Real-time Lipreading: Effects of Compression and Frame-rate

Master's Thesis

Riya Maan

Real-time Lipreading: Effects of Compression and Frame-rate

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK DATA SCIENCE

by

Riya Maan
born in India

Interactive Intelligence
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
http://ii.tudelft.nl

Real-time Lipreading: Effects of Compression and Frame-rate

Author: Riya Maan
Student id: 4719913
Email: riyamaan.aqua@gmail.com

Abstract

Speech recognition systems can be found all around us. From personal assistants in mobile phones and smart speakers to robots, we use speech recognition systems everyday. However, communicating with them can be troublesome in noisy environments because they only use audio signals for speech recognition. This problem can be solved by using visual speech recognition or lipreading systems. Research on lipreading systems has been going on since the 1980s but such systems are not being used in real-time systems yet. This can be attributed to the fact they need to process significantly higher amounts of data than audio speech processing which takes a lot of time and hence, they cannot be used in real-time.

This thesis aims at finding out if frame rate, jpeg compression or presence of noise have any impact on the performance of lipreading system. The LipNet system is used for this thesis and the Lip Reading in the Wild (LRW) dataset is used for the purpose of experiments. The frame rate of videos of the dataset is varied from 11 to 25, with an increment of 2 for each experiment. Also, compression ratio is varied between no compression and 30 % quality, to find out how compression affects the performance of lipreading systems. Also, salt and pepper noise is artificially added to the dataset for the purpose of experiments.

The results from the experiments showed that performance is not affected till frame rate 21, but it starts degrading gradually from frame rate 19 to 13 and after that there is sudden drop in the accuracy of LipNet. With compression of frames to 30 percent of their original quality, there is only a slight decrease in accuracy. However, there is a huge reduction in data size, which makes it easier to transmit data for cloud processing. We found substantial degradation in performance with the presence of noise with a probability of only 3 percent.

This means that if we decrease frame rate to 21 and compress the frames to 30 % quality, memory usage can be decreased to 25 % without much impact on performance of the system. However, quality of video capturing cameras and data transmission to cloud needs to be monitored to avoid noise.

Thesis Committee:

Chair:	Prof. dr. ir. Catholijn M. Jonker, Faculty EEMCS, TUDelft
University supervisor:	Dr. ir. Joost Broekens, Faculty EEMCS, TUDelft
Committee Member:	Dr. ir. Hayley Hung, Faculty EEMCS, TUDelft

Preface

This thesis is written as a part of my masters programme in Computer Science at TU Delft. It is a documentation of my work done from November 2018 to August 2019. This report aims at finding out means of making real-time lipreading computationally less expensive while maintaining the performance of the system.

For this, I would like to express my sincere gratitude to my supervisor, Dr. Joost Broekens, for his constant guidance and support throughout the process.

I would also like to thank my parents and brother for always supporting and encouraging me. Special thanks to my mom for always being there for me. I also wish to thank my friends for their support and being a family away from home.

Riya Maan
Delft, the Netherlands
August 26, 2019

Contents

List of Figures

Chapter 1

Introduction

Do you remember the time when you tried to read lip movement of your friend to share a secret or to share views about your manager with your colleagues? Not everyone is good at reading lips, but we try to do it unconsciously on a daily basis. Sometimes reporters try to read lips of celebrities to know what they're talking about, in case speech is not audible or the background is noisy. Also, while watching a poorly dubbed movie, the misalignment of audio and video makes it difficult for us to understand the otherwise clean audio signal. This is because we try to match the lip movements of actors to audio and we have trouble understanding the speech if they are misaligned.

Lipreading or visual speech recognition refers to understanding what a person is saying by observing movement of their lips and tongue. It is an important form of communication for deaf people and helps them understand what the other person is saying. For people with impaired hearing, when machines can't help them to listen the speech of the other person, for example in noisy environments, lipreading is helpful in conversation. However, they can't read lips with 100 percent accuracy, in which case, context can be used to fill in the gaps [12].

Even for people who do not have any hearing disability, lipreading is an important modality in conversation. We always correlate lip movements to the audio speech. It can be explained by the McGurk effect [20], which states that when different audio is dubbed over a video, it leads to perception of mismatched phonemes.

Humans can recognize 35 to 40 percent of speech using only lipreading [17], the remainder can be partially understood by context. But the performance of human lipreaders is still unsatisfactory and lipreading needs to be automated. Researchers identified this problem and started working on it in 1980s. It started slowly but the research caught off in late 1990s. With the advent of HMM-GMM decoders, the performance of lipreading systems improved drastically. HMM-GMM refer to Hidden Markov Model - Gaussian Mixture Model, in which GMM is used to recognize phoneme corresponding to a frame and HMM encodes the temporal aspect of speech. GMM recognizes phonemes by training a GMM model corresponding to each phoneme and then finding out the most probable phoneme for each frame, however it does not take temporal aspect into account, for which HMM is used. HMM breaks the speech in multiple states and then trains GMM models for each state according to the occurrence of phonemes in that state.

The recent development of deep learning networks proved to be another major

breakthrough for lipreading systems and eliminated the use of language models for visual speech recognition.

With the advent of personal assistants, audio speech recognition has become an important part of our day-to-day life. However, communicating with them is still troublesome. Especially when you have a lot of people talking around, it's difficult to get your message to the personal assistant. This problem is even more important when we have robots around us. To remove the bottleneck of robot cocktail party effect from audio speech recognition, we would need audio-visual speech recognition systems, which also take visual features into account when deciphering the speech.

These developments mean that we would need more processing power, as processing a video requires far more resources than processing audio alone. Robots and personal assistants use cloud-based systems for speech recognition [16]. For audio speech recognition, it only requires transmitting audio to the cloud. However, transmitting both audio and video means that we need more resources and time for speech recognition, which would make these systems difficult to be used in real-time. This problem can be solved by compressing the video for transmission or decreasing the frame-rate of the video stream, so that less data needs to be transmitted. Varied quality of cameras might also result in noisy and low-frequency video stream at the cloud. As per the best of my knowledge, the effect of these factors on the performance of lipreading systems has not been tested.

In this thesis, I assess how frame-rate, noise and compression of video affect the performance of lipreading systems. This study provides an understanding of applicability of video speech recognition for real-time systems.

1.1 Outline of report

The whole report is divided into 5 sections. The coming section discusses the literature on lipreading systems. Section 2 establishes the research questions which will be answered in this report. Third section explains the method used to perform experiments. It explains the model used in experiments and various dataset configurations. It also informs about experiment setup in brief and the measures used to evaluate the results and answer research questions. The next section reports results from experiment and discusses how they can be used to answer research questions. The last section concludes the report.

1.2 Background study

In this section, advancements of various speech recognition systems are discussed. Lip Reading and Audio Visual Speech Recognition are closely related problems. But both of them have vast literature and that needs to be dealt separately. An overview of different data-sets available for the training of speech recognition systems is also provided.

Automatic Speech Recognition

Lipreading: Initial work on lip reading did not include deep learning approaches [30]. Most of the researches used to process video frames to extract lip features, the temporal aspect was also explicitly extracted from the video frames. These carefully selected features were then used to train systems to identify speech [19, 24, 18].

Recently researchers have started using CNNs to detect characters [3], words [8] and sentences [28]. CNNs, convolutional neural networks, are used to recognize objects and classify objects in one of several categories. They consist of a combination of convolution layers, which can comprise of filtering, pooling and Rectified Linear Units (ReLUs). A ReLU layer applies the function $f(x) = max(0,x)$ to all the elements of the matrix, which means that all the negative values are changed to 0 and the postive values remain same. Chung et al. [8] used CNNs to directly classify multi-frame time series of lips extracted from videos. While Tatulli et al. [28] used ultrasound imaging in addition to video camera for visual features. These visual modalities were processed using a multi-modal CNN, combined with an HMM-GMM decoder for sentence prediction.

LipNet [3] extracts features using spatiotemporal convolution layers, these features are then aggregated by two Bi-GRUs (Gated Recurrent Units, a type of RNN). Spatio-temporal CNN is 3D CNN, which means that the filter matrices are 3D. Two dimensions correspond to space and the third dimension is time dimension. Bi-GRUs are a set of opposite directional GRUs, which are a modification of RNN. Lastly, word prediction is done using CTC loss on existing vocabulary. CTC Loss is one of the most commonly used loss functions in speech recognition systems. This loss function does not need aligned output for training of the model, while it takes summation over all possible permutations of output sequence. Detailed explanation of these layers is given in chapter 3. Shillingford et al. [27] used open-vocabulary data-set to train their Vision-to-Phoneme (V2P) 3D convolutional module, whose architecture is similar to LipNet. 3D convolutional module works in similar manner as spatio-temporal convolutions in LipNet, explained in the next section.

Audio visual speech recognition (AVSR): Similar to lip reading, there has been a lot of research on development of audio-visual speech recognition systems as well. Neti el al. [22] performed sentence-level speech recognition for the first time using HMM on a limited dataset, by extracting features from audio and visual components of video. They showed that the performance of audio speech recognition systems in noisy environments can be improved by adding visual features. They used IBM ViaVoice dataset for their study. Potamianos et al. [25] also worked on the same dataset and DIGIT dataset for their model of different levels of fusion for audio-visual speech recognition. They proposed features fusion, decision fusion and hybrid fusion techniques for the bi-modal speech recognition. The authors expressed concern that the usage of high-quality video would mean increased cost, storage and processing requirements. These issues are investigated in this thesis.

Petridis et al. [23] developed a system for audio-visual word recognition, within context. In real world settings, context may or may not be known so this cannot be used for real world application. Mroueh et al. [21] developed a deep network architecture for AVSR. However, they used an IBM dataset of 10,400 words. They worked on multimodal fusion in deep learning for AVSR.

Afouras et al. [1] developed a dataset, LRS2-BBC, for audio-visual speech recognition. This dataset consists of extracts from recordings of BBC shows, encompassing around 26M words. They tested two models for lip reading, one with CTC loss and other with sequence-to-sequence loss. Sequence-to-sequence loss matches ground truth labels to output character probabilities and train the model with cross entropy loss, while CTC loss calculates loss of all possible permutations of output sequence across given frames. The authors also investigated the extent till which lip reading complements audio speech recognition, especially in noisy backgrounds.

Datasets

Many datasets are available for speech recognition. Details about some of the large scale lip reading datasets are given below:

- **GRID**[10]: This is an audio-visual dataset of 1000 sentences each spoken by 34 speakers. Out of these 34 speakers, 16 were female and 18 male. They were staff and students at University of Sheffield and spoke English as first language. Each sentence was a 6 word sequence with given form. Videos are recorded in quiet or low-noise conditions.

- **Modality**[11]: This dataset includes recording of 35 speakers, 26 male and 9 female divided between native and non-native English speakers. It was customized to simulate scenarios of voice control for mobile devices, thus it consists of 231 words. In order to assess both isolated and continuous speech, the words were combined in a form of 42 sequences, which use all the 231 words. Half of these recording sessions were conducted in a quiet room, three kinds of noise were added in the other half of sessions in the background using loudspeakers.

- **LRW**[8]: It is a large-scale visual speech recognition dataset, which consists of 500 of different words spoken by over 1000 different speakers collected from British television channels.

- **LRS**[7]: This consists of thousands of hours of video with sentences from BBC shows recorded between 2010 and 2016.

- **LRS2-BBC**[1]: It is a large scale dataset for audio-visual speech recognition, extracted from different BBC programs including Dragonâs Den, Top Gear and Countryfile. It consists of around 26M words.

- **LRS3-TED**[2]: This is a multi-modal dataset for visual and audio-visual speech recognition. It consists of face videos of over 400 hours from TED and TEDx talks. Videos contain speakers at different angles from camera, that is, head-pose variation.

Chapter 2

Research Questions

This thesis focuses on real-time lipreading. It studies the effect of frame rate, noise and jpeg compression of frames on lipreading, ultimately aiming at reduction of computation, storage and transportation cost for video (visual data) and making these systems suitable for real-time cloud-based systems. Streaming video at 25 Hz for visual speech recognition, transferring the data to cloud and then processing it would mean substantially higher cost as compared audio speech recognition. This might lead to delay in getting the speech output from the cloud, which deems this system unusable for real-time applications. This problem can be solved by:

1. **Lower frame rate**: Lowering the frame rate means that less data needs to be transmitted and processed for video corresponding to equal time frame. As per the best of my knowledge, no one has tested impact of frame rate on the performance of lipreading systems. This might prove to be an efficient method for decreasing processing time of the system.

2. **Compress frames while streaming video to cloud system**: Compressing frames will lead to loss of some data, however, if we compress data to right level of quality so as to preserve useful information for lipreading and compress it enough to decrease data size, then transfer cost can be decreased significantly.

We know that adding visual features improves speech prediction over audio-only speech prediction systems [22] in noisy environments. However, no one has focused on impact of presence of noise in the video signal. The noise might be generated while transferring video to cloud or because of faulty camera output. Studying the impact of noise on these lipreading systems would help in assessment of reliability of these systems for real-world application.

These provide a base for the following research questions:

- What is the impact of frame rate of video stream on accuracy of lipreading systems?

- Does the impact of frame rate also depend on type of character to be detected?

- How does jpeg compression of frames impact the accuracy of lipreading systems?

- How does the presence of noise in video frames impact the accuracy of lipreading systems?

Chapter 3

Method

To answer the research questions, an experimental approach is adopted. For this purpose, LipNet is being used as the model lipreading system and Lip Reading in the Wild (LRW) [8] dataset is being used. The reasons for choosing them is explained later in detail.

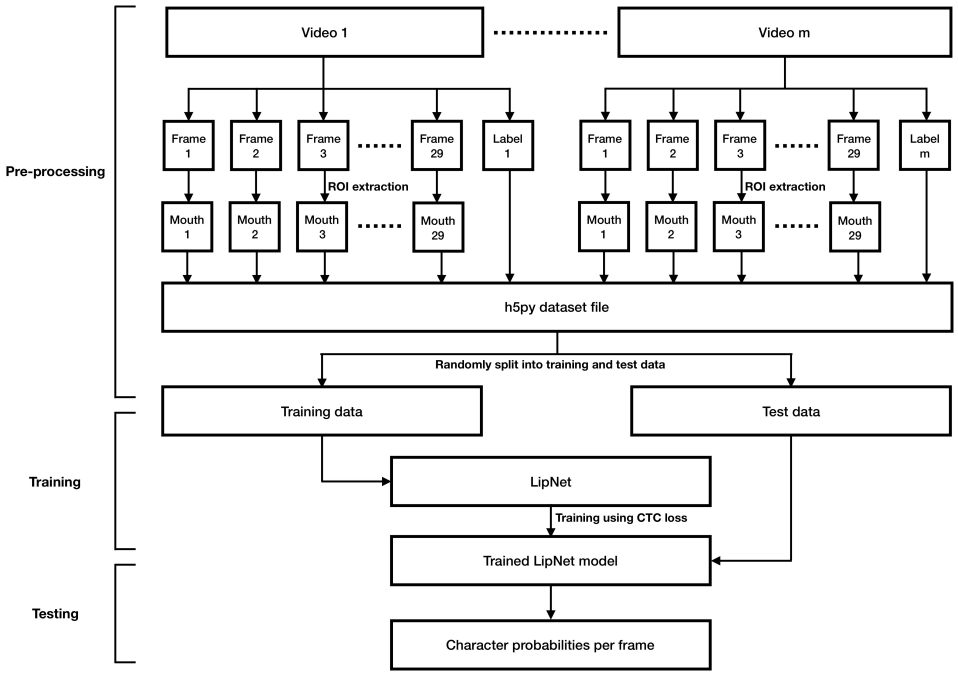


Figure 3.1: Overview of experimental setup

First of all, region of interest (mouth region) is extracted from each frame of all videos of the dataset. This is done using dlib library and OpenCV. Then, the resulting dataset is stored in the form of h5py file for easier access. The whole dataset is randomly divided into training and testing data, where 80 percent of the whole dataset goes to training set and 20 percent goes to test set. Once the dataset is pre-processed, it is used to train LipNet to detect characters per frame using CTC loss. After training

is complete, the model is used to recognize speech from unseen videos and character error rate is calculated. An overview of the experimental setup is shown in figure 3.1.

3.1 Lipreading model: LipNet

For the purpose of this thesis, LipNet [3] is used for experimentation purpose. LipNet is chosen because it is the state-of-the-art lipreading system available. Also, its model was publicly available, which made it possible to replicate the system for this thesis.

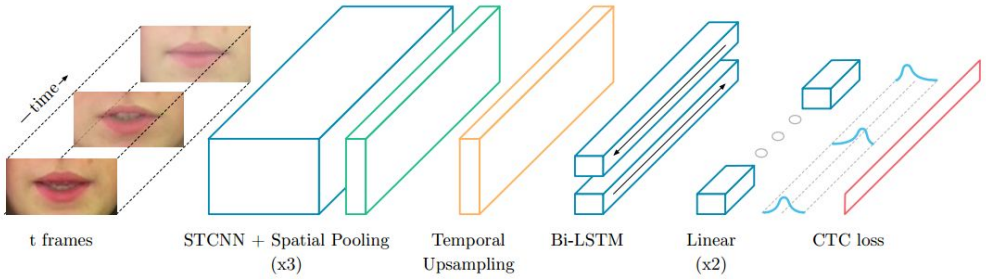


Figure 3.2: Overview of LipNet architecture [3]

An overview of the LipNet architecture is shown in figure 3.2. LipNet is an end-to-end trainable lipreading architecture, which maps video frames to text. It consists of three spatio-temporal CNN layers, followed by two Bi-GRU LSTMs and a fully connected layer in the end. CTC loss is calculated for the purpose of training the model.

Various layers of LipNet are explained in detail in this section.

1. **Spatio-temporal CNN**
   In 2D CNN, two-dimensional convolutions map local information from one layer onto another layer. This mapping is followed by an additive bias and sigmoid activation function. This can be represented as:

   $v_{ij}^{xy} = \tanh(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)})$ [15]

   where, v_{ij}^{xy} is the value of *(x,y)* coordinate in *j*th feature map of *i*th layer, tanh is the hyperbolic tangent function, b_{ij} is the bias for this mapping, m covers set of features in *(i-1)*th layer over present feature map, w_{ijm}^{pq} refers to the value at *(p,q)* position of kernel connected to mth feature map of previous layer, P_i and Q_i refers to height and width of the kernel. The resolution of these convolutional layers is decreased by performing a pooling operation. This decreases the impact of distortion of input over output from the network. A complete CNN network consists of multiple convolution and pooling layers stacked over each other. Weights w_{ij} and bias b_{ij} are learned during model training, using supervised or unsupervised learning techniques [4].

   Spatio-temporal CNNs have three dimensional kernels, instead of two-dimensional ones as in 2D CNN. The time dimension is the third dimension. Three dimensional kernels learn features in space and time dimensions, hence the name, spatio-temporal CNN. These kernels are applied over stack of frames from a

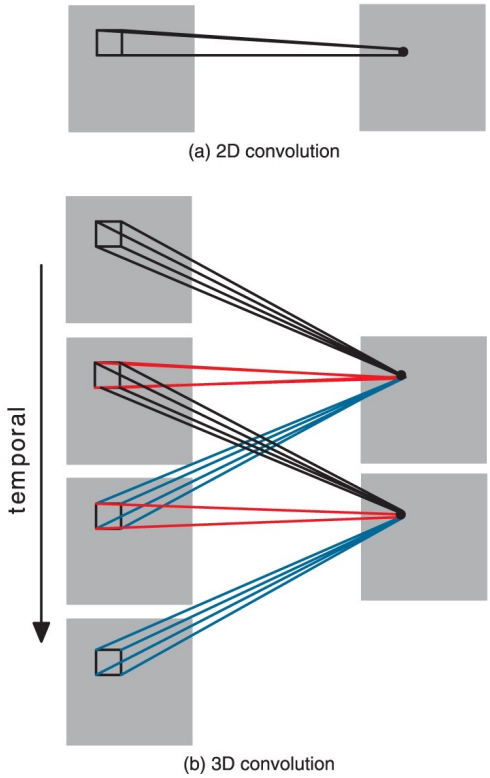


Figure 3.3: Comparison of 2D and 3D convolutions [15]

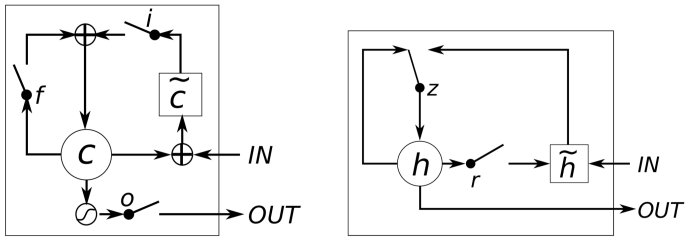
video. Hence, the kernels are connected to multiple continuous frames and learn motion-based features. Similar to the equation for 2D CNN, each value corresponding to position *(x,y,z)* in *i*th feature map of *j*th layer in 3D CNN is calculated by:

$$v_{ij}^{xyz} = \tanh(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{R=0}^{r_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)})[15]$$

Here, w_{ijm}^{pqr} value of kernel at *(p,q,r)*th position corresponding to *m*th feature map from previous layer. In depth explanation of 3D CNN architecture is provided by Shuiwang Ji et al. in [15].

To summarize, the difference between 2D and 3D convolutions is the kernel and input dimensions. This in turn results in a difference in kind of features extracted. Spatial features are extracted in 2D convolutions, while spatio-temporal features are extracted in 3D convolutions. Pictorial representation of difference of 2D and 3D convolutions is shown in figure 3.3.

2. **Bi-GRU**
   Gated Recurrent Units were introduced by Kyunghyun Cho et al. [6]. They were a modification of the original RNN, used to capture relationship between different time differences. These are similar to LSTMs, as they can also control flow of history information inside the unit. However, they do not have separate memory cells like LSTMs. This makes them computationally more efficient as



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 3.4: Graphical representation of LSTM and GRU[9] (a) i is the input gate, o is output gate and f is forget gate. c and $\tilde{c}$ are memory cell and new memory cell respectively. (b) r is reset gate and z is update gate. h and $\tilde{h}$ are activation function and candidate activation function respectively.

they expose the whole content without control and have less complex structure. The difference between LSTMs and GRUs is depicted in figure 3.4.

In Lipnet [3], standard formula of GRU is used.

$[\mathbf{u}_t, \mathbf{r}_t]^T = \text{sigm}\ (\ \mathbf{W}_z\ \mathbf{z}_t + \mathbf{W}_h\ \mathbf{h}_{t-1} + \mathbf{b}_g)$

$\tilde{\mathbf{h}}_t = \tanh\ (\ \mathbf{U}_z\ \mathbf{z}_t + \mathbf{U}_h\ (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h)$

$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t$

here, $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_t)$ is the input sequence for GRU, $\mathbf{h}_t$ is the activation function at time t and $\tilde{\mathbf{h}}_t$ is the candidate activation function. W and U are parameter matrices that are calculated during model training and b is a parameter vector.

In bidirectional GRU [14], one RNN maps input $(\mathbf{z}_1, ..., \mathbf{z}_t)$ to $(\overrightarrow{\mathbf{h}}_q, ..., \overrightarrow{\mathbf{h}}_t)$ and another RNN maps $(\mathbf{z}_1, ..., \mathbf{z}_t)$ to $(\overleftarrow{\mathbf{h}}_1, ..., \overleftarrow{\mathbf{h}}_t)$. And then the resulting output is $\mathbf{h}_t := [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t]$. In LipNet, the output of three spatio-temporal CNN layers is passed onto bi-GRU layer, which is followed by another bi-GRU layer.

3. **CTC Loss**
   Connectionist Temporal Classification Loss or CTC loss [13] is one of the most prevalent loss functions used in speech recognition systems. This is because it does not need the input to be trained to aligned outputs. Because of this same reason, it works well for handwriting recognition. It can be used for systems where a strict label boundary is not present.

   If a model gives a sequence of probabilistic distributions over tokens as output, then this loss function computes summation of probability of all possible permutations of output sequence, along with the blank character. Hence, the output sequence can be variable length. For example, suppose the output sequence is "*abc*" for T = 4, then, CTC defines its probability as *p(abc_) + p(_abc) + p(a_bc) + p(ab_c) + p(aabc) + p(abbc) + p(abcc)*, here "_" refers to blank character. The blank characters and adjacent repeated characters are removed by CTC to get the final string. However, if the final output has a set of adjacent repeated characters, for example "*p*" is repeated in "*apple*", then in these cases a blank character is always inserted between two *p*s, so that one of them doesn't get deleted automat-

ically. Because of this reason, number of frames always need to be more than number of characters in output sequence to accommodate repeating characters.

CTC Loss works well with label sequences, that is, if two labels frequently occur together, then it tends to group them. This makes it useful for predicting words from character sequences. This property is exploited in LipNet system and helps in prediction of words directly from frames without the need of language model.

3.2 Dataset and its variations

For the purpose of this thesis, Lip Reading in the Wild (LRW) [8] dataset is used. This dataset consists of 500 words spoken by over 800 speakers each. This makes the dataset speaker independent. It also makes it closer to real-world application as it has more words than in the dataset used by Assael et al. to train LipNet. They used GRID corpus [10], which consists of 26 distinct words and 26 english letters. Also, the GRID corpus has sentences of a given grammar, which would result in over training of model for this grammar and would make it difficult to access its usability for real-life systems.

Using the LRW dataset increases the number of words teh system is trained on. Also, it makes it possible to train the system in reasonable time, as due to the absence of huge computation power it would not have been possible to train the system on other huge dataset, like LRS, LRS2-BBC and LRS3-TED, in the duration of this thesis.

Also, as the videos were of 29 frames each, there was no need to do much pre-processing for feeding them to the network. All the videos were recorded at 25 Hz each. They were sub sampled for other frame rates, from 11 to 25 with a step size of 2. The frame rate was not decreased further as an average human speaks about 10 letters per second [26], hence, we need at least 10 frame rate to predict 10 characters. The dataset was stored in h5py files and generators were used to load the dataset in batches and converted them to numpy arrays before passing them to the network for training.

3.2.1 Frame rate reduction

The videos were sub-sampled to reduce the frame rate. This was done by first calculating number of frames with the given frame rate. This is calculated as: $t' = int(t/25 * f)$, where t is the total number of frames with 25 Hz rate (29 in our case), f is the lower frame rate required and t' is the total number of frames with f frame rate. Once total number of frames are known, we know that 29 frames needs to be reduced to t' frames. Video is then sub-sampled at equal interval to get t' frames, however, this results in fractional frame numbers. Hence, they need to be rounded down to get exact frame numbers. This would not introduce any bias as there are more than 800 utterances for each word by different speakers and speech of speech also differs per user. After sub-sampling, the total length of video was made 29 frames by adding repeated frames between sampled frames. This was done to avoid bias during the comparison of CTC loss for various frame rates. This is explained via an illustration in figure 3.5.

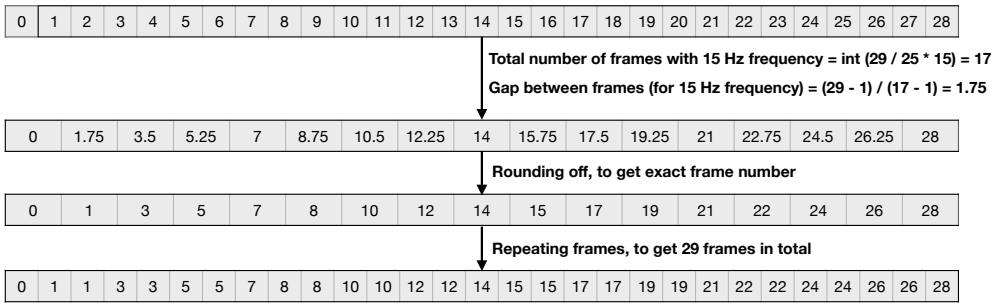
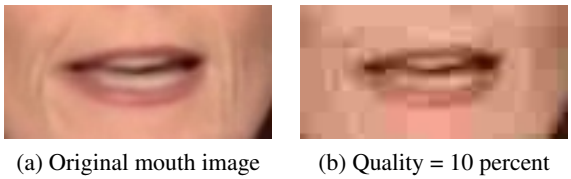


Figure 3.5: Frame rate reduction from 25 Hz to 15 Hz

3.2.2 Jpeg compression

One of the main problems of using cloud based systems for visual speech recognition is the size of data. Videos take up more space than audio, which makes it challenging to transfer it to the cloud for processing. Compressing the frames might be helpful in this situation. However, effect of compression on the performance of lipreading system is unknown. To check how this compression might affect the performance of system, ROI of each frame is compressed with 30 percent quality. This is done in order to significantly decrease the size of data (to 30 percent in this case) and still keeping the ROI good enough for lipreading. As can be seen in figure 3.6, with quality 10 percent the picture quality decreases a lot and the ROI is not clearly visible. With 70 percent quality, there's not much reduction in size of data. However, with 30 percent quality ROI is still clearly visible with a significant reduction in size of data.



(a) Original mouth image (b) Quality = 10 percent

(c) Quality = 30 percent (d) Quality = 70 percent

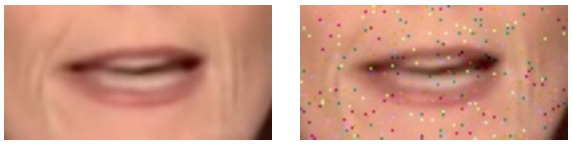
Figure 3.6: Original ROI image and after jpeg compression-decompression retaining given percentage of quality

For the purpose of this thesis, the ROI for each frame of all videos are first compressed with 30 percent quality and then they are decompressed. These compressed-decompressed frames are then used to investigate how compression might affect performance of LipNet.

3.2.3 Added noise

Visual speech recognition can be used for variety of applications, including smart assistants, human-robot interaction and human-virtual agent interaction. It would be lead to usage of variety of cameras, which will produce varying quality of images. Also, transferring data to the cloud might lead to noisy visual signal. To artificially create these artifacts, salt and pepper noise is added to the dataset. The choice of salt and pepper noise was based on the fact that this type of noise can be generated as a result of analog to digital signal conversion in cameras or due to bit error in transmission [5]. For adding artificial noise, some randomly selected values of the RBG color matrix of the frame are set to 0 or 255.

The probability is set to be 3 percent, which means that each pixels has 3% probability of noise being added to it. As can be seen in figure 3.7, if 1 percent noise is added it leads to negligible noise. 7 percent noise leads to really bad image quality. However, 3 percent noise is a noticeable amount of noise, without a lot of deterioration in the quality of the image.



(a) Original mouth image (b) Noise = 1 percent

(c) Noise = 3 percent (d) Noise = 7 percent

Figure 3.7: Original ROI image and after addition of salt and pepper noise

3.3 Experimental setup

The experiments were conducted on the TU Delft INSY cluster. It can be remotely accessed from the system to upload batch jobs via ssh connection. Once a job is finished, the output is written in a text file, which can be used to evaluate results. It has multiple GPUs which are helpful for training the network using tensorflow-gpu. CUDA 9.0 and CuDNN 7.0.5 libraries also had to be activated to use GPUs for training. Tensorflow GPU version 1.9.0 was used in Python 3.4.0.

For training, one GPU was used for tensorflow and two CPUs to load dataset batch and convert it to numpy array. Using this configuration, each epoch took around 5 hours to train the model on whole dataset. For each configuration, model was trained for around 40 epochs. Hence, training model took around 200 hours for each configuration. There were 10 configurations in total, this means total training time was 2000 hours.

3.4 Evaluation measures

To answer the research questions, following quantities are measured:

1. **CTC loss** of test data after each epoch: To compare the performance of models, CTC loss of test data after each epoch is recorded. It also helps to track the training of models and see if models get over-trained. With the increase in epochs, difference CTC loss between adjacent epochs decreases, which means that models fits the dataset better. Loss for different conditions can be compared to see which model performs better.

2. **Character error rate** of the trained model: Once the models are trained, character error rate for the test data is measured to compare the performance of model over various configurations, like different frame rate and presence of noise.

3. **Confusion matrix** of the trained model: Confusion matrix is plotted for error in character-level prediction by trained model. This is plotted to evaluate if the performance of the system depends on the type of characters that it has to predict.

Chapter 4

Results and Discussion

In this section, results of the experiments are being discussed. It includes a graph of CTC loss over epochs for all different configurations of dataset. It shows how the model trained for lipreading and loss decreased with progressing runs. The section is divided into three subsections, each dealing with a different set of experiments. The first section discusses results from varying frame rate of the videos in the dataset. The next section is about the compression of frame images. The last section deals with results from the noisy dataset.

4.1 Varying frame rate

In this section, the results from varying frame rate are shown and discussed. As mentioned in previous section, frame rate for videos is varied from 11 to 25, with an increment of 2 for each experiment. First of all, dataset is created for all these different frame rates. Then for each experiment, model is first trained with a specific frame rate dataset for 40 epochs and the CTC loss for these runs are recorded. Once the model has been trained, the character error rate (CER) of the test set is calculated.

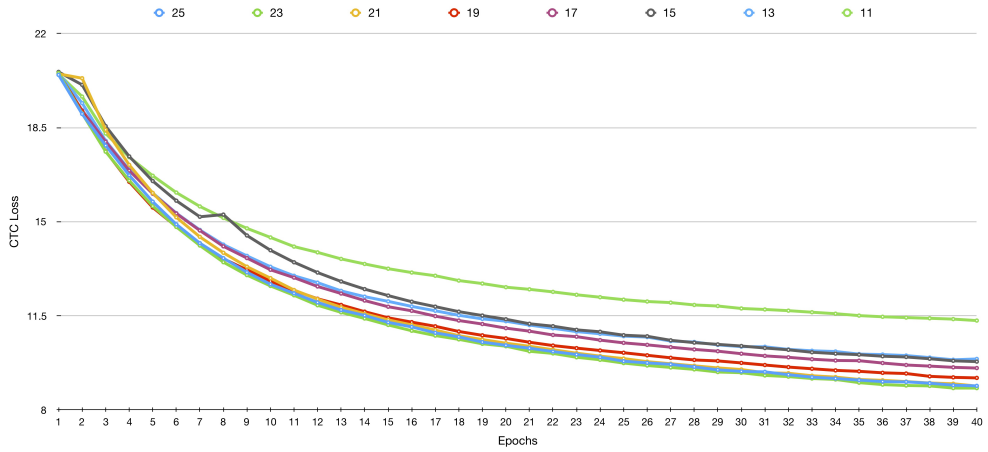


Figure 4.1: CTC loss over number of epochs for various frame rates

Figure 4.1 depicts CTC loss during model training. As can be seen from the graph,

Table 4.1: LipNet CER for videos at different frame rates

Frame rate	CER
11	51.46%
13	45.32%
15	44.97%
17	43.75%
19	42.21%
21	40.89%
23	40.64%
25	40.91%

during the start models performed almost similar. However with more epochs, a noticeable difference between performance of models for different frame rates can be noticed. It depicts that frame rate 25 to 21 have almost similar performance. The performance gradually decreases from frame rate 19 to 13 and there's a sudden drop in performance of model for frame rate 11.

Similar observations can be seen in table 4.1, which shows character error rate for different frame rates after models have been trained for 40 epochs. From these observations, it can be said that frame rate can be reduced to 21 Hz without any major impact on performance of system. However, if the frame rate is decreased further, accuracy of the system gradually decreases until it reaches 13 Hz. However, it should not be decreased further as the performance of system will drop drastically.

The sudden increase in error at frame rate 11 can be attributed to the fact that an average human speaks about 10 characters per second [26], so for a frame rate of 11, number of frames is almost equal to the number of characters being spoken. However, some characters might be made up of a set of phonemes and hence, it would not be possible to detect them in one frame as they compose of a combination of sounds. With the frame rate being 11, there are not sufficient frames to detect these characters. Also, if the true label has a set of adjacent repeated characters, for example "p" is repeated in "apple", then in these cases a blank character is always inserted between two *p*s, so that one of them doesnât get deleted automatically. Because of this reason, number of frames always need to be more than number of characters in output sequence to accommodate repeating characters.

There is a technical explanation for this as well. For a video of a total length of 29 frames at 25 frame rate, it will have 12 frames for frame rate 11. In LRW dataset, the maximum word length is 12 characters. This is exactly equal to number of frames. During prediction, first two outputs from Bi-GRU are discarded, as first few outputs from RNN tend to be garbage [29]. Here, number of character predicted from the model are 10, while number of characters in true label can be up to 12. Hence, there is a hike in error for frame rate 11.

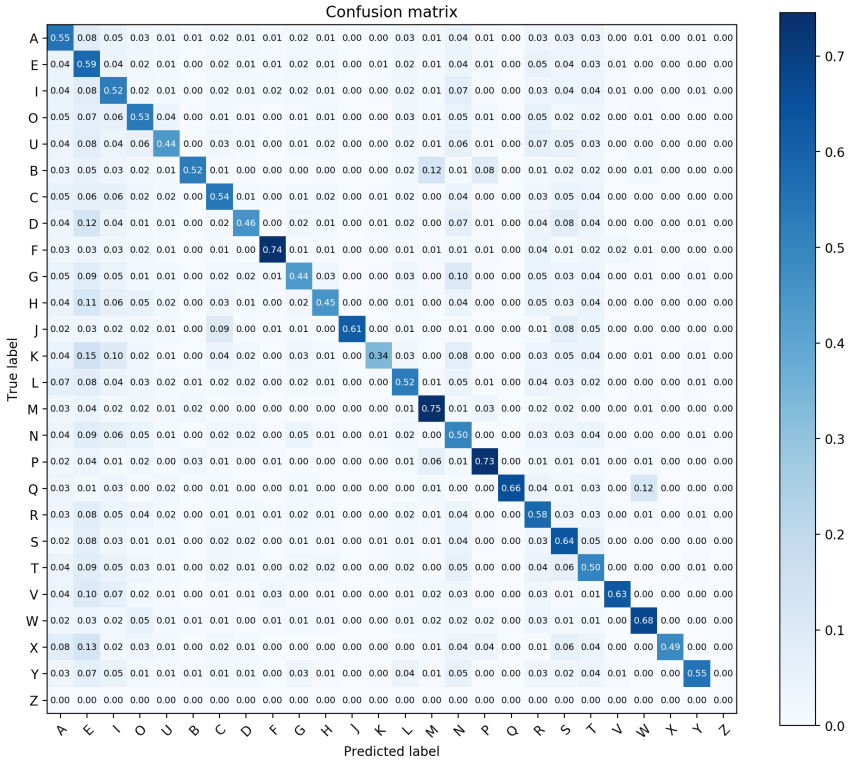


Figure 4.2: Confusion matrix for frame rate 25

To find out if characters have any impact on the CER for different frame rates, character-wise confusion matrix is plotted. Figure 4.2 shows confusion matrix for frame rate 25, after 40 epochs of model training. Figure 4.3 shows confusion matrix for frame rate 11. Confusion matrices for other frame rates are included in appendix A.

As can be seen from figure 4.2 and 4.3, letter 'K' has the least probability of being identified correctly. In figure 4.3, letter 'K' has 20% probability of bring identified correctly and 15, 12 and 11 % probabilities of being identified as 'E', 'N' and 'I' respectively. In figure 4.3, letter 'F' has highest probability of being correctly recognized, while in figure 4.2 letter 'M' has the highest probability, however 'F' also has a high probability here.

In both figures, it can be seen that the vowels perform below average with an error rate more than average character error made of the model. In figure 4.2, average character error rate is 40.91%, but all the vowels have an error rate of more than 41%. And in figure 4.3, average character error rate is 51.46%, but all the vowels have an error rate of more than 52%.

In both matrices, it can be seen that general trend is similar, with characters 'F', 'M' and 'P' having highest chances of being correctly recognized and letter 'K' having the least chances of being correctly recognized. Also, vowels perform below average. Hence, it can be concluded that frame rate does not have any impact on the type of character to be detected.

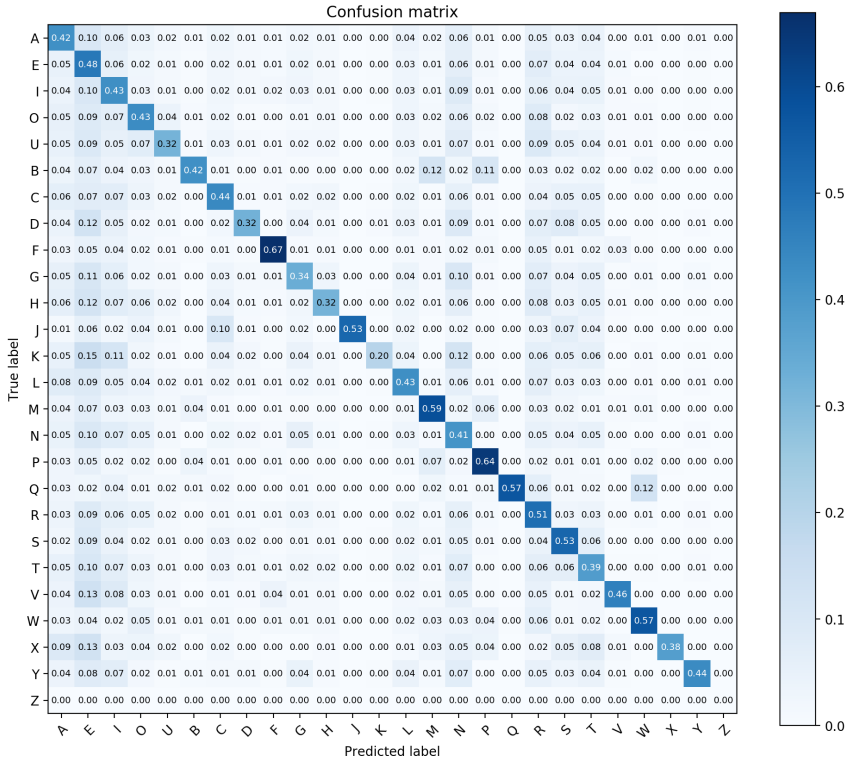


Figure 4.3: Confusion matrix for frame rate 11

4.2 Jpeg compression

In this section, the results corresponding to jpeg compressed dataset are discussed. Figure 4.4 shows the CTC during training the model using original and the compressed dataset. It can be seen that their graphs start converging and there is not much difference in the performance of the models.

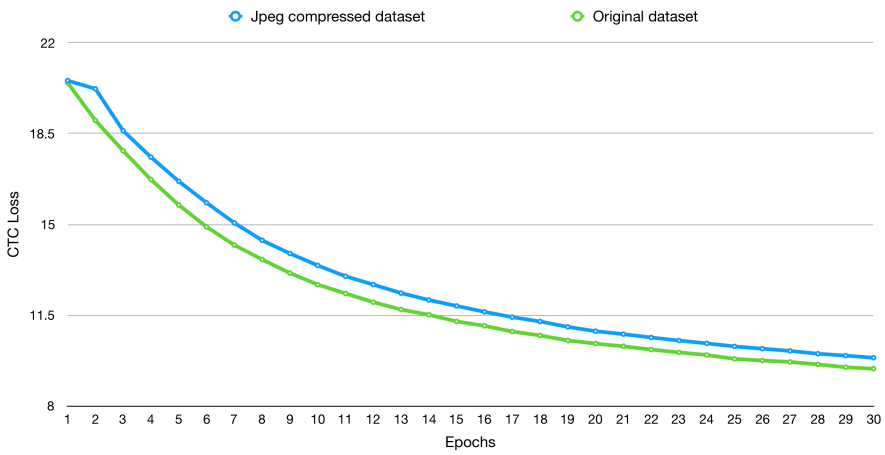


Figure 4.4: CTC loss over number of epochs for original and compressed dataset

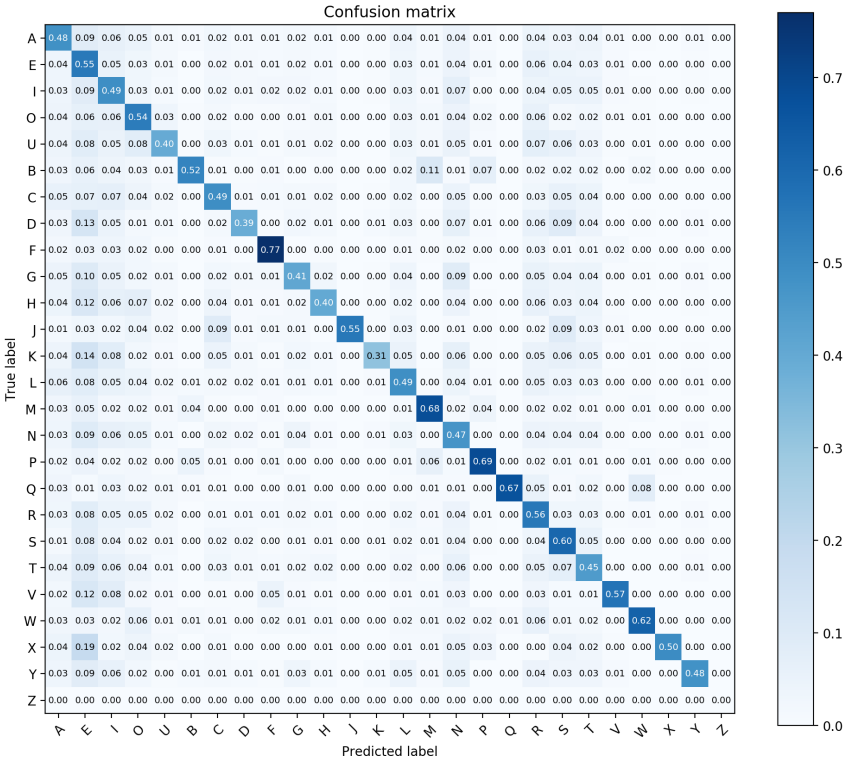


Figure 4.5: Confusion matrix for frame rate 25 with compressed dataset

Table 4.2 shows CER for the original dataset and the compressed one after model is trained for 30 epochs for each condition. It can be seen that there is not much different in error rate, it does not increase a lot on data compression. This is similar to the observation from figure 4.4. Storage space can be reduced to 30 percent of original size without much impact on performance of system.

Table 4.2: LipNet CER for original dataset and compressed dataset

Condition	CER
Original dataset	43.39%
Compressed dataset	44.71%

Figure 4.5 shows confusion matrix of compressed dataset after model has been trained for 30 epochs. Overall trend of CER is similar to the original dataset, figure 4.2. Letter 'K' has least probability of being correctly identified. Vowels has more than average character error rate, with the lowest character error rate being 45% for letter 'E' and the highest character error rate being 60% for letter 'U'. Also, letters 'F', 'P', 'M', and 'Q' have highest probabilities of getting correctly identified.

4.3 Noisy dataset

In this section, results for noisy dataset are discussed. Figure 4.4 shows the CTC during training the model using original and the noisy dataset. It can be seen that their graphs have substantial difference between them and it does not seem to converge beyond a certain limit.

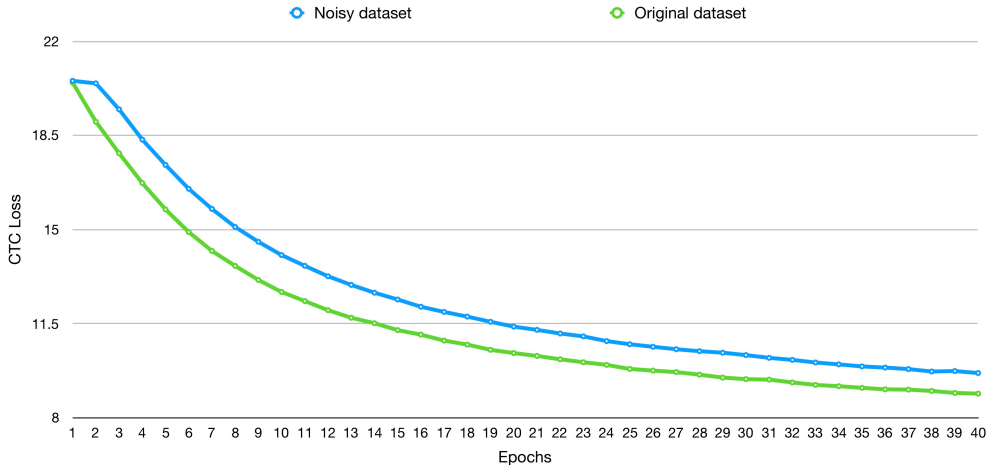


Figure 4.6: CTC loss over number of epochs for original and noisy dataset

Table 4.3 shows CER with original and noisy dataset. The noisy dataset only has 3 percent noise added to it. There is a big difference in character error rate of noisy and original dataset. This observation aligns with the observation from figure 4.6. It can be said that presence of noise can deteriorate results from the system and hence, much care needs to be taken to avoid noise in video streaming. We know that salt and pepper noise can be generated as a result of analog to digital signal conversion in cameras or due to bit error in transmission [5]. Hence, quality of video capturing camera and video transmission to the cloud needs to be optimum to avoid faulty results.

Table 4.3: LipNet CER for original dataset and noisy dataset

Condition	CER
Original dataset	40.91%
Noisy dataset	44.08%

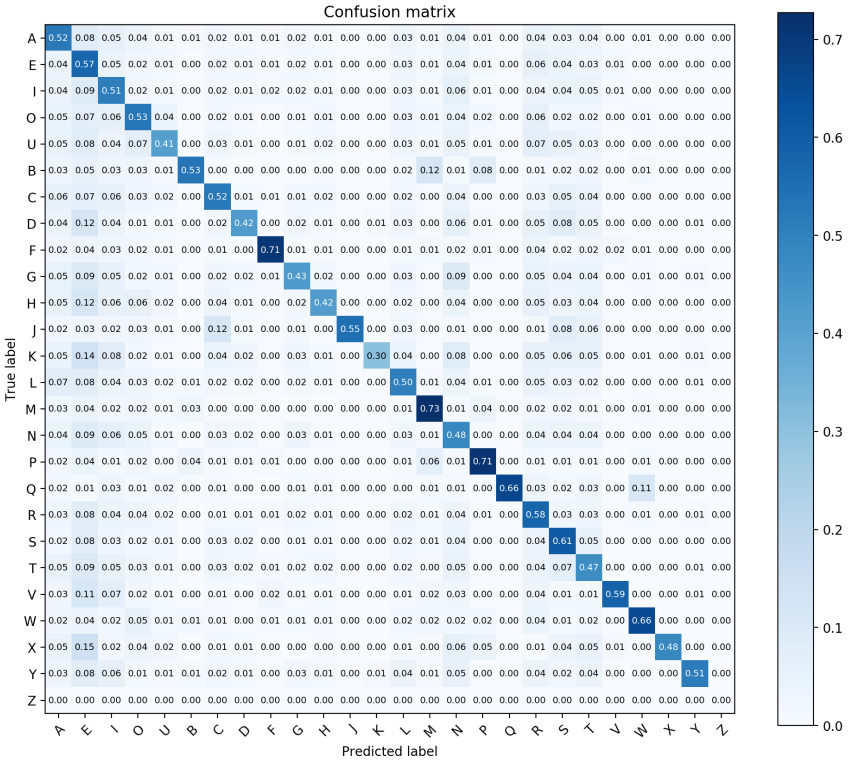


Figure 4.7: Confusion matrix for frame rate 25 with noisy dataset

Figure 4.7 shows confusion matrix of noisy dataset after 40 epochs. Here as well, the trend of CER is similar to the confusion matrix for original dataset, 4.2. Letter 'K' has least probability of being correctly identified. And letters 'M', 'F', 'P', and 'Q' have highest probabilities of getting correctly identified.

Chapter 5

Conclusions and Future Work

This report investigated the impact of various factors on the performance of LipNet. First variation was frame rate, for this frame rate of videos was varied from 11 to 25 with an increment of 2 for each experiment. Reduction in frame rate means, less data needs to be transmitted and processed. This in turn would make lipreading faster and easier to implement in real-time. Through experiments it was found out that when frame rate is reduced from 25 to 21, there was almost no impact on performance. After that performance started decreasing gradually till the frame rate of 13 but performance suddenly dropped at frame rate 11. However, the error increase does not depend on type of character being recognized and it increases proportionally for all the characters.

When frames of all videos of the dataset are compressed to 30 percent quality, there was slight drop in performance. By compressing the frames to 30 percent quality, the size of the dataset can be decreased by 70 percent. This will result in far less memory usage. Hence, data can be compressed for transmission without worrying about much degradation in performance.

Also, presence of 3 percent noise in the dataset lead to substantial increase in error rate of system. As we know that salt and pepper noise can be generated as a result of analog to digital signal conversion in cameras or due to bit error in transmission [5]. This suggests that more care needs to be taken when using different cameras for video streaming to maintain the quality of video.

Overall, this study shows that data usage for video streams for lipreading can be reduced without loosing much performance. For example, reducing the frame rate from 25 to 21 and compressing the frames with 30% quality loss will result in an effective reduction of 25% in data storage and transfer needs with only a marginal reduction in performance. Reduction to 13 Hz can reduce data bit-rate to about 12,5% of the original when combined with the same compression with potentially only reasonable increase in error rates. Although the combined effect of compression and such dramatic frame rate reduction needs to studied in future work, our results show that current state of the art real-time lipreading systems do not need high frame rates or high raw video frames. This opens up the potential for real-time cloud-based lipreading for example social robot.

In the future, more elaborate analysis need to be done for various lipreading systems and using multiple dataset. This would help in generalizing the results. Also, combinations of various dataset alterations can be used to find out an optimal set of

variations that can be used to make real-time lipreading less expensive.

When evaluating the model trained on different frame rates' datasets, we need to note that these results correspond to system trained on given frame rate. In real-life, system might be trained on original data (25 Hz) and then if we use it to predict speech from videos recorded at other frame rates, results might differ. This also applies in case of compressed dataset and noisy dataset. Hence, choice of training set needs to be carefully evaluated for training real-life lipreading systems as it might affect the performance of the system.

Bibliography

[1] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.

[3] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando De Freitas. Lipnet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599*, 2016.

[4] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. In *New advances in machine learning*. IntechOpen, 2010.

[5] Charles Boncelet. Handbook of image and video processing, chapter image noise models, 2005.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[7] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3444–3453. IEEE, 2017.

[8] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, pages 87–103. Springer, 2016.

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[10] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.

[11] Andrzej Czyzewski, Bozena Kostek, Piotr Bratoszewski, Jozef Kotus, and Marcin Szykulski. An audio-visual corpus for multimodal automatic speech recognition. *Journal of Intelligent Information Systems*, 49(2):167–192, 2017.

[12] Cletus G Fisher. Confusions among visually perceived consonants. *Journal of speech and hearing research*, 11(4):796–804, 1968.

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[14] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[16] Veton Kepuska and Gamal Bohouta. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 99–103. IEEE, 2018.

[17] Christine Chong-hee Lieu, Georgia Robins Sadler, Judith T Fullerton, and Paulette Deyo Stohlmann. Communication strategies for nurses interacting with patients who are deaf. *Dermatology nursing*, 19(6):541, 2007.

[18] Patrick Lucey and Sridha Sridharan. Patch-based representation of visual speech. In *Proceedings of the HCSNet workshop on Use of vision in human-computer interaction-Volume 56*, pages 79–85. Australian Computer Society, Inc., 2006.

[19] Iain Matthews, Timothy F Cootes, J Andrew Bangham, Stephen Cox, and Richard Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[20] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746, 1976.

[21] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2130–2134. IEEE, 2015.

[22] Chalapathy Neti, Gerasimos Potamianos, Juergen Luettin, Iain Matthews, Herve Glotin, Dimitra Vergyri, June Sison, and Azad Mashari. Audio visual speech recognition. Technical report, IDIAP, 2000.

[23] Stavros Petridis, Themos Stafylakis, Pingehuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. End-to-end audiovisual speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6548–6552. IEEE, 2018.

[24] Vassilis Pitsikalis, Athanassios Katsamanis, George Papandreou, and Petros Maragos. Adaptive multimodal fusion by uncertainty compensation. In *Ninth International Conference on Spoken Language Processing*, 2006.

[25] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

[26] Lawrence R Rabiner and Ronald W Schafer. *Theory and applications of digital speech processing*, volume 64. Pearson Upper Saddle River, NJ, 2011.

[27] Brendan Shillingford, Yannis Assael, Matthew W Hoffman, Thomas Paine, Cían Hughes, Utsav Prabhu, Hank Liao, Hasim Sak, Kanishka Rao, Lorrayne Bennett, et al. Large-scale visual speech recognition. *arXiv preprint arXiv:1807.05162*, 2018.

[28] Eric Tatulli and Thomas Hueber. Feature extraction using multimodal convolutional neural networks for visual speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2971–2975. IEEE, 2017.

[29] Chengwei Zhang. How to train a keras model to recognize text with variable lengths. 2018. `https://www.dlology.com/blog/how-to-train-a-keras-model-to-recognize-variable-length-text/` [Online; accessed 19-August-2019].

[30] Ziheng Zhou, Guoying Zhao, Xiaopeng Hong, and Matti Pietikäinen. A review of recent advances in visual speech decoding. *Image and vision computing*, 32(9):590–605, 2014.

Appendix A

Appendix

Confusion matrices for various frame rates are listed here.

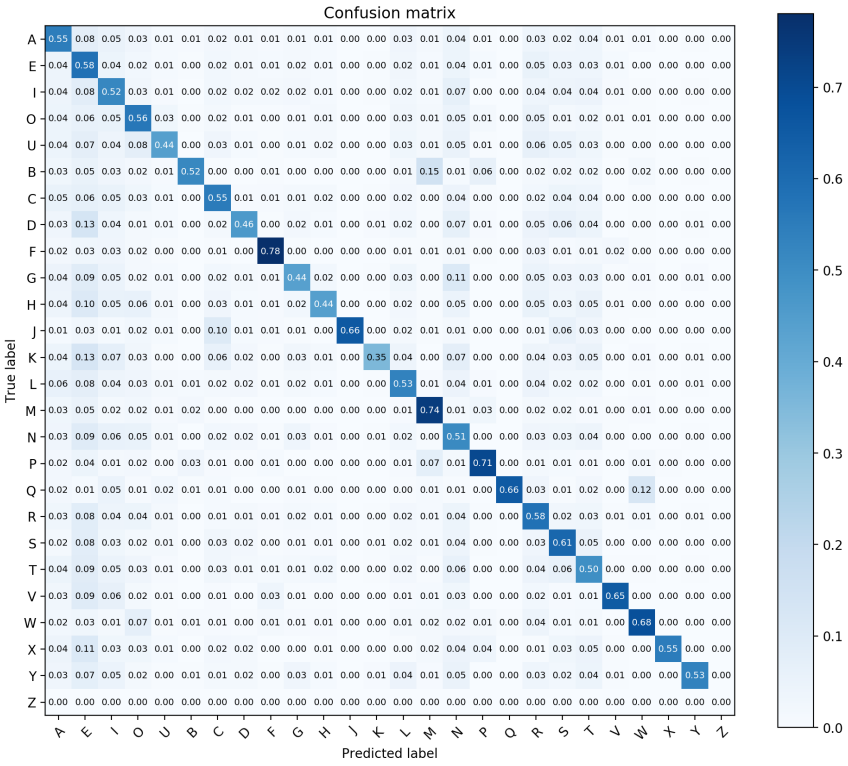


Figure A.1: Confusion matrix for frame rate 23

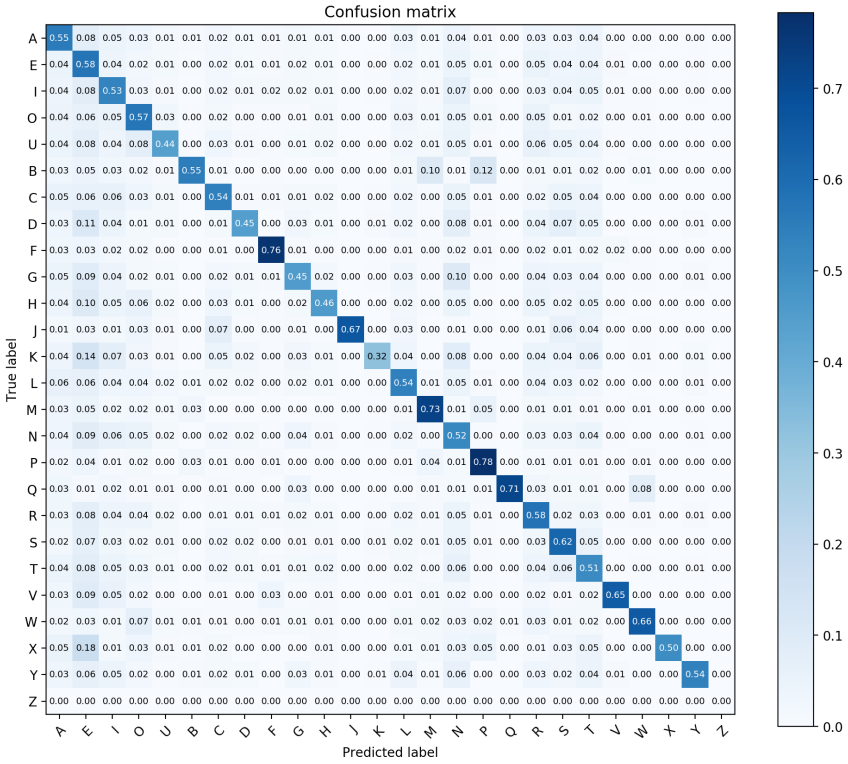


Figure A.2: Confusion matrix for frame rate 21

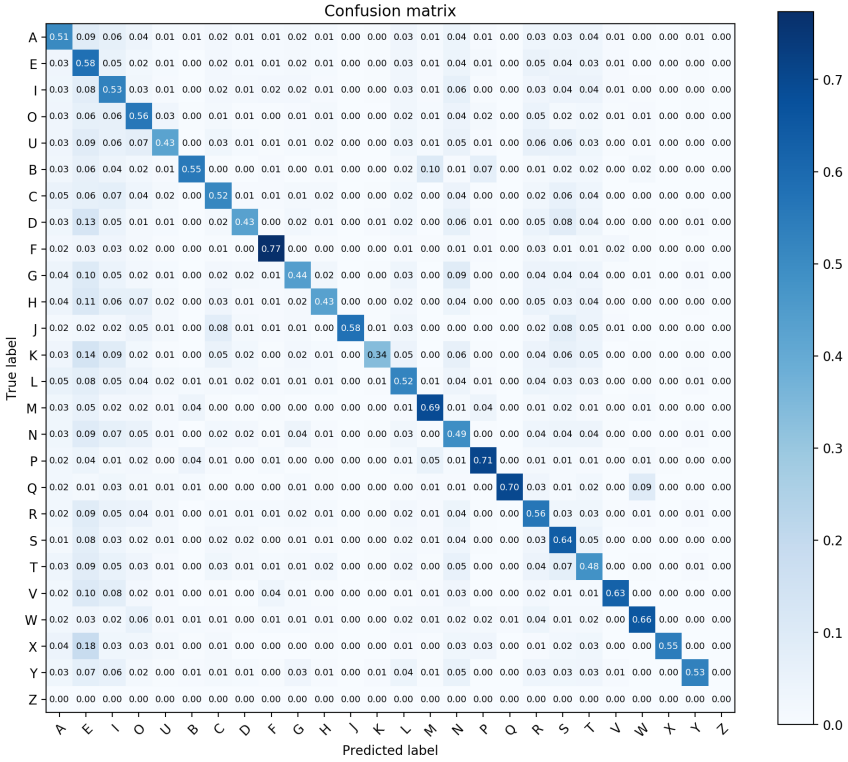


Figure A.3: Confusion matrix for frame rate 19

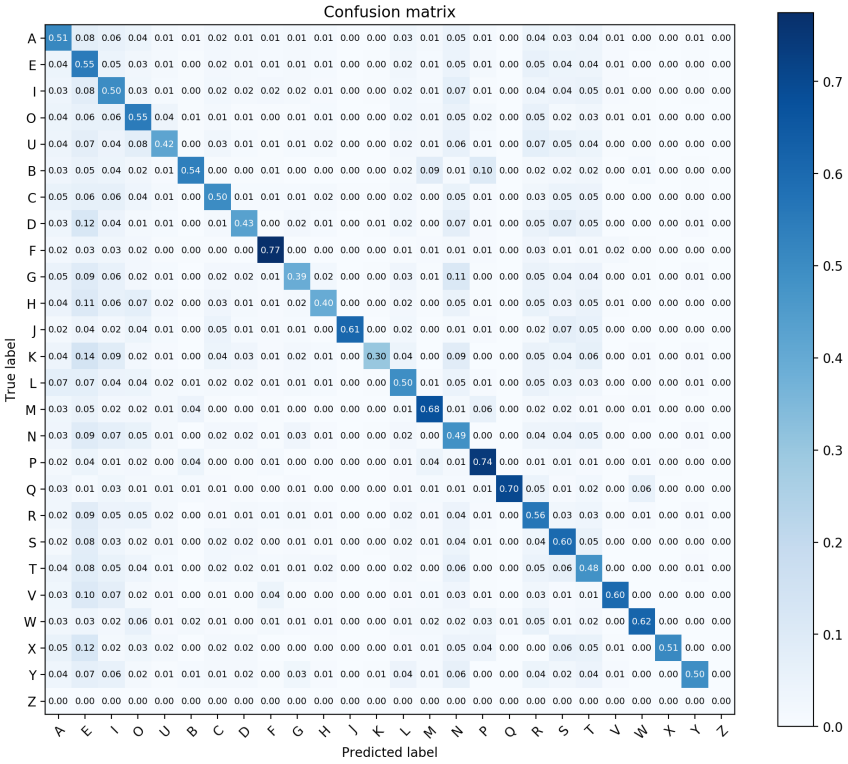


Figure A.4: Confusion matrix for frame rate 17

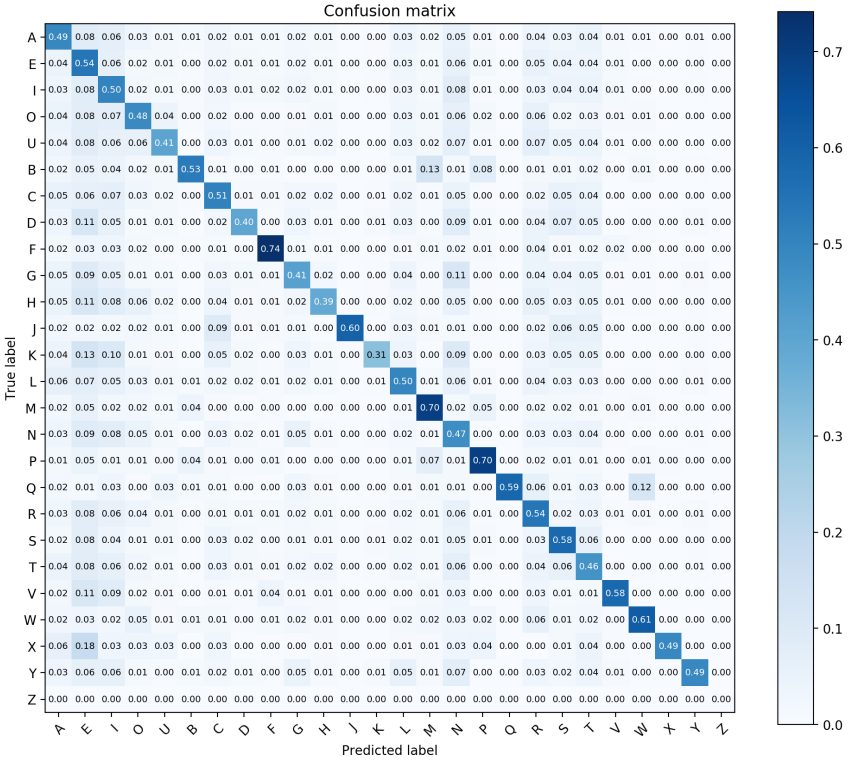


Figure A.5: Confusion matrix for frame rate 15

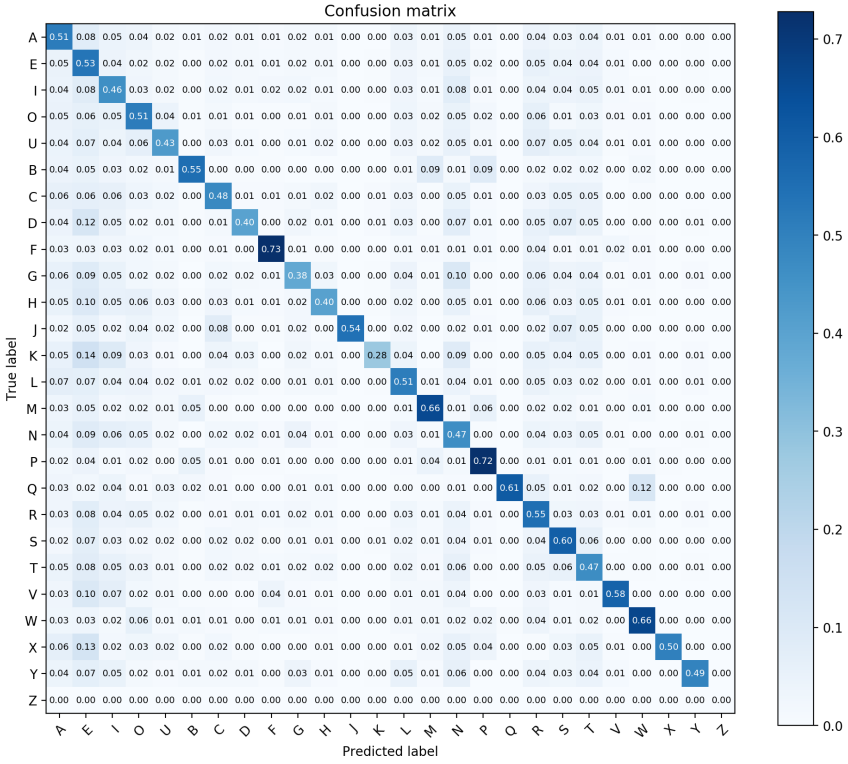


Figure A.6: Confusion matrix for frame rate 13