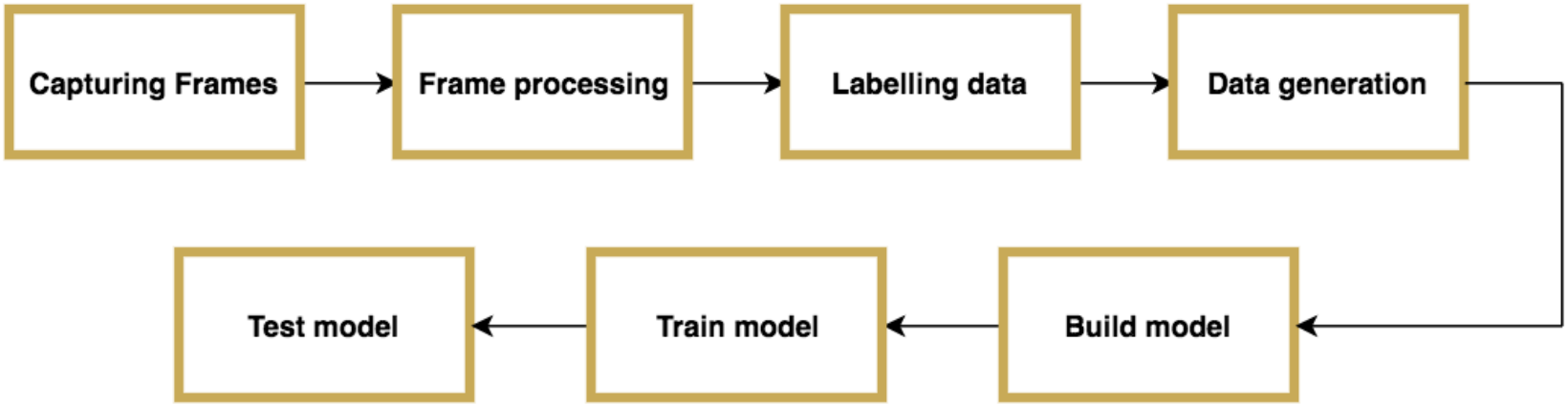
SIMULATION OF LANE SWITCHING IN SELF-DRIVING AUTOMOBILES

INTRODUCTION

- Why self-driving automobiles?
- How simple is autonomous driving?

- In this project we:
  - build and train a CNN model that can learn to drive a car in a very diverse set of virtual environment
  - use a computer video game to train and test our model.
  - train the CNN using the data that is generated by manual driving in a computer video game

PROCESS

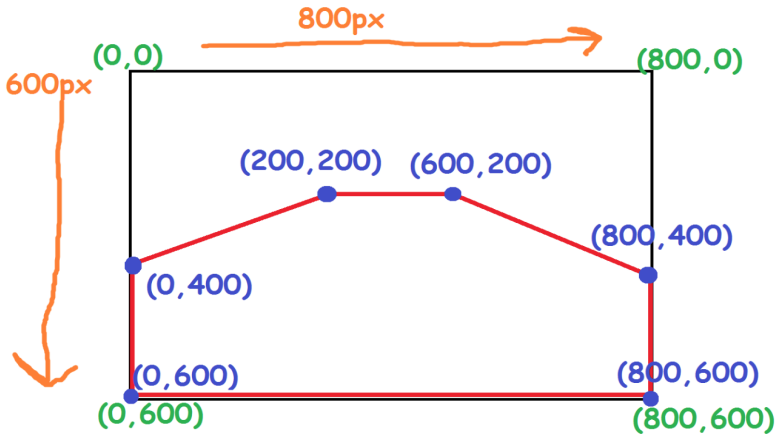

Capturing Frames
Frame processing
Labelling data
Data generation
Build model
Train model
Test model

CAPTURING FRAMES

- Frames form the bulk of our dataset required to train our neural network
- ImageGrab.grab() function in PIL module
- Conversion to grayscale using OpenCV module

FRAME PROCESSING - REGION OF INTEREST

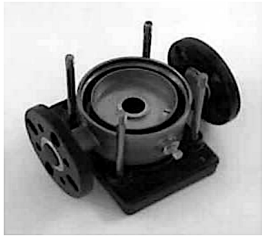
- Only a certain region of image is required to be processed and stored as a data set.
- ROI is obtained using Numpy Indexing.

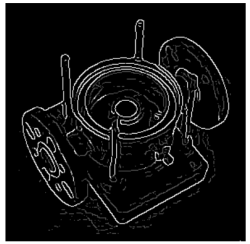
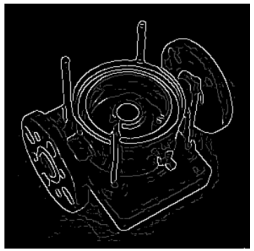
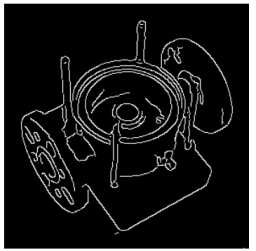


FRAME PROCESSING - EDGE DETECTION

We use the Canny algorithm which detects edges in 5 steps

- Smoothing
- Finding gradients
- Non-maximal suppression
- Double thresholding
- Edge tracking by hysteresis



FRAME PROCESSING - LINE DETECTION

- The lines are detected from the given image using Houghline transform mechanism
- Represents any shape in mathematical form

- This has 2 parameters associated with it: Minimum line length, Maximum line gap

LABELLING AND DATA GENERATION

LANE DETECTION

- Filters unwanted lines by determining their length
- Calculates slope of each line
- Pairs of lines whose slopes are approximately equal and of opposite sign are labelled as lanes

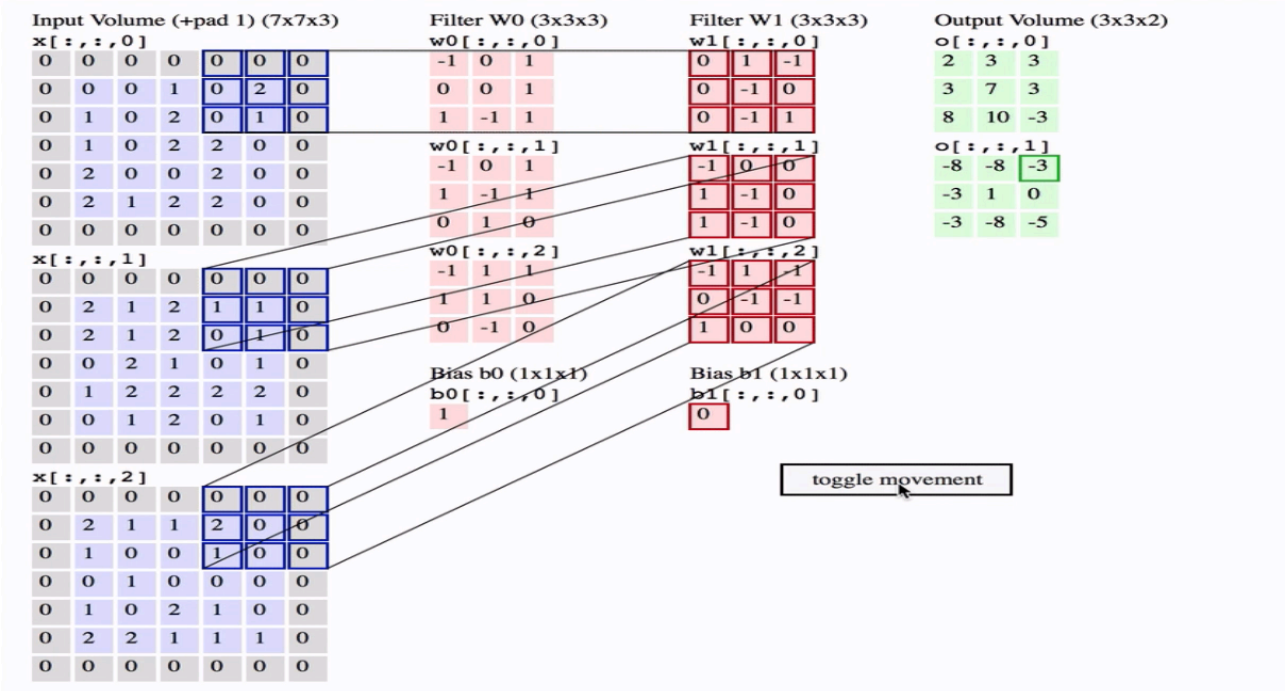
LABELLING

- Dataset is in the form of a numpy array
- Each element is a tuple
- First element is an image
- Second element is an array - consists of the key pressed while that image was captured

Dataset = [{image, keys}, {image, keys},……., {image, keys}]

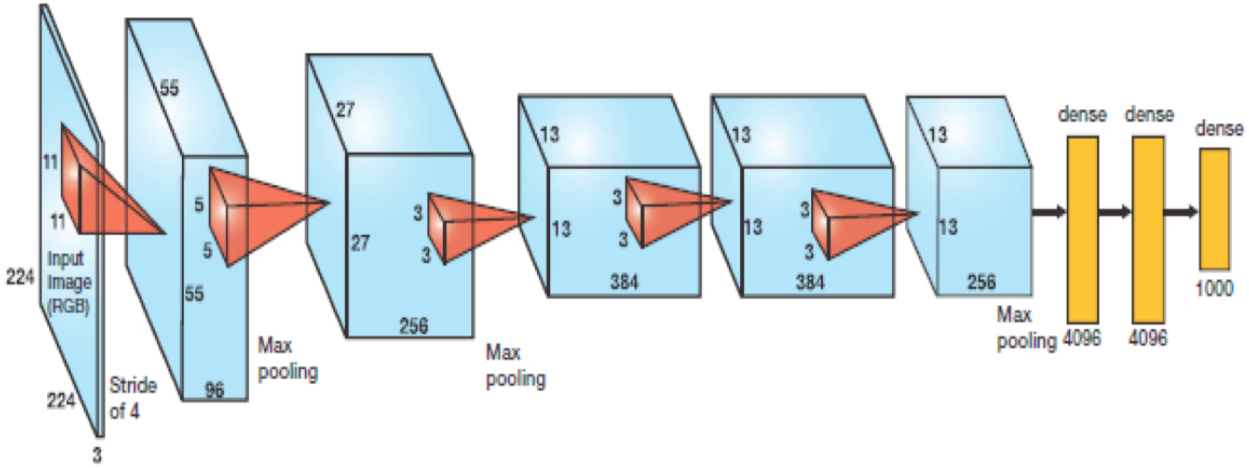
BUILDING THE MODEL

Why CNN?



BUILDING THE MODEL - ALEXNET

- Composed of 5 convolutional layers followed by 3 fully connected layers, each used for feature extraction
- A weight matrix called filter slides over the input image produces a feature map
- The network learns values of these filters on its own during the training process.



TRAINING THE MODEL

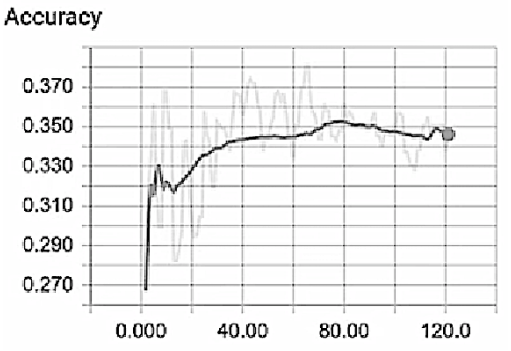
- TensorFlow’s TFlearn module is used for model training
- The input for this model is the dataset that we have generated
- The system is configured to take the best possible decision independently

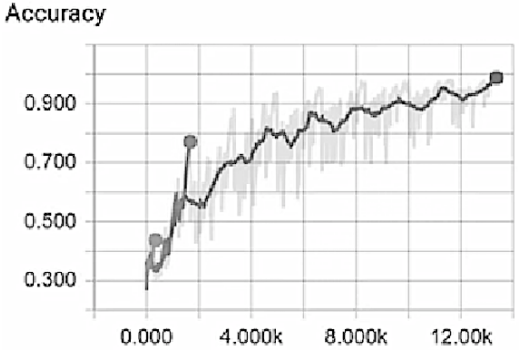
MODEL TESTING

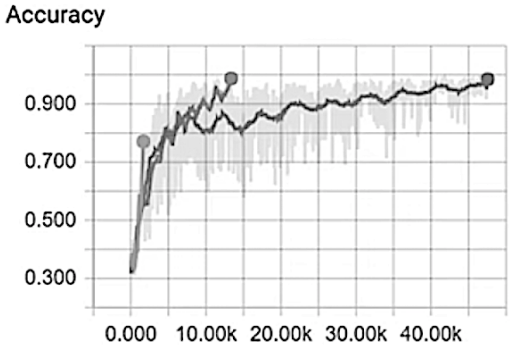
Once the model has been trained the following steps take place:

- Takes screenshots
- Analyses the images
- Takes decisions
- Executes action

MODEL TESTING







The accuracy of the neural model increases with increase in number of frames served as the input

CONCLUSION

Through this paper we have:

- Introduced the concept of lane switching in self driving automobiles
- Utilized AlexNet to create a model that can appropriately predict steering angles
- Demonstrated changes in model accuracy with varying volumes of training data

THANK YOU