ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

«УНИВЕРСИТЕТ “ДУБНА”»

ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА И УПРАВЛЕНИЯ

Кафедра распределенных информационно-вычислительных систем

**Курсовая работа**

по дисциплине

**«Теория принятий решений»**

|  |  |
| --- | --- |
| **ТЕМА:** | Распознавание речи по губам на видео без использования информации |
|  | об аудиосигнале с помощью алгоритмов машинного обучения |
|  | (наименование темы) |

|  |  |
| --- | --- |
| **Выполнил:** студент группы | **2253** |
| **Козловский А.А.** | |
| (Ф.И.О.) | |
|  | |
| (подпись студента) | |

|  |  |
| --- | --- |
| **Руководители:** по дисциплине теория принятия решений | |
| **cт. преп. Булякова И.А.** | |
| (учёная степень, учёное звание, занимаемая должность, Ф.И.О.) | |
|  | |
| Дата: |  |
| Оценка: |  |
|  | |
| (подпись руководителя) | |
|  | |

Дубна, 2021

# Содержание

[Содержание 2](#_Toc1538374108)

[Введение 3](#_Toc854701266)

[Постановка задачи 4](#_Toc1259887406)

[Теоретическая часть 5](#_Toc1508403088)

[Используемые термины 5](#_Toc1851317774)

[Операция свертки в задаче распознавания 5](#_Toc1897311045)

[Градиентные фильтры 6](#_Toc1139685382)

[Фильтр Собеля 7](#_Toc1474957085)

[Используемые алгоритмы 8](#_Toc1559817659)

[Гистограмма направленных градиентов 8](#_Toc1061306321)

[Метод опорных векторов 8](#_Toc1651404354)

[Классификатор случайного леса 8](#_Toc426576303)

[Практическая часть 9](#_Toc524334910)

[План решения задачи 9](#_Toc210404620)

[Сегментация аудиодорожки 10](#_Toc1042304128)

[Выделение области губ 10](#_Toc478472436)

[Автоматический сбор датасета 10](#_Toc747050072)

[Анализ датасета 11](#_Toc1825632179)

[Выявленные проблемы 11](#_Toc1359760398)

[Заключение 12](#_Toc807193011)

[Список литературы 13](#_Toc1911003744)

# Введение

Задача распознавания образов очень актуальна в наши дни и по праву заслуживает большого внимания. На данный момент распознавание речи по аудиосигналу является обыденной задачей. Однако, этот вид распознавания сталкивается с некоторыми проблемами, вызванными сильной зависимостью с качеством аудиосигнала. Одними из самых распространенных проблем при классическом распознавании речи по аудиосигналу являются:

1. Высокий уровень шума, в котором сложно выделить речь;
2. Параллельная речь нескольких человек, которую сложно разделить.

Актуальность работы заключается в том, что в случае распознавания речи на видео выше перечисленные проблемы можно решить, используя распознавание основанное на видеосигнале.

Чтобы ещё больше подчеркнуть актуальность решаемой задачи следует перечислить примеры некоторых практических задач, которые можно было бы решить, благодаря разработанной модели:

1. Улучшение качества распознавания на готовых видеозаписях.
2. Улучшение качества

В работе рассматривается план решения проблемы распознавания речи по видеосигналу, без использования информации об аудиосигнале, а также сбор датасета, необходимого для обучения модели распознавания.

# Постановка задачи

Цель: спроектировать, реализовать и обучить модель для распознавания речи по видеосигналу.

Исходные данные: датасет отдельных видеофрагментов с движениями губ говорящего на русском языке, сегментированный по словам и слогам.

Средства реализации: модель должна быть реализована на языке программирования *python* с использованием библиотек: *numpy, cv2, bs4, pytube, tensorflow, dlib.*

Модельные представления: модель должна представлять собой рекурентную сверточную нейронную сеть, возможно с *LSTM* слоем.

Ожидаемый результат: модель распознавания русскоязычной речи по видеосигналу.

Критерий оценки результатов: модель должна иметь точность распознавания не менее 75% на тестовой выборке.

# Теоретическая часть

## Используемые термины

В данном разделе введем, необходимые для дальнейшего описания, термины.

***Speech-to-text*** - в общем случае это некий набор алгоритмов (в большей степени машинного обучения), которые принимают на вход аудио-файл, а на выход выдают строки текста (https://www.silero.ai/what-is-speech-to-text/).

**Датасет** - это обработанная и структурированная информация в табличном виде. Строки такой таблицы называются объектами, а столбцы – признаками .

**Свертка** (в функциональном анализе) - операция, показывающая «схожесть» одной функции с отражённой и сдвинутой копией другой. Иными словами, свертка - это мера перекрывания одной фукнции с отраженной и сдвинутой копией другой. В дискретном случае:

(1)

В случае непрерывных переменных свретка двух функций записывается в виде интеграла:

(2)

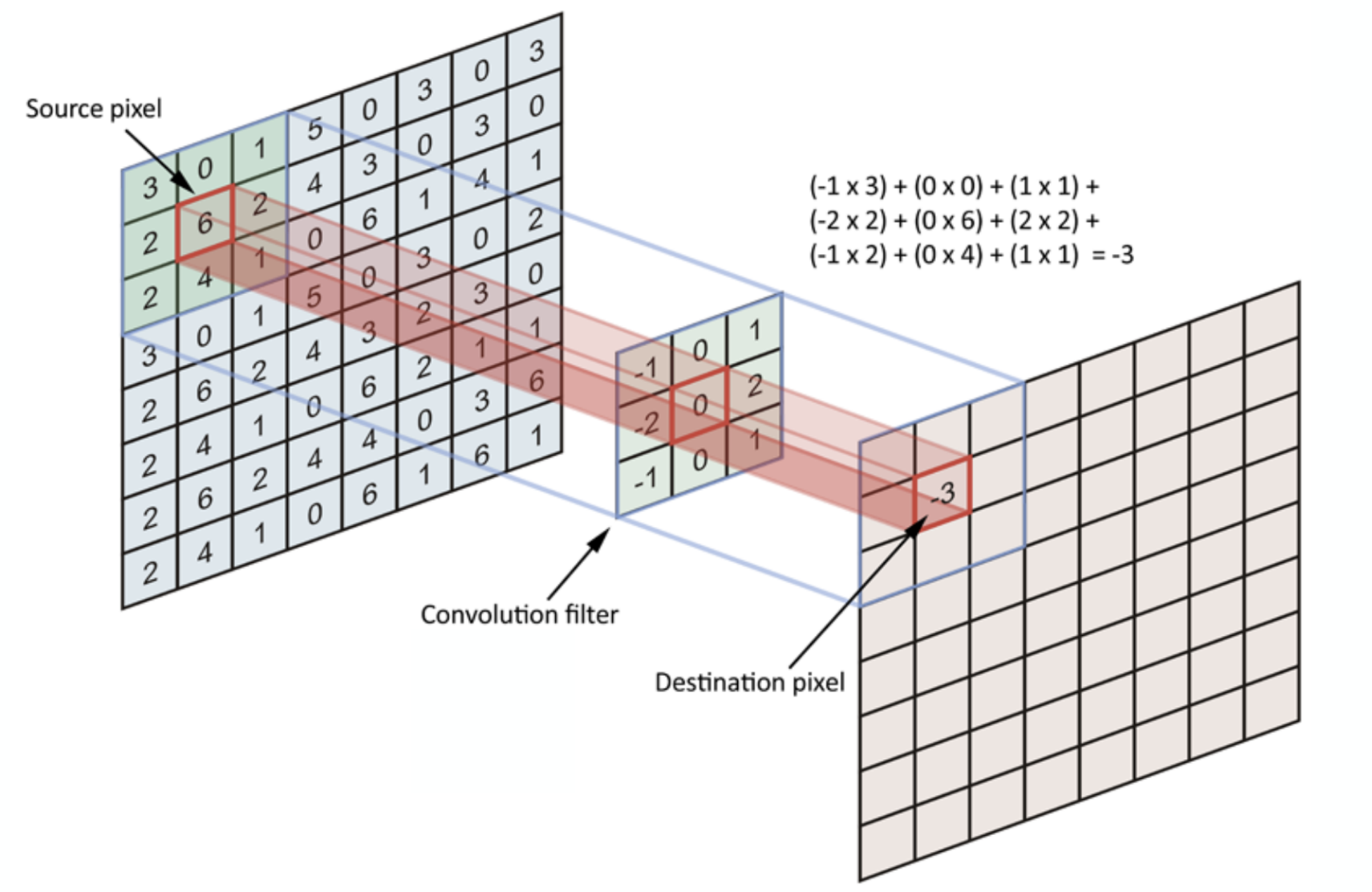
### Операция свертки в задаче распознавания

В нашем случае, так как мы работаем с изображениями, свертка будет выглядеть, как скольжения фильтра свертки по матрице, характеризующей изрбражение для вычисления суммы произведений наложившихся друг на друга элементов. Таким образом будет формироваться матрица, характеризующая новое изображение, получившееся в результате свертки. В матрице, характеризующей изображение значения элементов будут характеризовать интенсивность соответствующего пикселя в градациях серого (т.е. значение каждого элемента такой матрицы будет от 0 до 255).

Свертку изображений можно производить фильтрами разных размеров, и разным шагом. В нашем случае будут использоваться фильтры 3 на 3, с шагом 1, т.е. двигаться такой фильтр будет попиксельно.

Далее рассмотрим конкретные фильтры, которые мы будем использовать.

### Градиентные фильтры



*Рис. 1. Операция свертки изображений*

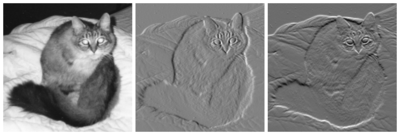
Допустим, нам необходимо посчитать изменение интенсивности пикселей по горизонтали. Очевидным решением будет использовать следующую матрицу (3) для образования горизонтального градиентного фильтра.

(3)

Аналогичным образом, если нам понадобиться считать изменения интенсивности пикселей по вертикали мы предложим для образования вертикального градиентного фильтра следующую матрицу (4).

(4)

Градиентные фильтры представляют важность, так как сильное изменение интенсивности пискелей может говорить и найденных краях (границах) каких-либо объектов на изображении. Результат работы горизонтального и вертикального градиентного фильтров можно увидеть на рис. 2.



*Рис.2 Результаты свертки изображения градиентными фильтрами*

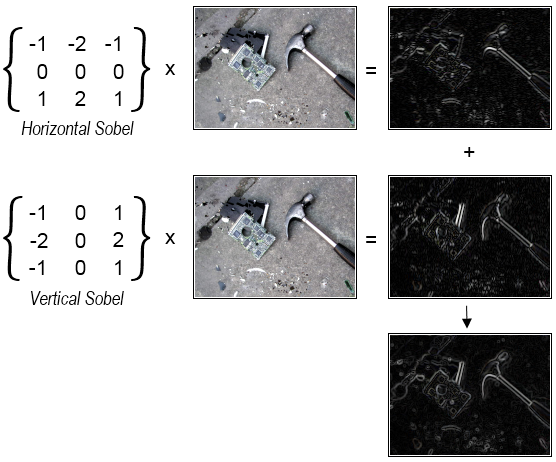
### Фильтр Собеля

Недостаток градиентных фильтров заключается в рассмотрение на каждом этапе свертки учитывает градиент только в одной точке. Фильтры Собеля позволяют это расширить и используют следующие матрицы в качестве ядер свертки: (5) - для вертикальной оси, (6) - для горизонтальной.

(5)

(6)

Результаты работы фильтров Собеля можно увидеть на рис. 3.



*Рис. 3. Результаты работы фильтров Собеля*

## Используемые алгоритмы

В данном разделе будут описаны алгоритмы, использованные для решения поставленной задачи. Для осуществления автоматического сбора датасета для обучения модели, необходимо было сегментировать видеозаписи по словам, выделить область лица и контуры губ на каждом фрагменте. Для нахождения области лица использовались алгоритмы *HOG* (*histogram of oriented gradients* - гистограмма направленных градиентов)и *SVM* (*support vector machine* - метод опорных векторов). Для выделения контуров губ было решено использовать классификатор случайного леса. Данные алгоритмы реализованы в библиотеке *dlib*, которая была использована для реализации.

Также для сегментации видеофайлов по словам была использована библиотка *vosk,* взятые откуда алгоритмы не будут рассмотрены подробнее, так как выходят за рамки поставленной задачи.

### Гистограмма направленных градиентов

Дописать.

### Метод опорных векторов

Дописать.

### Классификатор случайного леса

Дописать.

# Практическая часть

## План решения задачи

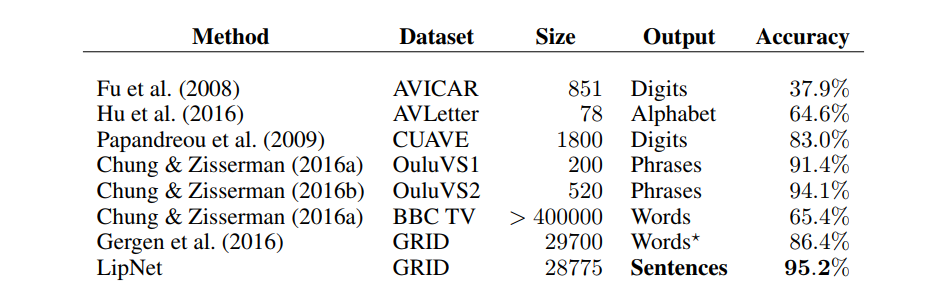
В данном разделе будет коротко рассмотрен план решения поставленной задачи.

Как видно из рис. 4, решение задачи разделяеится на 4 этапа. Рассмотрим подробно каждый из них:

1. Сбор датасета. Важность этого этапа заключается в том, что без его выполнения решить поставленную задачу невозможно. В ходе исследования предметной области был осуществлен поиск существующего датасета, однако, выяснилось, что датасета с движениями губ русскоговорящего человека нет (во всяком случае такой датасет найден не был). Таблицу с описанием существующих датасетов по данной задаче можно увидеть на рис. 5.



*Рис.4. План решения задачи*

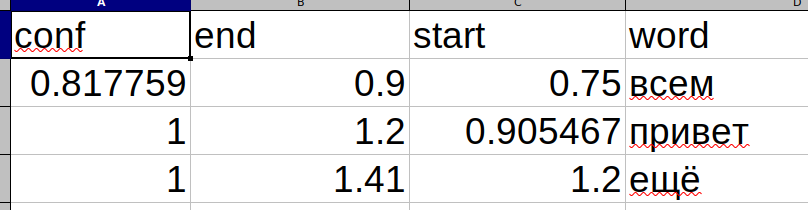


*Рис. 5. Существующие датасеты для обучение моделей для распознавания речи по губам*

1. Анализ данных. Необходимо произвести анализ получившегося датасета, так как возможно благодаря нему, мы сможем скорректировать нашу дальнейшую работу, а также проверить некоторые гипотезы, которые будут полезны для построения архитектуры модели распознавания.
2. Проектирование и реализация модели. На данном этапе необходимо спроектировать архитектуру нейронной сети, для распознавания. Планируется брать за основу архитектуру *CRNN* (сверточной рекурентной нейронной сети). Сверточные слои нам нужны будут для выделения признаков. Потребуется 1d свертка для корреляции во времени, так как в нашей задаче важно, как менялись положения точек губ. Рекурентные же блоки будут необходимы для восстановления адекватной речи и корректировки распознаных слов (или слогов) ранее. Дело в том, что при обучении модели по слогам, сверточные слои будут выделять просто набор слогов. Задача рекурентных блоков будет заключаться в том, чтобы преобразовать это все в корректную человеческую речь.
3. Тестирование модели и анализ результатов. Необходимо провести тестирование модели на реальных данных, выходящих за пределы тестовой выборки. Возможно провести интеграцию в видеочат и сравнивать распознавания речи по губам и по аудиодорожке или же просто проводить сравнения результатов распознавания нашей моделью и классического распознавания по аудиосигналу на готовых видеозаписях.

## Сегментация аудиодорожки

Было принято решение использовать сегментацию по словам для сбора датасета. Было решено использовать библиотеку *vosk* для распознавания речи и выделения иформации по каждому слову. Был написан скрипт, на вход которому подавается видеозапись, а на выходе получается *csv* файл с информацией по каждому распознанному слову. Пример *csv* файла, полученного в ходе работы вышеописанного скрипта можно видеть на рис. 6. Как мы видим, для каждого слова считается веорятность его корректного распознавания (*conf*), время, когда начинается слово (*start*), время, когда заканчивается слово (*end*).



*Рис. 6. Результат сегментации аудиодорожки с видеозаписи по словам*

## Выделение области губ

Выделение губ происходит уже на видеофграментах, получившихся в ходе сегментации видео. На рис. 7 можно схему алгоритма по выделению контуров губ. Для нахождения области лица были использованы алгоритмы *HOG* и *SVM,* а для выделения контуров губ - классификатор случайного леса. Для использования этих алгоритмов была выбрана библиотка *dlib*.

Классификатор случайного леса, для выделения контуров губ хорошо работает при уже найденной области лица, в противном случае - его результаты неточны, а иногда и вовсе абсурдны, что непозволительно для нашей задачи (атоматического сбора датасета). Поэтому для начала мы находим область лица. Для того чтобы это сделать, покадрово прогоним видео и найдем с помощью указанных выше алгоритмов области лица на каждом кадре. Мы столкнемся с проблемой, которая заключается в том, что на некоторых кадрах область лица не будет находится, а следовательно и контуры губ будут выделяться неправильно. Из-за этого могут происходить резкие изменения состояния контуров губ, что даст нам их неправильную скорость, траекторию движения и т.д., что вызовет проблемы как при обучении модели, так и при распознавании. Для решения данной проблемы будем находить среднее положения области лица на кадрах, где лицо в принципе было найдено. Для этого воспользуемся формулой (7), где *i -* это номер оси, а *n -* количество кадров, где было найдено лицо.



*Рис. 7. Схема алгоритма выделения области губ на сегментированных фрагментах вилео*

(7)

После этого область лица найдена и т.к. видеофрагемнты достаточно короткие, то стандартное отклонения распределения соответствующих координат лица невелико и мы можем использовать их средние.

После этого найдем контуры губ на каждом кадре с помощью классификатора случайного леса.

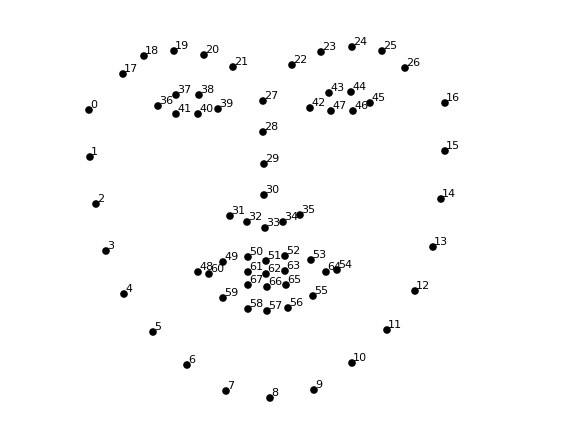
После того, как контуры найдены, нам необходимо вырезать область губ и формировать новый видеофрагмент фиксированного размера для датасета. Для этих целей была использована библиотека *opencv*. Рассмтроим процесс получения нового видеофрагмента подробнее. Во-первых, необходимо осуществить поворот картинки так, чтобы губы располагались вертикально. Для этого просто будем осуществлять поворот вокруг цетра губ (т.е. среднего координат точек контуров губ) на угол, показанный на рис. 8.

*Рис. 8. Поворот контуров губ*



Сделать поворот и другие преобразования мы сможем с помощью результатов, полученных ранее классификатором случайного леса. Библиотека *dlib* позволяет получать информациям по лендмаркам распознанного лица на изображении. В итоге у нас есть информация о координатах нужных нам точек (см. рис. 9). Как мы видим мы можем получить координаты 48 точки и 54, которые будут обозначать края губ и посчитать угол, на которой необходимо повернуть изображения, чтобы контуры губ находились параллельно оси *x*.

Аналогичным образом, мы можем посчитать насколько смещен центр контуров губ от центра изображения и смещать на каждом кадре контуры губ к центру.



*Рис. 9. Лендмарки лица после классификации случайным лесом*

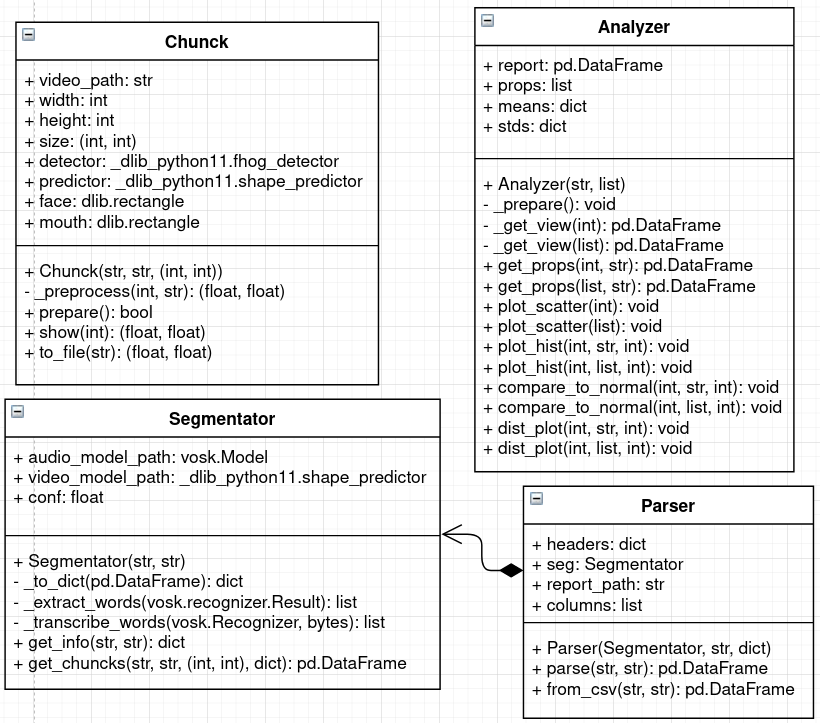
В результате контуры губ будут выглядеть как показано на рис. 10.

*Рис. 10. Скриншоты с фрагментов видео для добавления в датасет*



## Автоматический сбор датасета

Для автоматического сбора датасета была спроектировано и реализоано 4 класса. На рис. 11 можно увидеть *UML* диаграмму этих классов.



*Рис. 11. UML диаграмма классов, необходимых для сбора и анализа датасета*

## Анализ датасета

Дописать.

## Выявленные проблемы

Дописать.

# Заключение

1. Был написан скрпит для автоматического сбора датасета и метаданных.
2. Был написан скрипт для выделения, центрирования, нормирования и т.д.
3. В ходе работы был собран датасет видеофрагментов движений губ, сегментированный по словам русского языка, состоящий из {количество элементов}.
4. Был проанализирован собранный датасет и доказана гипотеза и схожести распределения средних скоростей движения губ одного человека с нормальным распределением.

# Список литературы

1. Паттанаяк, С. Глубокое обучение и *TensorFlow* для профессионалов. Математический подход к построению систем искусственного интеллекта на *Python* / С. Паттанаяк. — СПб : ООО "Диалектика", 2019.
2. Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. / П. Флах. — М : ДМК Пресс, 2015.
3. *Grebler, L. LipNet and Data Fusion Opportunity / L. Grebel* — Текст : электронный // *Medium* : [сайт]. — *URL*: *https://grebler.medium.com/lipnet-and-data-fusion-opportunity-6f4e0166da24* (дата обращения: 15.05.2021).
4. *Sighn, Y. Using AI and Deep Learning for Transcription and Translation / Y. Sighn.* — Текст : электронный // *Medium* : [сайт]. — *URL*: *https://medium.com/@cyogesh56/using-ai-and-deep-learning-for-transcription-and-translation-d7a44176336f* (дата обращения: 15.05.2021).
5. *Multi-Modal Methods: Visual Speech Recognition* (*Lip Reading*). — Текст : электронный // *MLReview* : [сайт]. — *URL*: *https://blog.mlreview.com/multi-modal-methods-part-one-49361832bc7e* (дата обращения: 15.05.2021).
6. *Rosebrock, A. Facial landmarks with dlib, OpenCV, and Python* / *A. Rosebrock.* — Текст : электронный // *pyimagesearch* : [сайт]. — *URL*: *https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/* (дата обращения: 15.05.2021).
7. *Blagojevic, B. How Facial Recognition Works, Part 1: Face Detection* / *B. Blagojevic.* — Текст : электронный // *Medium* : [сайт]. — *URL*: *https://medium.com/ml-everything/much-has-been-written-about-facial-recognition-and-its-role-in-society-2c59e66b9434* (дата обращения: 15.05.2021).
8. *Yiu, T. Understanding Random Forest* / *T. Yiu.* — Текст : электронный // *towardsdatascience* : [сайт]. — *URL*: *https://towardsdatascience.com/understanding-random-forest-58381e0602d2* (дата обращения: 15.05.2021).