Visual Speech Recognition with Convolutional Neural Network

Adam Kropiwnicki

22 October 2021

Project supervisor: Dr inż. Witold Paluszyński
Wroclaw University of Science and Technology
Faculty of Electronics, Photonics and Microsystems
Class: AIR AER

1 Abstract

The goal of this project was to develop an Artificial Intelligence capable of recognizing human speech based on only visual features. Such ability is commonly known as 'lip-reading'. Training process has been based on data from the LRS3 dataset, which consists of links to videos on a YouTube platform paired with labeling data describing their content. The main objective was to recognize whole words spoken in English. The achieved accuracy of predictions is satisfactory, but complexity of the project has been limited to a small set of words.

2 Introduction

Lip-reading is a complex task. Even after limiting ourselves to English language, there are over 170 000 words to be distinguished and that number only accounts for words frequently used. Many of them sounds differently but are represented by the almost identical set of lip movements. Similarly to the classic speech recognition problem, lip readers also come across the issue of speakers articulating words with different speed and different accent. In addition to all of that, the speaker's face can be viewed from many angles, and this angle may (and most likely will) vary in time during the speech. While human lip-readers can identify most of the speech while looking straight to the speaker's face, even the most prominent of them could not achieve more than 15% accuracy of predicted words when asked to recognize speech from the video. Cutting-edge Machine Learning algorithms are performing significantly better on this field, with Oxford/Google's LipNet architecture correctly inferring up to 48% of speech [3].

Due to limitations in both time and computing power, the speech recognition capability of the project has been downsized to only 5 arbitrarily chosen words: ABOUT, DIFFERENT, EVERY, GOING, HUMAN. This small set consist of words that have considerably distinct lip movement from each other.

This project has been written in Python language, with usage of Tensorflow/Keras libraries for neural network building and training, and OpenCV library for handling video and camera input processing.

3 Training data

3.1 LRS3 dataset

Project uses LRS3 dataset for model training. This dataset is based on publicly available recordings of TED Talks and consist of label text files providing the information about those records such as:

- Text of a spoken sentence
- link to a YouTube video from which sentence is taken
- Which frames in video contain the moment of articulating related sentence
- Bounding boxes of speaker's face in each frame in which he/she articulates the sentence
- Timestamps of each word in the sentence

Those recordings last about 1 hour on average. Each recording have a one label file for every uninterrupted period of time in which camera shows the speaker's face while he performs. Those label files are grouped in directories conveniently named after video IDs.

3.2 Data preprocessing

In order to convert provided label text files into viable labeled input for the model, data preprocessing pipeline has been created. It is actually builded from 3 seperate pipelines: Video downloading, Video proccessing and Training data fetching pipelines.
Video downloading pipeline operates on each label file in a following way:

1. Referenced video is downloaded from YouTube
2. If video has framerate different than 25 fps, it is converted to 25 fps. This step is required since frame labels of LRS3 dataset assume all videos have this particular framerate.
3. Based on provided labels, irrelevant frames are erased from video
4. Using provided face bounding boxes, each frame is resized to contained only speaker's face
5. Saves new video containing only processed frames.

Current iteration of video processing pipeline operates on each video generated in previous pipeline in a following way (Figure 1):

1. Upper 60% of each resulting frame is thrown away based on the fact, that almost all the people have their mouth area in the lower 40% of the face.

2. Mouth detecting classifier (pretrained Haar Cascade Classifier from OpenCV library) is applied to the remaining lower 40% of the face
3. If classifier have not found the mouth area for particular frame, it is taking it from the previous frame
4. Last but not least, extracted mouth area is reshaped to a 32x24 picture and converted to grayscale image.
5. Saves new video containing processed frames.

Training data fetching pipeline is capable of generating single word input samples from video files created in Video processing pipeline. It searches through the text labels to find the requested words and cuts specified by label video sample from relevant video file. Then, each such sample is scaled to contain only 12 frames, assuming each word takes about 0.5 seconds. Too short samples have their first frames repeated at the end, too long samples have some of their frames cut out (uniformly across the whole sample). Such sample, together with it's label, is ready to be fed to the model.

(a) Original frame

(b) Face from labels

(c) Extracted mouth

(d) Reshaped and converted input

Figure 1: Stages of video processing pipeline: (a) - Raw frame taken from video, (b) - Face frame based on provided bounding box, (c) - Mouth area extracted with Haar Cascade Classifier, (d) - Mouth area converted to 32x24 shape and converted to grayscale

4 Model

The currently best working neural network architecture is shown on Figure 2
All layers except for last one are using ReLU activation function. Last layer uses a softmax activation function to provide probability distribution output. Model utilises Adam optimizer with learning rate set to 0.0001 which has been tested to give the best learning time to accuracy ratio.

5 Results

The model achieved 78% percent accuracy on the test data. On top of that, model was evaluated empirically using a webcam. Evaluation program runs in a loop, feeding every 12 frames gathered from webcam to a model and printing its prediction. This approach has a major flaw: more often than not word will not fit perfectly a 12

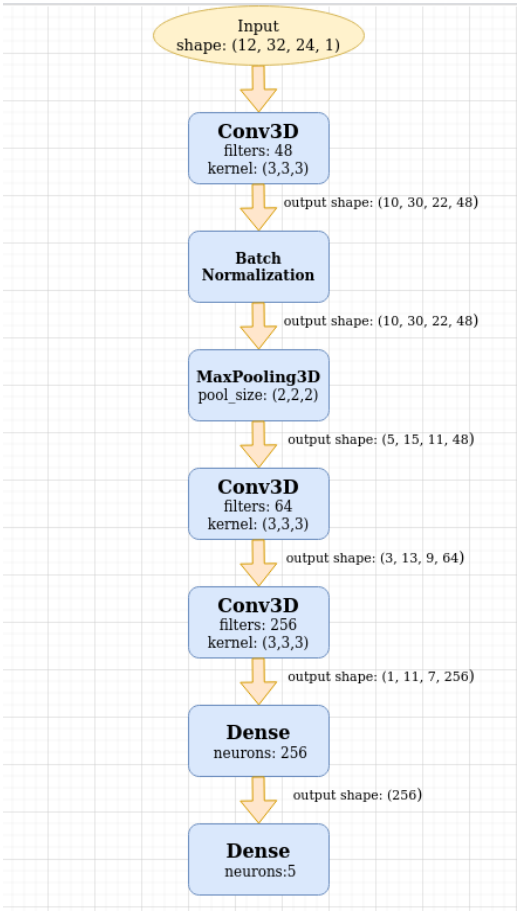


Figure 2: Model architecture

frame window, since we don't really control when the window starts and when it ends. To somewhat mitigate this problem user is required to repeat the word constantly, trying out different articulation speeds until he sees either model starting to correctly recognize his word most of the time, or that it fails to do so. Hence we cannot apply standard evaluation metrics here.

Nonetheless, the model learned to recognize words ABOUT, EVERY, GOING and HUMAN even at various articulation speeds and from various viewing angles. The word DIFFERENT proves to be more difficult, but under right circumstances (direct view at the face and synced articulation speed) the model stills recognizes this word correctly most of the time and with good confidence. This results are satisfactory considering that it makes correct predictions on faces that were not included in the training data. Example is shown on Figure 3

It is worth mentioning, that this model would probably work equally fine for a much larger set of words, which was not tested due to lack of time. Unfortunately this method could never work for a whole dictionary of English words due to sheer amount of them. For end-to-end speech recognition a different approach is required.

Code written for the purpose of this project is available here: https://github.com/Cropka/lipread/tree/master

6 References

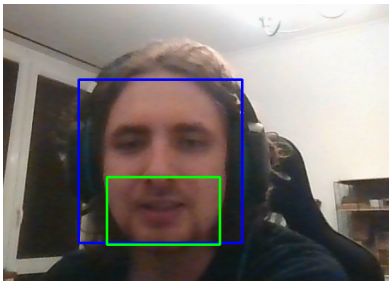
1 T. Afouras, J. S. Chung, A. Zisserman
LRS3-TED: a large-scale dataset for visual speech recognition
arXiv preprint arXiv:1809.00496

2 https://portfolios.cs.earlham.edu/wp-content/uploads/2019/08/CS488$_K aran_F inal_P aper.pdf$

3 https://arxiv.org/pdf/1611.01599.pdf

4 https://keras.io/about/

5 https://docs.opencv.org/4.x/d1/dfb/intro.html



(a) User's tracked face and mouth

```
[('GOING', 0.5218541), ('HUMAN', 0.37667638), ('ABOUT', 0.05538417), ('EVERY', 0.034426074), ('DIFFERENT', 0.011659265)]
[('GOING', 0.5953045), ('HUMAN', 0.3147661), ('ABOUT', 0.050655205), ('EVERY', 0.029750621), ('DIFFERENT', 0.009523531)]
[('GOING', 0.6043816), ('HUMAN', 0.306042), ('ABOUT', 0.046269506), ('EVERY', 0.03511739), ('DIFFERENT', 0.008189559)]
[('GOING', 0.53328055), ('HUMAN', 0.38902012), ('ABOUT', 0.04144131), ('EVERY', 0.031051947), ('DIFFERENT', 0.005206043)]
[('HUMAN', 0.8371861), ('GOING', 0.115630805), ('DIFFERENT', 0.017623298), ('EVERY', 0.016106525), ('ABOUT', 0.013453297)]
[('EVERY', 0.43220535), ('GOING', 0.29405192), ('ABOUT', 0.1690914), ('DIFFERENT', 0.09677481), ('HUMAN', 0.007876552)]
[('EVERY', 0.4401953), ('GOING', 0.21877237), ('DIFFERENT', 0.16527125), ('ABOUT', 0.16208495), ('HUMAN', 0.01367618)]
[('EVERY', 0.50443137), ('GOING', 0.18177234), ('ABOUT', 0.1516027), ('DIFFERENT', 0.14750582), ('HUMAN', 0.01468782)]
[('EVERY', 0.46082437), ('GOING', 0.23926505), ('ABOUT', 0.15700908), ('DIFFERENT', 0.12258372), ('HUMAN', 0.020317715)]
```

(b) Printed predictions

Figure 3: Model evaluation program: (a) - User providing input to the model, (b) - Predictions from model. Leftmost word is a top model prediction. User was requested to repeat a word GOING for a few seconds, then switch to a word EVERY for another few seconds. We can observe a mistake in classification during the switching period

7 Licence