

Государственное бюджетное общеобразовательное учреждение  
«ГБОУ Гимназия №1409»

**Проектно-исследовательская работа**

***«Автоматизация процесса обработки результатов ЕГЭ в методических целях»***

Работу подготовил:  
ученик 10 класса Б  
ГБОУ школы №1409  
Шерри Георгий  
Научный руководитель:  
учитель информатики  
Четверов А. В.

Москва, 2019

# Содержание

Введение.....	3-4 стр.
Исследовательская часть.....	5-11 стр.
Глава I. Что такое компьютерное зрение.....	5-7 стр.
Глава II. Существующее программное обеспечение для систем компьютерного зрения .....	8-9 стр.
Глава III. OpenCV .....	10-11 стр.
Практическая часть .....	12-13 стр.
Итоги .....	14 стр.
Литература .....	15 стр.

# **Введение**

## **Проблема**

В 2019 году 719 тыс. человек сдали ЕГЭ, и каждый год — это число растёт. После каждого экзамена учителям приходят протоколы о том, как их ученики сдали экзамен по определённому предмету. Им приходится вручную подсчитать процентное выполнение каждого экзамена, чтобы понять, где ученики больше всего ошибаются. И если это профильные предметы, как биология или физика, где у одного учителя такой экзамен сдадут ~ 30 человек, то как быть с обязательными экзаменами, которые сдают все выпускники старших школ, математикой и русским языком.

## **Актуальность**

По результатам опроса учителей, многие из них сталкивались с обозначенной проблемой. Данные подсчеты занимают много времени и довольно трудны, но помогают в учебном процессе. Замена ручного процесса подсчета на автоматический, с помощью программы, поможет учителям сэкономить время и уже в начале учебного года организовать более эффективный учебный процесс по подготовке к ЕГЭ.

## **Гипотеза**

Подсчет процентов о выполнении каждого задания учениками по определённому предмету быстрее сделать вручную, чем с помощью программы, которая и является продуктом данного проекта.

## **Цель**

Изучить методы компьютерного зрения и с помощью одного из них создать программу для обработки результатов ЕГЭ.

## **Задачи**

1. Понять есть ли аналоги программы и если они есть, то чем новая программа будет лучше их.
2. Изучить методы компьютерного зрения и научиться их применять.
3. Написать программу на языке программирования Python, в качестве GUI используя VK API.
4. Протестировать программу и провести опрос об эффективности ее использования.

# **Исследовательская часть**

## **Глава I. Что такое компьютерное зрение**

**Компьютерное зрение** (иначе **машинное зрение**) — теория и технология создания машин, которые могут производить обнаружение, отслеживание и классификацию объектов.

Как научная дисциплина, компьютерное зрение относится к теории и технологии создания искусственных систем, которые получают информацию из изображений. Видеоданные могут быть представлены множеством форм, таких как видеопоследовательность, изображения с различных камер или трехмерными данными.<sup>[1]</sup>

Как технологическая дисциплина, компьютерное зрение стремится применить теории и модели компьютерного зрения к созданию систем компьютерного зрения. Примерами применения таких систем могут быть:

1. Системы управления процессами
2. Системы видеонаблюдения
3. Системы организации информации
4. Системы моделирования объектов или окружающей среды
5. Системы взаимодействия
6. Системы дополненной реальности

Компьютерное зрение также может быть описано как дополнение (но не обязательно противоположность) биологическому зрению. В биологии изучается зрительное восприятие человека и различных животных, в результате чего создаются модели работы таких систем в терминах физиологических процессов. Междисциплинарный обмен между биологическим и компьютерным зрением оказался весьма продуктивным для обеих научных областей.

### **Компоненты системы машинного зрения**

Типовое решение системы машинного зрения включает в себя несколько следующих компонентов:

1. Одна или несколько цифровых, или аналоговых камер с подходящей оптикой для получения изображений
2. Программное обеспечение для изготовления изображений для обработки
3. Процессор
4. Программное обеспечение машинного зрения, которое предоставляет инструменты для разработки отдельных приложений программного обеспечения

Программное обеспечение, как правило, совершает несколько шагов для обработки изображений. Часто изображение для начала обрабатывается с целью уменьшения шума или конвертации множества оттенков серого в простое сочетание черного и белого (бинаризации). После первоначальной обработки программа будет считать, производить измерения и/или определять объекты, размеры, дефекты и другие характеристики изображения.

## Методы обработки

Коммерческие пакеты программ для машинного зрения и пакеты программ с открытым исходным кодом обычно включают в себя ряд методов обработки изображений, таких как:

- Счетчик пикселей: подсчитывает количество светлых или темных пикселей
- Бинаризация: преобразует изображение в серых тонах в бинарное (белые и черные пиксели)
- Сегментация: используется для поиска деталей
- Чтение штрих кодов: декодирование 1D и 2D кодов, разработанных для считывания или сканирования машинами
- Оптическое распознавание символов: автоматизированное чтение текста
- Измерение: измерение размеров объектов
- Обнаружение краев: поиск краев объектов
- Сопоставление шаблонов: поиск, подбор, и/или подсчет конкретных моделей

В большинстве случаев, системы машинного зрения используют последовательное сочетание этих методов обработки для выполнения полного инспектирования.

## Типовые задачи компьютерного зрения

**Распознавание:** один или несколько предварительно заданных или изученных объектов, или классов объектов могут быть распознаны, обычно вместе с их двухмерным положением на изображении или трехмерным положением в сцене.

**Идентификация:** распознается индивидуальный экземпляр объекта. Примеры: идентификация определённого человеческого лица, отпечатков пальцев или автомобиля.

**Обнаружение:** видеоданные проверяются на наличие определённого условия. Например, обнаружение возможных неправильных клеток или

тканей в медицинских изображениях. Обнаружение, основанное на относительно простых и быстрых вычислениях, иногда используется для нахождения небольших участков в анализируемом изображении, которые затем анализируются с помощью приемов, более требовательных к ресурсам, для получения правильной интерпретации.

**Оценка положения:** определение положения или ориентации определённого объекта относительно камеры. Примером применения этой техники может быть действие манипулятора робота в извлечении объектов с ленты конвейера на линии сборки.

**Оптическое распознавание знаков:** распознавание символов на изображениях печатного или рукописного текста, обычно для перевода в текстовый формат, наиболее удобный для редактирования или индексации.

### **Движение**

Несколько задач, связанных с оценкой движения, в которых последовательность изображений (видеоданные) обрабатываются для нахождения оценки скорости каждой точки изображения или 3D сцены. Примерами таких задач являются:

- Определение трехмерного движения камеры
- Слежение, то есть следование за перемещениями объекта (например, машин или людей)

### **Восстановление изображений**

Удаление шума (шум датчика, размытость движущегося объекта и т. д.). Наиболее простым подходом к решению этой задачи являются различные типы фильтров, таких как фильтры нижних или средних частот. Более сложные методы используют представления того, как должны выглядеть те или иные участки изображения, и на основе этого их изменение. Высокий уровень удаления шумов достигается в ходе первоначального анализа видеоданных на наличие различных структур, таких как линии или границы, а затем управления процессом фильтрации на основе этих данных.

## Глава II. Существующее программное обеспечение для систем компьютерного зрения.

Существует множество примеров программного обеспечения, некоторые распространяются платно, некоторые на основе различных лицензий, так же как открытый код. У всех имеются как плюсы, так и минусы. Вот несколько примеров:

**Caffe** — среда для глубинного обучения, разработанная Яньцинем Цзя в процессе подготовки своей диссертации в университете Беркли. *Caffe* является открытым программным обеспечением, распространяемым под лицензией BSD license. Написано на языке C++, и поддерживает интерфейс на языке Python. [2]

**Caffe** поддерживает много типов машинного обучения, нацеленных в первую очередь на решение задач классификации и сегментации изображений. **Caffe** обеспечивает свёрточные нейронные сети, RCNN, долгую краткосрочную память и полносвязные нейронные сети. При этом для ускорения обучения применяется система графических процессоров (GPU), поддерживаемая архитектурой CUDA и использующих библиотеку CuDNN от фирмы Nvidia.

**Caffe** манипулирует блобами — многомерными массивами данных, которые используются в параллельных вычислениях, которые помещаются в CPU или GPU. В качестве входа используются данные из памяти, из базы данных или со внешних носителей информации. В качестве скрытых слоёв используются традиционные для свёрточной сети - свёрточные слои, слои ReLU, полносвязные слои, а также слои разворачивания (деконволюции) для сетей RNN. Доступны также многие другие типы слоёв, фильтров, преобразований данных и функций ошибок.

*Не используется в проекте так как распространяется по лицензии и имеет больший потенциал для коммерческой разработки, чем для академических целей*

**Yandex Vision.** Весной 2019 компания Yandex представили сервис Yandex Vision, который позволяет распознавать текст и лица, но пока в тестовом режиме. Предполагается использование для создание крупных архивов текстовых файлов.

*Не используется в проекте так как распространяется в коммерческих целях и имеет ограниченный функционал.*



**Google Cloud Vision.** Данное API совмещает все наработки компании Google в области компьютерного зрения, что включает в себя распознавание текста, лиц, детектирование класса объекта и его геолокацию.

*Не используется в проекте так как распространяется в коммерческих целях и хоть функционал впечатляет, не подходит для точечных задач, которые рассматриваются в проекте.*

### Глава III. OpenCV

В ходе изучения методов компьютерного зрения и различных open-source проектов выбор для данного проекта пал на проект OpenCV.

**OpenCV** (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) — 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. <sup>[3]</sup> С 2000 по 2008 год OpenCV разрабатывалась и поддерживалась в основном Intel, и Нижегородское отделение корпорации с самого начала играло ведущую роль в формировании облика библиотеки. В первые годы существования OpenCV стремительно расширялась вширь, обрастая базовой функциональностью, такой как основные структуры данных, алгоритмы обработки изображений (image processing), базовые алгоритмы компьютерного зрения, ввод и вывод изображений и видео. Уже в это время были реализованы алгоритмы детекции человеческих лиц (каскадный классификатор), поиска стереосоответствия, оптического потока и другие. Второй мощный толчок к развитию проект получил с приходом компании Willow Garage, основной целью которой является создание персонального робота. OpenCV стала важной частью ROS (Robot Operating System), и на ее основе был создан ряд компонент для робота PR2. При поддержке Willow была сформирована команда, стартовавшая существенную переработку библиотеки. Именно следствием этих усилий стало то, что OpenCV приобрела C++ и Python API, модуль features2d, новую архитектуру, билд-систему на основе CMake, систему непрерывной интеграции на основе BuildBot, улучшенную документацию и массу других новшеств. Сейчас можно говорить уже о третьем значительном этапе в жизни библиотеки, наступившем с приходом Nvidia. В 2010 году эта компания поддержала создание CUDA-оптимизированной версии библиотеки, что

также было сделано инженерами компании Itseez. Первым публичным результатом стала реализация алгоритма стереосоответствия, способная в реальном времени производить обработку видео в FullHD-разрешении (1920x1080 пикселей). На сегодняшний день `opencv_gpu` — это полновесный модуль, нашедший применение во многих промышленных приложениях. <sup>[4]</sup> Библиотека предоставляет множество функций, которые могут пригодиться в изучении и применении систем компьютерного зрения:

1. Базовые структуры, вычисления (математические функции, генерация псевдослучайных чисел, DFT, DCT, ввод/вывод в XML и т.п.)
2. Обработка изображений (фильтры, преобразования и т. д.).
3. Методы и модели машинного обучения (SVM, деревья принятия решений и т. д.).
4. Анализ движения и отслеживание объектов (оптический поток, шаблоны движения, устранение фона).
5. Детектирование объектов на изображении (вейвлеты Хаара, HOG и т. д.).
6. Калибровка камеры, поиск стерео-соответствия и элементы обработки трехмерных данных.

## Практическая часть

В данной части проекта будет подробно описана структура и процесс работы продукта данного проекта.

**Используемый язык программирования:** Python 3.7

**Используемые библиотеки:** sys, numpy, cv2, math, VK\_api

Как работает программа?

1. Получение изображения
2. Применение фильтра
3. Нахождение координат точек рабочей области
4. Ориентирование и сегментирование рабочей области из всего изображения
5. Сегментирование и распознавание содержания нужных участков рабочей области
6. Вычисления и отправка результатов пользователю

### 1. Получение изображения

После нанесения специальных меток на бланк, учитель отправляет сканированный бланк с комментарием (количество бланков, количество учеников, предмет) боту в соцсети Вконтакте. В ходе работы над проектом было проведено сравнение api для чат-ботов двух популярных платформ Вконтакте и Telegram, выбор пал именно на первое, так как существуют некоторые проблемы с подключением к серверам Telegram, если хост бота находится на территории РФ

### 2. Применение фильтра

После получения фотографии с сервера мессенджера, вначале оно переводится из RGB в HSV. Это делается для удобства при работе с цветом, так как в HSV модели первый параметр является конкретным цветом, а с помощью остальных настраивается насыщенность и яркость цвета. А в цветовой модели RGB цвет описывается с помощью трёх цветов (Red – красный, Green – зеленый, Blue – синий), которые принято называть

основными. После на изображение накладывается фильтр, который меняет цвета изображения. Он состоит из двух переменных, каждая из которых состоит из трех параметров по цветовой модели HSV. В результате, цвета специальных меток становятся белыми, а всё остальное изображение черным.

### 3. Нахождение координат точек рабочей области

После применения фильтра было получено черно-белое изображение. С помощью встроенной функции библиотеки Open CV `findContours`, определяются границы меток, а затем с помощью функции `drawContours`, точки меток закрашиваются в определенный цвет. Данная процедура проводится два раза, чтобы метки были окрашены равномерно и были получены более точные координаты. После еще раз применяется фильтр, описанный в пункте два, и с помощью метода моментов находятся координаты четырех точек, которые помогают обнаружить границы рабочей области, а после вырезать её из полного изображения.

### 4. Сегментирование и распознавание содержания нужных участков рабочей области

По заданным пропорциям изображение последовательно сегментируется и каждый участок обрабатывается с помощью нейросети и результат заносится в матрицу  $X \times Y$  (где  $X$  – количество учеников, а  $Y$  – количество заданий).

### 5. Вычисления и отправка результатов пользователю

Для каждого задания вычисляется процентное выполнение, данные берутся из матрицы, которая заполняется в процессе, описанном в пункте 5. Отправка результатов учителю происходит также через бота в соцсети Вконтакте.

# **Итоги**

## **Опровержение гипотезы**

В результате эксперимента и последующего опроса, подсчет процентного выполнения каждого задания вручную гораздо дольше и сложнее, чем с помощью программы.

## **Результат работы над проектом**

Результатом работы над проектом является программа, которая может помочь учителям облегчить работу и более эффективно организовать учебный процесс в старших классах. Программу можно загрузить на сервер и благодаря выбору мессенджера Telegram, который распространяется в открытом доступе, то любой преподаватель может ей воспользоваться. Программа будет дорабатываться в будущем, будет организована возможность не сканировать бланки, а фотографировать с телефона, перечень предметов будет расширяться и улучшаться стабильность и скорость работы самой программы.

# Литература

## Веб-сайты:

1. [https://ru.wikipedia.org/wiki/Компьютерное\\_зрение](https://ru.wikipedia.org/wiki/Компьютерное_зрение)
2. <https://ru.wikipedia.org/wiki/Caffe>
3. <https://ru.wikipedia.org/wiki/OpenCV>
4. <https://habr.com/ru/company/intel/blog/146434/>

## Другие:

1. <https://docs.opencv.org/>
2. <http://robocraft.ru/blog/computervision>
3. <https://towardsdatascience.com/object-detection-via-color-based-image-segmentation-using-python-e9b7c72f0e11>
4. <https://vk.com/dev/methods>