

# «Мобильные системы компьютерного зрения»

## Лабораторная №1

«Основы создания системы компьютерного зрения на базе платформы Jetson Nano»

### Цель работы.

Ознакомиться с основами создания приложения системы компьютерного зрения на платформе Jetson Nano.

### Задание.

1. Ознакомиться с основами архитектуры аппаратного и программного обеспечения Jetson Nano.
2. Выполнить подключение модулей и сенсоров, необходимых для выполнения задания в соответствии с вариантом.
3. Разработать программное обеспечение в соответствии с вариантом задания.
4. Описанная в задании функциональность должна выполняться в реальном времени, скорость обработки должна выводиться на монитор.
5. Протестировать разработанное ПО следующим образом:
  - 5.1. Оценить функциональность полученной системы.
  - 5.2. Оценить максимальную скорость видеопотока, при которой видео будет обрабатываться корректно.
  - 5.3. Изменить указанные в задании параметры, объяснить их влияние на работу программы.

### Инструментальные средства.

Лабораторная работа выполняется на языке Python 3, в качестве платформы используется одноплатный компьютер Jetson Nano.

### Материалы и пособия.

1. Гайд по настройке Jetson Nano <https://www.pyimagesearch.com/2019/05/06/getting-started-with-the-nvidia-jetson-nano/>
2. ПО для работы с Jetson от NVIDIA <https://developer.nvidia.com/embedded/develop/software>
3. Richard Szeliski «Computer Vision: Algorithms and Applications» [http://szeliski.org/Book/drafts/SzeliskiBook\\_20100903\\_draft.pdf](http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)

### Критерии оценивания выполнения работы.

По результатам работы должен быть подготовлен отчет в электронном виде.

Максимальный балл – 10. Работа считается сданной при оценке минимум в 5 баллов.

Оценка складывается из следующих составляющих:

- Соответствие заданной функциональности – 0-4 баллов;
- Выполнены п. 5.2 и 5.3 задания 0-2 балла;
- Защита работы 0-3 балла;
- Составление отчета 0-1 балл.

# Варианты лабораторной работы №1

Заготовки для выполнения заданий по вариантам на языке Python доступны в репозитории: <https://github.com/zeanfa/mobileCV>

## Вариант 1

Заданная функциональность	Программа принимает на вход изображение с камеры и выводит его на монитор. В определенной области изображения отображается рамка области интереса. Если изображение в этой области имеет один из трех цветов (R, G, B), то загорается светодиод соответствующего цвета, иначе ни один светодиод не горит. <a href="https://github.com/zeanfa/mobileCV/lab1/src/v1_color.py">https://github.com/zeanfa/mobileCV/lab1/src/v1_color.py</a>
Используемые модули	Цифровая камера, блок светодиодов.
Изменяемые параметры	Пороговые значения параметров H, S, V детектирования цветов.

## Вариант 2

Заданная функциональность	Программа принимает на вход изображение с камеры и выводит его на монитор (в формате grayscale). Одновременно с этим производится расчет гистограммы изображения и его вывод. По сигналу нажатия кнопки программа переходит в режим выравнивания гистограммы, на монитор выводится обработанное изображение и гистограмма. <a href="https://github.com/zeanfa/mobileCV/lab1/src/v2_hist.py">https://github.com/zeanfa/mobileCV/lab1/src/v2_hist.py</a>
Используемые модули	Цифровая камера, кнопка.
Изменяемые параметры	Значение параметра количества корзин гистограммы histSize.

## Вариант 3

Заданная функциональность	Программа принимает на вход изображение с камеры и выводит его на монитор. Производится обработка изображения сглаживающим фильтром Гаусса и фильтром для выделения границ (в данном случае, фильтром Собеля). При нажатии кнопки отфильтрованное изображение на мониторе переключается между производными по осям x и y и суммой производных. <a href="https://github.com/zeanfa/mobileCV/lab1/src/v3_sobel.py">https://github.com/zeanfa/mobileCV/lab1/src/v3_sobel.py</a>
Используемые модули	Цифровая камера, кнопка.
Изменяемые параметры	Значение размера ядра фильтра.

#### Вариант 4

Заданная функциональность	Программа принимает на вход изображение с камеры и выводит его на монитор. Далее производится поиск ключевых точек на изображении с помощью алгоритма ORB. При нажатии кнопки включается или выключается отображение на изображении полученных ключевых точек. <a href="https://github.com/zeanfa/mobileCV/lab1/src/v4_orb.py">https://github.com/zeanfa/mobileCV/lab1/src/v4_orb.py</a>
Используемые модули	Цифровая камера, кнопка.
Изменяемые параметры	Значение количества признаков nfeatures.

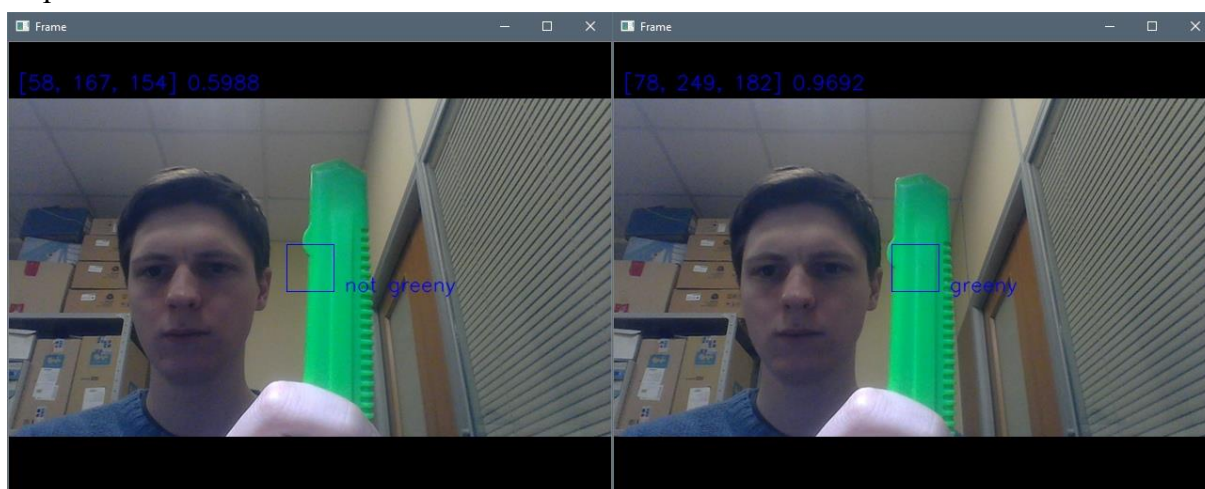
#### Вариант 5

Заданная функциональность	Программа принимает на вход изображение с камеры и выводит его на монитор. Далее производится бинаризация изображения с помощью алгоритма с адаптивным порогом. При нажатии кнопки происходит отображение бинарного изображения.
Используемые модули	Цифровая камера, кнопка.
Изменяемые параметры	Значение размера окружающей области blockSize.

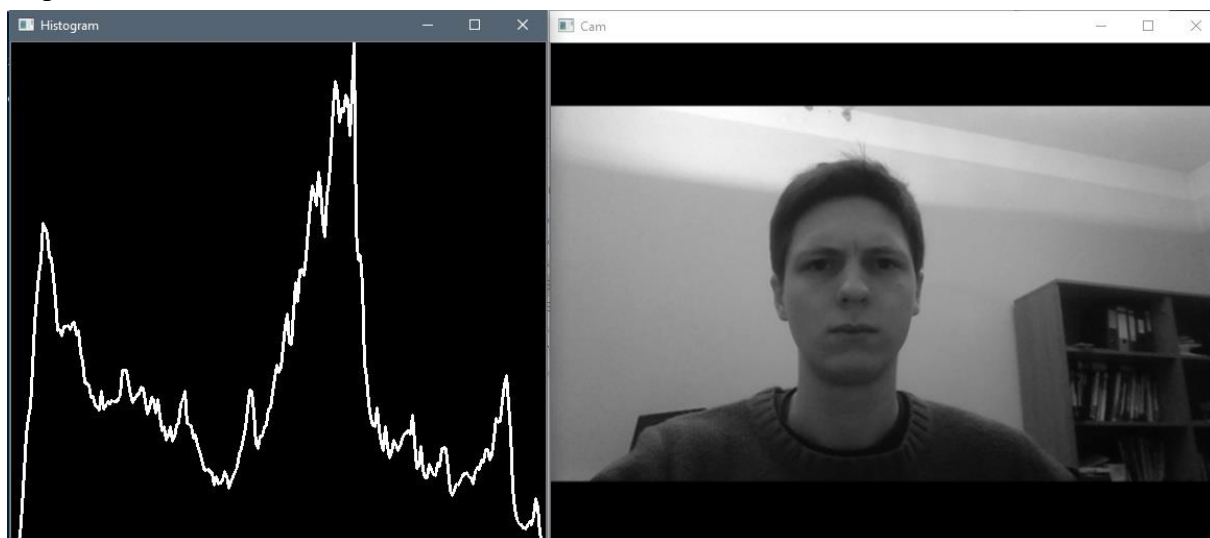
## Приложение

Далее приведены примеры работы алгоритмов по вариантам.

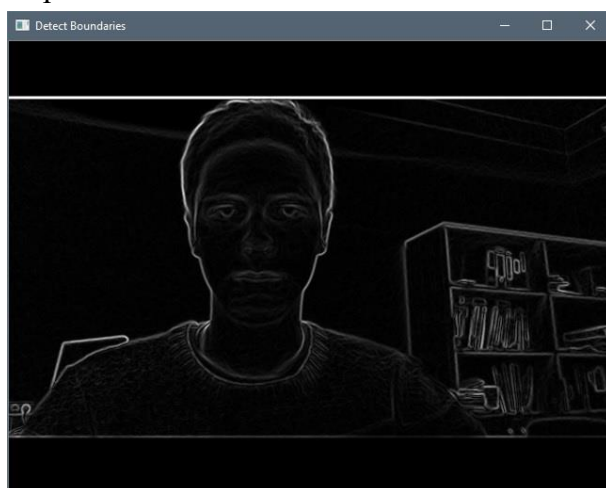
#### Вариант 1



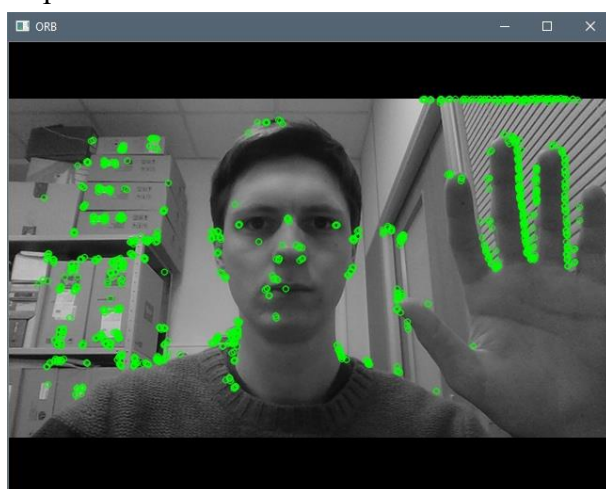
## Вариант 2



## Вариант 3



## Вариант 4



## Вариант 5

