

Automatic Parental Guide Ratings for Short Movies

BY

CHAI ZI XU

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2021

REPORT STATUS DECLARATION FORM

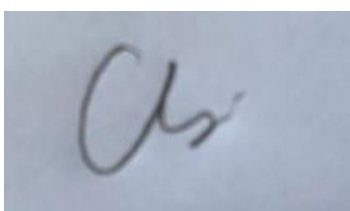
Title: Automatic Parental Guide Ratings for Short Movies

Academic Session: 01/2021

I CHAI ZI XU
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



(Author's signature)

Verified by,



(Supervisor's signature)

Address:

113, Lorong 6,

Taman Langkap Jaya

36700 Langkap, Perak.

Dr Aun Yichiet

Supervisor's name

Date: 15/04/2021

Date: 15/04/2021

Automatic Parental Guide Ratings for Short Movies

BY

CHAI ZI XU

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Automatic Parental Guide Ratings for Short Movies**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : CHAI ZI XU

Date : 15/04/2021

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Aun Yichiet who has given me this bright opportunity to engage in an AI development project. A million thanks to you for all the support and advice to make this project possible. Furthermore, I would also like to thanks to my course mates and friends for their unconditional support. Finally, I must say thanks to my parents and family for their love, support and continuous encouragement throughout the course

ABSTRACT

Video description is helpful for automatic movie ratings and annotating parental guides. However, human-annotated ratings are somewhat ambiguous depending on the types of movies and demographics. This project proposes a Machine-learning (ML) pipeline to generate a parental rating for short movies automatically. The ML pipeline infers and resolves various entities from 5 custom trained ML models trained using a corresponding public dataset. These ML models include nudity scene detection, violent scene detection, profanity scene detection, alcohol & drugs detection. A nudity detection scene is trained using YOLOv4 to detect possible scenes exposing private parts and genitals. Meanwhile, violent scene detection is trained using custom RNN-LSTM to detect possible fighting and gore scenes. Next, the profanity detection uses Google Text-to-Speech API to transcribe audio before feeding it into a custom better-profanity library. Lastly, the alcohol & drug models are trained using features extracted from VGG-16 then fed into a one-class CNN classifier. The experimental result showed that the proposed automatic rating is highly accurate when compared to manually annotated ratings.

Contents

ABSTRACT	vi
LIST OF FIGURES	ix
LIST OF TABLES	xi
LIST OF SYMBOLS	xii
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	1
1.1 PROBLEM STATEMENT AND MOTIVATION	1
1.2 BACKGROUND INFORMATION	2
1.2.1 Convolutional neural network (CNN)	2
1.2.2 Video Speech Recognition	5
1.3 PROJECT OBJECTIVES	6
1.4 PROPOSED APPROACH	6
1.5 HIGHLIGHTS	7
1.6 REPORT ORGANIZATION	7
CHAPTER 2 – LITERATURE REVIEW	8
2.1 Object Recognition	8
2.2 Human Activity Recognition and Detection	9
2.3 Violent Scenes Detection in Movies with Deep Learning	11
2.4 Dense-Captioning Events in Videos	13
2.5 Violent Scene Detection in Videos	14
2.6 Multimodal Data Fusion for Sensitive Scene Localization	16
2.7 CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet	17
CHAPTER 3- PROPOSED METHOD/APPROACH	20
3.1 SYSTEM DESIGN / OVERVIEW	20
3.1.1 Video Audio	21
3.1.2 Video Transcript	21
3.1.3 Video Frames	21
3.1.4 Violent Scene Detection	22
3.1.5 Profanity Scene Detection	24
3.1.6 Nudity Scene Detection	25
3.1.7 Alcohol & Drug Detection	25
3.1.8 Feature Aggregation And Generate Description	27
3.1.9 GUI	28
CHAPTER 4 – SPECIFICATION AND PLANS	29

4.1 TOOLS TO USE	29
4.2 VERIFICATION PLAN	30
CHAPTER 5 – EXPERIMENTS AND EVALUATIONS	33
5.1 EXPERIMENT SETUP	33
5.1.1 Violent Scene Detector	33
5.1.2 Nudity Scene Detector	34
5.1.3 Profanity Scene Detector	35
5.1.4 Alcohol & Drug Detector	36
5.2 EVALUATION RESULTS	38
5.2.1 Violent Scene Detector	38
5.2.2 Alcohol & Drug Detector	41
5.2.3 Nudity Scene Detector	43
5.3 IMPLEMENTATION EXAMPLES	44
CHAPTER 6 – CONCLUSION	47
6.1 OVERVIEW	47
6.2 FUTURE WORKS	48
Bibliography	49
APPENDIX A WEEKLY LOG	A-1
APPENDIX B- POSTER	B-1
APPENDIX C PLAGARISM CHECK RESULT	C-1
APPENDIX D TURNITIN FORM	D-1
APPENDIX E CHECKLIST	E-1

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.2.1.1	Neural network with convolutional layers	2
Figure 1.2.1.2	Some common filters in CNN	3
Figure 1.2.1.3	Types of Pooling	4
Figure 1.4.1	Proposed Method System Design	6
Figure 2.1.1	The YOLO Detection System	8
Figure 2.2.1	Block diagram of approach used in HAR	9
Figure 2.2.2	Framework of HAR	9
Figure 2.3.1	The key components of Violent Scene Detection System	11
Figure 2.3.2	The structure of the LSTM network	12
Figure 2.4.1	Dense-captioning events in a video	13
Figure 2.4.2	Complete pipeline for dense-captioning events in videos with descriptions	13
Figure 2.5.1	Overview of Two Stream CNN Architecture	14
Figure 2.5.2	Spatial and Temporal Stream CNN Input	15
Figure 2.6.1	Proposed fusion training pipeline	16
Figure 2.7.1	LeNet-5	17
Figure 2.7.2	AlexNet	18
Figure 2.7.3	GoogLeNet	18
Figure 2.7.4	VGGNet	19
Figure 2.7.5	ResNet	19
Figure 3.1.1	Proposed System Design	20
Figure 3.1.2	Proposed LSTM model	22
Figure 3.1.3	Architecture of the violent scene detector model	23
Figure 3.1.4	Sample of swear words in profanity_wordlist.txt	24
Figure 3.1.5	Schema of the proposed approach for alcohol & drug model	26
Figure 3.1.6	Architecture for the alcohol detector model	27
Figure 3.1.7	Architecture for the drug detector model	27
Figure 3.1.8	GUI for the explicit scene detector	28
Figure 4.2.1	Confusion Matrix	30
Figure 5.1.1	Sample of datasets that labelled by bounding box	34
Figure 5.1.2	Directory structure of dataset file for alcohol & drug scene detector	36

Figure 5.2.1	Accuracy chart for violent scene detector	38
Figure 5.2.2	Loss Chart for violent scene detector	38
Figure 5.2.3	Confusion matrix for violent scene detector model	40
Figure 5.2.4	Confusion matrix for alcohol detector model	41
Figure 5.2.5	Confusion matrix for drug detector model	42
Figure 5.2.6	Tensorboard metrics for nudity scene detector model	43
Figure 5.3.1	Sample Video 1	44
Figure 5.3.2	Sample Video 2	45
Figure 5.3.3	Sample Video 3	46

LIST OF TABLES

Table Number	Title	Page
Table 1.2.2.1	The advantage and disadvantage of the Google Speech-to-Text system	5
Table 2.2.1	Generalized Tabulation for UT-interaction dataset	10
Table 4.1.1	Current Setup of my laptop	29
Table 5.1.1	Parameter for violent scene detector model	33
Table 5.1.2	Parameter for nudity scene detector model	35
Table 5.1.3	Parameter for alcohol & drug model	37
Table 5.2.1	System performance of the violent scene detector model	39
Table 5.2.2	System performance for each class for violent scene detector model	40
Table 5.3.1	Result of Actual Label vs Predicted Label on sample video 1	44
Table 5.3.2	Result of Actual Label vs Predicted Label on sample video 2	45
Table 5.3.3	Result of Actual Label vs Predicted Label on sample video 3	46

LIST OF SYMBOLS

B	Reference frame
$C(t)$	Frames obtained at time interval t
$P[C(t)]$	Pixel value of current image frame
$P[B]$	Corresponding pixel value at the same position of the background image
$P[F(t)]$	Intensity components for the pixel locations
T	Threshold value
U	Input video frames
s_i	Set of sentences
t^{start}	Sentence start time
t^{end}	Sentence ending time
v_j	Sets of words with different length with vocabulary set V
P	Set of proposals
h_i	Hidden representation of the event
a_{ij}	Attention used to determine how relevant event j is to event i
Z	Normalization
w_i	Annotation vector
w_a	Learnt weights
b_a	Bias

LIST OF ABBREVIATIONS

<i>DPM</i>	Deformable Part Model
<i>YOLO</i>	You Only Look Once
R-CNN	Region-Based Convolutional Neural Network
SSD	Single Shot Detector
<i>CV</i>	Computer Vision
<i>NLP</i>	Natural Language Processing
<i>UGVs</i>	User-Generated Videos
<i>VSD</i>	Violent Scene Detection
<i>HOG</i>	Histogram of Oriented Gradient
<i>SVM</i>	Support Vector Machine
<i>CNN</i>	Convolutional Neural Network
<i>FC</i>	Fully Connected
<i>LSTM</i>	Long Short Term Memory
<i>IDT</i>	Improved Dense Trajectories
<i>HOF</i>	Histogram of Optical Flow
<i>MBH</i>	Motion Boundary Histogram
<i>TrajShape</i>	Trajectory Shape
<i>FV</i>	Fisher Vector
<i>STIP</i>	Space-Time Interest Points
<i>MFCC</i>	Mel-Frequency Cepstral Coefficients
<i>BLEU</i>	Bilingual Evaluation Understudy
<i>CIDEr</i>	Consensus-based Image Description Evaluation
<i>METEOR</i>	Metric for Evaluation of Translation with Explicit Ordering

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT AND MOTIVATION

The proliferation of smart devices has enabled anyone with access to technology to be content creators. In modern days, the momentum of video shooting and movie making has shifted from a studio-centric to crowd-source content creation. Not only that, the rapid growth of the Internet has led to an increase in the number of user-generated videos (UGVs). When the number of content creation increases and decentralized, content management and parental ratings become more challenging without a unified control mechanism. Amateur videos that are flooding the Internet are not properly screened and may contain explicit scenes that are not suitable for public viewing. The side effects from unhealthy exposures to violent, horror, profanity and explicit scenes can traumatize viewers and at the same time set up some domino effect in building bad characters from early ages. For instances, children who were exposed to violent films displayed more signs of emotional distress, in terms of depression and lack of excitement (Fitzpatrick, 2018). Therefore, to overcome this problem a violent scene detection system can be built to detect violent scenes so that parents can choose the suitable movie or video for their children.

In current research landscape, apart from violent scene detection (VSD), most of these explicit scenes have not been widely explored. Although scenes detection can be trained with transfer learning from some existing VSD, the training process is less straightforward since the resulting models will easily overfit. In automatic genre classification, machine-learning is used to pervasively categorize movies into corresponding genre based on scene detection. Genre classification leverages on the ‘coverage’ of scene detection models. Specifically, the types of genres that can be recognized depends on the number of classes (or scenes) supported by scene detection models. Existing scene detection are highly imbalanced in terms of scene diversity, they gravitate towards VSD recognition instead of a more general purpose scene recognition such as nudity scene recognition and profanity scene recognition.

In genre classification, movie categories are ambiguously defined. For example, movies category in Netflix are hand labeled by a pool of expert annotators. Currently, Netflix

assume viewers and annotators sharing the same perception of movies, despite every these individuals having different demographic and background. Although averaging these annotation can somewhat minimize these perception disparity, movies labels are often drawn from some specific scenes and intuitively generalized. Besides that, some movies with more complex storyline that contains a combination of genre elements are difficult to be conveyed. As a result, the recommendations given to viewers that browse movie based on category are mostly generic and are not reflected from an aggregated scenes of interests throughout the movies. Therefore, a system that can list out all the explicit scenes of a video or movie is needed for the convenience of the viewers to choose a movie they like.

1.2 BACKGROUND INFORMATION

1.2.1 Convolutional neural network (CNN)

Convolutional Neural Networks (CNN) is a neural network variant that specialises in processing and classifying of images. As the name suggests, the hidden layers of a CNN are composed of convolutional layers and pooling layers interweaving between them. Those layers are different with the usual layers because the convolution and pooling functions are replacing the activation functions. In addition, a CNN's hidden layers are also composed of pooling layers and fully connected layers. A full CNN flow to process an input image and perform value-based object classification is as shown in figure below.

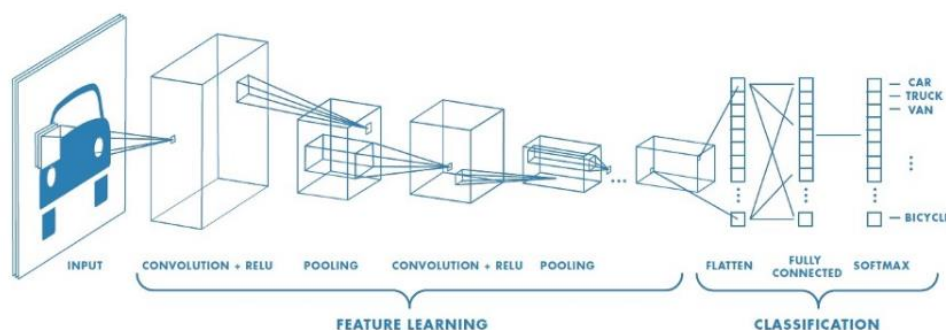


Figure 1.2.1.1: Neural network with convolutional layers

CHAPTER 1 INTRODUCTION

Convolution is the first layer in which features are extracted from an input image. By utilizing small squares of input data to learn image features, convolution remains the relationship between pixels. It is a mathematical operation that involves two inputs such as image matrix and a filter or kernel. Convolution of an image with various filters could perform operations like blurring, sharpening and edge detection. The example below shows different convolution image with various types of filters.



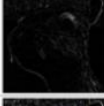



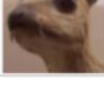
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Figure 1.2.1.2: Some common filters in CNN

Furthermore, when the images are too large, the pooling layers section will decrease the number of parameters. The spatial pooling is able to decrease the dimensionality of each map but the important information is retained. There are various kinds of spatial pooling: max pooling, average pooling and sum pooling.

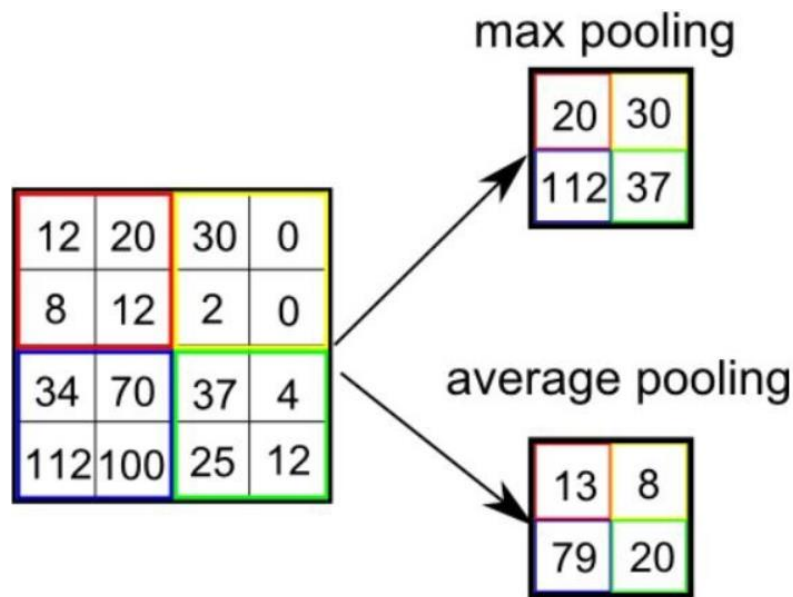


Figure 1.2.1.3: Types of Pooling

Max pooling returns the maximum value from the rectified feature map. On the other hand, average pooling returns the average of all the values from the rectified feature map and the sum pooling returns the sum of all elements in the feature map. The function of the max pooling is noise suppressant. It removes the noisy activations and conducts de-noise along with reduction of the dimensionality while the average pooling simply performs dimensionality reduction. Therefore, we can assume that max pooling does much better than average pooling.

Lastly, the fully-connected layer (FC layer) is used to learn non-linear combinations of the high-level features as represented by the convolutional layer's output. The image matrix is flattened into a column vector and fed to a feed-forward neural network and backpropagation applied to every iteration of training. These features are then combined to make a model. Finally, the activation function like softmax or sigmoid will do classification on the outputs.

1.2.2 Video Speech Recognition

Google Speech-to-Text helps developers to translate audio to text using powerful neural network models in a user-friendly API (Google Cloud, 2020). The API can recognize over 120 languages and variants to maintain global user base. One can make command - and - control voice, transcribe audio from call centre, etc. Google's machine learning algorithms can also process real-time streaming or pre-recorded audio. In addition, it will recognize spoken language automatically. Using Speech-to - Text one may recognize which language (up to four languages) is spoken in the utterance.

One can extract voice information from the video's audio channel. By using speech recognition, more comprehension of the video will be provided with additional semantic information. The speech recognition can also produce spoken word in natural language. With this, if there are vulgar language in the video, it can be detected and help users to choose movies that are appropriate for their children. The advantage and disadvantage of the Google Speech-to-Text system can be summarized in Table 2.4.1 below:

	Advantages	Disadvantages
Google Speech-to-Text system	<ul style="list-style-type: none"> -Automatic speech recognition -Global vocabulary -Auto-Detect Language -Noise robustness -Multichannel recognition -Automatic punctuation -Real-time streaming or prerecorded audio support 	<ul style="list-style-type: none"> -Not cost effective -Translation not completely accurate -may have problems with slang, technical words and acronyms

Table 1.2.2.1: The advantage and disadvantage of the Google Speech-to-Text system

1.3 PROJECT OBJECTIVES

There are some objectives in this project:

- To design a pipeline in detecting explicit scenes in short movies for automatic parental guide description
- To train a violent scene detection model using RNN-LSTM to detect the presence of violent scenes in a movie
- To train a profanity detection model that leverages Google Speech-to-Text to identify utterance of vulgar words in a movie
- To train a nudity detection model using YOLOv4 to detect scenes that contain nudity elements
- To train object classification model using a one-class classifier to detect the presence of alcohols and drugs in a movie

1.4 PROPOSED APPROACH

The project is developing a system involving a pipeline and model to extract information from the sensitive scene in the video including the visual and audio cues. The data would then be encoded and transcribed into a more correct sentence to describe the content of the video. In this stage, to access the feasibility of the project, partial semantic information is extracted and encoded into sentence as a prototype. The project can also automatically classify the selected video based on the results obtained from several sensitive scene detector.

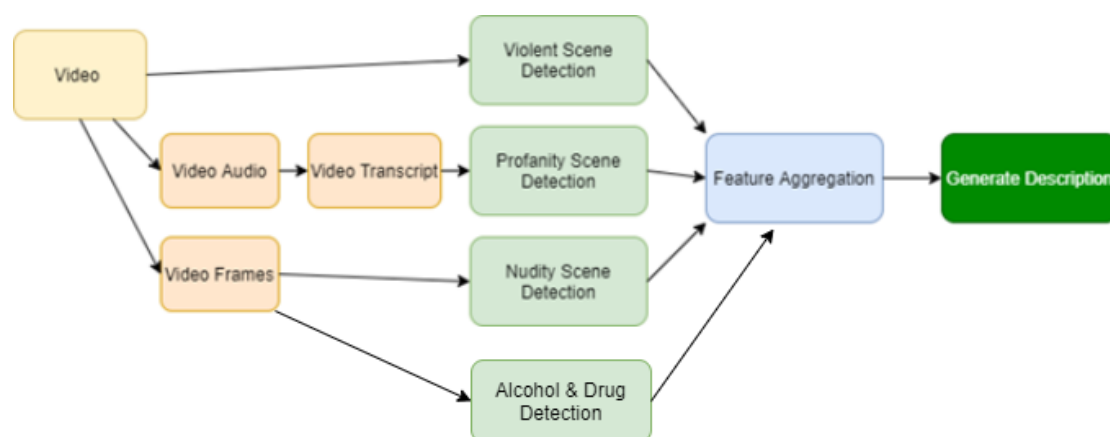


Figure 1.4.1 Proposed Method System Design

Figure 1.4.1 shows the design of the overall proposed system. In this approach, each task is broken down into an individual component in the pipeline. The individual component can be added, removed or replaced by a better model easily. The information that is extracted from the video selected by each scene detector will then be shown in the form of sentence (description).

1.5 HIGHLIGHTS

This contribution of this project is it will create a sentence with description regarding to the explicit scene of the video. This can benefit the viewers by helping them understand better about the video they watch. Also, it can automatically classify whether the video is a violent movie, nude movie, movie with profanity language or movie that contains alcohol and drug. This could act as a parental guide synthesizer because parents can understand better about the content of the video and choose suitable movie for their children.

Furthermore, this project extracts audio features in the video for the profanity scene detection. Most computer vision research has concentrated on visual feature without the assistance of audio feature, but audio is present in most videos, inevitably and indispensably. Audio can help in video description by providing important information such as the conversation between two persons, sound of train, ocean, traffic when there is no visual cue of their presence. Thus, the use of audio in video description models would definitely increase the performance (Nayyer Aafaq, 2020).

1.6 REPORT ORGANIZATION

The study is broken down into six parts. In Chapter 2, literature reviews are conducted on previous work in order to learn more about the methods used and to compare them to the proposed process. In Chapter 3, the proposed method's system design and implementation information are discussed. The system requirements / specifications, as well as verification plans, are addressed in Chapter 4. Continue to Chapter 5, where the outcomes of the experiments and evaluations are shown and analysed in order to assess the proposed method's efficiency. Finally, in Chapter 6, a conclusion is presented, as well as future project planning that are suggested and discussed.

CHAPTER 2 – LITERATURE REVIEW

2.1 Object Recognition

You Only Look Once (YOLO), a modern approach to detect objects. YOLO frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. In one test, a single neural network predicts bounding boxes and class probabilities straight from complete images. The detection performance can be optimized end-to - end directly since the whole detection pipeline is a single network. YOLO has extremely fast unified architecture. The foundation YOLO model processes images at 45 frames per second in real time. YOLO produces more localization errors compared with state-of-the-art detection systems but is less likely to predict false positives on the background. Finally, YOLO learns very general representations of objects. It surpasses other methods of detection, including DPM and R-CNN, when generalising natural images into other domains such as artwork (Joseph Redmon, 2016).

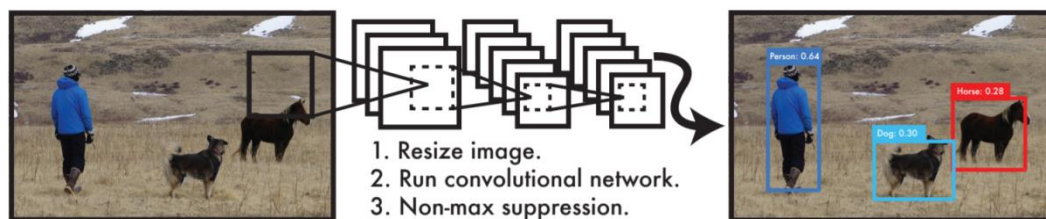


Figure 2.1.1: The YOLO Detection System

There is some updates that have been done to YOLO and YOLOv3 exists. YOLOv3 uses several tricks to boost training and performance, including multi-scale predictions, a stronger classifier for the backbone, etc. It is a little bigger but more precise than last time. It's still quick. At 320 x 320 YOLOv3 runs at 28.2 mAP in 22 ms, as precise as SSD but three times faster (Joseph Redmon, 2018). YOLOv3 model uses pre-trained weights for standard object detection problems.

2.2 Human Activity Recognition and Detection

This work focuses primarily on multiple identification and recognition of human activities (Chandrashekar M Patil, 2017). There are several human video databases being considered for the identification and monitoring of multiple people. Background subtraction technique is used to detect several people who are moving. It detects moving humans by taking difference between the current image frame with the reference image, the reference image is the background image of the video taken in static state.

Histogram of Oriented Gradient feature descriptor (HOG descriptor) is utilized for feature extraction. From each binary image, HOG features are extracted. This technique is used for calculating the occurrences of gradient in the localized portions of an image.

Support Vector Machine (SVM) classifier is utilized for the recognition of human activity. SVM uses nonlinear classification function that is defined by known samples to classify activities and thus solves the problem of parameter estimation in the state model method. The consideration of the distribution of probability isn't needed. Thus it has its own broad range of application. SVM uses an iterative training algorithm to create an optimal hyperplane to minimize an error function.

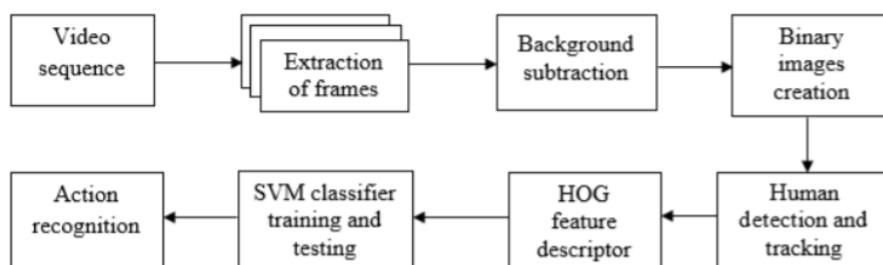


Figure 2.2.1: Block diagram of approach used in HAR

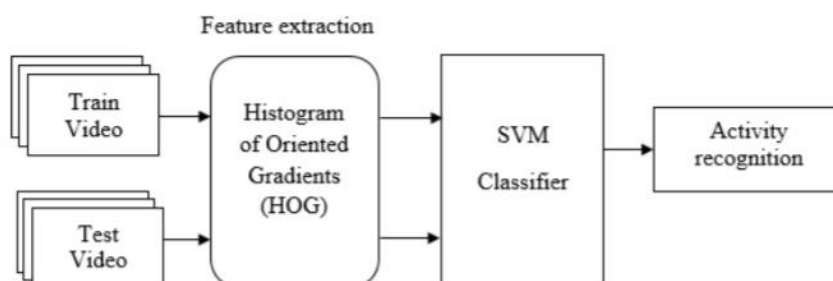


Figure 2.2.2: Framework of HAR

CHAPTER 2 LITERATURE REVIEW

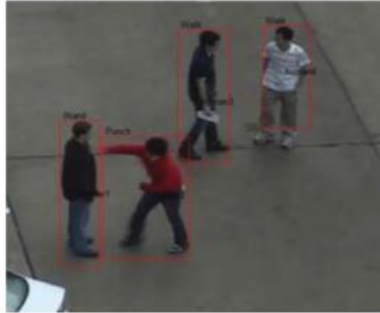

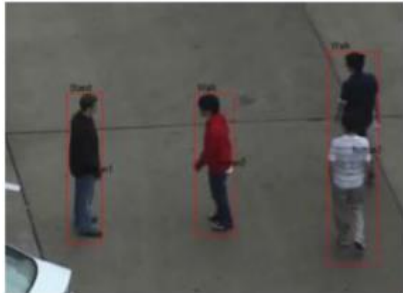
Frame	Number of frame	Action	Person
	1	Handshake	2
	2	Punch	1
		Stand	1
		Walk	2
	3	Stand	1
		Kick	1
		Walk	2
	4	Stand	1
		Walk	3

Table 2.2.1: Generalized Tabulation For UT-Interaction Dataset

The limitation of this system is that it uses the algorithm based on UT-interaction and own dataset for the recognition of human activity. Thus, it only can recognize few activities like walk, stand, punch, kick and handshake. To solve this limitation, the author can increase the number of training data or use other dataset from internet that contains more human activities.

2.3 Violent Scenes Detection in Movies with Deep Learning

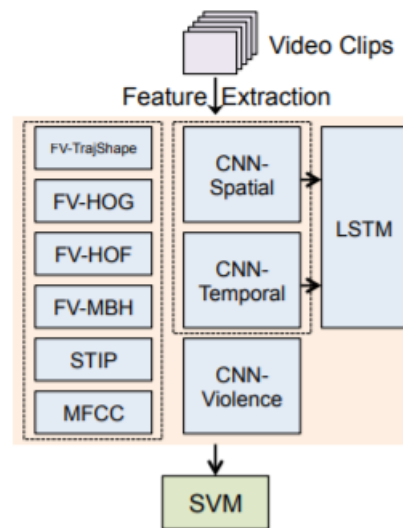


Figure 2.3.1: The key components of Violent Scene Detection System

This system consists of several deep learning features. First, they train a CNN model with a subset of ImageNet classes chosen particularly for the detection of violence (Dai et al., 2015). These classes are mainly among the scenes, people, weapons and actions categories. They trained CNN-violence which is an AlexNet model on video frames. In this model, individual frames are taken as network inputs followed by multiple convolutional layers, pooling layers and also fully-connected (FC) layers.

Second, they adopt a specially designed two-stream CNN framework (spatial stream and temporal stream) for extracting features on both static frames and motion optical flows. Next, on top of the two-stream CNN framework, Long Short Term Memory (LSTM) models are applied to capture the longer-term temporal dynamics. With a two-stream CNN model, video frames or stacked optical flows can be converted into a sequence of fixed-length vector representations. A LSTM model is aim to model such temporal information.

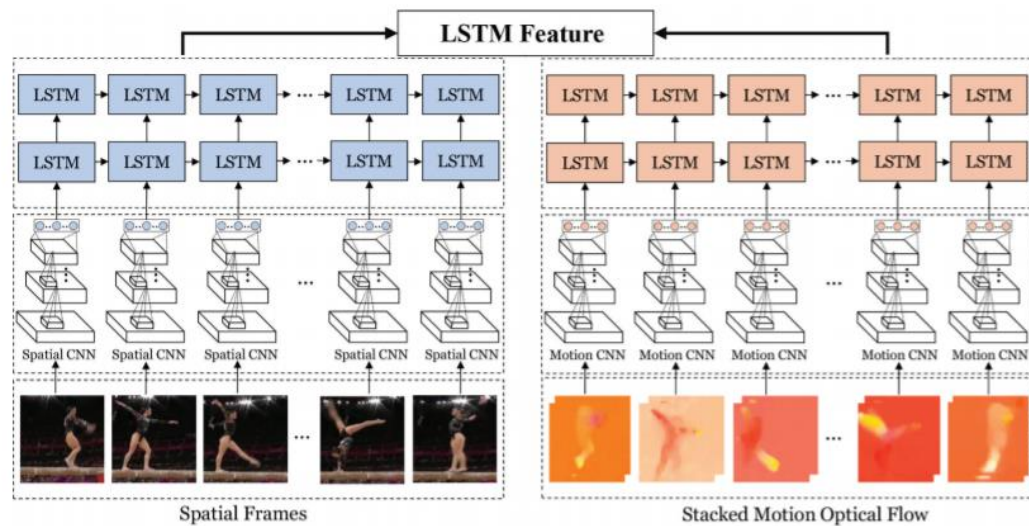


Figure 2.3.2: The structure of the LSTM network.

Also, many traditional motion and audio features are extracted in addition as the deep learning features as complementary details. Four improved dense trajectories (IDT) based features, including histograms of oriented gradients (HOG), histograms of optical flow (HOF), motion boundary histograms (MBH) and trajectory shape (TrajShape) descriptors are computed. The features are encoded with a codebook of 256 codewords by the Fisher vectors (FV). The other two types of conventional features are Space-Time Interest Points (STIP) and Mel-Frequency Cepstral Coefficients (MFCC).

Lastly, they choose SVM as the classifier. Linear kernel is used for the four IDT features, and 2 kernel is used for all the others. For feature fusion, kernel level fusion is adopted, which linearly combines kernels computed on different features. which linearly combines kernels computed on different features.

One common difficulty faced when training deep learning architectures is that video-based data are normally limited in both size and diversity. This might become a problem when training model because this requires a large amount of training samples to accomplish optimal performance. One of the method to solve this problem is to add regularization such as dropout after global average layers. This is to prevent overfitting during training.

2.4 Dense-Captioning Events in Videos

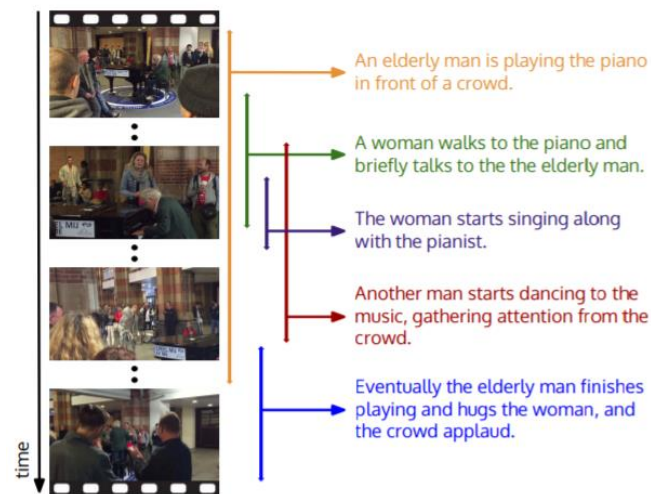


Figure 2.4.1: Dense-captioning events in a video

In this paper, they introduce the task of dense-captioning events to identify and describe all the events in a video in natural language (Ranjay Krishna, 2017). This event requires a model to make a set of descriptions for multiple events that occurred in the video and perform localization in time. They introduce a captioning module that uses the context from all the events from their proposal module to make each sentence.

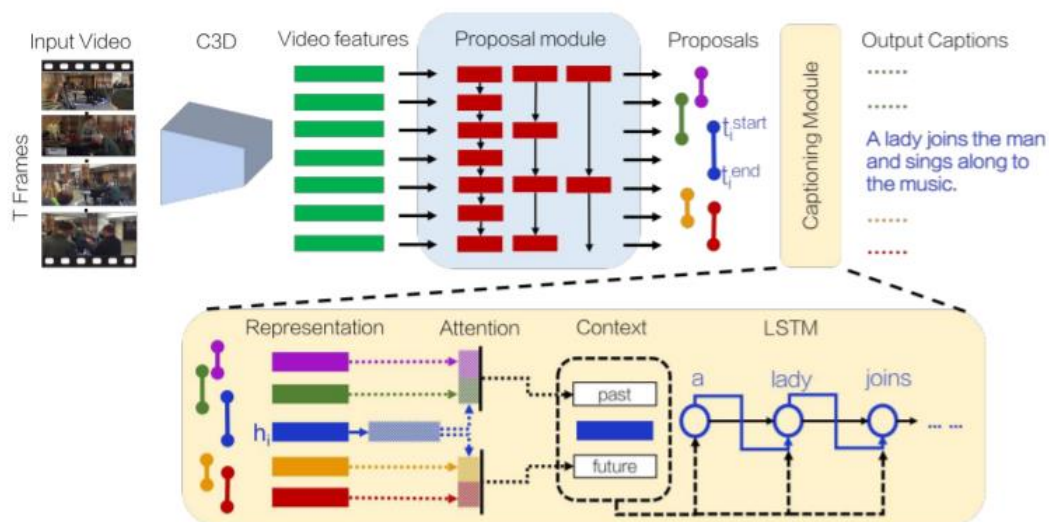


Figure 2.4.2: Complete pipeline for dense-captioning events

The model uses action proposal and social human tracking approach for detecting multiple events in both short and long video sequences and it use the context from past, concurrent and future events to generate each event descriptions. The author categorizes all events into two groups relative to a reference event, in order to collect the context

from all other neighbouring events. These two context categories capture events that have already happened (past), and events that occur after this event has ended (future). The concurrent events will be split into one of the two categories. It will be considered as ‘past’ when it ends early and future otherwise. At last, the concatenation of $(h_i^{past}, h_i, h_i^{future})$ is fed into LSTM to describe the event where h_i is a hidden representation for a video event.

2.5 Violent Scene Detection in Videos

Video data is composed of temporal and spatial components (Yew, 2019). The spatial component is the object details in each frame while the temporal component is the object motion across consecutive frames. Violent scenes normally have very distinct visual cues and these visual cues could be present in either components of video data. In this paper, a two stream CNN architecture is used to learn the features of violent scenes from both components for violent scene detection.

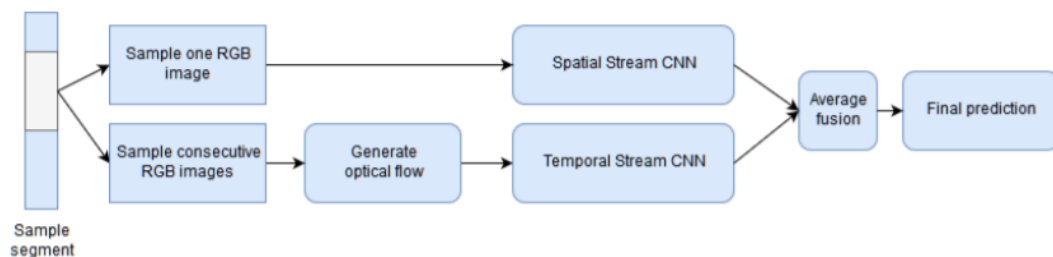


Figure 2.5.1: Overview of Two Stream CNN Architecture

As shown in Figure 2.6.1, this two stream CNN architecture uses 2 different modalities in video data for the detection of violent scene. The spatial stream CNN is modeled for the spatial component of video data, it extracts and learns useful features from individual RGB frames. On the other hand, temporal stream CNN is modeled for the video data’s temporal components. It extracts and learns valuable motion features from optical flow stack that generated from consecutive RGB frames. In addition, both spatial and temporal stream CNN use a variant of residual network, ResNet50 because it has a good balance between performance and number of parameters. Besides that, residual networks are easier to train as they have additional identity shortcut connections that decreases the effect of vanishing or exploding gradient during training.

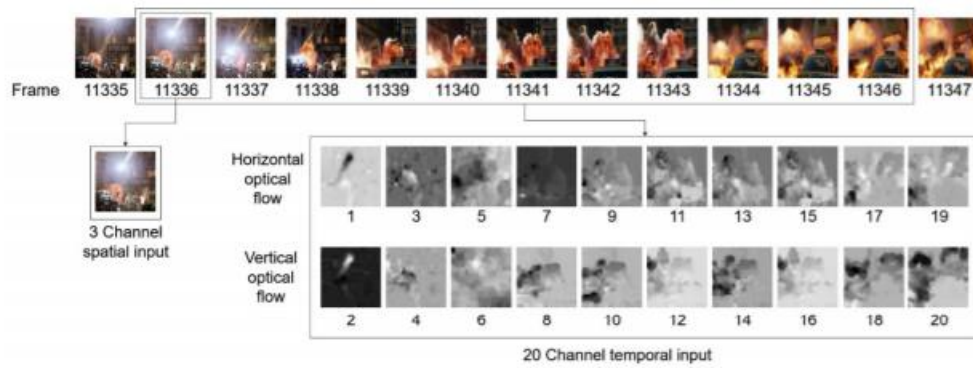


Figure 2.5.2: Spatial and Temporal Stream CNN Input

For the spatial stream CNN configuration, since the spatial stream CNN is taking RGB images at input, the input shape of ResNet50 is remain unchanged as $224 \times 224 \times 3$. The weights of ResNet50 that trained on ImageNet dataset is loaded as retraining for spatial stream CNN in order to allow transfer learning from image classification task to spatial stream CNN. Then, the last fully connected layer of the original ResNet50 is replaced with a new fully connected layer of one neuron with sigmoid as its activation function. The purpose of doing this is because the usual image recognition task has to classify 1000 classes, but the spatial stream CNN only need to classify violence and non-violence. Besides, all weights except for the final fully connected layer are set to be non-trainable to prevent overfitting during training.

On the other hand, for the temporal stream CNN configuration, the input for temporal stream CNN will be a stack for optical flow frames with a shape of $224 \times 224 \times 20$. The 20-channel optical flow frame stack is produced by stacking 10 dense optical flow frame pairs generated from 11 consecutive RGB frames. In order to allow transfer learning from image classification task to temporal stream CNN, ImageNet weights for ResNet50 trained on ImageNet dataset is loaded across modality to fit the shape of filter weights in the temporal stream CNN. The last fully connected layer of the original ResNet50 is also replaced with a new fully connected layer with one neuron and sigmoid as its activation function. For temporal stream CNN, none of the weights are set to non-trainable, thus the temporal CNN can learn new temporal features. At last, the prediction scores for both modalities are merged by late fusion.

2.6 Multimodal Data Fusion for Sensitive Scene Localization

This paper presents a new meta-learning late fusion approach that can be used to evaluate different sensitive content (either violence or pornography or other tasks). It allows the combination of time-overlapping snippets, as an attempt to identify the video content. The authors have proposed a late-fusion pipeline that can combine various snippet classifiers, even if they depend on different modalities of data like video, audio and so on (Moreira et al., 2019). Besides, the pipeline is built for general use, since it can easily be adapted to different kinds of sensitive material.

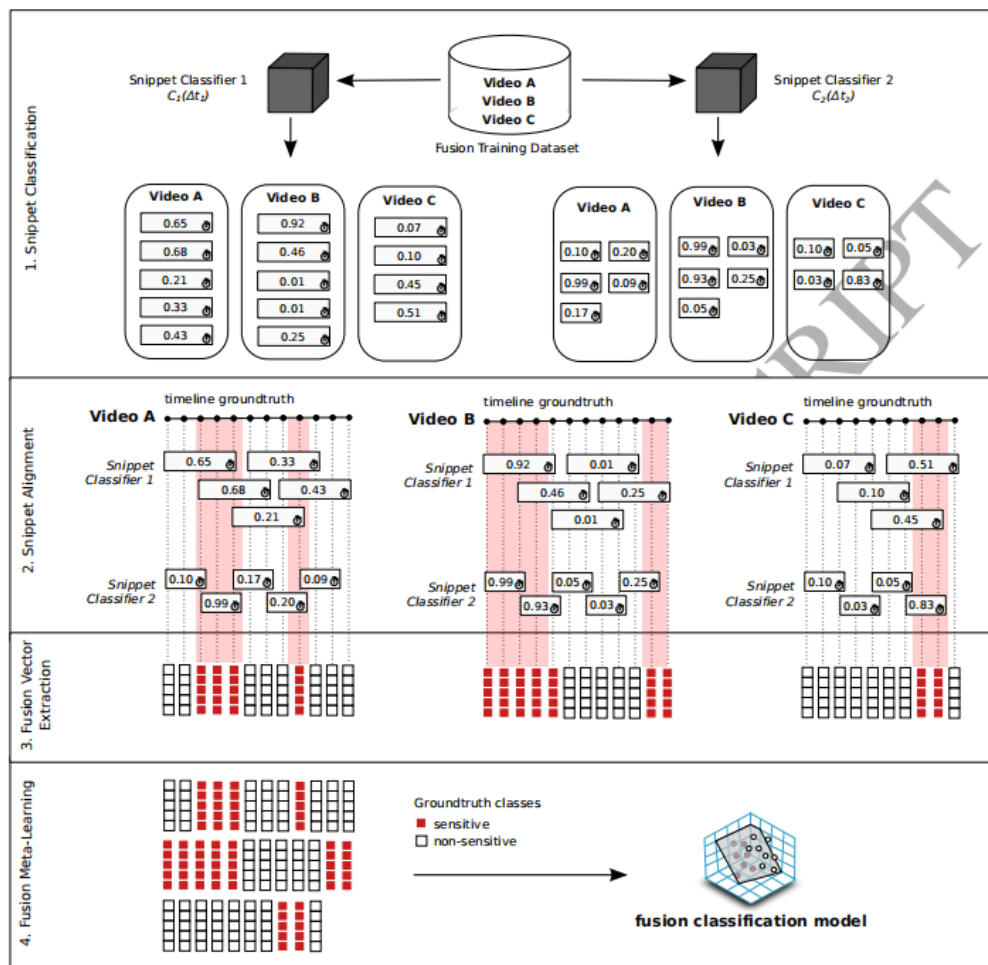


Figure 2.6.1: Proposed fusion training pipeline

For validation in this research, the localization capability of the pipeline for two of the most common kinds of sensitive content (pornography, violence) is analyzed. They validated important differences between the two sensitive content from the experiments. Audio is insignificant for localization of pornography and space-temporal features work as well as still-image features. This is because the audio aspect may be linked to the

abundance of pornographic amateur content in the dataset used, where audio streams have little to do with the visual content. The best method for the use of space-temporal compared to still-image features is combination of both of them. This is because they tend to be complementary in the pornographic case. In turn, audio is the key to enhance productivity for the localization of violence and space-temporal methods greatly surpass still-image approach.

In either case, for each scenario, the fusion pipeline could be nicely adapted, depending on the classification and fusion of multimodal time-overlapping video snippets. Nonetheless, the representatives of sensitive material are unheard other than pornography and violence and yet to be analyzed.

2.7 CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet

LeNet-5, a pioneering seven-level convolutional network that used in classification of digits, was applied by many banks in order to recognize hand-written numbers on cheques which digitized in 32x32 pixel greyscale input images. It requires larger and more convolutional layers to process a higher resolution images, thus This technique is limited by computing resources (Siddharth Das, 2017).

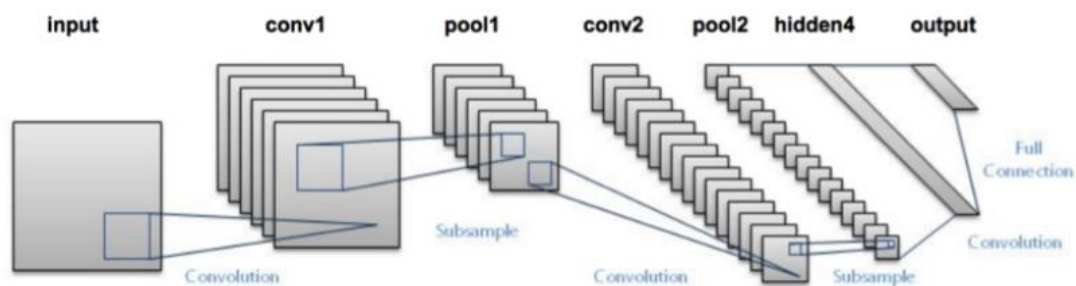


Figure 2.7.1: LeNet-5

AlexNet had an architecture somewhat similar to LeNet but deeper, and with more filters per layer, and with stacked convolutional layers. It composed of 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. After every convolutional and fully-connected layer, it will attach ReLU activations.

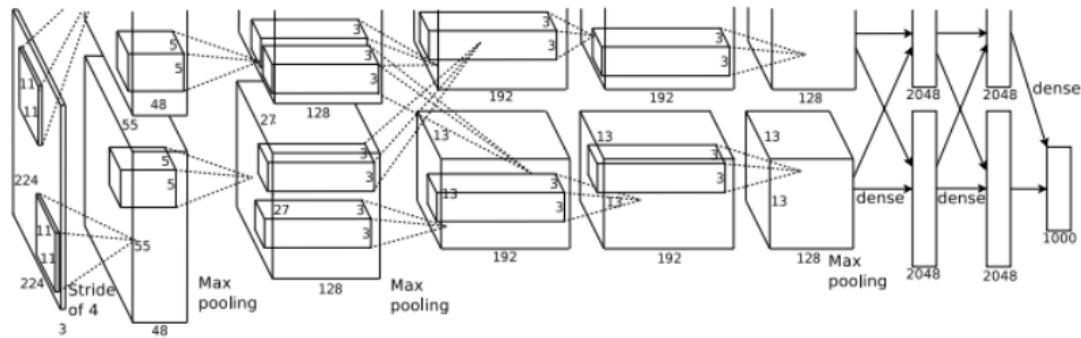


Figure 2.7.2: AlexNet

GoogLeNet (Inception v1) used a CNN that inspired by LeNet but with the implementation of a novel element called an inception module. It used batch normalization, image distortions and RMSprop. This module is based on many very small convolutions to decrease the number of parameters. Their architecture consisted of a 22 layer deep CNN and the number of parameters are reduced from 60 million (AlexNet) to 4 million.

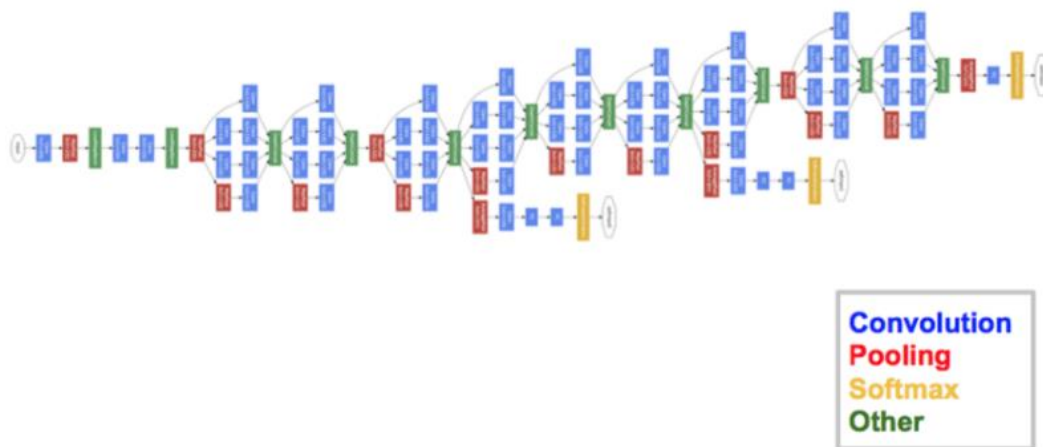


Figure 2.7.3: GoogLeNet

VGGNet consisted of 16 convolutional layers and its very uniform architecture made it highly attractive. It is quite similar to AlexNet, with only 3x3 convolutions, but lots of filters. Currently it is the community's most favoured option for image features extraction. The VGGNet's weight configuration is open to the public and has been used as a baseline feature extractor in many other applications and challenges. Even so, VGGNet contains 138 million parameters, which can be a bit difficult to manage.

CHAPTER 2 LITERATURE REVIEW

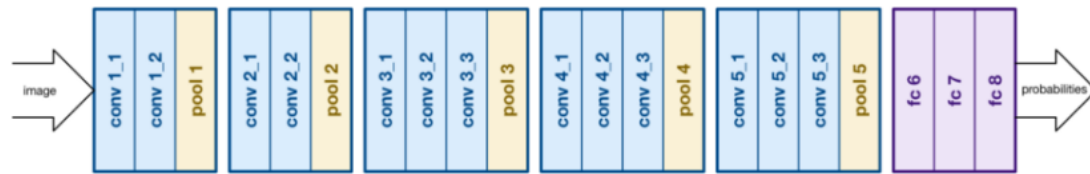


Figure 2.7.4: VGGNet

ResNet is a novel architecture with “skip connections” and features heavy batch normalization. The skip connections is to reduce the effect of vanishing or exploding gradient during training, by reusing activations from a previous layer until the adjacent layer learns its weights. Furthermore, skipping can simplify the network. For example, this technique can be used in training of a 152 layers neural network and still simpler than VGGNet.

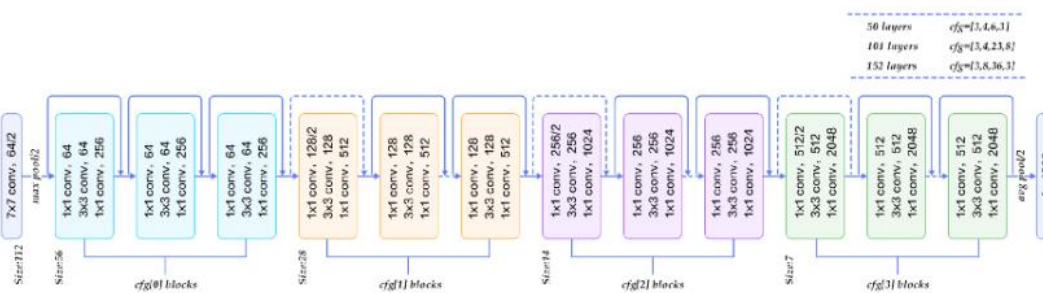


Figure 2.7.5: ResNet

CHAPTER 3- PROPOSED METHOD/APPROACH

3.1 SYSTEM DESIGN / OVERVIEW

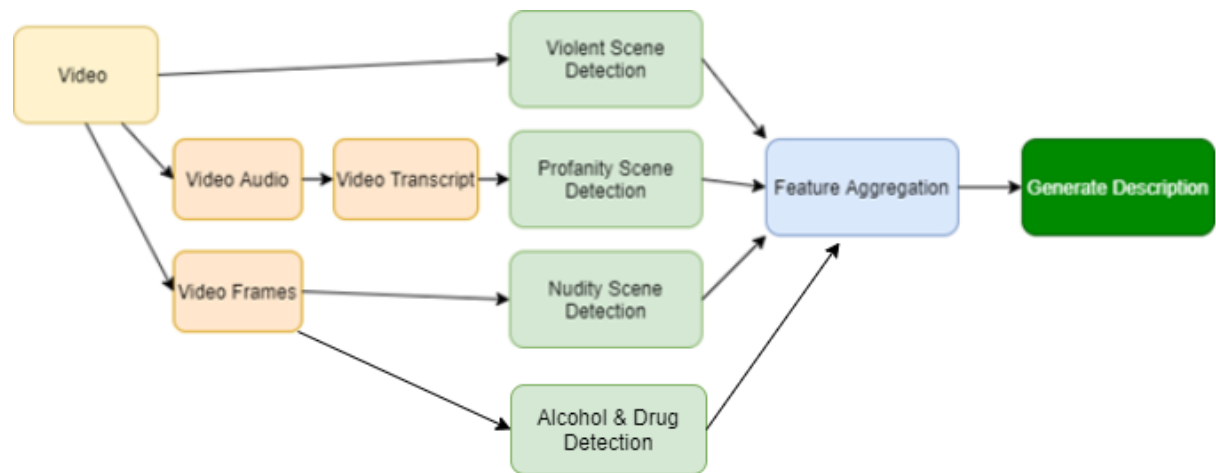


Figure 3.1.1 Proposed System Design

According to Figure 3.1.1, the pipeline / method is divided into a few sections. At first, a video will be inputted to the system. The video will be passed through the violent scene detector, profanity scene detector and also the nudity scene detector in order to identify whether the video contains these sensitive scenes. For the profanity scene detection, the video will be first converted into audio form (MP3 file) and further convert to transcript form (Text file). The transcript file will then send to the profanity scene detector to detect whether it contains any swear words. On the other hand, the video will be converted into several video frames for the violent scene detection, nudity scene detection and alcohol & drug detection. Firstly, the violence scene detector is trained by LSTM Neural Network. The nudity scene detector is based on Custom Scaled-YOLOv4 model. Moreover, the alcohol & drug detectors are trained with One Class CNN. After going through all the detectors, the result that obtained from each of the detector will be aggregated and filtered. Then, the description of the video will be shown in the form of sentences.

However, at this stage, the violent scene detector can only tell whether the video contains any violent scene but not describing the details of the violent scenes yet. Also, the prototype generated sentence only uses a template-based slot filling method without grammar checking.

3.1.1 Video Audio

The video audio is captured by converting the video from MP4 file to MP3 file. The python moviepy library can be used to convert MP4 to MP3 easily. In this project, we use the MP4 file to create a VideoFileClip object, then just get the audio object of this MP4 file, finally save the audio of this MP4 into MP3 file. The MP3 file will have to be further processed so that it can be used as the features of recognition.

3.1.2 Video Transcript

Spoken material is obtained from the video from the audio channel. Recognition of speech helps one to acquire more information about the video. The speech recognition can make the spoken word in natural language. In this project, after the MP4 file is converted into MP3 file. The MP3 file will then be stored into the Google Cloud Storage. After that, Google Speech-to-Text API service will be used to transcribe the speech into text as it is considered as a mature technology. The Google Speech-to-Text API service will then return a transcription of text for the video and store the text file into local machine.

3.1.3 Video Frames

The video frames are extracted from the video by OpenCV's function `cv2.VideoCapture(video)`, where *video* is the file path to the video file. The frames are captured at the frame rate and resolution of the video. For instance, if the duration of the video is 2s and its frame rate is 30, then 60 frames will be extracted from the video. The extracted frame will consist of 3 channels which are RGB channels. The captured frame will then store in a temporary file for further use.

3.1.4 Violent Scene Detection

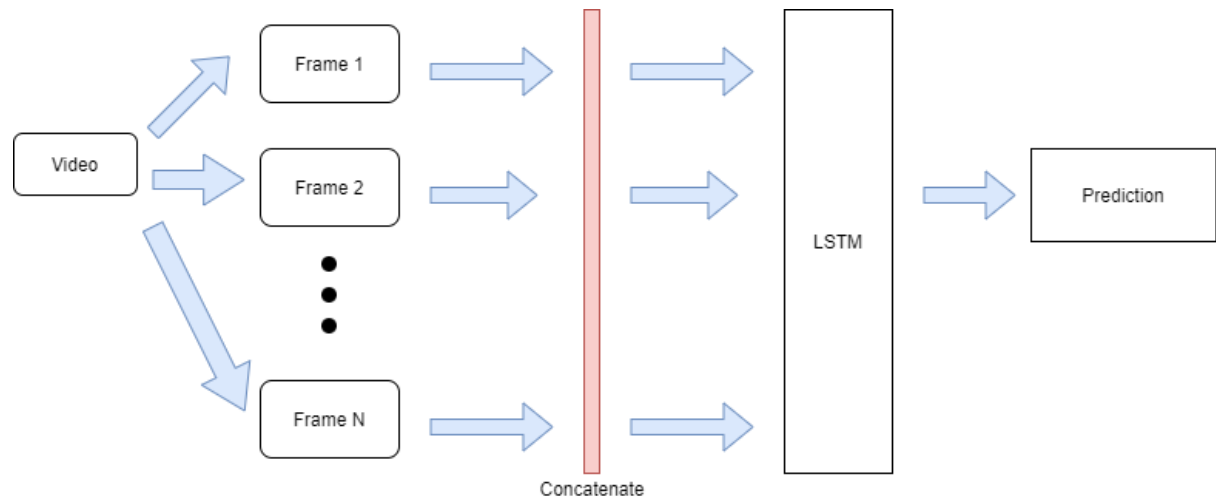


Figure 3.1.2 Proposed LSTM model

In this module, I downloaded the dataset video from the different resources. Firstly, the Peliculas.rar which can be downloaded at <http://academictorrents.com/details/70e0794e2292fc051a13f05ea6f5b6c16f3d3635/te ch&hit=1&filelist=1>. Second, the Movies.rar which can be downloaded at [www.cslab.openu.ac.il/download/violentflow/movies.rar](http://visilab.etsii.uclm.es/personas/oscar/FightDetection/index.html) - <http://visilab.etsii.uclm.es/personas/oscar/FightDetection/index.html>. Lastly, the HockeyFights.zip which can be downloaded at <http://visilab.etsii.uclm.es/personas/oscar/FightDetection/index.html>. Inside all these three file, there are a lot of short video that consists of violent video and non-violent video. The way of classifying them whether the videos are violent or not is through their name or folder name. For example, the file in HockeyFights is naming as fi1_xvid.avi. The 'fi' indicates that the video is containing the fight scene. Furthermore, for the Movies folder, there are two subfolders inside, one is NonViolence and another is Violence. The NonViolence folder is containing all the video that is classify as non-violent. My job is to combine all the videos and their label into a csv file for further processing. All the videos will be split into frames depending their resolution and frame rate and store into a folder for further training purposes.

In this project, the extracted frame is first resized to appropriate resolution and normalization will be performed to them. The purpose of doing this normalization is to speed up the training process. This normalization process is done by dividing all the

pixel values in the frame by 255 (maximum pixel number). Furthermore, all the extracted frames will be sorted to maintain the entire sequences of the data. Not only that, in order to make sure that each training set have an equal number of frame and to involve the temporal information, the time series data is grouped with the time step of 25. Long short-term memory(LSTM) which is an RNN architecture is used as the model to carry out classification, processing and prediction of the video scene. Before the data is passing into the LSTM model for the training purpose, the numpy array that are containing the pixel values of each frames is flatten and reshape into one dimensional array, which can fit the input shape of the LSTM input layer. The input size of our LSTM model is 25*12288 which the 12288 is the target size of our frames (64*64*3) and 25 is the time step. In our case, we only use one layer of LSTM followed by a SoftMax layer for the classification work. The reason is to prevent overfitting issue.

Model: "sequential_4"

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 10)	491960
dropout_6 (Dropout)	(None, 10)	0
dense_6 (Dense)	(None, 10)	110
dropout_7 (Dropout)	(None, 10)	0
dense_7 (Dense)	(None, 2)	22
Total params: 492,092		
Trainable params: 492,092		
Non-trainable params: 0		

Figure 3.1.3 Architecture of the violent scene detector model

For the video that are selected to carry out this violent scene detection, it should be split into numbers of frame and grouped into 25 frames per group. Therefore, to prevent that the extracted video frames cannot be divided completely by 25 frames, I wrote the function that are able to add the empty frame into the data so that the number of frames % 25 will equal to zero. The output from this model is in the form of array like:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

The 0 in the array indicate that there is no violent scene while the 1 indicate that there is a violent scene.

3.1.5 Profanity Scene Detection

A Python library (better-profanity 0.6.1) is used to check for profanity in strings. This library is significantly faster than the original one (profanity), as it uses string comparison instead of regex. Most of the words in the default word list are based on the full list of bad words and top swear words that banned by Google. There are some of the swear words that are inside the profanity_wordlist.txt.

```
1  2 girls 1 cup
2  anal
3  anus
4  areole
5  arian
6  arrse
7  arse
8  arsehole
9  aryan
10 asanchez
11 ass
--  .
```

Figure 3.1.4 Sample of swear words in profanity_wordlist.txt

The function `.contains_profanity()` will return True if there are any profanity words existing in the given string. There are some limitations in this library. One of the limitations is that the profanity checking could easily be bypassed by adding any character(s) to the word such as the word “jerkk off” will not be detected. However, this problem will not happen as the input string is the transcription from Google Speech-To-Text API, thus that is impossible to have such a weird word. In addition, this library is not supporting other language yet, such as Chinese.

This library provided the function of replacing each bad word with 4 asterisks. However, it cannot used to merely display the bad words. Thus, in order to show what is the profanity languages that are detected, I have written a function to compare the initial transcript and the transcript that has been censored. The differences between these two transcripts are the bad words. Thus, the detected bad words are to be now able to display.

3.1.6 Nudity Scene Detection

In this project, a custom Scaled-YOLOv4 detector was trained with the dataset that found at the https://github.com/notAI-tech/NudeNet/releases/download/v0/DETECTOR_AUTO_GENERATED_DATA.zip. There are a lot of pornography images inside the downloaded file. There is also an annotation file that already labelled the pornographic image (not all). The annotation file is consisted of the image name, xmin, xmax, ymin, ymax and label name such as EXPOSED_BELLY, EXPOSED_BUTTOCKS, EXPOSED_BREAST_F, EXPOSED_GENITALIA_F, EXPOSED_BREAST_M, EXPOSED_GENITALIA_M (6 classes). For those images that are unlabeled. I draw the bounding box and label them using Roboflow. If an image is detected as containing nudity scene, the custom YOLOv4 detector can detect the category of the nudity types and the coordinate of the bounding box. The labels and bounding box coordinates are displayed and stored in an array as a result. At the same time, the images that are detected as nudity scene will be drawn with the bounding box and the label and saved in the format of JPEG images for further use.

3.1.7 Alcohol & Drug Detection

There are two parts in this module: alcohol detection / drug detection. The two part that mentioned are one-class classification tasks. First of all, regarding to the datasets, I get both of the dataset for alcohol and drug from Bing. I have setup the Bing Search API in the Microsoft Azure Portal. The size of the dataset for alcohol and drug are 500 images and 300 images respectively. I also manually filtered the images of the datasets. After that, I do data augmentation for both datasets. I take the batch of images and apply a series of random transformations to every image such as resizing, shearing, random rotation, horizontal flips and translations.

In this module, I used One-Class Convolutional Neural Network architecture. The structure of the architecture that I proposed is consist of two parts. There are a feature extractor and a multi-layer perceptron. VGG-16 is chosen to extract feature from the images. On the other hand, the multilayer perceptron part is located after the feature

CHAPTER 3 PROPOSED METHOD/APPROACH

extractor. It gets the extracted features and perform classification on them. In the alcohol detector model, the output will be 1 if the image is alcohol while 0 means that the input sample is not alcohol. Same goes to the drug detector model.

In between the feature extractor and the multilayer perceptron, the embedded data samples (extracted feature) are ‘corrupted’ by adding some zero-centered gaussian noise. This allows us to combine feature extraction and classification in one step while only having one set of labels available. After that, these updated samples are batch-concatenated with their initial couples. As a result of this process, we have a duplicated batch of images made up of original samples (class 1) and corrupted samples (class 0). The purpose of doing this is to let the classification layers able to understand the difference and distinguish the real images among all.

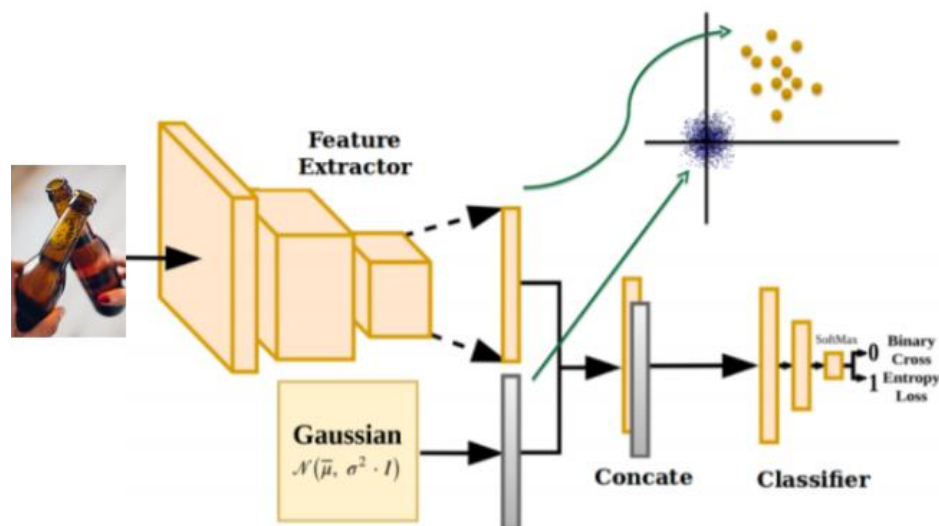


Figure 3.1.5 Schema of the proposed approach for alcohol & drug model

CHAPTER 3 PROPOSED METHOD/APPROACH

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	
gaussian_noise (GaussianNoise)	(None, 224, 224, 3)	0	input_2[0][0]
lambda (Lambda)	(None, 224, 224, 3)	0	gaussian_noise[0][0]
model (Functional)	(None, 4096)	117479232	lambda[0][0]
lambda_1 (Lambda)	(None, 4096)	0	model[0][0]
gaussian_noise_1 (GaussianNoise)	(None, 4096)	0	lambda_1[0][0]
lambda_2 (Lambda)	(None, 4096)	0	model[0][0] gaussian_noise_1[0][0]
activation (Activation)	(None, 4096)	0	lambda_2[0][0]
dense (Dense)	(None, 512)	2097664	activation[0][0]
dense_1 (Dense)	(None, 256)	131328	dense[0][0]
dense_2 (Dense)	(None, 128)	32896	dense_1[0][0]
dense_3 (Dense)	(None, 2)	258	dense_2[0][0]

=====
Total params: 119,741,378
Trainable params: 2,262,146
Non-trainable params: 117,479,232

Figure 3.1.6 Architecture for the alcohol detector model

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	
gaussian_noise (GaussianNoise)	(None, 224, 224, 3)	0	input_2[0][0]
lambda_1 (Lambda)	(None, 224, 224, 3)	0	gaussian_noise[0][0]
model (Functional)	(None, 4096)	117479232	lambda_1[0][0]
lambda_2 (Lambda)	(None, 4096)	0	model[0][0]
gaussian_noise_1 (GaussianNoise)	(None, 4096)	0	lambda_2[0][0]
lambda_3 (Lambda)	(None, 4096)	0	model[0][0] gaussian_noise_1[0][0]
activation (Activation)	(None, 4096)	0	lambda_3[0][0]
dense (Dense)	(None, 512)	2097664	activation[0][0]
dense_1 (Dense)	(None, 128)	65664	dense[0][0]
dense_2 (Dense)	(None, 2)	258	dense_1[0][0]

=====
Total params: 119,642,818
Trainable params: 2,163,586
Non-trainable params: 117,479,232

Figure 3.1.7 Architecture for the drug detector model

3.1.8 Feature Aggregation And Generate Description

The features that generated from the violent scene detector, profanity scene detector and nudity scene detector are then fused and filtered to create a sentence. A template-based slot filling method will be used according to linguistic semantic model. The model will overcome uncertainty in the sentence to make the sentence easier to understand and more meaningful.

3.1.9 GUI



Figure 3.1.8 GUI for the explicit scene detector

This is the GUI of this project. In order to make this GUI, Tkinter which is a standard Python GUI library is used. The reason why we choose this library is because it can create a GUI applications in an easy and fast way.

Firstly, the user should select a video to detect whether it contains any explicit scene. After the 'Browse' button is clicked, a file explorer will pop out and the user can go to any directory to choose the video. The selected video will then be copied into the video_selected folder for further use. Then, the users can click the buttons to run respective test such as VSD test, profanity test, nudity test and alcohol&drug test. The result for each test will finally be shown at the label beside the button.

CHAPTER 4 – SPECIFICATION AND PLANS

4.1 TOOLS TO USE

Hardware required to run:

(Most PC should be work)

Current Setup of my laptop:

Processor	Intel i5 7 th Gen
RAM	8GB
Storage	1TB
Graphic Card	Nvidia GeForce 940MX

Table 4.1.1: Current setup of my laptop

Software Requirement:

- Python 3.7.3
- OpenCV 4.1.0
- Pytorch 1.3.1
- Numpy 1.17.3
- Keras which is a high-level neural network API with TensorFlow as backend
- MoviePy (Python library for video editing)
- TensorFlow

PIP packages:

- better-profanity

External Packages:

<https://github.com/JunnYu/mish-cuda> (install mish activation function for cuda)

<https://github.com/roboflow-ai/ScaledYOLOv4.git> (clone Scaled_YOLOv4)

4.2 VERIFICATION PLAN

The purpose of this project is to detect the explicit scene such as violent scene, nudity scene, profanity scene and alcohol & drug scene as many as possible. The performance of this project can be evaluated by the confusion matrix, accuracy, precision, recall and f1 score of the scene detectors model.

First of all, a confusion matrix is a technique used in predictive analytics. It is, in particular, a table that shows and compares real values to the model's expected values. A confusion matrix is a metric used in machine learning to evaluate how well a machine learning classifier worked on a dataset. The precision, accuracy, specificity, and recall metrics are visualised using a confusion matrix. The confusion matrix is especially useful because, unlike other forms of classification metrics such as simple precision, it provides a more comprehensive image of how a model worked. With only one metric like accuracy can lead to a situation where the model repeatedly misidentifies one class, but this goes unnoticed because overall performance is good. The uncertainty matrix, on the other hand, compares various values such as False Negatives, True Negatives, False Positives, and True Positives. The meaning of True Positives (TP) is the frequency when the model correctly predicts the positive condition or class. True Negative (TN) is the frequency when the model predicts correctly the negative condition or class. False Positive (FP) is the frequency when the model incorrectly predicts a negative class or condition. False Negative (FN) is the frequency when the model incorrectly predicts the positive class or condition.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 4.2.1 Confusion Matrix

Second, accuracy in machine learning is known as the percentage of correct predictions out of all predictions made. Even though this appears to be adequate as a metric for a machine learning system's efficiency, but closer examination reveals that it is insufficient if the dataset contains unbalance number of positive and negative classes. The formula for calculating the accuracy is:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Third, precision is defined as the percentage of correct positive predictions among all cases that are classified as positive. The precision is used when we choose to concentrate mostly on false-positives, lowering the false-positive value while increasing precision. The precision can be calculated by the formula of:

$$Precision = \frac{TP}{TP + FP}$$

Fourth, recall is used to measure what is the actual positive label that is correctly predicted as positive. The recall is used when we want to concentrate more on FN, i.e., when we want to lower the false negative value while increasing the recall value. The formula to calculate the recall value is shown as:

$$Recall = \frac{TP}{TP + FN}$$

Lastly, F1-score is another good performance measure that takes into account both precision and recall. The F1-score is calculated by taking the 'Harmonic Mean' of precision and recall. Unlike precision, which is primarily concerned with false positives, and recall, which is primarily concerned with false negatives, the F1-score is concerned with both false positives and false negatives. The F1-score can be calculated with the formula:

$$F1 \text{ Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

CHAPTER 4 – SPECIFICATION AND PLANS

The violent scene detection, nudity scene detection, profanity scene detection and alcohol & drug detection will be tested with the testing dataset and some web-found videos for example the videos in Youtube for generalization task. The testing dataset is holding with its annotations and this can be used to determine the efficiency of the algorithms submitted. The videos that are found in Youtube are focusing on different domains and have their characteristics. For example, one of these videos is talking about the meaning of the “fuck” word or some bad words, the other video may contain some nudity scenes and violence scenes and another video may contain alcohol or drugs. The purpose of having the generalization set is to analyse how well the proposed algorithms generalize to video materials that vary from the training data.

CHAPTER 5 – EXPERIMENTS AND EVALUATIONS**5.1 EXPERIMENT SETUP****5.1.1 Violent Scene Detector**

Firstly, there are three datasets were combined for this task: Hockey Fight, Movies and Crowd Violence. This gave a total of 1446 videos, with 723 videos each violent and non-violent. The entire dataset was divided into training set and test set in the ratio 8:2. The training set is further divided into training set and validation set in the ratio 8:2. Image frames are extracted from these videos at their own frame rate. The images are then normalized by taking the images to divide by 255 before loading them into NumPy array with the target size of (64,64,3). The reason why I use the target size of (64,64,3) is to reduce the memory needed and speed up the training process. After that, the shape of images in training set and validation set is reshaped and flatten into 1-D array that can fit to the input layer of LSTM network. The input size of the LSTM model is (25,12288) where the 12288 is calculated by $64 \times 64 \times 3$ when the image array is reshaped to 1-D array and the 25 represents the time step. The time series data is then sent to the LSTM model for the training purposes. A SoftMax layer is implemented after the LSTM layer to carry out the classification task.

The model is trained based on the hyperparameter configurations that showed in Table below. The value of the hyperparameter will be fine-tuned during the model training phase to get an improved performance of the model.

Parameter	Value
Batch size	128
Epoch	50
Optimizer	Adam
Learning rate	1e-5
Loss Function	Categorical_crossentropy
Metric	Accuracy

Table 5.1.1 Parameter for violent scene detector model

5.1.2 Nudity Scene Detector

In this project, a custom Scaled-YOLOv4 detector was trained with the dataset that found at the https://github.com/notAI-tech/NudeNet/releases/download/v0/DETECTOR_AUTO_GENERATED_DATA.zip. There are a lot of pornography images inside the downloaded file. There is also an annotation file that already labelled the pornographic image (not all). The annotation file is consisted of the image name, xmin, xmax, ymin, ymax and label name such as EXPOSED_BELLY, EXPOSED_BUTTOCKS, EXPOSED_BREAST_F, EXPOSED_GENITALIA_F, EXPOSED_BREAST_M, EXPOSED_GENITALIA_M (6 classes). For those images that are unlabeled. I draw the bounding box and label them using Roboflow. Furthermore, I also uploaded the images followed by the annotation file to Roboflow. The Roboflow will help to draw the bounding box in the images according the coordinates in the annotation file.

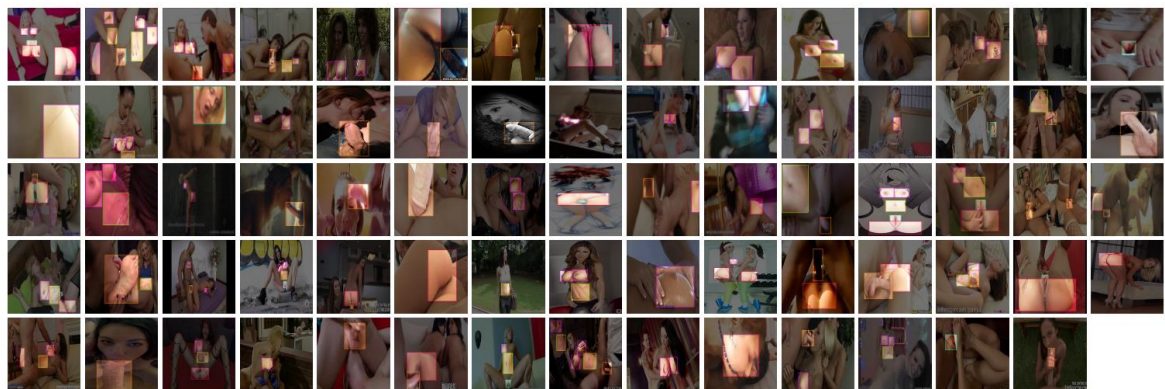


Figure 5.1.1 Sample of datasets that labelled by bounding box

It also helps to split the images into 70% training data, 20% validation data and also 10% testing data. The Roboflow can export the dataset in the format of YOLOv5 Pytorch so that it can be used to train the model. After that, we can start train the custom scaled-YOLOv4 Detector by passing several arguments such as input image size, batch size, epochs and etc.

Parameter	Value
Input image size	416x416
Batch size	16
Epochs	100

Table 5.1.2 Parameter for nudity scene detector model

5.1.3 Profanity Scene Detector

For the profanity scene detection, we will first convert the inputted video from MP4 file to MP3 file. The MP3 file will be converted to wav file. Not only that, due to the Google Speech to text API does not support the stereo audio files, thus it will convert the stereo audio file to mono file. In order to perform asynchronous request, the audio file will then be uploaded to Google cloud. The reason why we choose to use asynchronous request is because the duration of the audio file content can support up to 480 minutes (8 hours). Then, the transcribe function will perform all the operations necessary such as sending the audio file to the Google Speech-to-Text service to get the final transcripts. The transcripts will be stored in the 'transcript' variable. Once the Speech to text operation is completed, the final transcripts will be stored in a filepath for further use and the audio file that are uploaded to the Google cloud will also be deleted. Lastly, a python library (better-profanity 0.6.1) is used to detect whether the transcribe file contains any vulgar language that is existing in the profanity_wordlist.txt.

5.1.4 Alcohol & Drug Detector

Firstly, regarding to the datasets, I get both of the image dataset for alcohol and drug from Bing. I have setup the Bing Search API in the Microsoft Azure Portal. The size of the dataset for alcohol and drug are 500 images and 300 images respectively. I also manually filtered the images of the datasets by removing the images that are irrelevant. I have separated both of the datasets into 80% training data and 20% testing data. After that, I do data augmentation for the training data of alcohol and the training data of drugs. I take the batch of images and apply a series of random transformations to every image such as resizing, shearing, random rotation, horizontal flips and translations. The reason why I perform data augmentation is because it enables me to increase the size of the dataset and add variability of the dataset without actually collecting new data. In either case, the neural network considers these images as different images. Data augmentation also helps us to reduce over-fitting. Next, I used the `flow_from_dataframe()` in `ImageDataGenerator` class to read the images from the folders that containing the images of training data and testing data. The training dataset and the testing dataset are extracted into different folder. The `training_data` folder and `testing_data` should contain 2 folders each containing the images of respective classes. The directory structure of the dataset is shown in figure below.

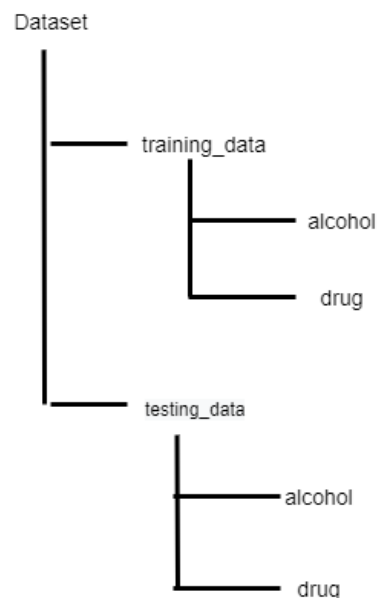


Figure 5.1.2 Directory structure of dataset file for alcohol & drug scene detector

CHAPTER 5 – EXPERIMENTS AND EVALUATIONS

In this module, I used One-Class Convolutional Neural Network architecture. The structure of the architecture that I proposed is consist of two parts. There are a feature extractor and a multi-layer perceptron. VGG-16 is chosen to extract feature from the images. On the other hand, the multilayer perceptron part is located after the feature extractor. It gets the extracted features and perform classification on them. In between the feature extractor and the multilayer perceptron, the embedded data samples (extracted feature) are ‘corrupted’ by adding some zero-centered gaussian noise. This allows us to combine feature extraction and classification in one step while only having one set of label available.

The model is trained based on the hyperparameter configurations that showed in Table below.

Parameter	Value
Verbose	2
Batch size	64
Epoch	20
Optimizer	Adam
Learning rate	1e-4
Loss Function	binary_crossentropy

Table 5.1.3 Parameter for alcohol & drug model

5.2 EVALUATION RESULTS

5.2.1 Violent Scene Detector

In order to evaluate the model's behavior, I have used the Matplotlib library to plot out the confusion matrix, accuracy chart and loss chart. According to the charts below, it is noticed that both the training and validation loss are considerably low while both the training accuracy and validation accuracy are high.

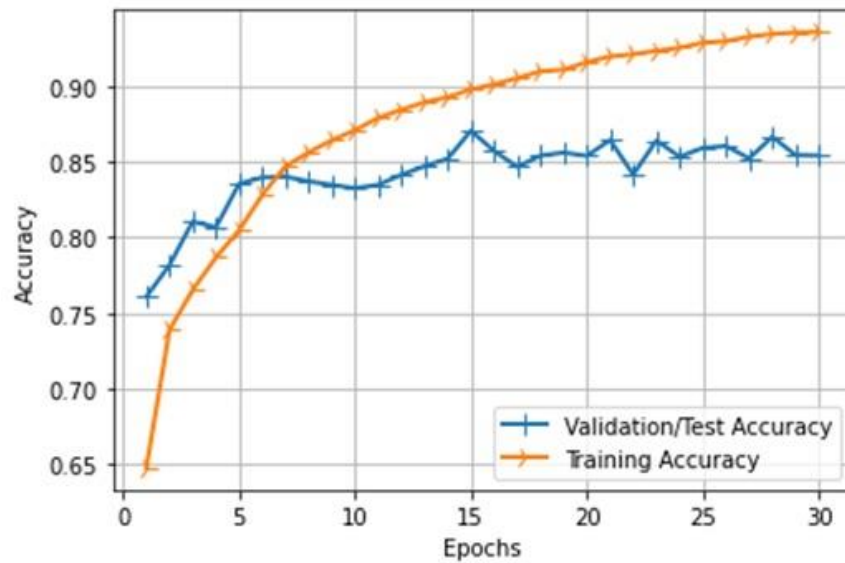


Figure 5.2.1 Accuracy chart for violent scene detector

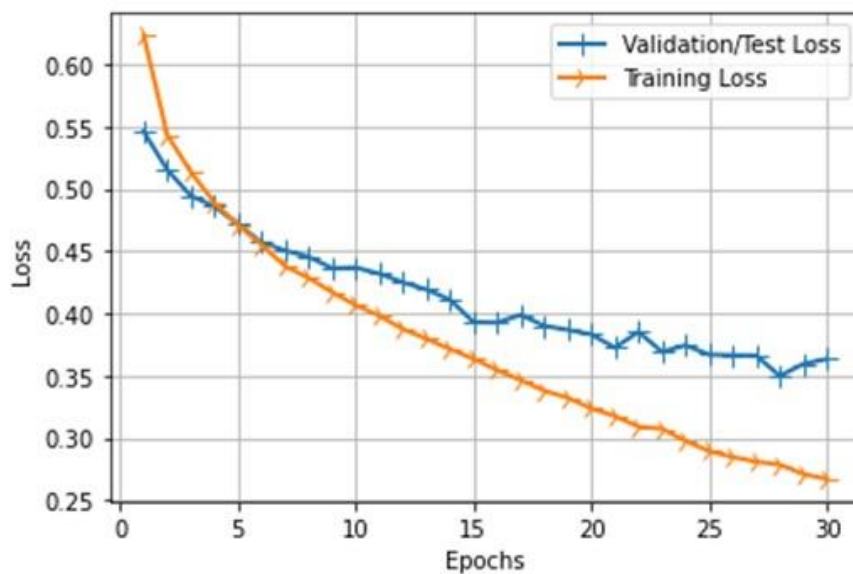


Figure 5.2.2 Loss Chart for violent scene detector

CHAPTER 5 – EXPERIMENTS AND EVALUATIONS

During the testing phase, the trained model is achieving an accuracy of 81.232%. From this accuracy, I can know that the trained model can well recognize the violent scene and non-violent scene. Furthermore, the precision value that I get is 81.309%. From this I can know that how accuracy that the violent scene is detected among all the predicted violent scene. On the other hand, the recall score of this model is 81.385%. This means I can know that how accuracy is the violent scene is detected among all the actual violent scene. Last, the F1-score of this model is 81.227%. It can give me a better measure of the incorrectly classified case than the accuracy metric.

Overall Accuracy	81.232
Overall Precision	81.309
Overall Recall	81.385
Overall F1 Score	81.227

Table 5.2.1 System performance of the violent scene detector model

From the confusion matrix of this model, I can see that there are 2952 non-violent scenes and 2839 violent scenes are predicted correctly. However, there are 806 non-violent scenes are misclassified as violent scene while 532 violent scenes are misclassified as non-violent scene.

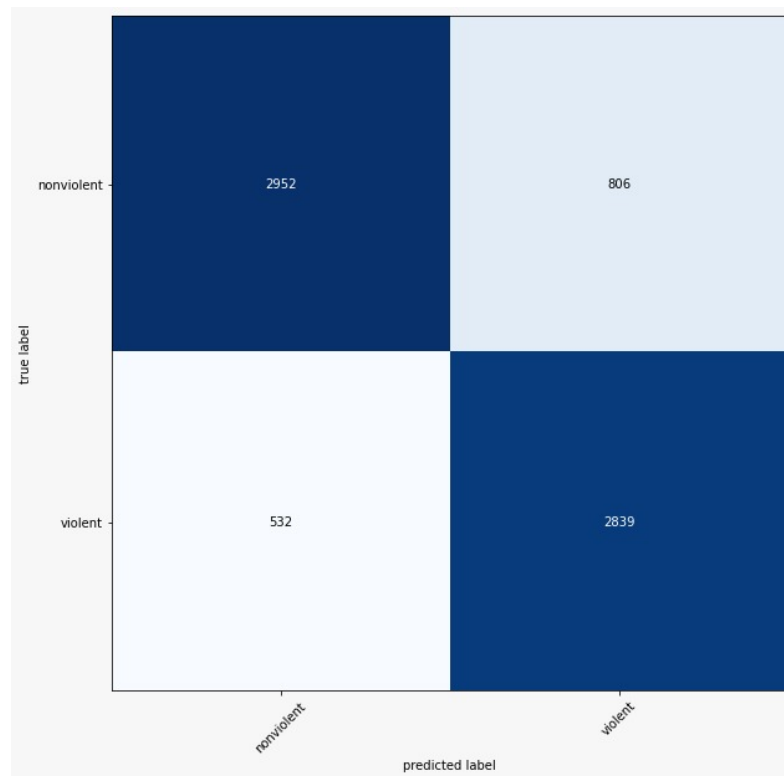


Figure 5.2.3 Confusion matrix for violent scene detector model

The table below show the performance metric for each of the class. The precision, recall and F1 score for each class have been calculated. From the table, it can notice that among all the metric of the non-violent class, the precision score is the highest value which is 0.8473. This value means that 84.73% of the non-violent scene is correctly predicted among all the scenes that predicted as non-violent scene. On the other hand, among all the metric of the violent class, the recall score is the highest value which is 0.8422. This value means that 84.22% of violent scene is correctly predicted among all the actual violent scene.

Class	Precision	Recall	F1 Score
Non-violent	0.8473020	0.7855242	0.8152444
Violent	0.7788752	0.8421833	0.8092930

Table 5.2.2 System performance for each class for violent scene detector model

5.2.2 Alcohol & Drug Detector

From the confusion matrix of this alcohol detection model, it can notice that the TP value is 0.94 and the TN value is 0.73. This means, among the testing data, 94% of the alcohol images are predicted correctly and 73% of the non-alcohol image are predicted correctly. Furthermore, the FP value of this model is 0.27 while the FN value is 0.06. This means, among the testing data, there are only 6% of alcohol images are misclassify as non-alcohol. However, there are 27% of non-alcohol images are incorrectly predicted as alcohol images.

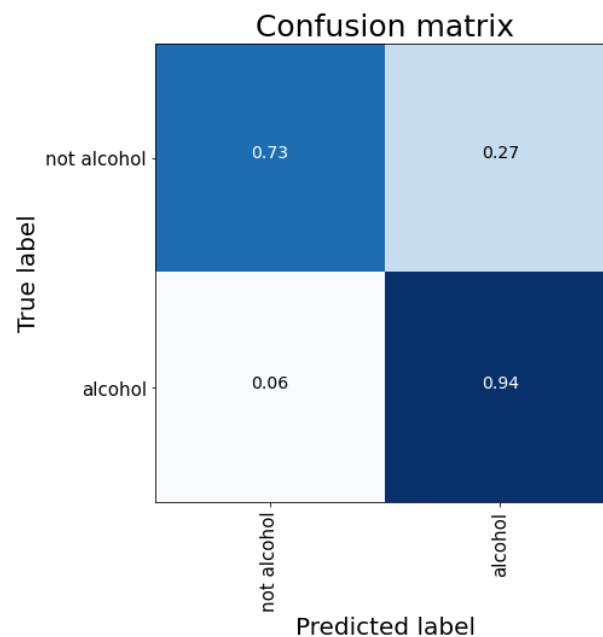


Figure 5.2.4 Confusion matrix for alcohol detector model

CHAPTER 5 – EXPERIMENTS AND EVALUATIONS

From the confusion matrix of this drug detection model, I can noticed that the TP value is 0.71 and the TN value is 0.80. This means, among the testing data, 71% of the drug images are predicted correctly and 80% of the non-drug image are predicted correctly. Furthermore, the FP value of this model is 0.20 while the FN value is 0.29. This means, among the testing data, there are 29% of drug images are misclassify as non-drug and 20% of non-drug images are incorrectly predicted as drug images.

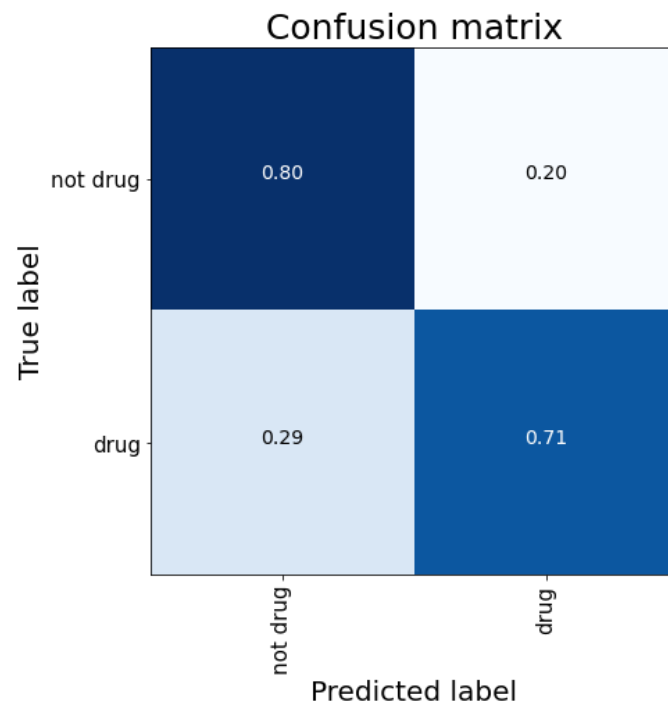


Figure 5.2.5 Confusion matrix for drug detector model

5.2.3 Nudity Scene Detector

From the Tensorboard metrics, the mAP (mean average precision) metric, precision metric and recall metric help to visualize the model. In this case, the obtained mAP_0.5 score achieved 0.43. The precision score and recall score of this model are 0.3 and 0.55 respectively. Honestly, the model obtained really low values of precision and recall score. In my opinion, the main reason that leads to this situation is because the small amount of dataset and there are 6 classes to classify.

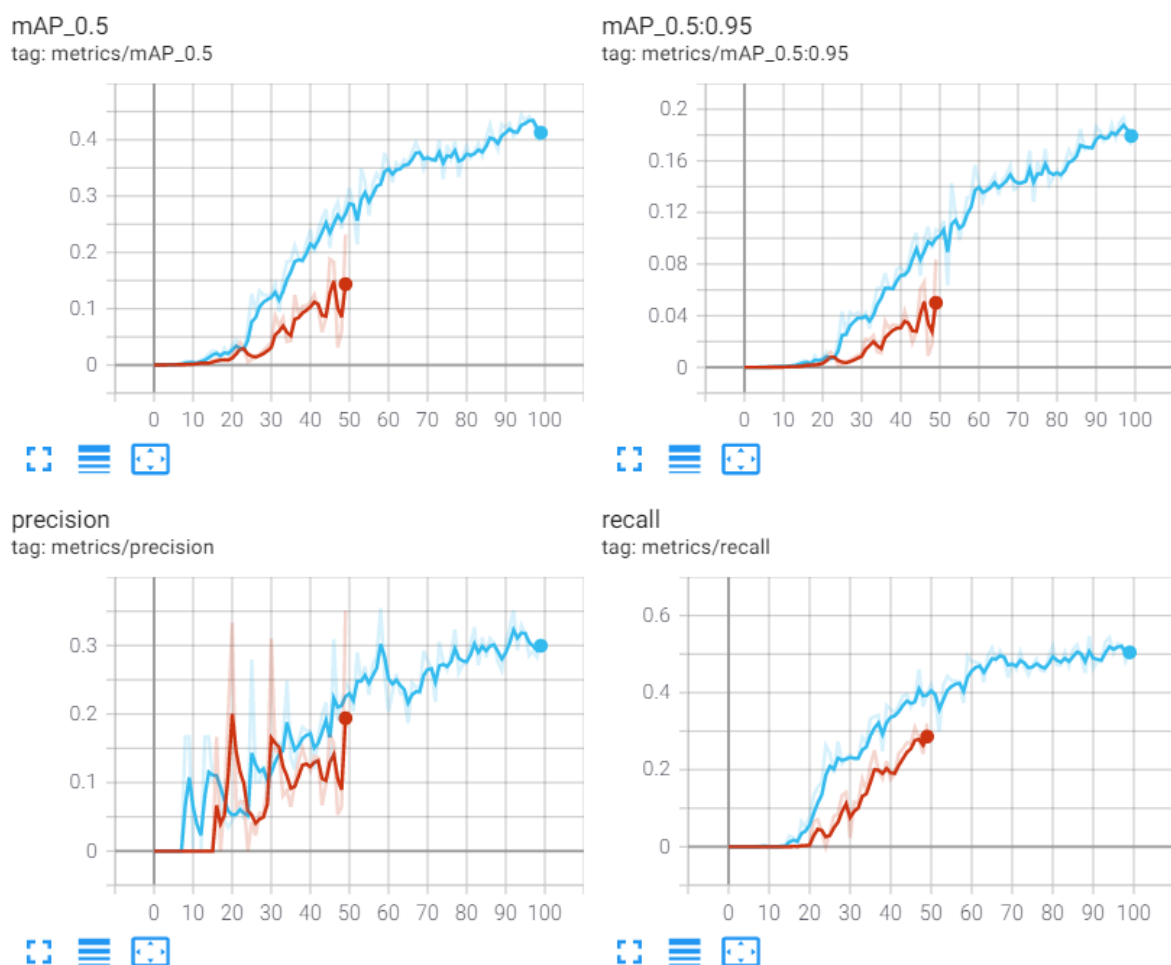


Figure 5.2.6 Tensorboard metrics for nudity scene detector model

5.3 IMPLEMENTATION EXAMPLES

In this section, the actual labels and the predicted labels will be shown in several video. The videos are downloaded from Youtube. The actual label/ ground truth of those video is annotated by three annotators:

First annotator: Male, 21 years old

Second annotator: Female, 53 years old

Third annotator: Male, 20 years old



Figure 5.3.1 Sample Video 1

Actual Label		Predicted Label	
Scene	Result	Scene	Result
Violent Scene	Not Exist	Violent Scene	Exist
Nudity Scene	Not Exist	Nudity Scene	Not Exist
Profanity Scene	Exist	Profanity Scene	Exist
Alcohol & Drug Scene	Not Exist	Alcohol & Drug Scene	Not Exist

Table 5.3.1 Result of Actual Label vs Predicted Label on sample video 1



Figure 5.3.2 Sample Video 2

Actual Label		Predicted Label	
Scene	Result	Scene	Result
Violent Scene	Exist	Violent Scene	Exist
Nudity Scene	Not Exist	Nudity Scene	Not Exist
Profanity Scene	Not Exist	Profanity Scene	Not Exist
Alcohol & Drug Scene	Not Exist	Alcohol & Drug Scene	Not Exist

Table 5.3.2 Result of Actual Label vs Predicted Label on sample video 2



Figure 5.3.3 Sample Video 3

Actual Label		Predicted Label	
Scene	Result	Scene	Result
Violent Scene	Not Exist	Violent Scene	Not Exist
Nudity Scene	Exist	Nudity Scene	Exist
Profanity Scene	Not Exist	Profanity Scene	Not Exist
Alcohol & Drug Scene	Not Exist	Alcohol & Drug Scene	Not Exist

Table 5.3.3 Result of Actual Label vs Predicted Label on sample video 3

CHAPTER 6 – CONCLUSION

6.1 OVERVIEW

Currently, the rapid growth of the Internet has led to an increase in the number of user-generated videos (UGVs). When the number of content creation increases and decentralized, content management and parental ratings become more challenging without a unified control mechanism. Amateur videos that are flooding the Internet are not properly screened and may contain explicit scenes that are not suitable for public viewing. The side effects from unhealthy exposures to violent, horror, profanity and explicit scenes can traumatize viewers and at the same time set up some domino effect in building bad characters from early ages. Therefore, this project aims to produce a system that can list out all the explicit scenes of a video or movie so that the viewers can roughly know what is going to play in the video.

In order to reach the aim, this project introduces a system involving a pipeline and model to extract semantic information from the sensitive scene in the video including the visual and audio cues. The data would then be encoded and transcribed into a sentence to describe the content of the video. The project can also automatically classify the selected video based on the results obtained from several sensitive scene detector.

There are some challenges during the implementation and development of the model. The dataset of this violent scene model is just containing the video that the people is fighting. Thus, the violent scene detector only can recognize the scene of people fight to each other. However, in real world, there are many other scenes such as blood, gunshot that can be classified as violent scene.

Secondly, the Google Speech Recognition API sometimes cannot return the converted text of entire audio. The reason is because there are some particular video sample has loud background music, which will interfere with the speech recognition. Not only that, sometimes the converted text is not completely accurate. It may affect by many factors like the language spoken and pronunciation of word in video. These factors may restrict the detection's accuracy and can lead to a misclassification. Thus, better methods for minimizing the detection error may need to be investigated.

6.2 FUTURE WORKS

In this project, there are still a lot of parts that can be improved such as the performance of different models. Firstly, the components (models) in the pipeline can always be replaced by a better and more efficient one. Not only that, the performance of the pipeline can also be improved by carrying out optimization. For example, more datasets will be added into the training process so that the performance can be boosted. Furthermore, more different settings also will be tested by tuning those hyperparameter (batch size, learning rate, weight decay) or adjusting the depth for the neural network.

Bibliography

1. Aafaq, N., Mian, A., Liu, W., Gilani, S.Z. and Shah, M. (2020). Video Description. *ACM Computing Surveys*, 52(6), pp.1–37.
2. Cerliani, Marco. "One-Class Neural Network in Keras." *Medium*, Towards Data Science, 2 Nov. 2020, towardsdatascience.com/one-class-neural-network-in-keras-249ff56201c0.
3. C. M. Patil, B. Jagadeesh and M. N. Meghana, "An Approach of Understanding Human Activity Recognition and Detection for Video Surveillance using HOG Descriptor and SVM Classifier," *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Mysore, 2017, pp. 481-485.
4. Dai, Q., Wu, Z., Zhao, R.W., Wu, Z., Wang, X., Gu, Z., Wu, W. and Jiang, Y.G..(2015). Fudan-Huawei at MediaEval 2015: Detecting Violent Scenes and Affective Impact in Movies with Deep Learning. In *MediaEval*.
5. Demarty, C.-H., Penet, C., Soleymani, M. and Gravier, G. (2014). VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation. *Multimedia Tools and Applications*, 74(17), pp.7379–7404.
6. Donahue, J., Hendricks, L.A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K. and Darrell, T. (2017). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] 39(4), pp.677–691. Available at: <https://arxiv.org/pdf/1411.4389.pdf> [Accessed 7 Sep. 2019].
7. Fitzpatrick, C. (2019). *Watching violence on screens makes children more emotionally distressed*. [online] The Conversation. Available at: <https://theconversation.com/watching-violence-on-screens-makes-children-more-emotionally-distressed-106757>.
8. Fotache, C. (2019). *Object detection and tracking in PyTorch*. [online] Medium. Available at: <https://towardsdatascience.com/object-detection-and-tracking-in-pytorch-b3cf1a696a98> [Accessed 15 Apr. 2020].
9. Google Cloud. (n.d.). *Cloud Speech-to-Text - Speech Recognition*. [online] Available at: <https://cloud.google.com/speech-to-text> [Accessed 15 Apr. 2020].

Bibliography

10. Joseph Redmon, A. F., 2018. *YOLOv3: An Incremental Improvement*, Washington: University of Washington.
11. Moreira, D., Avila, S., Perez, M., Moraes, D., Testoni, V., Valle, E., Goldenstein, S. and Rocha, A. (2019). Multimodal data fusion for sensitive scene localization. *Information Fusion*, 45, pp.307–323.
12. PyImageSearch. (2019). *Video classification with Keras and Deep Learning*. [online] Available at: <https://www.pyimagesearch.com/2019/07/15/video-classification-with-keras-and-deep-learning/> [Accessed 15 Apr. 2020].
13. Ranjay Krishna, K. H. F. R. L. F.-F. J. C. N., 2017. *Dense-Captioning Events in Videos*, s.l.: Computer Vision Foundation.
14. Redmon, J. (2012). *YOLO: Real-Time Object Detection*. [online] Pjreddie.com. Available at: <https://pjreddie.com/darknet/yolo/>.
15. Solawetz, Jacob. “How to Train Scaled-YOLOv4 to Detect Custom Objects.” *Medium*, Towards Data Science, 7 Jan. 2021, towardsdatascience.com/how-to-train-scaled-yolov4-to-detect-custom-objects-13f9077ebc89.
16. Siddharth Das (2017). CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more. [online] Medium. Available at: <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>.
17. Yew (2019). Violent Scene Detection In Videos.[online] Available at: <http://eprints.utar.edu.my.libezp2.utar.edu.my/3494/1/CS-2019-1507160-1.pdf>

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 2
Student Name & ID: CHAI ZI XU 17ACB05093	
Supervisor: DR AUN YICHIE	
Project Title: Automatic Parental Guide Ratings for Short Movies	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Meeting with supervisor to discuss the overall project process

2. WORK TO BE DONE

Need to online search for recent similar work done by others

3. PROBLEMS ENCOUNTERED

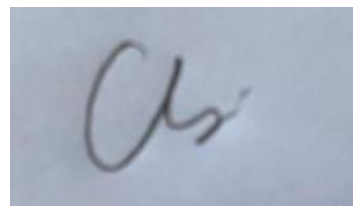
It takes time to find similar work done by others

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 4
Student Name & ID: CHAI ZI XU 17ACB05093	
Supervisor: DR AUN YICHIE	
Project Title: Automatic Parental Guide Ratings for Short Movies	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Installed required software and libraries for Python development environment

2. WORK TO BE DONE

- i. Build the model
- ii. Collect dataset

3. PROBLEMS ENCOUNTERED

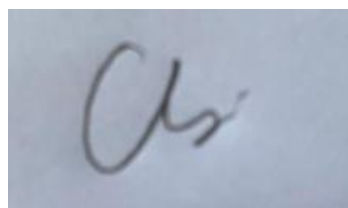
- i. Library dependencies and version conflicts.
- ii. Lack of computational power.

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 6
Student Name & ID: CHAI ZI XU 17ACB05093	
Supervisor: DR AUN YICHIET	
Project Title: Automatic Parental Guide Ratings for Short Movies	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- i. Updated FYP report structure, revised content for Chapter 1 & 2.

2. WORK TO BE DONE

- i. Build another model

3. PROBLEMS ENCOUNTERED

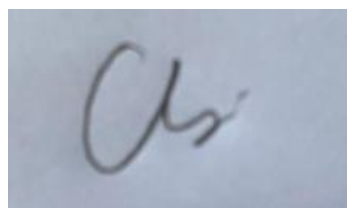
- i. Dataset size too large, and found difficult to use the dataset

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 11
Student Name & ID: CHAI ZI XU 17ACB05093	
Supervisor: DR AUN YICHIE	
Project Title: Automatic Parental Guide Ratings for Short Movies	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

ii. Every model has been done

2. WORK TO BE DONE

i. Write Report

3. PROBLEMS ENCOUNTERED

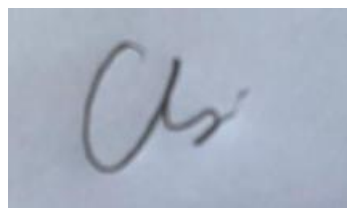
-

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.



Supervisor's signature



Student's signature

APPENDIX B- POSTER



FACULTY OF INFORMATION AND
COMMUNICATION TECHNOLOGY

AUTOMATIC PARENTAL GUIDE RATINGS FOR SHORT MOVIES

INTRODUCTION

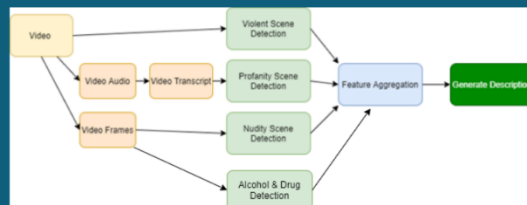
Video description is helpful for automatic movie ratings and annotating parental guides. However, human-annotated ratings are somewhat ambiguous depending on the types of movies and demographics. This project proposes a Machine-learning (ML) pipeline to generate a parental rating for short movies automatically.

Implementation Result



Actual Label		Predicted Label	
Scene	Result	Scene	Result
Violent Scene	Exist	Violent Scene	Exist
Nudity Scene	Not Exist	Nudity Scene	Not Exist
Profanity Scene	Not Exist	Profanity Scene	Not Exist
Alcohol & Drug Scene	Not Exist	Alcohol & Drug Scene	Not Exist

System Design



APPENDIX C PLAGARISM CHECK RESULT

APPENDIX C PLAGARISM CHECK RESULT

feedback studio

CHAI ZI XU | Automatic Parental Guide Ratings for Short Movies

-- /0 ?

CHAPTER 1 INTRODUCTION

The proliferation of smart devices has enabled anyone with access to technology to be content creators. In modern days, the momentum of video shooting and movie making has shifted from a studio-centric to crowd-source content creation. Not only that, the rapid growth of the Internet has led to an increase in the number of user-generated videos (UGVs). When the number of content creation increases and decentralized, content management and parental ratings become more challenging without a unified control mechanism. Amateur videos that are flooding the Internet are not properly screened and may contains explicit scenes that are not suitable for public viewing. The side effects from unhealthy exposures to violent, horror, profanity and explicit scenes can traumatize viewers and at the same time set up some domino effect in building bad characters from early ages. For instances, children who were exposed to violent films displayed more signs of emotional distress, in terms of depression and lack of excitement(Fitzpatrick, 2018). Therefore, to overcome this problem a violent

Match Overview

15%

1	www.yugangjiang.info	2%
2	Chandrashekar M Patil,...	1%
3	Daniel Moreira, Sandra ...	1%
4	github.com	1%
5	Nayyer Aafaq, Ajmal Mi...	1%
6	Submitted to University...	1%

Turnitin Originality Report

Document Viewer

Processed on: 15-Apr-2021 19:40 +08
ID: 1559882764
Word Count: 8627
Submitted: 1

Automatic Parental Guide Ratings for Short Mo... By CHAI
ZI XU

Similarity Index	Similarity by Source	
15%	Internet Sources:	9%
	Publications:	12%
	Student Papers:	7%

include quoted	include bibliography	excluding matches < 8 words	mode: quickview (classic) report	Change mode	print	download
2% match (Internet from 16-May-2019) http://www.yugangjiang.info						
1% match (publications) Chandrashekar M Patil, B Jagadeesh, M N Meghana. "An Approach of Understanding Human Activity Recognition and Detection for Video Surveillance using HOG Descriptor and SVM Classifier", 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017						
1% match (publications) Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, Anderson Rocha. "Multimodal data fusion for sensitive scene localization", Information Fusion, 2019						
1% match (publications) Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, Mubarak Shah. "Video Description", ACM Computing Surveys, 2019						
1% match (student papers from 11-Dec-2019) Submitted to Cyprus International University on 2019-12-11						

APPENDIX D TURNITIN FORM

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	CHAI ZI XU
ID Number(s)	17ACB05093
Programme / Course	CS
Title of Final Year Project	Automatic Parental Guide Ratings for Short Movies

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>15</u> % Similarity by source Internet Sources: <u>9</u> % Publications: <u>12</u> % Student Papers: <u>7</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

APPENDIX D TURNITIN FORM



Signature of Supervisor

Name: Dr Aun Yichiet

Date: 15/04/2021

Signature of Co-Supervisor

Name:

Date:

APPENDIX E CHECKLIST



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

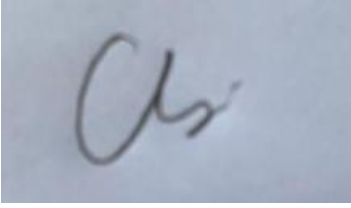

Student Id	17ACB05093
Student Name	CHAI ZI XU
Supervisor Name	DR. AUN YI CHIET

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.
--	---

APPENDIX E CHECKLIST

 (Signature of Student) Date:15/04/2021	 (Signature of Supervisor) Date:15/04/2021
--	--