SMAI PROJECT MID EVAL

VIOLATION DETECTION IN VIDEOS

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

Contents

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

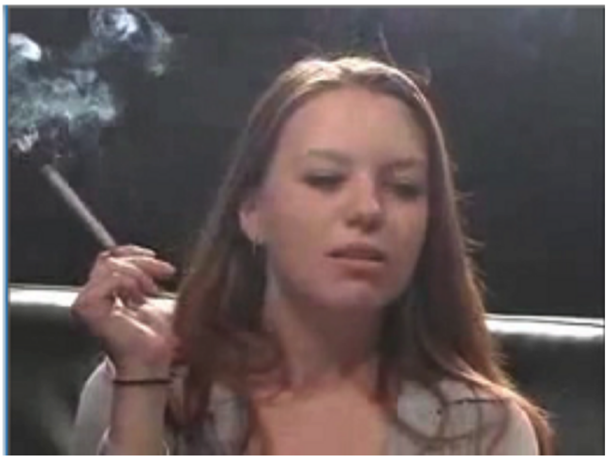
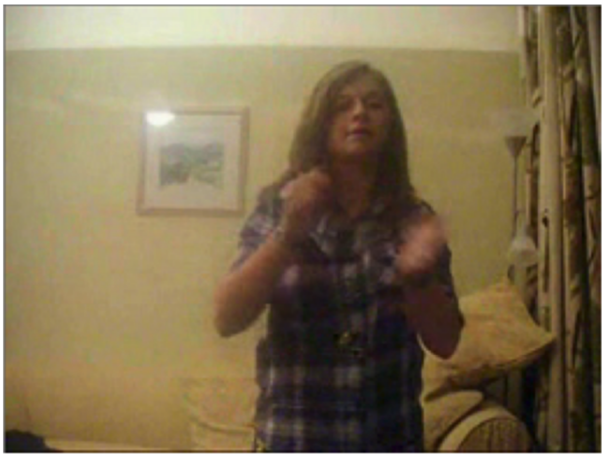


Figure 2: Label : Non-Smoking



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 Dataset Details

0.2.1 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.

0.2.2 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.

0.2.3 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.

0.3 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.

0.3.1 CNN

0.3.1.1 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.

0.3.1.2 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.

0.3.1.3 Disadvantages :

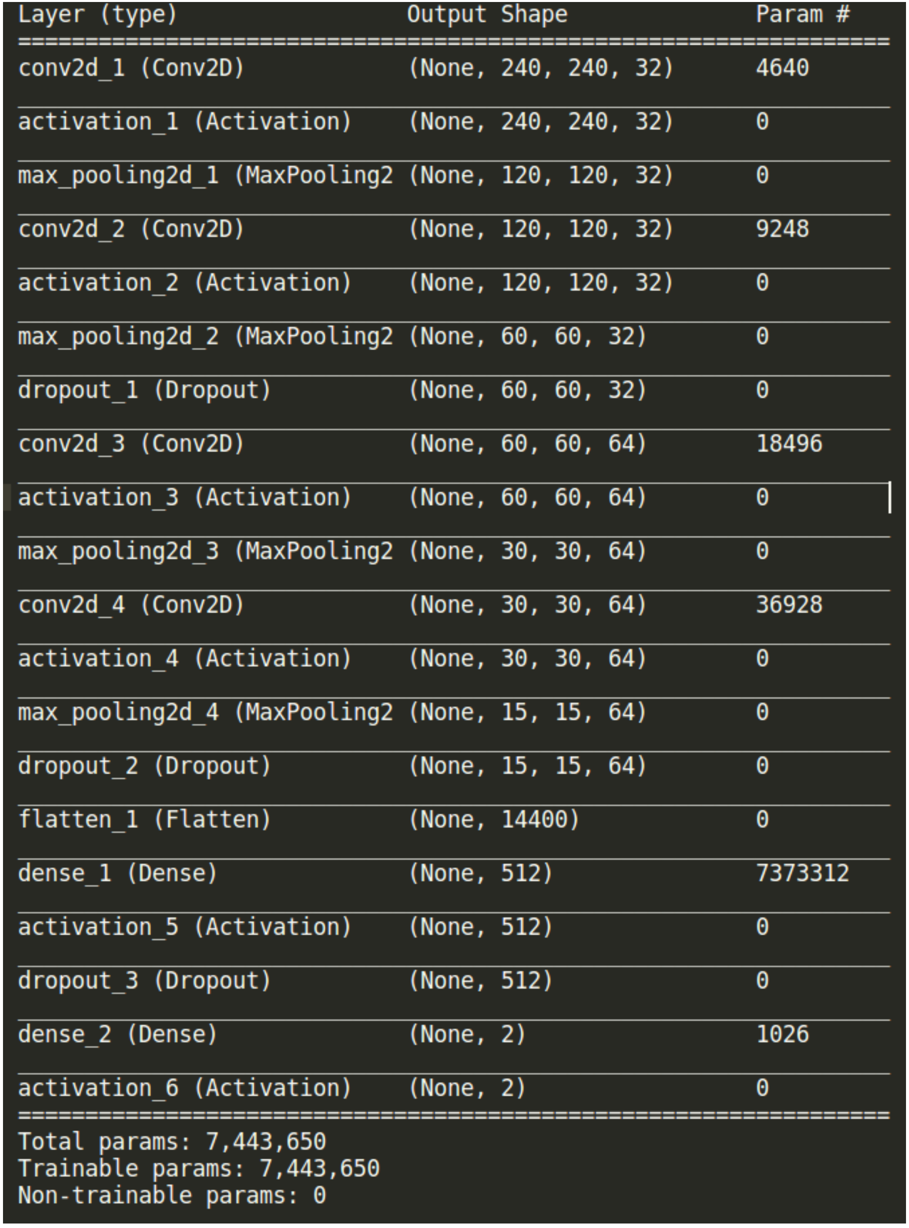
- Does not have any scope to take into account time dependency of data and hence doesn't work well on sequential data.
- Large number of parameters and values are required.

0.4 Implementation

0.4.1 CNN

0.4.1.1 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
```

0.4.1.2 Results :

- Smoking dataset
  - Confusion Matrix

		prediction outcome		
		p	**n**	**total**
actual value	**p′**	TP 0.46	FN 0.16	P′
	n′	FP 0.28	TN 0.10	N′
	total	P	N	

  - Performance measures

    Accuracy: 0.56

    Recall: 0.74

    Precision: 0.62

    F1 score: 0.67

References

- 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