SMAI PROJECT FINAL REPORT

VIOLATION DETECTION IN VIDEOS

Swapnil Pakhare - 20161199
Ritvik Sharma - 20161147
Ayush Anand - 20161085
Vikrant Deshmukh - 20161161
IIIT Hyderabad

Contents

0.1 Introduction 3

0.1.1 Aims 3

0.1.2 Motivation : 4

0.1.3 Application of ML : 4

0.2 Referenced Research Papers 4

0.3 Dataset Details 4

0.3.1 Smoking Dataset 4

0.3.2 Obscenery Dataset 5

0.3.3 Drinking dataset 6

0.4 Approach 7

0.4.1 CNN 7

0.4.1.1 Intro : 7

0.4.1.2 Advantages : 8

0.4.1.3 Disadvantages : 8

0.4.2 VGG16 8

0.4.2.1 Intro : 8

0.4.2.2 Advantages : 9

0.4.2.3 Disadvantages : 9

0.4.3 LSTMs 10

0.4.3.1 Intro : 10

0.4.3.2 Advantages : 10

0.4.3.3 Disadvantages : 10

0.4.4 Bi-directional LSTMs 11

0.4.4.1 Intro : 11

0.4.4.2 Advantages : 11

0.4.4.3 Disadvantages : 12

0.5 Implementation 13

0.5.1 CNN 13
0.5.1.1 Architecture : 13
0.5.1.2 Results : 14
0.5.2 LSTM 16
0.5.2.1 Architecture : 16
0.5.2.2 Results : 16
0.5.3 BLSTM 21
0.5.3.1 Architecture : 21
0.5.3.2 Results : 22
0.6 Work Distribution 27

0.1 Introduction

0.1.1 Aims

To detect violation in videos where violation is scenes of smoking, drinking and obscene imagery. Figure 1,2 show some typical examples of input.

Figure 1: Label : Smoking

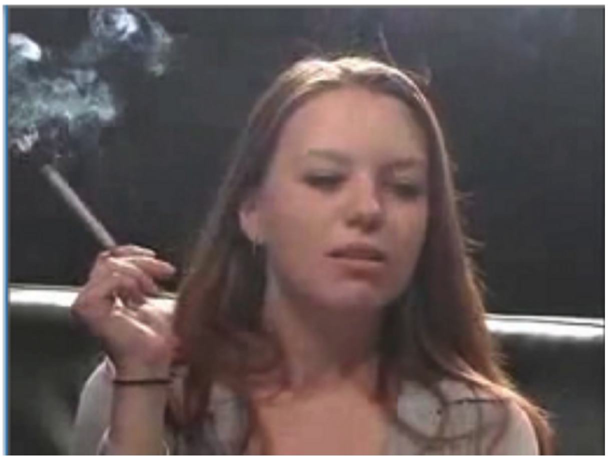
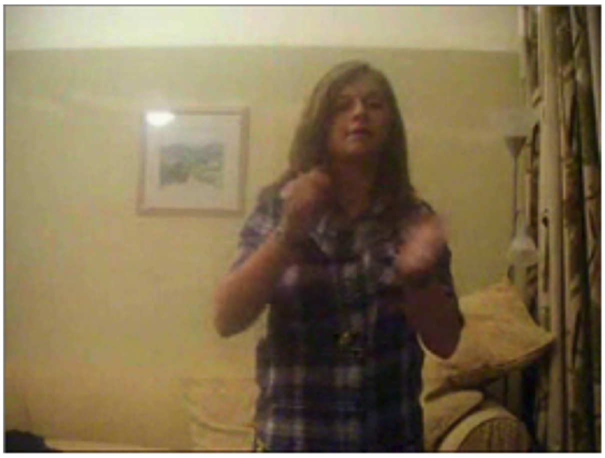


Figure 2: Label : Non-Smoking



0.1.2 Motivation :

Currently, all over the world, people manually scroll through videos to filter out content. We intend to build a classifier that returns time stamps of when violation detection takes place to an accuracy such that if initially the work took about 24 hours of manual time, with the help of our classifier, that time is reduced significantly.

0.1.3 Application of ML :

We can model the problem of detection of violation as a multi-class classification problem with classes being smoking/non-smoking, osbscene/not-obscene, and drinking/not-drinking. Implementing a paper on video classification and using it for our classification will solve our problem.

0.2 Referenced Research Papers

1. http://cs231n.stanford.edu/reports/2017/pdfs/709.pdf

0.3 Dataset Details

0.3.1 Smoking Dataset

- Link
- Number of videos : 1715 non-smoking videos, and 1749 smoking videos (Total 3464 videos)
- Length of each video (Approx.) : 4-5 secs

- Total size of dataset : 1.0 GB
- Summary : Some of the videos were taken from the smoking class of HMDB51 dataset. Other videos were downloaded from Youtube, Reddit etc. and manually annotated. For videos belonging to non-smoking class, videos from the rest 50 classes of the HMDB51 dataset were used.
- Challenges : The main challenge was to build a dataset where smoking is being done in all possible environments, postures, lighting conditions etc.

0.3.2 Obscenery Dataset

- Link
- Number of videos : 122 non-obscene, and 121 obscene (Total 243 videos)
- Length of each video (Approx.) : 8-9 secs
- Total size of dataset :135 MB
- Summary : Some of the videos were taken from the following dataset. More videos were downloaded from Reddit and manually annotated. For videos belonging to non-obscene class, videos from the rest all 51 classes of the HMDB51 dataset were used.
- Challenges : One of the challenges was to find obscene videos on the internet in the first place because mostly such videos are banned on the college LAN network and because Youtube doesn't allow nudity in it's videos. Videos from different angles had to be included so as to make the dataset robust.

0.3.3 Drinking dataset

- Link
- Number of videos : 69 non-drinking, and 75 drinking (Total 147 videos)
- Length of each video (Approx.) : 4-5 secs
- Total size of dataset :42 MB
- Summary : Videos were downloaded from Youtube and manually annotated for the drinking class. For videos belonging to non-obscene class, videos from the rest all 51 classes of the HMDB51 dataset were used.
- Challenges : Main challenge was to build a dataset which distinguishes between drinking a normal fluid such as water, juice etc. and drinking an alcoholic drink such as wine, whiskey, etc.

0.4 Approach

The problem that we have is essentially of video classification. Once the dataset was completed we had a variety of approaches ahead of us. The first objective was to extract frames out of the video data collected. This used the opencv2 package and created a video capture object. We created a frame for every one second and used the images to extract features for our classification.
The first approach we chose was to use a CNN model to classify the images extracted.

0.4.1 CNN

0.4.1.1 Intro :

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The preprocessing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

0.4.1.2 Advantages :

- For a completely new task / problem CNNs are very good feature extractors.
- In terms of performance, CNNs outperform NNs on conventional image recognition tasks and many other tasks.

0.4.1.3 Disadvantages :

- Does not have any scope to take into account time dependency of data and hence doesn't work well on sequential data.
- Large number of parameters and values are required.

0.4.2 VGG16

0.4.2.1 Intro :

VGG16 is used to extract the features present in the images. The model takes in 224X224 sized images. We use the weights extracted from imagenet. This features that we get as an output from VGG16 model is later used in future models for further classification.

0.4.2.2 Advantages :

It makes the improvement over AlexNet by replacing large kernel-sized filters(11 and 5 in the first and second convolutional layer, respectively) with multiple 3X3 kernel-sized filters one after another. With a given receptive field(the effective area size of input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features, and that too at a lower cost.

0.4.2.3 Disadvantages :

- It has so many weight parameters, the models are very heavy, 550 MB + of weight size.
- Which also means long inference time.
- Large width of convolution layer.

Once these features are extracted we have a dual pronged approach using LSTMs and bidirectional LSTMS. We create two classes that have the models stored in them to be trained and predicted.

0.4.3 LSTMs

0.4.3.1 Intro :

Long Short Term Memory networks, usually just called LSTMs, are a special kind of RNN, capable of learning long-term dependencies.

0.4.3.2 Advantages :

- The advantage an LSTM gives over a CNN would be the aspect of connectivity in the data.
- Further, one input to many outputs, many to many, and input flexibility in general is higher in LSTMs.
- The sequential aspect of the model can be preserved using LSTM that store attributes of the past to be used in calculating further features.

0.4.3.3 Disadvantages :

- Vanishing gradient can be one of the causes of low accuracy.

LSTM in its core, preserves information from inputs that has already passed through it using the hidden state. Vanilla LSTM only preserves information of the past because the only inputs it has seen are from the past. This is where bidirectional LSTMs comes in.This helped us run inputs using both the past and the future. Using the combined hidden states for the past and the future we create a more comprehensive classification for our features.

0.4.4 Bi-directional LSTMs

0.4.4.1 Intro :

Bidirectional LSTM (BLSTM) connect two hidden layers of opposite directions to the same output. With this the output layer can get information from past (backwards) and future (forward) states simultaneously.BLSTM are especially useful when the context of the input is needed. For example, in handwriting recognition, the performance can be enhanced by knowledge of the letters located before and after the current letter.

0.4.4.2 Advantages :

- Using both past and present features to classify. Handles the cases where past and future interdependancy is present.

0.4.4.3 Disadvantages :

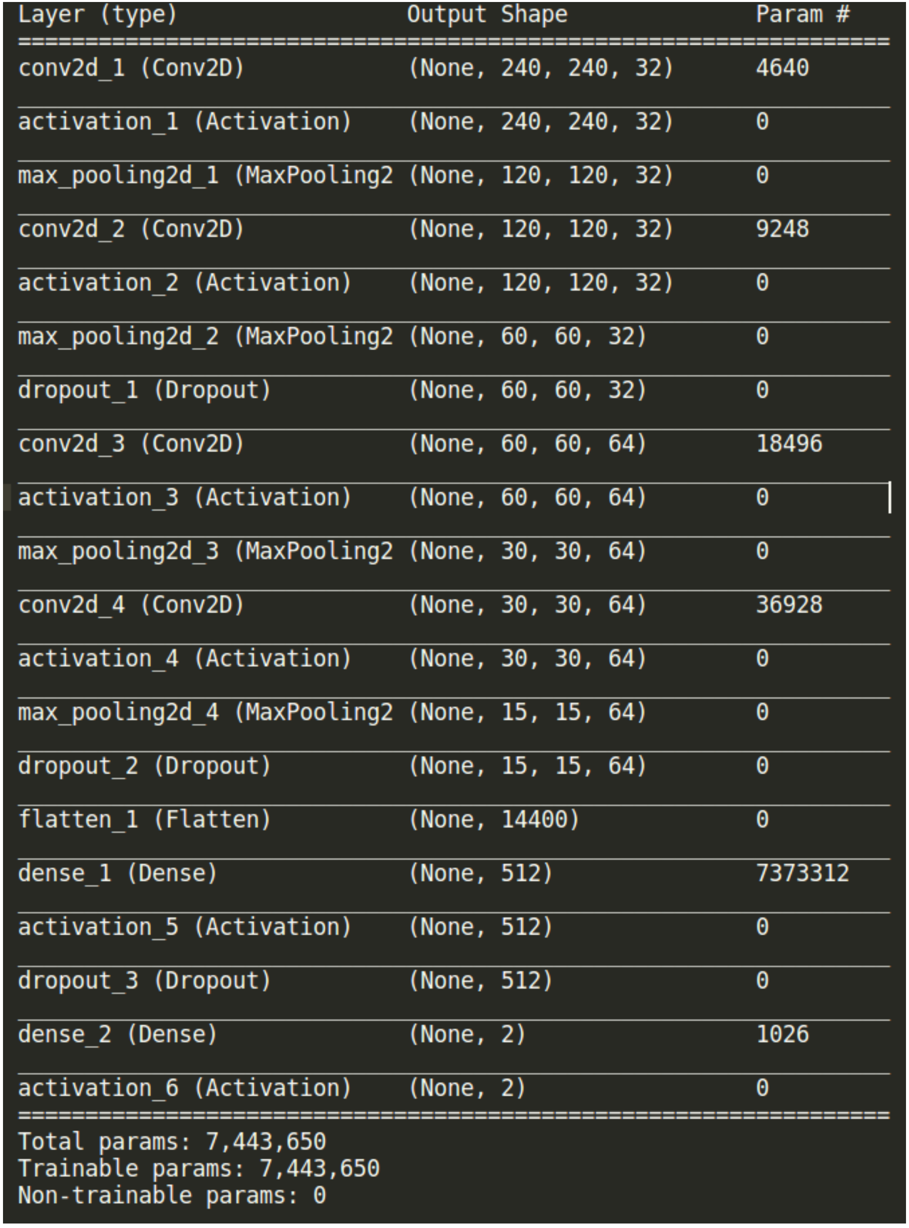
- Takes too much time to train.

0.5 Implementation

0.5.1 CNN

0.5.1.1 Architecture :

- Model summary



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 240, 240, 32)      4640
_________________________________________________________________
activation_1 (Activation)    (None, 240, 240, 32)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 120, 120, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 120, 120, 32)      9248
_________________________________________________________________
activation_2 (Activation)    (None, 120, 120, 32)      0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 60, 60, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 60, 60, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 60, 60, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 60, 60, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 30, 30, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 30, 30, 64)        36928
_________________________________________________________________
activation_4 (Activation)    (None, 30, 30, 64)        0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 15, 15, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 15, 15, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 14400)             0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               7373312
_________________________________________________________________
activation_5 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1026
_________________________________________________________________
activation_6 (Activation)    (None, 2)                 0
=================================================================
Total params: 7,443,650
Trainable params: 7,443,650
Non-trainable params: 0
```

0.5.1.2 Results :

- Smoking dataset
  - Confusion Matrix

prediction outcome

		p	**n**	**total**
actual value	**p′**	TP 0.48	FN 0.12	P′
	n′	FP 0.26	TN 0.14	N′
	total	P	N	

  - Performance measures

    Accuracy: 0.62

    Recall: 0.8

    Precision: 0.64

    F1 score: 0.71

- Obscenity dataset
  - Confusion Matrix

	p	**n**	**total**
p′	TP 0.39	FN 0.14	P′
n′	FP 0.33	TN 0.14	N′
total	P	N	

  - Performance measures

    Accuracy: 0.53

    Recall: 0.73

    Precision: 0.54

    F1 score: 0.62

- Drinking dataset

  - Confusion matrix

	p	**n**	**total**
p′	TP 0.29	FN 0.22	P′
n′	FP 0.20	TN 0.29	N′
total	P	N	

  - Performance measures

    Accuracy: 0.58

    Recall: 0.56

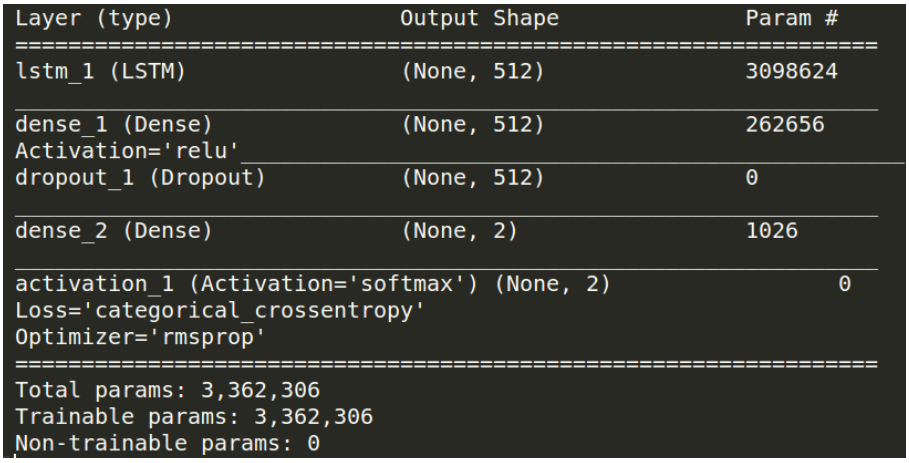
Precision: 0.59

F1 score: 0.57

0.5.2 LSTM

0.5.2.1 Architecture :

- Model summary



```
Layer (type)                     Output Shape              Param #
=================================================================
lstm_1 (LSTM)                    (None, 512)               3098624
_________________________________________________________________
dense_1 (Dense)                  (None, 512)               262656
Activation='relu'________________________________________________
dropout_1 (Dropout)              (None, 512)               0
_________________________________________________________________
dense_2 (Dense)                  (None, 2)                 1026
_________________________________________________________________
activation_1 (Activation='softmax') (None, 2)                  0
Loss='categorical_crossentropy'
Optimizer='rmsprop'
=================================================================
Total params: 3,362,306
Trainable params: 3,362,306
Non-trainable params: 0
```

0.5.2.2 Results :

- Smoking dataset
  - Confusion matrix

	p	**n**	**total**
p′	TP 0.48	FN 0.15	P′
n′	FP 0.26	TN 0.25	N′
total	P	N	

  - Performance measures

    Accuracy: 0.64

    Recall: 0.76

    Precision: 0.64

    F1 score: 0.70

- Obscenity dataset

  - Confusion matrix

	p	**n**	**total**
p′	TP 0.41	FN 0.08	P′
n′	FP 0.32	TN 0.17	N′
total	P	N	

  - Performance measures

    Accuracy: 0.59

    Recall: 0.83

    Precision: 0.56

    F1 score: 0.67

- Drinking dataset

- Confusion matrix

	p	**n**	**total**
p′	TP 0.32	FN 0.26	P′
n′	FP 0.23	TN 0.19	N′
total	P	N	

- Performance measures

  Accuracy: 0.51

  Recall: 0.55

  Precision: 0.58

  F1 score: 0.56

- Learning Curves

Figure 3: Label : Smoking Error Curve LSTM

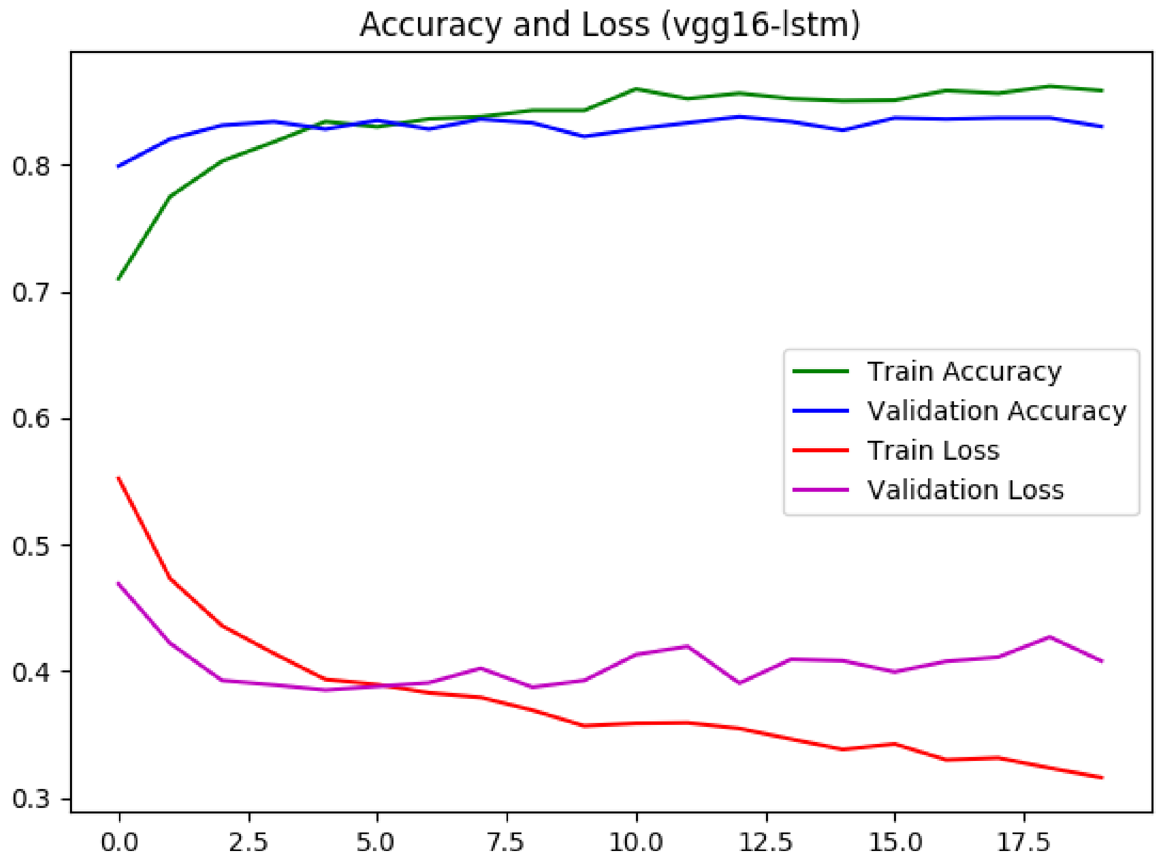


Figure 4: Label : Drinking Error Curve LSTM

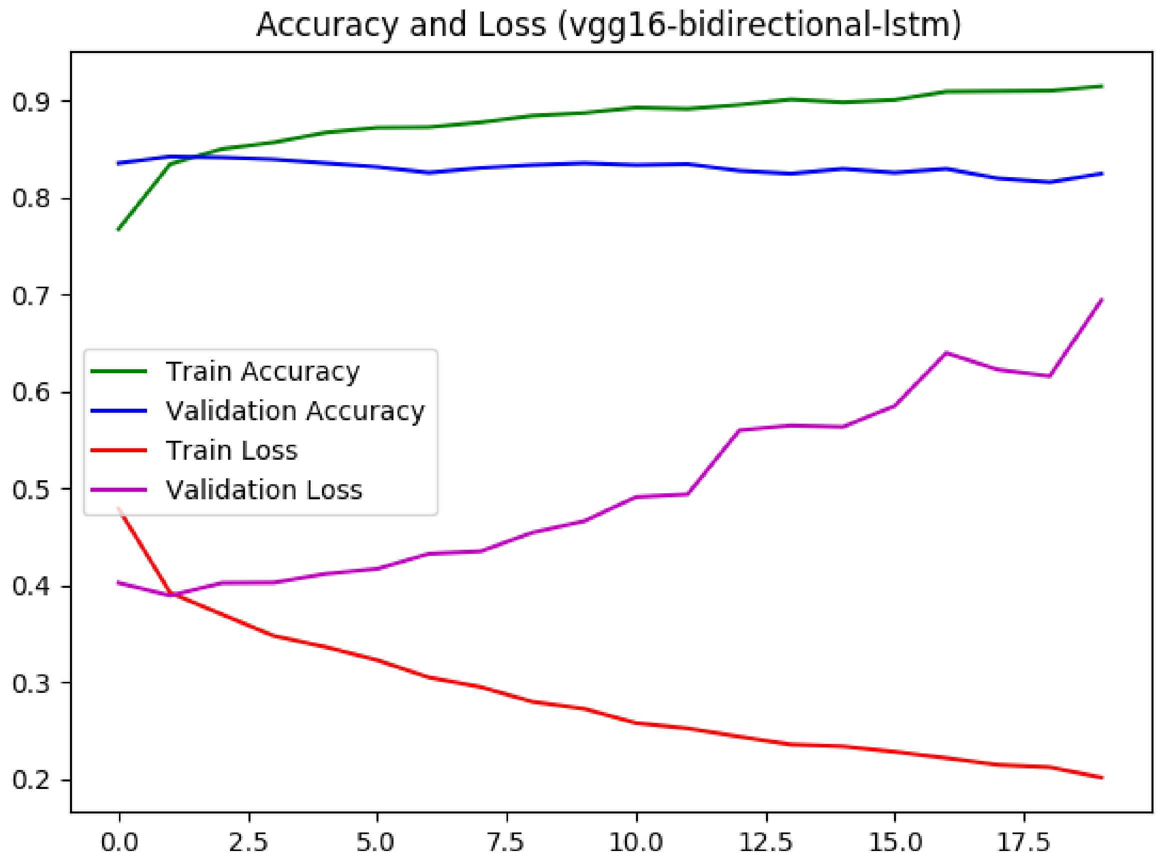
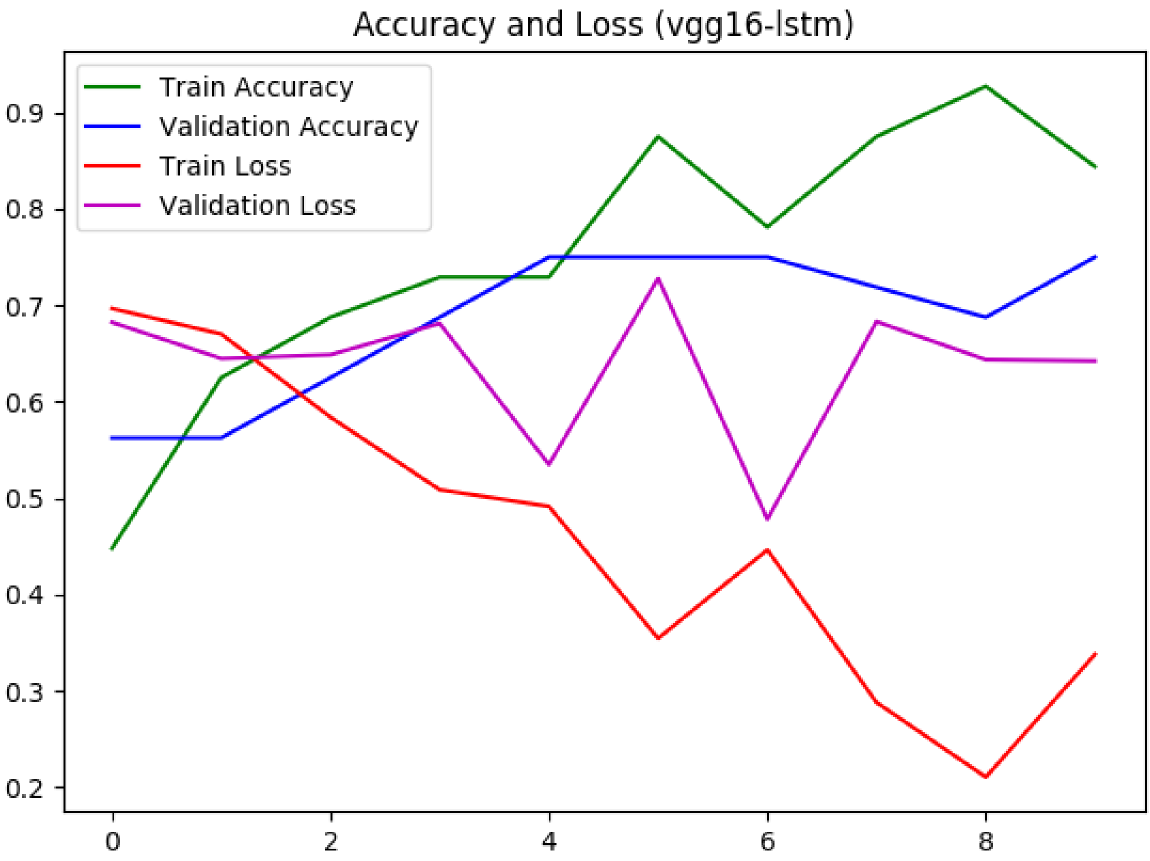


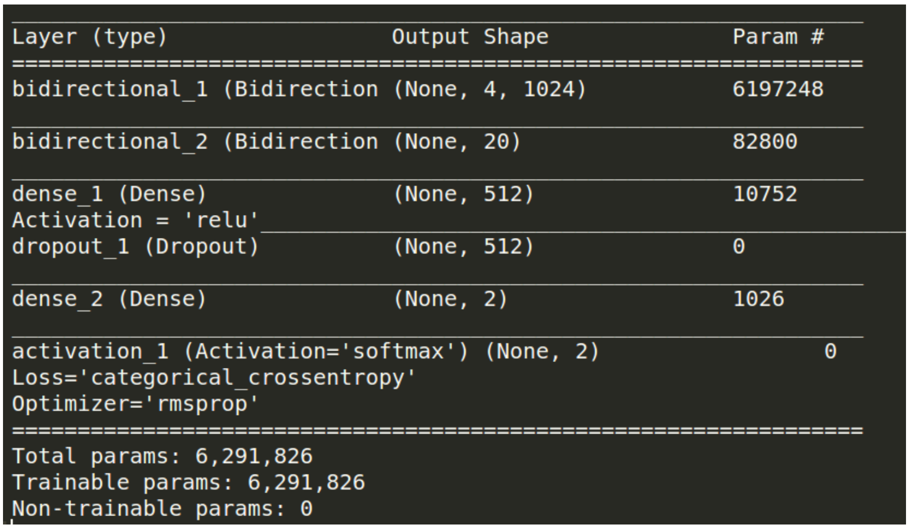
Figure 5: Label : Obscenery Error Curve LSTM



0.5.3 BLSTM

0.5.3.1 Architecture :

- Model summary



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_1 (Bidirection (None, 4, 1024)           6197248
_________________________________________________________________
bidirectional_2 (Bidirection (None, 20)                82800
_________________________________________________________________
dense_1 (Dense)              (None, 512)               10752
Activation = 'relu'_______________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1026
_________________________________________________________________
activation_1 (Activation='softmax') (None, 2)                 0
Loss='categorical_crossentropy'
Optimizer='rmsprop'
=================================================================
Total params: 6,291,826
Trainable params: 6,291,826
Non-trainable params: 0
```

0.5.3.2 Results :

- Smoking dataset
  - Confusion Matrix

	p	**n**	**total**
p′	TP 0.48	FN 0.04	P′
n′	FP 0.26	TN 0.22	N′
total	P	N	

  - Performance measures

    Accuracy: 0.7

    Recall: 0.92

    Precision: 0.64

F1 score: 0.76

- Obscenity dataset
  - Confusion Matrix

	p	**n**	**total**
p′	TP 0.35	FN 0.15	P′
n′	FP 0.17	TN 0.33	N′
total	P	N	

  - Performance measures

    Accuracy: 0.68

    Recall: 0.7

    Precision: 0.67

    F1 score: 0.68

- Drinking dataset
  - Confusion Matrix

	p	**n**	**total**
p′	TP 0.26	FN 0.24	P′
n′	FP 0.14	TN 0.36	N′
total	P	N	

- Performance measures

  Accuracy: 0.62

  Recall: 0.52

  Precision: 0.65

  F1 score: 0.57

- Learning Curves

Figure 6: Label : Smoking Error Curve BLSTM

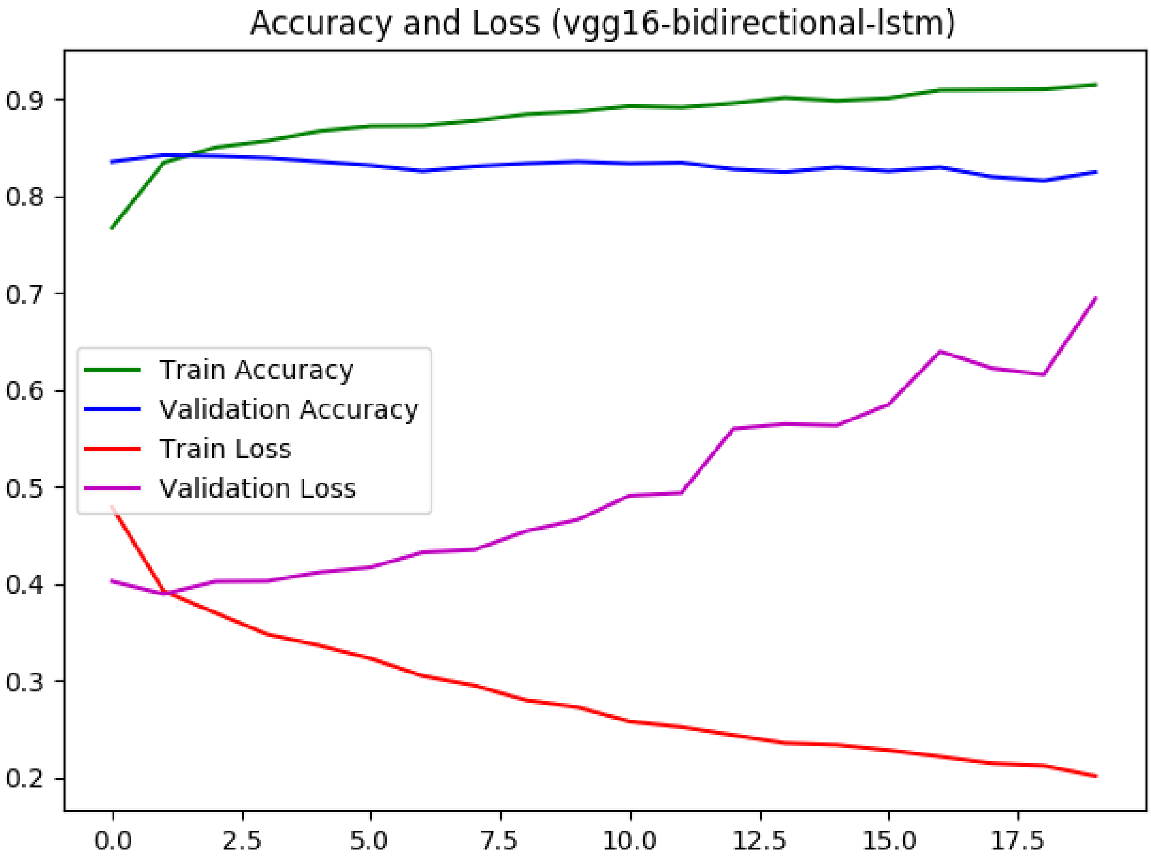


Figure 7: Label : Drinking Error Curve BLSTM

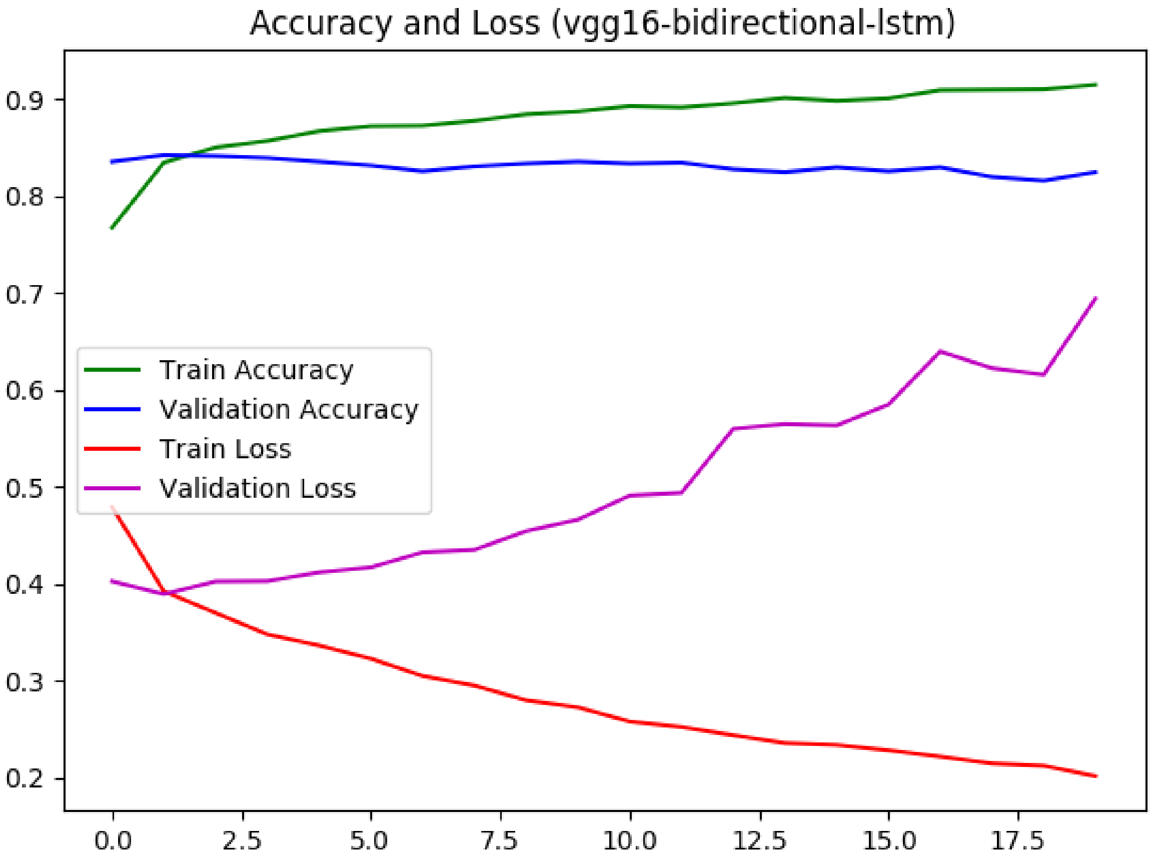
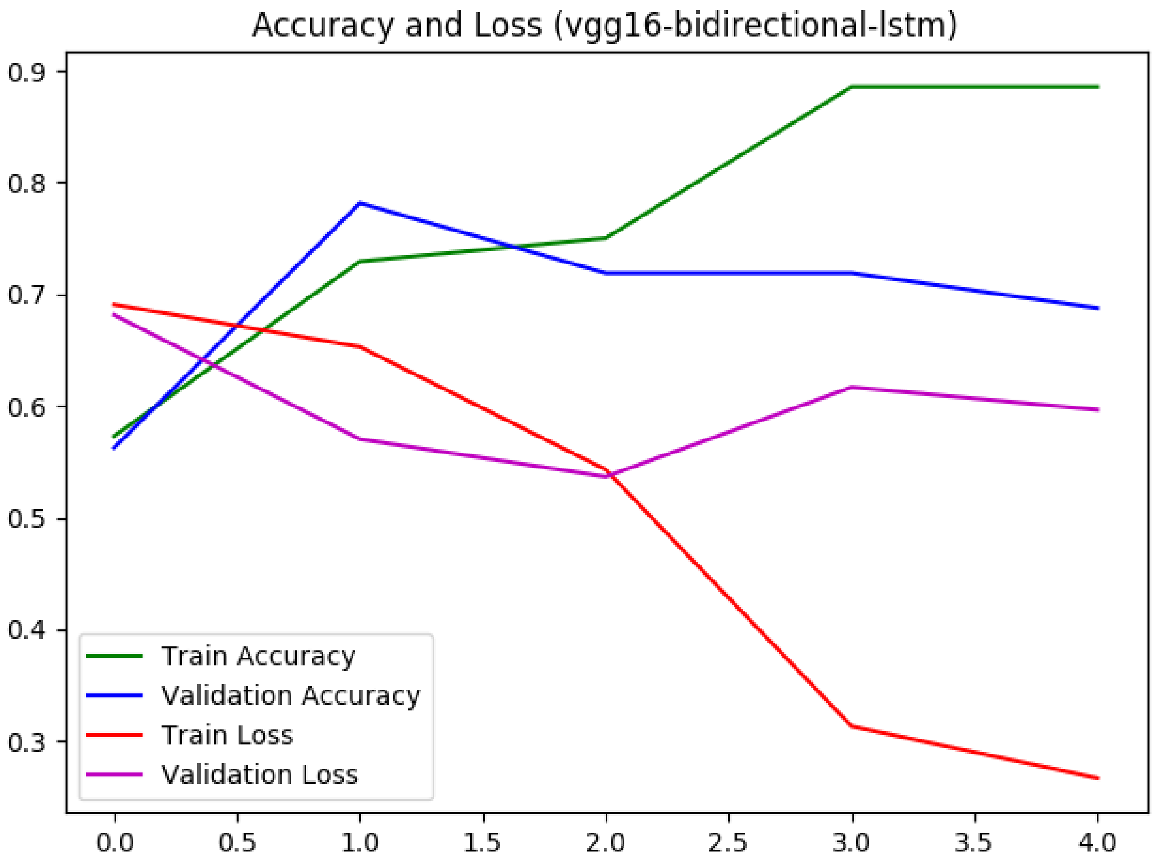


Figure 8: Label : Obscenery Error Curve BLSTM



0.6 Work Distribution

Work Distribution	
Team Member	Contribution
Swapnil	Worked on BLSTM model, code base, and data collection
Ritvik	Worked on VGG-16 LSTM model, data collection and annotation
Ayush	Worked on CNN-model, code base, and data collection
Vikrant	Worked on code base, Data collection and annotation

References

- LOPES, Ana P. B. ; AVILA, Sandra E. F. ; PEIXOTO, A. ; OLIVEIRA, Rodrigo S.; COELHO, Marcelo de M.; ARAÃŽJO, Arnaldo de A.. Nude Detection in Video using Bag-of-Visual-Features. In: 22th Brazilian Symposium on Computer Graphics and Image (SIBGRAPI), Rio de Janeiro, RJ, 2009.
- H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A Large Video Database for Human Motion Recognition. ICCV, 2011
- Jiajun Sun, Jing Wang, Ting-chun Yeh, Video Understanding: From Video Classification to Captioning, 2017