

Всероссийский конкурс научно-технологических проектов «Большие  
Вызовы»

# Система распознавания лиц для пропускного пункта школы «Тётя Наташа»

Выполнила: Мягкова Ирина Олеговна

МБОУСОШ №38 11»В» класс

2022 год, г. Воронеж

# Содержание

<b>Введение.....</b>	<b>3</b>
<b>Исследовательский этап.....</b>	<b>6</b>
Теория создания нейронных сетей и работы с ними.....	6
<b>Технологический этап.....</b>	<b>10</b>
Оборудование и программное обеспечение.....	10
План работы.....	11
Технология.....	12
Калибровка.....	17
Система распознавания лиц.....	18
<b>Заключительный этап.....</b>	<b>19</b>
<b>Список литературы.....</b>	<b>20</b>

# Введение

**Актуальность.** В современном мире безопасность выходит на первый план. Практически в каждом офисе, школе или университете существует пропускная система. В большинстве таких учреждений нужны пропускные карточки или пропуска. Но иногда эти маленькие пластиковые прямоугольники теряются или забываются в другой сумке или куртке. В некоторых местах без пропуска невозможно попасть внутрь, а где-то всё равно пропустят. И в том, и в другом случае система не эффективна. В первом случае работник/ученик не могут выполнять свою работу от чего страдают сами, а также учреждение. Во втором случае страдает безопасность, ведь охранники не могут знать всех в лицо, а значит они могут пропустить нарушителя. Возникает вопрос. Что же мы не можем потерять и что может служить нашим личным идентификатором. Ответ прост. Наше лицо. Лицо нельзя забыть или потерять (в прямом смысле). Так и появилась идея создания пропускной системы с использованием распознавания лиц.

Конечно, система распознавания лиц не новая технология, она применяется во многих сферах нашей жизни, но тем не менее, её интеграция в качестве пропускной системы почти не развита. Поэтому я считаю свой проект актуальным.

**Обоснование проблемы.** В моей школе существует система пропусков через турникет, но школьники часто забывают свой пропуск или теряют, а восстановление не только стоит денег, но и достаточно долгий процесс. Из-за этого многие ученики просят друзей пропускать их. В связи с этим в базе данных один человек мог входить в школу несколько раз подряд, а другой не появляться неделями. Вся суть этой системы потерялась и её наличие утратило смысл, а ведь сама идея пропускного пункта не только выступает в качестве безопасного барьера, но и может следить за опоздавшими и фиксировать количество учеников в школе каждый день. В связи с этим я решила посвятить свой проект именно созданию системы распознавания лиц для пропускных пунктов

**Основная аудитория.** Мой проект рассчитан на школы, высшие учебные заведения и организации доп. Образования в которых предусмотрен вход через турникеты и подобные системы безопасности, требующие пропуска.

Компоненты нужные для функционирования моей системы есть в каждой школе (ноутбук/компьютер, камера, электричество, интернет). Соответственно установка системы «Тётя Наташа» не нанесёт большого ущерба бюджету образовательных организаций. Единственный ресурс, который будет затрачиваться (при наличии выше указанных компонентов) – время на обучение нейронной сети.

**Цель:** создать систему распознавания лиц

**Задачи:**

- Собрать и разметить dataset для задач object detection
- Обучить нейронную сеть с помощью yolov3
- Проанализировать возможные способы совмещения весов нейронной сети и камеры
- Вывести видео в реальном времени одновременно с определением лиц
- Протестировать систему
- Сделать вывод о её эффективности

# Исследовательский этап

## Теория создания нейронных сетей и работы с ними

**Нейронная сеть** — последовательность нейронов, соединённых между собой синапсами. Эта структура пришла в программирование из биологии. Нейронные сети используются для решения задач, которые требуют аналитики каких-то данных. [4]

Самыми распространёнными применениями нейронных сетей является [4]:

- ❖ **Классификация** — распределение данных по параметрам. Например, нам даётся набор домов и нужно определить какие в аварийном состоянии, а какие нет. Эту работу может сделать нейронная сеть, анализируя информацию: возраст постройки, состояние труб, проводки и т.д..
- ❖ **Предсказание** — возможность предсказывать следующий шаг. Например, рост или падение акций, основываясь на ситуации на фондовом рынке.
- ❖ **Распознавание** — самое широкое применение нейронных сетей. Например маски в ВКонтakte, нейронная сеть определяет положение вашего лица и накладывает фильтр/маску и т.д..

**Нейрон** — вычислительная единица, которая получает информацию, производит над ней простые вычисления и передаёт её дальше. Нейроны бывают трёх типов: входной, скрытый и выходной. В том случае, когда нейросеть состоит из большого количества нейронов, вводят термин слоя. Есть входной слой, который получает информацию, некоторое количество скрытых слоев, которые её обрабатывают и выходной слой, который выводит результат. У каждого из нейронов есть 2 основных параметра: входные данные и выходные данные. В случае входного нейрона: входные данные=выходные

данные. В остальных, в поле входные данные попадает суммарная информация всех нейронов с предыдущего слоя, после чего, она нормализуется, с помощью функции активации и попадает в поле выходные данные. [4]

**Синапс** — связь между двумя нейронами. Параметр синапса - вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому. Допустим, есть 3 нейрона, которые передают информацию следующему. Тогда у нас есть 3 веса, соответствующие каждому из этих нейронов. У того нейрона, у которого вес будет больше, та информация и будет доминирующей в следующем нейроне [4]

**Функция активации** — способ нормализации входных данных. Если на входе у вас будет большое число, пропустив его через функцию активации, вы получите выход в нужном вам диапазоне. [4]

**YOLOv3** — это очень популярная архитектура CNN, которая используется для распознавания множественных объектов на изображении. Это усовершенствованная версия архитектуры YOLO. Она лучше детектирует небольшие объекты по сравнению с её предшественниками. Основная особенность YOLOv3 состоит в том, что на выходе есть три слоя каждый из которых рассчитан на обнаружения объектов разного размера. Оригинальная архитектура YOLOv3 реализована с помощью framework Darknet. [1]

Сейчас очень много задач в области компьютерного зрения решаются с помощью свёрточных нейронных сетей (Convolutional Neural Networks). Благодаря своему строению они хорошо извлекают признаки из изображения. CNN используются в задачах, которые я описала ранее.

Популярные архитектуры CNN для распознавания объектов [1]:

- ❖ **R-CNN.** Обычный классификатор изображений. На вход сети подаются разные регионы изображения и для них делается предсказание. Очень медленная.
- ❖ **Fast R-CNN.** Улучшенная и более быстрая версия R-CNN, работает по похожему принципу. Медленная для задач реального времени.
- ❖ **Faster R-CNN.** Главное отличие от предыдущих в том, что вместо selective search алгоритма для выбора регионов использует нейронную сеть для их «заучивания».
- ❖ **YOLO.** Совсем другой принцип работы по сравнению с предыдущими, не использует регионы вообще. Наиболее быстрая. Более подробно о ней говорится выше.
- ❖ **SSD.** По принципу похожа на YOLO. Тоже довольно быстрая и пригодная для работы в реальном времени.
- ❖ **Feature Pyramid Networks (FPN).** Еще одна разновидность сети типа Single Shot Detector, из-за особенности извлечения признаков лучше чем SSD распознает мелкие объекты.
- ❖ **RetinaNet.** Использует комбинацию FPN+ResNet и благодаря специальной функции ошибки дает более высокую точность.

**TensorFlow** — это ML-framework от Google, который предназначен для работы с моделями глубокого обучения. Глубокое обучение — это область машинного обучения, алгоритмы в которой были вдохновлены структурой и работой мозга. TensorFlow используют для численных вычислений. Само по себе это не кажется специфичным, однако эти вычисления производятся с помощью data-flow графов. В этих графах вершины представляют собой математические операции, в то время как ребра представляют собой данные.

[2]



**Anaconda** - Дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения. Основная цель - поставка единым согласованным комплектом наиболее востребованных соответствующим кругом пользователей тематических модулей с разрешением возникающих зависимостей и конфликтов, которые неизбежны при одиночной установке.

# **Технологический этап**

## **Оборудование, программное обеспечение**

- Ноутбук/компьютер
- Камера
- Google Colab
- Интернет
- Терминал
- Github
- CVAT

# План работы

1. Снять на видео людей для первого обучения
2. Разметить видео и скачать dataset
3. Загрузить нужные файлы на google диск
4. Обучить модель
5. Проверить модель
6. Подключить модель в веб камере

# Технология

1. Первым делом я сняла видео с тремя своими одноклассницами. Я решила, что четыре видео (включая моё видео) будет достаточно для прототипа. Видео снимались таким образом, чтобы девочки немного меняли причёски, закрывали какие-то части лица и меняли эмоции (это было нужно для разнообразия датасета).
2. Далее, с помощью программы CVAT, я разделила видео на кадры и на каждом кадре выделила лицо девочек. Это было долгое занятие, но мне нравится размечать данные, это довольно медитативный процесс. Вообще от разметки данных зависит очень много, поэтому этот этап очень важен. Одно из основных правил выделения объектов в разметке заключается в том, что если выделять объект квадратом, то стороны этого квадрата должны быть как можно ближе к объекту.
3. Третий этап – подготовка файлов. Для начала я конвертировала все файлы из формата PNG в JPG. После этого я переместила фотографии и текстовые данные со значением расположения объектов на изображении в одну папку и сжала её. Далее мне нужно было изменить под себя файлы `obj.name`, `obj.data`, `yolov3_custom.cfg`. В `obj.name` находятся названия наших объектов -labels. В моём случае это имена учеников. Важно расположить эти имена в том порядке, в котором были пронумерованы labels в CVAT. В `obj.data` хранятся пути сохранения весов и текстовых файлов теста и valid, а также количество классов, значение `classes` нужно изменить на 4, так как мне нужно, чтобы нейронная сеть определяла 4 класса. И последний файл `yolov3_custom.cfg`. Там содержится вся важная

информация для обучения. Сначала я сняла комментарий со строк для обучения. Потом изменила значение `max_batches` (количество проходимых эпох обучения) по формуле количество классов \* 2000 ( $4*2000=8000$ ). Также немного ниже нужно изменить значение `steps`. Эти два значения должны быть равны 80% и 90% от `max_batches` (в моём случае 6400 и 7200). В этом файле присутствует 3 идентичных части под название `[yolo]`. Во всех этих частях нужно поменять количество классов (для меня 4) и значение `filters` (над блоком `[yolo]`) по формуле количество (классов + 5) \* 3 ( $(4+5)*3=27$ )

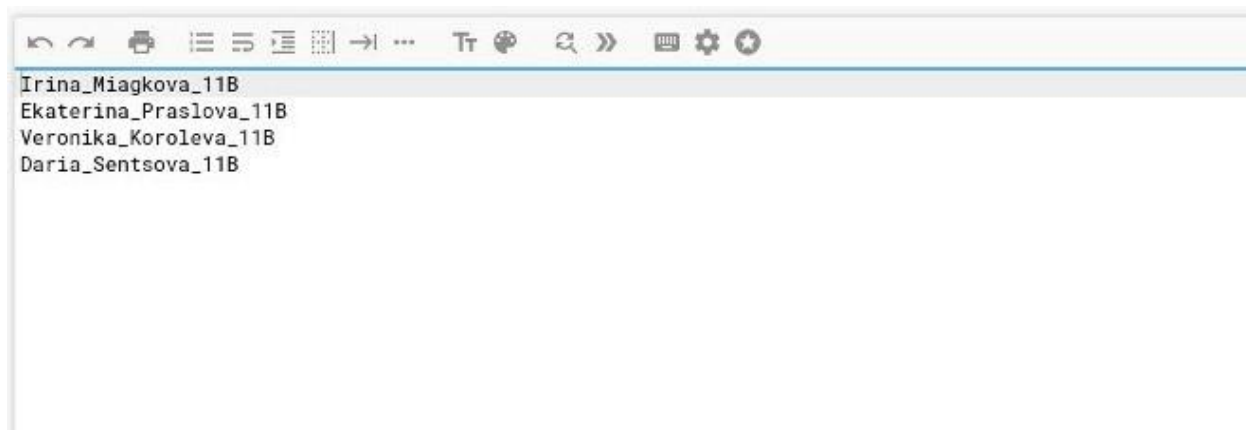


Рис.1 Изменение файла `obi.name`

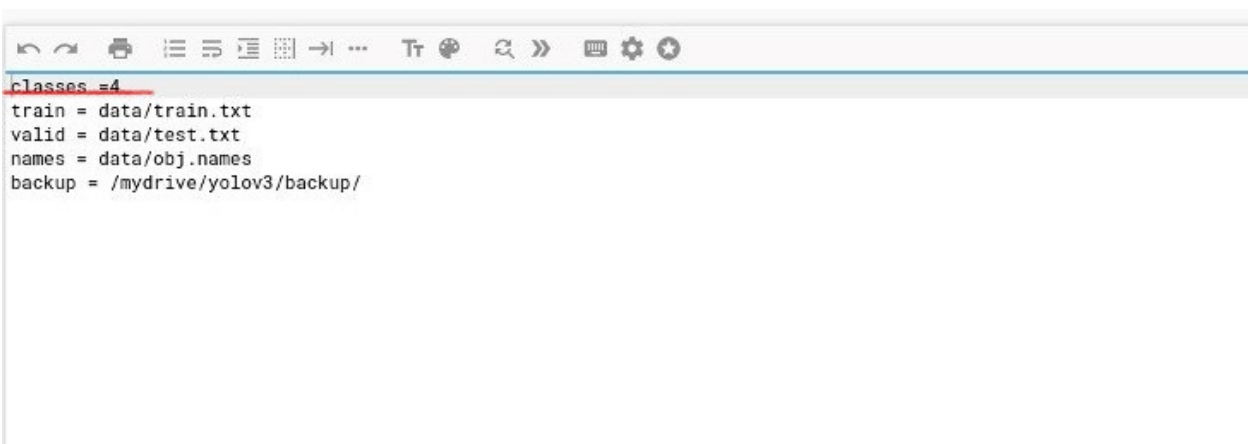


Рис.2 Изменение файла `obi.data`

```
1 [net]
2 # Testing
3 #batch=1
4 #subdivisions=1
5 # Training
6 batch=64
7 subdivisions=16
8 width=416
9 height=416
10 channels=3
11 momentum=0.9
12 decay=0.0005
13 angle=0
14 saturation = 1.5
15 exposure = 1.5
16 hue=.1
17
18 learning_rate=0.001
19 burn_in=1000
20 max_batches = 8000
21 policy=steps
22 steps=6400,7200
23 scales=.1,.1
24
25 [convolutional]
26 batch_normalize=1
27 filters=32
28 size=3
29 stride=1
30 pad=1
31 activation=leaky
32
```

Рис.3 Изменение файла yolov3\_custom.cfg.

```
759 size=1
760 stride=1
761 pad=1
762 activation=leaky
763
764 [convolutional]
765 batch_normalize=1
766 size=3
767 stride=1
768 pad=1
769 filters=256
770 activation=leaky
771
772 [convolutional]
773 size=1
774 stride=1
775 pad=1
776 filters=27
777 activation=linear
778
779
780 [yolo]
781 mask = 0,1,2
782 anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326
783 classes=4
784 num=9
785 jitter=.3
786 ignore_thresh = .7
787 truth_thresh = 1
788 random=1
789
```

Рис.4 Изменение файла yolov3\_custom.cfg.

4. Теперь переходим к обучению. Я обучала с помощью Google colab. Google colab даёт доступ к gpu (ускоряет обучение) и вообще этот сервер отлично подходит для машинного обучения. Единственный минус, что пользоваться gpu можно 6 часов в 24 часа, поэтому обучение затягивается на несколько дней. Я обучала по уже готовому коду для обучения yolov3

1. Сначала я обучила нейронную сеть на своих фотографиях, да, она определяла меня, но, по всей видимости, нейронная сеть запомнила не какие-то особенные черты моего лица, а просто вид человеческого лица, потому что она определяла всех, как меня.

2. Далее я обучала нейронную сеть на dataset со всеми участниками, но я поступила опрометчива и не проверила количество фотографий каждой, поэтому dataset получился неравномерный и нейронная сеть неправильно определяла лица. Так я подумала первоначально. Когда я запускала программу повторно, то делала это при ярком освещении и тогда нейронная сеть определила меня правильно, а чужого человека не определила. Но даже тут была небольшая ошибка, моих родителей программа определяла так же, как меня. Такой результат меня не устроил, и я решила заново обучить нейронную сеть.

3. Учитывая предыдущие ошибки, я сравняла количество фотографий, насколько это было возможно и дополнила dataset обработанными фотографиями (изменение контрастности, чёрно-белый эффект, рябь, зелёный фильтр, красный фильтр, синий фильтр). На новом наборе данных я обучила нейронную сеть. Теперь при плохом свете лица определяются лучше, но проблему с определением родственников решить не удалось.

5. Последний этап – вывод видео с камеры совместно с нейронной сетью. Над этим пунктом я думала долго, изначально я планировала использовать yolov5 и flask, но, к сожалению, моих знаний не хватило для реализации моей идеи именно этим способом. Поэтому я начала поиски решения. На просторах интернета я нашла способ вывода видео с помощью терминала с использованием Anaconda и TensorFlow (см. Исследование). Для начала я скачала все нужные библиотеки, а также набор файлов с GitHub для реализации. После этого я перевела файлы из формата weights в формат tf (формат для работы TensorFlow) с помощью команды

```
python load_weights.py --weights './weights/yolov3_custom_last.weights' --output  
 './weights/yolov3_custom.tf' --num_classes '4'
```

Потом, когда файлы приобрели нужно расширение я ввожу команду, для вывода видео с видео камеры, а также записи этого видео. Зачем я решила записывать видео? Это довольно просто, пусть по статистике нейронные сети ошибаются меньше людей (см. Исследование), но всё же ошибаются, поэтому наверняка понадобятся записи в случае спорных ситуаций, а также это просто как видео наблюдение. Я думаю, что такая функция довольно полезна.

```
python detect_video.py --weights 'weights/ yolov3_custom.tf' --output  
'data/video/video.avi' --video 0 -num_classes '4' --classes 'data/labels/obj.names'
```



# Калибровка

Самая большая проблема у меня возникла с dataset-ом. Сначала я неправильно расположила labels и поэтому с одним номером у меня были разные люди. Эта проблема решается новой разметкой с правильным расположением классов. Следующая проблема заключалась с обучением из-за неравномерности dataset-а, но эту проблему и её решение я рассмотрела в ходе работы, пункте 6.2.

# Система распознавания лиц

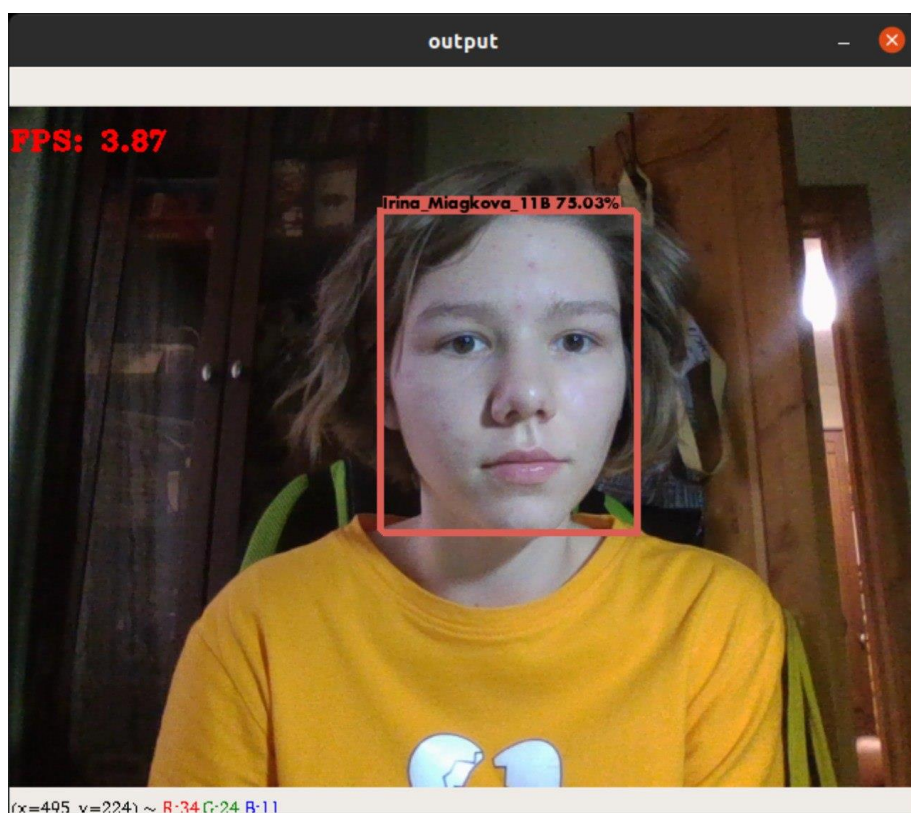


Рис.5 Работа системы при хорошем освещении

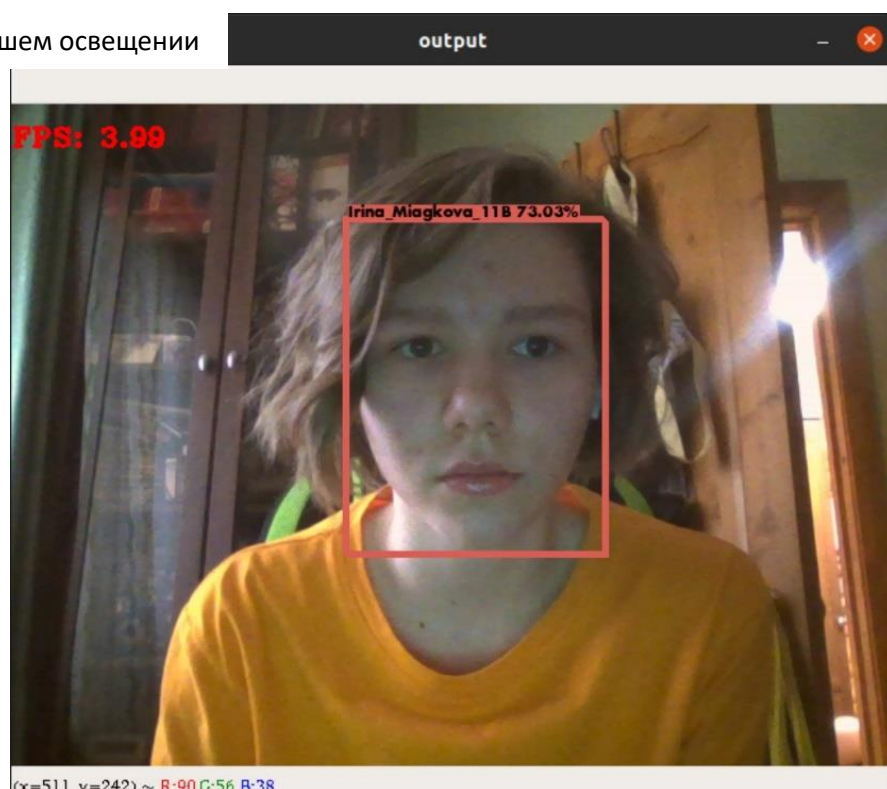


Рис.6 Работа системы при плохом освещении

# Заключительный этап

**Вывод.** В результате работы был создан прототип системы распознавания лиц для пропускного пункта школы. Эта система способна распознавать лица учеников, выводить их имя, фамилию и класс, а также сохранять видео с распознаванием. В моём прототипе есть ряд недочётов, которые я буду исправлять в дальнейшем. Первое – определение родственников ученика, как его самого. Второе - при дальнейшем расположении ученика при определении могут происходить ошибки. Для решения этой проблемы около информации об ученике есть процент верности, поэтому если процент меньше 85, то нужно подойти ближе к камере или сделать свет ярче для точного распознавания. Эти недочёты я планирую решать с помощью расширения dataset-a (дальнее и ближнее расположения человека)

Стоимость моей системы зависит от компонентов: компьютер/ноутбук, камера, электричество, время и интернет. Учитывая эти факторы можно предположить, что система будет стоить около 43000, но может быть и дешевле, и дороже.

**Перспективы развития.** В дальнейшей работе над проектом планируется совмещение системы распознавания лиц с турникетом, а также доработка самой нейронной сети. Рассматриваются так же варианты изменения методики обучения и анализирование новых методик создания систем распознавания лиц.

Когда проект будет доведён до полной работоспособности планируется распространение системы «Тётя Наташа» по школам.

# Список литературы

В тексте обозначается как [№ссылки]

1. Распознавание объектов в режиме реального времени с помощью YOLOv3 (дата обращения: декабрь 2021)  
<https://habr.com/ru/post/460869/>
2. Библиотека глубокого обучения Tensorflow дата обращения: январь 2022 <https://habr.com/ru/company/ods/blog/324898/>
3. Работа в режиме реального времени с использованием YOLOv3 Абхинав Сагар (дата обращения: декабрь 2021)  
<https://www.machinelearningmastery.ru/pedestrian-tracking-in-real-time-using-yolov3-33439125efdf/>
4. Нейронные сети (дата обращения: октябрь 2021)  
<https://habr.com/ru/post/312450/>
5. Улучшение алгоритма yolo-v3 для распознавания небольших объектов Лоу Гуанпин (дата обращения: февраль 2022)  
<https://cyberleninka.ru/article/n/uluchshenie-algoritma-yolo-v3-dlya-raspoznavaniya-nebolshih-obektov>
6. Детектирование и классификация дорожных знаков в реальном времени на основе алгоритма YOLO версии 3 Сичкар В.Н. Колубин С.А. (дата обращения: январь 2022)  
<https://cyberleninka.ru/article/n/detektirovanie-i-klassifikatsiya-dorozhnyh-znakov-v-realnom-vremeni-na-osnove-algoritma-yolo-versii-3>
7. Разработка программы распознавания лиц Ахметзянов К.Р. Сазонов В.И. Липин Ю.Н. Южаков А.А. (дата обращения: декабрь 2021)  
<https://cyberleninka.ru/article/n/razrabotka-programmy-raspoznavaniya-lits>
8. Обзор методов распознавания лица на изображении

А. В. Антончик А. А. Дерюшев (дата обращения: октябрь 2021)

<https://cyberleninka.ru/article/n/obzor-metodov-raspoznavaniya-litsa-na-izobrazhenii>