ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

«УНИВЕРСИТЕТ “ДУБНА”»

ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА И УПРАВЛЕНИЯ

Кафедра системного анализа и управления

**Курсовая работа**

по дисциплине

**«Теория принятия решений»**

|  |  |
| --- | --- |
| **ТЕМА:** | Распознавание речи по губам на видео без использования информации |
|  | об аудио сигнале 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

[Введение 3](#_Toc72200576)

[Постановка задачи 4](#_Toc72200577)

[Теоретическая часть 5](#_Toc72200578)

[Используемые термины и понятия 5](#_Toc72200579)

[Операция свертки в задаче распознавания 5](#_Toc72200580)

[Градиентные фильтры 6](#_Toc72200581)

[Фильтр Собеля 7](#_Toc72200582)

[Используемые алгоритмы 8](#_Toc72200583)

[Практическая часть 9](#_Toc72200584)

[План решения задачи 9](#_Toc72200585)

[Сегментация аудиодорожки 10](#_Toc72200586)

[Выделение области губ 11](#_Toc72200587)

[Автоматический сбор датасета 14](#_Toc72200588)

[Анализ датасета 15](#_Toc72200589)

[Выявленные проблемы 18](#_Toc72200590)

[Заключение 20](#_Toc72200591)

[Список литературы 21](#_Toc72200592)

# Введение

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

1. Улучшение качества распознавания на готовых видеозаписях.
2. Улучшение качества связи в видео чатах путем автоматической генерации субтитров по распознанной речи с видеосигнала в зашумленных местах.
3. Разработка систем автоматического перевода видеозаписей с языка на язык вместе с технологиями машинного перевода и моделями для решения задач *text-to-speech* (такими как *WaveNet*).

В работе рассматривается:

1. План решения проблемы распознавания речи по видеосигналу без использования информации об аудио сигнале.
2. Сбор и анализ датасета, необходимого для обучения модели распознавания.
3. Выявление основных проблем и возможных способов их решения.

# Постановка задачи

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

Исходные данные: датасет отдельных видеофрагментов с движениями губ, сегментированный по словам и слогам русского языка.

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 рекуррентную 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 и понятия

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.

**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 функции с отраженной и сдвинутой копией другой. В дискретном случае:

. (1)

В случае непрерывных переменных свертка 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 изображение 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.

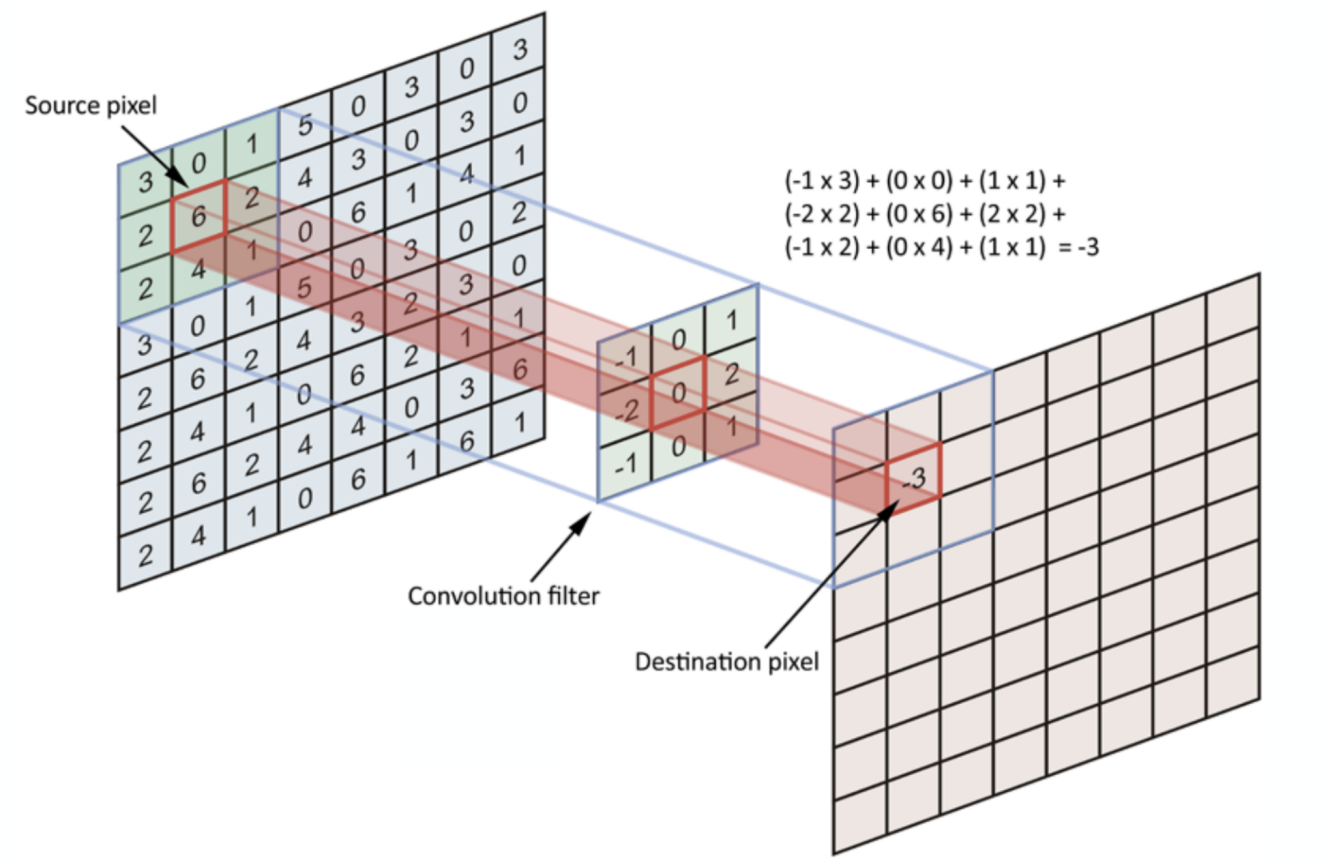


Рис. 1. Операция свертки

Далее рассмотрим конкретные фильтры, которые мы будем использовать.

### Градиентные фильтры

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 пикселей может говорить и найденных краях (границах) каких-либо объектов на изображении. Результат работы горизонтального и вертикального градиентных фильтров можно увидеть на рис. 2.

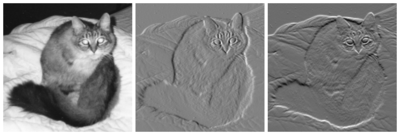


Рис. 2. Результаты свертки изображения градиентными фильтрами

### Фильтр Собеля

Недостаток градиентных фильтров заключается в рассматривании на каждом этапе свертки градиента 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.

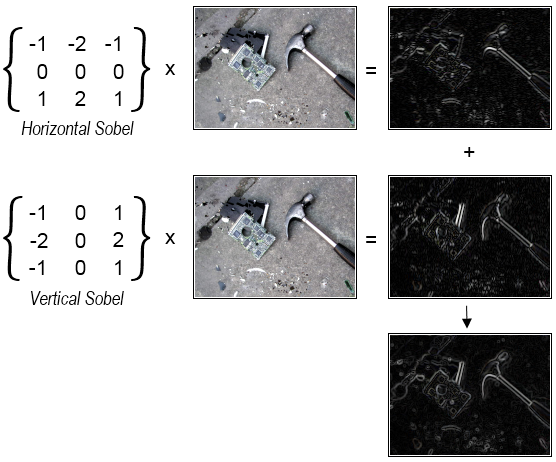


Рис. 3. Результаты работы фильтров Собеля

## 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 библиотека *vosk,* взятые откуда алгоритмы не будут рассмотрены подробнее, так как выходят за рамки поставленной задачи.

# Практическая часть

## План решения задачи

В данном разделе будет коротко рассмотрен план решения поставленной задачи.

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



Рис. 4. План решения задачи

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.

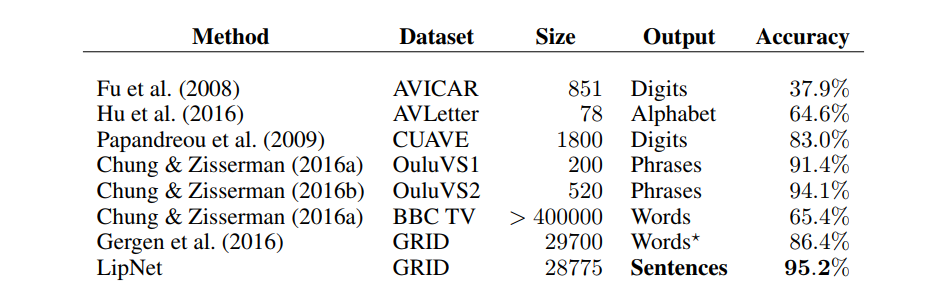


Рис. 5. . Существующие датасеты для обучения моделей распознавания речи по губам

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 рекуррентной нейронной сети). Сверточные слои нам будут нужны для выделения признаков. Потребуется 1*d* свертка для корреляции во времени, так как в нашей задаче важно, как менялись положения точек контуров губ. Рекуррентные же блоки будут необходимы для восстановления адекватной речи и корректировки распознанных слов (или слогов) ранее. Дело в том, что при обучении модели по слогам, сверточные слои будут выделять просто их набор. Задача рекуррентных 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 видео чат и сравнивать распознавания речи по губам и по аудиодорожке, или же просто проводить сравнения результатов распознавания нашей моделью и классического распознавания по аудио сигналу на готовых видеозаписях.

## Сегментация аудиодорожки

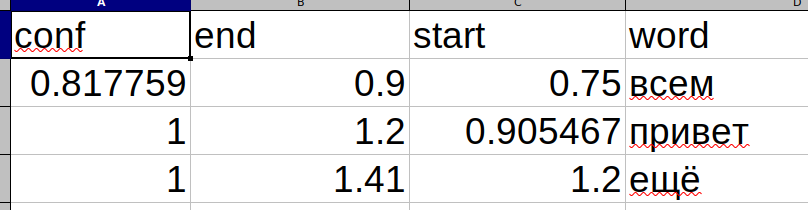


Рис. 6. Результат сегментации аудиодорожки с видеозаписи по словам

Было принято решение использовать сегментацию по словам для сбора датасета. Было решено использовать библиотеку *vosk* для распознавания речи и выделения информации по каждому слову. Был написан скрипт, на вход которому поступает 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 вероятность его корректного распознавания (*conf*), время, когда начинается слово (*start*), время, когда заканчивается слово (*end*).

## Выделение области губ

Выделение губ происходит уже на видеофрагментах, получившихся в ходе сегментации видео. На рис. 7 можно увидеть 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*.



Рис. 7. Схема алгоритма выделения области губ на сегментированных фрагментах видео

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 (автоматического 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.

. (7)

После этого область лица найдена и т.к. видеофрагменты 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.



Рис. 8. Поворот контуров губ

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*. Рассмотрим процесс получения нового видеофрагмента подробнее. Во-первых, необходимо осуществить поворот картинки так, чтобы губы располагались вертикально. Для этого просто будем осуществлять поворот вокруг центра губ (т.е. среднего координат точек контуров губ) на угол, показанный на рис. 8.

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*.

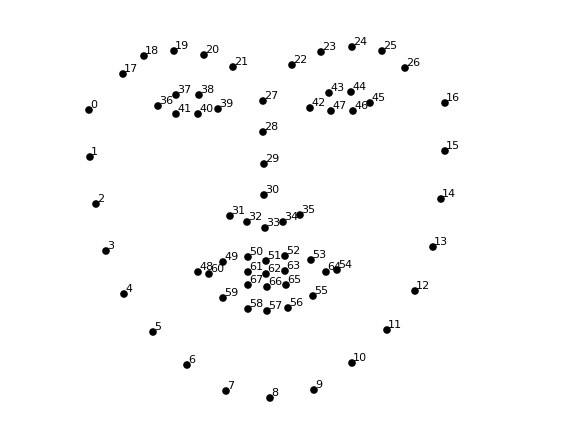


Рис. 9. Лендмарки лица после классификации случайным лесом

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

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 было спроектировано и реализовано 4 класса. На рис. 11 можно увидеть *UML* диаграмму этих классов.

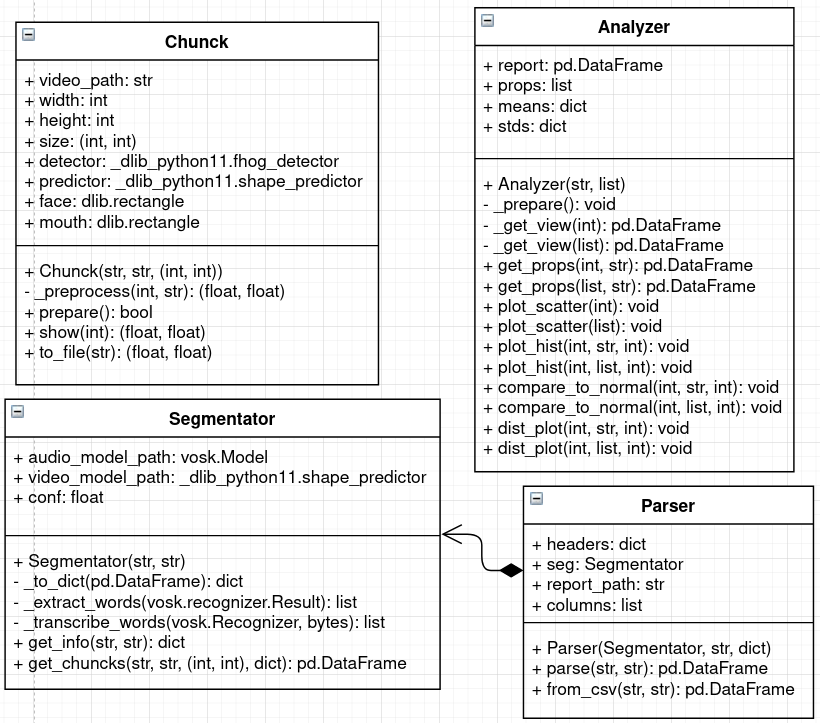


Рис. 11. UML диаграмма классов, необходимых для сбора и анализа датасета

Рассмотрим данные классы подробнее.

1. *Chunck* — инкапсулирует в себе логику работы с отдельными фрагментами видео: позволяет выделять область лица, контуры губ, а также производит необходимую обработку найденных контуров (нормирование, центрирование, поворот) для добавления итогового видеофрагмента в датасет. Также позволяет посчитать среднюю скорость губ на видеофрагменте.
2. *Segmentator* — инкапсулирует в себе логику работы с необработанными видеозаписями для дальнейшей работы класса *Chunck.* Позволяет осуществить сегментацию видеозаписи по словам и получить отдельный фрагменты видео.
3. *Parser* — класс для автоматического скачивания видео с *youtube*. Также позволяет получать метаданные по скачанным видеозаписям, например, автора видео, длину и т.д.
4. *Analizer —* класс, позволяющий удобно анализировать получившийся в итоге датасет. Благодаря данному классу возможно посчитать различные статистические характеристики (математическое ожидание и стандартное отклонение средних скоростей движения губ на видео), проверить распределение средних скоростей движения губ на видео, построить гистограммы и диаграммы рассеивания.

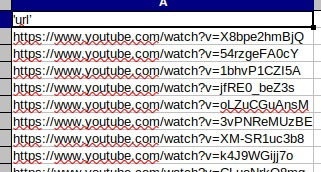


Рис. 12. . Структура csv файла для осуществления автоматического сбора датасета

С помощью классов, описанных выше был написан скрипт, позволяющий по исходному *csv* файлу (см. рис. 12) со ссылками видео на *youtube* автоматически собирать датасет. Итоговую структуру датасета можно увидеть на рис. 13.

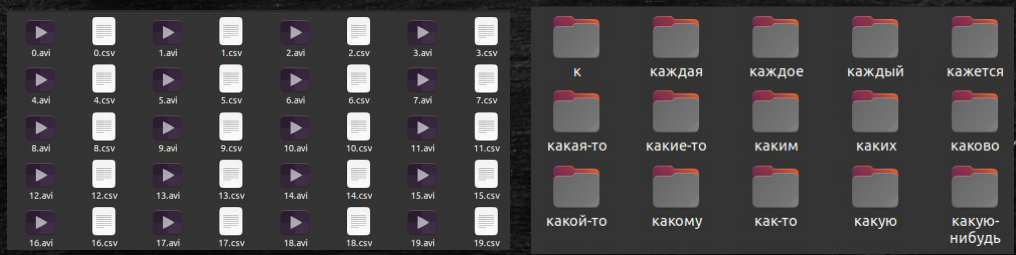


Рис. 13. Итоговая структура датасета

## Анализ датасета

После сбора датасета был проведет его анализ. Необходимо было проанализировать распределения средних скоростей губ, длин произнесенных слов, выяснить насколько коррелирует средняя скорость движения губ с длиной произнесенного слова. Также, в ходе сбора датасета, была выдвинута гипотеза о малом стандартном отклонении средних скоростей движения губ одного человека. Гипотеза была выдвинута исходя из здравого смысла и интуитивного представления о том, что артикуляция у каждого разная и не сильно отличается при произнесении разных слов.

На рис. 14 изображена диаграмма рассеивания средних скоростей движения губ и длин произнесенных слов на выборке видеозаписей с речью одного человека. Как мы можем заметить, гипотеза, упомянутая ранее, подтверждается, так как график позволяет нам сделать вывод о том, что скорость движения губ остается примерно одинаковой.

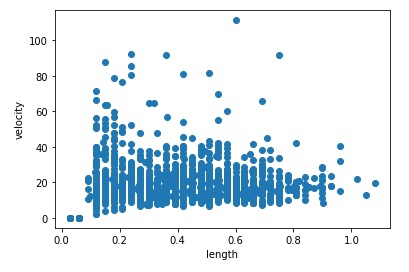


Рис. 14. Диаграмма рассеивания средних скоростей движения губ и длин слов, произнесенных одним человеком

Для более строгого подтверждения выдвинутой гипотезы были посчитаны статистические характеристики средних скоростей движений губ одного человека (8), (9).

, (8)

. (9)

Также была построена гистограмма средних скоростей движений губ (см. рис. 15) для большей наглядности и упрощения исследования распределения.

Как мы видим из диаграммы рассеивания, гистограммы и посчитанных статистических характеристик, выдвинутая гипотеза была подтверждена.

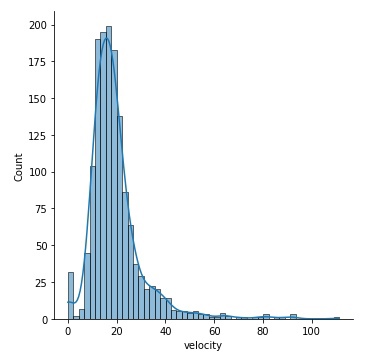


Рис. 15. Распределение средних скоростей движения губ

Было решено провести анализ распределения длин произнесенных слов. Аналогично была построена гистограмма (см. рис. 16) и посчитаны соответствующие статистические характеристики (10), (11).

, (10)

. (11)

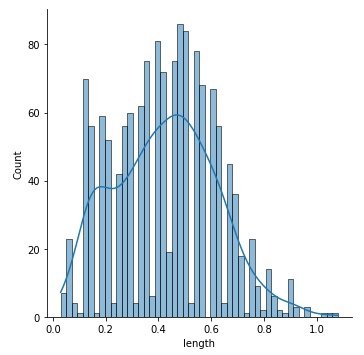


Рис. 16. Распределение длин слов

Однако, как мы видим из построенной гистограммы, математического ожидания и стандартного отклонения, распределение длин слов, гораздо меньше похоже на нормальное, чем распределение средних скоростей движения губ, что далее приведет к ряду проблем, которые будут рассмотрены в следующем разделе.

Хотелось бы отметить, что исследование датасета позволила провести логика, инкапсулированная в классе *Parser* и *Analizer,* т.к. в итоге сбора датасета у нас имелся не только набор видеофрагментов, но и *csv* файл с характеристиками по каждому собранному фрагменту (см. рис. 17).

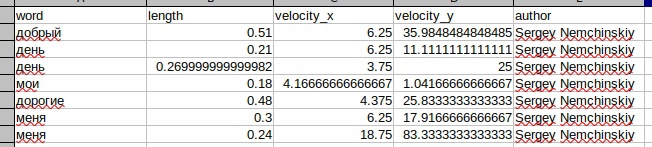


Рис. 17. Csv файл с информацией по каждому фрагменту видео из датасета

## Выявленные проблемы

В ходе анализа датасета были выявлены некоторые серьезные проблемы, которые могут сильно влиять на точность планируемого распознавания, а именно:

1. Корректная сегментация видеосигнала без использования информации об аудио сигнале. Так как ранее было показано, что распределение длин слов имеет распределение не похожее на нормальное и стандартное отклонение достаточно велико, мы не можем сегментировать видеосигнал для дальнейшего распознавания по среднему. Мы также не можем прибегать к тому же способу сегментирования, что и при сборе датасета, так как по постановке задачи, мы не можем использовать информацию об аудио сигнале. Для решения данной проблемы планируется собрать новый датасет с сегментацией по слогам, так как распределение их длин гораздо больше похоже на нормально и позволит более точно осуществлять сегментацию по средней длине.
2. Проблемы корректного распознавания в реальном времени. Ранее было описано, что для корректного определения контуров губ на каждом кадре необходимо прогонять видеофрагмент 2 раза. Однако, в случае распознавания в реальном времени — это невозможно. Как следствие, мы теряем точность распознавания, потому что, либо часть кадров остается без определения контуров губ, либо с их некорректной детекцией. Для решения данной проблемы планируется проверить, насколько точно работают другие алгоритмы детекции лица и определения контуров губ и найти вариант, который позволяет делать это сразу корректно (но, при этом, практически без задержки).
3. Проблема необходимости огромного датасета при сегментации по словам. При выполнении сбора датасета было решено сегментировать речь по словам, что приводит к необходимости огромного размера датасета для корректного обучения модели распознавания (т.к. если слова нет в датасете, распознано оно не будет). Кроме того, для модели остается принципиально невозможным распознавание личных неологизмов каждого человека, так как невозможно использовать их на этапе сбора датасета. Для решения этой проблемы планируется собрать новый датасет с сегментацией по слогам, так как количество слогов гораздо меньше количества слов и, кроме того, позволяет распознавать слова, которые невозможно включить в датасет при обучении модели.
4. Проблемы невысокой точности распознавания за пределами тестовой выборки. С данной проблемой остро сталкиваются уже имеющиеся модели распознавания (такие как *LipNet*). Несмотря на возможный высокий уровень точности распознавания на тестовой выборке, на реальных данных модель будет работать значительно хуже. И если у *LipNet* такая проблема возникает из-за искусственно созданного датасета в хороших лабораторных условиях (человек смотрит строго в камеру, не мотает головой, имеет достаточно четкую артикуляцию), то в нашем случае данная проблема будет вызвана менее точной сегментацией и самим распознаванием слов, которых было либо мало в датасете, либо вообще не было (т.е. проблемами 1, 2 и 3).

# Заключение

В заключении хотелось бы перечислить то, что уже было сделано в рамках работы над проектом.

1. Был написан скрипт 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.
4. Был проанализирован собранный датасет и доказана гипотеза о схожести распределения средних скоростей движения губ одного человека с нормальным распределением.

Также отдельно хотелось бы упомянуть планы по развитию проекта, такие как:

1. Решение проблем, описанных в разделе «Выявленные проблемы».
2. Проектирование и реализации модели распознавания, архитектура которой будет основана на *CRNN*.
3. Обучение модели, тестирование и анализ результатов.
4. Интегрирование работающей модели в видео чаты и видеоплееры для более точного автоматического генерирования субтитров.
5. Разработка системы автоматического перевода с языка на язык, с использованием спроектированной модели, моделей машинного перевода и нейронных сетей (таких как *WaveNet*) для решения задачи *text-to-speech*.

# 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).