующего объекта для распознавания) и 200-250 (примерно в 2 раза меньше) положительных. Положительную выборку рекомендуется формировать, помещая на каждое изображение по одному искомому объекту (хотя на фотографии одновременно может находиться и несколько). Также каскад Хаара работает с изображениями, имеющими только один цветовой канал, поэтому все изображения необходимо сначала переводить в черно-белый формат (если важен цвет, то будет уместно перед этим сменить цветовую модель, например, с RGB на CMYK).

Таким образом, для выполнения задачи по поиску объектов на изображениях необходим каскад Хаара, обученный минимум на паре сотен изображений, содержащими пример искомого объекта, и в 2 раза большем количестве изображений, которые этого объекта не содержат.

## **2. СОЗДАНИЕ ПРИЛОЖЕНИЯ**

### **2.1. Подготовка датасетов и обучение каскадов**

Для выполнения задания были подготовлены датасеты изображений для каждого из необходимых каскадов Хаара (для определения мест отсутствия трех разных деталей на платах). Так как в качестве рабочего материала имелось очень маленькое число фотографий как правильно собранных плат, так и плат, сборка которых не завершена, то выборка отрицательных примеров получена разбиением каждой фотографии правильной платы на несколько частей и переводом каждой из них в черно-белый формат. Положительные же выборки представляют собой наборы из черно-белых изображений мест пропусков разных деталей, полученных из соответствующих снимков.

Сначала требуется собрать вектора положительных примеров (их было по 240 штук на каждый каскад), после чего с учетом отрицательной выборки (по 400 изображений) запускается, непосредственно, обучение каскада. Эта процедура может занимать от нескольких часов до нескольких суток в зависимости от размеров датасетов. Операции по созданию вектора и обучения каскада совершаются с помощью утилит, поставляющихся вместе с OpenCV – `opencv_createsamples` и `opencv_traincascade`. На выходе получается для каждой интересующей детали свой вектор положительных примеров и папка, в которой хранится xml-файл каскада Хаара и несколько вспомогательных файлов, создающихся в процессе обучения модели.



## **2.2. Возможности полученного приложения и оценка результатов**

Полученное в ходе выполнения задания приложение принимает на вход путь к исследуемому изображению, после чего, переведя его в черно-белый формат, при помощи функции библиотеки OpenCV `detectMultiScale()` применяет к нему последовательно все три созданных ранее каскада. После этих операций программа получает координаты мест, где находятся предполагаемые пропуски деталей. Позже эти пропуски обозначаются прямоугольными областями разных цветов (в зависимости от пропущенного элемента платы). Изображение проверенной платы сохраняется на компьютер пользователя и выводится на экран. Если программа все обозначила (или не обозначила) правильно, можно согласиться на обновление датасетов. В таком случае приложение само дополнит выборки изображениями, полученными из данного конкретного примера для последующего обучения модели.

В ходе тестирования удалось подобрать такие параметры обучения и использования каскадов, при которых будет наблюдаться наименьшее число так называемых ложных срабатываний, однако из-за маленького количества уникальных фотографий, предоставленных предприятием, результат не идеален, но он, насколько это возможно, максимально приближен к желаемому. На рис. 1 приведен пример работы программы. Избавиться от неточностей в работе текущего приложения можно будет путем расширения выборки до 500 положительных примеров и 1000 отрицательных (с использованием большего числа уникальных снимков).

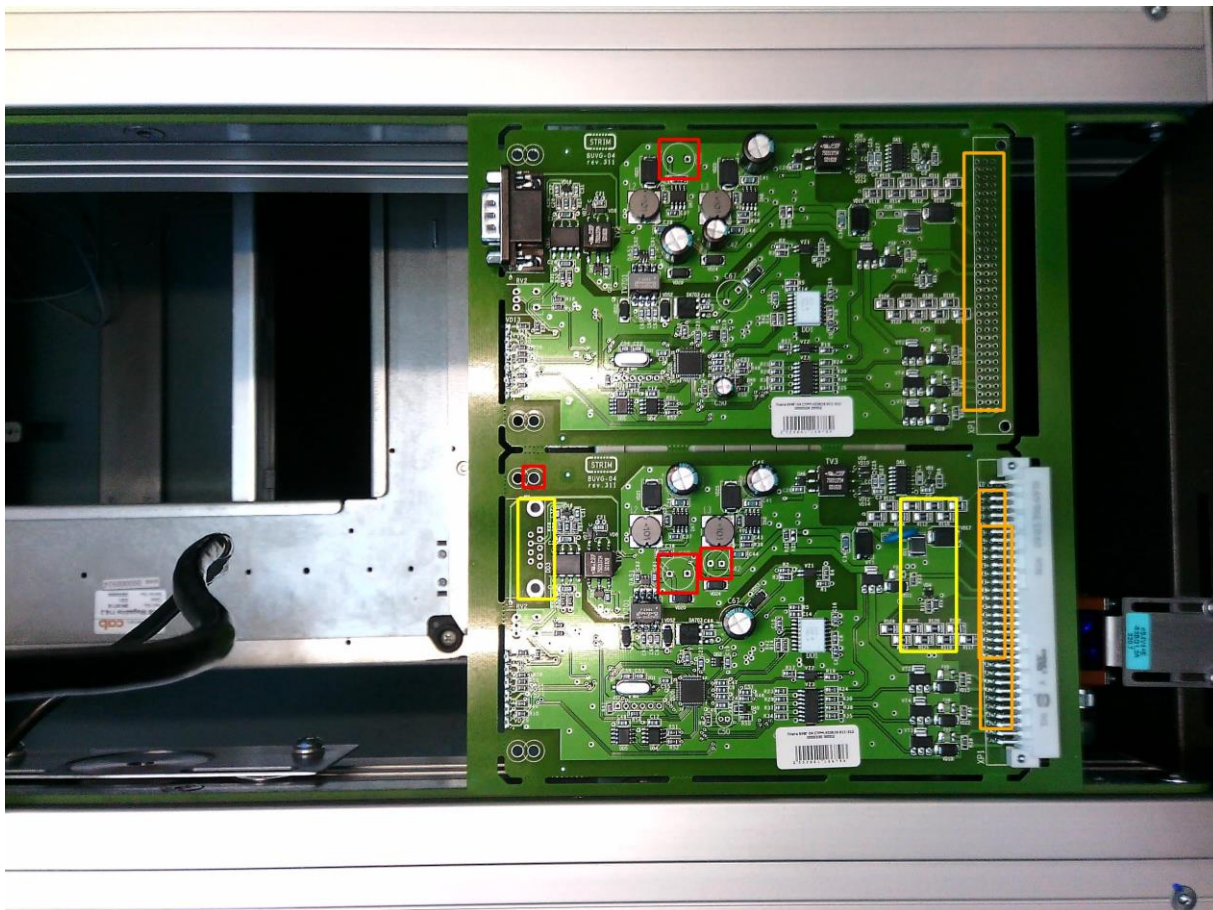


Рис. 1 – Пример работы программы

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения задачи производственной практики были закреплены навыки работы с языком Python. Разработано приложение, способное находить указанные объекты на фотографиях. В частности, требовалось определять места отсутствия элементов плат. Путем корректировки параметров обучения модели результат работы приложения был максимально приближен к желаемому, настолько, насколько это было возможно при имеющемся наборе оригинальных изображений, использовавшихся для формирования каскадов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Описание базовых функций для работы с изображениями OpenCV – Python // Tproger. URL: <https://tproger.ru/translations/opencv-python-guide/> (дата обращения: 01.07.2020).
2. Обучение OpenCV каскада Хаара // Habr. URL: <https://habr.com/ru/post/208092/> (дата обращения: 02.07.2020).
3. Общие сведения о компьютерном зрении // Habr. URL: <https://habr.com/ru/post/350918/> (дата обращения: 01.07.2020).

## ПРИЛОЖЕНИЕ А

### ОТЗЫВ РУКОВОДИТЕЛЯ ПРЕДПРИЯТИЯ О ПРОХОЖДЕНИИ ПРАКТИКИ

Студент 3 курса СПбГЭТУ «ЛЭТИ» Тарасенко Е.А. проходил производственную практику в Обществе с Дополнительной Ответственностью «Стрим» под моим руководством с 29 июня 2020 г. по 19 июля 2020 г.

В его обязанности входило:

1. Прохождение требуемого для решения поставленных задач обучения (знакомство с документацией, возможностями библиотеки);
2. Составление подробного плана, по которому будет решаться поставленная задача, разработка прототипа программы;
3. Разработка приложения, результат работы которого будет максимально приближен к желаемому.

За время практики студент проявил ответственность, исполнительность, знание языка программирования Python, за короткий промежуток времени изучил материал, необходимый для работы, показал высокий уровень обучаемости, внимательность к деталям, рациональный подход к решению поставленных в рамках проекта задач. Результат работы максимально приближен к желаемому.

Оценка:	Отлично
Контакт руководителя:	eugene@strim-tech.com
ФИО руководителя:	Якубов Евгений Николаевич
Дата:	20.07.2020