**Перечень источников информации по проекту скилфактори и подход к его выполнению.**

Тема «Идентификация наличия медицинской маски на человеке на потоковом видео (on-line)»

Подход:

1. Найти размеченные данные (фотографии) с людьми в масках и без.
2. Предобработать данные для их последующей обработки на модели
3. Разработать модель (выбрать модель и выбрать датасет предобучения)
4. Разработать механизм захвата потокового видео с вебкамеры.

Источники информации для реализации проекта:

Датасет 1: <https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset>

Датасет 2: [https://github.com/prajnasb/observations/tree/master/experiements/data (480](https://github.com/prajnasb/observations/tree/master/experiements/data%20(480) -без масок,480 с масками)

**Информация по проекту**

**Структура отчетного документа.**

**Обоснование выбора темы проекта:**

Маски для лица стали жизненно важным элементом нашей повседневной жизни за время глобальной пандемии (COVID-19). Поэтому их ношение важно для безопасности и контроля распространения вируса. Медицинские маски способны значительно снизить уровень распространения смертельного вируса. По мере того как мы начали продвигаться вперед в этом "новом нормальном" мире, необходимость в маске для лица возросла. На сегодняшний день актуальным является вопрос контроля наличия маски на людях в общественных местах (торговых центрах, магазинах, кафе, ресторанах и так далее). В большинстве случаев сегодня эта задача решается с помощь человеческого труда или не контролируется должным образом.

**Целью** данного проекта является ограничение возможности людей совершать различные операции в общественных местах без наличия медицинской маски автоматическим способом.

Для достижения поставленной цели в проекте придется решить следующие **задачи**:

* Определить критерии правильно надетой на человека маски
* Определить критерии выбора механизмов для реализации проекта
* Рассмотреть возможные варианты внедрения полученных результатов в повседневную жизнь.

**Применимость результатов.** С помощью результатов данного проекта можно будет осуществлять, например, следующие виды контроля:

* Контроль наличия медицинской маски на посетителе торгового центра/магазина (пример решения в Азии:<https://twitter.com/larrykim/status/1318243765979615233?s=21>)
* Контроль наличия медицинской маски на покупателе в магазине на кассе – при наличии маски может активироваться терминал оплаты.
* Контроль наличия медицинских масок на посетителях общественного транспорта, а также подсчет нарушителей режима. Информация может поступать на пульт диспетчерской)

Для реализации данного проекта определены следующие **этапы**:

1. Определение критериев правильности надетой на человеке маски
2. Сбор и разметка данных для обучения по сформированным категориям
3. Определение критериев выбора механизмов реализации распознавания маски на лице человека
4. Обучение модели
5. Тестирование модели на потоковом видео
6. Подведение итогов эксперимента

**Этап 1. Определение критериев правильности надетой на человеке маски**

Для того, чтобы приступить к сбору данных и формированием датасета для обучения необходимо определиться с критериями правильности надевания медицинской маски человеком. Например, на сегодняшний день люди могут носить маску на подбородке, не закрывая нос и так далее. Как мы будем их классифицировать.

В рамках текущего проекта мы выделим 3 класса для идентификации наличия маски на человеке:

* **Маска надета** – критерием для данного класса будет следующее расположение маски: маска закрывает нос человека, и маска закрывает рот человека
* **Маска надета неправильно** – критерием для данного класса будет следующее расположение маски на лице человека: маска не закрывает нос, маска не закрывает рот человека, маска не закрывает часть носа или рта человека.
* **Маска не надета** – критерием для данного класса будет полное отсутствие маски на лице человека

**Этап 2. Сбор и разметка данных для обучения по сформированным категориям**

Для формирования датасета были использованы открытые источники. По итогам сбора данных фотографии людей были классифицированы в соответствии с определенными для проекта критериями. Итого для будущего обучения собрано следующее количество фотографий людей:

* Фотографии с людьми в масках - \_\_ шт.
* Фотографии людей с неправильно надетыми масками \_\_\_ шт.
* Фотографии людей без масок \_\_\_ шт.

**Этап 3. Определение критериев выбора механизмов реализации распознавания маски на лице человека**

Перед тем как переходить к определению критериев нам необходимо определиться с перечнем технологий и подходов, которые будут использованы в данном проекте. Верхнеуровнево в проекте необходимо обучить модель по фотографиям различать то, как надета маска на человека и далее применить обученную модель для распознавания медицинской маски на потоковом видео. Для этого нам в проекте потребуются следующие технологии:

1. Архитектура нейросети – «голова» для transfer learning и fine-tuning
2. Датасет, на котором была предобучена «голова» нашей нейросети
3. Технология детекции лица на фотографии/видео
4. Технологии захвата и работы с потоковым видео

**Архитектура нейросети**

Основным критерием при выборе архитектуры сети для распознавание медицинских масок на потоковом видео является ее легковесность. Данный критерий повлияет как на время обучения, так и на время ее работы на реальных данных. В связи с тем, что функционал по большей части будет реализовываться на портативных устройствах, то их мощности и памяти должно хватать для обработки потока видео и распознавания с должной частотой на нем медицинской маски и лиц людей.

#### Для решения стоящих перед проектом задач была выбрана архитектура нейросети MobileNetV2. Сравним между собой несколько архитектур сетей. Возьмем для примера Xception, о котором был прошлый пост, глубокую и старую VGG16, а также несколько вариаций MobileNet

|  |  |  |  |
| --- | --- | --- | --- |
| **Архитектура сети** | **Количество параметров** | **Top-1 accuracy** | **Top-5 accuracy** |
| Xception | 22.91M | 0.790 | 0.945 |
| VGG16 | 138.35M | 0.715 | 0.901 |
| MobileNetV1 | 4.20M | 0.709 | 0.899 |
| MobileNetV2 | 6.06M | **0.750** | **0.925** |

Также, в статье про MobileNetV2 показываются весьма интересные результаты на других задачах. В частности, авторы демонстрируют, что SSDLite-архитектура для задачи object detection, использующая MobileNetV2 в свёрточной части, превосходит известный детектор реального времени YOLOv2 по точности на датасете [MS COCO](http://cocodataset.org/#home), при этом показывая в 20 раз большую скорость и в 10 раз меньший размер (в частности, на смартфоне Google Pixel сеть MobileNetV2 позволяет делать object detection с 5 FPS).

Для проекта выбираем MobileNetV2 предобученную на датасете ImageNet. Также в качестве эксперимента попробуем использовать следующее поколение архитектуры сети MobileNetV3\_Lite и MobileNetV3\_Large

**Технология детекции лица на фотографии/видео**

Для целей распознавая наличия медицинской маски на человеке необходимо сначала ограничить область поиска маски. В нашем случае мы ее будем искать на лице. Данный подход позволит, во-первых, снизить время расчета по модели, во-вторых снизить требования к аппаратному обеспечению (вычислительному), а также исключить потенциальную возможность обнаружения медицинской маски в иных местах (нас такие значения не интересуют).

Два основных подхода к обнаружению лиц:

* классический каскадный классификатор на основе признаков-фильтров ([метод  Виолы-Джонса](http://api-2d3d-cad.com/viola-jones-method/));
* совместное обнаружение и выравнивание лиц с использованием многозадачных каскадных сверточных сетей.

Первый подход реализуется через библиотеку OpenCV, второй — через библиотеки DLIB. MTCNN и др. Ниже представлен сравнительный анализ подходов:



В рамках проекта посмотрим, как работают оба этих метода. В части каскадного классификатора будем использовать каскад Хаара (haarcascadefrontalface) а в части многозадачных сетей будем использовать решение от MTCNN.

**Технологии захвата и работы с потоковым видео**

Для работы с потоковым видео в данном проекте будет использоваться библиотека OpenCV. Данная библиотека позволяет работать с изображениями, совершать преобразования. Работа с видео будет осуществлена следующим образом: 1 кард – это картинка, соответственно, чтобы работать с потоковым видео необходимо создать бесконечный цикл картинок (кадров), а также предусмотреть выход из этого цикла по команде. Захват видео будет осуществляться с встроенной камеры ноутбука.

**Этап 4. Обучение модели**

**Этап 5. Тестирование модели на потоковом видео**