School of Engineering and Applied Science (SEAS), Ahmedabad University

B.Tech(ICT) Semester V: Wireless Communication (CSP 311)

- **Group No : 21**
- **Group Members**

Jay Patel (AU1741018)
Mohit Vaswani (AU1741039)
Manav Shah (AU1741042)
Prima Sanghvi (AU1741045)
Jaydeep Modi (AU1741070)

- **Title :** Vehicular Cyber Physical Security

  1) Robust Hierarchical Deep Learning for Vehicular Management

  2) Traffic congestion detection with deep feature using Convolutional Neural Network and Regression.

1 Introduction

1.1 Background

- Vehicle management system has expanded to add more fields in recent years. And an important system is traffic congestion detection. Vehicle traffic congestion is a major problem in the society. Accidents, diversions, poor traffic signal management has increased the volume of traffic, hence traffic congestion detection is necessary.
- Traffic Congestion Detection has been researched a lot and many methods are proposed , but the problem with these proposed methods are that they under-perform in real scenarios. So basically to solve this problem , a such algorithm must be designed which has a pretty acceptable generalization ability.

1.2 Motivation

For congestion detection there are existing algorithms but they work on traditional features which are not that discriminative hence results in rather poor performance under complex scenarios. Hence to solve this problem we need something realistic . Hence motivated by this fact we decided to contribute to this problem using Deep learning.

1.3 Contributions

- **Convolutional Nueral Network:-**

  In deep learning, Convolutional Neural Network(CNN) is standard neural network specifically designed to process data such as images [1].The components of CNN are convolutional layer, pooling layer and followed by a dense layer.Deep learning CNN models to train and test, each input image and pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply softmax function to classify an object with probabilistic values between 0 and 1[2].

  Convolutional Nueral Network is used because it helps extracting relevant features for traffic congestion detection , which than results into better accuracy for more complex scenarios.

  Convolutional Layer : Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

  Pooling Layer : Pooling layers basically reduce the number of parameters when the image, input data has too many parameters.

  Fully Connected Layer : Fully connected layers connect every neuron in one layer to every neuron in another layer. The last fully-connected layer uses a activation function for classifying the generated features of the input image into various classes based on the training dataset.

  The proposed Hierarchical Network consists of 4 convolutional layers, 4 pooling Layers and 4 fully connected layers.

- **Explanation of our Model :-**

  Convolutional Layer : Convolutional layers extract the features using filters,in the proposed network padding is used, so basically there is a filter, center element of which is placed on every element of input matrix and convolution is computed. By computing the convolution the features are extracted.

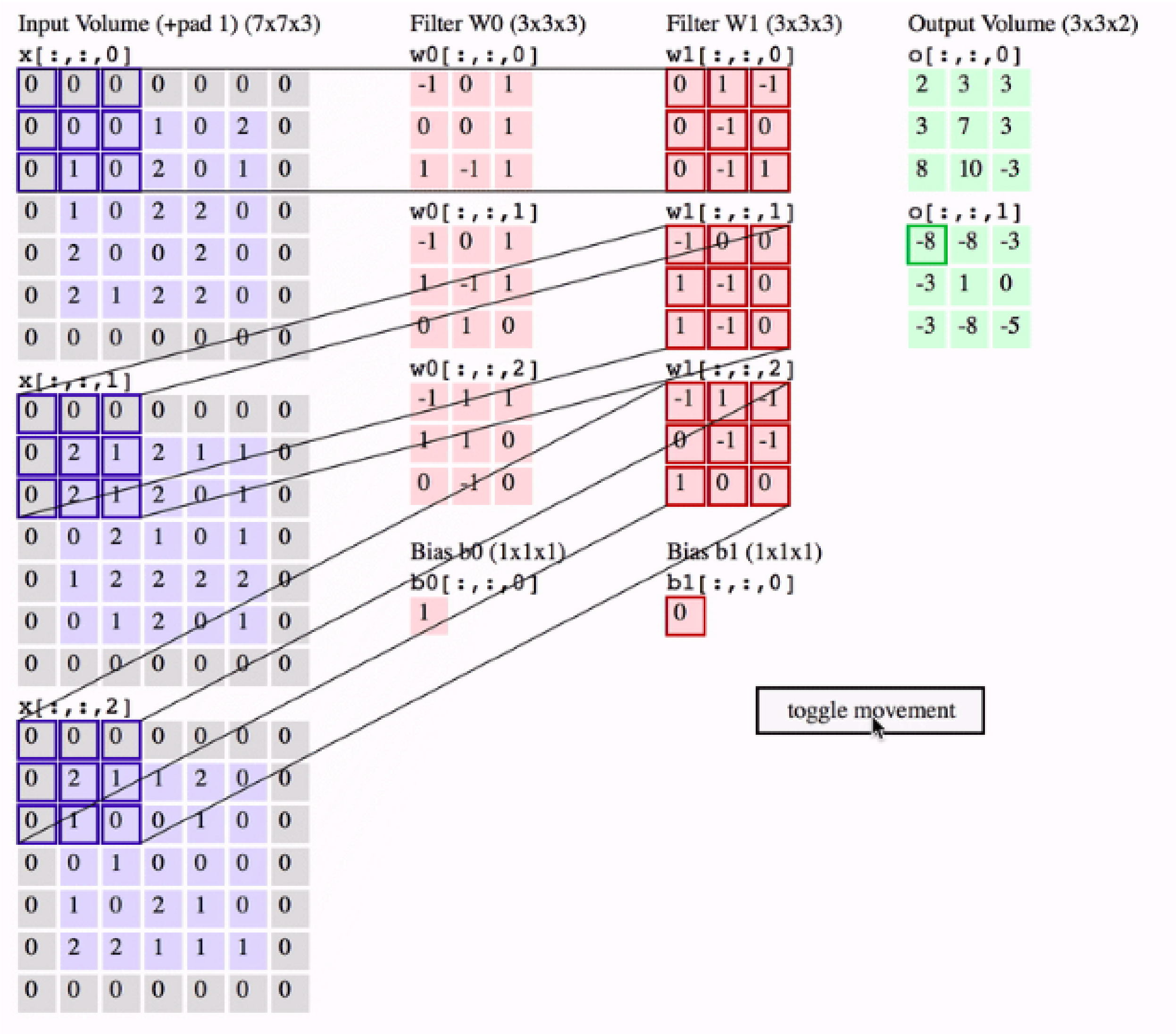

Figure 1: working of convolutional layer

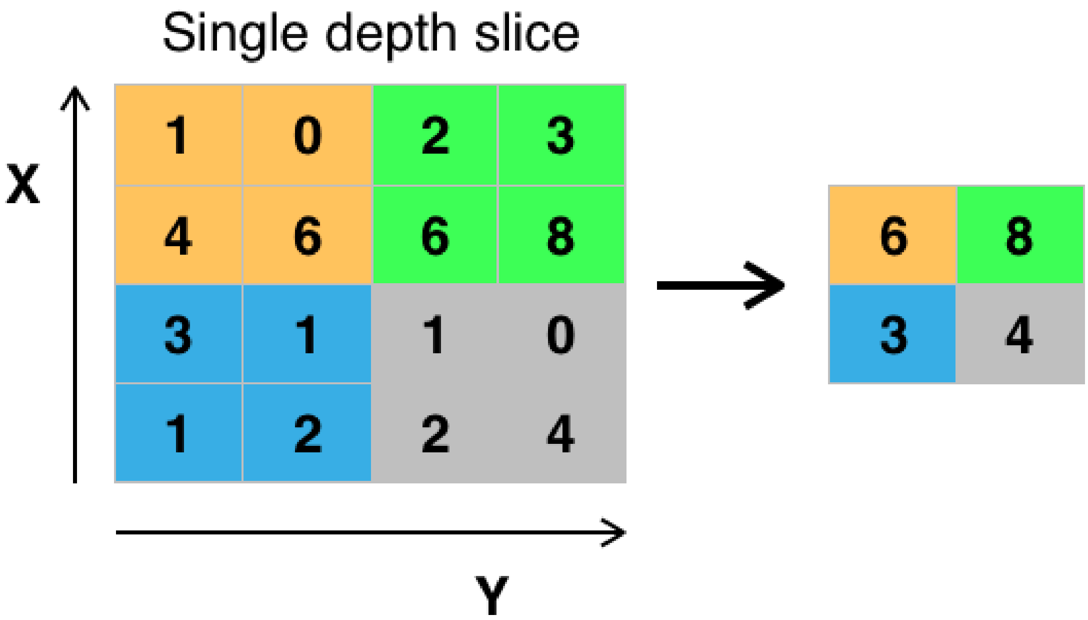

Figure 2: working of pooling layer

Pooling Layer : Pooling Layers Basically shrinks the matrix into smaller matrix with similar parameters.Max pooling is method where most important element of a particular part of the matrix is taken into consideration , all other smaller values are ignored.

2 Performance Analysis

- **Symbols and their respective description:**

Symbol	**Description**
$f_h(x)$	Hierarchical feature
$\varnothing$	Activation function
W_i^c	Projection of ith layer
b_i^c	Bias
L	Feature Transformation
M	Distance Metric
x	Tranning example
y	Labels of x

System model image:

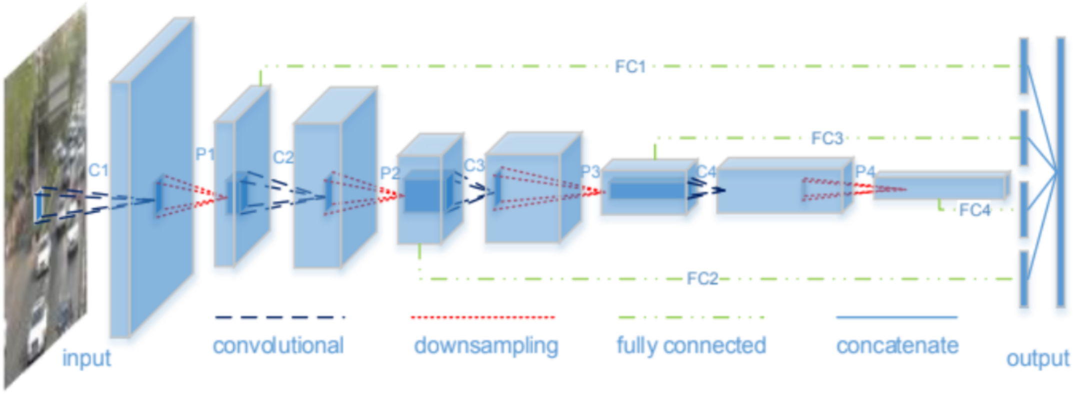

Figure 3: Transmitter Relay

- Analytical expression derivation:

  $f_1(x) = \varnothing(w_1^c x + b_1^c)$

  i.e. output of first pooling layer

  $\rightarrow$ For deep extraction we use $f_1(x)$

  $\therefore f_2(f_1(x)) = \varnothing(w_2^c * f_1(x) + b_2^c)$ $\quad w_2^c * f_1(x)$ is convolution layer output

  $\rightarrow$ After applying $\varnothing$ we get pooling layer output.

  Now concatenating all four features :

  $f_h(x) = f_1(x)||f_2(x)||f_3(x)||f_4(x)$

  $Mahalanobis Distance$:

  $d(x, x_i) = (f_h(x_i) - f_h(x))^T M(f_h(x_i) - f_h(x))$

  $= (f_h(x_i) - f_h(x))^T L^T L(f_h(x_i) - f_h(x))$

  $d(x, x_i) = ||Lf_h(x_i) - f_h(x)||^2$

  Now,

  In definition of Gaussian distance

  $P(X/W_i) = \frac{1}{(2\pi)^{\sigma/2}|\Sigma_i|^{1/2}} exp\{\frac{-1}{2}(x - u_i)^T \Sigma_i^{-1}(x - u_i)\}$

  $k(x.x_i) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{d(x,x_i)}{\sigma})$ $\quad$ i.e. weight based on distance of x and x_i

  Loss Function

  $L = \Sigma\phi_{hub}(y_i, \hat{y}_i)$

  $$\phi_{hub}(y_i, \hat{y}_i = \begin{cases} (y_i - \hat{y}_i)^2 & \text{if}|y_i - \hat{y}_1| \leq \tau \\ \tau(2|y_i - \hat{y}_i| - \tau) & \text{if}|y_i - \hat{y}_1| > \tau \end{cases} \quad (1)$$

  $L = \Sigma\phi_{hub}(y_i, \hat{y}_i) + (\frac{\alpha}{2}\Sigma_w W^2)$

3 Performance Analysis of New contributions

- System Model
- CNN-LSTM Model

  In our new contribution, we tried to merge and implement LSTM Model in our CNN system Model. CNNs are used in modeling problems related to spatial inputs like images.

  LSTMs are used in modelling tasks related to sequences and do predictions based on it. Standard LSTM cant be used directly on sequences where input is spatial. So to perform tasks which need sequences of images to predict something we need more sophisticated model. That's where CNN-LSTM model comes in. Also the accuracy of this model is more comapared to CNN system model.

  The CNN Long Short-Term Memory Network (CNN-LSTM) is an LSTM architecture specifically designed for sequence prediction problems with spatial inputs, like images or videos. The above picture describes how a general CNN-LSTM model works[4]. CNN-LSTMs are generally used when their inputs have spatial structure such as the 3D structure or pixels in an image or 1D structure of words in a sentence, paragraph, or document and also have a temporal structure in their input such as the order of images in a video or words in text, or require the generation of output with temporal structure such as words in a textual description.

  Steps for our Model -

  1) We can define a 2D convolutional network as comprised of Conv2D and MaxPooling2D layers ordered into a stack of the required depth.
  2) Conceptually, there is a single CNN model and a sequence of LSTM models, one for each time step. We want to apply the CNN model to each input image and pass on the output of each input image to the LSTM as a single time step.
  3) So we have 3 pairs of 2D convolutional network comprised of Conv2D and MaxPooling2D layers and after every depth we define a LSTM model.

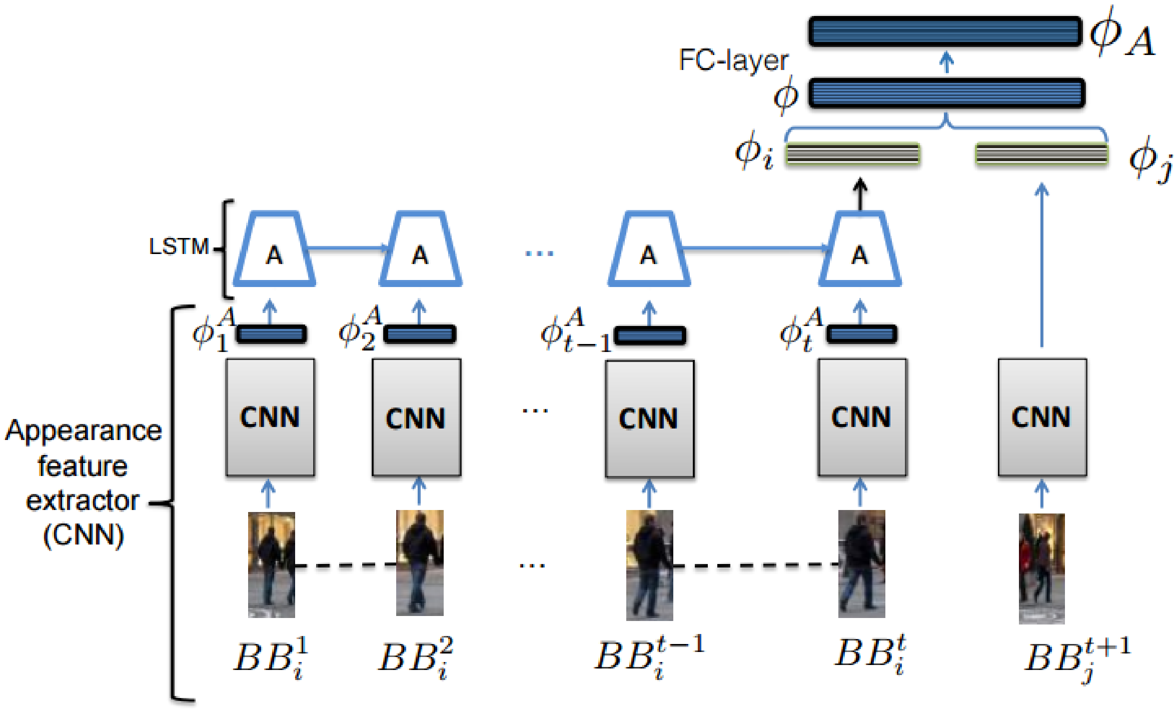

Figure 4: System Model.

Model :

model = Sequential()

define CNN model

model.add(TimeDistributed(Conv2D(...))

model.add(TimeDistributed(MaxPooling2D(...)))

model.add(TimeDistributed(Flatten()))

define LSTM model

model.add(LSTM(...))

model.add(Dense(...))

4 Numerical Results

4.1 Simulation Framework

Traffic Video Database:-

This traffic video database contains only one scene(same camera angle). It is used to evaluate the performance of different methods under simple circumstance. The database contains 254 highway video clips in which the resolution is 320 240. There are different light conditions and weathers in the videos. However, all videos are recorded by the same camera and angle. It is a populardataset used for congestion classification which is divided into 3 classes. We have tested our proposed model using this dataset and we are achieving 62% -65%.

We are keeping learning rate = 0.0005 and tested dataset with 4 epoches. This data-set contains 254 samples each containing frames of different videos having traffic in low to high range. We have 4 sets for training our model and 4 sets for testing our model.

We have also tested this model by capturing traffic videos at Panchvati cross road and we have converted that into frames and produced figures.

Reproduced Figures:

Figure 5: Contains of comparision between article,s results and figures produced by our model.

Figure 6,7,8 : Results of Traffic Database.

Figure 9,10,11,12 : Results of Our Dataset.

4.2 Reproduced Figures

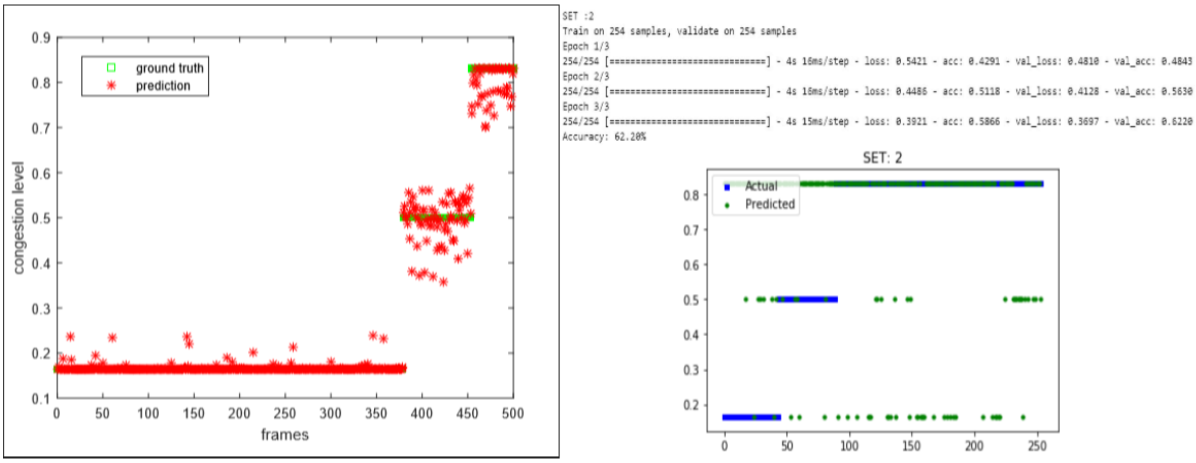

Figure 5:

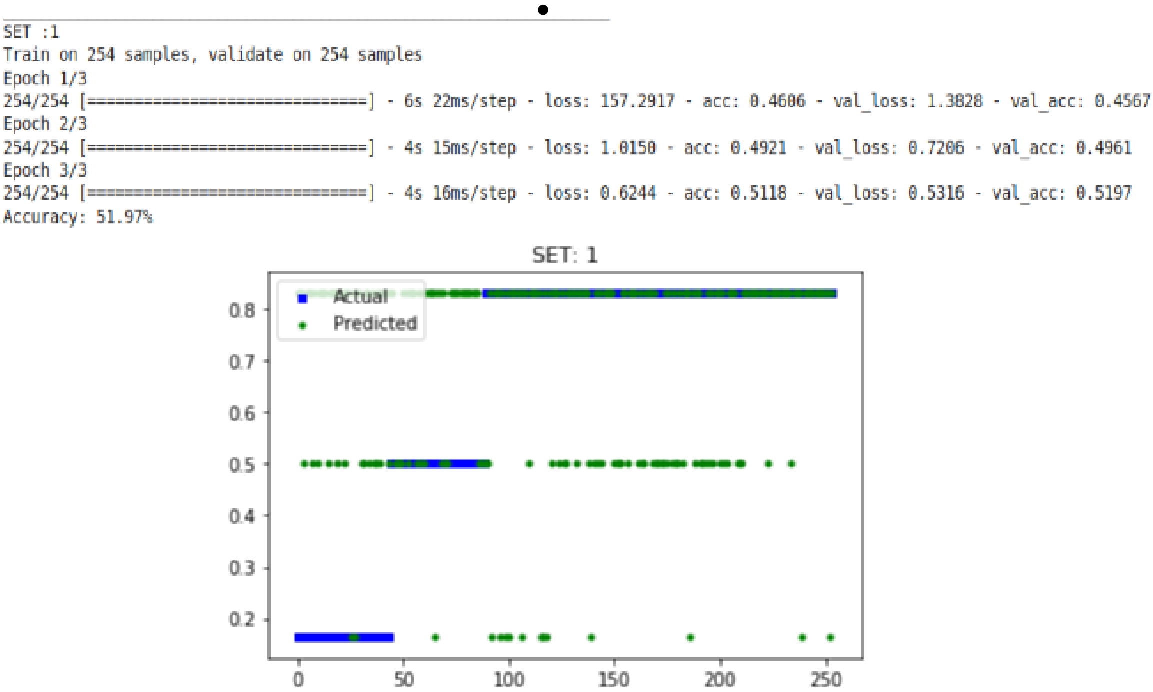

Figure 6:

```
SET :3
Train on 254 samples, validate on 254 samples
Epoch 1/3
254/254 [==============================] - 4s 16ms/step - loss: 0.3568 - acc: 0.6220 - val_loss: 0.3425 - val_acc: 0.6299
Epoch 2/3
254/254 [==============================] - 4s 16ms/step - loss: 0.3337 - acc: 0.6260 - val_loss: 0.3244 - val_acc: 0.6220
Epoch 3/3
254/254 [==============================] - 4s 16ms/step - loss: 0.3188 - acc: 0.6260 - val_loss: 0.3124 - val_acc: 0.6378
Accuracy: 63.78%
```

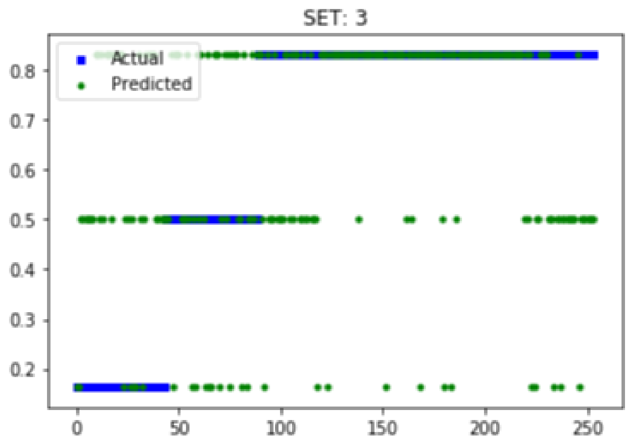

Figure 7:

```
SET :4
Train on 254 samples, validate on 254 samples
Epoch 1/3
254/254 [==============================] - 4s 17ms/step - loss: 0.3082 - acc: 0.6457 - val_loss: 0.3034 - val_acc: 0.6457
Epoch 2/3
254/254 [==============================] - 4s 16ms/step - loss: 0.3002 - acc: 0.6417 - val_loss: 0.2964 - val_acc: 0.6457
Epoch 3/3
254/254 [==============================] - 4s 16ms/step - loss: 0.2937 - acc: 0.6378 - val_loss: 0.2908 - val_acc: 0.6417
Accuracy: 64.17%
```

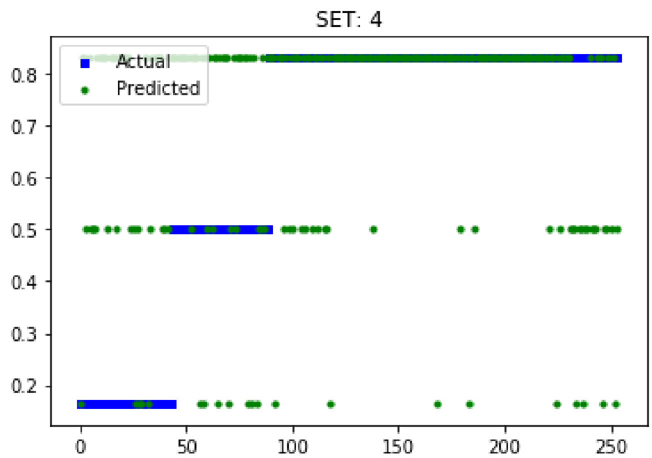

Figure 8:

4.3 New Results

```
SET :1
Train on 254 samples, validate on 254 samples
Epoch 1/3
254/254 [==============================] - 5s 21ms/step - loss: 186.8727 - acc: 0.6024 - val_loss: 0.8477 - val_acc: 0.5669
Epoch 2/3
254/254 [==============================] - 4s 16ms/step - loss: 0.6426 - acc: 0.5354 - val_loss: 0.4862 - val_acc: 0.5039
Epoch 3/3
254/254 [==============================] - 4s 15ms/step - loss: 0.4375 - acc: 0.5000 - val_loss: 0.3908 - val_acc: 0.5236
Accuracy: 52.36%
```

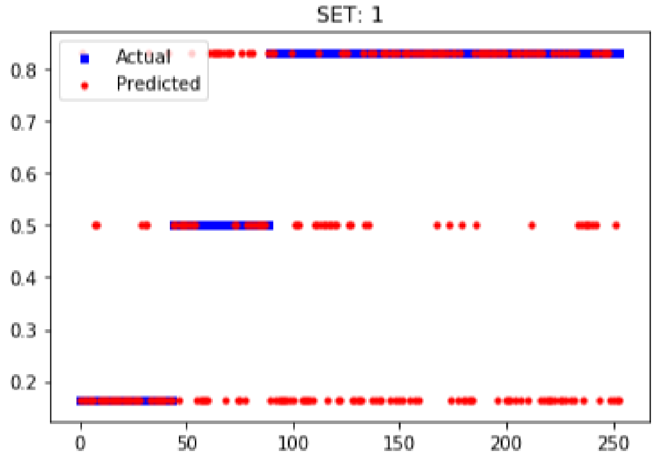

Figure 9:

```
SET :2
Train on 254 samples, validate on 254 samples
Epoch 1/3
254/254 [==============================] - 4s 15ms/step - loss: 0.4507 - acc: 0.4882 - val_loss: 0.4103 - val_acc: 0.5197
Epoch 2/3
254/254 [==============================] - 4s 15ms/step - loss: 0.3900 - acc: 0.5472 - val_loss: 0.3688 - val_acc: 0.5748
Epoch 3/3
254/254 [==============================] - 4s 15ms/step - loss: 0.3564 - acc: 0.5787 - val_loss: 0.3433 - val_acc: 0.5827
Accuracy: 58.27%
```

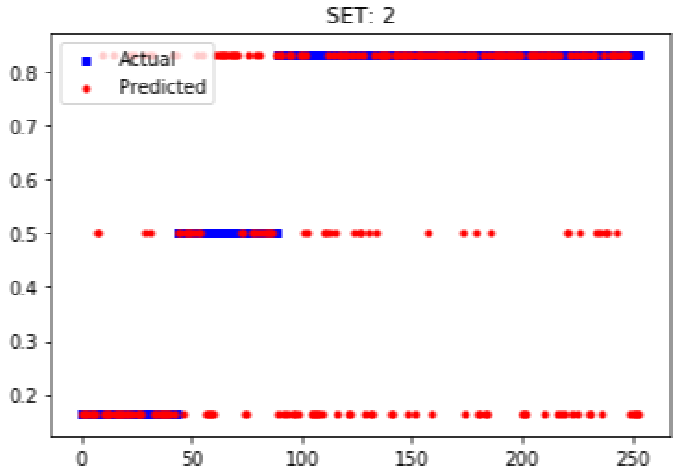

Figure 10:

```
SET :3
Train on 254 samples, validate on 254 samples
Epoch 1/3
254/254 [==============================] - 4s 15ms/step - loss: 0.3351 - acc: 0.5709 - val_loss: 0.3259 - val_acc: 0.5866
Epoch 2/3
254/254 [==============================] - 4s 15ms/step - loss: 0.3203 - acc: 0.5827 - val_loss: 0.3141 - val_acc: 0.6024
Epoch 3/3
254/254 [==============================] - 4s 15ms/step - loss: 0.3099 - acc: 0.6181 - val_loss: 0.3048 - val_acc: 0.6220
Accuracy: 62.20%
```

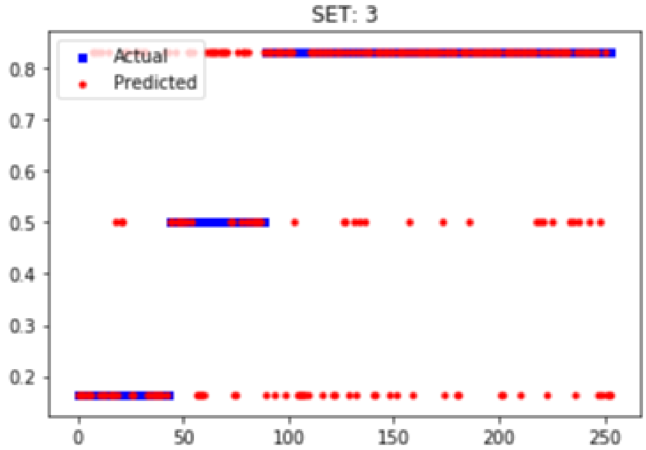

Figure 11:

```
SET :4
Train on 254 samples, validate on 254 samples
Epoch 1/3
254/254 [==============================] - 4s 15ms/step - loss: 0.2811 - acc: 0.6024 - val_loss: 0.2788 - val_acc: 0.6063
Epoch 2/3
254/254 [==============================] - 4s 16ms/step - loss: 0.2771 - acc: 0.6063 - val_loss: 0.2752 - val_acc: 0.6220
Epoch 3/3
254/254 [==============================] - 4s 15ms/step - loss: 0.2740 - acc: 0.6181 - val_loss: 0.2725 - val_acc: 0.6181
Accuracy: 61.81%
```

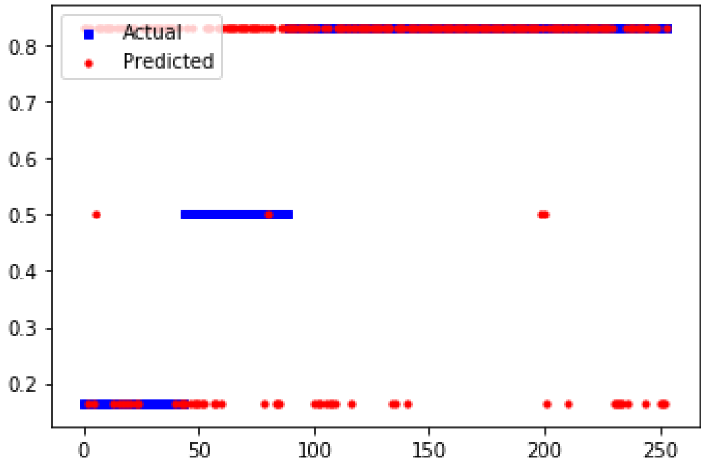

Figure 12:

5 Conclusions

As Deep learning is a subset of machine learning which requires great amount of skill set and hands on experience. We started by learning the basics of machine learning and browsed the parts of deep learning regarding our research paper. Our base article included advanced convolutional neural network, simultaneously we were searching for dataset using which we can execute this network. We found a categorical dataset(trafficdb) on which we implemented our model. On finding the dataset effective generated our results on it and acquired 60% accuracy. By improving our model subsequently we acquired 65% accuracy. We generated our dataset on the streets of Ahmedabad. Tested it and acuired an accuracy of 66%. In future we would like to improve its accuracy using LSTM and make our result more specific using regression.

6 Contribution of team members

6.1 Technical contribution of all team members

Tasks	**Jay Patel**	**Mohit Vaswani**	**Manav Shah**	**Prima Sanghvi**	**Jaydeep Modi**
Task-1	Simulation of code	Simulation of code	Simulation	Theory Concepts	Integration
Task-2	Building CNN Network	CNN Network	CNN Network	Derivation	Derivation
Task-3	Data-set Creation	Data-set management	LSTM	Regression Code	Data-set Creation

6.2 Non-Technical contribution of all team members

Tasks	**Jay Patel**	**Mohit Vaswani**	**Manav Shah**	**Prima Sanghvi**	**Jaydeep Modi**
Task-1	Creating our dataset	Literature Survey	Graph plotting	Report making	Report making
Task-2	Video to frames	Report Content	Literature Survey	Labeling of dataset	creating our dataset
Task-3	Labeling of dataset	Plotting Graphs	Importing	Report content	Labeling of dataset

References

[1] The best explanation of Convolutional Neural Networks on the Internet! *Retrieved 20 November 2019,https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8*

[2] Prabhu, R. (2018). *Understanding of Convolutional Neural Network (CNN) — Deep Learning. Retrieved 20 November 2019, from https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148*

[3] Convolutional Neural Networks (CNN, or ConvNets) (2017). *Retrieved 16 October 2019, from https://medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207*

[4] Brownlee, J. (2019). *CNN Long Short-Term Memory Networks. https://machinelearningmastery.com/cnn-long-short-term-memory-networks/*

[5] Thompson, D. (2014). Jet propulsion laboratory, California institute of technology. *https://www.youtube.com/watch?v=M0EjrFQH49o*

[6] Bouvrie, J. (2004). Hierarchical Learning: Theory with Applications in Speech and Vision. *http://web.mit.edu/jvb/www/papers/bouvrie*$_t$*hesis*$_2$009*.pdf*

[7] Verma, N. (2017). Metric Learning with some recent advances. *http://cseweb.ucsd.edu/ naverma/talks/metric*$_l$*earning*$_t$*utorial*$_v$*erma.pdf*