Audio-Visual Speech Recognition using DNN

Himanshu Pandotra
Kanhaiya Kumar

Agenda

- Introduction
- Neural Network Components
  - Convolution Neural Network
  - RNN/LSTM
  - Connectionist Temporal Classification
- Feature Extraction
- Deep learning Architecture
- Experiments and Results
- Challenges
- Conclusion and Future work

Introduction

- Audio Visual Speech Recognition

In AVSR, the aim is to combine two modalities (audio and video) to improve speech recognition results.

The idea is to extract meaningful information from the lip movement of the speaker along with the audio information to infer speech.

Here we will be using a deep learning based approach for solving this problem.

Introduction

- Why Neural Networks for AVSR?
  - Generalized Models
    - No complex language models
    - Robust to noise
  - End-to-End systems
  - Modelling Temporal Information
  - Doesn’t require phone level segmentation
  - Integration of Features from different modalities
    - Audio and Video Features

Neural Network Components

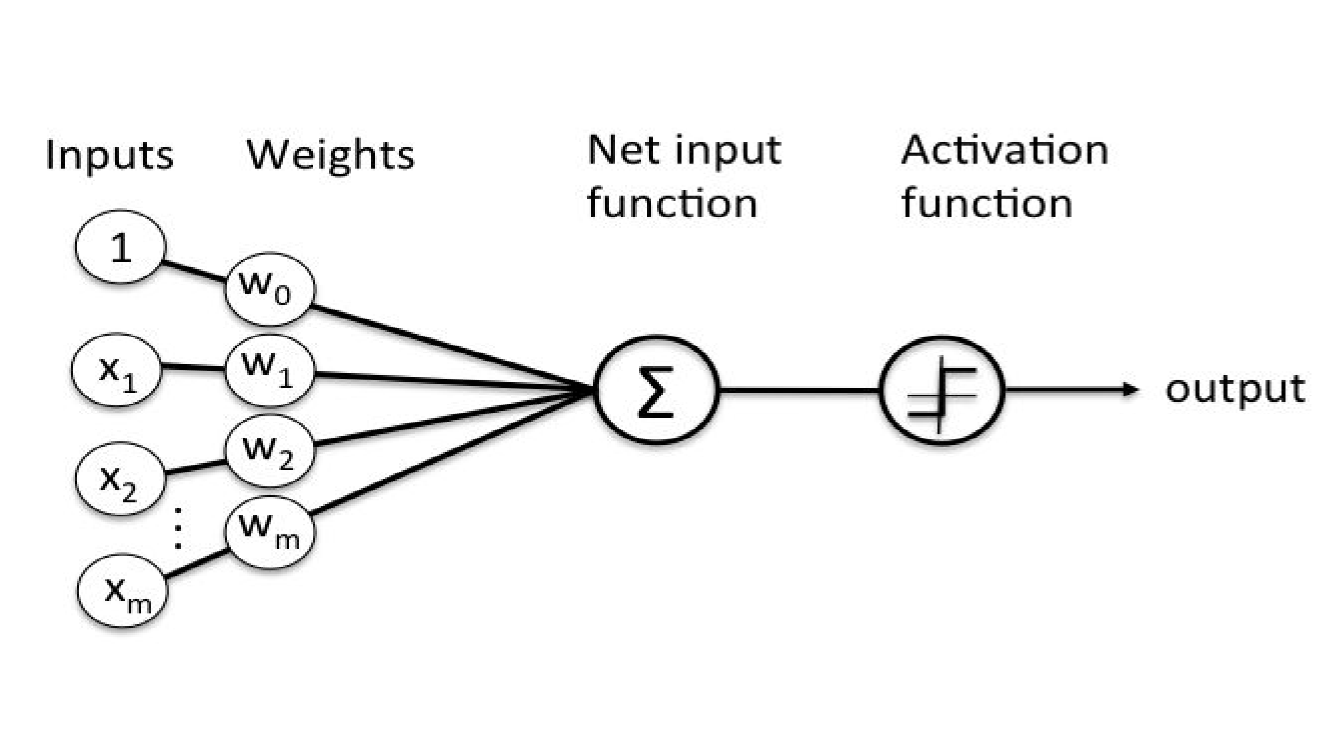
- Neural Network Basics
- Convolution Neural Network
- Recurrent Neural Network / LSTM
- Connectionist Temporal Classification Loss

Neural Networks Basics

- Perceptron
- Activation
  - Sigmoid
  - tanh
  - ReLU
- Feed forward Neural Network

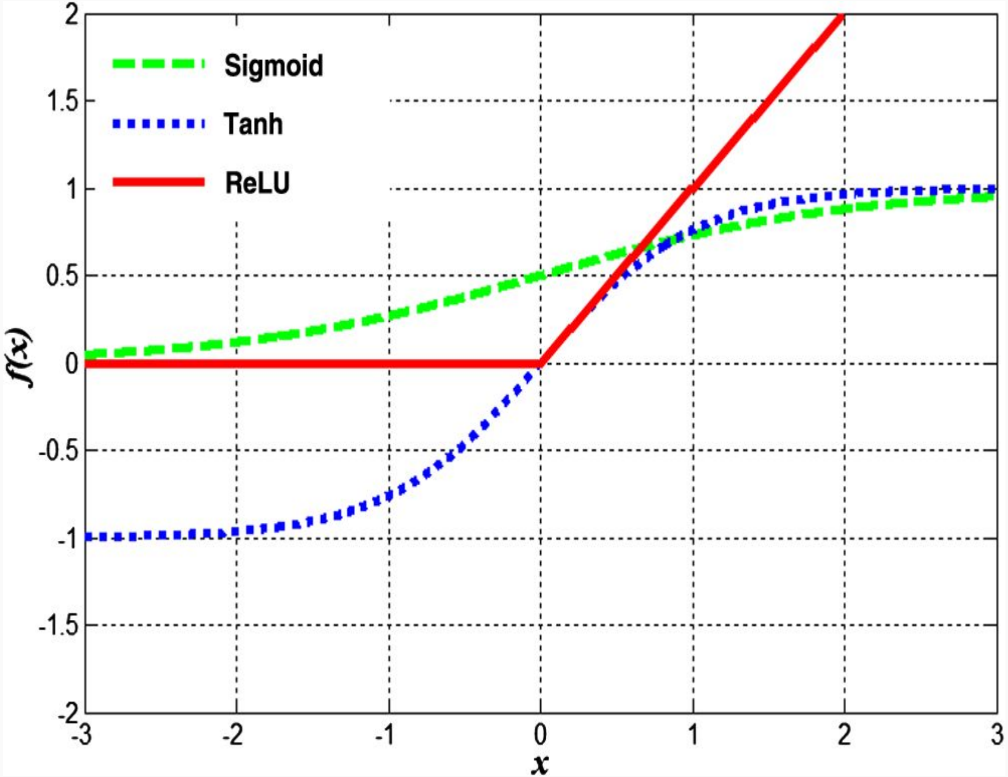
Perceptron

Building Block of Neural Network



Activation Function

- Activation functions break the linearity of a neural network, allowing it to learn more complex functions than linear regression.

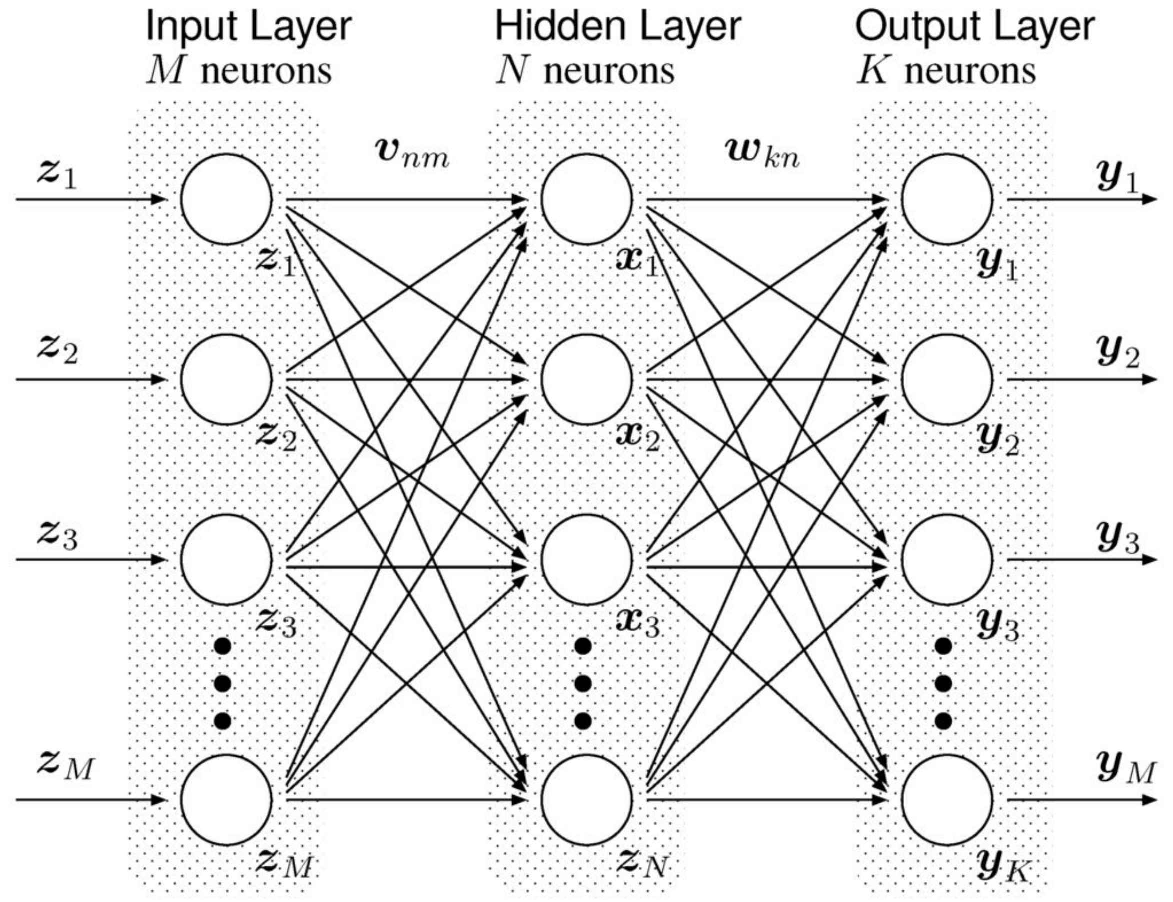


Source : Google Images

Advantage of ReLU over other activations

- Computing gradient for ReLU is easier because of its linear nature on positive side and zero on -negative side.
- Non- vanishing gradient ensuring the motion towards minimum.
- Sparsity:
  - for -negative activation it gives zero, so has higher sparsity.
  - Sparsity is good from computational point of view.

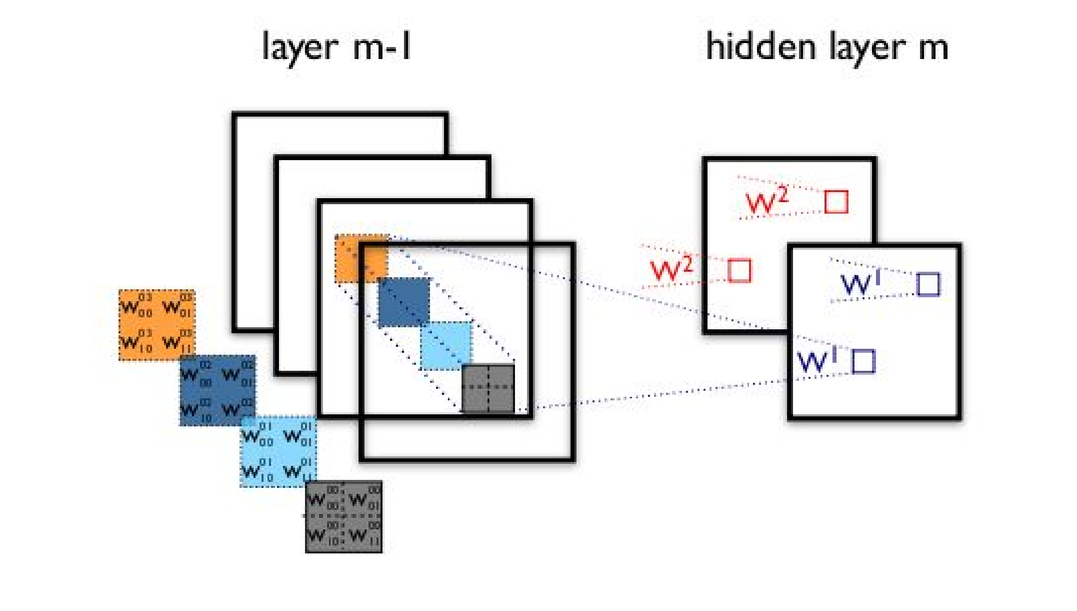
Feed Forward Neural Network



Source : Google Images

Convolutional Neural Networks

- The CNN are very similar to standard Neural Network but instead of fully connected neurons between two layers,we have only local connections.
- These weights act as filters which are locally connected.
- The benefit of convolution layer is the reduced number of weight parameters to learn.



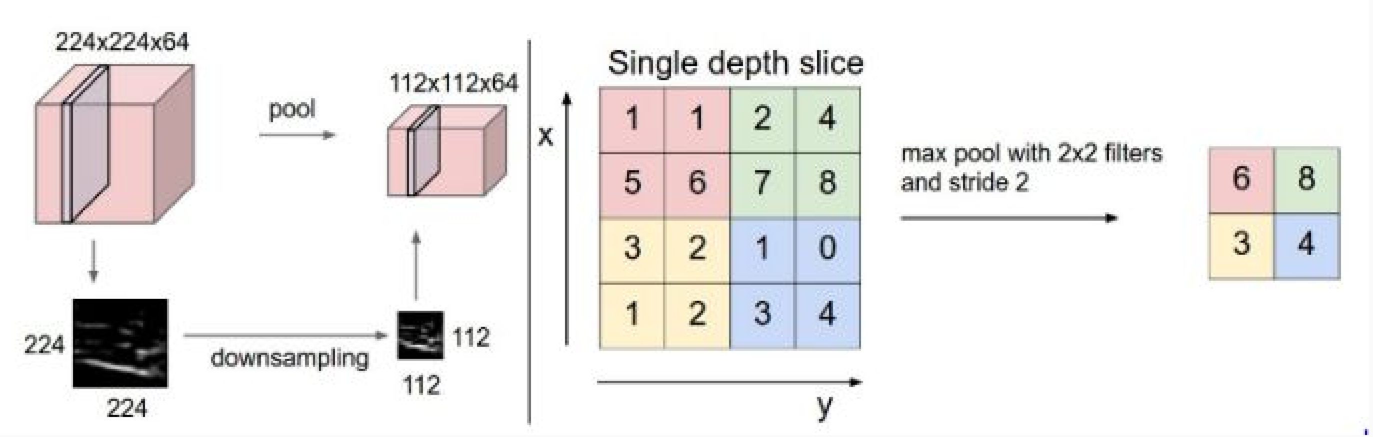
layer m-1
hidden layer m
W^{03}_{00} W^{03}_{01}
W^{03}_{10} W^{03}_{11}
W^{02}_{00} W^{02}_{01}
W^{02}_{10} W^{02}_{11}
W^{01}_{00} W^{01}_{01}
W^{01}_{10} W^{01}_{11}
W^{00}_{00} W^{00}_{01}
W^{00}_{10} W^{00}_{11}
W^2
W^2
W^1
W^1

Source : Google Images

Convolutional Neural Networks

Max pooling

- It is a form of non-linear down-sampling.
- It decreases the dimension of its input by taking only the maximum value from a fixed region of convolutional layer. Hence, reduces the number of parameters to train.



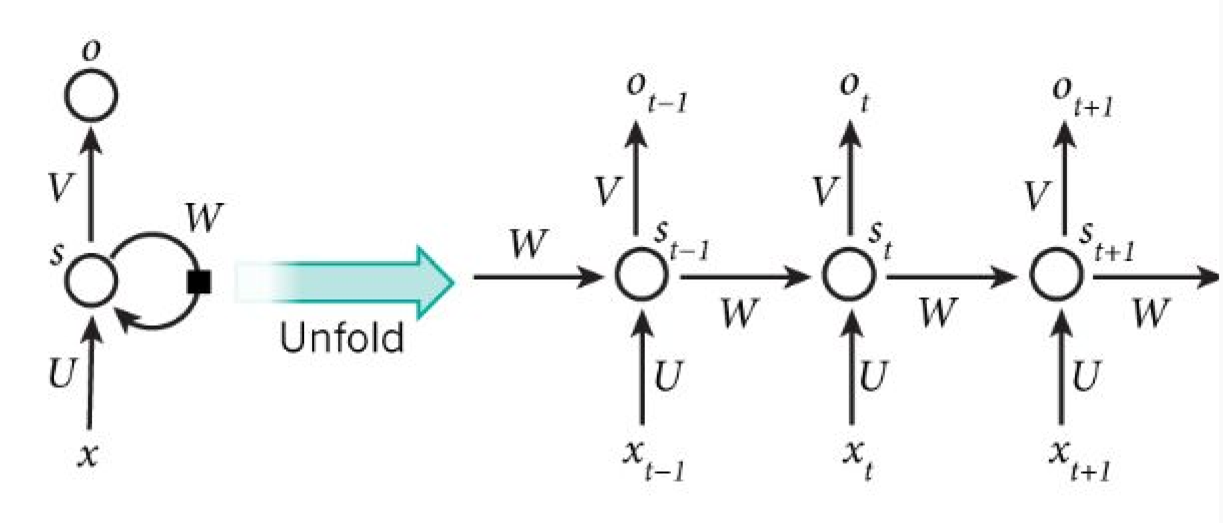
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Source : Google Images

Recurrent Neural Network

- Recurrent Neural Network is a model which takes into account the temporal aspects of data.
- The neuron in RNN has a looping structure and the information from current time step is carried to the next time steps.

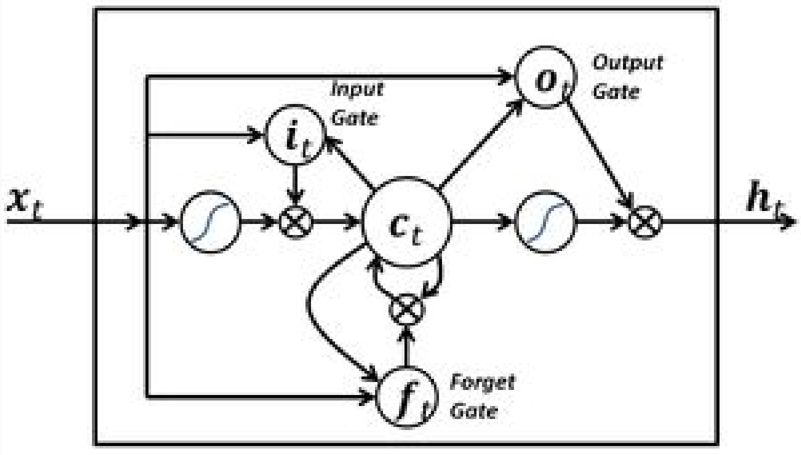


Source : Google Images

Recurrent Neural Networks/Long Short Term Memory

- The RNN suffer from the problem of vanishing gradients. The limitation of learning small context exists in RNN.
- A LSTM unit is a recurrent network unit that excels at remembering values for either long or short durations of time.

- The forget gate decides which information from previous time steps should flow to the current state.
- So, during gradient descent via backpropagation through time, the information of only few previous states is taken into account and rest is ignored via forget gate, preventing vanishing gradients.



Source : Google Images

Connectionist Temporal Classification

- CTC is a loss function useful for performing supervised learning on sequence data, without needing an alignment between input data and labels.
- It maximizes the probability:

$$p(l|\mathbf{y}) = \sum_i p(l|\pi_i)p(\pi_i|\mathbf{y})$$

P(_ _ CC _ _ _ AAA _TT)
....
....   +   ⇒ P(CAT)
....
P(_C_ _ A A _ A _ _T _ _)

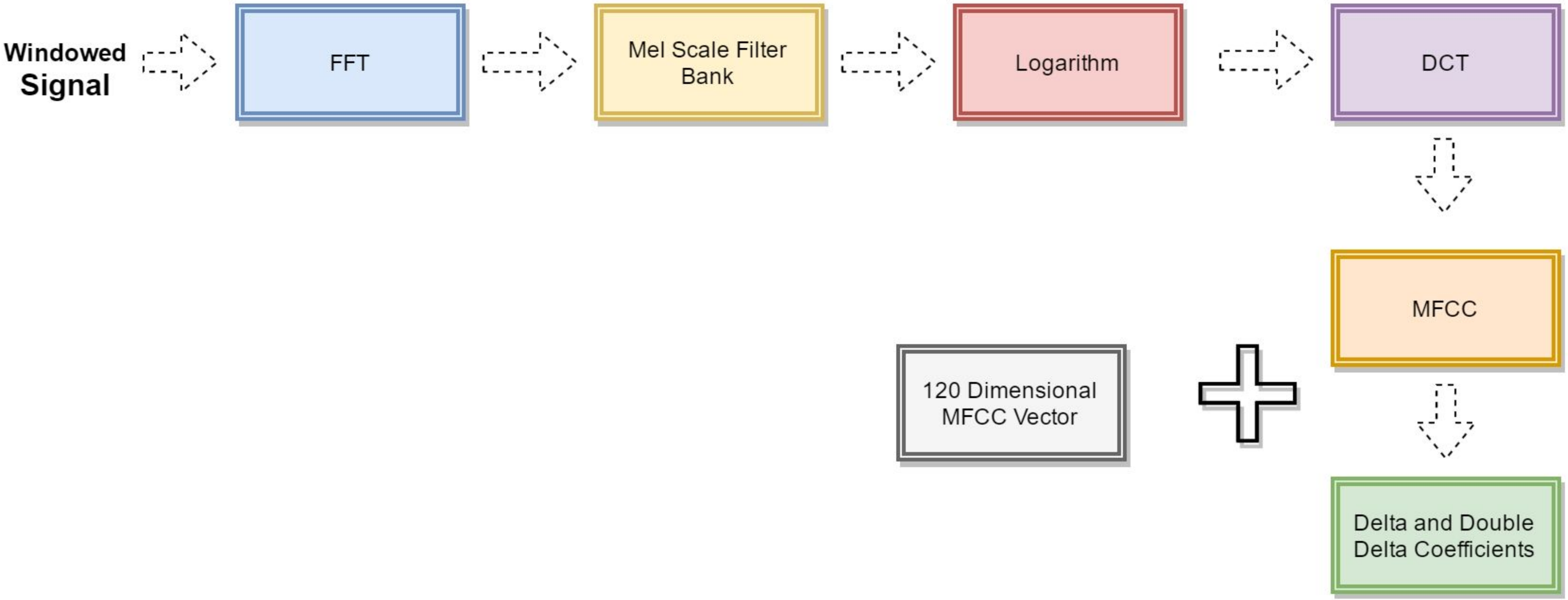
where, **l** is the actual label and **y** is the input to the CTC layer,

$p(l|\pi_i)$ is the probability of the label given a path **π_i**,

$p(\pi_i|y)$ is the probability of a path **π_i** given the input y.

Feature Extraction

- **Audio Features → Mel-Frequency Cepstral Coefficients**

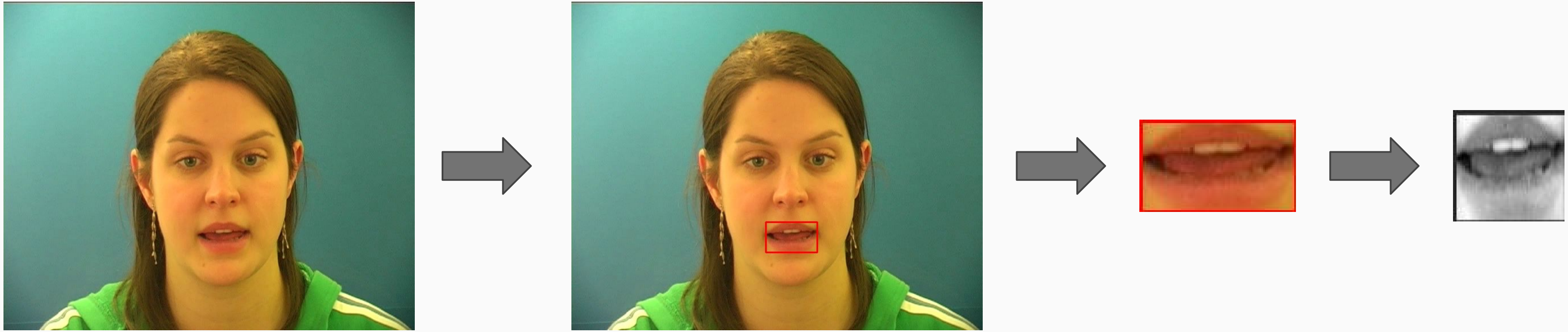

Windowed Signal
FFT
Mel Scale Filter Bank
Logarithm
DCT
MFCC
Delta and Double Delta Coefficients
120 Dimensional MFCC Vector

Feature Extraction

- Audio Features :
  - We use the standard MFCC coefficients of length 40 as audio features.
  - We also concatenate the derivatives and double derivatives of the cepstral coefficients.
  - We get a 120x1 feature vector for every audio frame.

Video Features

- The video feature corresponds to the mouth region of the speaker in each frame which gives the movement of lips of the person speaking.
- We extract the mouth region from the video frames and rescale to a 64x64 image.



Video Features

- The extraction of the mouth region is done by recognition of important landmark points on the face.
- We use a **pre-learnt** classifier which has been trained on face images. The model is trained by classification using ‘Histogram of oriented Gradients’ (HOG) features.

Source : http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2

Deep Learning Architecture

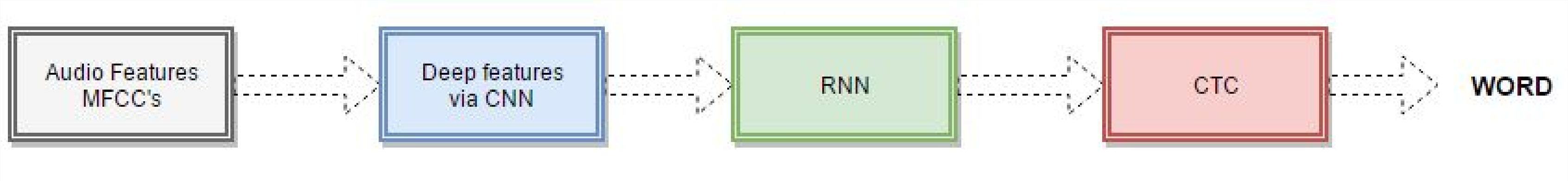
We consider three separate Speech Recognition networks:

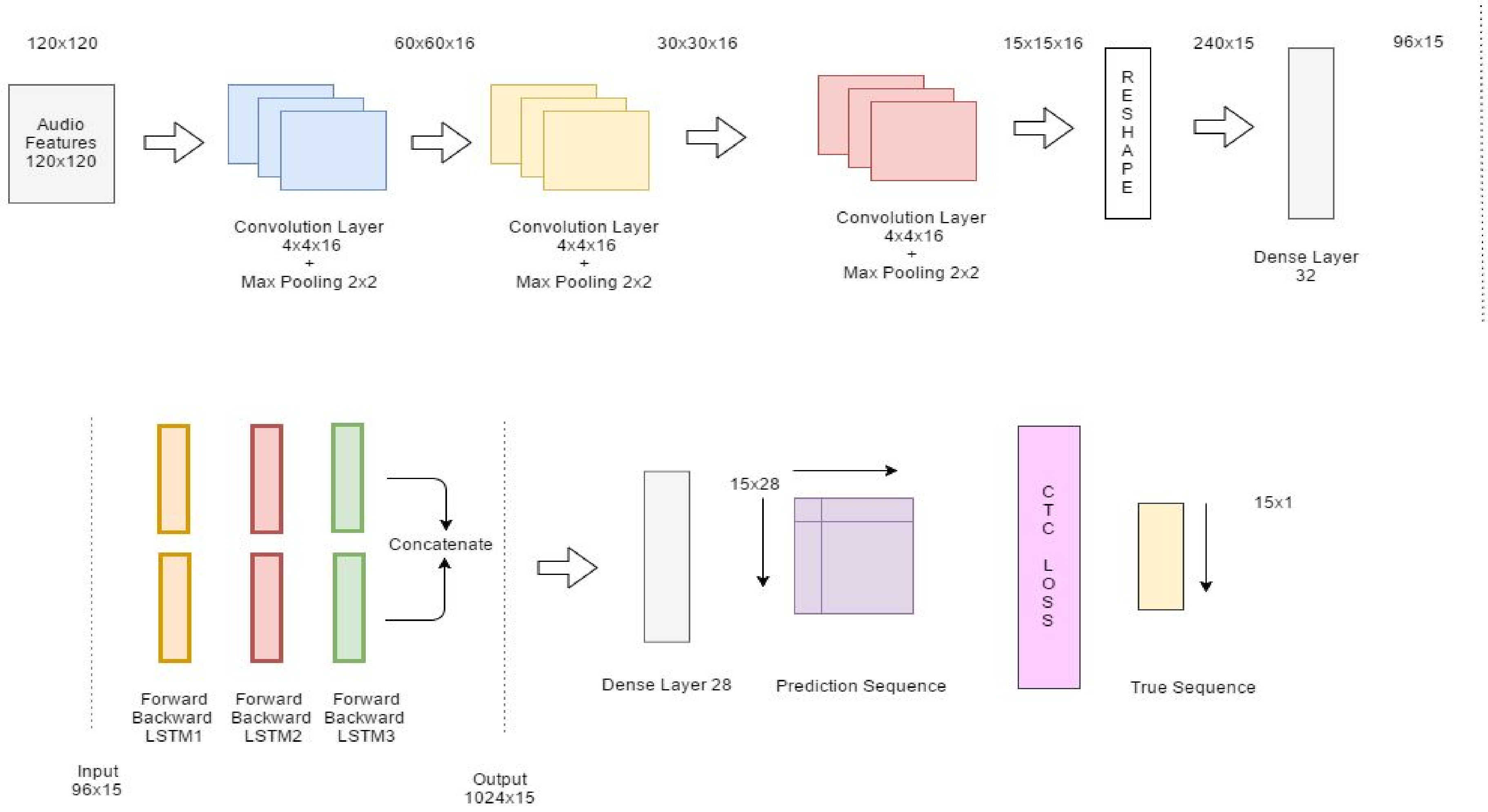
1. Using only Audio-Features (Audio-Network)
2. Using only Visual Features (Visual-Network)
3. Using combined Audio-Visual Features (Audio-Visual Network)

Audio Neural Network

The input audio rate is 50 KHz. The audio MFCC vectors are calculated for a frame length of 10ms with 5ms hop length.

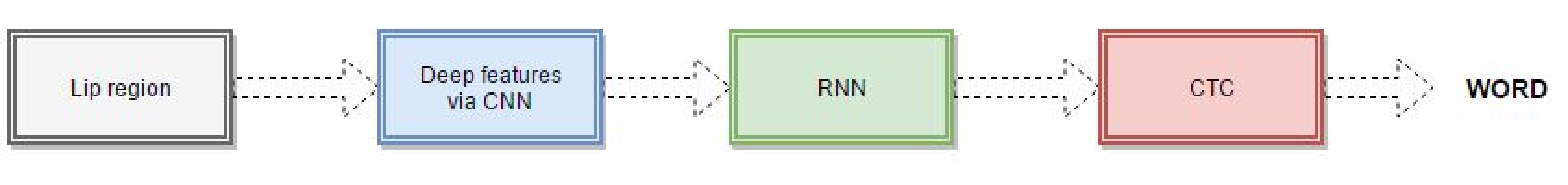
Since each spoken word has different duration, the number of MFCC feature vectors associated with each word is different. By zero padding, this is normalized to length 120 for each word. So we get a 120 x 120 audio feature associated with each word.

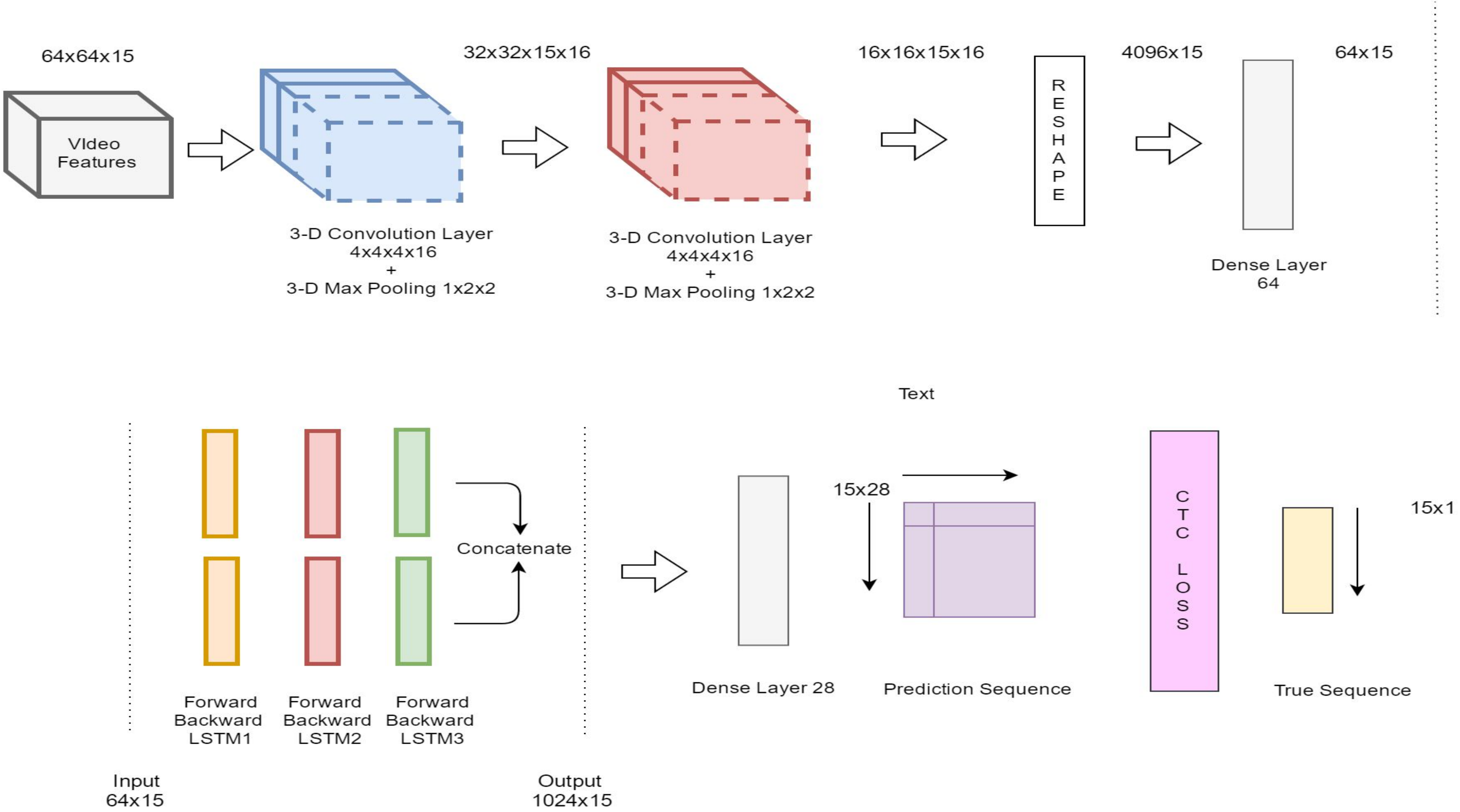



120x120
Audio Features 120x120
60x60x16
Convolution Layer 4x4x16 + Max Pooling 2x2
30x30x16
Convolution Layer 4x4x16 + Max Pooling 2x2
15x15x16
Convolution Layer 4x4x16 + Max Pooling 2x2
RESHAPE
240x15
Dense Layer 32
96x15
Forward Backward LSTM1
Forward Backward LSTM2
Forward Backward LSTM3
Concatenate
Input 96x15
Output 1024x15
Dense Layer 28
15x28
Prediction Sequence
CTC LOSS
15x1
True Sequence

Visual Neural Network

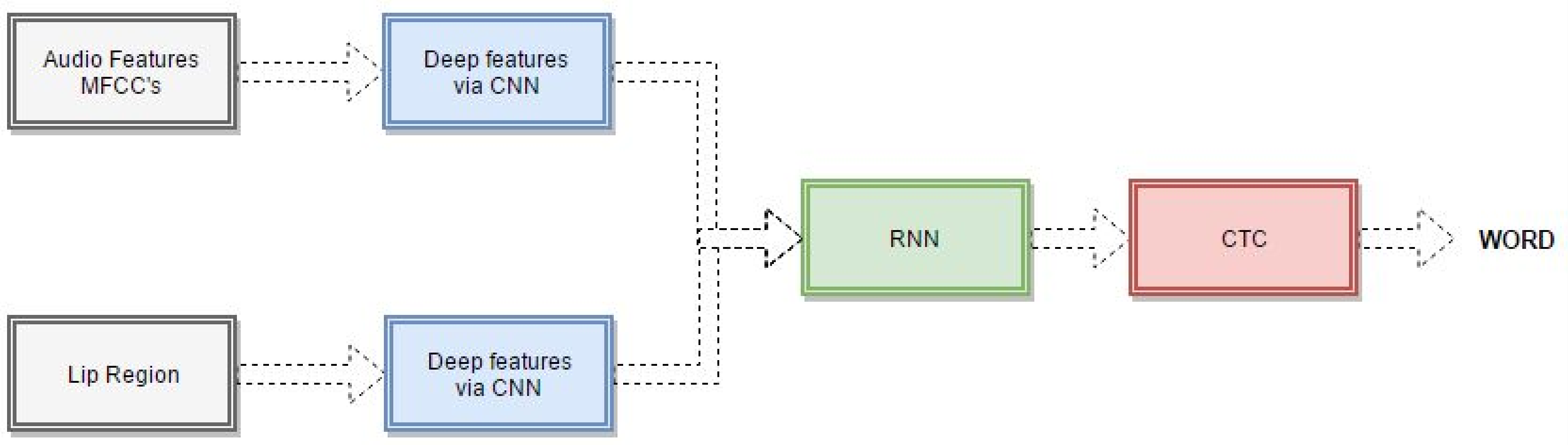
The input visual feature is the 64x64 lip region image. Similar to the audio case, we zero pad features for each word to have a length of 64x64x15. Now the input to the network is a 3D feature vector.

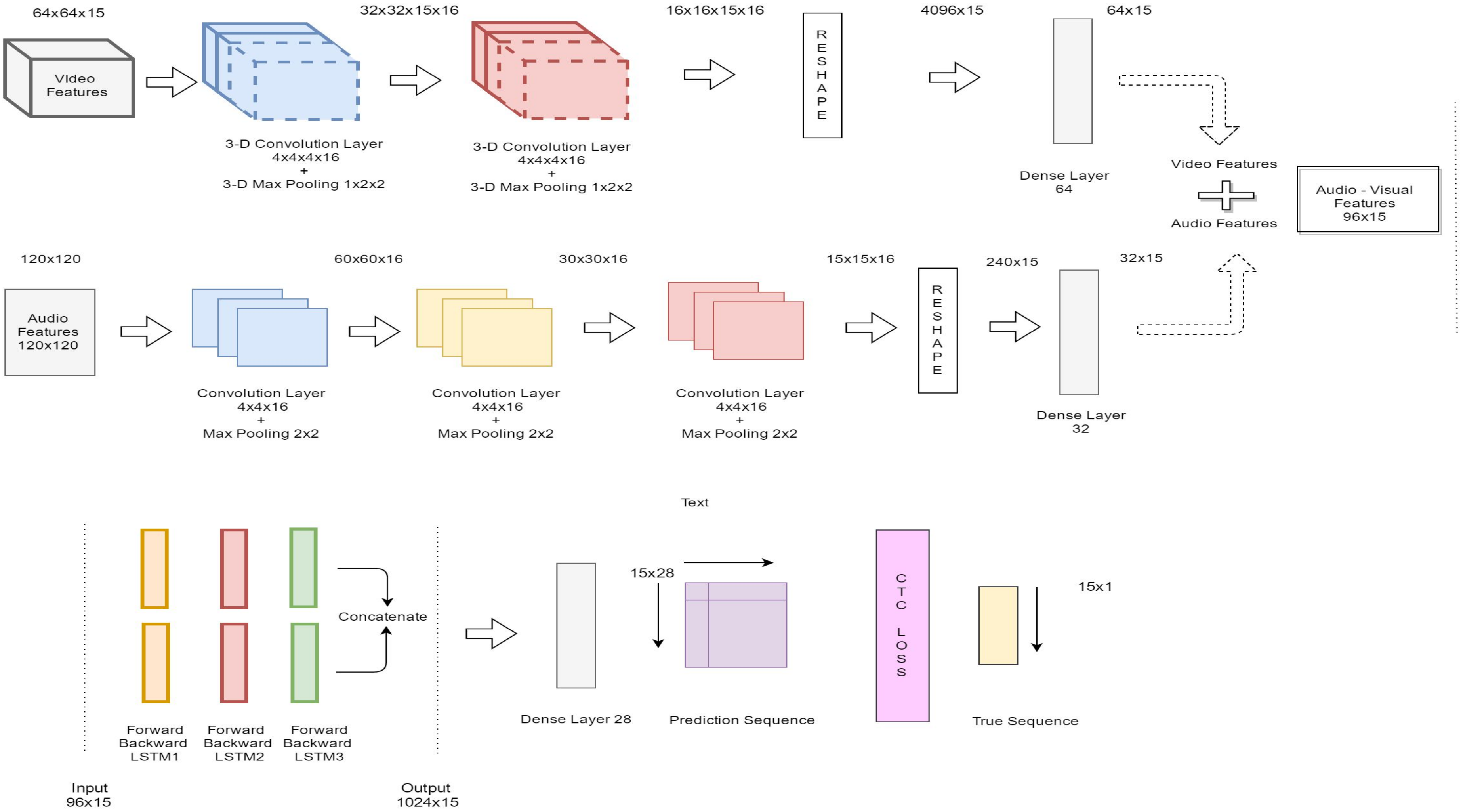



64x64x15
VIdeo Features
32x32x15x16
3-D Convolution Layer 4x4x4x16 + 3-D Max Pooling 1x2x2
16x16x15x16
3-D Convolution Layer 4x4x4x16 + 3-D Max Pooling 1x2x2
RESHAPE
4096x15
64x15
Dense Layer 64
Forward Backward LSTM1
Forward Backward LSTM2
Forward Backward LSTM3
Concatenate
Input 64x15
Output 1024x15
Dense Layer 28
Text
15x28
Prediction Sequence
CTC LOSS
15x1
True Sequence

Joint Audio+Visual Network

- We extract features from the audio and video streams in parallel and then concatenate them before giving it to LSTM. The rest of the network structure remains same.




64x64x15
VIdeo Features
32x32x15x16
3-D Convolution Layer 4x4x4x16 + 3-D Max Pooling 1x2x2
16x16x15x16
3-D Convolution Layer 4x4x4x16 + 3-D Max Pooling 1x2x2
RESHAPE
4096x15
64x15
Dense Layer 64
Video Features
Audio Features
Audio - Visual Features 96x15
120x120
Audio Features 120x120
60x60x16
Convolution Layer 4x4x16 + Max Pooling 2x2
30x30x16
Convolution Layer 4x4x16 + Max Pooling 2x2
15x15x16
Convolution Layer 4x4x16 + Max Pooling 2x2
RESHAPE
240x15
32x15
Dense Layer 32
Forward Backward LSTM1
Forward Backward LSTM2
Forward Backward LSTM3
Concatenate
Input 96x15
Output 1024x15
Dense Layer 28
Text
15x28
Prediction Sequence
CTC LOSS
15x1
True Sequence

Experiments and Results

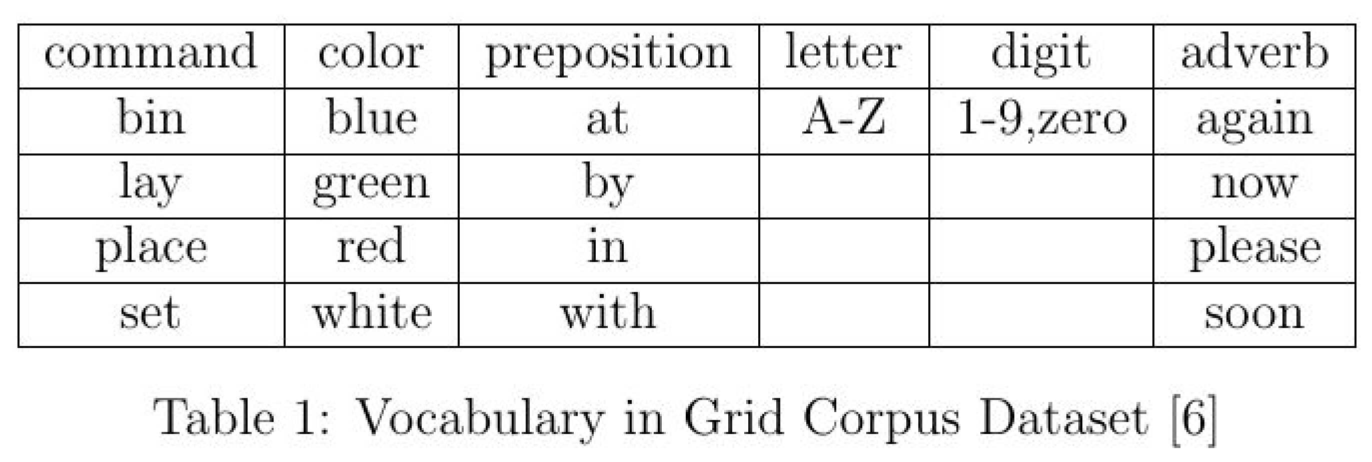
Dataset : ‘**The GRID AudioVisual Sentence Corpus**’

http://spandh.dcs.shef.ac.uk/gridcorpus/

- The dataset consists of video recordings of 33 people with 1000 sentences per speaker.
- Each video recording is of 3 second duration.
- The length of each sentence is 6 words.
- Finite vocabulary consisting of 52 words.
- The audio data is available at 50kHz and video at 25 fps.
- The speaker’s face is aligned with camera front and there is little motion.

Experiments and Results

Vocabulary : 52 words



command	color	preposition	letter	digit	adverb
bin	blue	at	A-Z	1-9,zero	again
lay	green	by			now
place	red	in			please
set	white	with			soon

Table 1: Vocabulary in Grid Corpus Dataset [6]

Experiments and Results

We use a subset of 6 speakers from the above dataset and learn and test our models on them.

- Training
  - Two speakers with 1000 sentences each.
- Testing
  - Four other speakers in our data subset.

Experiments and Results

Evaluation : We calculate prediction accuracy in following two ways:

1. The number of exact matches between the predicted word and the label.
2. The nearest neighbour of the predicted string from the vocabulary based on ‘edit distance’.

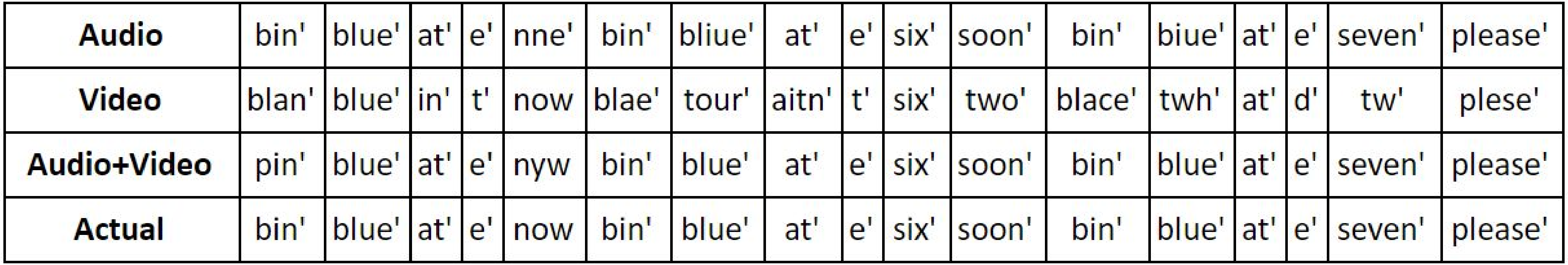
Edit Distance :

Edit Distance between seq ‘A’ and seq ‘B’ is defined as the minimum number of operations {deletion, insertion, substitution} required to convert one sequence to another.

Example: ED(‘number’ , ‘nnubr’) = 3

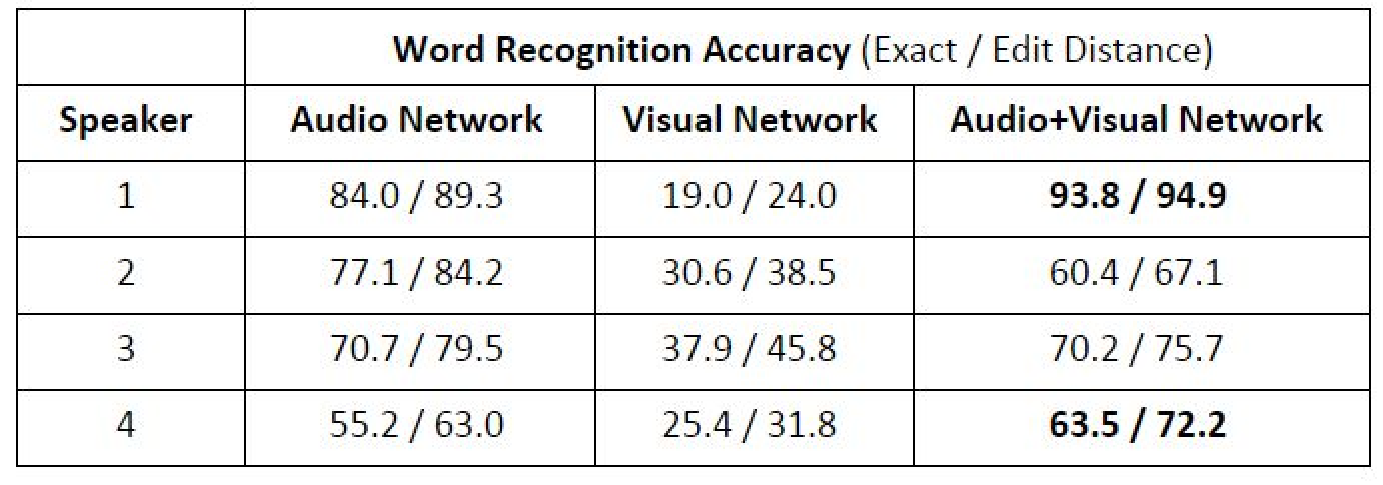
Results

- Examples of predicted words for the three trained networks



Audio	bin'	blue'	at'	e'	nne'	bin'	bliue'	at'	e'	six'	soon'	bin'	biue'	at'	e'	seven'	please'
Video	blan'	blue'	in'	t'	now	blae'	tour'	aitn'	t'	six'	two'	blace'	twh'	at'	d'	tw'	plese'
Audio+Video	pin'	blue'	at'	e'	nyw	bin'	blue'	at'	e'	six'	soon'	bin'	blue'	at'	e'	seven'	please'
Actual	bin'	blue'	at'	e'	now	bin'	blue'	at'	e'	six'	soon'	bin'	blue'	at'	e'	seven'	please'

Experiments and Results



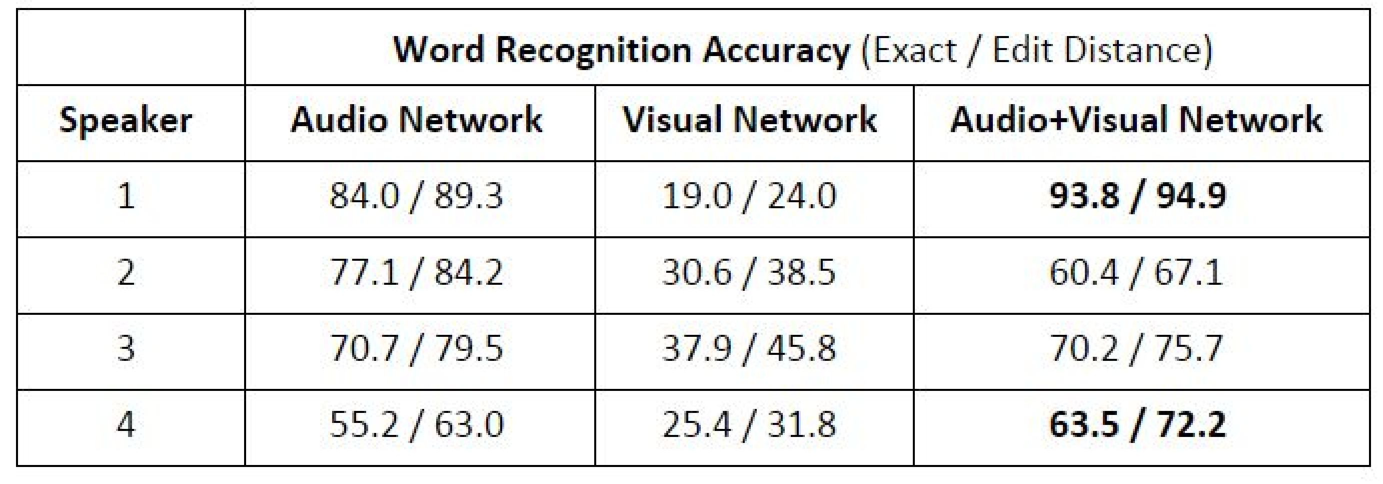
	Word Recognition Accuracy (Exact / Edit Distance)		
Speaker	**Audio Network**	**Visual Network**	**Audio+Visual Network**
1	84.0 / 89.3	19.0 / 24.0	**93.8 / 94.9**
2	77.1 / 84.2	30.6 / 38.5	60.4 / 67.1
3	70.7 / 79.5	37.9 / 45.8	70.2 / 75.7
4	55.2 / 63.0	25.4 / 31.8	**63.5 / 72.2**

Observations

1. We can observe that the recognition accuracy results for exact spelling match of predicted and actual word is reasonably close to the results calculated by edit distance. So the limited vocabulary is not a strong limitation.

2. The results for visual network are considerably bad as compared to other two networks. This was a bit expected because the ‘viseme’ to phone mapping is not one-to-one.

Experiments and Results



	Word Recognition Accuracy (Exact / Edit Distance)		
Speaker	**Audio Network**	**Visual Network**	**Audio+Visual Network**
1	84.0 / 89.3	19.0 / 24.0	**93.8 / 94.9**
2	77.1 / 84.2	30.6 / 38.5	60.4 / 67.1
3	70.7 / 79.5	37.9 / 45.8	70.2 / 75.7
4	55.2 / 63.0	25.4 / 31.8	**63.5 / 72.2**

Observations

3. We see that for speaker 1 and 4, the results for Audio+Visual network give better results than for just audio network. We expect this because of the extra information from video stream adding to audio features to give better results.

4. Though we can see that the above in not true for speaker 2,3. This suggests the model has not generalized well.

Experiments and Results

Possible reasons for lower accuracy with joint network for two speakers

1. The model for audio+visual network is complex with more training parameters. Since we have trained on smaller dataset, the model has not fully generalized.

2. The relative importance of audio and video features was not considered. We need to do discriminative training, where we switch off one data stream for some time and train on the other and then train on both later. Basically, we want to achieve a better local minimum for our objective function.

Challenges

- Handling large dataset
  - Extraction of lip region from all video.
  - Used HDF5 format to store data.
- Synchronization of audio and video streams
  - The video frame rate is ⅛ of the audio MFCC rate (200/sec).
  - We tried interpolation of video frames to match frame rates and then extract features.
  - Build a joint end-to-end network.
- Extracting meaningful video features
  - We tried using DCT features instead of deep features for video stream but the results were not good.

Future Work

- Explore and identify the different ways in which most relevant video features can be extracted and utilized (possibly changing Neural network architecture)
- Discriminative training for audio and video data to give importance to more relevant data
- Combining HMM based models with deep neural networks
- Handling multi-speaker environments with pose variations and speaker movement

Thank You