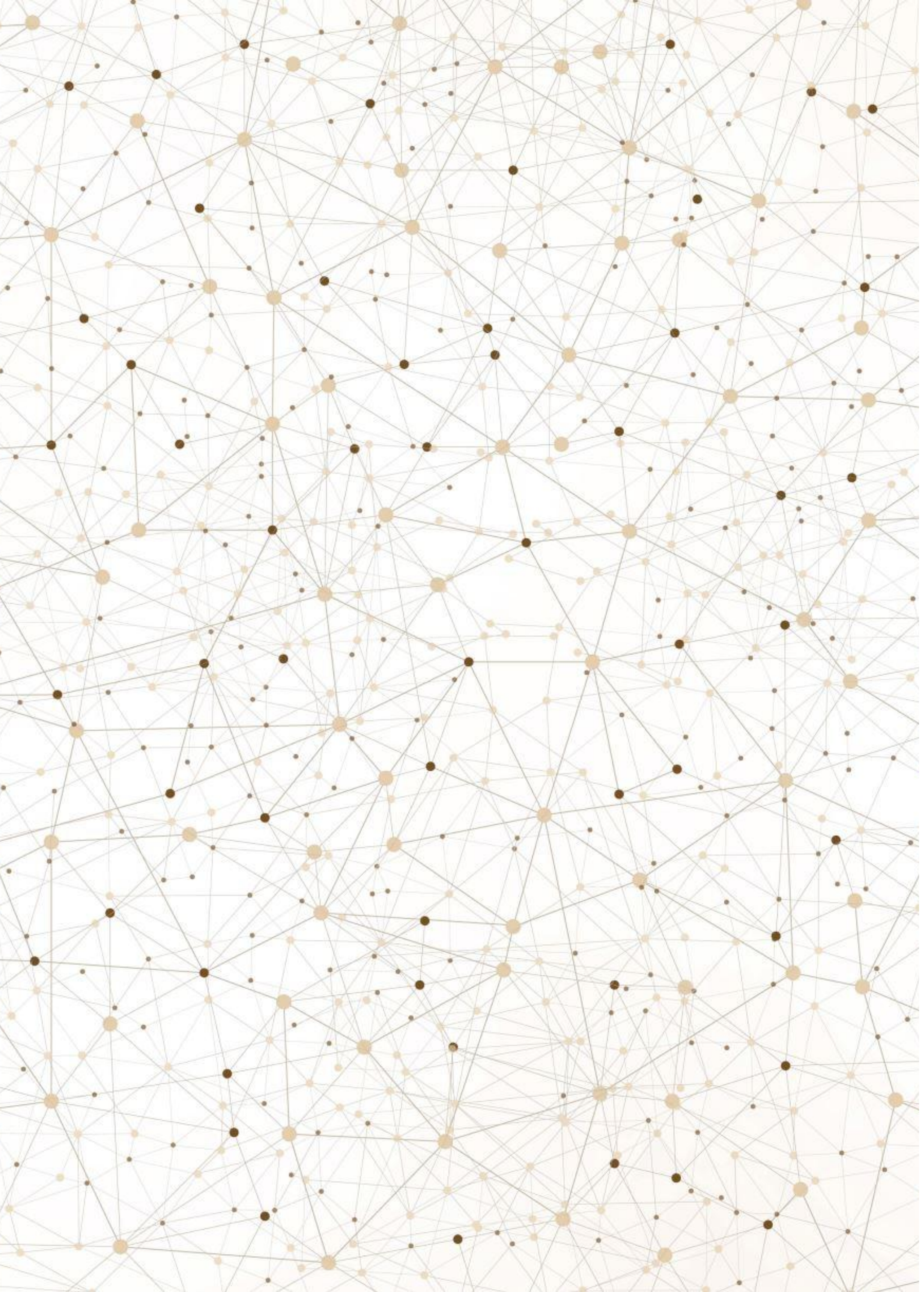


AUTOMATED LIP READING USING DEEP LEARNING

AI2100 – Deep Learning

INTRODUCTION

- Lip reading is a technique to understand words or speech by visual interpretation of face, mouth, and lip movement without the involvement of audio.
- It plays an important role in places where speech recognition is difficult.
- Using Deep Learning methods, we can now build neural networks that can help classify the image sequences to a particular label.
- We are presenting a CNN-based classifier and its comparison with a pre-trained ResNet model, which is a very famous model for handling image data.

Dataset and Features

- We used MIRACL-VC1 dataset for our project.
- Used for visual speech recognition, face detection and biometrics.
- 15 speakers - 10 female, 5 male
- Each speaker speaks 10 instances of 10 different words, and 10 different phrases.
- 3000 total instances. (15x20x10)
- Every instance is a numbered sequence of color and depth images of 640x480 pixels dimension.

ID	Words	ID	Phrases
1	Begin	1	Stop navigation.
2	Choose	2	Excuse me.
3	Connection	3	I am sorry.
4	Navigation	4	Thank you.
5	Next	5	Good bye.
6	Previous	6	I love this game.
7	Start	7	Nice to meet you.
8	Stop	8	You are welcome.
9	Hello	9	How are you?
10	Web	10	Have a good time.

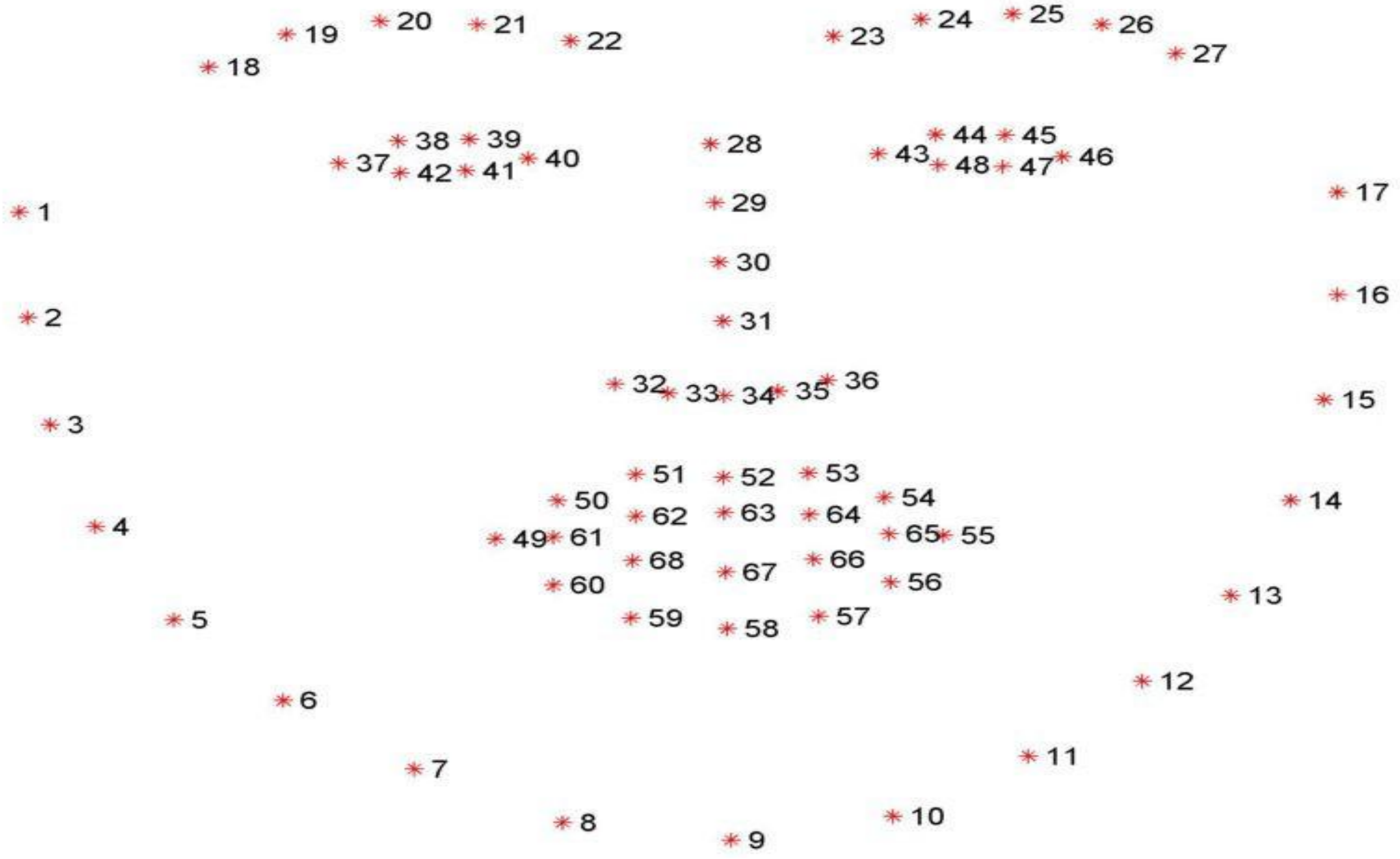
Dataset we used

- We discarded the depth images as we couldn’t come up with a proper use of those.
- We used only words for classification and discarded the phrases, for simplicity.
- The length of image sequences varied from 4 to 22 images, for different words.
- So, we had 1500 instances in total.
- We decided to use 70% of this data for training (1050 instances), and 30% for validation (450 images)

Data Pre Processing

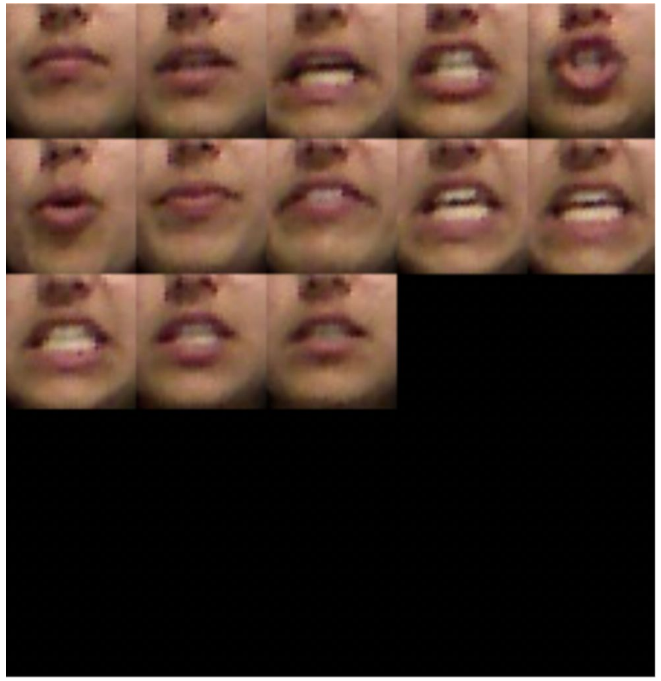
Our data pre processing included 2 steps.

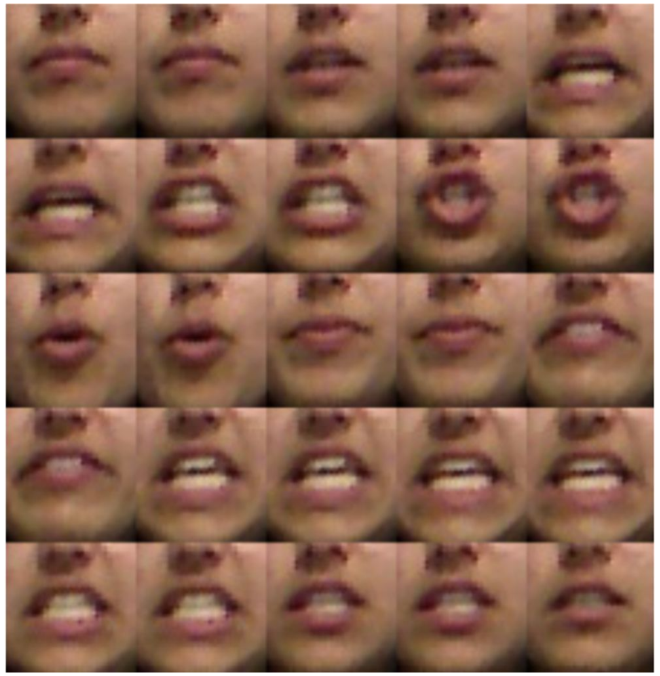
- Step 1: Identifying the lip region of a face
  - For this step we used DLIB.
  - Dlib is a modern C++ toolkit containing machine learning algorithms and tools.
  - Using that we first identified the face of the person speaking.
  - Different regions of the face are identified as shown. So, we mainly have to focus on the lip region which includes regions numbered from 49 to 68.


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

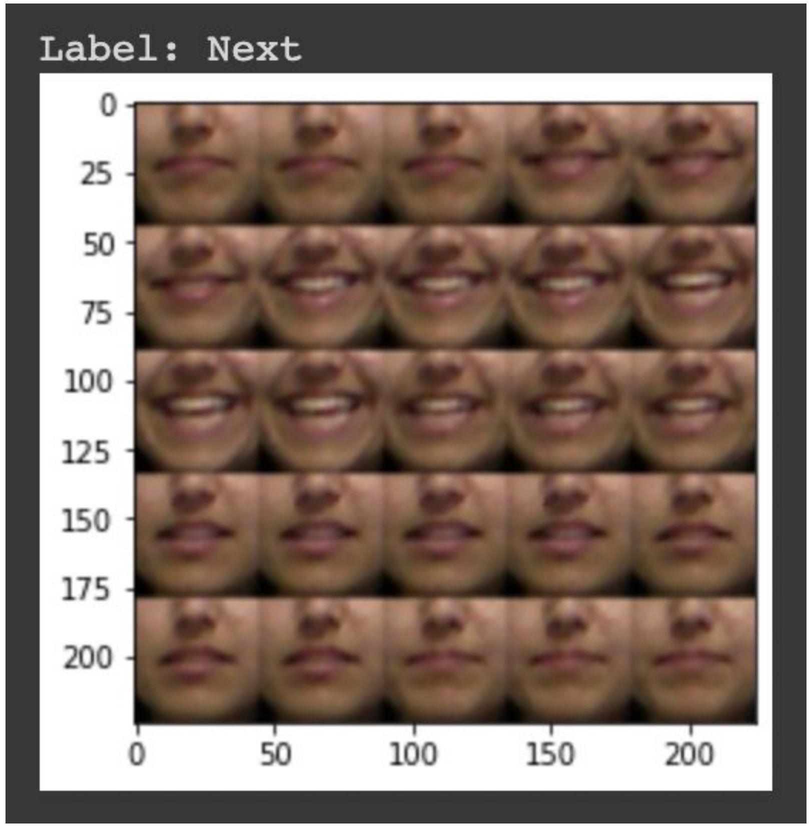
Data Pre Processing

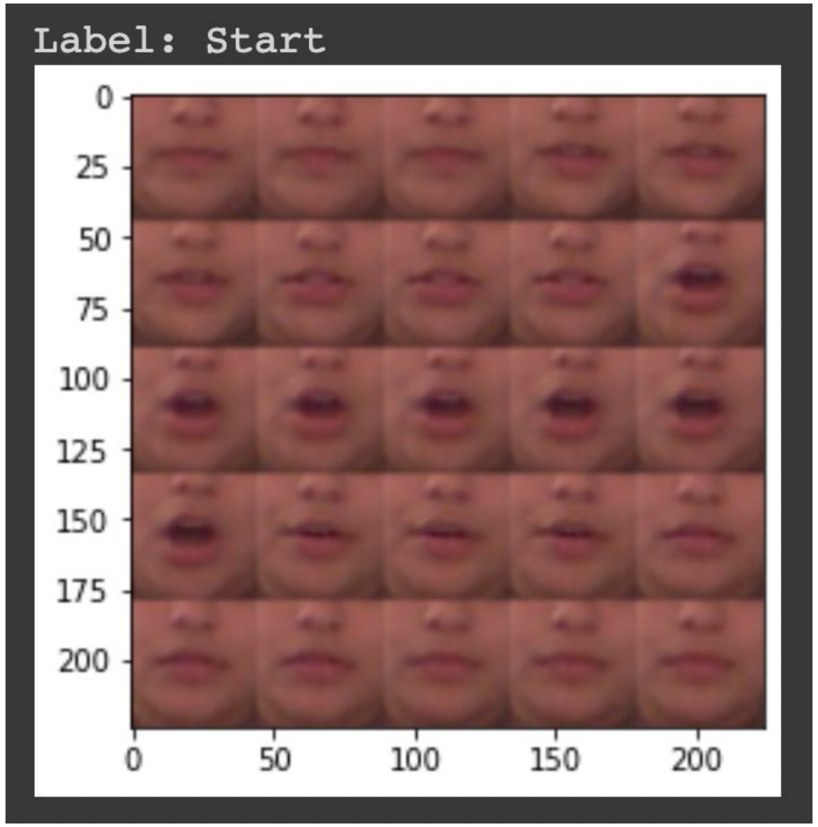
- We got the rectangular region that consists all these points and padded it such that the the regions width and height are 50 and then resized it to get an image of size 100 x 100.
- Step 2 - Merging all the frames of a word to a single image.
  - For this we decided to fit all the frames into a grid of 5 x 5.
  - If the current frames are less than 25, first we divided the slots equally to each frame.
  - Then we gave the excess slots to the images which differ a lot from its previous frame.
  - Then we merged all the frames in the order of occurring and resized the merged image of size 224 x 224.


Before Stretching


After Stretching

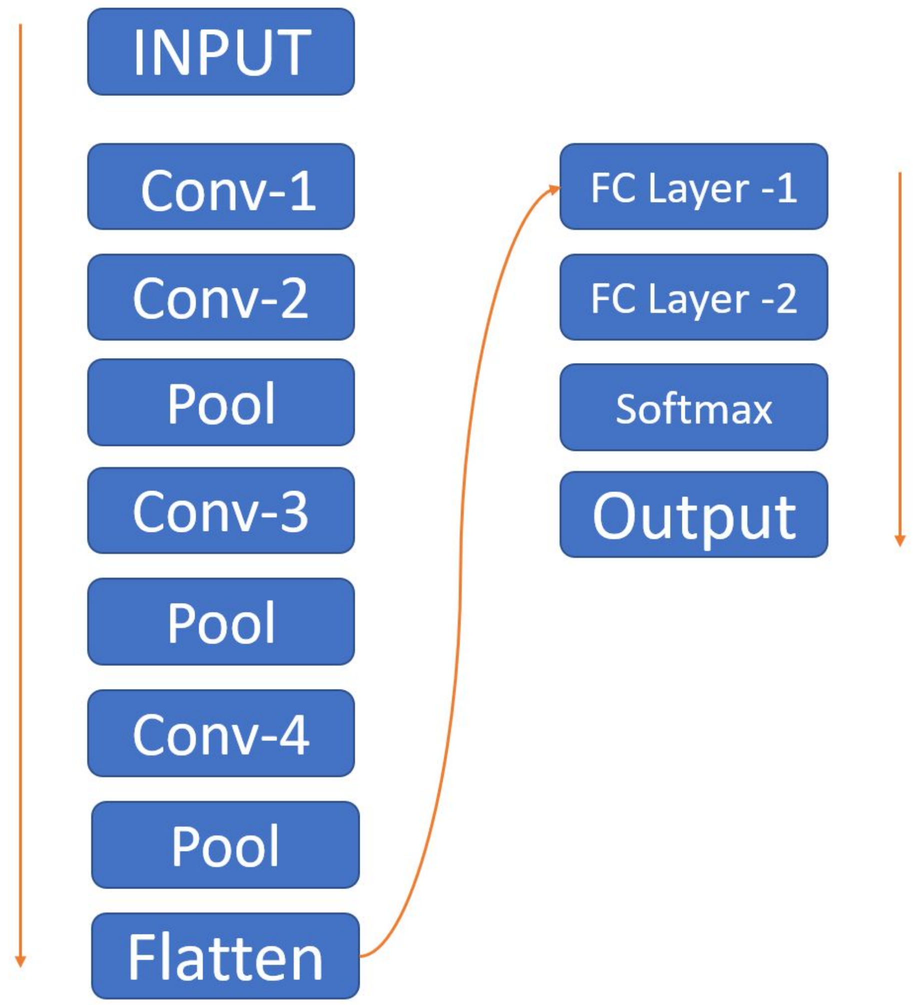
This is how the pre-processed data looks like





CNN Model

- Our CNN model is a linear stack of CNN Layers and fully connected layers.
- First we added two Conv2D layers with ReLu activation, then a maxpool layer.
- Then we added 2 sets of a Conv2D layer with ReLu activation and a maxpool layer.
- Then we flattened the output.
- We added two fully connected layers and softmax activation at the end, to get a 10 length vector output.

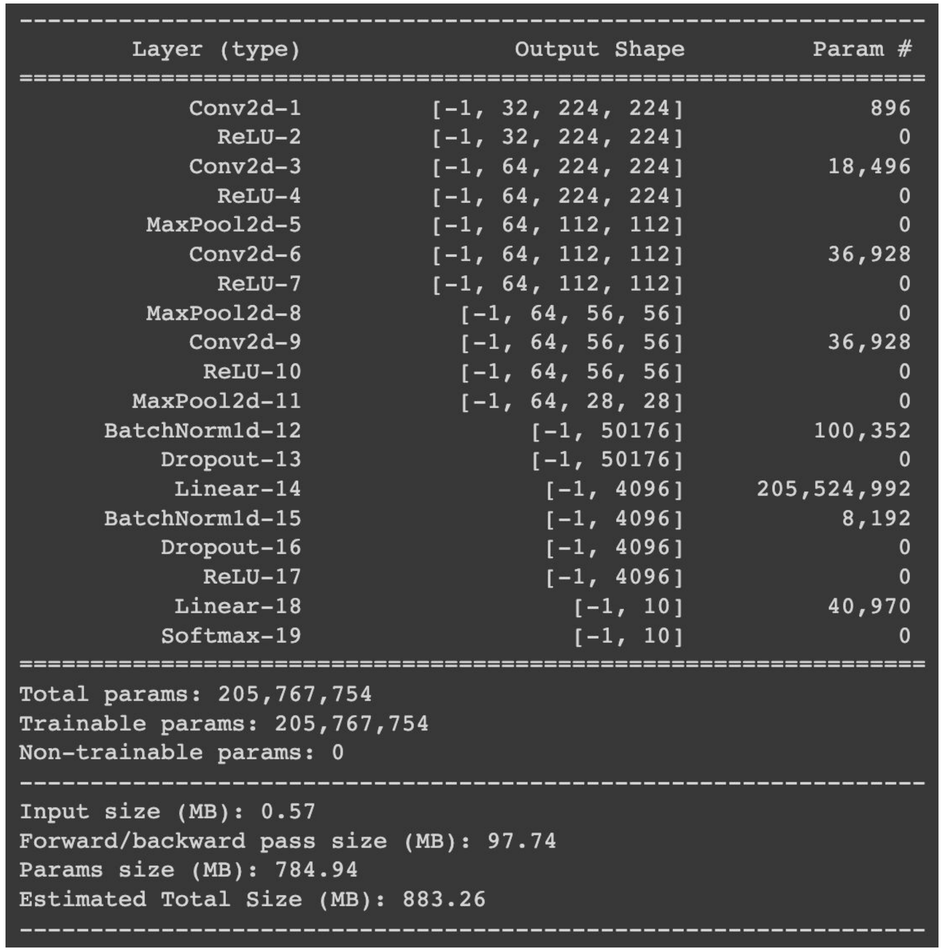


CNN Model

- We added dropouts and batch-normalisation before and after the first fully connected layer.
- Loss function used - Cross Entropy between the predictions and labels.
- Optimizer used while backpropagation - RMSprop with learning rate 0.001 and α=0.9.

Parameters in CNN Model

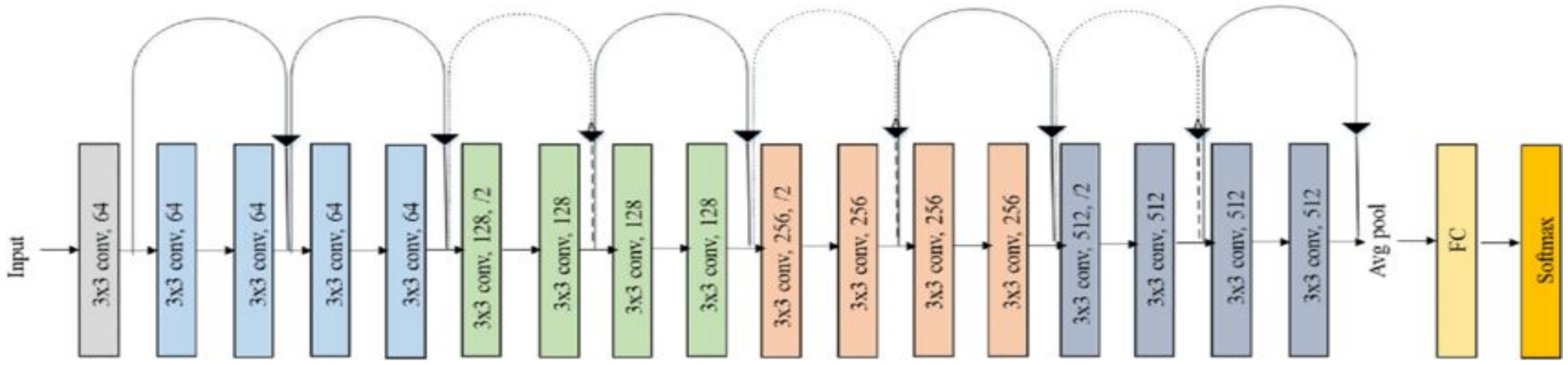
- The parameters are described in the figure attached, which total to 205,524,992 learnable parameters.
- The convolution layers contribute 93248 (896+18496+36928+36928) learnable parameters to our model.
- The fully connected neural network has 205,524,992 parameters in hidden layer and 40,970 in output layer.
- The batchnorm used before hidden and output layer have 108544 parameters.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 32, 224, 224]             896
              ReLU-2         [-1, 32, 224, 224]               0
            Conv2d-3         [-1, 64, 224, 224]          18,496
              ReLU-4         [-1, 64, 224, 224]               0
         MaxPool2d-5         [-1, 64, 112, 112]               0
            Conv2d-6         [-1, 64, 112, 112]          36,928
              ReLU-7         [-1, 64, 112, 112]               0
         MaxPool2d-8           [-1, 64, 56, 56]               0
            Conv2d-9           [-1, 64, 56, 56]          36,928
             ReLU-10           [-1, 64, 56, 56]               0
        MaxPool2d-11           [-1, 64, 28, 28]               0
      BatchNorm1d-12                [-1, 50176]         100,352
          Dropout-13                [-1, 50176]               0
           Linear-14                 [-1, 4096]     205,524,992
      BatchNorm1d-15                 [-1, 4096]           8,192
          Dropout-16                 [-1, 4096]               0
             ReLU-17                 [-1, 4096]               0
           Linear-18                   [-1, 10]          40,970
          Softmax-19                   [-1, 10]               0
================================================================
Total params: 205,767,754
Trainable params: 205,767,754
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 97.74
Params size (MB): 784.94
Estimated Total Size (MB): 883.26
----------------------------------------------------------------
```

ResNet Model

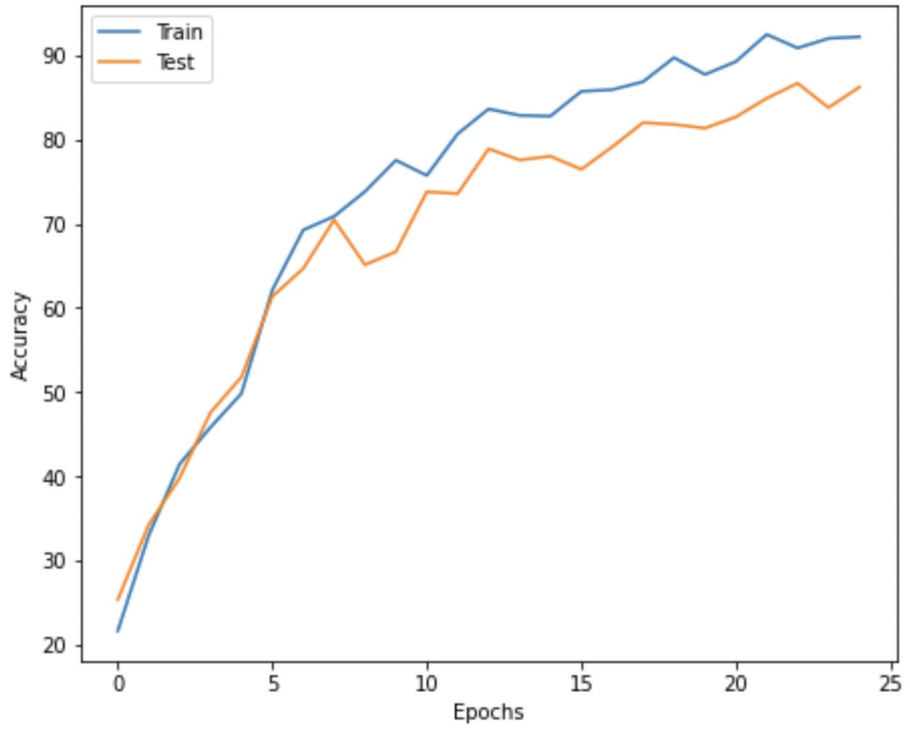
- We compared our CNN model with the Residual Network (ResNet) model.
- It is deeper network than CNN, and is still less complex because of using residual block.
- We used ResNet 18 model, pre-trained with human faces of celebrities from IMDB



Results after Training

1. **CNN Model :**

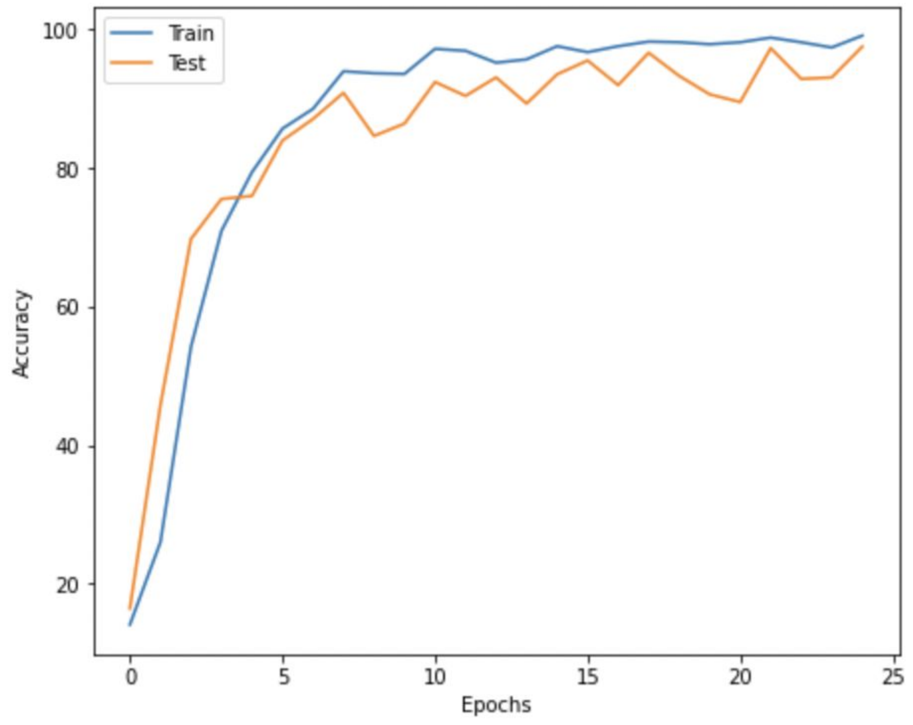
- The model had 205,767,754 trainable parameters and non-trainable parameters.
- We executed the training for 25 epochs.
- First epoch: Train accuracy=21.62%, Test Accuracy=25.33%.
- 25th epoch: Train accuracy=92.19%, Test Accuracy=86.22%.
- Accuracies improve over epochs and attain a decent stability at the end of 25 epochs.



Results after Training

2. **ResNet Model :**

- The model had 11,181,642 trainable parameters and 0 non-trainable parameters.
- We executed the training for 25 epochs.
- First epoch: Train accuracy=14.09%, Test Accuracy=16.44%.
- 25th epoch: Train accuracy=99.14%, Test Accuracy=97.55%.
- Accuracies attain a decent stability at the end of 25 epochs.



Comparison of the two Models

- We used the same dataset to train two different models. So their accuracies can be compared to know which performed better.
- The CNN model produced lesser accuracy than the ResNet model both in training and testing.
- So, ResNet performed better than our CNN model overall.
- ResNet is deeper than CNN, with more number of layers, which boosts its accuracy.
- It has residual block which makes it easier to train models with many layers,

Conclusion

- We compared two approaches for implementing Automated Lip Reading, our CNN model and the pre-trained ResNet model.
- We got the result that ResNet is a better model for Lip Reading task, as compared to our CNN model.
- However, our CNN model also gave a decent train and test accuracy, even though it had lesser number of layers.
- CNNs are very powerful deep networks.
- We also found that data pre-processing is an essential task for such DL models.
- We were unable to implement LSTM models, due to longer training times and difficult tuning of parameters.

References

The Links to the Literatures, Python Libraries Used, and Dataset can be found below:

- Dataset Used : MIRACL-VC1
- Library : Dlib
- Library : OpenCV
- Literature : Lip Reading Using CNN and LSTM
- Literature : Lip Reading Word Classification

Contributions

- CS19BTECH11045 – Kommana Vikas
  - Literature Review
  - Coding part for pre-processing
- CS19BTECH11054 – Nistala Praneeth
  - Coding part for pre-processing
  - Helped in Report
- AI20BTECH11020 – Sujal
  - Making the Report
  - Literature Review
  - Coding part for splitting train and test sets
- AI20BTECH11022 – Vaibhav Chhabra
  - Literature Review
  - Coding part for the models
  - Making the Report
- AI20BTECH11023 – Vishwanath Hurakadli
  - Coding part for the models
  - Making the Report