innovate achieve lead

BITS Pilani
K K Birla Goa Campus

Self-Driving car using Deep learning

Kanika Gera (2018H1030041G)

Satyajeet Kumar (2018H1030050G)

Kunwar Vikrant (2018H1030067G)

Motivation

- Automatic navigation is an open challenging issue in the real-life traffic situation
- The motivation behind this is to improve car safety and efficiency.
- Motivation is to help prevent traffic accidents, free up people's time and reduce carbon emissions.

Problem Definition

innovate achieve lead

- Problem is to implement self driving car‘s steering wheel component with front dashcam video as a sensory input .
- We shall implement the software component of deciding the angle of rotation of the steering wheel according to the curvature of the road, and showcase our work visually.
- We shall make use of various deep learning methods used to solve similar problems and compare their performances in this domain. Our objective includes reducing log loss on the training data.

Action Plan

innovate achieve lead

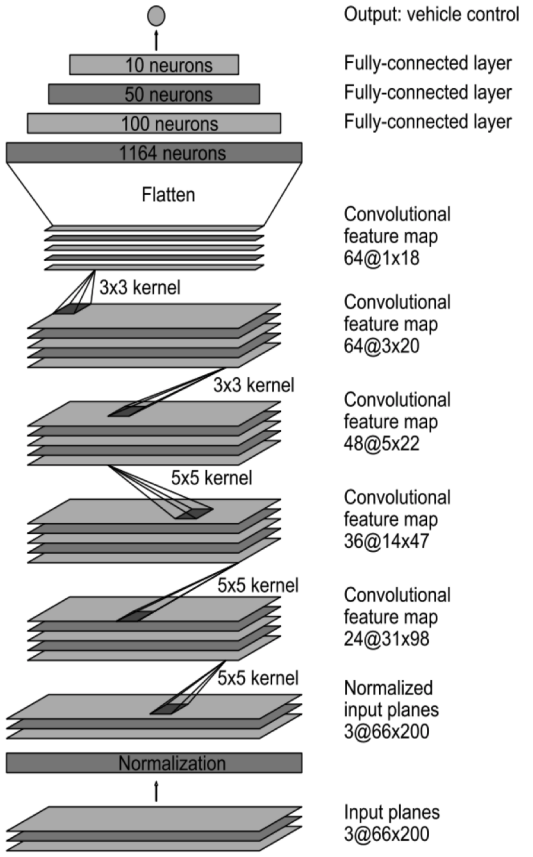
Approach 1

- Develop an end to end Convolution Neural Network to implement steering wheel rotation in a self driving car.
- In addition to the architecture,we will perform hyper-parameter tuning like adding dropout layers, changing optimizer etc.

Corresponding Literature Survey

Title: *End to End Learning for Self-Driving Cars*
NVIDIA CORP.

Architecture:



Action Plan

Approach 2

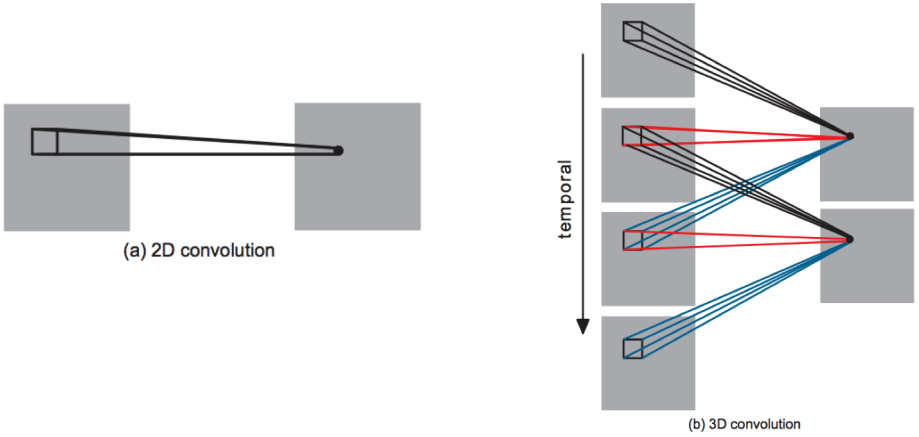
- We shall improve model implemented using approach 1, by considering the sequential information that video sequence provides us.
- We will make use of temporal-spatial information to improve self-driving car model with the help of 3D Convolution network.
- Study shows that 3D Convolution has outstanding capabilities for the extraction of spatial features.

Corresponding Literature Survey

Title: *3D Convolutional Neural Networks for Human Action Recognition*

Summary:

- In 2D Conv feature maps used are computed from the spatial dimensions only.
- It is desirable to capture the motion information encoded in multiple contiguous frames.
- The 3D convolution is achieved by convolving a 3D kernel to the cube formed by stacking multiple contiguous frames together.



Action Plan

Approach 3

Conv-LSTM can make use of timing relationships with the help of LSTM component and can characterize local spatial features with the help of CNN. Similar idea will be used for our problem statement.

Corresponding Literature Survey

Title: *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*

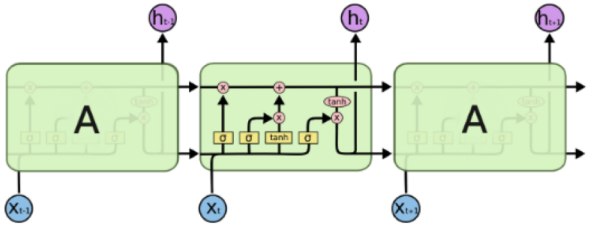


Figure 1 - Repeating Module in LSTM

The ConvLSTM , the is replaced by '*' i.e. Convolution Operator and is replaced by 'o' i.e. Hadamard Operator

Title: *End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners*

Baidu has developed an automated driving training model based on Conv-LSTM network structure only. It has used front cameras along with TomTom route planner

PHASE 2

Objective

- The objective was to develop an end to end Convolution Neural Network model to implement steering wheel rotation in a self driving car.

- In addition to the architecture, we have performed hyper-parameter tuning like adding dropout layers, changing optimizer etc.

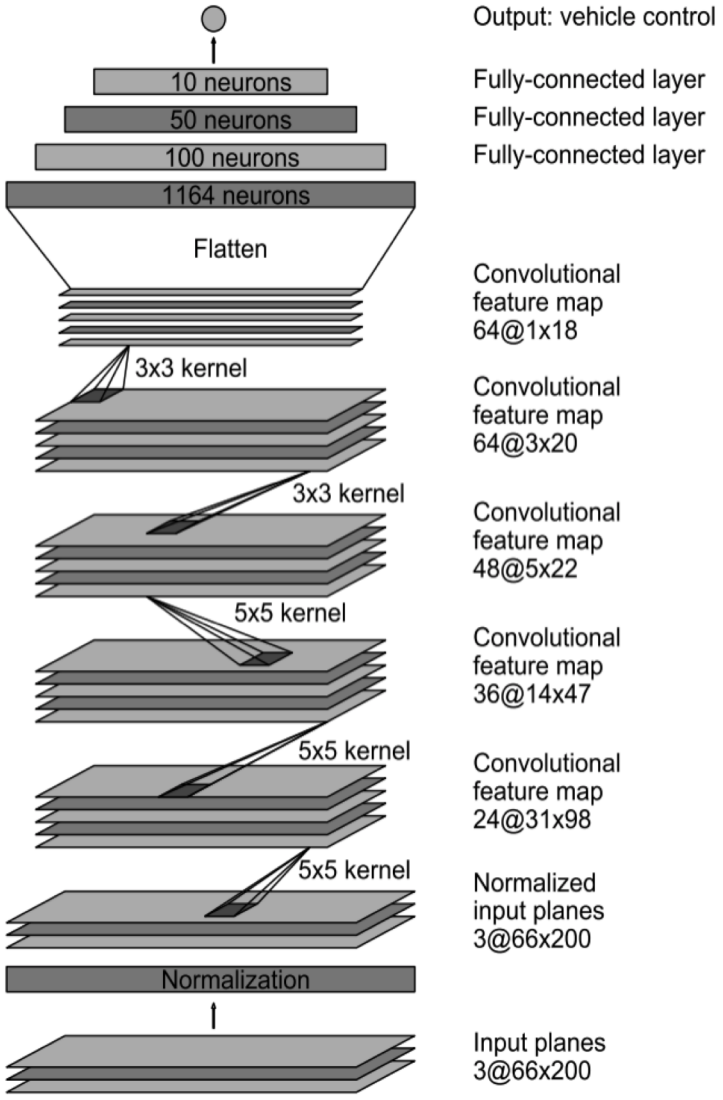
Data Preprocessing

- Load Images
- Normalize pixels
- Train-Test Split
- Split Training Data into batch size of 100 and number of epochs as 30.

45378	45378.jpg	1.51
45379	45379.jpg	1.51
45380	45380.jpg	1.51
45381	45381.jpg	1.61
45382	45382.jpg	1.61
45383	45383.jpg	1.61
45384	45384.jpg	1.51
45385	45385.jpg	1.51
45386	45386.jpg	1.31
45387	45387.jpg	1.21
45388	45388.jpg	1.21

Model Architecture

- Adding untied bias on each CNN Layer.
- Adding Dropout Layer after each fully connected layer, with dropout rate =0.5.
- Tried various Adam Optimizer Values.



Results

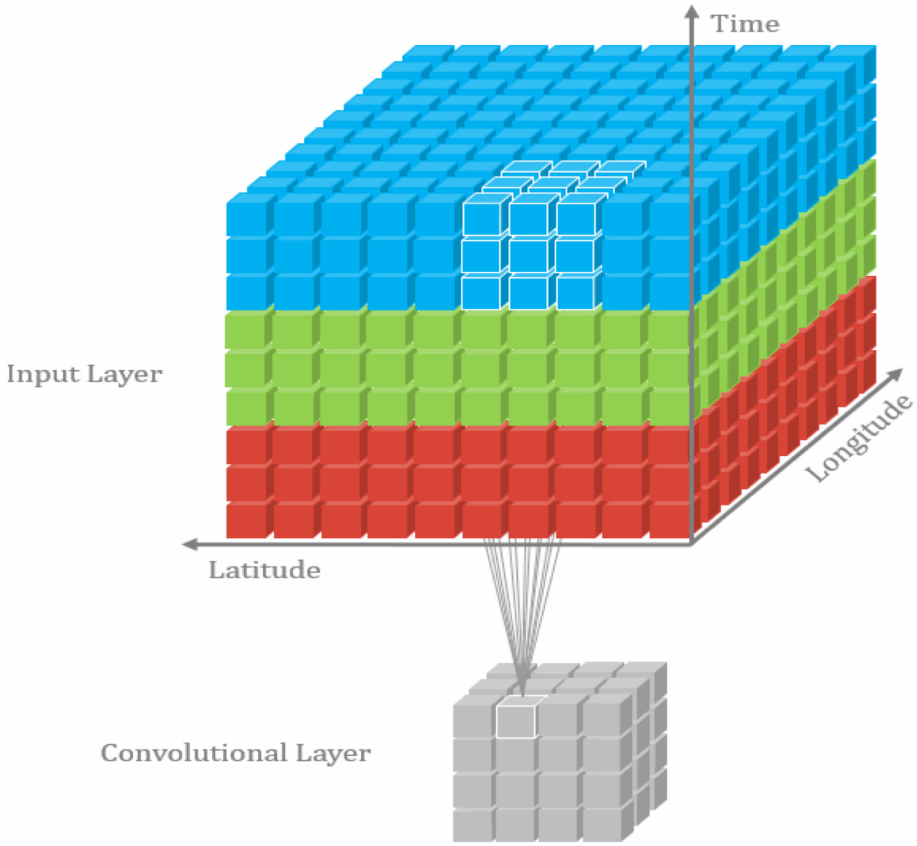
- With the changes incorporated into the architecture, we also managed to achieve a loss of 0.141.

```
Epoch: 29, Step: 3260, Loss: 0.155983
Epoch: 29, Step: 3270, Loss: 0.155054
Epoch: 29, Step: 3280, Loss: 0.159188
Epoch: 29, Step: 3290, Loss: 4.32049
Epoch: 29, Step: 3300, Loss: 1.17218
WARNING:tensorflow:*********************************************************
WARNING:tensorflow:TensorFlow's V1 checkpoint format has been deprecated.
WARNING:tensorflow:Consider switching to the more efficient V2 format:
WARNING:tensorflow:   `tf.train.Saver(write_version=tf.train.SaverDef.V2)`
WARNING:tensorflow:now on by default.
WARNING:tensorflow:*********************************************************
Epoch: 29, Step: 3310, Loss: 0.149182
Epoch: 29, Step: 3320, Loss: 0.147925
Epoch: 29, Step: 3330, Loss: 0.148542
Epoch: 29, Step: 3340, Loss: 0.149045
Epoch: 29, Step: 3350, Loss: 0.141557
Model saved in file: ./save/model.ckpt
```

PHASE 3

Objective

- Objective of this phase is to develop an 3D Convolution Neural Network that can captures temporal-spatial information to implement steering wheel rotation in a self driving car.



Data Preprocessing

- Images are resized to 66 x 200 .
- MinMaxScaler technique is used to normalize the pixel values of the images.

- 3D Convolution has one extra dimension , hence images are reshaped according to [Batch_Size,Time, Width, Height, Channels], we took 100 images for time dimension i.e. 100 images of 66X200X3 are sent together in the network.

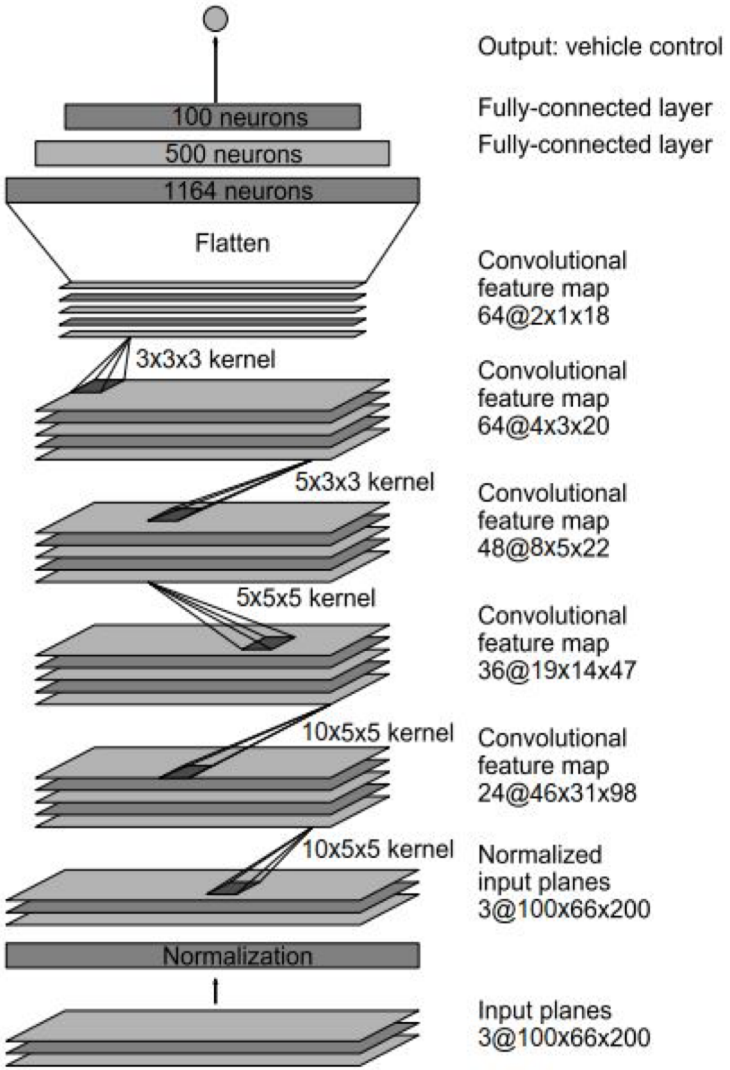
45378	45378.jpg	1.51
45379	45379.jpg	1.51
45380	45380.jpg	1.51
45381	45381.jpg	1.61
45382	45382.jpg	1.61
45383	45383.jpg	1.61
45384	45384.jpg	1.51
45385	45385.jpg	1.51
45386	45386.jpg	1.31
45387	45387.jpg	1.21
45388	45388.jpg	1.21

Model Architecture

- **Input Layer**
  3D Convolution has one extra dimension , hence images are reshaped according to **[Batch_Size,Time, Width, Height, Channels]**

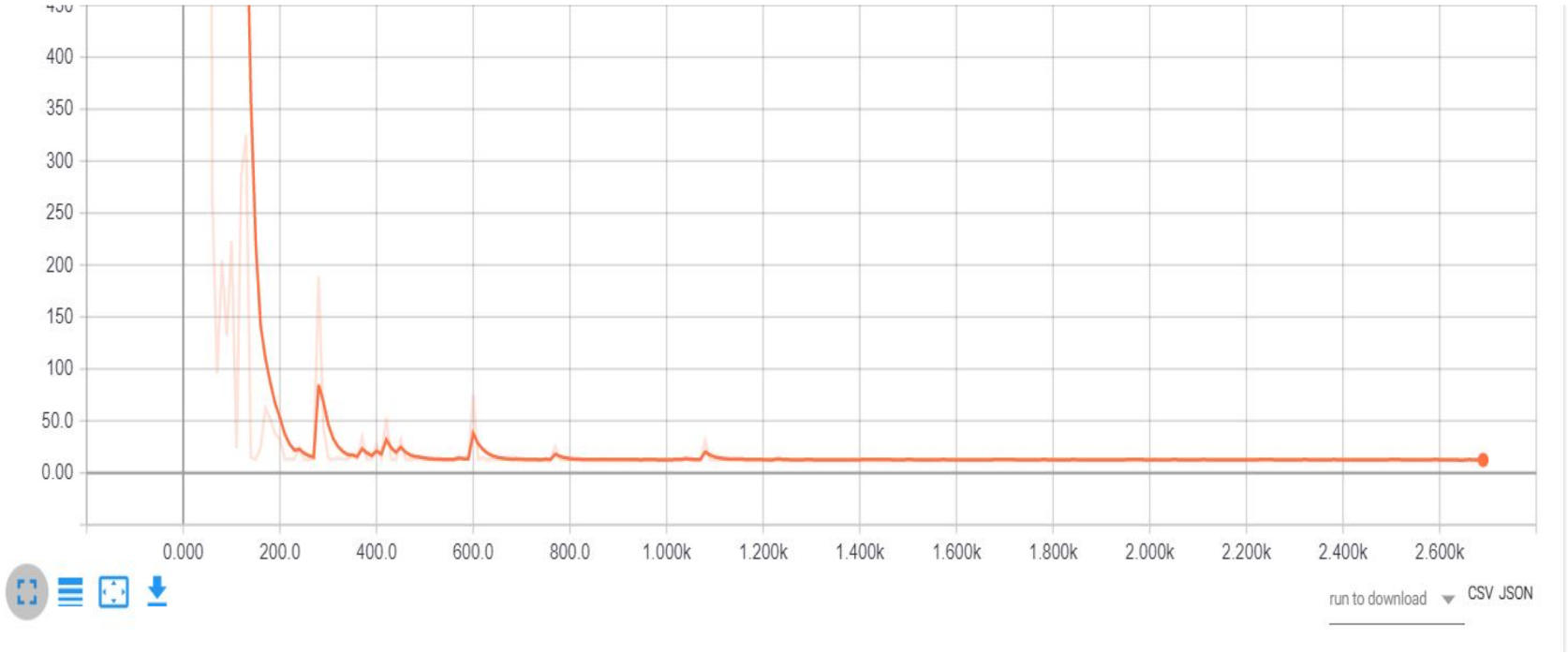
- **Convolution Layers**
  The kernel in 3D Convolution is of shape **[T_kernel, W_kernel, H_kernel, in_channels, out_channels],** where T_kernel is time dimension shape of the kernel.

- **Fully Connected Layers**
  Fully Connected Layer of 1164 neurons is used first. Last convolution layer was of shape [batch_size,2,1,18,64]. Hence 2304 X 1164 trainable parameters exist during flattening.



Results

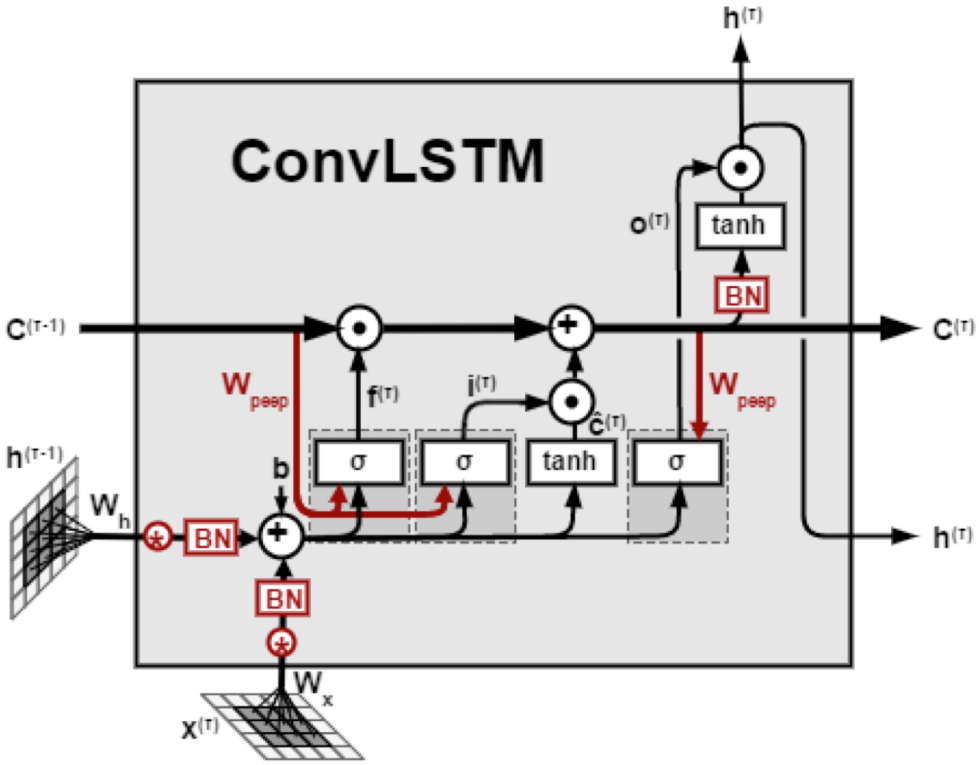
- With the changes incorporated into the architecture, loss of 12.07 was observed.



PHASE 4

Objective

- Objective of this phase is to develop an end to end Conv2D-LSTM model that can capture temporal-spatial information to implement steering wheel rotation in a self driving car.



Data Preprocessing

- Images are resized to 66 x 200 .
- MinMaxScaler technique is used to normalize the pixel values of the images.

- Conv2D-LSTM has one extra dimension , hence images are reshaped according to [Batch_Size,Time,Width,Height,Channels], we took 100 images for time dimension i.e. 100 images of 66X200X3 are sent together in the network.

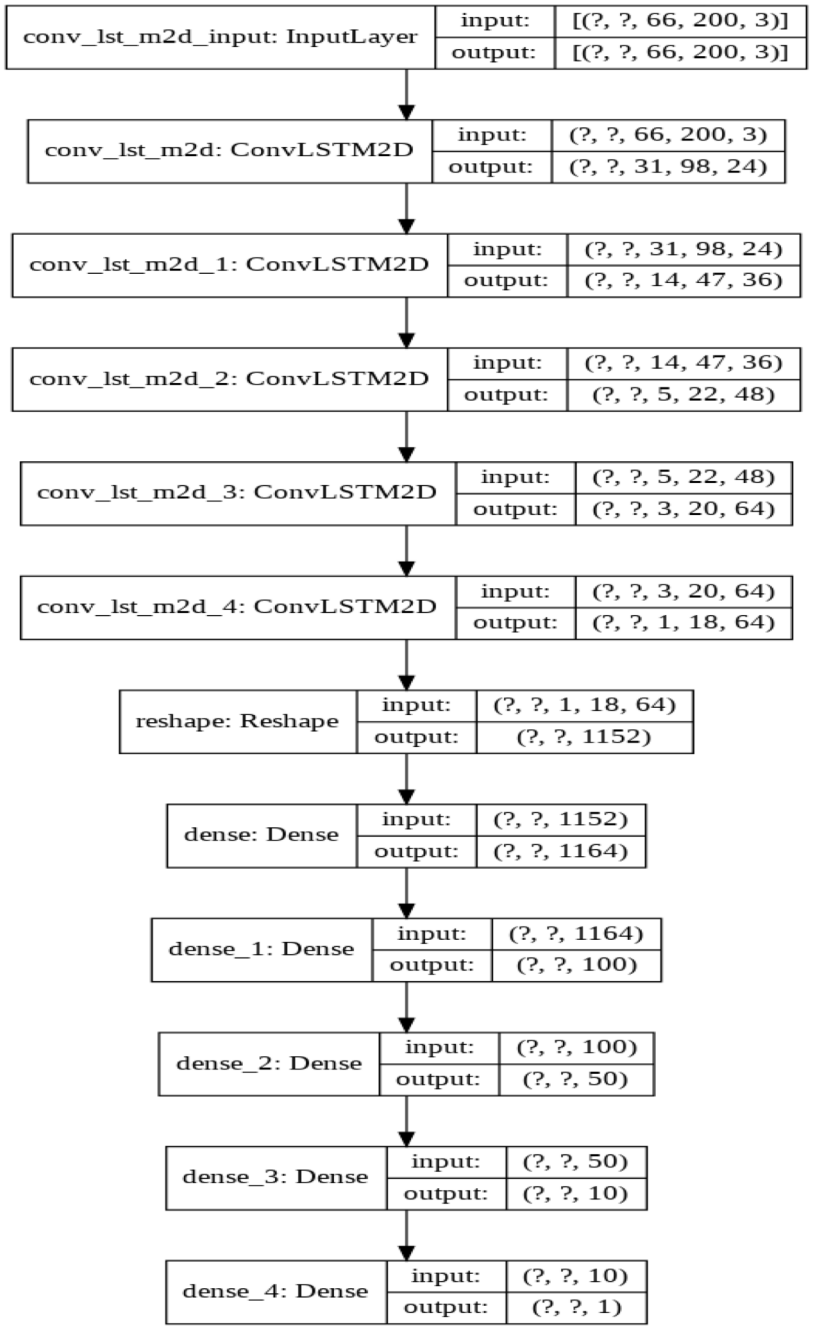
45378	45378.jpg	1.51
45379	45379.jpg	1.51
45380	45380.jpg	1.51
45381	45381.jpg	1.61
45382	45382.jpg	1.61
45383	45383.jpg	1.61
45384	45384.jpg	1.51
45385	45385.jpg	1.51
45386	45386.jpg	1.31
45387	45387.jpg	1.21
45388	45388.jpg	1.21

Model Architecture

- ## Input Layer
  3D Convolution has one extra dimension , hence images are reshaped according to **[Batch_Size,Time, Width, Height, Channels]**

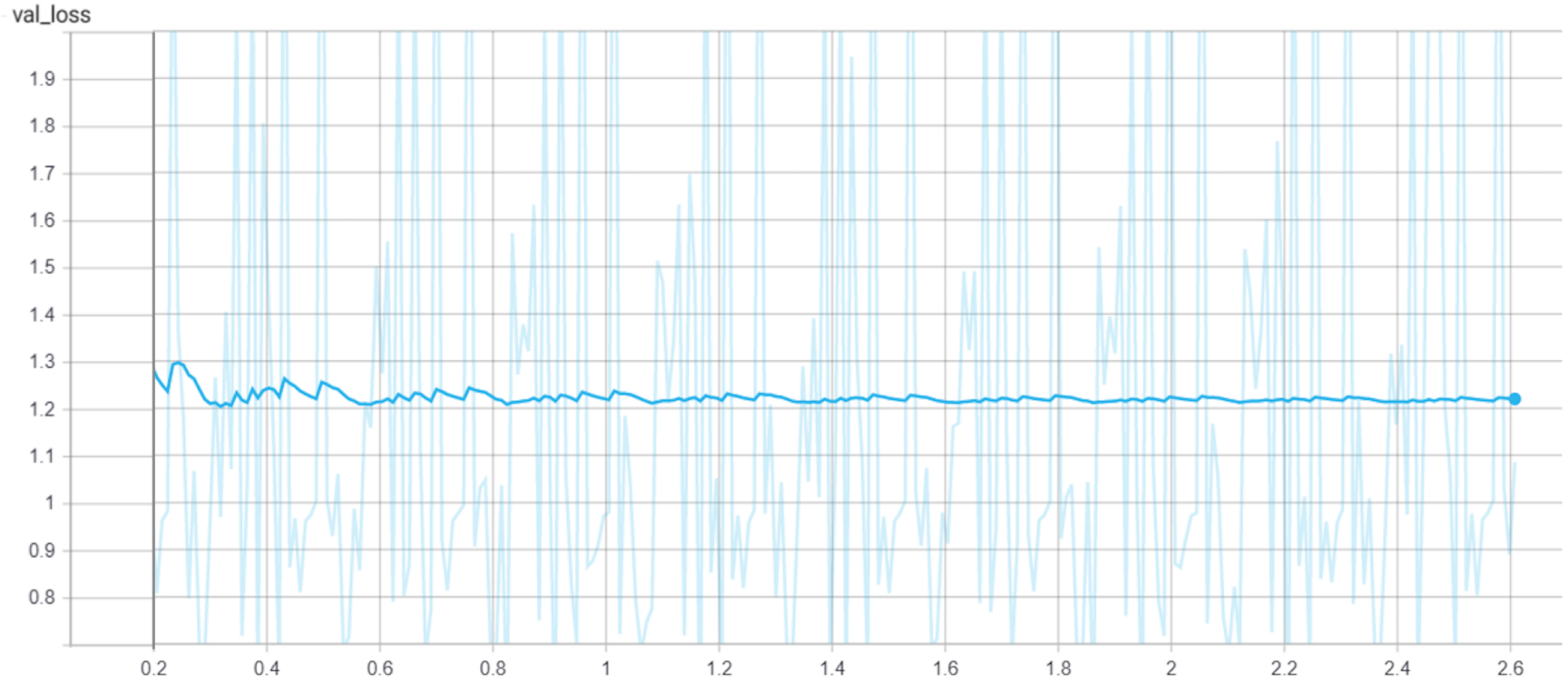
- ## Convolution-LSTM Layers
  The kernel used here are of 2D Convolution is of shape **[W_kernel, H_kernel, in_channels, out_channels],** we have used convolution layers used in phase 2 , as internal functions inside LSTM layer.

- ## Fully Connected Layers
  Fully Connected Layer of 1164 neurons is used first. Last convolution layer was of shape [batch_size,100,1,18,64]. Hence (1X18X64)1152 X 1164 trainable parameters exist during flattening.



Results

- With the changes incorporated into the architecture, loss of 1.2 was observed.

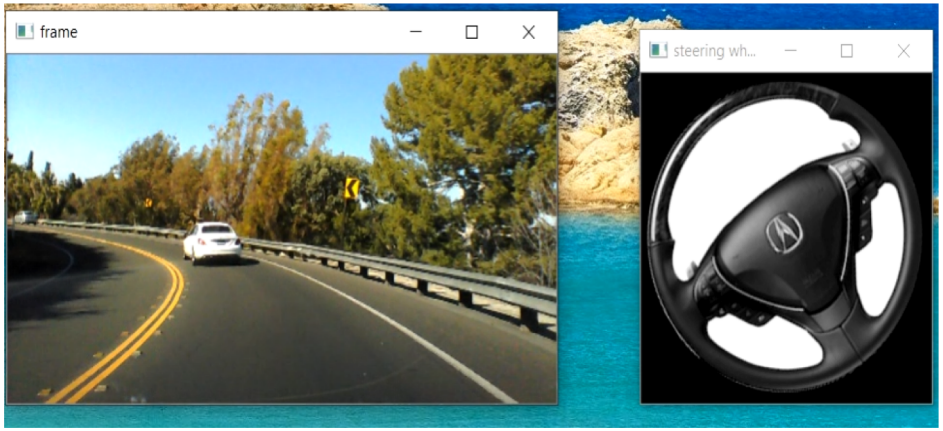


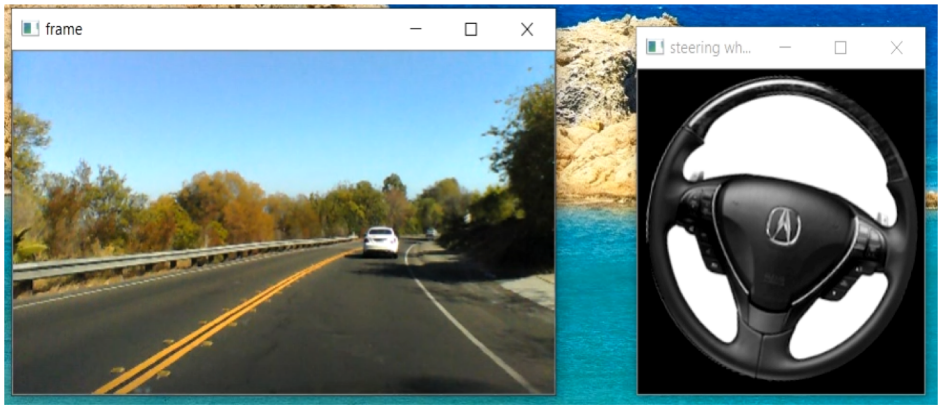
Model comparison

Model	Loss (Mean Squared Error)
Conv2D	0.14
Conv3D	12.07
Conv-LSTM	1.2

Visualization

We ran the model on images of road, and visualized the rotation of the steering wheel according to the input, using openCV library .





Conclusion

We tried three different models to build our automation system and we observed the following points

- The initial Conv2D model with hyperparameter tuning gave the best results among all of the models.
- The second model we tried was Conv3D, this model performed significantly worse than the first and the last model. Also,the training time taken by the 3D-CNNs was significantly larger as compared to the other two models.
- The last model we tried was Conv-LSTM model which did not perform as good as the first model but still performed much better than the Conv3d model.

Certain inferences can be drawn from our observation. The most important one being that using sequence information among images did not yield a better result so more experimentation can be done in the areas of the end-to-end learning model we started with to achieve even better results.

Thank you