

Муниципальное бюджетное учреждение дополнительного образования  
«Станция юных техников»  
имени Героя Социалистического труда Б. Г. Никитина

## **ПРОЕКТ**

### **«Модель нейронной сети для распознавания лиц в медицинской маске»**

**Выполнил: Русанов Глеб Александрович**

**Класс: 8**

**Руководитель: Безумова Надежда Михайловна,  
педагог дополнительного образования**

В последнее время нейронные сети стали очень востребованны. С их помощью значительно облегчается структурирование и анализ данных. Создание нейронной сети – задача непростая. Я решил попробовать свои силы в создании собственной модели нейронной сети для определения лиц без медицинской маски. Нейросеть в данном случае должна анализировать изображения лиц с видеокамеры, установленной на входе в здание. В условиях пандемии Covid-19 решение данной проблемы **актуально**.

**Цель проекта:** Создать рабочую нейросеть для проверки наличия медицинской маски.

#### **Задачи:**

- Познакомиться с типами нейронных сетей и с основными принципами их работы;
- Познакомиться со средствами разработки;
- Изучить необходимые Python-библиотеки;
- Найти необходимый набор данных для обучения сети;
- Создать и обучить модель нейронной сети и опробовать её на практике.

#### **Этапы реализации проекта**

##### **Первый этап - изучение видов и принципов работы нейронных сетей.**

Начав изучение этой темы я узнал, что искусственные нейронные сети предназначены для решения разных задач.

Затем я приступил к знакомству с платформой для машинного обучения TensorFlow с открытым исходным кодом. Здесь можно создавать модели, соединяя «строительные блоки» - блоки кода, написанные на языке программирования Python.

##### **Второй этап - продолжение изучения языка программирования Python и выбор среды разработки.**

Я продолжил изучение языка программирования Python на платформе Stepik.org. Я выбрал краткий курс «Python и нейронные сети». Полученных знаний оказалось недостаточно, и я продолжил обучение на сайте «Академия высоких технологий».

Выбор среды написания кода Jupyter Notebook или Google Colaboratory.

Я начал построение кода в блокноте Google Colaboratory. Это позволяет писать и выполнять код Python в браузере (также как и Jupyter Notebook). При этом:

- не требуется никакой настройки;
- вы получаете бесплатный доступ к графическим процессорам;
- предоставлять другим людям доступ к документам очень просто;
- Colab позволяет использовать для анализа и визуализации данных все возможности популярных библиотек Python [3]

Работать в Google Colaboratory оказалось достаточно удобно, предоставив доступ моему руководителю, который мог проверить и поправить ошибки. Хотя для этих целей можно было использовать и Jupyter Notebook.

### **Третий этап - изучение и подключение Python-библиотек.**

### **Четвёртый этап - Поиск наборов данных для обучения сети.**

Затем приступил к поиску наборов данных (Dataset) для обучения сети.

Dataset с фотографиями людей в масках и без я скачал на платформе Kaggle.

Далее я загрузил эти фото на свой Google диск в папки «With\_mask» и «No\_mask» и добавил ещё в эти папки фотографии, которые сделал сам.

Количество изображений должно быть достаточно большим, чтобы не произошло переобучения сети. Этот Dataset содержит 7545 изображений.

### **Пятый этап – работа с данными.**

1. В список сохраняем пути к изображениям и перемешиваем их между собой.
2. Обходим список и открываем изображения с помощью библиотеки OpenCV. Меняем размер изображений, который становится 40 на 40 пикселей.
3. Сохраняем метку, в качестве которой выступает имя папки с в которой находится данная картинка.
4. Сохраняем изображения и их метки в формате массива в numpy в отдельные файлы.

### **Шестой этап - создание нейронной сети и её обучение.**

Первый слой нейронной сети полносвязный и содержит 2500 нейронов, 7500 входов, а второй два нейрона.

В процессе обучения сети не всё сразу получалось: были ошибки, которые приходилось решать, варьируя некоторые параметры, например:

1. Изменяя количество эпох (EPOCHS) с 10 до 85.
2. Экспериментируя со скоростью обучения (INIT\_LR) изменяя с 0.001 до 0.1 и 0.3
3. Эксперименты с размерами пакетов (batch\_size), задавая размеры от 20 до 200.
4. Экспериментируя с оптимизаторами SGD и Adam т.д.

После применения различных оптимизаторов:

- Стохастический градиентный спуск (SGD)
- Стохастический градиентный спуск с инерцией (SGDm)
- Метод адаптивного градиента (Adagrad)
- Метод адаптивного шага обучения (Adadelta)
- Метод адаптивной инерции (Adam)

Я пришёл к выводу, что наилучшие результаты даёт SGD.

После проведённых экспериментов удалось добиться приемлемого результата точности обучения (ассигасу) 93%.

### **Седьмой этап - тестирование нейронной сети и вывод результатов в виде графика. Сохранение модели.**

#### **Описание результата проекта**

В результате, получилась обученная, полностью готовая к использованию модель нейросети, программа на Python, и .exe файл, который может исполняться на любом компьютере с системой Windows и Веб- камерой. За время исследования я узнал много

нового о нейронных сетях, понял их принцип работы, научился разрабатывать их самостоятельно. Создал свою, действующую нейронную сеть.

### Список информационных источников

1. <https://bio-smart.ru/tpost/d4klnv9x1o-raspoznavanie-po-litsu>
2. <https://www.goal.ru/sistemy-bezopasnosty-ceny/neuronet-price/>
3. <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ru>
4. <https://avt.global/contacts>
5. <https://www.reg.ru/blog/keras/>
6. [Шпаргалка по разновидностям нейронных сетей. Часть первая. Элементарные конфигурации \(tproger.ru\)](#)
7. [Общий круг задач, решаемых искусственными нейронными сетями \(neuronus.com\)](#)