ЧТЕНИЕ ПО ГУБАМ РУССКОЙ РЕЧИ НА ОСНОВЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ ЛИЦ

Г. С. Афанасенко

Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь, mmf.afanasenGS@bsu.by
Научный руководитель — А. Э. Малевич, кандидат физико-математических наук, доцент

В данной статье описываются этапы создания системы для чтения по губам русской речи, а также уделяется внимание различным техническим особенностям. Подробно описан процесс проектирования и создания автоматизированной системы для сбора и разметки обучающих данных, а также процесс обучения модели.

Ключевые слова: чтение по губам; компьютерное зрение; обучающий набор; разметка данных; нейронные сети; 3D-свёрточные слои.

ВВЕДЕНИЕ

Задача чтения по губам является достаточно сложной как для людей, так и для решения с помощью технологий компьютерного зрения. Данная задача имеет множество применений, например, в медицине для людей, имеющих проблемы с голосом; для улучшения точности распознавания голоса голосовыми ассистентами с помощью добавления визуальной информации.

Предыдущие исследования по данной теме по большей части касались только английского языка, поэтому в данной статье рассматривается применение этих исследований на русскую речь. Кроме того, в открытом доступе нет достаточно больших и разнообразных датасетов с видео русскоговорящих людей, поэтому в рамках этой работы такой датасет требуется собрать.

СБОР ДАННЫХ

Поскольку поставленная задача является сочетанием задачи компьютерного зрения, а также задачи обработки естественного языка, то требуется большой объём данных для обучения, так как даже обычная языковая модель требует большого количества данных для того, чтобы выучить все нюансы языка. Предыдущий обучающий набор данных LRWR для данной задачи был собран и упомянут в [1]. Однако в данном датасете обучающие примеры представляли собой отдельные слова, а одной из целей данной работы является создание датасета с обучающими примерами, состоящими из целых предложений.

Для начала требуется определить требования к обучающим примерам, которых нужно добиться. В условиях данной задачи были определены следующие требования:

- Ровное положение головы относительно границ кадра.
- В каждом кадре должно быть только лицо без лишних областей.
- Частота кадров должна быть выше 25, а также высокое разрешение – 720p и выше.

Как уже упоминалось, данных должно быть много, следовательно собирать их нужно не вручную, а автоматизировать весь процесс. Аналогично [1] будем брать неподготовленные данные с интернета, а уже затем подготавливать их для обучения. В качестве основного источника взят видеохостинг youtube.com. На нём достаточно много видео в требуемом или похожем формате: подкасты, влоги, блоги и прочее. Неподготовленные данные представляют собой длинные видео одного из описанных выше типов. Далее используем это видео для генерации небольших обучающих видео. Опишем процесс обработки данных схематично (рис. 1).

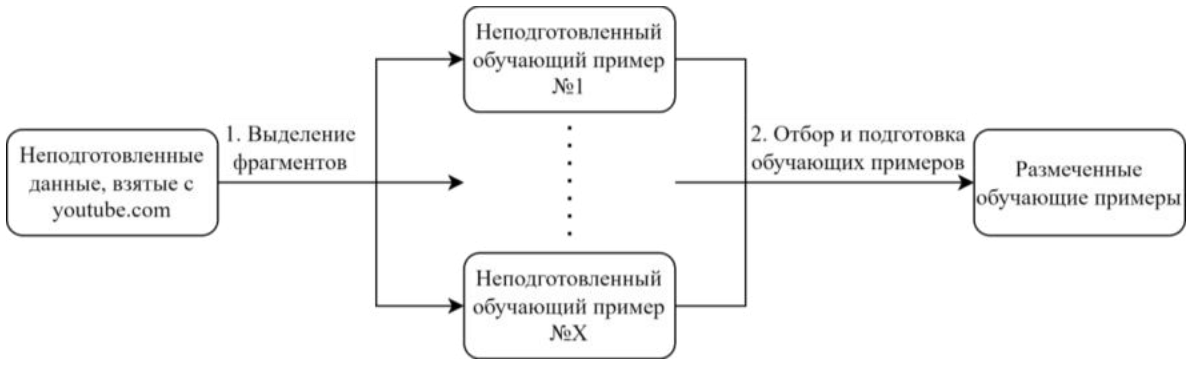

Рис. 1. Схема работы системы для разметки данных

Цель первого этапа заключается в том, чтобы разделить длинное видео на небольшие фрагменты одинаковой длины [2]. Изначально фрагменты были по 10 секунд, однако в результате дальнейшего анализа и экспериментов стало ясно, что такая длина слишком велика, и фрагменты получаются перенасыщенными речью спикеров. Поэтому далее длина фрагментов была сокращена до 5 секунд.

На втором этапе каждый полученный фрагмент нужно проверить на годность к дальнейшей разметке и далее разметить. Качество размеченных данных зависит от качества начальных данных, а также качества работы размечающей системы. Поскольку начальные данные выбираются вручную, то требуется сделать так, чтобы их качество меньше всего влияло на размеченные данные. Следовательно, требуется сделать систему отбора и разметки данных достаточно гибкой.

В процессе отбора требуется проверить, что фрагмент содержит лицо спикера на протяжении всех кадров, а также, что это лицо не повёрнуто сильно вбок. Если эти условия выполнены, то следующими шагами являются: поворот кадра таким образом, чтобы лицо находилось ровно, и последующее кадрирование кадра по границам лица спикера. В завершении нужно транскрибировать аудио из видео и записать текст в отдельный файл.

Все скрипты были реализованы, используя Python 3. Для работы с видео и аудио использовался инструмент FFmpeg и библиотека OpenCV, а для транскрибирования библиотека SpeechRecognition. Для более быстрой обработки важно использовать ускорение FFmpeg с помощью видеокарты.

ПРЕДСКАЗАТЕЛЬНАЯ МОДЕЛЬ

Выбор нейронных сетей в качестве предсказательной модели очевиден. Прочие варианты, такие как скрытые марковские цепи, уже устарели. Архитектурой модели будет LipNet [3]. Данная архитектура состоит из двух блоков: блок слоёв 3D-свёртки и пара рекуррентных Bi-GRU слоёв. Одна из особенностей данной архитектуры – это использование CTC Loss в качестве функции потерь. Основная идея данной функции потерь состоит в том, чтобы учитывать следующие факторы при распознавании речи: некоторые звуки могут растягиваться спикером, иногда спикер может допускать неравномерные и неодинаковые паузы при произношении.

Поскольку текущий собранный датасет достаточно крупный, то для тестирования реализации LipNet использовался датасет LRWR [1], который был предоставлен в исследовательских целях. Для быстрого обучения на мощных видеокартах использовался облачный сервис Google Colaboratory Pro. Однако даже в такой связке обучения модели заняло много времени, из-за чего модель до сих пор находится в состоянии обучения. На данный момент можно посмотреть, каким образом обучились различные каналы первого свёрточного слоя модели (рис. 2).

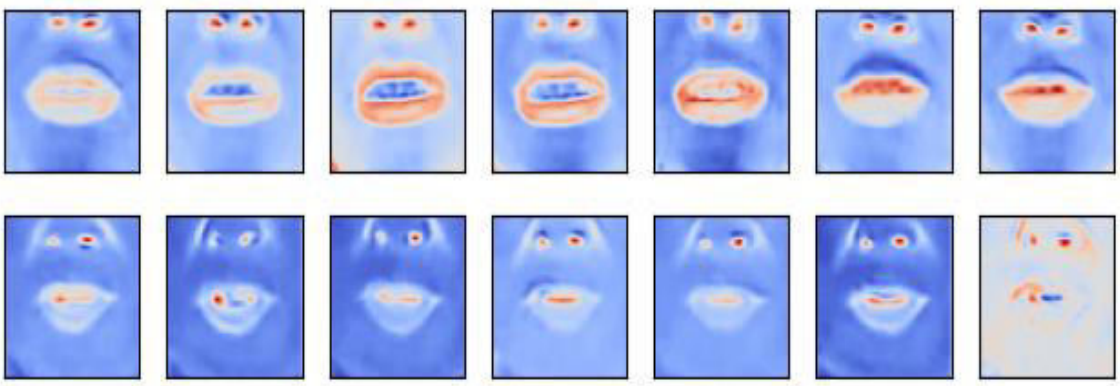

Рис. 2. Возбуждение свёрточного слоя на входные данные.
Красным цветом помечены области, которые больше всего возбудили модель.

Реализовывалась и обучалась модель с использование библиотеки PyTorch 2.3 и ряда других вспомогательных библиотек из Torch-семейства. Стоит отметить, что оригинальная модель из [3] была реализована с использованием TensorFlow, то есть в рамках данной работы архитектура LipNet была полностью переведена на PyTorch 2.3 с учётом всех тонкостей этой библиотеки. Для более качественного обучения начальная инициализация весов нейросети проводилась вручную с применением различных методов для каждого отдельного слоя.

ЗАКЛЮЧЕНИЕ

В результате данной работы удалось создать достаточно гибкую систему и минимизировать зависимость качества размеченных данных от качества начальных данных. Тем самым это позволяет с лёгкостью масштабировать обучающую выборку. Также стоит отметить, что реализованная система может быть полезна и в некоторых других задачах аудиовизуального распознавания, что делает эту систему ещё более полезной. Также реализована архитектура LipNet с использованием библиотеки PyTorch 2.3.

Библиографические ссылки

1. LRWR: Large-Scale Benchmark for Lip Reading in Russian language [Electronic resource] / E. Egorov [et al.] // arXiv, 2023. URL: https://doi.org/10.48550/arXiv.2109.06692 (date of access: 17.06.2024).

2. Command Line Video Splitter [Electronic resource] // GitHub, 2022. URL: https://github.com/c0decracker/video-splitter (date of access: 17.06.2024).

3. LipNet: End-to-End Sentence-level Lipreading [Electronic resource] / Y. M. Assael [et al.] // arXiv, 2023. URL: https://doi.org/10.48550/arXiv.1611.01599 (date of access: 17.06.2024).