Inappropriate Scene Detection in a Video Stream

Thesis submitted in partial fulfilment of the requirement for the degree of

Bachelor of Computer Science and Engineering

Under the supervision of

Dr. Jia Uddin

By

Kamrun Nahar Tofa (14101063)

Farhana Ahmed (14101069)

Arif Shakil (14101031)

School of Engineering and Computer Science

14th December 2017

BRAC University, Dhaka, Bangladesh

DECLARATION

We hereby declare that this thesis is based on results obtained from our own work. All the materials that were used for the purpose of completing this thesis are duly acknowledged and mentioned in reference. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor:

_____

Dr. Jia Uddin,
Assistant Professor,
Department of Computer Science and Engineering,
BRAC University, BRAC

Signature of Authors:

_____

Arif Shakil, 14101031

_____

Farhana Ahmed, 14101069

_____

Kamrun Nahar Tofa, 14101063

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network to Detect Object
ConvNets	Convolutional Neural Network to Detect Object
FCN	Fully Connected Network
MLP	Multi-Layered Perception
FCL	Fully Connected Layer
HoG	Histogram of Gradients
SVM	Support Vector Machine

ACKNOWLEDGEMENT

We would like to begin by thanking the Almighty Allah, the creator, the owner of the universe, the most merciful, beneficent and the most gracious, for giving us the power and will to initiate and complete our research and put our best into it to conclude the paper successfully.

Next, we would like to express my deep gratitude to Dr. Jia Uddin sir, our thesis supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. We would also like to thank Dr. Jia Uddin sir, for his critical advises and assistance in keeping our progress on schedule. My grateful thanks are also extended to the BRAC University Faculty Staffs of the Computer Science and Engineering department, who have been constantly acting as our guidance throughout the study period at BRAC University, especially in building and enhancing our base in education and knowledge.

Finally, we would like to express our gratitude and gratefulness to our beloved parents, brothers, sisters for their love and care. We are grateful to all of the participants who helped us directly, or indirectly in completing our research work.

ABSTRACT

In this paper, we attempted to propose a model to detect inappropriate scenes, such as nudity, weapons, danger, drugs, gore, etc. in any video stream. The detection part is divided into different steps. The very first step of the proposed model is to convert the video stream into its frames. After fragmentation is completed, the image detection algorithms are used on the individual extracted frames to detect our required inappropriate scenes. For the detection part, we decided to fuse three different algorithms to detect the required inappropriate scenes. The nude detection part is handled using two algorithms. The first algorithm would find human figures on the fragmented frames and if found, the human figures would be cropped out and a separate image file would be formed containing the cropped-out part only. Next, the second algorithm would use this cropped image to find the presence of nudity in the original uncropped image. To detect dangerous objects like knives, guns, swords and detect gore, bloody scenes in the fragmented frames, the object detection algorithm is used. For object detection, we used CNNs Object Detection [1,2] algorithm to detect objects and scenes, and for nudity detection, we have used nudepy library from python which is based on detecting skin-colored pixels and identify nudity based on pixel count and its region [18]. After successful detection via the two algorithms, the model is going to give an output to show the percentage of how much of violent scenes and nudity is present in the video sequence. Furthermore, the model will also be able to determine and classify the video as pornography if the percentage nudity detected is over our base scale of what percentage of nudity in a video may be considered as a pornography.

CHAPTER 01
INTRODUCTION

The sole purpose of developing this model of detecting inappropriate scenes in any video stream is from the realization that with the rapid growing digital media, usage of videos and images are becoming more and more common. Sometimes the videos are resourceful, but one might not prefer to watch it because of containing inappropriate scenes like adult scenes, nudes or any crime scene. Furthermore, many may prefer to know the proportion of the inappropriate contents in the chosen media before watching it or presenting it somewhere. Thus, our scheme will help an individual detect whether the video contains any unwanted scene prior to watching it.

1.1 Motivation

Development of this scheme is quite necessary for children or even teenagers as they have started using internet quite a lot. They may come across documentaries or movies on internet that they shouldn't watch. Like many parents doesn't want children to see a fighting scene like a man killing another man with a knife. Yet he might end up watching it in any movie as this type of scenes started to become normal. Therefore, having this model will help parents have better control over the videos their children watch.

Also, Autistic people or people with mental illness may not be able to tolerate certain violence and thus this system will be of good use to them as well.

Moreover, people may not prefer watching a nude scene, adult scene or a crime scene due to personal preferences or even religious reasons. Thus, this system will be of help to them as well.

For the above given reasons, we came up with a proposed model to tackle and give the viewers an idea of what to expect in a given video.

1.2 Contribution Summary

In past, there had been research work on developing algorithms that would detect pornography in a video or an algorithm that would detect nudes from an image or even a video clip. Experimental

work on detecting crime scenes, gore scenes or any certain inappropriate scene had been done as well. However, all of these algorithms focused on one particular aspect only. Inappropriate scene detection model is a system that defines inappropriate scenes on a wide range of spectrum. From pornography, to small durational adult scenes in any movie/video/documentary, to just a nude scene the definition of inappropriate ranges upon criminal scenes, gore scenes, murder scenes, extreme violence, showing of dangerous weapons, etc. The convolutional neural network that had been used in detecting particular objects or scenes can range and vary according to the preference of individual users. Thus, this system will provide better detection over a wide range of scenes, making it more useful and handy.

Also, previously many work on nude detection had been done mostly on the basis of skin pixel detection and then relative analysis of skin pixel with non-skin pixel to conclude whether the scene have nudes or not. However, to increase the accuracy of proper detection, our system ensures that human bodies are perfectly identified and separated from the original picture into a different image containing only the human figures before performing skin detection algorithm on it. This greatly enhances the performance of nude detection as now even a small nude portion in a large image can be identified and handled. Hence the combination makes our nude detection algorithm much more efficient and effective.

1.3 Thesis Outline

- Chapter 02 contains background study that provides the overall work flow, stating the algorithms and techniques used.
- Chapter 03 discusses Methodology that gives a detailed description of every step of this model along with implementation details.
- Chapter 04 shows the results of all the algorithms used at each step of all the algorithms used.
- Chapter 05 analyses the results.
- Chapter 06 concludes the paper while stating all the limitations faced and also discusses about future opportunities from this model.

Chapter 02

BACKGROUND STUDY

2.1 Fragmentation of video

The very first step is to convert the video file into its individual frames. In order to do that, we used MATLAB and its video reader and image reader libraries to convert the video to frames.

First, we take a video file as an input and then make a directory with the corresponding name of the video file. After this, we extract the number of frames of the video using the video reader and store it to run a loop to extract each frame from the video. On extraction of each frame, we store it in the recently created directory. The filename of each of these extracted frames corresponds to their frame number in the video. After reading all the frames of the video, we end the MATLAB simulation and move on to our next step of the model. The above process is presented in the flow chart in Fig. 1. And, a sample simulation has been shown in Fig. 2.

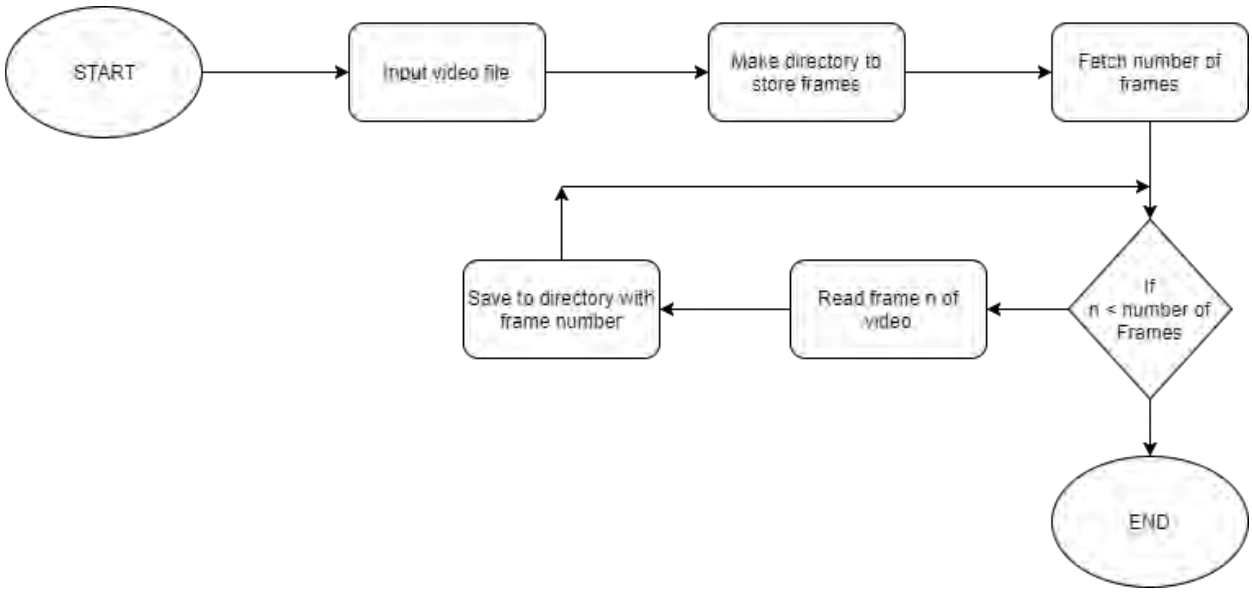


Fig. 1 – Video to frames fragmentation process.

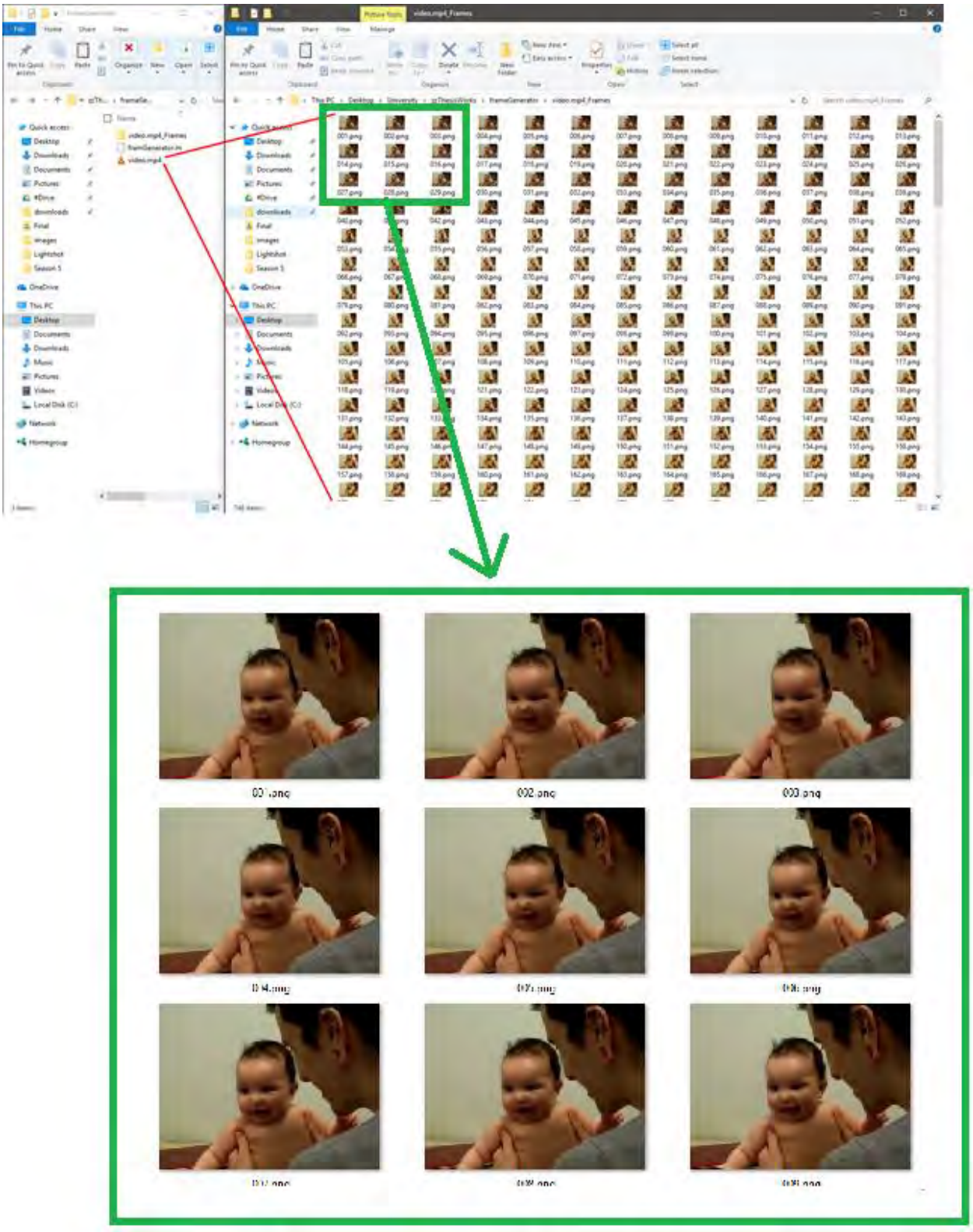


Fig. 2 – Sample simulation of video to frames fragmentation

2.2 Nude Detection

The nude detection part is implemented with the combined use of two algorithms. The first algorithm mainly detects human figures from the fragmented frames and crop them out in a separate image file. Then the second algorithm, that is the nude detection algorithm, is used on the new image files to determine if the images are nudes or not.

2.2.1 Detecting Human Figures

For human detection we used an open source library, OpenCV. OpenCV has HoG descriptor and SVM classifier which can be used to detect humans. Firstly, the image is resized because their aspect ratio needs to be 1:2 for finding out the HoG descriptors. Next, the image is divided into 8 x 8 cells (see figure 03(a)) and then the HoG (histogram of oriented gradient) of each pixel are calculated in each cell. Gradient contains two values, the direction and the magnitude. The gradients are then stored in a 9-bin 1D array/vector and this is called the histogram of oriented gradient. The histograms are then normalized over a 16 x 16 block (see figure 03(b)) by concatenating the vectors of the block into a 36 x 1 vector. This is done so that light intensity does not become a factor in calculating the gradients. Finally, the 36 x 1 vectors are concatenated into one vector to calculate the final feature vector. In Fig. 3(c), the Hog descriptor is shown.

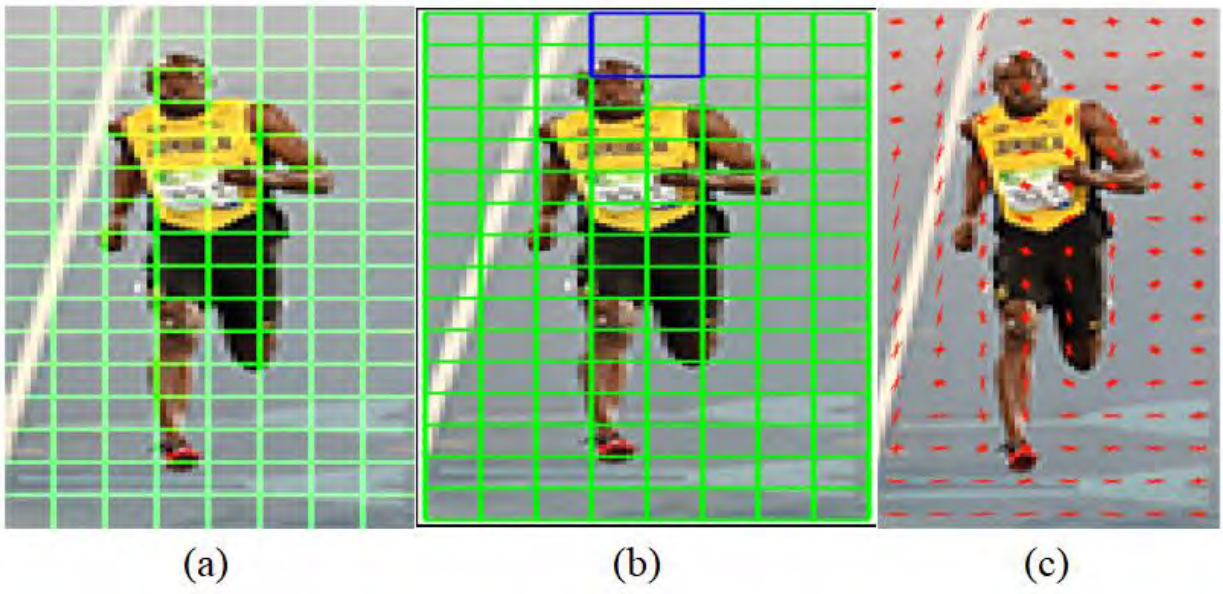


Fig. 3 – (a)Cells of HoG (b) Block used for normalization (c) HoG descriptor [19]

The classification is done by training a SVM (Support Vector Machine) classifier [5]. The SVM classifier used in OpenCV is pretrained with the human HoG features. The classification of different objects in SVM is done by an optimal hyperspace in the feature space [6], in this case the feature space is the final feature vector resulting from the HoG detector. Linear SVM finds the maximum distance between the multidimensional vectors of two classes. [7] If we consider two dimensions only (to view a simplified scenario), the SVM is given the 2D coordinates of two classes, the two classes have dots of different colors (see Fig. 4). SVM chooses H_3 because it defines the maximum distance between the two classes. SVM then classifies the classes by seeing which side of the plane, the dots lie in. If SVM has to be trained for a new class, first it has to been seen that where the points lie and then assign their specific class.

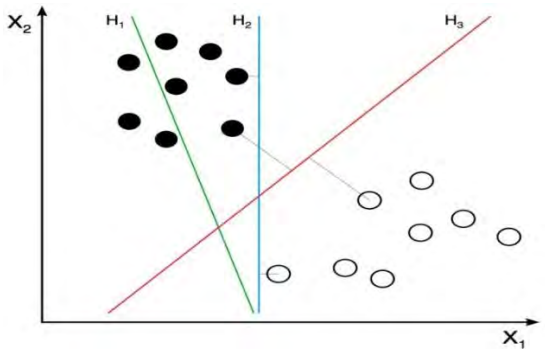


Fig. 4 – Example of how SVM works [19]

2.2.2 Nude Detection Algorithm

As discussed above, detection of Human Figures algorithm is applied on the fragmented image files from the video input which further created separate image files containing only the human figures. On these new image files, the nude detection algorithm will be performed to conclude whether an image is nude or not.

The very first step in the nude detection algorithm is to find the percentage of skin pixels relative to percentage of non-skin pixels in the given image. A relative comparison of skin to non-skin pixels helps determine whether the image is nude or not. Previous works on experiments shows that a threshold of 15% is good enough for determining nude scenes. Therefore, if the given image contains less than 15% skin pixel then it is not a nude image. Hence, the first step of the algorithm scans the image from upper left corner to the lower left corner and label all the pixels as either skin pixel or non-skin pixel type.

To verify whether a certain pixel is skin pixel or not, a skin color distribution model is used. From the observation on previous experiments and research work, it is seen that only a single-color space is more often used to determine whether the pixel is skin-pixel or not [1,2,3,4]. However, to improve the efficiency this model will combine three color spaces before classifying a pixel as skin or non-skin pixel. Thus, along with RGB values, Normalized RGB and HSV color spaces are also used while constructing the skin color distribution model. The data set for the testing of the model ensures that a good range of light, brown and dark images are used, so that the variation in skin tone doesn't hamper the results of our model. After scanning the image, the corresponding RGB values along with normalized RGB and HSV values are calculated and then compared with the parameters ranged by skin color model to determine whether it's skin or not.

If the image contains more than 15% of skin pixels compared to the whole image, then the next step is to identify skin regions by evaluating the continuity or adjacency between the skin pixels. These skin regions are then evaluated on the basis of the number of skin regions, the size of the skin regions and the relative position of the skin regions to classify the picture as nude or not.

Among the calculated skin regions, the three largest skin regions are taken into account. The percentage of these three skin regions relative to the image size is calculated. The leftmost, the uppermost, the rightmost and the lowermost skin pixels of the three largest regions are identified to plot three bounding polygons. Then the area of the polygons is calculated and the number of skin pixels within those polygons is counted. Percentage of the skin pixels within the bounding area of the polygon relative to the area of polygon is determined. If the number of skin pixels within the bounding polygon is less than 55 percent of the size of the polygon, the image is not nude. If the number of skin pixels in the largest skin region is less than 35% of the total skin count, the number of skin pixels in the second largest region is less than 30% of the total skin count and the number of skin pixels in the third largest regions is less than 30% of the total skin count, the image is not nude. If the number of skin regions is more than 60 and the average intensity within the polygon is less than 0.25, the image is not nude. Otherwise, the image is nude.

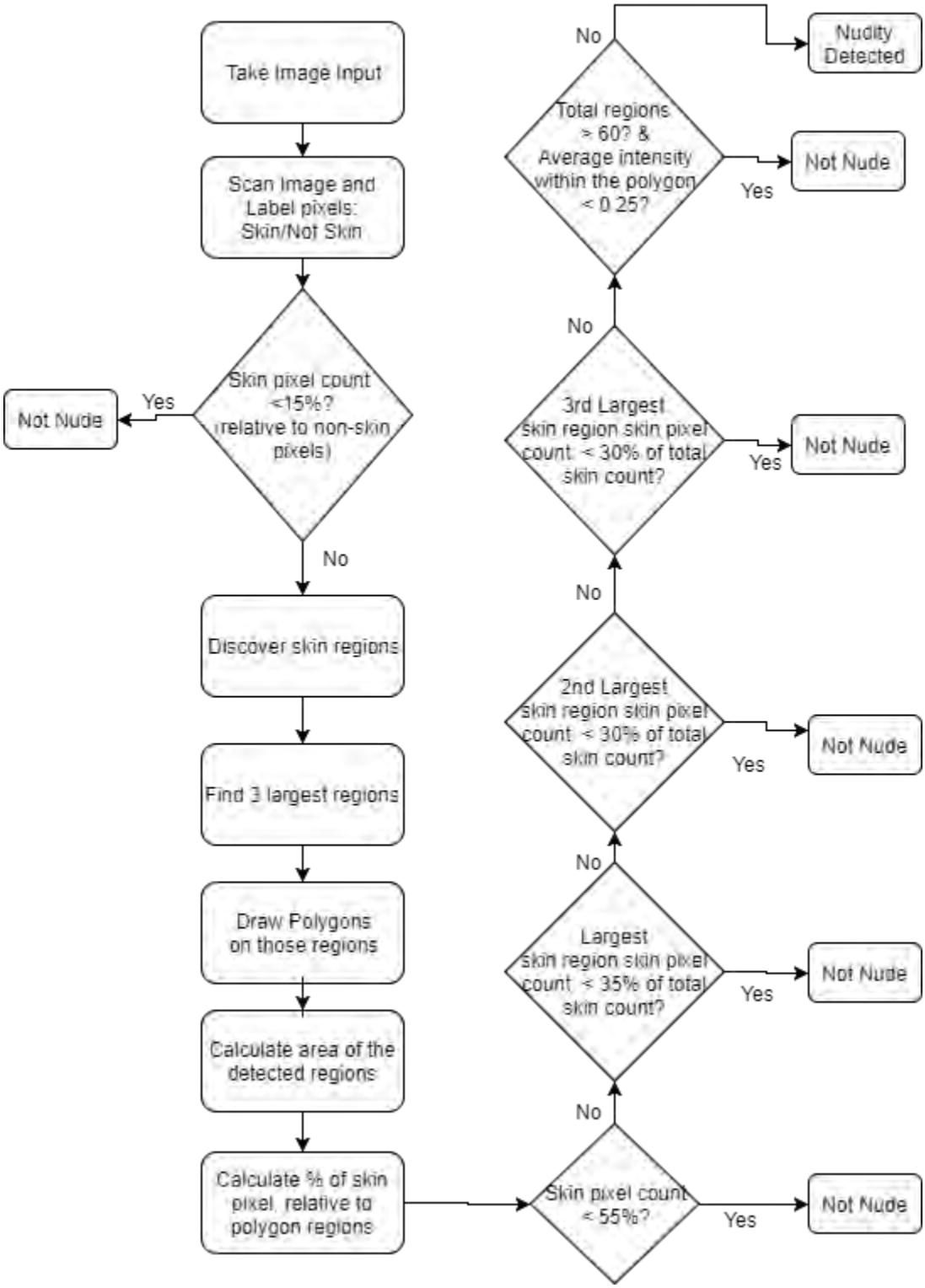


Fig. 5 – Flowchart for nudity detection algorithm

In the figure, Fig. 6, a sample picture is taken, then the human figure is cropped out of the original image and stored on a new image file. Now, nudity detection algorithm will be performed on this new image and finally output for the presence of nude is given.

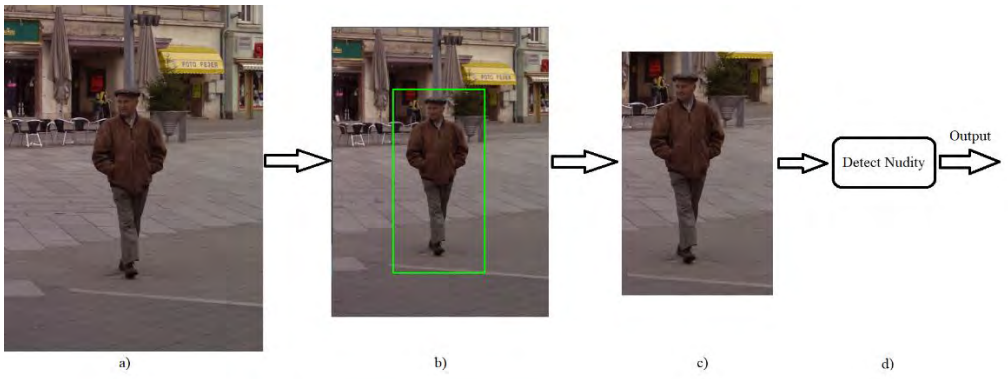


Fig. 6 – Combined output of human detection and nude detection algorithm

2.3 Object and Scene Detection

As mentioned above numerous times, the main goal of this research paper and proposed model is to detect unwanted scenes from a video stream. The first step was to detect nudity, the next step is to detect objects and scenes such as dangerous weapons, drugs, gore items from a given image. These scenes are usually disturbing to quite a fair amount of individuals, furthermore it may cause even further damage to any individual with mental illness. Moreover, it may also encourage certain people to act likewise, however, it can be such that their guardian won't allow, for this reason, for them to watch anything related to the required unwanted scenes. The following section is going to describe how that can be possible to be attained and get an idea prior to watching a video to what the viewer can expect from the video. In order to do so, we included the method of Convolutional Neural Network mechanism to detect our required segments from a given image. We loop through each image of the fragmented video file and check each image for our required content and then output a percentage of the images that contained unwanted objects.

2.3.1 Convolutional Neural Network to Detect Object

One of the specialized category in the study of Neural Networks is the Convolutional Neural Networks, i.e. ConvNets or CNNs, that is proven to be a very effective detector in areas of image recognition and classification. It is known to work on a grid-like topology [9]. The CNNs is successful in identifying a wide array of objects given in an image with very high success rate.

Furthermore, CNN are an important tool for most of the machine learning algorithms currently in practice. The figure, Fig. 7 demonstrates an example of the output generated by detecting objects by using the CNN algorithm.

Fig. 7 Example of detection of objects in an image using CNNs [2]

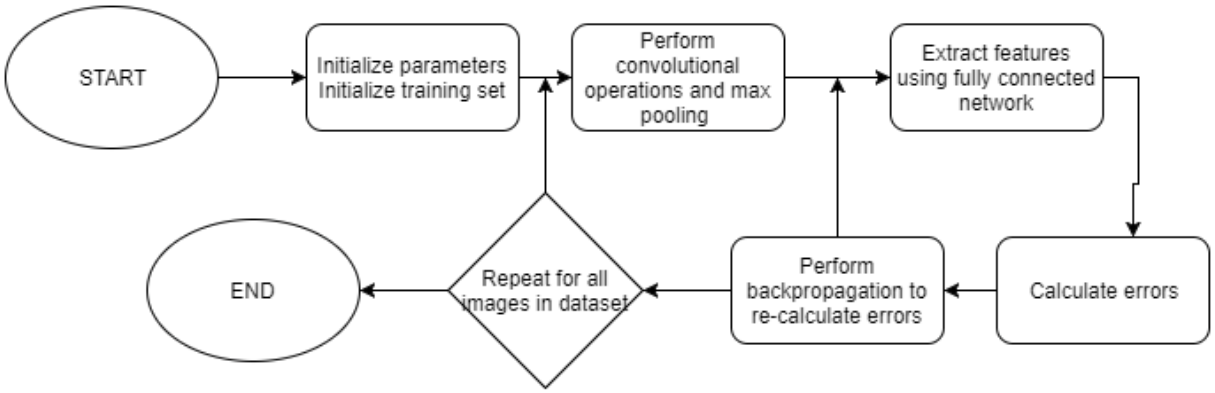


Fig. 8 – Flowchart of detecting objects from the image.

The CNN comprises of three major steps to detect object successfully. They are, convolution, pooling, getting fully connected network and finally, giving the output with probability matching. Fig. 9 shows the basic building blocks of a CNN. It goes through two phases of convolution and pooling followed by fully connected phase and finally giving the required output.

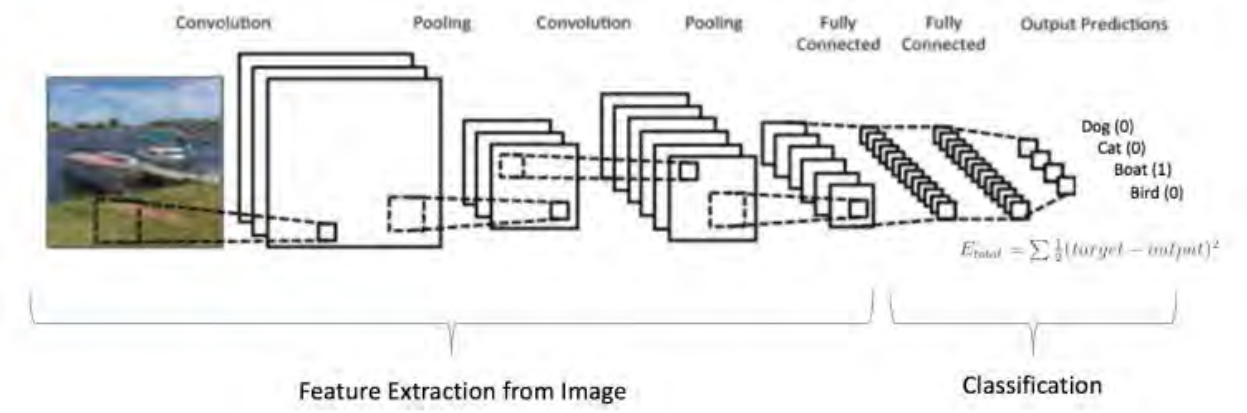


Fig. 9 – Example of a CNNs method of detecting object from a given image [8]

2.3.2 Definitions

Convolution

The term 'Convolution', in its most general form, means an operation on two functions of a real-valued argument. In order to correctly identify or detect an object, with least noise possible, it takes several measurements and then takes the average of it. However, the more recent the measurement is, the more accurate it is in terms of detection, as a result of that, the convolution process uses a weighting function to take weighted average of the measurements taken in account for the detection purpose [9]. Let x be the object or input that we are trying to detect, and t is the position of x at time t (these are the two real-valued data), next, let w(a) be the weighting function or the kernel where a denotes the age of the measurement that it took. When we apply convolution, we get the following function that is used to compute the position of the object

$$s(t) = \int x(a)w(t-a)da. \tag{1}$$

The output that we get from the above equation is a feature map of the input image [1]. Furthermore, the image is compared by flipping the kernel, w. This is done so to find the commutative properties of the input matched with the kernel [9, 13].

Pooling (Max Pooling)

The next step in CNNs is called pooling. In the above convolution process, it creates several parallel convolutions to produce a set of linear activations, next, each linear activation is then passed through a non-linear activation function, like the rectified linear activation function. This

step of the process is known as the Detector Stage. After this, to modify the output of the layer further, we use a pooling function [10]. For the CNNs, it performs max pooling (Zhou and Chellappa, 1988) where the maximum pixel matched is chosen out of three neighboring convoluted nodes. In the figure, Fig. 10, the pooling stage compares three neighboring nodes as shown in the figure to select the maximum value out of the three to select for its pooling.

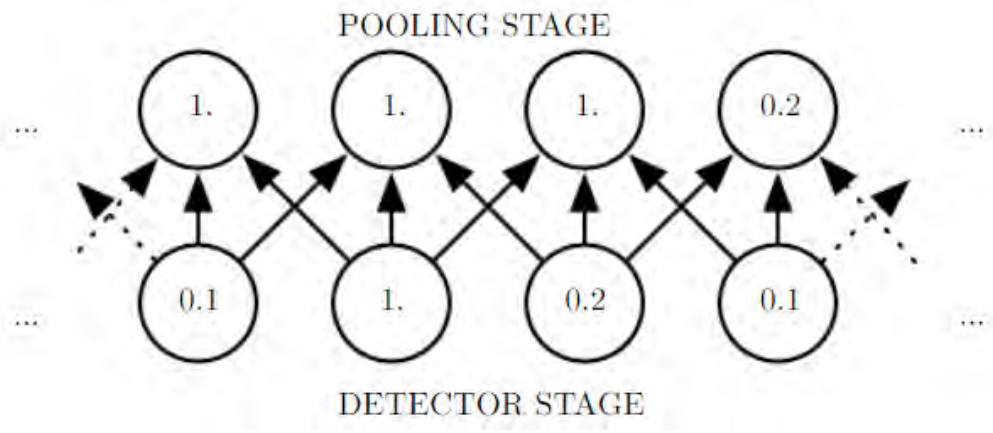


Fig. 10 An example of max pooling [9]

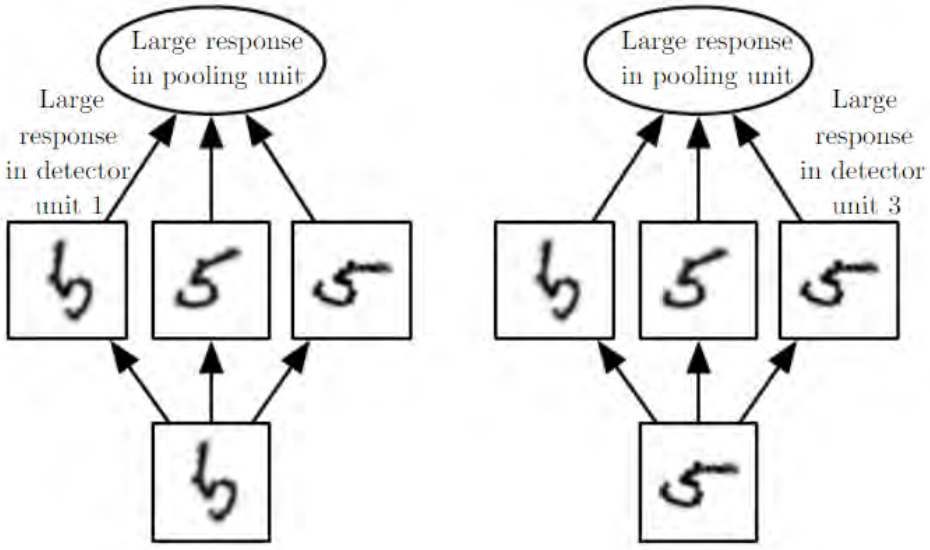


Fig. 11– An example of pooling from three different sets of detector [9]

Fully Connected Network

The final step of the CNN is forming the fully connected network and giving an output of probability. The fully connected network (FCN) may also be referred as Multi-Layer Perception [12]. The FCN uses the concept of a single neuron. It takes a series of inputs, which are weighted, from the previous step of the process (Fig. 12).

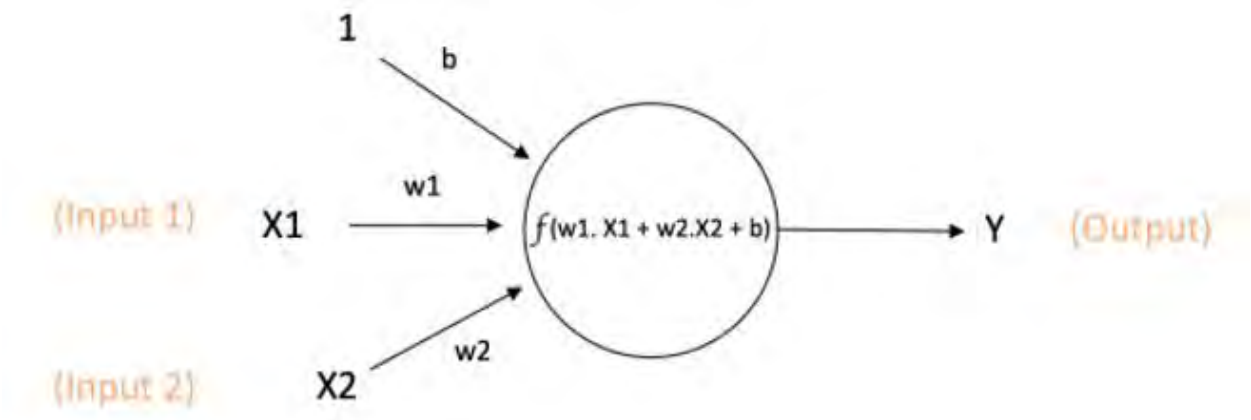


Fig. 12 – A single neuron [12]

$$Output\ of\ neuron = \ Y = f(w1.X1 + w2.X2 + b) \quad (2)$$

A Multi-Layer Perceptron (MLP) contains one more hidden layers in the process. The input layer is simply the input that it is fed from the previous step, and that is not altered by the FCN. Next, in the hidden layer, it takes the weights of all input nodes and creates new hidden nodes. These are called hidden nodes because, it has no direct connection with the outside world. Furthermore, the required calculations and computations required to give outputs is all done in this layer. The values are then fed to the output layer where each node in the output layer is again calculated using the concept of neurons, shown in Fig. 12. After this, all the outputs are compared to see which has the best probabilistic value out of all to properly identify the given object that was fed to the CNNs algorithm.

The figure, Fig. 13 shows the input neurons sending information to the next layer, which is the hidden layer. It can have one or more hidden layers. The hidden layers then send to output layer where the probabilistic results are calculated and given as output.

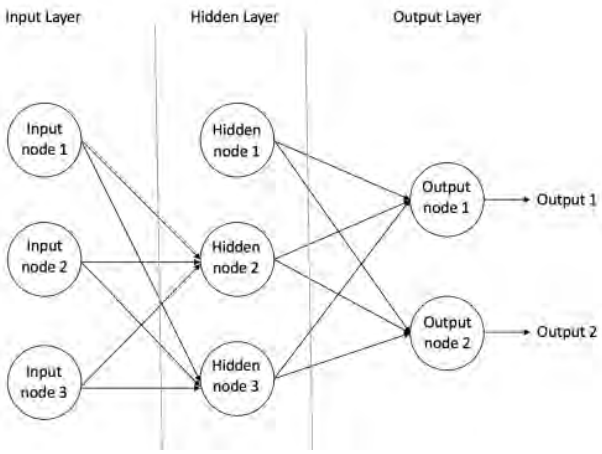


Fig. 13 – Example of a feed forward neural network [12]

2.3.3 LeNet Architecture

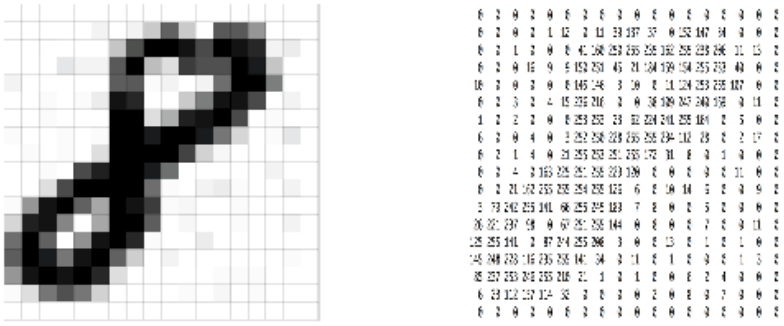
All the image recognizing architectures developed in the recent years have been based on the LeNet which was built in the 1990s, to recognize the images. The latter works are basically improvements done on LeNet to enhance its recognizing capabilities and increase accuracy and efficiency [14]. The architecture showed in fig x is similar to that of the original structure of LeNet. The example model presented in the image is capable of categorize objects found into dog, cat, boat, and a bird.

As mentioned above, CNNs has three major operations that are performed to detected objects, which are [11]:

1. Convolution
2. Pooling or Sub Sampling
3. Classification (Fully Connected Layer)

But before we get into how the architecture works, its required to know that, an image is a matrix of pixel values. Thus, the CNNs uses these pixel values to compare. If the image is a grayscale image, a 2d array is good enough. However, if it's a colored picture, the convolution step divides itself into layers, separating each color lair (i.e. red, green blue) and then performs its operations

on each layer. The CNNs treats each block of pixel as the corresponding color value, as shown in the figure below.



a) b)

Fig. 14(a) – Pixelated picture, (b) – Equivalent pixel color data [11]

2.3.4 Convolutional Neural Network Algorithm

2.3.4.1 Convolutional Operations and Max Pooling

The main purpose of this step is to extract features from the given input image. The convolution step preserves the spatial relationship between pixels by earning image features using small squares of input data. The mathematical concept is described is in the above section ‘Convolution’.

As discussed above, each pixel of an image gives the corresponding pixel color value, thus forming a matrix of these corresponding values. The convolution step takes the image, selects an area of the image, and then performs max pooling, to collect the highest valued feature match from that particular section of the convoluting task. This process is illustrated in the figure Fig. 15.

Furthermore, an image may go through the convolution process more than once. In other words, a single image may be convoluted and pooled more than once in order to increase feature extraction. This might be needed when an image is very large, in which case, it would mean that a given feature, in the original image, conquers more number of pixels. However, in the kernel, the dataset we need to match with has a fixed pixel block. Thus, when there’s a large image, it needs to be convoluted multiple times to bring it within the range for the kernel to pinpoint our required features

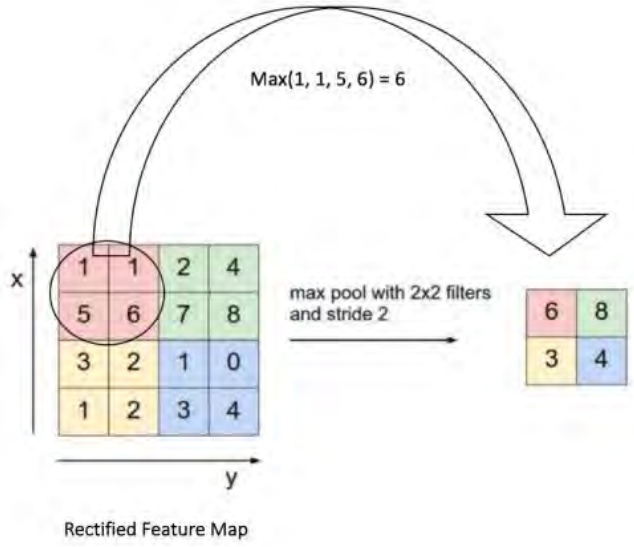


Fig.15 – Example of convolution segmenting and max pooling from a 4x4 to 2x2 matrix. [11]

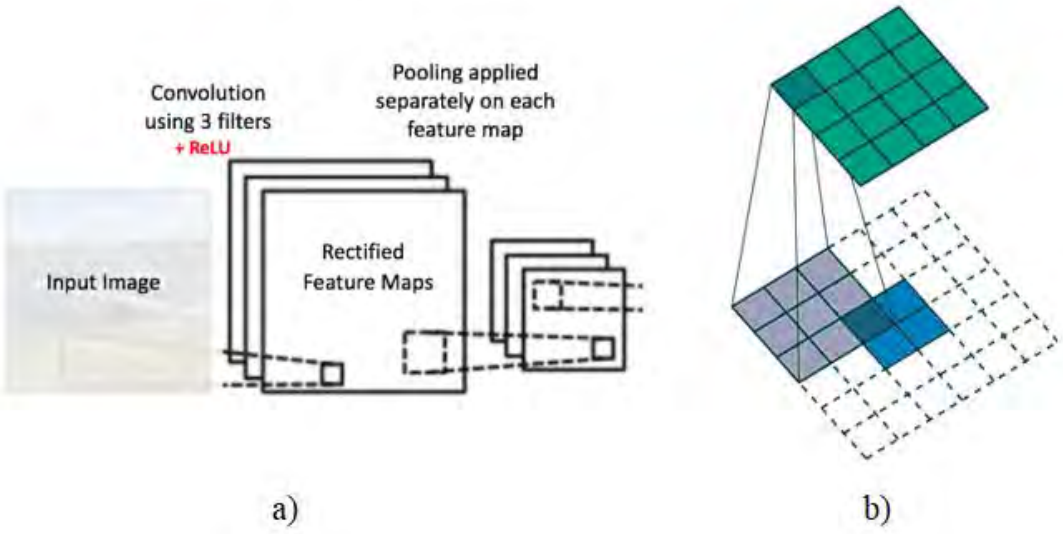


Fig. 16 – A different view of the above figure [11]

In CNN terminology, the 2×2 matrix, in Fig. 15, is called a 'filter' or 'kernel' or 'feature detector' and the matrix formed by sliding the filter over the image and computing the dot product is called the 'Convolved Feature' or 'Activation Map' or the 'Feature Map'. It is important to note that filters acts as feature detectors from the original input image [9,11].

The layers of the Featured Map are determined by the depth level, in other words, the number of filters we will use for the convolution operation [8].

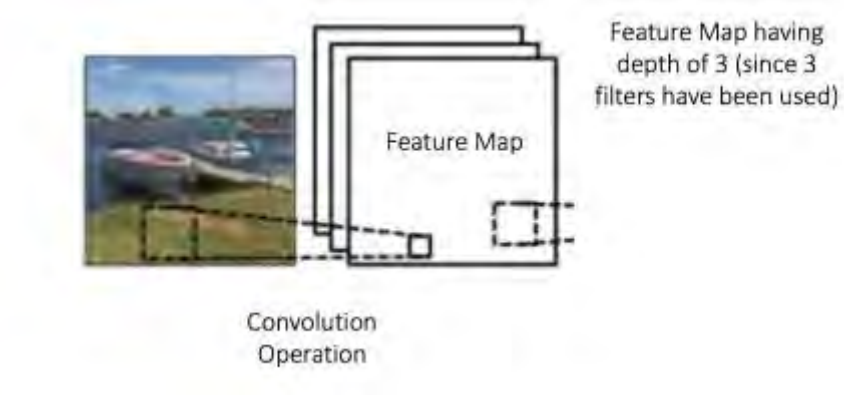


Fig. 17 – Depth of the feature map [11]

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Fig. 18 - In the figure above, in the table, we can see the effects of convolution of the above image with different filters by simply changing the matrix values of the image. [8, 9]

From the steps performed so far, together, these layers extract the useful features of the image [16] and the output of the 2nd stage of Pooling (Fig. 9), acts as an input to Fully Connected Layer.

2.3.4.2 Fully Connected Layer

The Fully Connected Layer (FCL) implies that every neuron in the previous layer is connected to every neuron on the next liar, discussed above in the current section. The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset. For example, in the example figure or model we have been using in this paper, the image classification task is set out to perform four possible outputs as shown in Fig. 19.

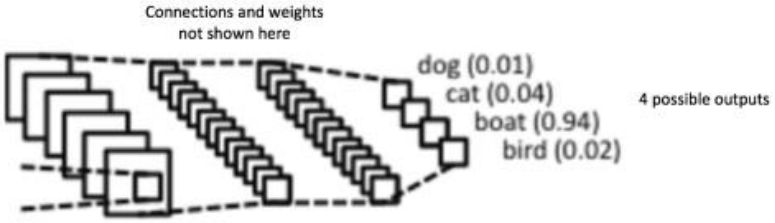


Fig. 19 – Probabilistic output after fully connected layer [15]

The output consists of an error function given by,

$$Total\ Error = \sum \frac{1}{2}(target\ probability - output\ probability)^2 \quad (3)$$

On calculating this error, the CNNs uses backpropagation to calculate gradients of the error with respect to all the weights in the network and use gradient descent to update all the filter values and parameter values to minimize the output error.

In the figure, Fig. 20 on the next page, the received data in the output neurons is sent backwards to the nodes to recalculate the function with newly assigned weighted measurements to achieve higher accuracy

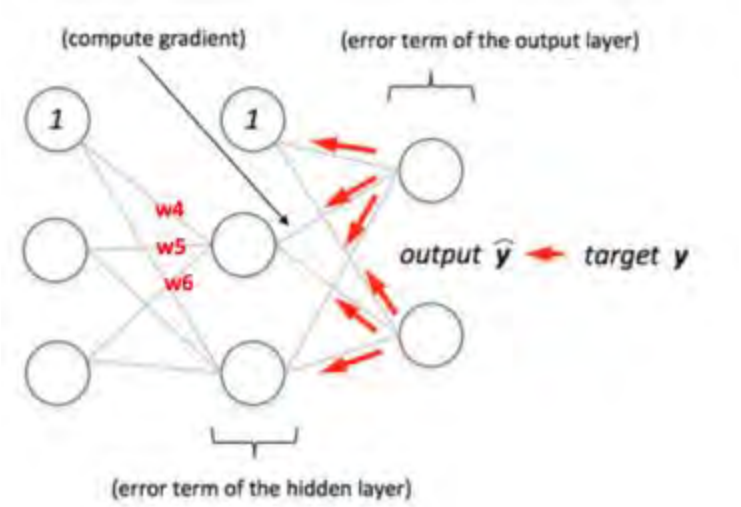


Fig. 20 – Backpropagation and weight adjustment [12].

$$Gradient = \frac{\partial y}{\partial w_{i,j}^{(l)}} l(W) = a_j^{(l)} \delta_i^{(l+1)} \tag{4}$$

$$Error\ term\ of\ the\ output\ layer = \delta^{(3)} = a^{(3)} - y \tag{5}$$

$$Error\ Term\ of\ the\ hidden\ layer = \delta^{(2)} = (W^{(2)})^T \delta^{(3)} * \frac{\partial g(z^{(2)})}{\partial z^{(2)}} \tag{6}$$

CHAPTER 03

PROPOSED MODEL

3.1 Flowchart

The data flowchart of the proposed model is presented in Fig. 21.

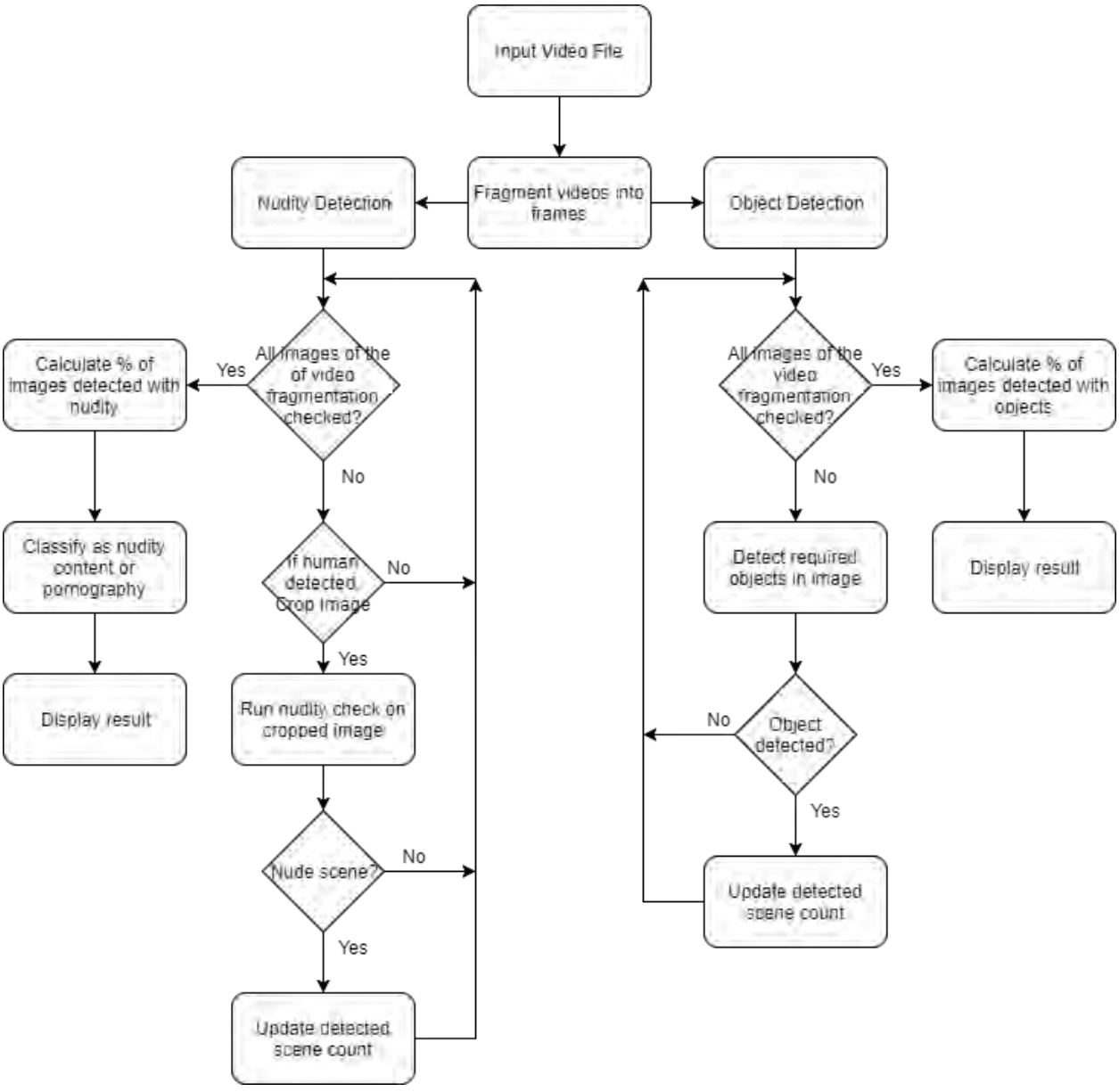


Fig. 21 – General flowchart of the proposed model

3.2 Workflow Description

As mentioned above in this paper, the proposed model has been divided into different steps in order to successfully detect our required scenes. The model, on its very first step, a video is selected as an input to the model. Once the input is taken successfully, the video file is then fragmented into individual video frames. This is done in order to use our detection algorithms, since they are image processing algorithms. We will be running each image via the algorithms via a loop and thus check each image one by one for both object and nudity detection separately.

The next step in the proposed model is running the detection algorithms over each images, extracted from the video file. Two tasks are performed here side by side, by taking advantage of the processor's parallel computing power, thus reducing run-time by half. One task is to detect objects, where we used the algorithm of Convolutional Neural Networks. This algorithm works by treating each image pixel as color pixels, and then extracts features and forms layers by using convolution. Next, the algorithm will perform max pooling to select the best matched feature block (Fig. 15) and then feed it to the Fully Connected Network for it to perform all the necessary calculations to get the best match of the object and correctly identify the object given in the fed image. On the other hand, simultaneously, the other task is to handle nudity in the image. This task utilizes two algorithms side by side to properly detect the nudity in an image. The original nudity detecting algorithm has a limitation, which is, it gives false negative whenever skin-pixel ratio to whole image pixel ratio was quite large, even if the picture contained some sort of nudity. However, to tackle this limitation, we added a human detecting algorithm first, which uses Histogram of Gradients and Linear SVM to detect humans. With the help of this human detecting algorithm we crop out the human from the original image and then perform nudity check to get a more correct result. However, if there's no human detected in the image at all, the model would skip the image and move on to check the next image in line. The above two task is ran over each frame of the fragmented video and a count is kept for each category (nudity and object) to be used later in the model for necessary calculations.

The final step of the proposed model is to display the percentage content of nudity and violence detected in the given video file. It does so by keeping a count of detected images in the above step in each category (nudity and object). After all the images are processed, the model compares the number of detected images to the total number of images of the fragmented video, and displays

the percentage of images detected in each category of nudity and violence separately. Furthermore, the model uses pre-defined threshold values in order to classify the video into a total of 6 categories, 3 for nudity and violence each presented in Table 1 and Table 2.

Table 1 – Threshold parameters for nudity detection

Level of Nudity	Percentage Threshold
Safe	< 5%
Unsafe	< 80%
Pornography	> 80%

Table 2 – Threshold parameters for violence detection

Level of Violence	Percentage Threshold
No Violence	< 5%
Mild Violence	< 60%
High Violence	> 60%

3.3 Algorithm

The algorithm for the proposed model can be divided into six major steps –

1. Take video as input.
2. Individual frames extraction from the video using fragmentation.
3. Simultaneously run nude detection and object detection algorithms on the collected frames.
4. Calculate percentage of nude scenes and unwanted object detected scenes.
5. Classify conclusion based on comparison of percentages detected in step 4 and pre-defined threshold values.
6. Display the output obtained from step 4 and 6.

A sample simulation of the above algorithm is illustrated in Fig. 23.

CHAPTER 04

EXPERIMENTAL SETUP AND RESULT ANALYSIS

4.1 Experimental Setup

The proposed model is divided into two parts as already discussed. One part deals with nudity detection which further determines adult scene like censored physically intimate scene in any movie, nude scene or pornography. Another part deals with finding scenes like crime scene, gore scene, violent scene, etc.

4.1.1 Dataset for nude detection and its improvements

While collecting the dataset for nude detection, we chose a wide range of images from different movies, online videos, documentaries and pornography. The main purpose of testing on such a wide range of data is to ensure the efficiency of this part of our model. As per the first step of the proposed model, the videos chosen were fragmented into different frames, that is different image files. The dataset for nudity detection varied in the following way

1. Nude pictures of people with different skin tones.
2. Black and white images of nude pictures.
3. Pictures were people wearing skin color dresses or has a background that contains objects which are close to skin color.
4. Images containing nude in a small portion relative to the size of the image.

A sample dataset has been illustrated in Fig. 22.

Fig. 22(a) Sample dataset for nudity detection (b) Sample dataset for violence detection

While testing the nudity detection algorithm on the above four types of dataset, some limitations of the algorithm were found. The algorithm works very well on different skin tones due to using combined color model distribution. However, the analysis of skin pixel to non-skin pixel where canceling out pictures which had less portion of nude image compared to the total image size. Therefore, we decided to fuse the nudity detection algorithm with human detection algorithm to come up with a model which is more efficient in determining nude scenes. Therefore, even if the background contains skin color theme or even if there is another object of skin color, still the algorithm works for determining nudes as the human bodies are first detected and cropped out. A sample example is shown in Fig. 23. In the given figure, the original image returned a false value when detecting for nudity, however, when the image was cropped out, the cropped image returned true value. In contrast, while dealing with data containing skin color dresses or images that are black or white, the nude detection part of the model fails to determine it as nude.

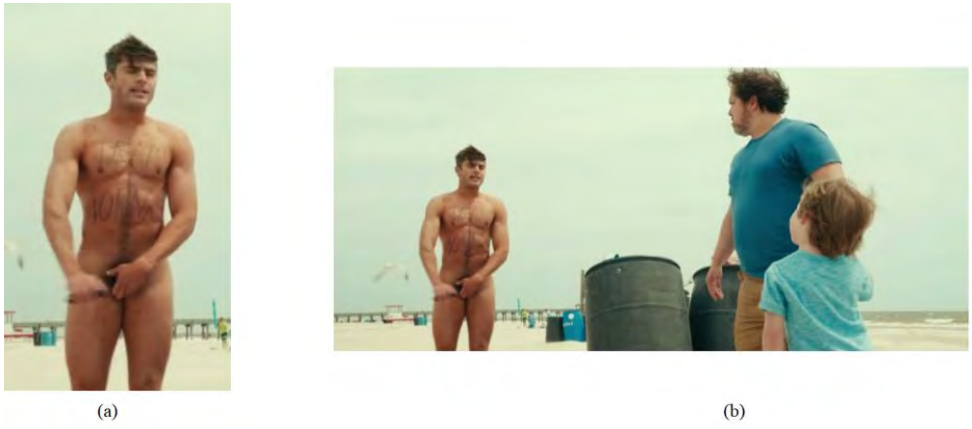


Fig. 23(a) – Cropped Image (b) Uncropped, original image.

4.1.2 Dataset for object detection and its wide range opportunity

For the object detection, we initially used the Clarifai dataset and tested on it to find different kind of objects in our fragmented images of the videos. After this testing part, we have used it to test on images containing gun, blood and knife as we considered crime scenes or gore scenes as inappropriate too. While testing on the object detection algorithm, we understood the huge opportunity of detecting any kind of scenes in videos from this aspect of the model. Therefore, user may customize the model according to his individual preference to identify any kind of scene he wants in a video stream, making this model greatly flexible.

4.1.3 Experimental Setup Workflow

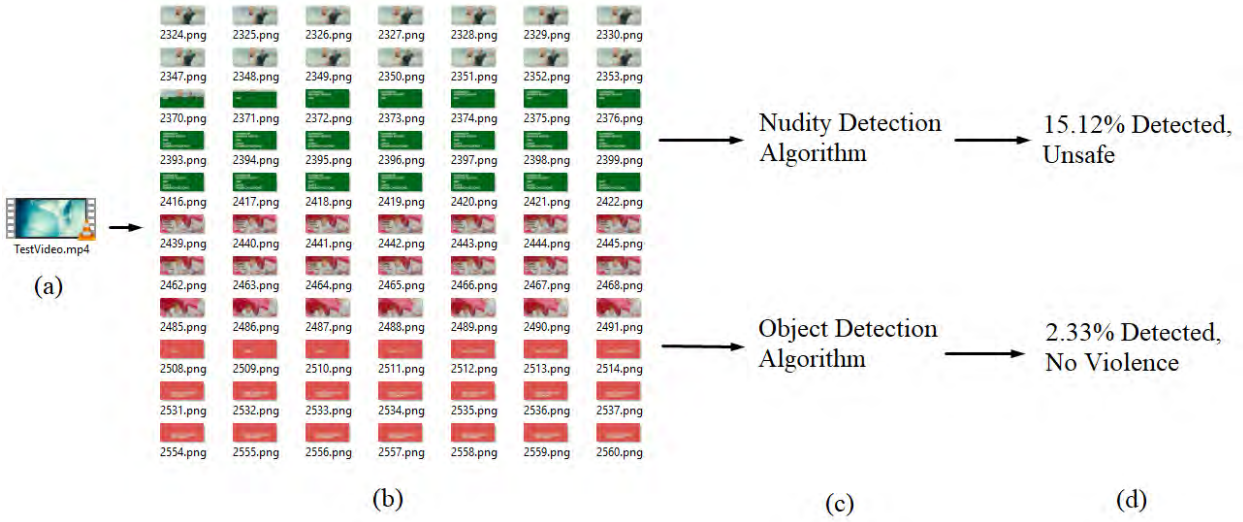


Fig. 24(a) – The input video file (b) Fragmented video file into frames (c) Running the model algorithms of detection (d) Output given in percentage detected with each algorithm used.

4.2 Tools Used

The tools that we used are listed below

- MATLAB R2017a (64bit) and R2015b (32bit)
- PyCharm Python IDE 2017 (64bit)
- CMDer (command line)
- CMD
- Eclipse Oxygen (32bit)

The libraries of python (unless, otherwise mentioned) that were used to test simulate the model

- OpenCV 2
- Pillow
- PIL

- clarifai
- nudepy
- pip
- Video Reader (MATLAB)
- Image Writer (MATLAB)

The model performs nude detection and object detection simultaneously which reduces the time for giving output to half and make the model more useful.

4.3 Results

To test the proposed model against our requirements, we simulated the whole model and collected the output result. Furthermore, we have set threshold parameters to categorize the detected videos into various levels of the detected portions, as mentioned in our workflow (Section 3.2). The threshold representation is shown in the table, 1 and 2 for nudity detection and violence detection respectively.

Moving on, given the threshold parameters in Table 1 and 2, to test the proposed model implied in this research paper, we tested with various number of videos and test images [27, 28]. A sample of the whole dataset has been presented in the tables Table 3 for nudity detection and Table 4 for violent scene detection.

Table 3 – Output result for nudity detection

Video Sample	Total Frames	% Nudity Estimated	% Nudity Detected	First Frame Detected with Nudity	Classification
1	748	0	0	-	Safe
2	152793	17	15.12	42900	Unsafe
3	600	0	0	-	Safe
4	620	0	0	-	Safe
5	580	56	49.46	212	Unsafe
6	600	80	85.36	53	Pornography

Table 4 – Output result for violence detection

Video Sample	Total Frames	% Violence Estimated	% Violence Detected	First Frame Detected with Violence	Classification
1	748	0	0	-	No Violence
2	152793	4	2.33	52233	No Violence
3	600	80	85.12	23	High Violence
4	620	90	89.35	10	High Violence
5	580	18	15.69	361	Mild Violence
6	600	0	0	-	No Violence

4.4 Analysis

First, looking at table 5, for nudity detection, it can be seen that, with the tools that we used for the proposed model gave approximately 90% accuracy for determining both the nudity and violent content of the video. Furthermore, as previously mentioned in our background study, we will be classifying the output percentage to a threshold, as per table 1 and 2, value to identify the level of the nudity and violence in the video stream respectively. Also, to give the result a visual illustration, a plotted graph has been added to Fig. 27 and 28.

Moreover, even from the graph, Fig. 25 and 26, it can be seen that, the curves of the estimated and detected nudity and violent content of the video goes very closely, proving it to have a good estimated efficiency.

Table 5 – Accuracy representation of the results obtained.

Video Sample	% Accuracy of Nudity Detection	% Accuracy of Violence Detection
1	100	100
2	88.94	91.5
3	100	93.6
4	100	99.32
5	88.32	87.17
6	93.3	100

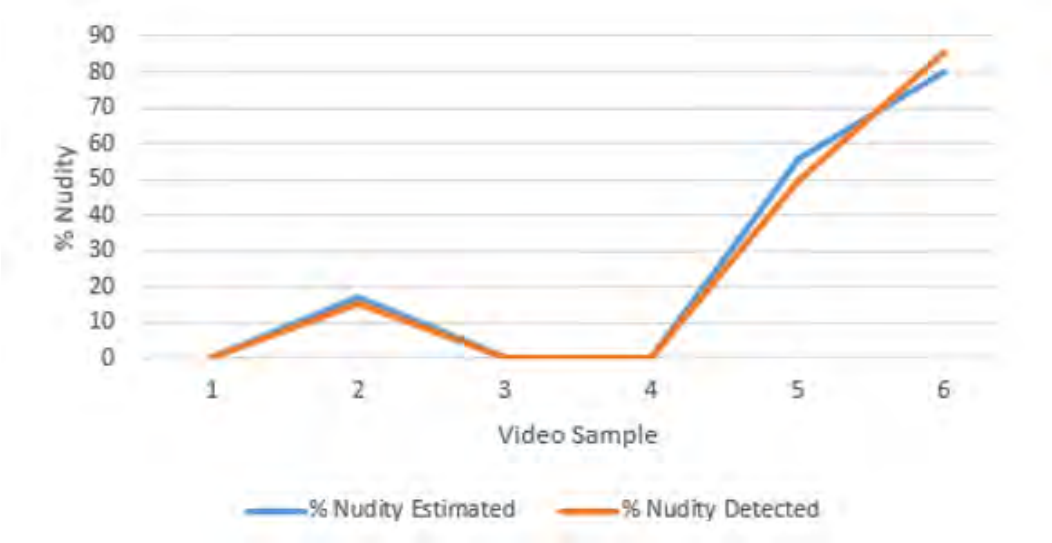


Fig. 25 – Graphical representation of nudity detection

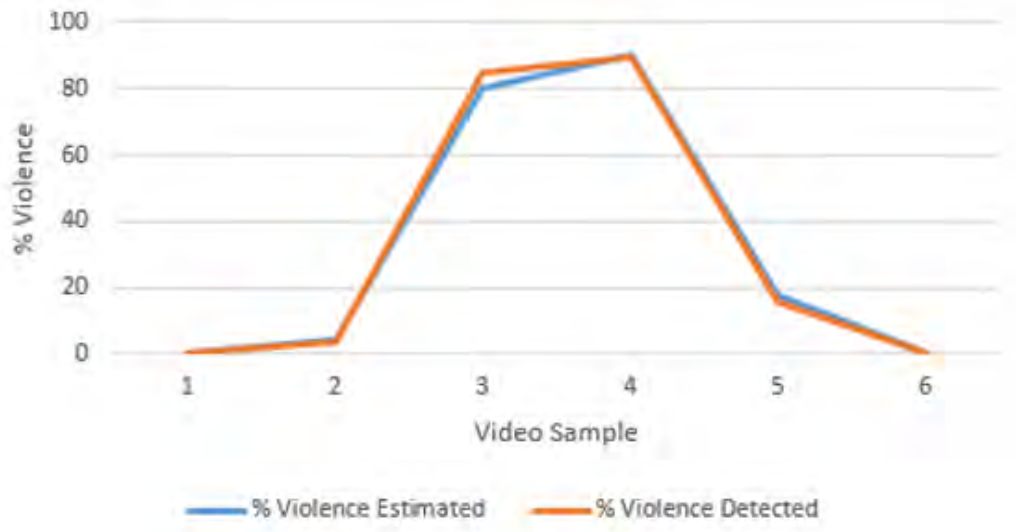


Fig. 26 – Graphical representation of violence detection

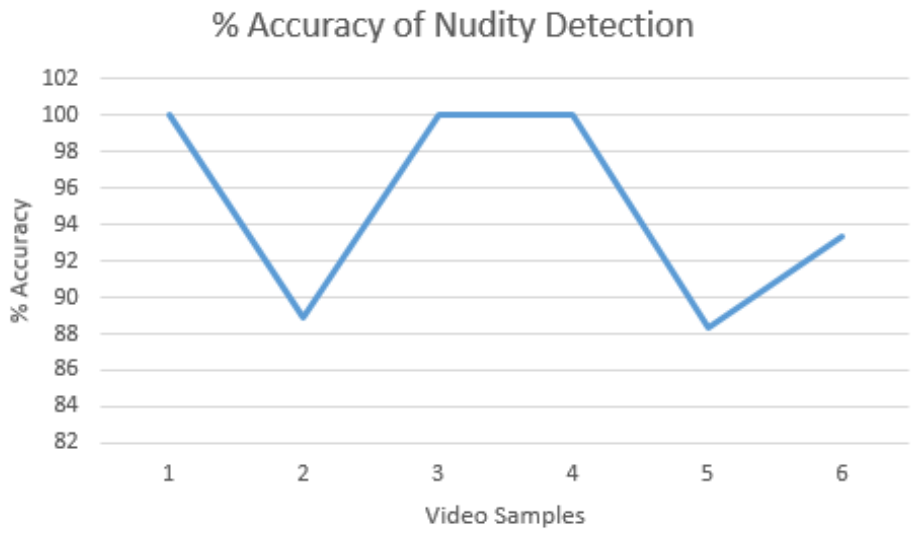


Fig. 27 – Graphical representation of percentage accuracy of nudity detection

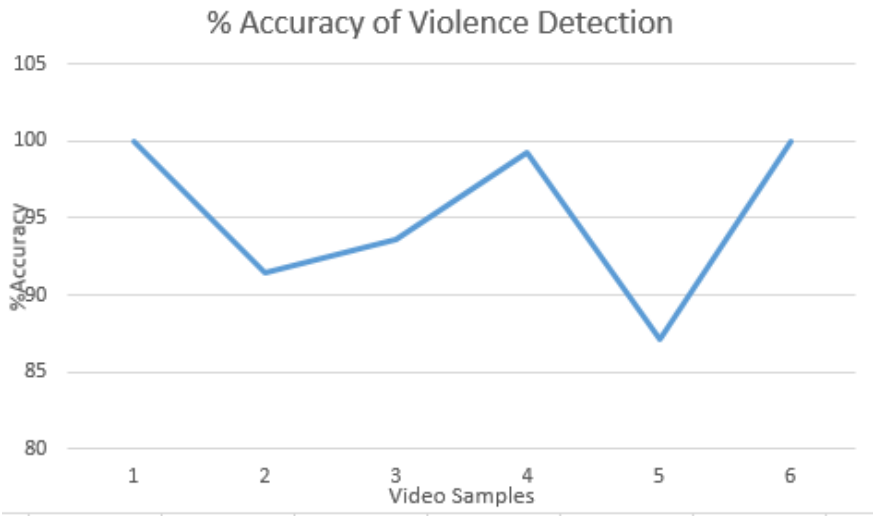


Fig. 28 – Graphical representation percentage accuracy of violence detection

CHAPTER 05

CONCLUSION

5.1 Conclusion

In conclusion of the above proposed model in this paper and from experimental results, it can be said that the proposed model was quite successful in detection of inappropriate scenes in a video stream. Although, the process is just in its preliminary stage, it can be deduced that the proposed model can be pursued further to increase efficiency and for further development for professional use. It can be also said that, the proposed model can be of great help to various groups of people in this world. Furthermore, it can be used with various apps and on video websites or applications as filtration process and precisely control the amount of content that can be allowed on the particular platform.

5.2 Limitations and Future Works

5.2.1 Limitations

Despite the proposed model working properly, there are a few limitations that we faced with the algorithms we selected to test our proposed model. Firstly, while detecting nudity, the algorithm fails to detect proper nudity on black and white images, since our algorithm is mainly based on skin color detection in order to detect nudity. However, since the modern time videos are rarely black and white nowadays, thus, this shouldn't be much of a problem for our current model's limitation. Furthermore, the algorithm gives false positive result when a person is wearing a skin-toned dress. Lastly, the nudity algorithm also fails to detect nudity if most of the body portion is not within the frame, the reason being, failing to extract all the features of a human body to be detected thus giving a false negative output. Moving on, secondly, just like nudity detection, while detecting objects, it again fails to detect objects properly when the color composition of the video is black and white. Furthermore, again like nudity detection, this object detection algorithm also fails to detect objects if one of the object's crucial part is hidden by some other obstacle or if the object is not fully visible within the frame of the video, thus, giving a false negative output.

5.2.2 Future Works

In addition to the current proposed model, presented in this research paper, it can be further improved in order to make the detection stronger. A few of the possible future developments that can be done with this model is mentioned below.

1. This model can be used as a background mobile app or desktop application which can be installed in home PC and mobile phones of teenagers, children, institutional premises. The model will detect any inappropriate video content before the computer or mobile plays any video. If detected, crossing the chosen threshold level, the model can be used to not allow the platform to play the video. Thus, giving the parents or the care taker a greater control over the video content usage of their children.
2. Another upgrade could be to detect and then delete or censor the detected scenes from the video stream. This would allow the video to be played safely anywhere necessary.
3. Detection of audio can also be included with this model. The purpose of this would be to detect certain pattern of language or detect specific words in order to detect or predict a probably unwanted scene. When detected, it can be either muted, or the scene can be fully deleted or censored as per required.

REFERENCES

[1] D. Chai and A. Bouzerdoum, "A Bayesian approach to skin color classification in YCbCr color space", in *Proc. of IEEE Region Ten Conference*, vol. 2, pp. 421-4124.

[2] Y. Lin, H. Tseng and C. Fuh, "Pornography Detection Using Support Vector Machine ",*16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP 2003), Kinmen,ROC*

[3] D.A. Forsyth and M.M. Fleck. (1999). Automatic Detection of Human Nudes. In *Proc. International Journal of Computer Vision*. 32(1), pp. 63-77.

[4] Michael J. Jones and M.Rehg James. (2002). Statistical color models with application to skin detection. *International Journal of Computer Vision*. 46(1), pp.81-96.

[5] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection,"Conference on *Computer Vision and Pattern Recognition* (CVPR), 2005.

[6] C. Cortes and V. Vapnik. (1995). Support vector networks. *Machine Learning*. 20(3), pp.273 – 297.

[7] C. Papageorgiou and T. Poggio. (2000). A Trainable System for Object Detection. *International Journal of Computer Vision*. 38(1), pp.15-33.

[8] S. Ren, K. He, R. Girshick and J. Sun. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. pp. 91-99.

[9] Ian Goodfellow, Yoshua Bengio and Aaron Courville. "Convolutional Networks," in *Deep Learning*, Deep Learning, MA: MIT Press, 2016, pp. 327-335.

[10] Ian Goodfellow, Yoshua Bengio and Aaron Courville. "Convolutional Networks," in *Deep Learning*, Deep Learning, MA: MIT Press, 2016, pp. 335-339.

[11] Ujjwal Karn. "An Intuitive Explanation of Convolutional Neural Networks." Internet: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/, August 11, 2016[Nov. 15,2017]

[12] Ujjwal Karn. "A Quick Introduction to Neural Networks." Internet: https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/, August 9, 2016[Nov. 15,2017]

[13] Eugenio Culurciello, “Algebraic Properties”, 2000, pp. 24.

[14] Eugenio Culurciello. "Neural Network Architectures." Internet: https://towardsdatascience.com/neural-network-architectures-156e5bad51ba, [Nov. 20,2017]

[15] Mathew Zeiler, “Implementation of CNNs” Internet: https://www.clarifai.com/, Aug.19, 2017.

[16] R. Girshick, J. Donahue, T. Darrell and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[17] "Clarifai | Image & Video Recognition API", *Clarifai.com*, 2017. [Online]. Available: http://www.clarifai.com. [Accessed: 21- Dec- 2017].

[18] Rigan Ap-Apid. "An algorithm for nudity detection," in *Proceedings of the 5th Philippine Computing Science Congress, Cebu City*, 2005.

[19] Satya Mallick. "Image Recognition and Object Detection: Part 1." Internet: https://www.learnopencv.com/image-recognition-and-object-detection-part1/, Nov.14, 2016 [Dec.1,2017]

[20] G. Gomez. "On Selecting Colour Components for Skin Detection," in *Proc. Of the ICPR*, vol. 2, pp.961-964, 2000.

[21] Birgitta Martinkauppi. *Face colour under varying illumination: analysis and applications*. Finland: University of Oulu, 2002.

[22] P. Peer, J. Kovac, and F. Solina. "Human Skin Colour Clustering for Face Detection," in *Proc. Of EUROCON 2003 – International Conference on Computer as a Tool*, 2003, vol. 2, pp. 144-148.

[23] V. Vezhnevets, V. Sazonov, and A. Andreeva. "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon*, 2003, vol. 3, pp. 85-92.

[24] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647-655.

[25] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation," in *Asian Conference on Computer Vision*, 1998, pp. 687-694.

[26] J. Wang, J. Li, G. Wiederhold, and O. Firschein. "System for Screening Objectionable Images Using Daubechies' Wavelets and Color Histograms," in *Proc. of the International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, 1997, pp. 10–30.

[27] "Google Images", *Google.com*, 2017. [Online]. Available: http://www.google.com/images. [Accessed: 08- Nov- 2017].

[28] "YouTube", *Youtube.com*, 2017. [Online]. Available: http://www.youtube.com. [Accessed: 21- Nov- 2017].