JIMMA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Pornographic Content Classification in Video Using Convolutional Neural Network and Gated recurrent unit (GRU)

By: Boaz Berhanu Tulu

Advisor: Kinde Anlay (PhD)

Co-Advisor: Mr. Fetulhak A.

A STUDY SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF JIMMA UNIVERSITY

IN PARTIAL FULFILLMENT FOR MASTERS OF SCIENCE

IN ELECTRICAL COMPUTER ENGINEERING

Date Aug -2021

Jimma, Ethiopia

DECLARATION

I do hereby declare that this Study is my original work and that it has not been submitted partially; or in full by any other person for an award of a degree in any other university/institution.

Name of Participant__________________________ Signature __________________

Date________________

This Study has been submitted for examination with my approval as College supervisor.

Name of P. Advisor_____________________________ Signature____________________

Date________________

Name of Co-advisor_____________________________ Signature___________________

Date________________

APPROVAL

The undersigned certify that they have read and recommend to the Faculty of Electrical and Computer Engineering to accept this Study submitted by “Boaz Berhanu” and entitled “Pornographic Video Classification Using Convolutional Neural Network and Gated recurrent unit (GRU)” in partial fulfillment of the requirements for the award of a Master’s Degree.

Name of Advisor ______________________________ Signature___________________

Date____________________________

Name of Internal Examiner_______________________ Signature _________________

Date ___________________________

Name of External Examiner _________________________ Signature _______________

Date ___________________________

Name of Head of Department ______________________ Signature _________________

Date __________________________

ACKNOLEDGEMENT

First, I would like to thank the almighty God, the most beneficent, the most merciful, for helping me to complete the work successfully. I express my gratitude and deep appreciation to my advisor, Dr. Kinde Anlay and co-advisor, Fetulhak Abdurahman (MSc), for their guidance, suggestions, and encouragement throughout this study. Finally, I extend my gratefulness to all of my families and friends for their directly and indirectly support during the study work.

ABSTRACT

Changes and advancement in many fields become eminent in the world where we live. In the last ten years, there has been a high coverage and availability of internet connection. These the advancement of technology and emerging of advanced computing platform bring a lot of advantages and negative influences on our community. From those negative threats, one that harshly attacks the youth's productive class of the population is the videos with pornographic contents. According to the annual statistics released by PornHub videos, 64 million people visited PornHub every day. This is the noticeable number. This will lead youths to exercise unsafe sexual behavior and of course they are exposed to sexually transmitted diseases like HIV. In this study we have proposed an automated classification of the pornographic videos. A combined effect of CNN pretrained models along with GRU has been employed to tackle this problem. The CNN pretrained model such as EfficientNet has been used for relevant feature extraction. Where the sequential learner bidirectional GRU is responsible for detecting the instance of video frames as porn video content or not. To evaluate the proposed model a publicly available 2K NPDI dataset from the university of Campinas, Brazil and applied has been used. A preliminary preprocessing steps such as normalization and cleaning has been applied on the dataset. We have used EfficientNet as a fine-tuned feature extractor in order to extract important features from the frames of the video and then the sequential information from the frame is learnt by DB-GRU network. In this DB-GRU network multiple layers are stacked together in both forward and backward pass in order to increase its depth and get good accuracy. Beside this various parameter optimization has been applied to increase the accuracy and performance of the proposed model. Following this, the experimental evaluation has showed a significant result of 99.68%. This result has improved by 0.68% and there is an improvement in efficiency in training and testing when compared to previous attempts on similar datasets. Various visualization methods have been used to present the result in more interpretable and human interpretable way. This will help readers to reproduce our work. Finally, we have tried to show the real time application of the proposed system by integrating and deploying the model, which has been already developed in this study along with web-based API. Thus, any user can scan to detect early whether a pornographic content present in a certain video stream or not.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

ARIMA:	Auto Regressive Integrated Moving Average
CNN:	Convolutional Neural Network
FC-LSTM:	Fully connected long short term memory
FC-RNN:	Fully connected recurrent unit
FLOPS:	Flouting Point Operation per Second
FN:	False Negative
FP:	False Positive
GRU:	Gated Recurrent Network
HMM:	Hidden Marcov Model
ILSVRC:	ImageNet Large scale visual recognition challenge
LSTM:	Long Short Term Memory
PPV:	Positive Predictive Value
RAM:	Random Access Memory
RGB:	Red Green Blue
RNN:	Recurrent Neural Network
SEM:	Sexually Explicit Material
TP:	True Positive
TRoF:	Temporal Robust Feature
UN:	United Nation

CHAPTER 1. INTRODUCTION

1.1 Background of the Study

Internet pornography is increasingly a prevalent part of the culture within the Ethiopia and in many countries worldwide. These days, pornography revenues have significantly decrease due to free online sexually explicit material (SEM). However, a recent study has showed there were 12.2 billion visits to the popular pornographic website PornHub in 2015 equaling approximately 4.3 billion hours of viewing [1]. Among PornHub visitors worldwide 60% of them are millennia's (i.e., the generation of people born between 1982 and 2004 [2]. On the other hand, 50% of college students report viewing pornography online. These statistics undoubtedly associate the proliferation of internet pornography with the technological progression of the modern world. The availability, anonymity(i.e., the perception of inconspicuousness when viewing online sexual content) and affordability of internet pornography has significantly affected the consumption of SEM for adults.

Following the evolution of advanced computers and existence of high-speed digital cameras, large number of pornographic resources over the internet is growing rapidly. Nowadays, the Internet has become an essential part of people's lives, including children. The uncontrolled access to the Internet, anonymity of online porn content and easy access of such resources potentially leads to serious social and cultural problems[3]. The precise definition of pornography may be subjective, and we consider it as "any sexually explicit material with the aim of sexual arousal or fantasy"[4]. Sexuality is a highly entertained topic through every available media worldwide in this generation more than ever. It is especially the hottest issue for adolescents and young adults. According to the 2012 United Nations (UN) report in[5][6], the world is home to 1.2 billion individuals aged 10–19 years and this age group is defined to be adolescents. Since, adolescence age is a transition period from childhood to adulthood many youths develop sexual behaviors at this age mainly derived by pornographic content of videos.

Figure 1- 1: Pornographic image using smart phone

Figure 1- 1: Pornographic image using smart phonedescribes Sexuality influences a number of the motives of people's behavior, e.g. at work, at school, or in free time, and it affects their values and everyday needs. The respondents access pornography almost exclusively on mobile devices via the Internet (streaming) and prefer free pornographic content. Only 5.9% of men and 3.6% of women stated that they also view porn at school or at work.

Nearly 800,000 young women aged 15-19 years become pregnant in the United States each year, most of them unintentionally, and half of the roughly 19 million new sexually transmitted infections (STIs) diagnosed each year are among 15- to 24-year- olds[7]. In Ethiopia, a research in [8] has conducted a study in Addis Ababa, Gulele sub-city to analyze the pornographic usage of students at a high school level. Thus, the result has showed that most students 569 (72.5%) has been exposed at least once and 368 (47%) of them has consumed (used pornographic materials) repeatedly. Among those students who have been exposed to pornography were in a higher proportion of reported not to have used condoms and have multiple sexual partners. They have found also pornographic consumers to be 6 times more likely to have sexual debut compared to youths who have not sexual experiences. Moreover, the result has reported also students consumed pornography to get information and experience about sexuality.

Figure 1-2 below shows the annual report of Pornhub, the popular pornographic video provider. Pornhub is the icon of pornographic video. It is a multi-million company based in the USA. Pornhub release aannual report about their user traffic along with frequently searched searched and viewed contents information. In the Figure 1-2 up to Figure 1-4 below we have put three most eminently growing reports of Pornhub. The first happens in every minute in the Pornhub industris. This infers large amount of bandwidth has been consumed transferring large amounts of video data. Beside this, large amounts of people are wasting their time to view such contents on Pornhub each minute. Figure 1- 3 showed that an enormous amount of females and working part of the community (youths) have been watching porn in the developing country. We strongly believe that practicing these activities leads extensive amount of crisis in terms of psychologically, socially and economically in the developing countries. In the Figure 1- 2 below we have shown the yearly review of Pornhub website.

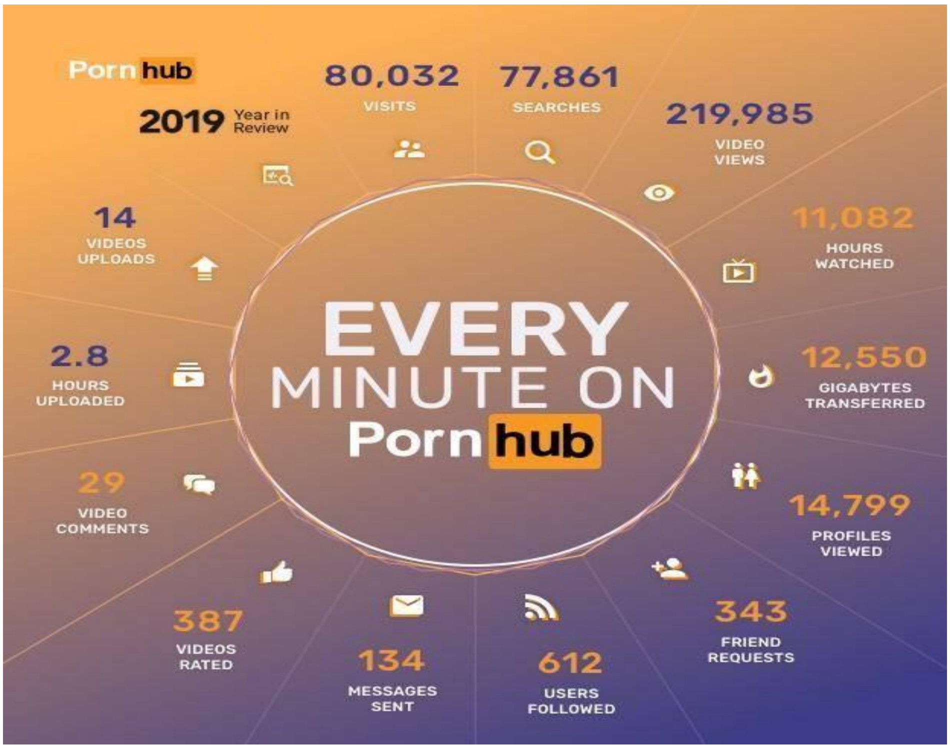


Figure 1- 2: PornHub annual report in 2019

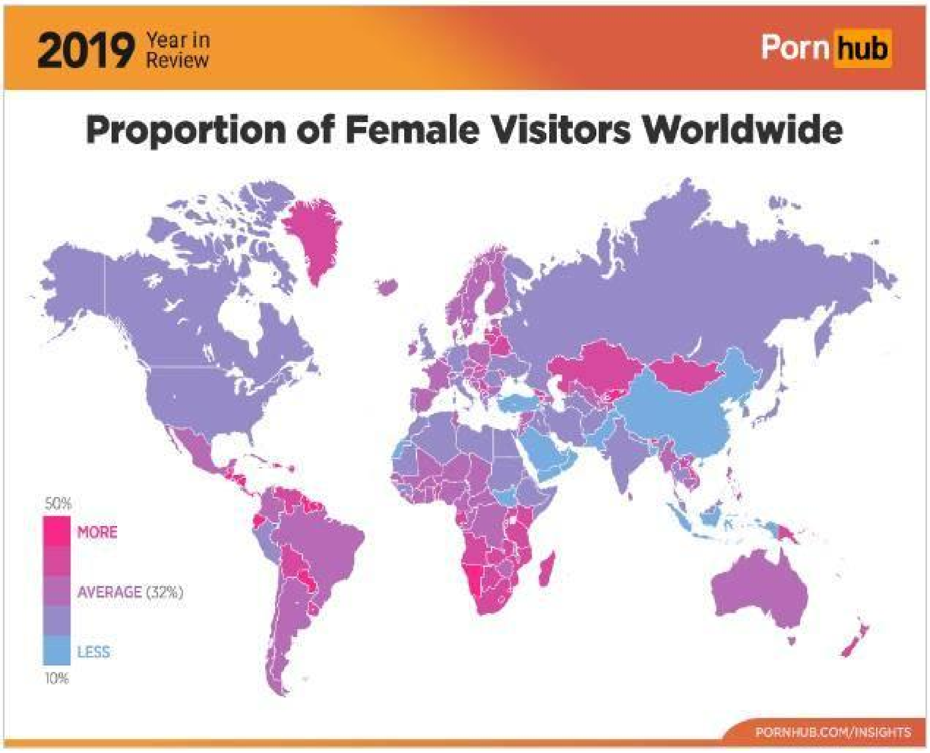


Figure 1- 3: Pornhub annual report on Female visitors in 2019

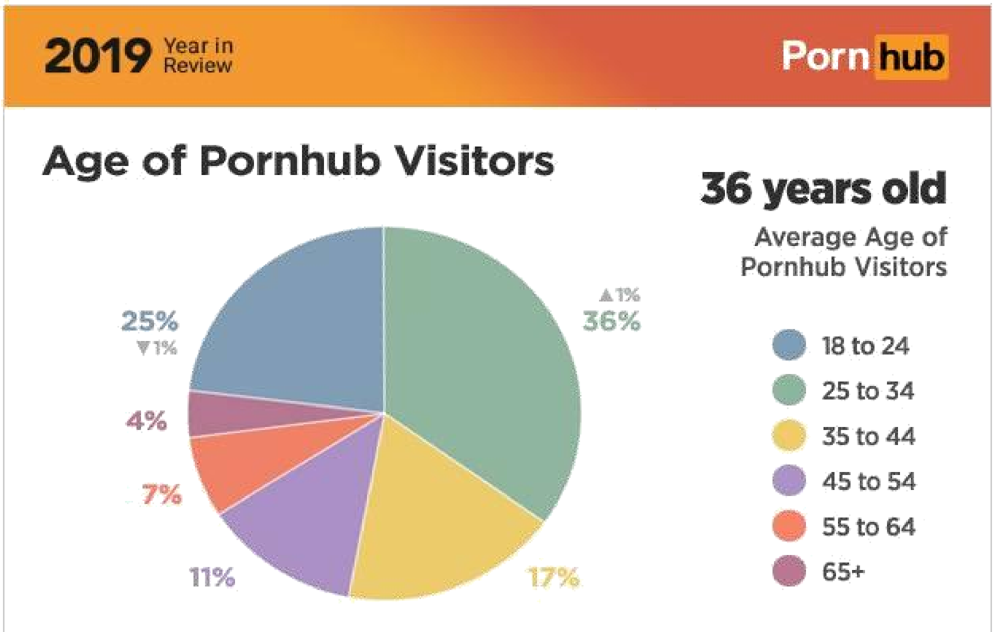


Figure 1- 4: Pornhub annual report on Age of visitors in 2019

The Figure 1-2 above has showed the statistics what could happen in every minute over the Pornhub. The detailed segmentation has described that each minute they upload 3 hours of porn video to Pornhub[2]; which is one of the million porn websites worldwide and each minute 80,000 people visit pornhub website. Figure 1-3 has also clearly shows the participation of women on Pornhub website. Figure 1-4 has shown the age distribution of users in Pornhub website; Thus it indicates ages between 18 to 24 are widely experienced. This statistic clearly infers the most productive age groups or dominantly youths has been wasting their time on watching pornography videos. As a result, policy makers and other corresponding stakeholders in the market has to take serious measure on restriction of posting such contents over the web to decrease the sexual harassment.

Having all these considerations, in this study we have proposed to classify destructive pornographic contents from videos. To tackle this problem a generic neural network architecture such as; Convolutional neural network (CNN) [9] and Gated Recurrent Unit (GRU) version of sequential models [10] has been employed. An experiment has been conducted on publicly available NPDI dataset [11]. The proposed method has showed significant result with better accuracy and short training time than existing state of the art research work.

1.2 Statement of the problem

These days, there exists large amount of pornographic contents over the we than any time in human history. Addiction to pornographic video has been increasing habit in the developed and developing country; most recently, this becomes a case for many suicide and homicide activities. The serial killer Ted Bundy who killed more than 40 women's has said Pornographic contents motivated him to kill all those innocent 40 women's [12]. Now a day's, teens often seek social and sexual information from social media sources rather than their parents and they have been dominantly exposed to articles with sexual contents. According to findings of different studies such as [13] [14]; teens who watch sexual content on television are more likely to engage in sexual activities. Besides this, they have also exposed to sexually transmitted diseases such as HIV and STD. Most importantly, teens who have been frequently watching sexual content rich videos on television, mobile device and over the web tends to have more negative attitudes and worry about to stay without sexual partners.

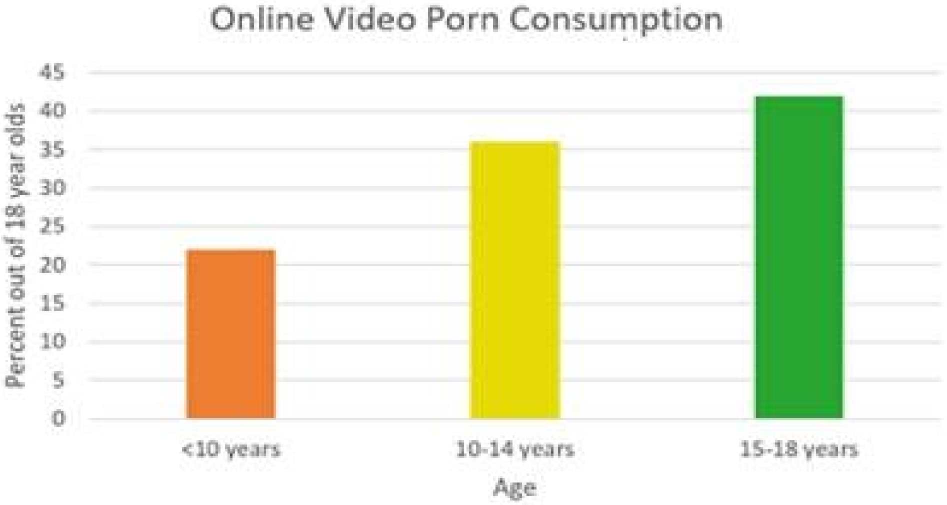


Figure 1- 5 Online Pornographic video consumption(age<18)

According to the annual statistics released by PornHub videos viewed in the year 2015 is 91,980,225,000 videos and 3,110,400,000 GB bandwidths is used each day, 64 million people visited PornHub every day[1]. In Figure 1-5 it shows the consumption of pornographic contents in the young children below 18 years. In most of the country it is prohibited to watch pornographic contents below the age of 18 years old[15]. Because of all those problems there is a high demand for automatic pornographic video classification tasks. Even though, there are many attempts at identifying pornographic video using different methods there still a challenge. Such, as in [16] which uses images for pornographic video classification and we use the two deep learning algorithms [17] the combination of CNN and RNN to classify pornographic videos based on the static and motion information we get from the video with less amount of time and good accuracy.[18][19]

1.3 Research Question

This is study attempts to explore best techniques to classify pornographic video contents. Based previous trials potential video classification methods have been used in similar works. Dominantly, powerful deep learning models has gained attention in multimedia data classification and detection. Thus, we have identified significant possible questions to address in this study; by employing CNN and sequential models such as GRU and LSTM.

Therefore, the research question we are planning to answer under this study is:

- ***Research Question 1:*** *How much improvement can we get in the runtime efficiency and classification accuracy on the NPDI Dataset when considering both the static and motion in-formations available in the video?*
- ***Research Question 2:*** *How efficient are the popular per-trained feature extractors (ResNet50, VGG19, Inception and Mobile Net) on extracting relevant feature from the NPDI Dataset?*

1.4 Objectives

1.4.1 General objective

The general objective of this study is to design a pornographic video classification system using powerful deep learning techniques.

1.4.2 Specific objective

In order to achieve the general objective, the specific objectives are:

- Identifying related studies that has been conducted so far on pornographic video classifications to understand the benchmarks.
- Exploring and pre-process the proper dataset
- Extracting relevant features from video frames Using CNN
- Propose and design a generic CNN-GRU pornographic classification technique.
- Evaluate the performance and accuracy of the proposed model with various hyperparameters tuning.
- Comparing and discussing the experimental results
- Develop a prototype of the proposed classification model.

1.5 Motivation the study

In watching typical pornographic video they are also exposed to series of scenarios including kissing, sexual touching, masturbation. I have been working on this topic starting from my internship program at VLM office Addis Abeba there I meet difference youth who have been addicted to porn video when I hear there testimonial I was very motivated to have a solution by using my knowledge I got from my internship program even in my internship program I developed an application called Nesa Hiwet which help peoples addicted to porn videos. This is what motivated me to have my research work on this topic.

1.6 Scope and limitation of the study

The scope of this research is to cover the proposed classification mechanism for pornographic Videos by combining two popular deep learning algorithms CNN (Convolutional Neural Network) and GRU (Gated Recurrent Unit). To develop a prototype using python programming language which is used as a tool because of its rich feature and powerful deep learning libraries. We mainly interested on detecting pornographic contents, which exists in videos only. However, this study will not consider pornographic contents found in materials such as text base and voice base contents.

1.7 Significance of the study

It is urgent and necessary for families and society to take effective measures to prevent teenagers from accessing Pornographic videos. To reduce the negative impact of these videos, a variety of methods have been proposed to detect the presence of pornographic videos. Due to the fast and regular availability of pornographic videos it becomes a difficult task to detect phonographic videos and this has helped porn videos to easily circulate in different websites and in youths mobile phone. This research work will provide a new, fast and efficient way to classify pornographic videos from the non-pornographic videos.

1.8 Methodology

In this study, we have applied a design science approach [38] to tackle the research questions mentioned in section 1.3 as part of quantitative and experimental research; we have followed the following high-level pipelines.

Literature review: - we have conducted Various literature reviews on the areas of Pornographic video classification, feature extraction techniques that are relevant to our work. During the review, available books, journals, case studies, previous research works, and guidelines has been explored to clearly understand the subject.

Dataset Pre-processing and feature extraction: -At this step potential dataset has been identified and appropriate preprocessing techniques has been employed. This helps to perform to enhance dataset quality, and to create a convenient condition feature extraction first by converting the video to frame and then by applying image augmentation technique in order to enhance the quality of images.

Method selection: - once the relevant features has been extracted using transfer learning the immediate task is classifying pornographic videos using Gated Recurrent unit.

Evaluation and discussion: - A standard evaluation metrics such as precision, recall, F1-score and accuracy has been employed to evaluate the proposed model.

1.9 Outline of the study

The structure of this study has organized as follows : In **CHAPTER 2** theoretical background of the study works are presented. It elaborates on the terms and definitions we have used in this study on deep learning and describes the variants of the RNNs and CNN used in this study. Following this, in **CHAPTER 3** we have examined different literature reviews related to pornographic video classification task. It also describes the earlier studies where researchers worked on classifying events in the Pornographic videos with deep learning combined with other techniques.

CHAPTER 4 has described the design of this study. The design section consists major steps such as: fine-tuned feature extraction using transfer learning followed by training a sequence of feature vectors using Gated Recurrent Unit has presented in detail. **CHAPTER 5** describes the methods and the experimental setup. It also explains the evaluation metrics considered comparing the models used in the study. Here, it also presented the study experimental results with their corresponding discussions. **CHAPTER 6** is the last section of our study work and it consists of the conclusion and recommendation.

1.10 Main Findings

The main finding of this study is the full-fledged set-up that starts at video dataset preprocessing and feature extraction using EfficientNet[9] ;which is a pre-trained transfer learning method to prepare the input for the pornographic video classification model. Following this, the extracted spatial video features passes for GRU [10]. The result has been evaluated by using different evaluation matrices[16] precision, accuracy and F1 scores. The proposed model has showed a significant improvement from the previous methods for classifying pornographic videos.

CHAPTER 2. THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Pornography

Depending on various previous empirical and theoretical based researches, [17] had categorized pornography in to four types: violent pornography, non-explicit sexual aggression, nonviolent, low degradation pornography and nonviolent, high degradation pornography. [17] Explains those categories of pornography in the following way:

The first category of pornography, violent pornography, usually depicts sexual coercion in a sexually explicit context. Such kind of pornography depicts a common feature what many researchers call it, "positive victim outcome" or rape. On the other hand, non-explicit sexual aggression, the second type of pornography, reveals male sexual aggression against women, which is more or less similar to the first one. Pornography that consists of the depiction of mutually pleasurable sexual acts between adults or pictures of nude women is said to be non-violent, low degradation pornography. The final one is nonviolent, high degradation pornography, which uncovers degrading and dehumanizing of women through the depiction of women in subordinate positions, as being abused in some manner or over sexed or highly promiscuous. [17]

In general Pornography is the graphic sexually explicit subordination of women or men through picture and/or texts [4]. Some of the exploitation of pornography include one or many of the following phrases:

(i) Women are presented dehumanized as sexual objects, things or commodities

(ii) Women are presented as sexual objects who experience sexual pleasure in being raped

(iii) Women are presented as sexual objects who enjoy pain or humiliation

(iv) Women are presented as sexual objects tied up or cut up or mutilated or bruised or physically hurt

(v) Women are presented in postures or display

(vi) Women's body parts including but not limited to vaginas, breasts, or buttocks-are exhibited such that women are reduced to these parts

(vii) Women are presented as whores by nature

(viii) Women are presented being penetrated by objects or animals

(ix) Woman are presented in scenarios of degradation, injury, torture, filthy or inferior, bleeding, bruised, or hurt in a context that makes these conditions sexual.

2.2 Neural networks

Figure 2- 1 shows a basic neural network, composed of 3 layers, an input layer, a hidden layer, and an output layer. In this network, each unit (also called a neuron) in each layer is connected to all other units in the next layer. When a neural network is applied for prediction task, the input data is fed into the input layer, following the arrows of the connections to go through each subsequent layer of the network, and finally producing the output by the output layer [18]

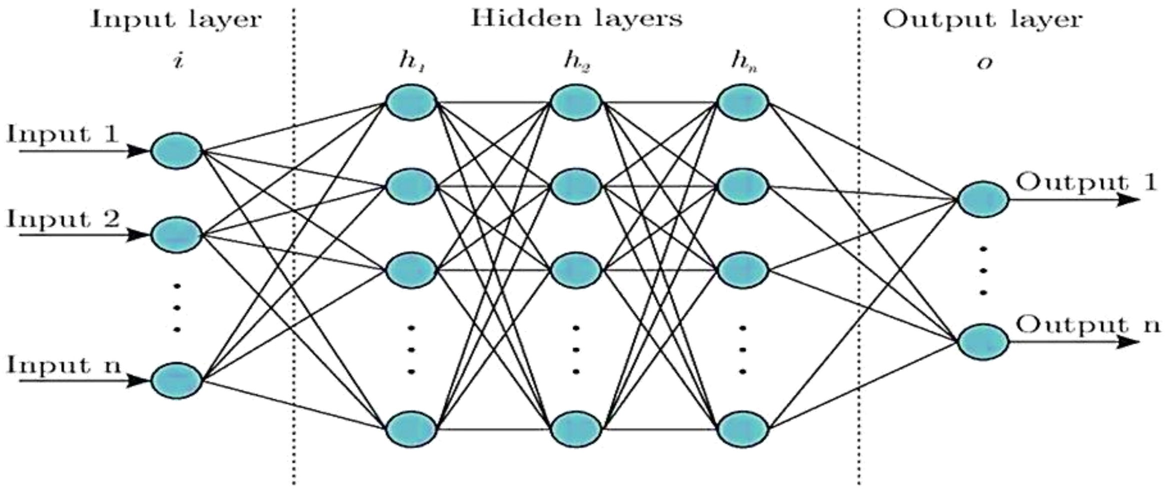


Figure 2- 1: Basic neural network structure

2.2.1 Deep neural networks

Neural networks with more than one hidden layer are called deep neural networks. Neural networks for handling real world applications are usually deep, say at least l0 layers, hence the name deep learning.Figure 2- 2: Deep Neural Network structure Recent neural network architectures can go even further as to reaching the depth of over one hundred of layers. This makes neural networks very flexible in producing its output, to the extent that deep neural networks are sometimes called universal function approximates. Which roughly means that a neural network can approximate any function from one finite-dimensional space to another with any desired nonzero amount of error, provided that the network is given enough hidden units [19]. The Figure 2- 2 below shows the basic structure of a deep neural network.

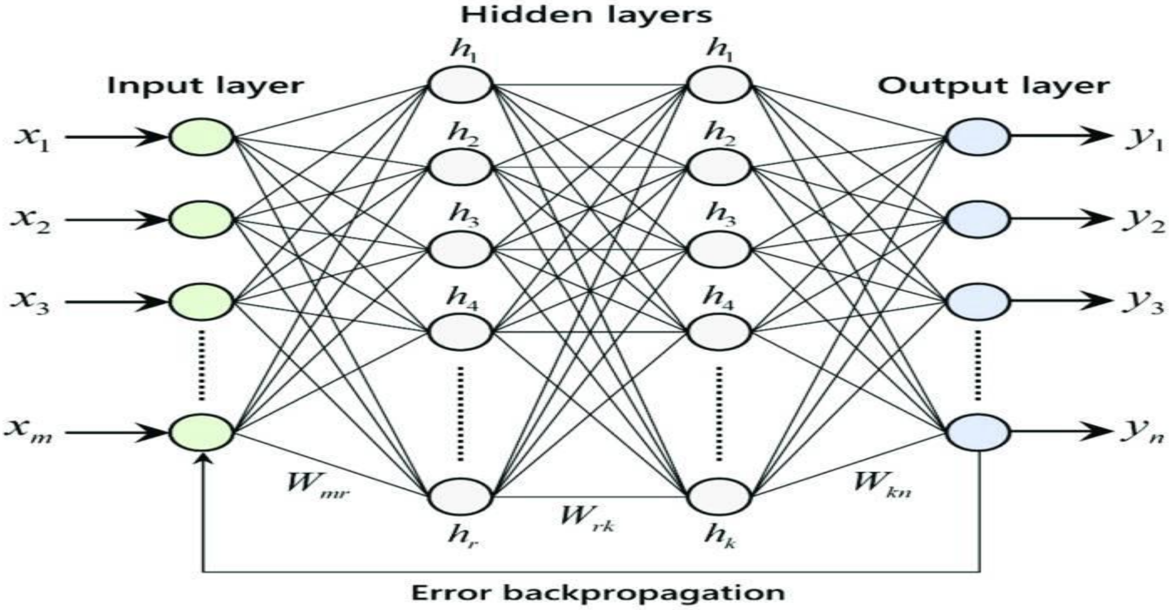


Figure 2- 2: Deep Neural Network structure

2.2.2 Convolutional neural network

A convolutional neural network (CNN) [20] is a feed forward neural network. CNNs has been primarily designed to solve problems related to image recognition. These models, unlike ordinary feed forward neural networks, use convolutions in place of general matrix multiplication. It biologically inspired them variants of ordinary neural networks. CNN architectures are specialized for learning data that has a grid-like topology. Primarily, CNNs assume that the input data are images [20]. This allows them to encode some properties in the network structure. A convolutional layer is the basic building block of a CNN. The parameters of a convolutional layer comprise a set of small learn- able filters. When learning we convolve each filter across the image and take a dot product. The network learns filters that activate when they "see" some visual feature, for example, vertical edges, or wheel-like patterns. Convolutions build on the observation that, in images, global features are constructed from combinations of local features hierarchically. We have discussed the connectivity between units and their parameter sharing scheme.

They often insert a pooling layer between two convolutional layers. The primary function of the pooling layer is to provide translational invariance. It consequently reduces the spatial size of the representation of the image. Doing this also reduces the network parameters, thus limiting the effect of over fitting. The most common operation used for pooling is the MAX [21] operation. A very common form of the pooling layer is a pooling layer with a filter of size 2×2 applied with a stride of 2. In this case, it performed the max over 4 numbers and it discards 3 to 4 of the activations. Besides MAX, the pooling layer can also use other functions like average pooling, L2-norm pooling, or global average pooling.

2.3 EfficientNet

Convolutional Neural Networks (ConvNets) are commonly created at a fixed asset budget and after that scaled up for way better precision in the event that are more assets are accessible. According to [9] EfficientNet model has showed better accuracy and effectiveness than past ConvNets. Scaling up ConvNets is broadly used to realize way of better precision. Where ResNet can be scaled up from ResNet-18 [22] to ResNet-200 by utilizing more layers; As of late, GPipe accomplished 84.3% ImageNet top-1 precision by scaling up the standard shows four times bigger. The method of scaling up ConvNets has never been well caught on and there are as of now many ways to do it.

It is possible to scale up ConvNets by increasing the depth or width. Another less common, but progressively well-known, the strategy is to scale up models by picture resolution. Thus, width, depth and resolution of pictures in combination have considered as a scaling parameter for ConvNet models. Although it is conceivable to scale two or three measurements subjectively, self-assertive scaling requires repetitive manual tuning and still regularly yields sub-optimal accuracy and efficiency.

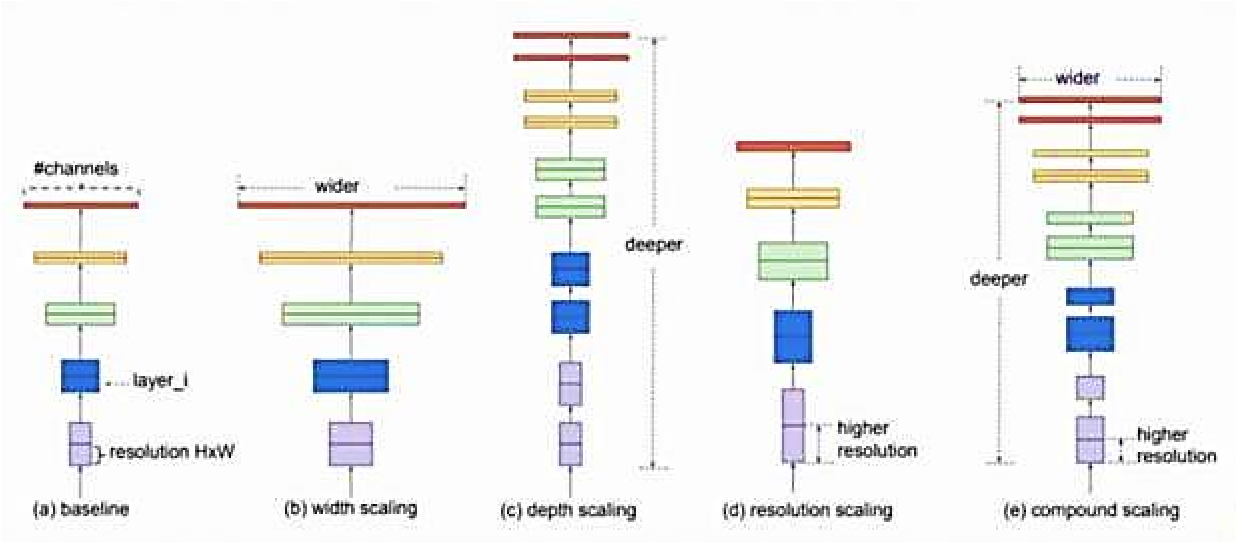


Figure 2- 3: Efficient model scaling parameters

EfficientNet uses a new compound scaling method, which uses a compound coefficient φ to consistently scale network depth, width, and image resolution in a mathematical principled way. Choice of determination, depth and width are moreover confined by many variables as shown in Figure 2- 3: Efficient model scaling parameters. As a result, the profundity, width and determination of each variation of the EfficientNet models has been hand-picked and showed to create great outcomes.

Compound Scaling is the combination of increasing the width, adding more layers, and increasing the input image resolution.

$$width : w = b^{i} \tag{2.1}$$

$$depth: \quad d = a^{i} \tag{2.2}$$

$$resolution: \quad r = y^{i} \tag{2.3}$$

$$a . b^{2} . y^{2} \approx 2 \tag{2.4}$$

$$a \geq 1, b \geq 1, y \geq 1 \tag{2. 5}$$

The author in [9] has proposed a straightforward, but successful scaling strategy that employments a compound coefficient *is* to consistently scale organize width, depth, and resolution in a principled way. It may be a user-defined, worldwide scaling factor (numbers) that controls how many assets are accessible through α, b, and y in decide how to relegate these resources to arrange depth, width, and resolution individually. The FLOPS(Flouting point operation per second) of a convolutional operation are corresponding to d, w^2, r^2, since doubling the profundity will double the FLOPS whereas multiplying width or resolution increments FLOPS nearly by four times. So, scaling the network using condition 3 will increment the full FLOPS by $(\alpha * b^2 * y^2) \wedge i$ subsequently, in order to form beyond any doubt that the whole FLOPS don't exceed 2^i, the constraint $(\alpha * b^2 * y^2) \approx 2$ is applied. What this implies, is that in the event that we have twice the resources accessible, we can basically utilize a compound coefficient of 1 to scale the FLOPS by 2^1.

The parameters i, b, and y- in equation 2.1 can be decided using framework look by setting ϕ=1 and finding parameters that result within the best accuracy. Once found, these parameters can at that point be settled, and the compound coefficient i can be expanded to induce bigger but more accurate models.

2.4 Recurrent Neural Networks

Recurrent neural networks (RNNs) [23] are powerful models which have been used to generate and model sequences in various domains like text, image captions and music. RNNs can process sequential data one step at a time. RNNs can also generate new sequences by iteratively sampling from the network's output distribution and using the output as the input for the next step. A fundamental limitation of feed forward neural networks is their inability to process variable length inputs. Feed-forward neural networks, both "vanilla" neural networks and convolutional neural networks, can accept a fixed size input vector and produce a fixed size output vector. Unlike these, recurrent neural networks are models which allow operations over a sequence of vectors. The network is designed to have cyclic connections. The recurrent units have connections that form a directed cycle with the hidden state. Figure 2- 4 below shows the generic architecture of simple recurrent neural network.

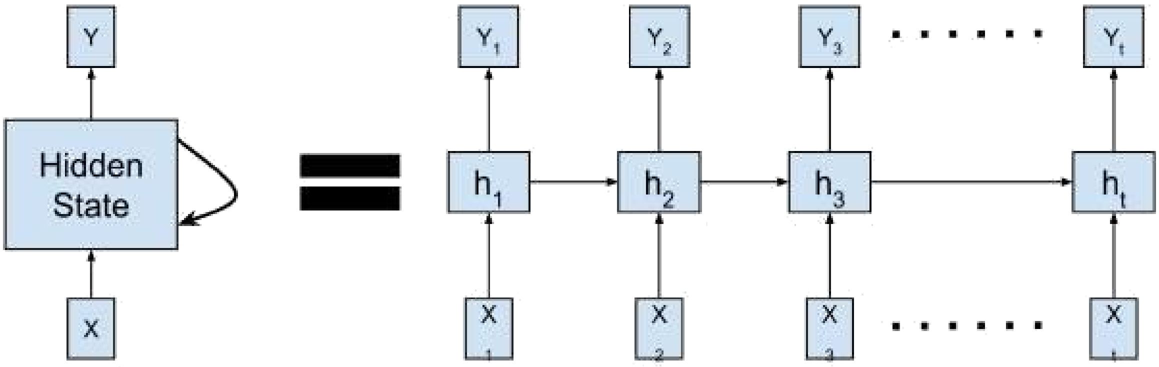


Figure 2- 4: General Structure of Recurrent Neural network

Though RNNs are very rich and dynamic, it is difficult to train them to modeling term dependencies. This is in part because of the vanishing and exploding gradient problem. because of propagation of the gradient through many layers of the unfolded recurrent network. LSTM [25] and GRU [10] provides a very effective solution to the vanising gradiant problem.

2.4.1 Vnishing Gradients Problems

The back-propagation algorithm [24] works by computing the gradients regarding every neuron and back propagating the gradient from the output layer all the way to the input layer. Unfortunately, as the computation goes through layers, gradients often get increasingly smaller, and eventually vanish. As the gradient vanishes, the stochastic gradient descent algorithm will not have any clue on how to update the parameters, especially the parameters in the early layers, thus the training will stop progressing and the model will not be useful. It made major progress on understanding and handling this issue in [25] where the author suggests that the saturation of the non-linear activation function is a major cause of the vanishing gradients problem. At that time, the dominate activation function that was used by most researcher in the field of neural networks is Logistic function and Hyperbolic tangent. Figure 2- 5 below shows the vanishing gradient problem.

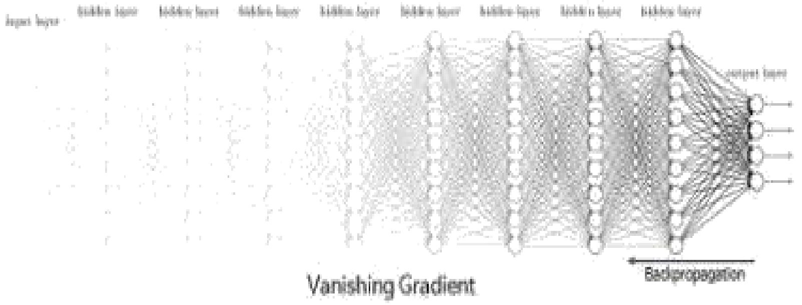


Figure 2- 5: The Vanishing Gradients Problems

2.4.2 Long Short – Term Memory (LSTM)

Long Short-Term Memory (LSTM) [26] networks are a changed version of the RNN architecture. Hoch Reiter Schmidhuber introduced them in 1997. Many people have since refined and popularized the LSTM architecture. LSTMs long-term dependencies. The LSTM changes the cell state using various operations, namely forget and input gates and the previously hidden state. It changes the hidden state using information from the cell and output gate.

The forget gate governs how much information is kept from the previous time step. It looks at the previous hidden state and the current input and outputs a number between 0 and 1. This number determines the amount of information that is kept from the previous cell state. The LSTM then has to decide what new information has to be added to the cell. It does so use the input gate and a candidate state generated using the input and hidden state of the LSTM units at the previous time step.

The input gate, like the forget gate, modulates how much of the new candidate state percolates into the cell. Finally, using the forget gate, input gate, and the new candidate state the cell is updated. Next, the LSTM needs to decide what to output. The output of the LSTM is the updated hidden state. The update to the hidden state is controlled by two factors: the current cell state that we updated as stated above and an output gate. The current cell state is passed through a hyperbolic tangent function to get a value between -1 and 1. The output gate, like the for-get and input gates, determines how much the cell state will affect the new hidden state and has a multiplicative interaction.

The control flow of LSTM is similar to that of RNN; the difference is in their internal control flow. Cell states are the core concept of LSTM and act as an information transport highway that carries information all the way down to the sequence chain. Important information from the early time steps will travel to the later time steps by doing so it will reduce the effect of short-term memory.

2.4.2.1Forget gate

Information that is going to be thrown away or kept is determined by this gate. Forget gate controls how the information from the current input x(t) and the output from the previous time step h(t-1) in the current cell after an activation function has been calculated. Following the equation 2.6 it outputs a vector f(t) with all elements between 0 and 1. This vector points which information is allowed to pass. It accepts input information from the previous hidden state and the current input through the sigmoid function. Since it is a sigmoid function it will give the value between 0 and 1. When the value is closer to 0 it means to forget and when the value is closer to 1 it means to keep. Figure 2- 6 below shows LSTM forget gate .

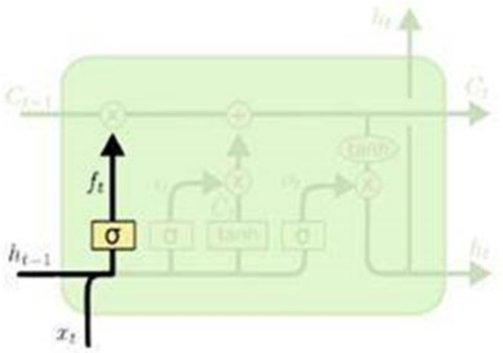


Figure 2- 6: LSTM forget gate

$$f_t = a\left(w_f . [h_{t-1}, x_t] + b_f\right)\#2.6$$

2.4.2.2Input Gate

The current cell has a state and in order to update this cell state we use an input gate. The input gate decides how much new input information should be added to the cell state. First, a sigmoid function decides which information needs to be updated, and a tanh function generalizes the Ct which means the contents available for update. Then, the old cell state Ct-1 in equation 2.8 can be replaced by adding new information into the cell state and get C(t). In this gate we use both of the activation function the tanh and the sigmoid function first it accepts the hidden state and current input which have the value between 0 and 1 closer to 0 means it is not that much important and forget. Closer to 1 means it is important information so keep it. After this the hidden state and the current input also pass through the tanh function in order to regulate the network. Finally, the output of the tanh function is multiplied by the output of the sigmoid function by doing so the sigmoid function output will decide what to keep or forget from the output of the tanh function. By this process the input gets updated to the current cell status

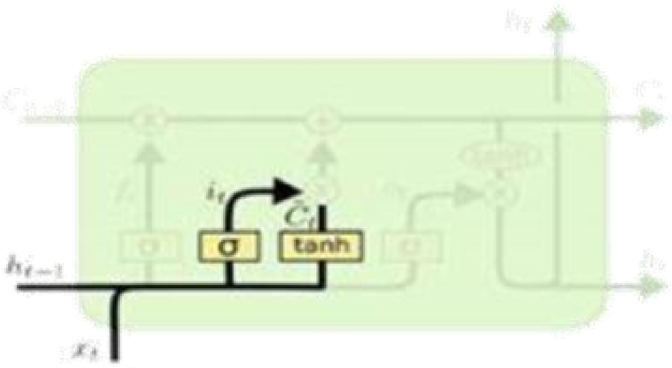


Figure 2- 7: LSTM input gate

$$i_t = a(w_i.\left[h_{t-1,x}\right] + b_i) \qquad 2.7$$

$$C_t = tanh\,(W_c.[h_{t-1}, x_t] + b_c \qquad 2.8$$

2.4.2.3Cell State

Information passes to all the cells in the LSTM using the cell state. This cell state is responsible for carrying information throughout the network. After it came from the previews cell it is multiplied by the output of the forget gate which means if the output of the forget gate is closer to 0 it means that it will forget most of the information from the previews gate if the output of the forget gate is closer to 1 it pass most of the information from the previous cell. Now in order to calculate the cell state we have all the necessary information the new cell state we gate by multiplying it by the forget gate is now concatenated with the output of the input gate it updates the cell state to a new value which will pass to the next cell.

2.4.2.4Output gate

All the information that is needed to pass to the next cell is calculated since the cell state is calculated from the forget gate and the input gate. Now we have to calculate the hidden state for the next cell and we do this by the output gate. The sigmoid function accept the current hidden state and the current input the newly modified cell state is pass to the tanh function for regulating the network then the output of the tanh is multiplied by the sigmoid output which is calculated by the current hidden state and current input, by this multiplication we decide which information have to pass to the next hidden state as shown in equation 2.9 and 2.10. The final output of a single cell is the next hidden state and the cell state.

$$o_t = \partial(W_0.[h_{t-1}, x_t] + b_0 \qquad 2.9$$

$$h_t = o_t * tanh\,(C_t) \qquad 2.10$$

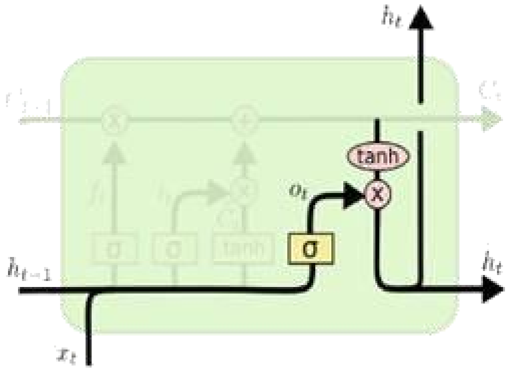


Figure 2- 8: LSTM output gate

To generalize the overall working principle of the LSTM from the perspective of each gate, forget gate decides what to keep based on the previous information, input gate decides what information has to be added to the cell state based on the current information. The output gate determines what information have to be included in the next hidden state which will be given to the next cell.

2.4.3 GRU (Gated recurrent units)

Gated recurrent units are another variant of gated RNNs proposed by Khandelwal, Lecouteux, & Besacier [10], which are designed to be simpler than LSTM in terms of computation. While discussing in terms of simplicity, GRUs [10] do not incorporate memory cells, they have reset gates instead. These reset gates allow the hidden state to leave the unimportant information and thus, focusing on the quality of content. Further, the update gate controls the flow of information from the hidden state in terms of quantity, which makes the process compact.Though GRUs act the same as LSTM in terms of learning the long term dependencies, they are expected to perform efficiently in terms of computation.

GRUs have been so far used in many prediction and classification tasks where they were able to outperform many traditional models. For example, auto regressive integrated moving average (ARIMA) [27] model was outperformed by GRUs for traffic flow. Other examples where GRUs performed better than state-of-the-art RNNs and LSTM models are session recommendation system for language modeling and sequence modeling [28].

In the task of video classification, [29] used multi-rated GRUs as encoders to encode the frames of YouTube2Text video dataset. Their model coupled with Res-Net-200 features produced an accuracy of 34.45% on Meteor which was comparatively higher than CNNs and LSTM. The data they experimented had variable input length of sequences similar to the dataset used in the present study.

Previously we have seen that in LSTM the cell state is used to transfer information to all the cells from the beginning to the end but in GRU the cell state is removed and the hidden state is used instead of the cell state to transfer information to all the cells. GRU only has 2 gates, the reset gate and the update gate and unlike LSTM GRU have only one output. Now let us see the two gates in detail. Figure 2- 9 below shows GRU’s update and reset gate.

$$z_t = \partial(W_z.[h_{t-1}, x_t]) \tag{2.11}$$

$$r_t = \partial(W_r.[h_{t-1}, x_t]) \tag{2.12}$$

$$\dot{h}_t = \partial(\,tanh\,(W.[r_t * h_{t-1}, x_t]) \tag{2.13}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \dot{h}_t \tag{2.14}$$

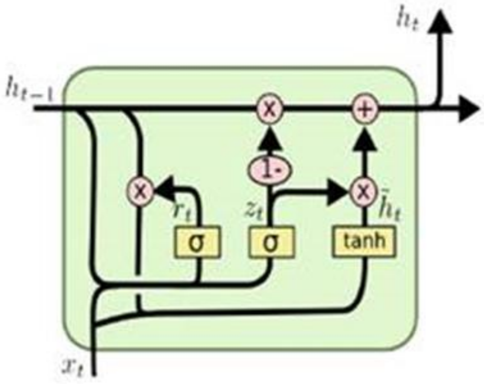


Figure 2- 9: Update and reset gate of GRU

2.4.3.1Tanh

This activation function is used to regulate the network value throughout the network. In Figure 2- 10 the tanh [30] furcation makes the value always to be between -1 and 1. When a vector which is converted from image, text or voice passes through the network it undergoes different mathematical operations those mathematical operations make the same value to be insignificant. This problem is solved by the tanh function regulating the output of the network by ensuring all the values to be between -1 and 1.

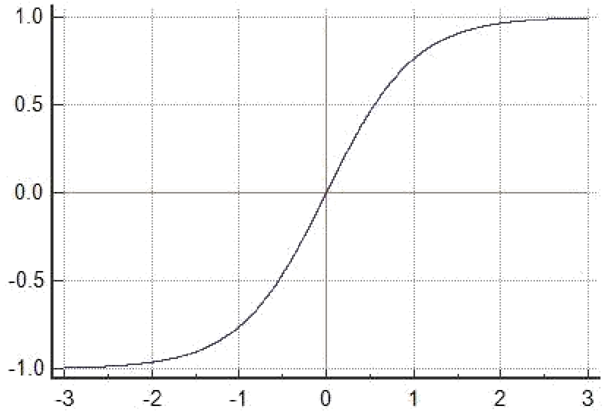


Figure 2- 10: Tanh activation function

2.4.3.2Sigmoid

The sigmoid [31] is similar to the Tanh function as shown in Figure 2- 11 instead of making all the values to be between -1 and 1 it makes the output to be 1 and 0 in other words it causes the values to disappear whenever the value is 0. And it causes the output to be kept, whenever the value is 1. By doing so the network learns which data to keep and which data to forget.

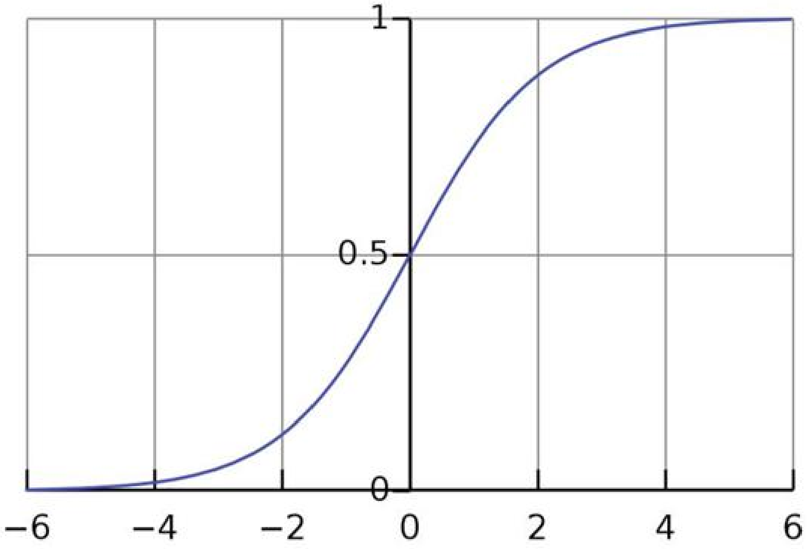


Figure 2- 11: sigmoid activation function

2.4.3.3Update Gate

This gate determines what information to forget and what information to keep based on the current information on the cell. Similar to the LSTM input gate. In order to do so it concatenates the previous hidden state with the current input and feeds it to the sigmoid function which will give a result between 0 and 1. In this gate the information from the previous hidden state and the current information are used to generate a sigmoid output.

2.4.3.4Reset Gate

This gate determines how much information to forget from the past information. This gate affects the final output of the cell; it follows the following steps to do so.

Step 1. First, we multiply the previous hidden state by the reset gate output. Remember the reset gate output is the output of a sigmoid activation function which means between 0 and 1 multiplying the previous hidden state by this value means we are determining what information should pass.

Step 2. Then we add the result of step 1 and the concatenation of the previous hidden state and the current input and finally we apply the tanh function in order to regulate the network.

Step 3. This is the final and the critical step for the reset gate, in this step we are going to calculate the current hidden state unlike LSTM here in GRU the current hidden state does not only store the cell state of the current cell it also stores the information that should pass to the next cell respective of the previous and current information. There are two measure steps that should be taken. First we have to perform an element wise vector multiplication of the output of the update gate and the output we gate from step 2 then since we want to calculate the info that need to pass to the next cell it is stored in the update gate output (z(t)) when we subtract it from 1 we know how much information should not pass to the next cell (1- z(t)) so we multiply the previous hidden state by one minus the output of the update gate then we sum up the result with the result we gate from step one and this finally give us the current hidden state or the memory at current timestamp which is also passed to the next cell.

2.5 Literature Review

2.5.1 Introduction

In this section, different published works related to pornographic video classification is assessed. The classification is done in two main approaches first using the image and video processing, second using deep learning methods. At the end of this section, we summarize attempts that have been discussed in this section.

2.5.2 Pornographic video classification using image processing methods

In [32] uses fast motion features to classify pornographic videos. For each video they performed shot change detection from a single clip they extracted 20 frames and fast motion feature was computed for each clip. A threshold value is determined and if it is greater than that threshold value then the video will be classified as porn else it is not porn. They are able to get an accuracy of 88.5% on the publicly available NPDI [11] 800 datasets. This study has a major drawback in correctly classifying pornographic videos because it has many false positive and negative since it only takes in to consideration motion information it will easily misled because other than pornographic videos there are many types of non-pornographic videos with repetitive motion like boxing, swimming etc. in the other hand we do also have many porn videos without repetitive motion so at this time it will give us false negative result.

In [29] propose repetitive motion to detect videos just like the one in the above paragraph. They select a clip then extract the frame then they calculate the motion vector from the extracted frames using Hamming window [33]. The authors in [29] has reported two significant contributions first, they have increased the repetitive motion detection performance by 10%. Secondly, they have reduced the computational complexity in detecting repetitive motion form a video like the previous work they have determined a threshold value for classifying it as porn or non porn. If the calculated motion vector value is greater than the threshold value it is porn otherwise it is not porn. Aside from the contribution of this study it is still suffering from false positive and false negative problem. We have identified gaps from this work as; there are many instances where repetitive motion occurs in the short section without the video a hole being indecent. All repetitive motion is not pornographic video and all pornographic videos does not have a repetitive motion.

In [34] identifies reciprocating motion form pornographic video using Hidden Marcov model (HMM) [35]. First, the motion vector is obtained by decoding the compressed MPEG video then the feature vectors are extracted by calculating the direction and magnitude of the motion vector then the extracted feature is given to the Hidden Markov Model for training and classification of the action. They only classify six action Porn, walk, run, jump, basketball and Trampoline and they are able to get an accuracy of 90%. The main gap in this study is that it is not possible to detect pornographic videos; which include reciprocating action other than the listed classes. The other problem is that all pornographic videos do not have the same motion so it is hard for their model to detect pornographic videos with less motion and with unfamiliar motion.

In [11] they have introduced a special temporal interest point detector and descriptor with the help of temporal Robust feature (TRoF). They aggregated local information which is extracted by TRoF into a mid-level representation using fisher vectors. The main contribution of this study is the creation of a pornographic video dataset which contains 2000 videos of nearly 140 hours of 1000 pornographic and 1000 non-pornographic videos [11]. This dataset contains all kinds of pornographic videos from animation to live porn videos; it also includes all ethnic groups. The Main drawback of this study is that space-temporal leads to more effective pornographic video classification but it needs high computational time and memory footprint so it makes it hard to apply dense strategies space-temporal approach on hardware, which does not have that much Capacity.

2.5.3 Pornographic video classification using Deep learning methods

The first attempt to use deep learning for pornographic video classification has reported by [21]. They apply the convolutional neural network to solve pornographic images and video Classification. They use a combination of two convolutional neural network models the AlexNet[37] and the GoogleNet [38]. AlexNet was trained using 1.2 million images and it has 1000 different classes. The other is GoogleNet this one is deeper than AlexNet. They combine these two convolutional Networks to classify pornographic images and video frames (image extracted from a video) they have used the NPDI [11] pornographic image and video dataset. In this study, they do not use pre-trained future extractor and classifier for pornographic videos and images. They do not even consider other features from the pornographic videos.

In [39] they have used CNN to extract future and to classify frames extracted from Pornographic videos. They apply face detector and age classification to infer the age group represented in different images which are extracted from the pornographic videos. They get a majority vote from the individual frames to decide whether it is porn or not porn and then they try to detect a face from those frames and when they find one, they will feed it to the age classifier. They have used the NPDI [11] dataset for training and testing. There are mainly two drawbacks on this study. The first one is detecting face from pornographic videos is not an efficient way because it is hard to get clear and suitable faces from the frames of the video. Where the second one is, they have limited the input to a static frame extracted from each video they did not take the temporal information provided from the frames.

According to [40] they have incorporated the motion information and deep learning architecture. They use optical flow and MPEG motion vectors [41] for combining the static frames and the dynamic (motion) information and they have used the NPDI 800 training and testing pornographic video dataset. In the static information first they choose sampling of the video frame and extract their future using convolutional network they feed the extracted feature to the SVM [42] classifier from the final classification after the static information the motion information extract the motion information from the feature extracted from the static information using MPEG motion vector those extracted information are changed to feature extraction using generate image representation and they are feed to motion CNN classification the result is then used for prediction they have got an accuracy of 97.9. The main gap in this study is that the motion information may create a high false negative for repetitive videos other than porn videos like tense playing, boxing, bike riding. And they do not consider the relationship between frames for a better classification.

In [36] they have proposed a deep learning method using both convolutional and recurrent neural networks to classify pornographic videos. They perform their experiment on the dataset provided from the NPDI [11] pornographic video dataset. They have used a pre-trained ResNet50 [22] Convolutional neural network to extract a feature from the frames. Before extracting the feature from the image, they have used 10 corps for every image which improves the resulting accuracy. After they extract the feature, they have used extracted from the frames. LSTM [26] as a sequence learner causes values of Yn to be generated by using information from all previous frames F(t) for all the elements between 1 and n-1. They are able to achieve a 99.0% accuracy level. The LSTM sequence learner is more robust in identifying adult content in scenarios with large skin exposure but some pornographic videos do not have that much long skin exposure so at that time it will perform low and using LSTM have high computational latency with respect to our dataset the NPDI 2K [11] pornographic dataset.

Table 2.1 Compression of different Literature done on Pornographic videos

Author	Algorithm	Accuracy	Dataset	Method	Year
[32]J.-J. Yu,	Repetitive motion	85.5%	NPDI	Image Processing	2015
[29]J. Garcia	Repetitive motion Hamming window	-	-	Image Processing	2008
[34]Z. Qu	Hidden Marcov model (HMM)	90.0%	NPDI 800	Image Processing	2009
[11]A. Gangwar	Temporal Robust feature (TRoF).	-	NPDI 800	Image Processing	2017
[21]K. He	CNN	94.0%	NPDI 800	Deep learning	2015
[39]J. Jung	CNN	94.1%	NPDI 800	Deep Learning	2017
[40]B. Petrovska	Motion CNN	97.9%	NPDI 2K	Deep Learning	2017
[36]J. Wehrmann	CNN and RNN(LSTM)	99.0%	NPDI 2K	Deep Learning	2018

2.5.4 Summery

The research by [32][29][34][11] use image processing to classify pornographic videos as we have seen they have manly focus on the repetitive information that can be obtained from a video so they have used different methods to extract this repetitive motion and then formulate a threshold value to classify videos in to porn and non porn. The problem comes when we encounter a non porn video with repetitive motion and a porn video without repetitive motion. From this we can understand only just by using image processing we cannot accurately classify a pornographic video. The researches [21][39][40][36] use deep learning. We can see how accuracy has changed when a deep learning is introduced to the pornographic video classifications especially when motion information is included it produces good results. All the researchers use the same dataset which is best for comparing the results. We discuss more about the dataset in the upcoming chapters. From the above literature review we can conclude that the integration of deep learning for porn video classification has good accuracy and when motion information is added with deep learning [36] it shows the best accuracy.

CHAPTER 3. PORNOGRAPHIC VIDEO CLASSIFICATION SYSTEM DESIGN AND ARCHITECTURE

3.1 Introduction

In this section, a detailed description of the designed architecture of the proposed deep learning model has explained. In which manner does the architecture performs and what crucial steps are taken to tackle the video classification task. Moreover, essential frames are extracted from the video and the model takes an extracted feature from these extracted frames as an input to classify whether or not the video is a pornographic. The overall pipeline of the proposed model is comprised of the following processes:

- Data preparation for feature extraction
- Fine-tuned feature extraction from prepared frames and
- Prediction by learning the sequence information from the extracted features.

3.2 Architecture of Pornographic video classifier

As we have mentioned earlier in section 4.1 the proposed model has three main procedural components as shown below in Figure 4.1. The architecture classifies pornographic video out of non-pornographic video by only using visual features that have been performed well in small dataset and low processing power of the machine. Transfer learning has been used in order to solve the problem of having small dataset and low processing power in which each component processed independently. The first component, which is called data pre processing, has been accomplished independently on NPDI public 2K dataset [11]. Following this, the Fine-Tuned Feature extraction has employed by using EfficientNet; which is one of a Transfer learning models. Lastly, the training was undertaken using GRU and LSTM in order to learn the sequence information in the video.

The extracted frames for the publicly available dataset are extracted and stored in a permanent storage in order to feed it to the fine-tuned feature extractor. Before feeding to the fine-tuned feature extractor, necessary pre processing has been done on the extracted frames.

In order to extract the visual feature, a per-trained EfficientNet model has been used and the extracted feature information is saved on a permanent storage. The extracted features used by the sequence learner to decide the final class of the video. Each component will be discussed in the next section. Figure 4- 1 below shows the proposed architecture of CNN and GRU based pornographic video classification.

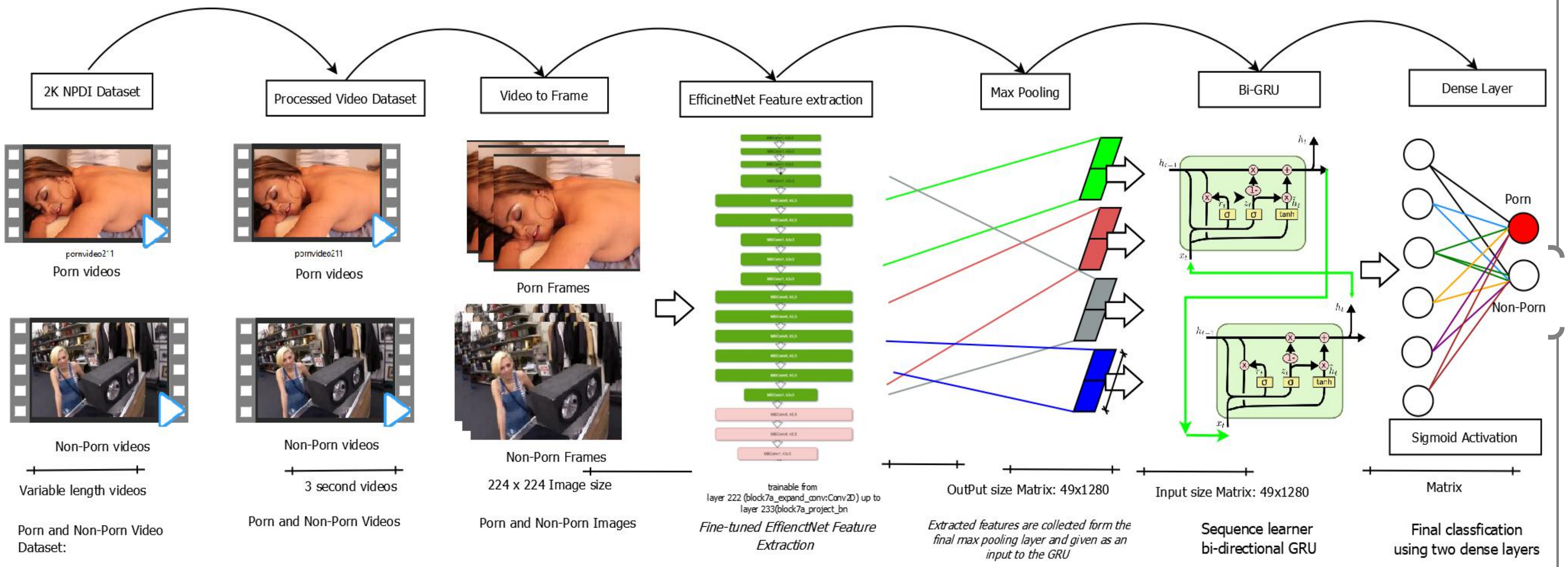

2K NPDI Dataset
Processed Video Dataset
Video to Frame
EfficinetNet Feature extraction
Max Pooling
Bi-GRU
Dense Layer
pornvideo2'1
Porn videos
pornvideo211
Porn videos
Porn Frames
Non-Porn videos
Non-Porn videos
Non-Porn Frames
Porn
Non-Porn
Sigmoid Activation
Variable length videos
3 second videos
224 x 224 Image size
trainable from
layer 222 (block7a_expand_conv:Conv2D) up to
layer 233(block7a_project_bn
OutPut size Matrix: 49x1280
Input size Matrix: 49x1280
Matrix
Porn and Non-Porn Video
Dataset:
Porn and Non-Porn Videos
Porn and Non-Porn Images
Fine-tuned EffiencNet Feature
Extraction
Extracted features are collected form the
final max pooling layer and given as an
input to the GRU
Sequence learner
bi-directional GRU
Final classfication
using two dense layers
h_{t-1}
h_t
x_t
r_t
z_t
$\tilde{h}_t$
σ
σ
tanh
1-

Figure 4- 1 Architecture of pornographic video classification using CNN and GRU

The first part of the architecture is the fine-tuned feature extraction part. EfficientNet [43] has been used to extract important features from the generated frames from the video clips of the NPDI 2K Pornographic video datasets. Those extracted features are aggregated to a single video and given to the GRU [10] sequence learner. GRU sequence learner is a modified version of RNN which eliminates the problem of vanishing gradients. According to [26] GRU has proven to be the best method for learning sequence information from video data. Thus, we have adopted the GRU model for the sequence labeling and learning task. To do so, by accepting those extracted features, the GRU learn sequence information from the provided input and predicting the video class.

3.3 Dataset

Dataset is an integral part of the field of machine and deep learning powered applications[44]. We strongly believe that whenever there is a clear and accurate data, it is not too hard to go in hands with good deep learning powered solutions. The NPDI 2K [11] publicly available dataset is collected from the university of Campinas. Due to the data sensitivity, too big and associated copyright issues, an agreement sign were handled in between Jimma Institute of Technology, Faculty of Electrical and Computer Engineering and university of Campinas. Thus, in this study we have used this publicly available dataset. The argument signed between Jimma Institute of Technology (JiT) and the University of Campinas (UoC) has presented in Appendix D.

The statistics of the dataset distribution shows it consists of 2000 pornographic and non-pornographic videos with total length of 140 hours of 40GB size. The dataset is composed of different ethnic groups. Animated pornographic contents are also included. Sample frames are listed below in Figure 4- 2 just to show the type of videos available frames at the top are all taken from pornographic video. Frames at the second and third row are taken from non-pornographic videos.

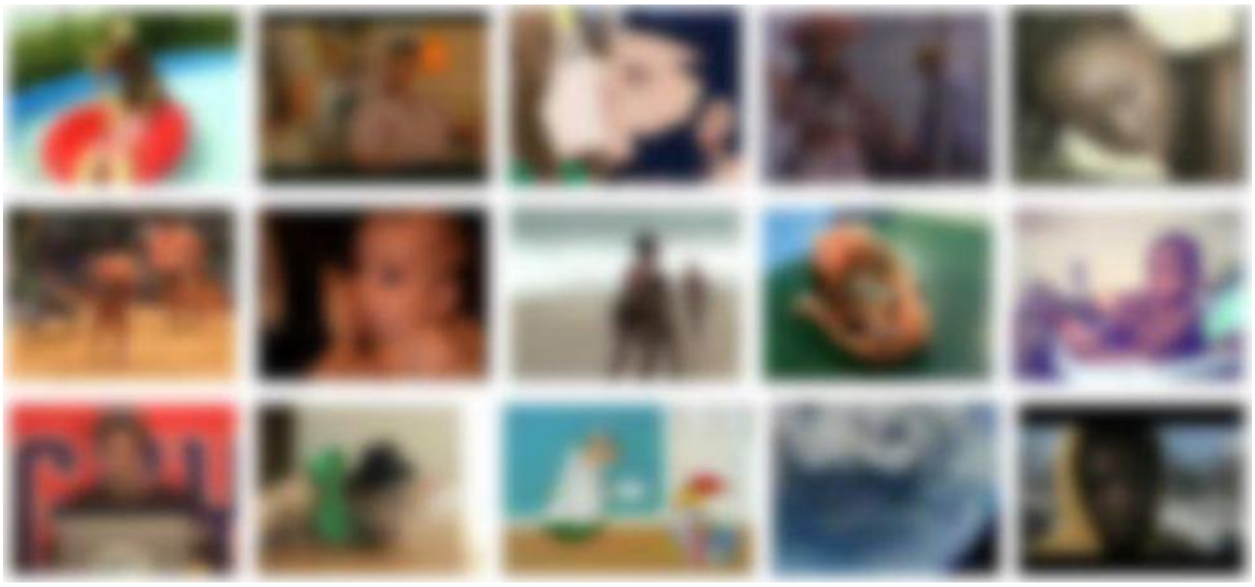


Figure 4- 2: Images from NPDI Pornographic video dataset

To maximize the potential of classifying the videos as whether pornographic or not, we believe that it is mandatory to included pornographic and non-pornographic video of different ethnic background. This could help to balance distribution of our dataset in order to perform best results with the proposed model. Thus, the NPDI dataset consists videos of different ethnic background compositions. Numerically it is described as follow in Table 4- 1 and

Table 4- 2.

Table 4- 1: Number of videos and hour in the NPDI Pornographic dataset

Class	**Video**	**Hour**
Porn	1000	64
Non-Porn	1000	76
Total	**2000**	**140**

Table 4- 2: Ethnic diversity in the NPDI Pornographic dataset

Ethnicity	**% of Videos**
Asian	16%
Black	14%
White	46%
Multi - ethnic	24%

Dataset preparation and preprocessing steps that were performed in this study are described in the sections 4.4.

3.4 Data preparation

Data preparation or data pre processing is a process of transforming raw or unprocessed data to meaningful insights where it is at ready state to feed into the machine. [45] An appropriate pre processing has been done on the video by applying a relevant pre processing techniques. In order to feed the video to the feature extractor, short videos from our video dataset is required. We have selected two parts of certain video in order to take a 3 second video [11] [72] from each part of the video. Which are the 25th % and 75th % of the total length of the video duration. By applying this technique for the whole video dataset we have gotten 4000 a three seconds videos with the total duration of 3.33hrs. let, if the video length is 7 minute our video extractor will extract a 3 seconds video from 1:05 – 1:08 (25th%) and 5:25 – 5-28 (75th%) by doing this we are able to have 4000 videos of each 3 second.

By having those 3 seconds videos, we have extracted the sequence of frames from the video –clips. To convert the videos into the sequence of frames we have used Opencv2 library; which is one of computer vision libraries. By extracting 20 frames/second, we got 60 frames from each three second videos. Thus, from those all 4000 a three second videos, we have got 240,000 frames. Generally, to get a 3 second videos and to extract an essential features from the videos, we have used opencv2 [46] and FFMPEG [47] together from the on shelf python libraries.

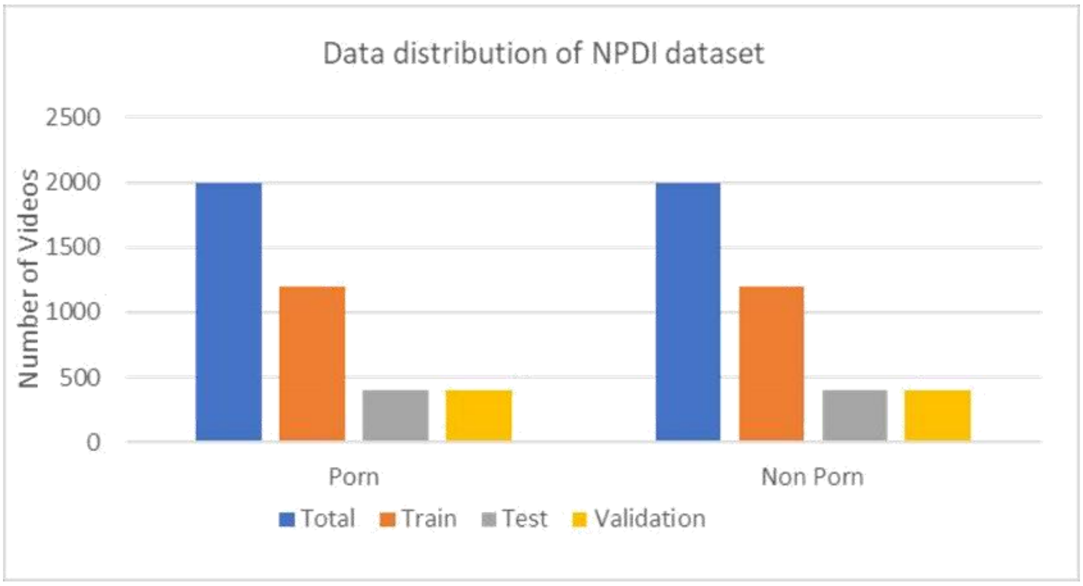


Figure 4- 3: Data distribution of NPDI dataset

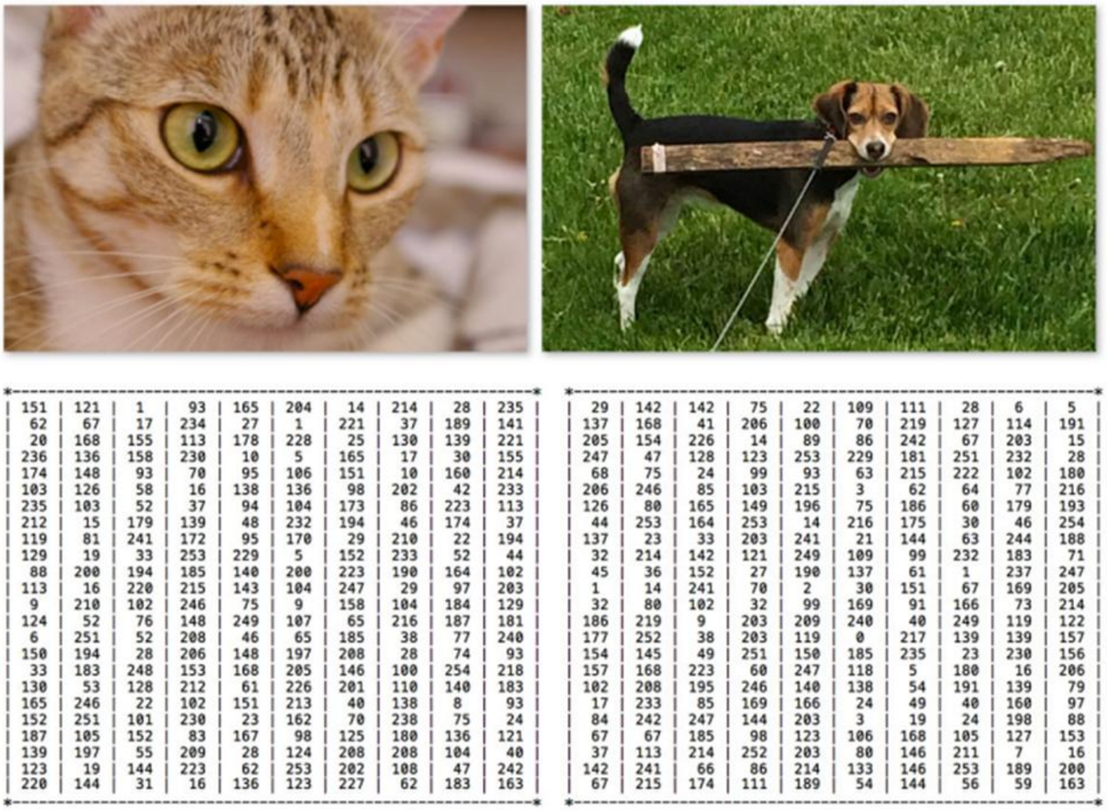
Finally, the total frames extracted from the 2000 videos are divided into three set. Train set, test and validation set. The frames extracted from 1200 videos are for training, 600 for testing and 400 for validation. The video-clips extracted from both pornographic and non-pornographic videos are stored in a folder named as per their class respectively.

Table 4- 3: Frames extracted from the NPDI dataset

Class	**Total Video**	**Short Video (3 Second)**	**Extracted frames form short videos**
Porn	1000	2000	120,000
Non-Porn	1000	2000	120,000

3.5 Fine-Tuned Feature Extraction

Take a look at the two photos in Figure 4-4. It should be fairly trivial for us to tell the difference between the two photos – there is clearly a cat on the left and a dog on the right. But all a computer sees is two big matrices of pixels (bottom). Given that all a computer sees is a big matrix of pixels, we arrive at the problem of the semantic gap. The semantic gap is the difference between how a human perceives the contents of an image versus how an image can be represented in a way a computer can understand the process. Again, a quick visual examination of the two photos below can reveal the difference between the two species of an animal. But in reality, the computer has no idea there are animals in the image to begin with.



```
*-----------------------------------------------------*
| 151 | 121 |   1 |  93 | 165 | 204 |  14 | 214 |  28 | 235 |
|  62 |  67 |  17 | 234 |  27 |   1 | 221 |  37 | 189 | 141 |
|  20 | 168 | 155 | 113 | 178 | 228 |  25 | 130 | 139 | 221 |
| 236 | 136 | 158 | 230 |  10 |   5 | 165 |  17 |  30 | 155 |
| 174 | 148 |  93 |  70 |  95 | 106 | 151 |  10 | 160 | 214 |
| 103 | 126 |  58 |  16 | 138 | 136 |  98 | 202 |  42 | 233 |
| 235 | 103 |  52 |  37 |  94 | 104 | 173 |  86 | 223 | 113 |
| 212 |  15 | 179 | 139 |  48 | 232 | 194 |  46 | 174 |  37 |
| 119 |  81 | 241 | 172 |  95 | 170 |  29 | 210 |  22 | 194 |
| 129 |  19 |  33 | 253 | 229 |   5 | 152 | 233 |  52 |  44 |
|  88 | 200 | 194 | 185 | 140 | 200 | 223 | 190 | 164 | 102 |
| 113 |  16 | 220 | 215 | 143 | 104 | 247 |  29 |  97 | 203 |
|  9  | 210 | 102 | 246 |  75 |  9  | 158 | 104 | 184 | 129 |
| 124 |  52 |  76 | 148 | 249 | 107 |  65 | 216 | 187 | 181 |
|  6  | 251 |  52 | 208 |  46 |  65 | 185 |  38 |  77 | 240 |
| 150 | 194 |  28 | 206 | 148 | 197 | 208 |  28 |  74 |  93 |
|  33 | 183 | 248 | 153 | 168 | 205 | 146 | 100 | 254 | 218 |
| 130 |  53 | 128 | 212 |  61 | 226 | 201 | 110 | 140 | 183 |
| 165 | 246 |  22 | 102 | 151 | 213 |  40 | 138 |  8  |  93 |
| 152 | 251 | 101 | 230 |  23 | 162 |  70 | 238 |  75 |  24 |
| 187 | 105 | 152 |  83 | 167 |  98 | 125 | 180 | 136 | 121 |
| 139 | 197 |  55 | 209 |  28 | 124 | 208 | 208 | 104 |  40 |
| 123 |  19 | 144 | 223 |  62 | 253 | 202 | 108 |  47 | 242 |
| 220 | 144 |  31 |  16 | 136 | 123 | 227 |  62 | 183 | 163 |
*-----------------------------------------------------*
```

```
*-----------------------------------------------------*
|  29 | 142 | 142 |  75 |  22 | 109 | 111 |  28 |  6  |  5  |
| 137 | 168 |  41 | 206 | 100 |  70 | 219 | 127 | 114 | 191 |
| 205 | 154 | 226 |  14 |  89 |  86 | 242 |  67 | 203 |  15 |
| 247 |  47 | 128 | 123 | 253 | 229 | 181 | 251 | 232 |  28 |
|  68 |  75 |  24 |  99 |  93 |  63 | 215 | 222 | 102 | 180 |
| 206 | 246 |  85 | 103 | 215 |  3  |  62 |  64 |  77 | 216 |
| 126 |  80 | 165 | 149 | 196 |  75 | 186 |  60 | 179 | 193 |
|  44 | 253 | 164 | 253 |  14 | 216 | 175 |  30 |  46 | 254 |
| 137 |  23 |  33 | 203 | 241 |  21 | 144 |  63 | 244 | 188 |
|  32 | 214 | 142 | 121 | 249 | 109 |  99 | 232 | 183 |  71 |
|  45 |  36 | 152 |  27 | 190 | 137 |  61 |  1  | 237 | 247 |
|  1  |  14 | 241 |  70 |  2  |  30 | 151 |  67 | 169 | 205 |
|  32 |  80 | 102 |  32 |  99 | 169 |  91 | 166 |  73 | 214 |
| 186 | 219 |  9  | 203 | 209 | 240 |  40 | 249 | 119 | 122 |
| 177 | 252 |  38 | 203 | 119 |  0  | 217 | 139 | 139 | 157 |
| 154 | 145 |  49 | 251 | 150 | 185 | 235 |  23 | 230 | 156 |
| 157 | 168 | 223 |  60 | 247 | 118 |  5  | 180 |  16 | 206 |
| 102 | 208 | 195 | 246 | 140 | 138 |  54 | 191 | 139 |  79 |
|  17 | 233 |  85 | 169 | 166 |  24 |  49 |  40 | 160 |  97 |
|  84 | 242 | 247 | 144 | 203 |  3  |  19 |  24 | 198 |  88 |
|  67 |  67 | 185 |  98 | 123 | 106 | 168 | 105 | 127 | 153 |
|  37 | 113 | 214 | 252 | 203 |  80 | 146 | 211 |  7  |  16 |
| 142 | 241 |  66 |  86 | 214 | 133 | 146 | 253 | 189 | 200 |
|  67 | 215 | 174 | 111 | 189 |  54 | 144 |  56 |  59 | 163 |
*-----------------------------------------------------*
```

Figure 4- 4 Image vs correspondent feature

We might describe the image as follows:

- Spatial: The cat is at the top of the image and the dog is at the bottom.
- Color: The grass is dark green; the dog is a lighter dark than the wood, while   the cat is yellow.
- Texture: The cat has a relatively uniform pattern, while the grass is very coarse.

How do we go about encoding all this information in a way that a computer can understand it? The answer is to apply feature extraction to quantify the contents of an image. Feature extraction is the process of taking an input image, applying an algorithm, and obtaining a feature vector (i.e., alist of numbers) that quantifies our image.

Fine-tuning technique has been employed in the process of feature extraction since, it is the process in which parameters of a model must be adjusted very precisely in order to fit with certain observations. When performing feature extraction we did not re-train the original CNN. We treated the CNN as an arbitrary feature extractor and then trained a simple machine learning model on top of the extracted features. In order to feed into the machine, essential features have to be extracted from the videos. To embed spatial information, we have used EfficientNet network as building block in order to extract frame level spatial features from the overall frames of 2K NPDI [11] video dataset.

In order to achieve Fine-tuning we have followed the following steps(process) Removing the fully connected layers at the end of the network(i.e: where the actual class label predictions are made) then replacing the fully connected layers with freshly initialized ones. Then freeze earlier CONV layers earlier in the network (ensuring that any previous robust features learned by the CNN are not destroyed. Starting training, but only train the full connected layers heads Optionally unfreeze some/all of the CONV layers in the network. The general layer structure of Efficient Net B0 model has shown in Appendix C.

```
for layer in eff_model.layers[:221]:
     layer.trainable = False
for layer in eff_model.layers[222:234]:
     layer.trainable = True
```

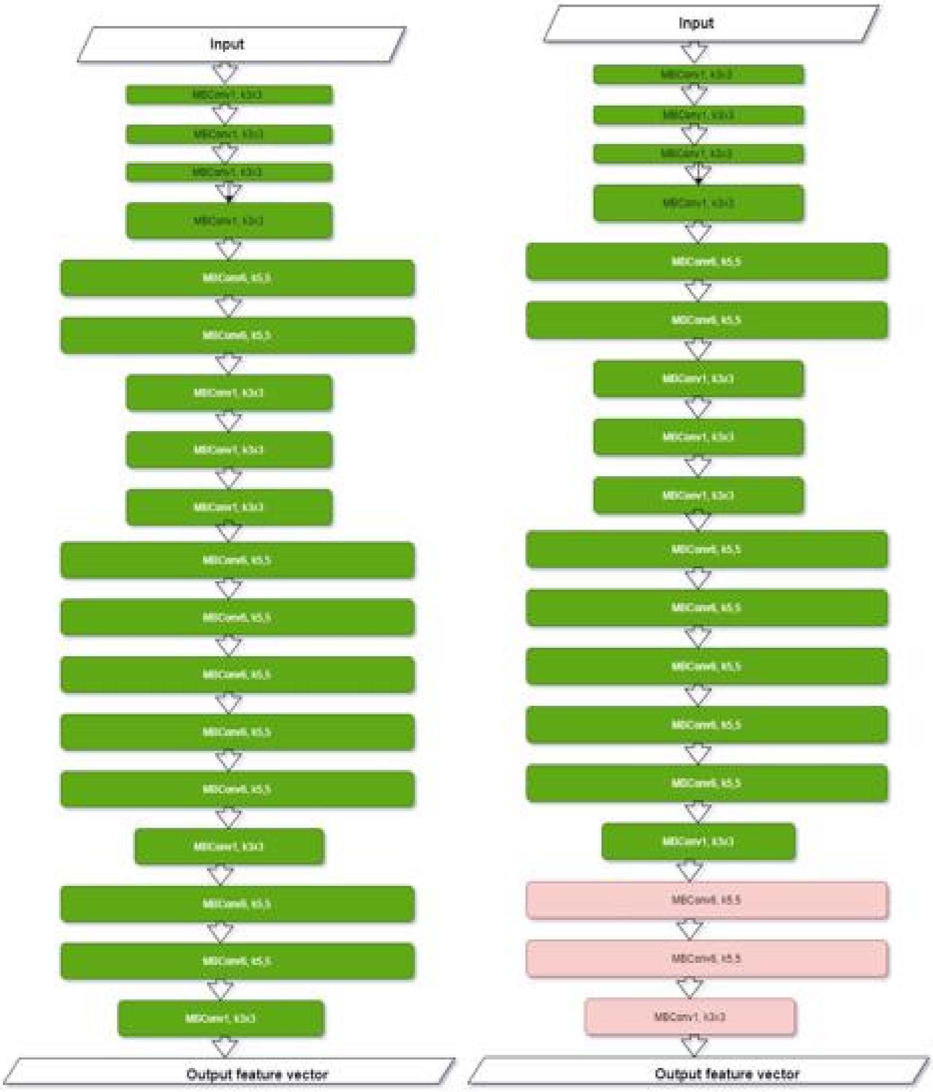


Figure 4- 5 : Fine-tuned feature extraction

In Figure 4- 5 left side: When we start the fine-tuning process, we freeze all CONV layers in the network and only allow the gradient to back propagate through the FC layers. Doing this allows our network to "warm up". Figure 4- 5 Right: After the FC layers have had a chance to warm up, we may choose to unfreeze all or some of the layers earlier in the network and allow each of them to be fine-tuned as well. We have set the last block to be trainable from layer 222(block7a_expand_conv:Conv2D) up to layer 233(block7a_project_bn).

The CNN layer is applied to extract visual features and the RNN layer also used to encode the sequential relationship between frames of video and classify the sequence features within the video. Due to the little amount of data we ought to prepare for the network, overfitting was our number one challenge and In order to fight this overfitting, rather than preparing the network from scratch, we preferred to use pre-trained EfficentNet model for feature extraction. EfficentNet is pre-trained on ImageNet Large scale visual recognition challenge (ILSVRC) [29]. Unlike the previous traditional transfer learning neural networks, EfficentNet uses a Compound scaling which increases the network performance and ability for better feature extraction

Video feature extraction is the method by which we extract visual feature from the video. This process happens within the video feature extracting component. The video feature extraction handle starts by changing each images extracted from in each video to their respective RGB values. At that point the RGB values are given to a pre trained CNN so that visual features are extracted. Next, GRU layer comes to extract the successive data. Since we are working with a sequence of pictures (frames) we require the GRU layer to understand the relationship between the past frames and the present frame. By doing this, motion or development of the video may be classified into porn or non-porn.

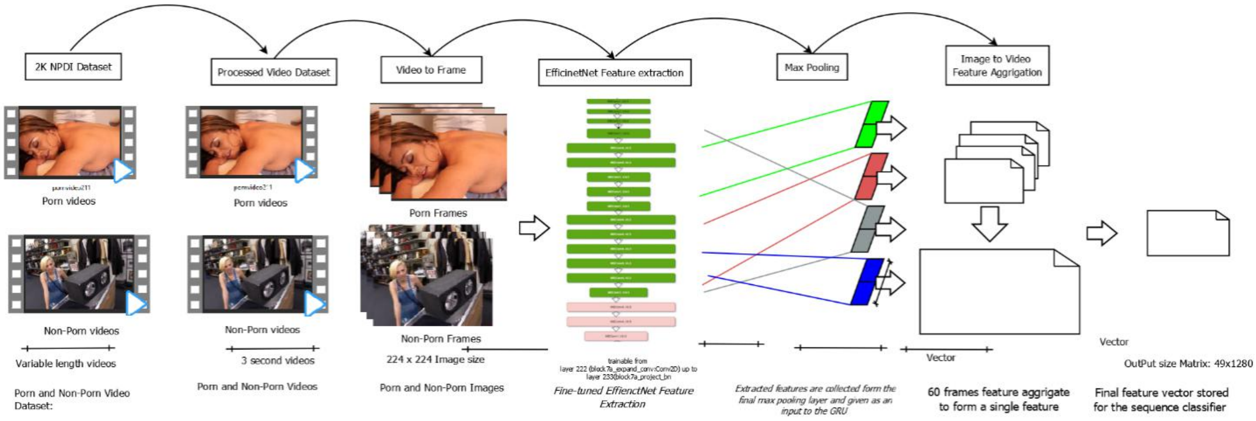
Within the video feature extraction preparation, we use transfer learning by applying pre-trained network as a way for extracting important features from frames of the video.. Since the last layers of pre-trained model is task specific, we have removed them and customized it to extract high level features from the video information. A pre-trained network may be a spare arrangement that was already trained on a large dataset, regularly on a large-scale image-classification challenge. The model that we are going to utilize to extract features is EfficeintNet. Frames from each video will pass through this pre-trained model shown in Table 4- 4. At that point features from each frame will be aggregated to make the entire video data. We have also used ResNet50 [25] and DenseNet [48] for comparison with EfficientNet and we have shown the result we get using all the three feature extractors at the result and discussion section.

Pre-trained network is a spare arrangement in which it was already trained on a large dataset basically on a large-scale image classification challenge. EfficientNet is the one which we used to extract features. The frames extracted from each video passes through this model and features from each frame get aggregated to make the entire video data. We have also applied ResNet50 [25] and DenseNet [48] in comparison to EfficientNet for the same target.

Table 4- 4: Efficient Net feature extraction model structure

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

From a collection of video clips V where each clip v ⊂ V contains a sequence of frames with a specific temporal order *{f1,..,fn}* and label *lv*. We pull out RGB frames to represent spatial information from the frame sequence (local) motion information. The RGB frames are processed at 20 fps from a single video. Distinct pee-trained EfficeinetNet layer network models are then used to extract temporal features *fva* form the frames extracted from the video. The EfficeinetNet pee-trained model on the ImageNet ILSVRC-2012 [49] dataset is responsible for extracting temporal feature. We have collected F*va* from the last Fully-Connected layer of the pee-trained model as the spatial embedding video vector sequences. The extracted feature has dimension: 49x1280.



2K NPDI Dataset
Processed Video Dataset
Video to Frame
EfficinetNet Feature extraction
Max Pooling
Image to Video
Feature Aggrigation
pornvideo211
Porn videos
pornvideo211
Porn videos
Porn Frames
Non-Porn videos
Non-Porn videos
Non-Porn Frames
Variable length videos
3 second videos
224 x 224 Image size
Porn and Non-Porn Video Dataset:
Porn and Non-Porn Videos
Porn and Non-Porn Images
trainable from
layer 222 (block7a_expand_conv:Conv2D) up to
layer 233(block7a_project_bn
Fine-tuned EffienctNet Feature Extraction
Extracted features are collected form the final max pooling layer and given as an input to the GRU
Vector
60 frames feature aggrigate to form a single feature
Vector
OutPut size Matrix: 49x1280
Final feature vector stored for the sequence classifier

Figure 4- 6: Feature extraction component

A frame picture size of (224, 224, 3) is at that point entered as the input for the EfficeintNet within the feature extractor. The output of EfficeintNet feature extractor is a vector in size of 49x1280 extracted as a static feature from a single frame, a grouping of 60 static features is extracted for a single video fragment composed of 60 examined frames. The image to video feature aggregation step comes here since we extract 49x1280 vector size of feature form a single frame and a single video have 60 frames the extracted feature of each frame is put in a single chunk which will be used as an input to the GRU cell. This single chunk of feature of a single video is feed to the GRU with Ts time interval of each feature. Each chunk features are given to the next sequence learner GRU form a single directory for a single video with a time interval of each frame feature. By doing this we have prepared the temporal feature of each video to be utilized by the sequence learner GRU.

3.6 Training Classifier

To capture temporal feature from a given video, it is possible to use Recurrent Neural Network (RNN) on the top of frame-level spatial features. However, basic FC-RNN is known to suffer from the vanishing gradient problem especially when the input time sequence goes longer. Vanishing gradient problem is encountered when training artificial neural networks with gradient-based learning methods and back propagation. In such methods, each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. As one example of the problem cause, traditional activation functions such as the hyperbolic tangent function have gradients in the range (0, 1), and back propagation computes gradients by the chain rule. This has the effect of multiplying n of these small numbers to compute gradients of the early layers in an n-layer network, meaning that the gradient (error signal) decreases exponentially with n while the early layers train very slowly.

To avoid these Vanishing gradient problem, basically, two well-known alternatives are used to avoid the problem which are Long-Short Term Memory (LSTM) and Gate Recurrent Unit (GRU) [50]. They can avoid the Vanishing Gradient problem by implementing "Gating Units" that decide what to ''forget' and what to 'Recall' or 'update'. In our approach we have chosen FC-GRU [25] to further encode frame-level spatial features, since it has comparative performance to FC-LSTM [26] and requiring less computation cost that is an important considerable factor for video analysis.

When working with complex sequence information like video data it is recommended to use stacked (bi-directional) units of GRU layers this will enable it to learn more long-term dependency [43]. Each features extracted through the above-mentioned method is utilized to prepare a visual feature. Given the spatial embedding video vector sequences f(va) encode each of them independently with two bidirectional FC-GRUs. The Fully-Connected Gated Recurrent Unit (FC-GRU) uses the reset gate rt and the update gate z(t) (both gates take values between 0 and 1 due to the logistic sigmoid activation function) to fuse input with previous memory and define how much of the previous memory to keep, thus avoiding the gradient vanishing problem while maintaining the power to discover temporal correlation.

The FC-GRU architecture shown in Figure 4- 7 that is used in this research contains 1024 hidden units for each FC-GRU hidden weight (the dimension of ht is 1024). We experimented with various numbers of units and choose 1024 to best trade off computation expense and performance.

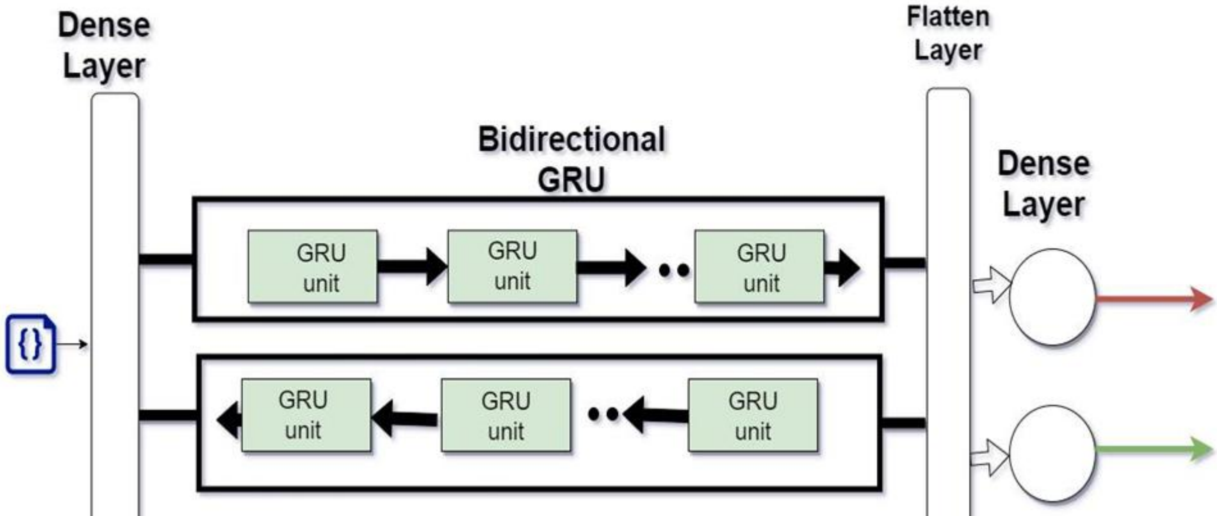


Figure 4- 7: Training the pornographic video classification using GRU

We then pooled the last hidden state h(t) as outputs for the RGB frames sequence. The outputs of our FC-GRU are the feature appearance of video representation having 1024 dimensions. Finally, a simple concatenation is performed to combine the output of the two FC-Bidirectional GRU representations at the video level. This results in a temporally representation of 2048 dimensions. To satisfactorily reflect the non-linearity of the high-dimensional classification hyperspace comprising of high-dimensional descriptor vectors, as a comparatively basic and non-linear learning schema, a multi-layer perception model comprising of an FC layer of size 1024 with the ReLU [51] activation function and another FC layer of size 2 with the SoftMax activation [52] are utilized. In most deep neural network classification, it is better to use SoftMax in order to have the final classification process. This last dense layer with the SoftMax activation layer outputs the probability distribution of the two classes (Porn and Non-Porn). In all classifiers, the FC layers (of size 2) at that point get 1024 inputs and deliver the final 2-dimensional output either a porn or a non-porn.

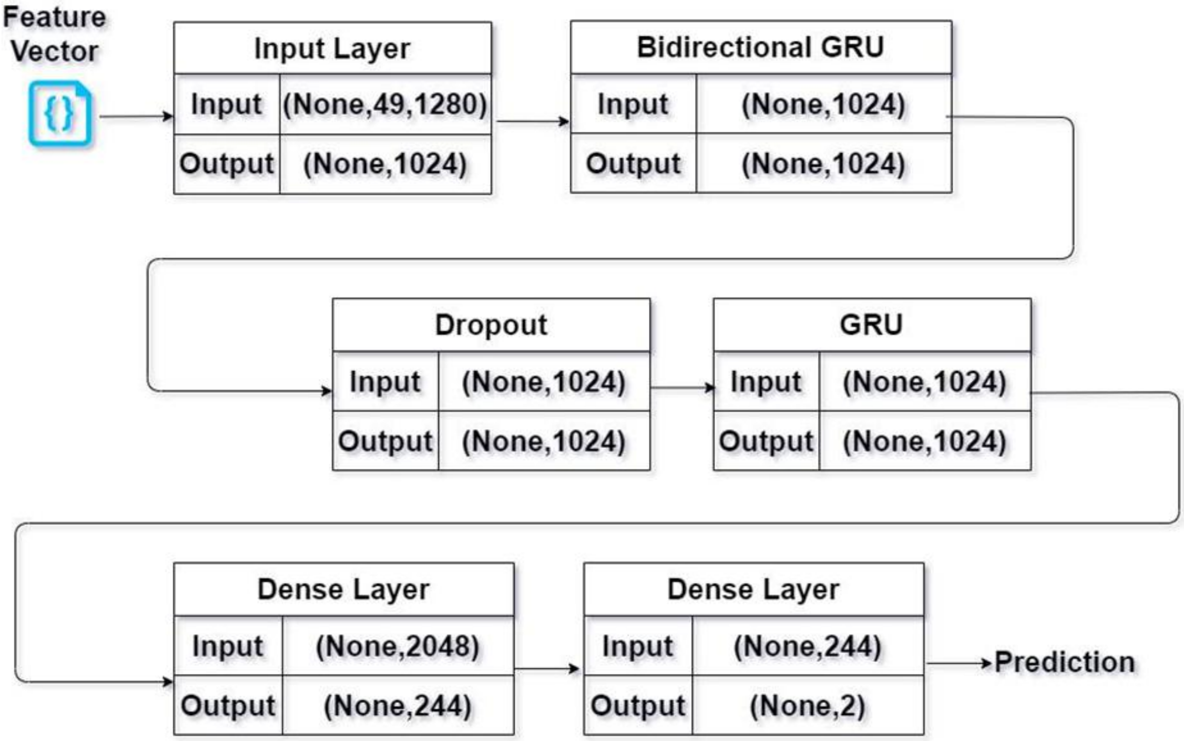


Figure 4- 8: Layer structure of the proposed Architecture

Since it is important to have a measuring method for the performance of the model so we have used categorical binary cross entropy to measure the loss in the distribution of prediction respect to the actual labeled classes or in the other word to measure the error rate. We have defined the optimizer to be Adam, and the learning rate to 0.0001.

3.7 System evaluation techniques

There are many supervised learning algorithms methods to train and measure the accuracy of the model. The data set is split into three groups: training, validation and testing. The first is the set of data that the model trains and learns, while the second is used to provide an unbiased assessment of the fit model on the training data set while tuning the model hyper-parameters. After training, the test set is used to validate the model with data that has not been before. In some instances, the cost of setting aside a large portion of the data set as required by the holdout approach is too high. In these cases, a re-sampling based technique used, such as cross validation as a solution instead. The k-fold validation method shown in Figure 4- 9 is used for our project. The complete training set is split into k folds (subsets) in this approach. A combination of k–1 folds used to train the model over k iterations, while the remaining fold used for validationpurposes. An average score used to measure the generalization error. Figure 4.6 describes the cross-validation technique used in this study

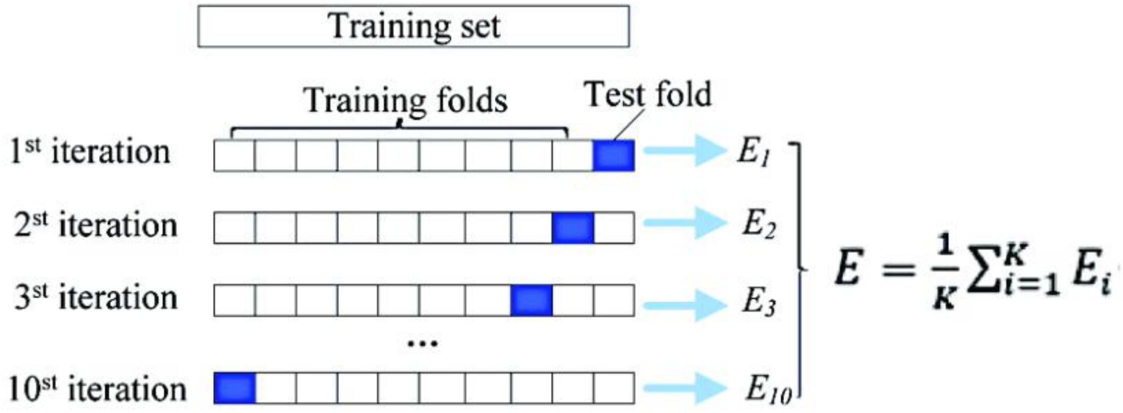


Figure 4- 9: K fold cross validation

To perform the project model measurements, k = 10 and four standard performance metrics are determined: Accuracy, Precision, Recall, F1 score and confusion matrix [53]. These four measures are used in deep learning, artificial intelligence, sentiment analysis, and compute vision.

True positive: When evaluating the results of classification, an actual positive rate measures the number of instances that have classified in a category and belong to that category.

True negative: The real negative rate describes the number of cases that have not organized in a class, and that efficiently does not belong to that class.

False positive: The false positive rate indicates the number of instances that have been classified in a category but do not belong to that class.

False negative: The false negative rate indicates the number of instances that have not classified in the category but that do belong to that class.

3.7.1 Accuracy

Accuracy is one evaluation metric of Machine Learning models[54]. It measures the classifier's performance by determining the ratio of the correct number of trials, which are achieved by the classifier over the total number of trials. The formula for accuracy is shown in equation 4.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \qquad 4.1$$

3.7.2 Precision

Is the number of classes correctly marked as belonging to a positive class divided by a total number element which are labeled as belonging to the positive class[54]. First it is labeled to be a positive class and dividing it by the total number of elements labeled by the model into the positive class. The value of precision indicates how much the system returns the correct classes rather than the incorrect one. Having a higher precision value means our model is giving us correct values in equation 4.2.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad 4.2$$

3.7.3 Recall

It is the number of relevant instances forecasted over the total relevant instances[54]. In general recall is used to measure the ability of our model to recognize all the entities in the dataset. If the result of Recall value is high, it means that the model predicts the most relevant result.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad 4.3$$

3.7.4 F1 score

It is the combined measure of precision and recall[54]. In other words, it is the harmonic mean of precision and recall the balanced F1 score is calculated as follows.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad 4.4$$

3.7.5 Confusion Matrix

The confusion matrix is a method used for evaluating the performance of a classification algorithm [54]. The number of correct and incorrect forecast are given by count values and decomposed by a group. It gives intution not only into the errors produced by the classifier but also, most importantly, into the types of errors which have occurred

CHAPTER 4. EXPERIMENTAL RESULT AND DISCUSSION

4.1 Software and hardware tools

In this study, we have used different software tools and packages. Python is a general-purpose programming language commonly used in this study. Python has various standard machine learning library to manipulate multimedia data on specific tasks. The main Python modules which has been used in this study include Scrapy, IPython, Jupyter Notebook, Pandas, Matplotlib, Scikit-learn, Tensorflow and Keras.

IPython and Jupyter Notebook:- IPython is an interactive python shell with a Jupyter Note book kernel that enables interactive code development in the browser[55]. It is commonly usedin data science and data processing, mathematical modelling, machine learning, numerical simulation and visualization. Throughout this study, Jupyter Notebook used for data processing, machine learning, analysis and visualization.

Scikit-learn:- Scikit-learn is a Python open-source machine learning, data mining, and data analysis library[56]. The library includes, among other items, a logistic regression model used as a basis for this study.

Panads: Pandas is an open-source library for manipulation of data and analysis [57]. Convenient handling of data structures and high performance. Pandas have used throughout this study for the processing and processing of data and data analysis.

Matplotlib:- Matplotlib is a Python module devoted to plotting figures, graphs and diagrams [58]. It used for visualization in combination with Jupyter Notebook.

TensorFlow:- TensorFlow is an open-source library for artificial intelligence and numerical computing which was created by Google [59]. The library is particularly well suited for deep learning, where the scalable architecture enables computation to apply to one or more CPUs or GPUs. Tensorflow has been used as a backend program to facilitate deep learning in this study.

Keras:- Keras is a high-level API that can operate with TensorFlow and other deep learning libraries [57]. The main focus is to facilitate efficient prototyping and experimentation. The

Keras API used to build the hybrid CNN-GRU model with several architectures analyzed in this study

Numpy1.19.3: it is a python module used for powerful mathematical operations and we have used it to perform same mathematical operations in the image file

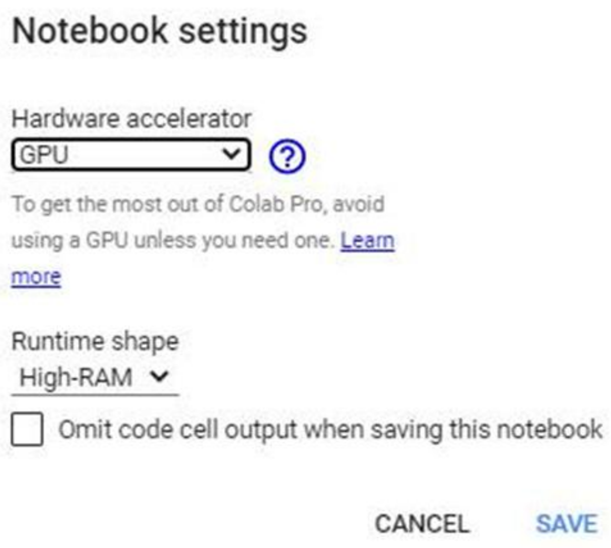
OpenCV: library built in by C++ for computer vision operations and we have used it for preprocessing the video files [60].

Ffmpgi: it is a video processing tool which we used for extracting frames from the video in different particular time [47].

4.2 Experimental setup

One of the key challenges we have been experienced when working with computer vision and related tasks using deep learning models is requirement of high performance computing device. Beside this, requirement of high storage devices has been also challenging to implement this study. In order to solve the need for a high-performance environment, we have used a special computing environment developed by Google called Colab Pro; which has 24 GB RAM and 135 GB storage. We have used Colab pro has totally avoided session crashing problem because colab pro provided 2x of the normal colab environment. In Figure 5- 1 it shows the type of runtime and other notebook settings.

The other reason we have used google colab pro is that it has a fast-training environment with GPU and since we are working on video/image data. It needs an outrageous amount of processing power so in this case google colab is the preferable environment for training the model. Same of the codes we have written in google colab are found in the Appendix A.



Notebook settings

Hardware accelerator

GPU

To get the most out of Colab Pro, avoid using a GPU unless you need one. Learn more

Runtime shape

High-RAM

□ Omit code cell output when saving this notebook

Figure 5- 1: Colab Pro Notebook setting

4.3 Evaluation of pornographic video classification using deep learning

This section runs smoothly with evaluating the performance of proposed and developed pornographic video classification system by using 2K NPDI [11] pornographic video dataset as an input. The standard evaluation metrics what have been discussed in chapter 4 are used here in order to evaluate our model's accuracy, precision, recall, F1 score and confusion matrix. The evaluation techniques helps to evaluate the combined efficiency of using GRU together with EfficientNet feature extraction method on NPDI dataset [11]. We can evaluate how efficiently those model perform in extracting certain features from the pornographic videos and classify them according to their respective class.

4.3.1 Hyperparamters

All learning algorithms, even simple ones, require hyper-parameters to be determined by the practitioner. Neural networks share many hyper-parameters like activation function, learning rate, optimization, and dropout . In addition to all shared parameters for the neural network, each particular type of neural network still has more hyperpara meters. To optimize each hyperparameter, one can see either reason about sensible values for these parameters based on the learning task or complete a full grid-search on the model. A grid search is a step-by-step optimization of hyperparameters and is expensive for learning lessons with many input data. Our proposed model's structure has been modified for the following most frequently adjusted hyperparameters and network architecture parameters using a grid search. Table 5- 1 shows the different hyperparameters that tested in the grid search optimization.

Table 5- 1: Different values hyperparameters used during the grid search

Hyperparameter	**Values ranges**
Optimizer	SGD and Adam
Learning rate	0.1,0.01,0.001,0.0001,0.00001
Activation function	Relu,tanh, Sigmoid
Batch size	8,16,32,64,128
Number of epochs	10,20,30,40,50,60,70,80,90,111
Dropout	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.9

4.3.1.1Epoch

The number of epochs is a hyperparameter that determines the number of times the learning algorithm can operate through the entire training dataset. One epoch means each sample in the training dataset has had the chance to update the model's internal parameters. The epoch consists of one or more batches; we can think of a for-loop over the number of iterations where each loop continues over the training dataset. In this for-loop, there is another nested for loop that iterates across each batch of variables, where one batch has the number of observations defined in the batch size. We have shown the number of epoch for the given input sets and the results are shown in Appendix E.1 Epoch vs Accuracy Figure 0- 1.

4.3.1.2Optimization algorithm

An optimization algorithm is a procedure which is executed iteratively by comparing various solutions until an optimum or a satisfactory solution is found. There are two distinct types of optimization algorithms widely used today. Deterministic Algorithms: They use specific rules for moving one solution to other. These algorithms are in use to suite some times and have been successfully applied for many engineering design problems and Stochastic Algorithms: The stochastic algorithms are in nature with probabilistic translation rules. They do well in optimal problem formulation. Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. ADAM [61] (Adaptive Moment Estimation) optimizer was utilized for learning control and binary cross entropy was utilized for loss calculation. Parameter optimization is performed with Adam we have also work with SGD [62] for comparative analysis and shown the result in Appendix E Figure 0- 2.

4.3.1.3Batch size

The batch size is a hyperparameter that describes the number of units to work through before renewing the internal model parameters that means a batch as a for-loop was iterating over one or more pieces and making predictions at the end of the batch, the forecasts compared to the expected output variables as shown in Appendix E Figure 0- 3.

4.3.1.4Learning Rate

Another crucial issue in training neural network is the learning rate [63]. The Learning rate is one of the critical hyperparameters that maintains the speed at which the model parameters need to be updated to deduce a satisfactory output neural network. The process of obtaining optimum value can become an entangled effort in many practical problems. According to [55] The choice of a reasonable learning rate can turn out to be a difference between a model that is unable to learn from the data and a model that delivers outstanding results. On the other hand, a neural net with a small learning rate examines data slowly, requiring a significant amount of time to converge. On the other side, excessively high learning rates lead to weight dispersion and fluctuation around the minimum. The learning rate increases the training stage's potential to achieve a global minimum, eliminating the chance of being confined at a local minimum that could give rise to incorrect outputs as shown in Appendix E Figure 0- 4.

4.3.1.5 Dropout rate

dropout is a basic co-adaptation technique used during training to prevent neural networks from overfitting by co-adaptation [64]. This technique compared to a random sampling of function weights in a network, i.e. an unexpected decrease in some of the network's weight connections to prevent over-dependence on a single feature.The result we get while testing different dropout is shown in Appendix E Figure 0- 5

4.3.2 Hyperparameter Tuning

The hyperparameters explained in section 5.3.1 have been optimized using a grid search. The best parameters after the grid search shown in Appendix E. For each step of the grid search, all the hyperparameters fixed apart from the one that is optimized. The best hyperparameters summarized in

Table 5- 2 below.

Table 5- 2: Best Hyper-parameters

Hyper Parameter	**Value**
optimizer	Adamax
Learning rate	0.0001
Activation function	Sigmoid
Droupout rate	0.3
Batch size	32
Number of Epochs	50

After getting the best suitable parameters, we experimented our model in different scenarios or by using different feature extraction methods, and measured the performance in terms of accuracy and loss for both classes in the NPDI [11] dataset. The graphs in Figure 5-4 show the comparison between the four feature extraction methods for the two classes (Porn and Non-Porn) video classification. The graph shown at Figure 5-3 is the loss vs epoch for EfficeintNet feature extraction method. We get the best result when we use EfficeintNet as a feature extraction and combine it with the GRU in order to learn sequence information from the pornographic videos in the NPDI [11] dataset.

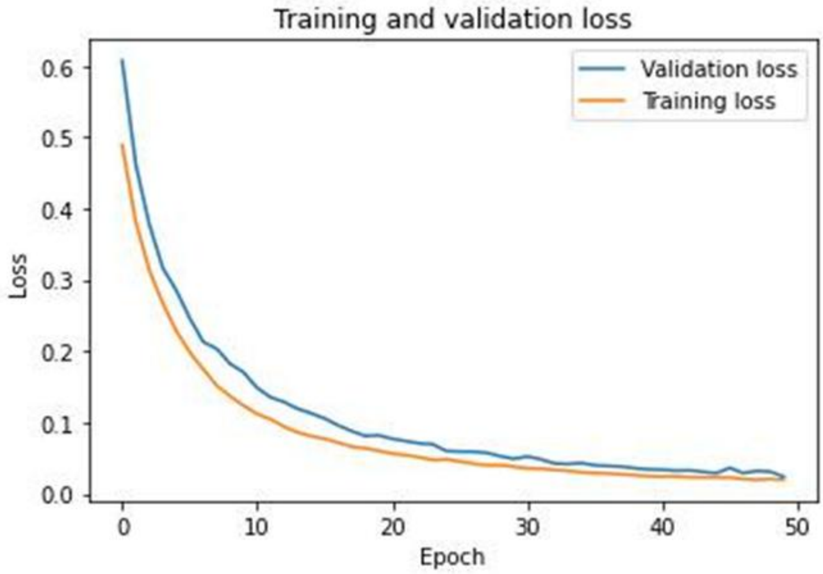


Figure 5- 2: Loss vs epoch of the model

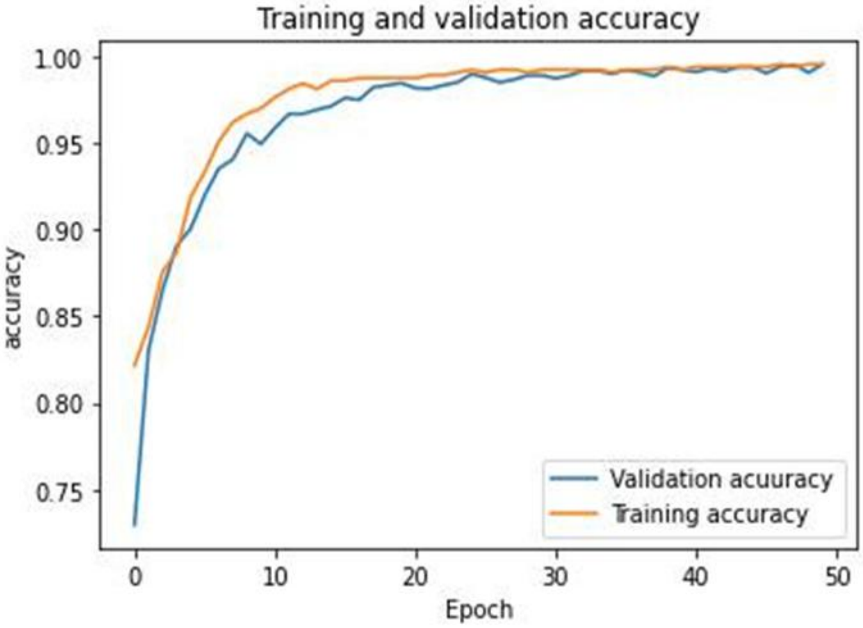
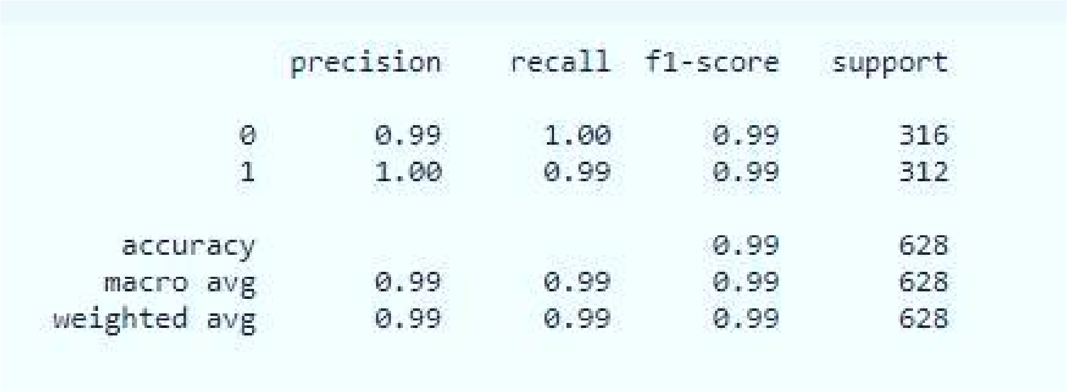


Figure 5- 3: Accuracy vs epoch of the pornographic video classification model

Figure 5- 2 shows the accuracy vs epoch graph for the use of EfficeintNet for feature extraction and GRU for learning sequence information. We have got the best result in this scenario. After the training is finished, the model is exported and tested on the test dataset with the accuracy shown in Figure 5- 2 and Figure 5- 3. In Table 5- 4 we are going to evaluate the precision, recall and F1 score from the prepared dataset for evaluation purpose.

Table 5- 3: The precision, recall, and F1-Score of porn video classification



```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       316
           1       1.00      0.99      0.99       312

    accuracy                           0.99       628
   macro avg       0.99      0.99      0.99       628
weighted avg       0.99      0.99      0.99       628
```

Table 5- 4: The precision, recall, and F1-Score of porn video classification

In Figure 5- 4 below, we try to show the loss for using different feature extraction methods combined with the GRU sequence information learner. We have used VGG 19, ResNet50 [25], Inception [38] and DenseNet [65] and shown below in Figure 5- 4 the loss vs epoch graph for each feature extraction method.

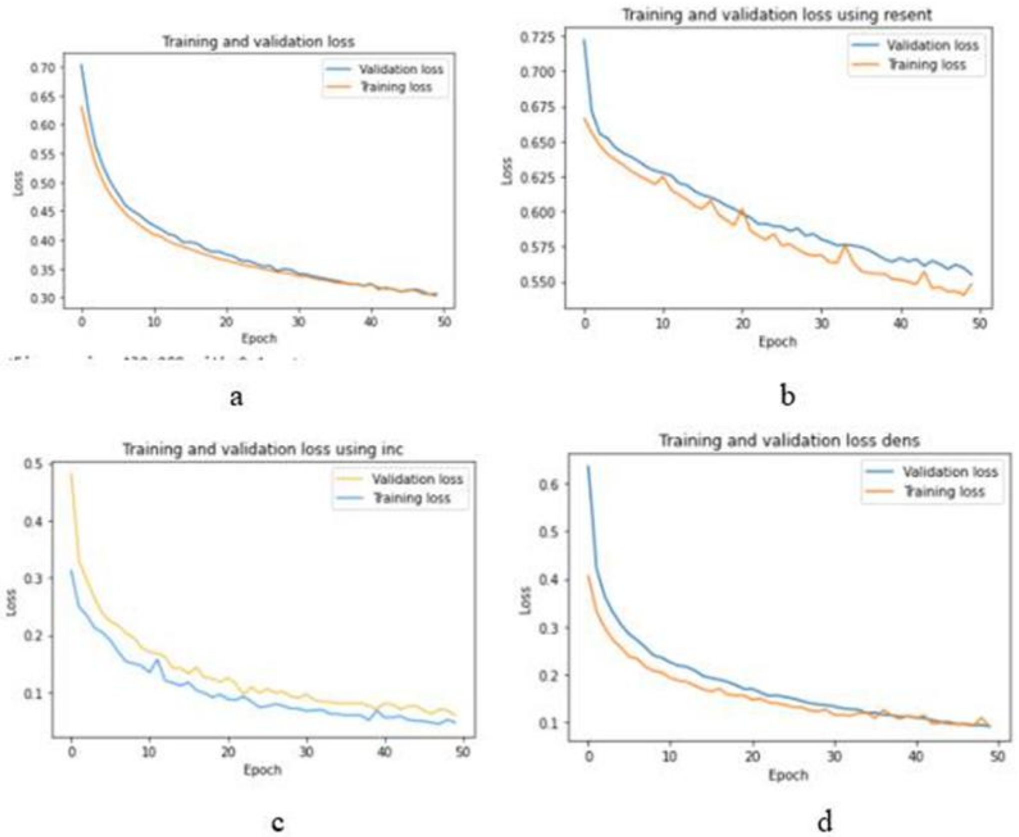
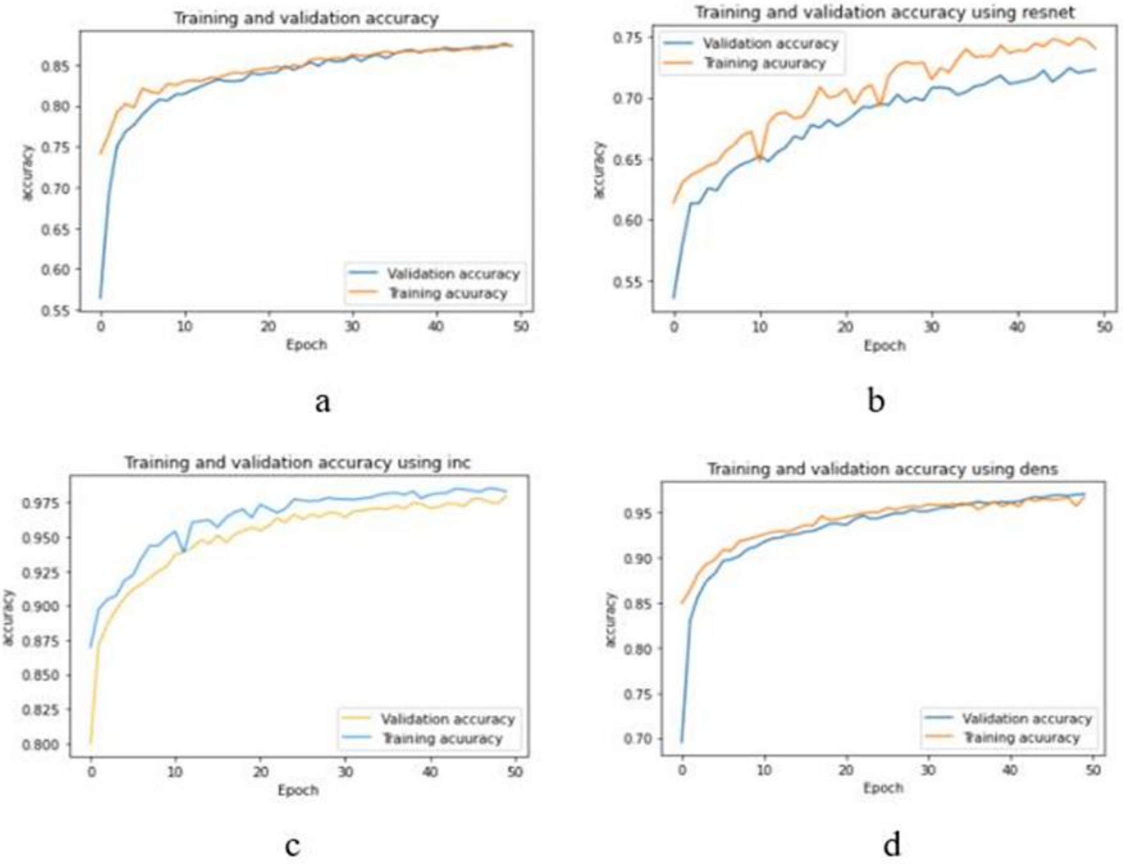


Figure 5- 4: Loss for different feature extraction methods a) VGG 19 b) ResNet50 c) Inception d) densnet

In Figure 5- 5 below, we have tried to show the accuracy for using different feature extraction methods combined with the GRU sequence information learner. We have used VGG 19 [66], ResNet50 [25], Inception and DenseNet [65] and shown below in Figure 5- 5 the accuracy vs epoch graph for each feature extraction methods. We have used all those feature extraction and found that EfficientNet [49] out preform them from the perspective of accuracy and loss.



a

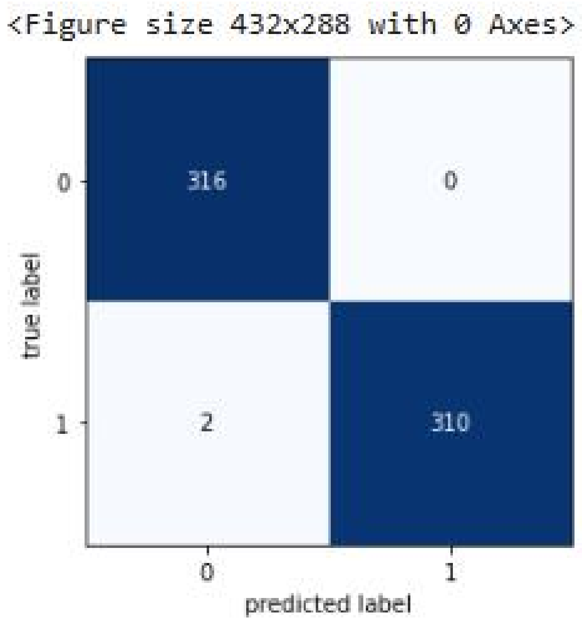
b

c

d

Figure 5- 5: Accuracy for different feature extraction methods a) Vgg19 b) ResNet50 c) Inception d) densnet

A confusion matrix [67] is a 2-D matrix used to describe the overall performance of the models. The rows and columns of the matrix are marked by the classes[67]. The diagonal cells correspond to observations that are correctly identified and represent the true positives in a binary-class confusion matrix. In Figure 5- 6 the off-diagonal cells correspond to incorrectly identified observations.



```
<Figure size 432x288 with 0 Axes>
```

Figure 5- 6: Confusion Matrix of Pornographic video classification using CNN and GRU

Our confusion matrix shows the performance of our system in the visual information. It shows how much class is correctly classified from the test dataset. It classifies correctly for both classes but we see same misclassification for the non-porn class. Since there are many activities that may look like pornographic activity, it is mis-classifying for this reason.

4.3.3 Runtime

We have also evaluated the runtime efficiency of the model in order to compare the LSTM and GRU and it can be analyzed from the figure below [68].

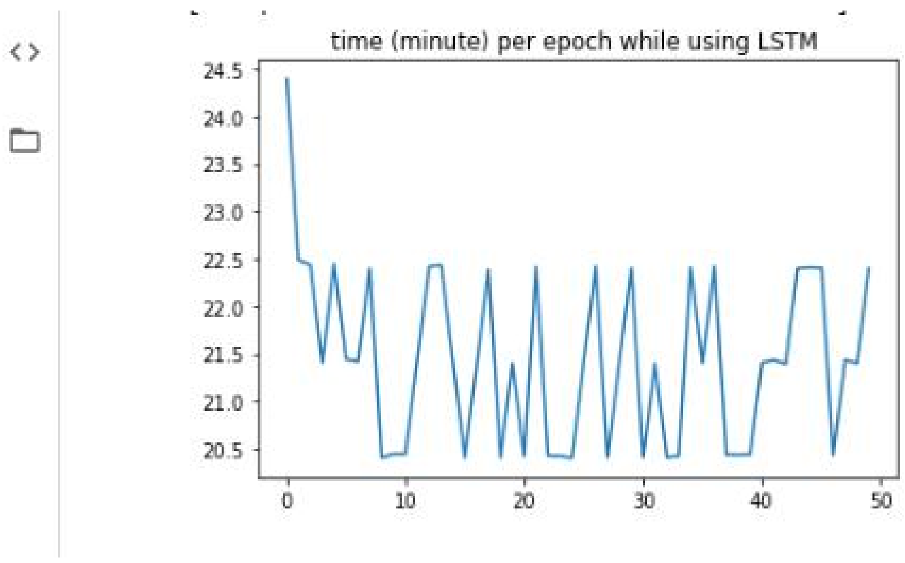
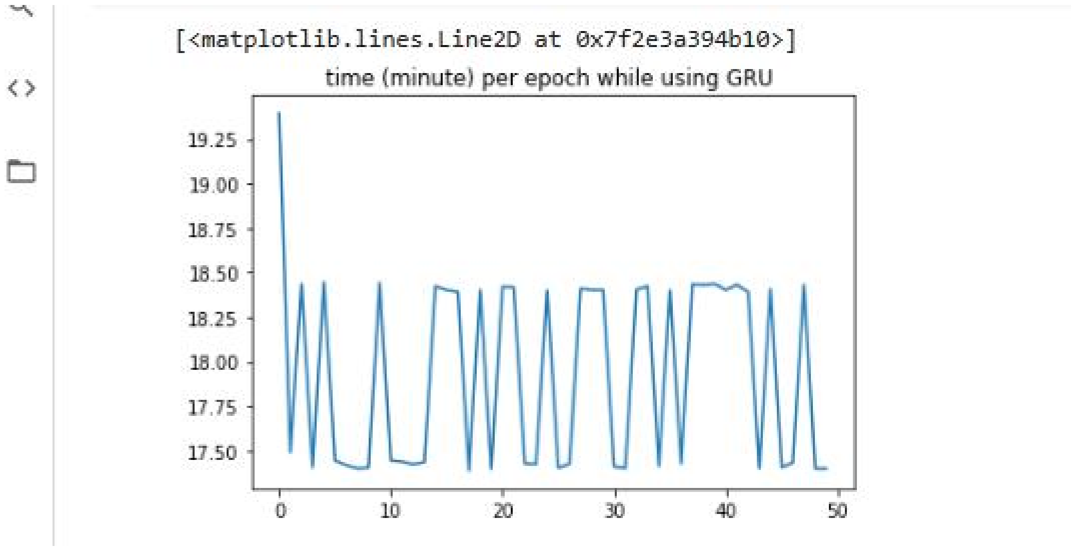


Figure 5- 7: Time per epoch for LSTM



```
[<matplotlib.lines.Line2D at 0x7f2e3a394b10>]
```

Figure 5- 8 Time per epoch for GRU

It is very visible that the training runtime is smaller for GRU compare to LSTM. The reason for the computational simplicity of GRU is that the number of gates in GRU are smaller than LSTM. Some research works have also try to show these phenomena. [43] [69]

4.3.4 The K-Fold cross validation

The k-fold cross-validation methodology is implemented in this experiment to optimize the approach used to assess the deep learning method. This methodology has the advantages of using all samples for both training and test, with each sample being used only once for a test. We use k=4 in this experiment, we choose four due to the computational scarcity after 4 folds our colab runtime expires. We divided the original dataset into four subgroups based on this. A single subset was chosen for testing from the four subsets. The training sets were made up of the remaining 3 subsets. By changing the test and training sets, the cross-validation process was repeated four times. The parameter and hyper parameters of the model were used again. Figure 5- 9 shows the results of the 4-folds cross validation experiments. The average accuracy we get from the 4 fold validation is 0.9904 and the average loss we get from the 4 fold validation is 0.0053. Since it is hard to display the result in a graph the numerical value is presented in Appendix E.

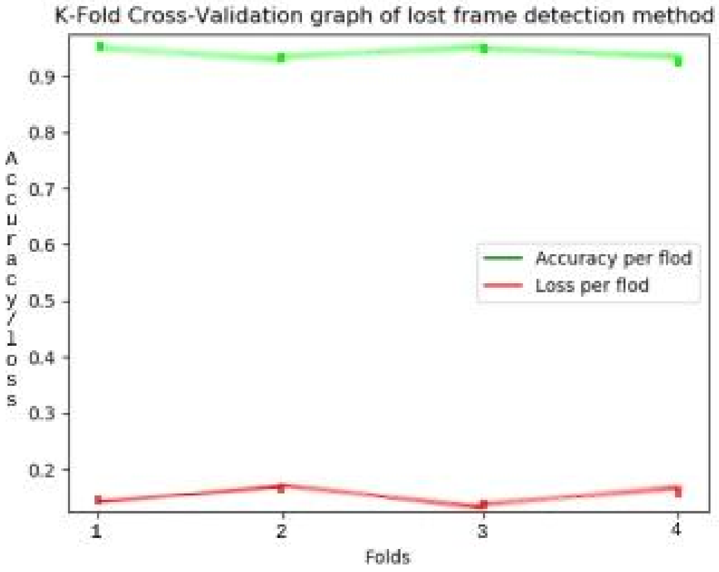
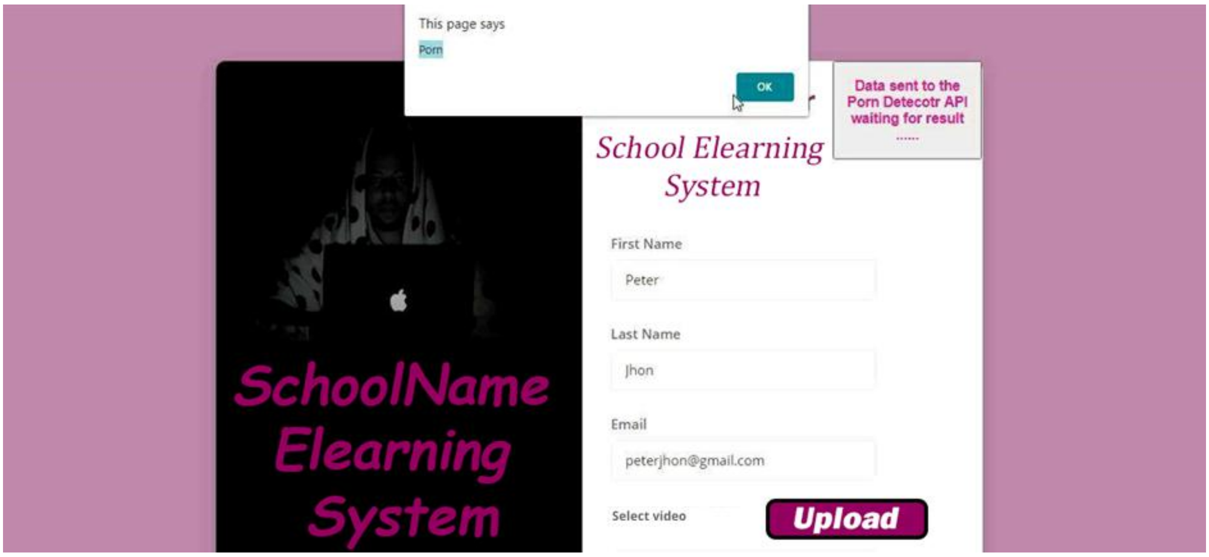


Figure 5- 9: K-Fold cross validation

4.4 Prototype of the system

We believe that every developed research should have its own contribution in making the world a better place. Researchers have an impact when the finding or outcome of research is applied into the life of the ordinary peoples. After we have done developing the model, we started thinking about how we can apply the finding of this research to the community and we have developed two mainly real time and real-world pornographic video detection systems.

The first application is a real time web API which is used to analyze video. The web API is developed using Django [70] which is python web framework. Using this API any website can detect a video or image if it is porn or non-porn. Many websites in Ethiopia do't use a content filtering web or application API. For example, Jimma University has an eLearning system but the materials which are uploaded to the system does not go through the API, so this leads to the easy circulation of pornographic contents in the web. But with our API people can detect a video before it goes to their web server. In Figure 5- 10 below shows a web using our API.



This page says

Porn

Data sent to the Porn Detecotr API waiting for result ......

School Elearning System

First Name: Peter

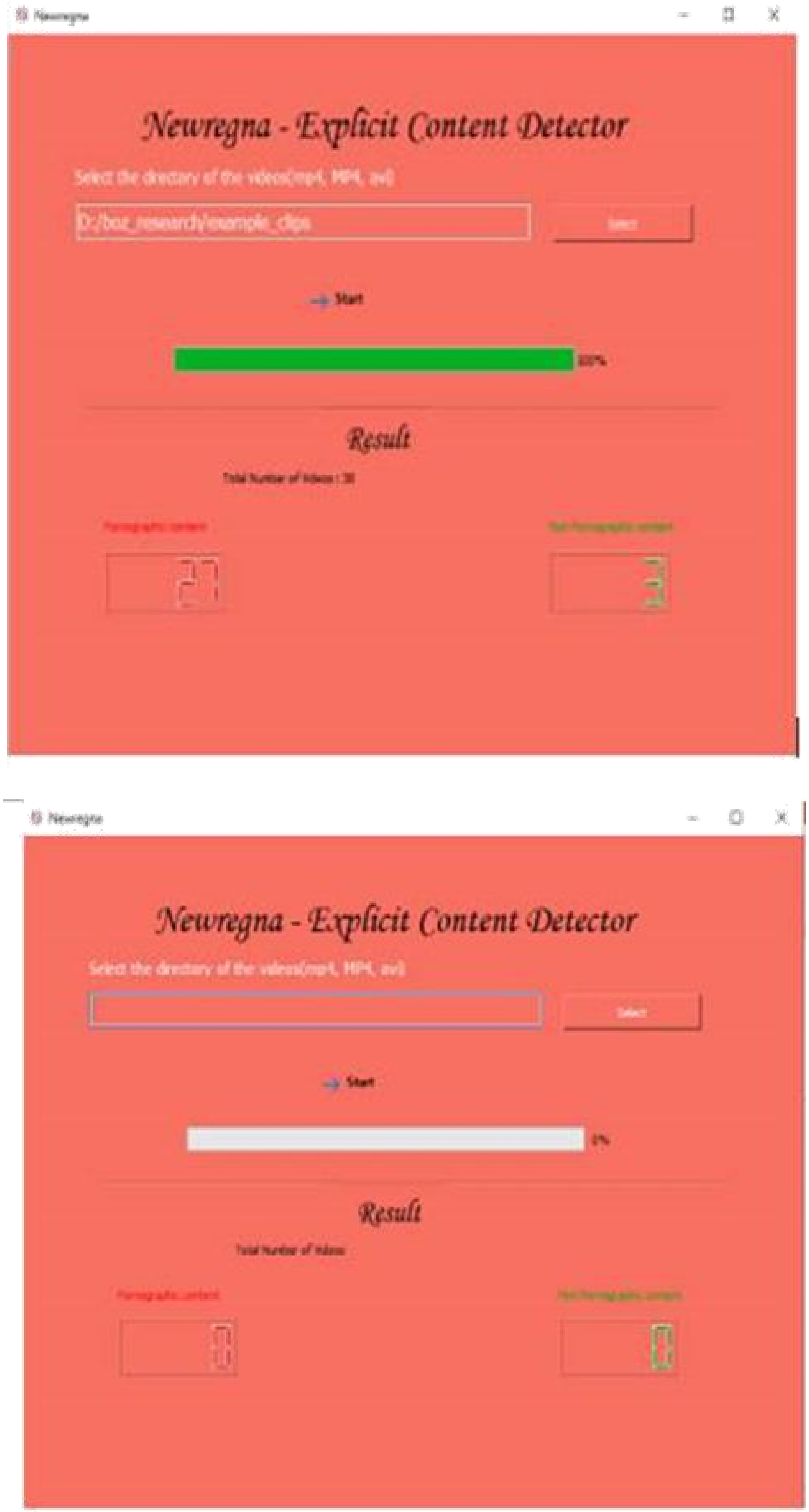
Last Name: Jhon

Email: peterjhon@gmail.com

Select video

Figure 5- 10: API for porn image and video detection

The second real time application is a desktop application shown in Figure 5- 11 which enables us to analyze an enormous amount of video files. If we have a large amount of video data and want to identify the pornographic content from the non-pornographic content, it is hard to hire people to do this task for two reasons. First the content they see is a pornographic content which by itself is a distracting and dangerous thing. Second, it is very tedious work and may lead to false classification. The developed desktop application solved those two problems.



Newregna - Explicit Content Detector

Select the directory of the videos(mp4, MP4, avi)

D:/boz_research/example_clips

Total Number of Videos : 30

Pornographic content: 27

Non Pornographic content: 3

Newregna - Explicit Content Detector

Select the directory of the videos(mp4, MP4, avi)

Total Number of Videos

Pornographic content: 0

Non Pornographic content: 0

Figure 5- 11: Desktop application for porn detection from large data of videos

The desktop application is developed using a python module named PyQt5 [71] and other python libraries. The application accepts a directory having many videos, and it gives an output which shows how much of the video is pornographic and how much is non-pornographic. In order to identify each video, it uses our trained model for accurate and fast classification of the videos. Below is the same as the screenshots of our desktop application.

4.5 Discussion

Working with pornographic content is a really challenging task since the content we are working with is very graphic and disturbing for a normal human mind. But we believe every research works no matter how challenging it is, it should solve the problem of the community so our challenging work has solved one of the most destroying problems of our community by developing an automatic and accurate way for classification of pornographic videos.

Since we have used GRU the best and the state-of-the-art method for video classification, we have got a very good accuracy in pornographic video classification. Our developed model uses both CNN and RNN, we use CNN for the feature extraction part we have used EfficeintNet which is a pre-trained transfer learning method for the process of feature extraction.This EfficeintNet extraction important features automatically form the sequence of frames which are generated from the video clips of the NPDI pornographic video dataset. Those extracted features are in reach with motion information which letter will be used by the GRU sequence learner.

The developed prototype system which is developed using the model we have created in this research work highly applies to our community. University, local social media and other organizations can use the web API to detect pornographic contents before they are uploaded to their system. The desktop application also applies for analyzing large amounts of video data for pornographic contents.

4.6 Comparison with previous related works

To the best of our survey, we have found previous work on pornographic video classification using deep learning and image processing methods. But there are about three works in general we have selected the one which use Convolutional neural network together with the recurrent neural network methods. Hence, we have done the comparison with them, since we use the same dataset, the 2k NPDI [11] pornographic dataset. We have used the publicly available NPDI pornographic dataset as a baseline to compare our work with the others. In the Figure 5- 12 and Table 5- 5 we have presented the accuracy that is achieved by those works in compression with our work.

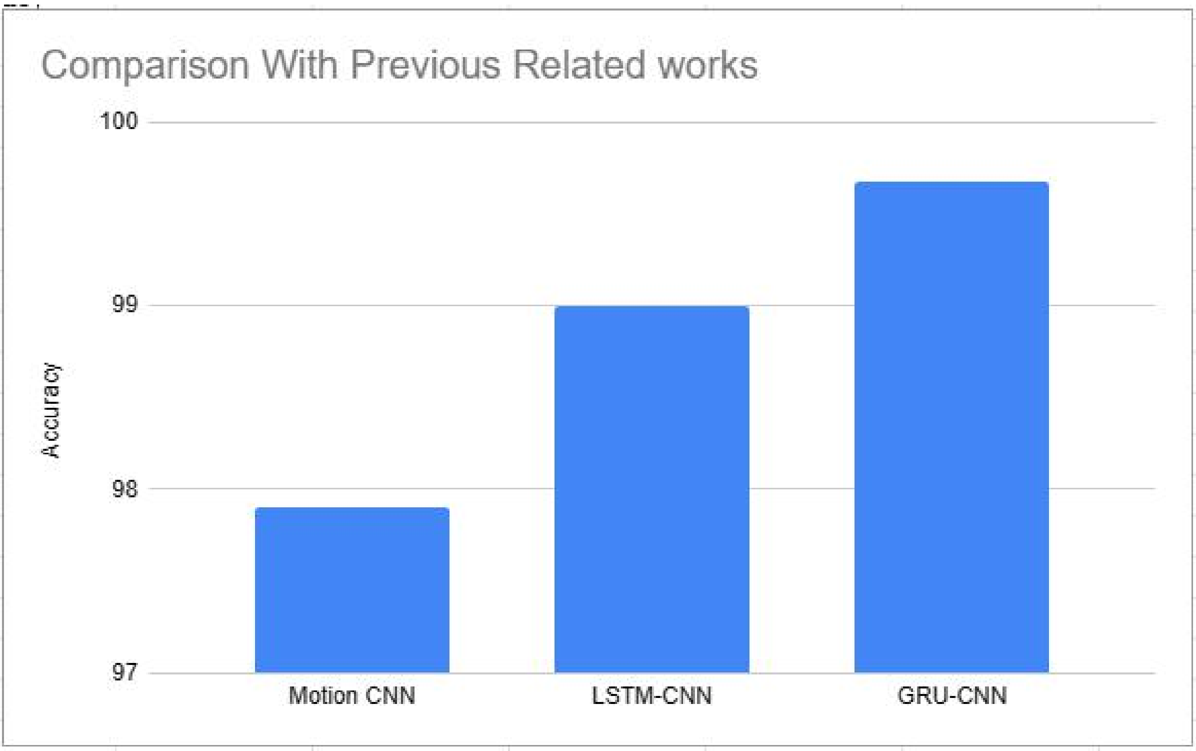


Figure 5- 12: Comparison with previous related works

Table 5- 5: Comparison with previous related works

Research Year	Method	Feature extraction approach	Dataset used	Tool	Result
2017	Motion CNN	GoogleNet	NPDI 2K	Python	97.9% accuracy
2018	CNN - LSTM	ResNet50[25]	NPDI 2k	Python	99.0 % accuracy
	CNN - GRU	EfficeintNet	NPDI 2k	Python	99.68% accuracy

CHAPTER 5. CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Africa is the continent with a high number of young children aged population (the most productive age span). The exposure of this population class to a pornographic content will affect the morality and economic potential of the continent negatively since youths are considered as a backbone of any economic growth. Thus, wise control over the pornographic video releases plays a crucial role in escaping the community from contemporary negatively targeting mind war. Due to increased amount of digital content in a web, it is hard to find pornographic videos out of non-pornographic one. According to 2019 Pornhub's report [2], 115 million people visit pornhub, which is one of the largest pornographic video site, daily. Only in 2019, the calculated average shows that people spent 1.36 million hrs. By watching pornographic videos in pornhub. Also, in 2019 there is a data transfer of 6597 Petabyte which means each day there is a data transfer of 18,073 Terabytes which is a significant amount of the waste of bandwidth of the internet. This and other statistics from Pornhub indicate how pornographic videos are circulating in the community. The amount of damage they create is more than our imagination. It is believed that there is an advantage to solve the community problem by creating an automated way for pornographic video classification tasks which use the latest available methods of learning and identifying pornographic videos.

In order to overcome the above listed potential damages we come up with a solution which classifies whether the web content video is pornographic by using deep learning algorithm that is a branch of machine learning. Since it is tedious activity to classify whether the video is porn for each individual being manual, this solution is promising since no human intervention is required. In machine learning, there is a tedious task which is feature engineering, but by using deep learning we have to get rid of the task instead we replace it with an automated way. Since we are working with sequential data, we have used recurrent types of layers in our deep learning layers.

We have used supervised learning in order to classify the pornographic video. Meaning that all the training data are labeled, and all the extracted features are categorized based on either they are extracted from the porn or non-porn videos. The focus of this research is to accurately classify pornographic videos by using latest deep learning methods. In order to achieve our goal, we have used the combination of CNN together with RNN. In our architecture we have used EfficeintNet for feature extraction and GRU for learning of the extracted features and predicting the type of video. Our architecture is a novel way for classifying pornographic videos.

In our EfficeintNet feature extractor, we have used RGB image frames extracted from a video clip. Those clips are carefully generated from the videos of the publicly available Pornographic dataset NPDI Dataset. For each video clip its length is 3 second we have extracted RGB image frames in the rate of 20 fps and we got a total of 60 frames from a single video clip after we made basic preprocessing on those extracted images, we feed it to the ImageNet pre trained model which is EfficientNet to extract feature using the transfer learning principle. After extracting features from each single image frame, all frames related to a single video file are aggregated to their respective video file before it is fed to the sequence learning GRU. Finally, it is given to the GRU which learns the motion information from the extracted feature and predicts the type of video.

We have used different evaluation metrics for evaluating the performance, like accuracy, F1 score recall and confusion matrix. we evaluated our model using this evaluation metrics and found it while performing well in respective to other previous works done in this area. We have used the publicly available dataset for this problem called NPDI [11] dataset and for developing our model and the final prototype product we have used python as a base programming language and Keras and TensorFlow as a deep learning framework and we also used other image processing and different python libraries for mathematical operations and data processing.

5.2 Contribution of the study

The followings are the chief contributions of this research work.

- We have designed an architecture for identifying pornographic video using the per-trained EfficeintNet model in combination with the GRU recurrent neural network.
- The per-trained EfficeintNet is used to extract feature from the generated frames for each video clip of the available pornographic video dataset and thus extracted important features are aggregated to form a single video and fed to the GRU sequential information learner to classify each video to Porn and Non-porn.

- We have shown that the use of GRU for pornographic video classification will lead to an increase in the accuracy of the classification task. We have used GRU by combining it to the EfficeintNet feature extraction because of this we have utilized the advantage of using GRU for such kind of problem it has a small amount of computational complexity and it also has a better accuracy and this will make our research to be the first to use GRU together with EfficeintNet for pornographic video classifications.
- Finally, our research shows that the use of motion information for the classifications of pornographic information will play a substantial amount of role in having better accuracy.

5.3 Recommendation

Dataset is the key for having an accurate deep learning model in the case of computer vision, big data is necessary. Here in our research, we have used the publicly available NPDI dataset, which is organized in Brazil, where an enormous amount of porn content is available. We have got very good accuracy in this dataset. The fact is that it is possible to get a better accuracy with more dataset having all ethnic and many pornographic categories. In [73] it have been shown that in general video classification considering the sound available in the video will increase the performance up to 8%. Here we have used the motion information we get from the pornographic videos to classify pornographic videos but the researcher believes that it may have a great role of using sound in combination of images and video together with the motion information, we can get a good accuracy based on the research done on [73]. A large amount of research has to be done in the pornographic video classification task since pornographic videos are destroying the youth the most productive part of the community, all of our deep learning researchers, computer vision researchers, psychologists and health researchers have to come together to solve this community problem.

REFERENCE

[1] Pornhub, “Pornhub Review.” [Online]. Available: https://www.pornhub.com/insights/pornhub-2015-year-in-review.

[2] Pornhub, “Pornhub Review,” 2019. [Online]. Available: https://www.pornhub.com/insights/pornhub-2015-year-in-review.

[3] T. B. Smidt, “Pornography as Work Culture and Cultural Phenomenon,” p. 34, 2011.

[4] E. W. Owens, R. J. Behun, J. C. Manning, and R. C. Reid, “The Impact of Internet Pornography on Adolescents: A Review of the Research,” *Sex. Addict. Compulsivity*, vol. 19, no. 1–2, pp. 99–122, 2012.

[5] NHRC and OSRT, “Trafficking in persons, especially women and children. National report 2011,” 2012.

[6] UN General Assembly, “Report of the Special Rapporteur on the sale of children, child prostitution and child pornography, Najat Maalla M’jid,” *Hum. Rights Counc.* , no. March, 2012.

[7] A. D. Dwulit and P. Rzymski, “Prevalence, patterns and self-perceived effects of pornography consumption in polish university students: A cross-sectional study,” *Int. J. Environ. Res. Public Health*, vol. 16, no. 10, 2019.

[8] Yesserie, “The status of pornogrphy,” *Addis Abeba Univ. thesis*, vol. 151, no. June, pp. 10–17, 2015.

[9] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[10] R. Rana, “Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech,” pp. 1–9, 2016.

[11] A. Gangwar, E. Fidalgo, and E. Alegre, “Pornography and Child Sexual Abuse Detection in Image and Video : A Comparative Evaluation,” pp. 37–42, 2017.

[12] A. Pedneault, “Ted Bundy on the" malignant being": An analysis of the justificatory discourse of a serial killer,” *Sfu.Ca*, no. 1957, 2005.

[13] L. Ugalde, J. I. Martínez-de-Morentin, and M. C. Medrano, “Adolescents’ Tv viewing patterns in the digital era: A cross-cultural study,” *Comunicar*, vol. 25, no. 50, pp. 67–75, 2017.

[14] C. Bell, “An Overview of Research on the Impact that Viewing Pornography has on Children, Pre-Teens and Teenagers,” *Bravehearts*, no. September 2011, p. 2017, 2017.

[15] Council of Europe, “Media Regulatory Authorities And Protection Of Minors,” 2019.

[16] H. M and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 01–11, 2015.

[17] N. Malamuth, “Pornography’s Impact on Male Adolescents.,” *Adolescent medicine (Philadelphia, Pa.)*, vol. 4. pp. 563–576, 1993.

[18] C. M. Bishop, “Neural networks and their applications,” *Rev. Sci. Instrum.*, vol. 65, no. 6, pp. 1803–1831, 1994.

[19] A. Shrestha and A. Mahmood, “Review of deep learning algorithms and architectures,” *IEEE Access*, vol. 7, no. c, pp. 53040–53065, 2019.

[20] N. Milosevic, “Introduction to Convolutional Neural Networks,” *Introd. to Convolutional Neural Networks*, pp. 1–31, 2020.

[21] H. Wu and X. Gu, “Max-pooling dropout for regularization of convolutional neural networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9489, pp. 46–54, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[23] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Phys. D Nonlinear Phenom.*, vol. 404, no. March, pp. 1–43, 2020.

[24] P. Munro *et al.*, “Backprop,” *Encyclopedia of Machine Learning*. pp. 69–73, 2011.

[25] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, “Overcoming the vanishing gradient problem in plain recurrent networks,” no. Section 2, pp. 1–20, 2018.

[26] S. Hochreiter and J. Urgen Schmidhuber, “Ltsm,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] N. A. Bakar and S. Rosbi, "Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction," *Int. J. Adv. Eng. Res. Sci.*, vol. 4, no. 11, pp. 130–137, 2017.

[28] Y. Zhu *et al.*, "What to do next: Modeling user behaviors by Time-LSTM," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 0, no. March 2018, pp. 3602–3608, 2017.

[29] T. Endeshaw, J. Garcia, and A. Jakobsson, "Classification of indecent videos by low complexity repetitive motion detection," *Proc. - Appl. Imag. Pattern Recognit. Work.*, 2008.

[30] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv*, pp. 1–20, 2018.

[31] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Appl. Sci.*, vol. 10, no. 5, 2020.

[32] J.-J. Yu, "Porn Video classfication using fast motion features," vol. 6.

[33] P. Podder, T. Zaman Khan, M. Haque Khan, and M. Muktadir Rahman, "Comparative Performance Analysis of Hamming, Hanning and Blackman Window," *Int. J. Comput. Appl.*, vol. 96, no. 18, pp. 1–7, 2014.

[34] Z. Qu, Y. Liu, Y. Liu, K. Jiu, and Y. Chen, "A porn video detecting method based on motion features using HMM," *Isc. 2009 - 2009 Int. Symp. Comput. Intell. Des.*, vol. 2, pp. 461–464, 2009.

[35] M. Chains, "Hidden Markov Models," 2019.

[36] J. Wehrmann, G. S. Simões, R. C. Barros, and V. F. Cavalcante, "Adult content detection in videos with convolutional and recurrent neural networks," *Neurocomputing*, vol. 272, pp. 432–438, 2018.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "2012 AlexNet," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[38] C. Szegedy, S. Reed, P. Sermanet, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," pp. 1–12.

[39] J. Jung and A. Morlot, “Combining CNNs for detecting pornography in the absence of labeled training data.”

[40] B. Petrovska, E. Zdravevski, P. Lameski, R. Corizzo, I. Štajduhar, and J. Lerga, “Deep learning for feature extraction in remote sensing: A case-study of aerial scene classification,” *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–22, 2020.

[41] A. Bonzanini, R. Leonardi, and P. Migliorati, “Semantic video indexing using MPEG motion vectors,” *Eur. Signal Process. Conf.*, vol. 2015-March, no. March, 2000.

[42] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. M. Nishikawa, “A support vector machine approach for detection of microcalcifications,” *IEEE Trans. Med. Imaging*, vol. 21, no. 12, pp. 1552–1563, 2002.

[43] K. Chandani, “An Experimental Comparison of Deep LSTM and GRUs for Event Classification in Sports,” 2019.

[44] H. Wehle, “ML – AI- COGNITIVE,” no. July, 2017.

[45] N. M. Nawi, W. H. Atomi, and M. Z. Rehman, “The Effect of Data Pre-processing on Optimized Training of Artificial Neural Networks,” *Procedia Technol.*, vol. 11, no. February 2014, pp. 32–39, 2013.

[46] M. Khan, S. Chakraborty, R. Astya, and S. Khepra, “Face Detection and Recognition Using OpenCV,” *Proc. - 2019 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2019*, vol. 2019-Janua, pp. 116–119, 2019.

[47] FFmpeg Developers, “FFmpeg Developers. (2016). ffmpeg tool (Version be1d324) [Software].” .

[48] E. Mocsari and S. S. Stone, “Colostral IgA, IgG, and IgM-IgA fractions as fluorescent antibody for the detection of the coronavirus of transmissible gastroenteritis.,” *Am. J. Vet. Res.*, vol. 39, no. 9, pp. 1442–1446, 1978.

[49] E. Is, S. U. Cnn, and R. Process, “EfficientNet.”

[50] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” *Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016*, no. December, pp. 324–328, 2017.

[51] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical Evaluation of Rectified Activations in Convolutional Network,” 2015.

[52] S. Kanai, Y. Yamanaka, Y. Fujiwara, and S. Adachi, "SigsofTmax: Reanalysis of the softmax bottleneck," *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 286–296, 2018.

[53] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification," 2020.

[54] S. N. Brohi, T. R. Pillai, S. Kaur, H. Kaur, S. Sukumaran, and D. Asirvatham, "Accuracy Comparison of Machine Learning Algorithms for Predictive Analytics in Higher Education," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 285, no. August, pp. 254–261, 2019.

[55] P. Jupyter, "jupyter notebook," 2021.

[56] G. S. of C. Projec, "scikit-learn," 2021.

[57] "pandas," 2021.

[58] Matlab, "Matplotlib," 2021.

[59] Google, "Tensorflow," 2021.

[60] Https://opencv.org/, "https://opencv.org/," 2021.

[61] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[62] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016.

[63] Y. Li, C. Wei, and T. Ma, "Towards explaining the regularization effect of initial large learning rate in training neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, pp. 1–12, 2019.

[64] B. Mele and G. Altarelli, "Lepton spectra as a measure of b quark polarization at LEP," *Phys. Lett. B*, vol. 299, no. 3–4, pp. 345–350, 1993.

[65] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[67] "Confusion Matrix," *Encycl. Stat. Sci.*, 2004.

[68] H. D. Hlynsson, A. N. Escalante-B, and L. Wiskott, “Measuring the data efficiency of deep learning methods,” *ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, no. March, pp. 691–698, 2019.

[69] “Pornhub Insights :: Digging deep into the data.” [Online]. Available: https://www.pornhub.com/insights/. [Accessed: 05-Aug-2021].

[70] “Django The Web framework for perfectionists with deadlines | Django.” [Online]. Available: https://www.djangoproject.com/. [Accessed: 30-Mar-2021].

[71] “PyQt5 · PyPI.” [Online]. Available: https://pypi.org/project/PyQt5/. [Accessed: 30-Mar-2021].

[72] Video Classification | Video Classification Model In Python (analyticsvidhya.com) [Accessed : 26-Oug-2021]

[73] AENet: Learning Deep Audio Features for Video Analysis Naoya Takahashi, Member, IEEE, Michael Gygli, Member, IEEE, and Luc Van Gool, Member, IEEE

APPENDIX A: SOURCE CODE AND TRAINING THE MODEL

A.1 Data Prepocessing

```
directory = os.fsencode(directory_in_str) count = 0
pcount = 0 npcount = 0 result = []

for file in os.listdir(directory): count=count+1

filename = str(os.fsdecode(file))

if filename.endswith(".MP4") or filename.endswith(".mp4") or filenam
e.endswith('.avi'):

path = directory_in_str + '/' + filename

# print(path, filename)

vs = cv2.VideoCapture(path) writer = None

(W, H) = (None, None) count = 0

# loop over frames from the video file stream frame_count = int(vs.g
et(cv2.CAP_PROP_FRAME_COUNT))

# print('Total number fo Frames :', frame_count)

f25 = int((frame_count * 25) / 100)

f50 = int((frame_count * 50) / 100) f75 = int((frame_count * 75) / 1
00)
# print(f25, f50, f75)

r = [] print(path) try:

for i inrange(f25,f75+1,f25): vs.set(cv2.CAP_PROP_POS_FRAMES, i)
# print('Position:', int(vs.get(cv2.CAP_PROP_POS_FRAMES)))
_, frame = vs.read()

# cv2.imwrite("merz"+str(i)+".png", frame)

# read the next frame from the file
```

```
(grabbed, frame) = vs.read()

# if the frame was not grabbed, then we have reached the end
# of the stream

if not grabbed:
```

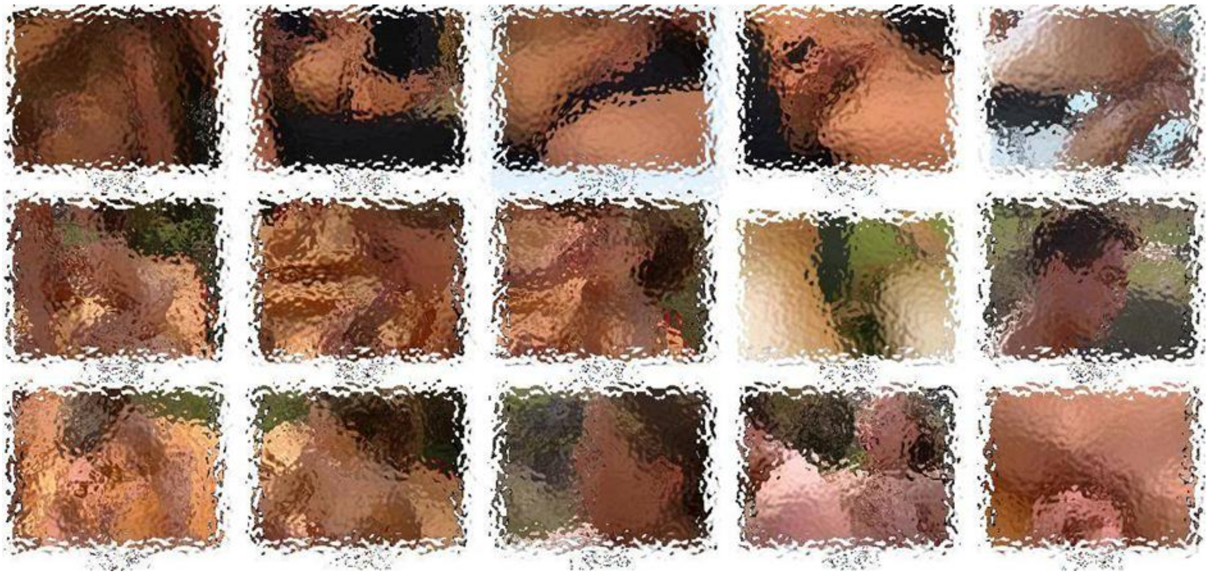
A.2 Fine-tuned Feature extraction

```
import numpy as np
import matplotlib.pyplot as plt # used to plot fig
from mlxtend.plotting import plot_confusion_matrix  # used to draw confusion matrix
import os  # This is used to read files to indict floders in the operating system
import cv2 # cv2 is used in opencv and deeplearning foe images transform, flip, change color and so on..

from sklearn.model_selection import train_test_split # this is used in Machine learning so i used it to split the train and test file 80/20 respectively
from tensorflow.keras.utils import to_categorical #Converts a class vector (integers) to binary class matrix.
from sklearn.preprocessing import LabelBinarizer  #Fit label binarizer and transform multi-class labels to binary labels
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Conv2D, MaxPooling2D, AveragePooling2D, Input, Dropout, Flatten, BatchNormalization
from keras.models import Sequential, Model
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from tensorflow.keras.optimizers import Adam, SGD
from sklearn.metrics import accuracy_score, confusion_matrix
import tensorflow as tf
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from keras.applications.vgg19 import VGG19
from keras.applications.vgg16 import VGG16
from keras.applications.resnet50 import ResNet50
from keras.applications.densenet import DenseNet121
from keras.applications.mobilenet_v2 import MobileNetV2
from keras.applications.inception_v3 import InceptionV3

import keras as k
from numpy import array
from keras.datasets import imdb
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import GRU
from keras.layers import Dropout
import matplotlib.pyplot as plt
from keras.layers import BatchNormalization
from keras.preprocessing import sequence
# fix random seed for reproducibility

import os
import pickle
#from data_proc import filesAndLabels

#data_proc

import numpy as np
import matplotlib.pyplot as plt # used to plot fig
from mlxtend.plotting import plot_confusion_matrix  # used to draw c
onfusion matrix
import os  # This is used to read files to indict floders in the ope
rating system
import cv2 # cv2 is used in opencv and deeplearning foe images trans
form, flip, change color and so on..

from sklearn.model_selection import train_test_split # this is used
in Machine learning so i used it to split the train and test file 80
/20 respectively
from tensorflow.keras.utils import to_categorical #Converts a class
vector (integers) to binary class matrix.
from sklearn.preprocessing import LabelBinarizer  #Fit label binariz
er and transform multi-class labels to binary labels
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Conv2D, MaxPooling2D, AveragePooling
2D, Input, Dropout, Flatten, BatchNormalization
from keras.models import Sequential, Model
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, Earl
yStopping
from tensorflow.keras.optimizers import Adam, SGD
from sklearn.metrics import accuracy_score, confusion_matrix
import tensorflow as tf
from tensorflow.keras.applications.mobilenet_v2 import preprocess_in
put
from tensorflow.keras.preprocessing.image import img_to_array
from keras.applications.vgg19 import VGG19
from keras.applications.vgg16 import VGG16
from keras.applications.resnet50 import ResNet50
from keras.applications.densenet import DenseNet121
from keras.applications.mobilenet_v2 import MobileNetV2
from keras.applications.inception_v3 import InceptionV3
np.random.seed(7)


from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input
# from keras.applications.vgg16 import VGG16
from tensorflow.python.keras.applications.efficientnet import Effici
entNetB0
# from keras.applications.densenet import DenseNet121
import efficientnet.keras as efn
model = EfficientNetB0(weights='imagenet', include_top=False)
baseModel = efn.EfficientNetB0(weights="imagenet", include_top=False,
 input_shape=(224, 224, 3))

# construct the head of the model that will be placed on top of the
# the base model
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="sigmoid")(headModel)
# place the head FC model on top of the base model (this will become
# the actual model we will train)


def extract(model, img_path):
  #first go with 128 then 224
  img = image.load_img(img_path, target_size=(128, 128))
  img_data = image.img_to_array(img)
  img_data = np.expand_dims(img_data, axis=0)
  img_data = preprocess_input(img_data)
  vgg16_feature = model.predict(img_data)
  return vgg16_feature.flatten()

def filesAndLabels(parDir = "data partition"):
  os.chdir("/content/drive/MyDrive/boaz/GRU ResNet50/data partition")
  res = []
  sets = ['train','val','test']
  for set in sets:
    labels = os.listdir(os.path.join(parDir, set))
    for label in labels:
      seqs = os.listdir(os.path.join(parDir, set, label))
      for seq in seqs:
        res.append((os.path.join(parDir, set, label, seq), label))
  return res
```

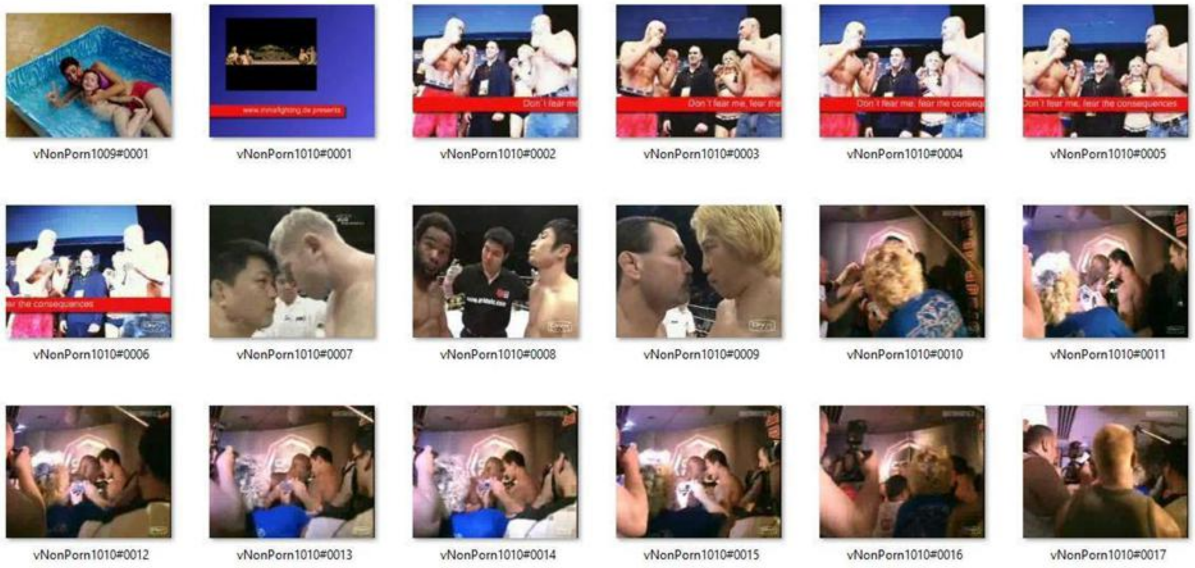
```
def run_preproc():
  for set in sets:
    for label in labels:
      seqs = os.listdir(os.path.join(set, label))
      for seq in seqs:
        frames = os.listdir(os.path.join(set, label, seq))
        framePaths = [os.path.join(set, label, seq, frame) for frame
 in frames]
        featureVectors = []
        for fp in framePaths:
          featureVectors.append(extract(model, fp))
        newSetPath = set.replace("data partition", newparDir)
        np.savetxt(os.path.join(newSetPath, label, seq+".txt"), feat
ureVectors)



run_preproc()
```

APPENDIX B: SAMPLE FRAMES FROM THE NPDI DATASET

Pornographic frames category



Non-Pornographic frames category



APPENDIX C: LAYER STRACTURE OF EFFICIENT NET

0 input_2

1 rescaling_1

2 normalization_1

3 stem_conv_pad

4 stem_conv

5 stem_bn

6 stem_activation

7 block1a_dwconv

8 block1a_bn

9 block1a_activation

10 block1a_se_squeeze

11 block1a_se_reshape

12 block1a_se_reduce

13 block1a_se_expand

14 block1a_se_excite

15 block1a_project_conv

16 block1a_project_bn

17 block2a_expand_conv

18 block2a_expand_bn

19 block2a_expand_activation

20 block2a_dwconv_pad

21 block2a_dwconv

22 block2a_bn

23 block2a_activation

24 block2a_se_squeeze

25 block2a_se_reshape

26 block2a_se_reduce

27 block2a_se_expand

28 block2a_se_excite

29 block2a_project_conv

30 block2a_project_bn

31 block2b_expand_conv

32 block2b_expand_bn

33 block2b_expand_activation

34 block2b_dwconv

35 block2b_bn

36 block2b_activation

37 block2b_se_squeeze

38 block2b_se_reshape

39 block2b_se_reduce

40 block2b_se_expand

41 block2b_se_excite

42 block2b_project_conv

43 block2b_project_bn

44 block2b_drop

45 block2b_add

46 block3a_expand_conv

47 block3a_expand_bn

48 block3a_expand_activation

49 block3a_dwconv_pad

50 block3a_dwconv

51 block3a_bn

52 block3a_activation

53 block3a_se_squeeze

54 block3a_se_reshape

55 block3a_se_reduce

56 block3a_se_expand

57 block3a_se_excite

58 block3a_project_conv

59 block3a_project_bn

60 block3b_expand_conv

61 block3b_expand_bn

62 block3b_expand_activation

63 block3b_dwconv

64 block3b_bn

65 block3b_activation

66 block3b_se_squeeze

67 block3b_se_reshape

68 block3b_se_reduce

69 block3b_se_expand

70 block3b_se_excite

71 block3b_project_conv

72 block3b_project_bn

73 block3b_drop

74 block3b_add

75 block4a_expand_conv

76 block4a_expand_bn

77 block4a_expand_activation

78 block4a_dwconv_pad

79 block4a_dwconv

80 block4a_bn

81 block4a_activation

82 block4a_se_squeeze

83 block4a_se_reshape

84 block4a_se_reduce

85 block4a_se_expand

86 block4a_se_excite

87 block4a_project_conv

88 block4a_project_bn

89 block4b_expand_conv

90 block4b_expand_bn

91 block4b_expand_activation

92 block4b_dwconv

93 block4b_bn

94 block4b_activation

95 block4b_se_squeeze

96 block4b_se_reshape

97 block4b_se_reduce

98 block4b_se_expand

99 block4b_se_excite

100 block4b_project_conv

101 block4b_project_bn

102 block4b_drop

103 block4b_add

104 block4c_expand_conv

105 block4c_expand_bn

106 block4c_expand_activation

107 block4c_dwconv

108 block4c_bn

109 block4c_activation

110 block4c_se_squeeze

111 block4c_se_reshape

112 block4c_se_reduce

113 block4c_se_expand

114 block4c_se_excite

115 block4c_project_conv

116 block4c_project_bn

117 block4c_drop

118 block4c_add

119 block5a_expand_conv

120 block5a_expand_bn

121 block5a_expand_activation

122 block5a_dwconv

123 block5a_bn

124 block5a_activation

125 block5a_se_squeeze

126 block5a_se_reshape

127 block5a_se_reduce

128 block5a_se_expand

129 block5a_se_excite

130 block5a_project_conv

131 block5a_project_bn

132 block5b_expand_conv

133 block5b_expand_bn

134 block5b_expand_activation

135 block5b_dwconv

136 block5b_bn

137 block5b_activation

138 block5b_se_squeeze

139 block5b_se_reshape

140 block5b_se_reduce

141 block5b_se_expand

142 block5b_se_excite

143 block5b_project_conv

144 block5b_project_bn

145 block5b_drop

146 block5b_add

147 block5c_expand_conv

148 block5c_expand_bn

149 block5c_expand_activation

150 block5c_dwconv

151 block5c_bn

152 block5c_activation

153 block5c_se_squeeze

154 block5c_se_reshape

155 block5c_se_reduce

156 block5c_se_expand

157 block5c_se_excite

158 block5c_project_conv

159 block5c_project_bn

160 block5c_drop

161 block5c_add

162 block6a_expand_conv

163 block6a_expand_bn

164 block6a_expand_activation

165 block6a_dwconv_pad

166 block6a_dwconv

167 block6a_bn

168 block6a_activation

169 block6a_se_squeeze

170 block6a_se_reshape

171 block6a_se_reduce

172 block6a_se_expand

173 block6a_se_excite

174 block6a_project_conv

175 block6a_project_bn

176 block6b_expand_conv

177 block6b_expand_bn

178 block6b_expand_activation

179 block6b_dwconv

180 block6b_bn

181 block6b_activation

182 block6b_se_squeeze

183 block6b_se_reshape

184 block6b_se_reduce

185 block6b_se_expand

186 block6b_se_excite

187 block6b_project_conv

188 block6b_project_bn

189 block6b_drop

190 block6b_add

191 block6c_expand_conv

192 block6c_expand_bn

193 block6c_expand_activation

194 block6c_dwconv

195 block6c_bn

196 block6c_activation

197 block6c_se_squeeze

198 block6c_se_reshape

199 block6c_se_reduce

200 block6c_se_expand

201 block6c_se_excite

202 block6c_project_conv

203 block6c_project_bn

204 block6c_drop

205 block6c_add

206 block6d_expand_conv

207 block6d_expand_bn

208 block6d_expand_activation

209 block6d_dwconv

210 block6d_bn

211 block6d_activation

212 block6d_se_squeeze

213 block6d_se_reshape

214 block6d_se_reduce

215 block6d_se_expand

216 block6d_se_excite

217 block6d_project_conv

218 block6d_project_bn

219 block6d_drop

220 block6d_add

221 block7a_expand_conv

222 block7a_expand_bn

223 block7a_expand_activation

224 block7a_dwconv

225 block7a_bn

226 block7a_activation

227 block7a_se_squeeze

228 block7a_se_reshape

229 block7a_se_reduce

230 block7a_se_expand

231 block7a_se_excite

232 block7a_project_conv

233 block7a_project_bn

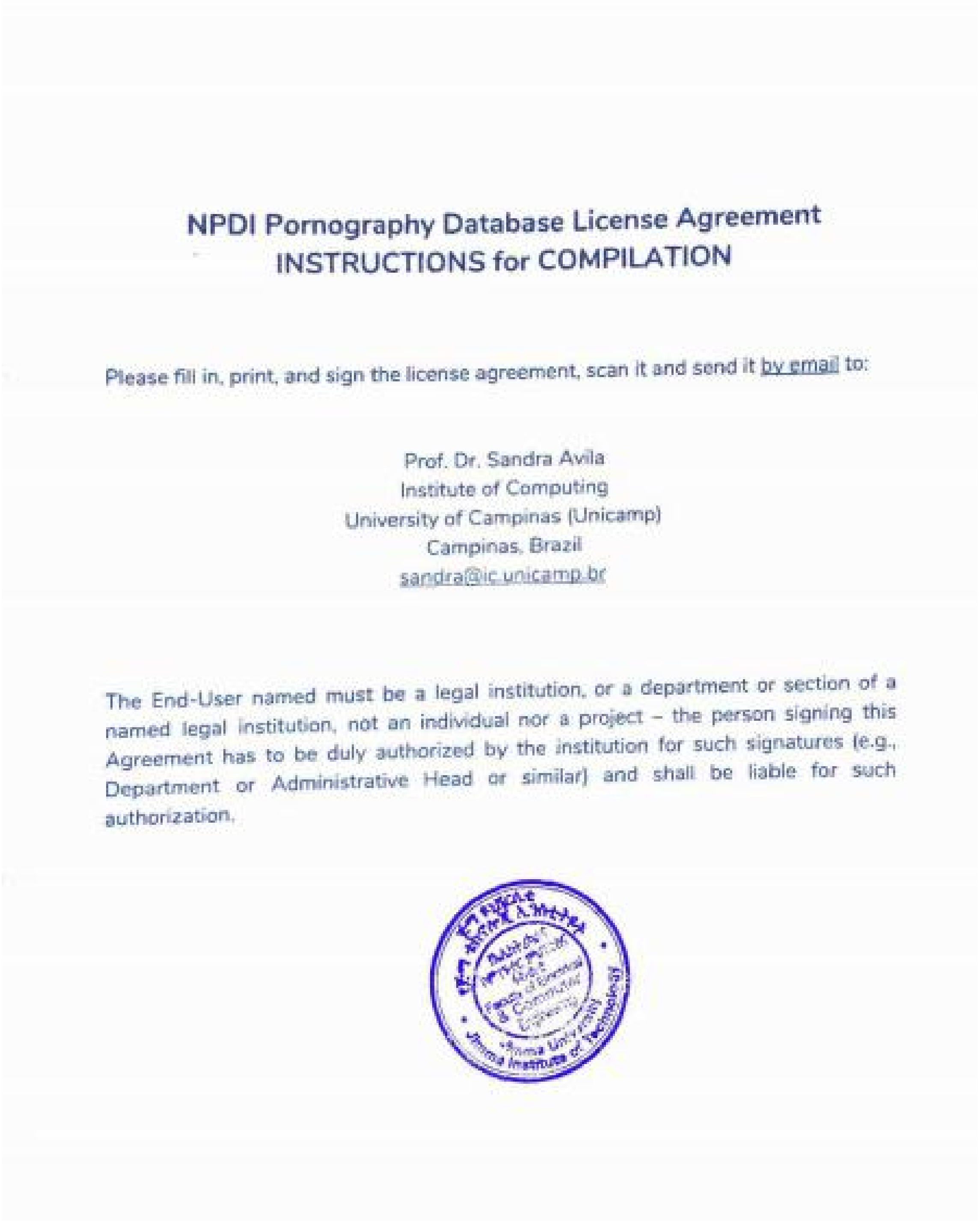
234 top_conv

235 top_bn

236 top_activation

\

APPENDIX D: NPDI LICENSE AGREEMENT



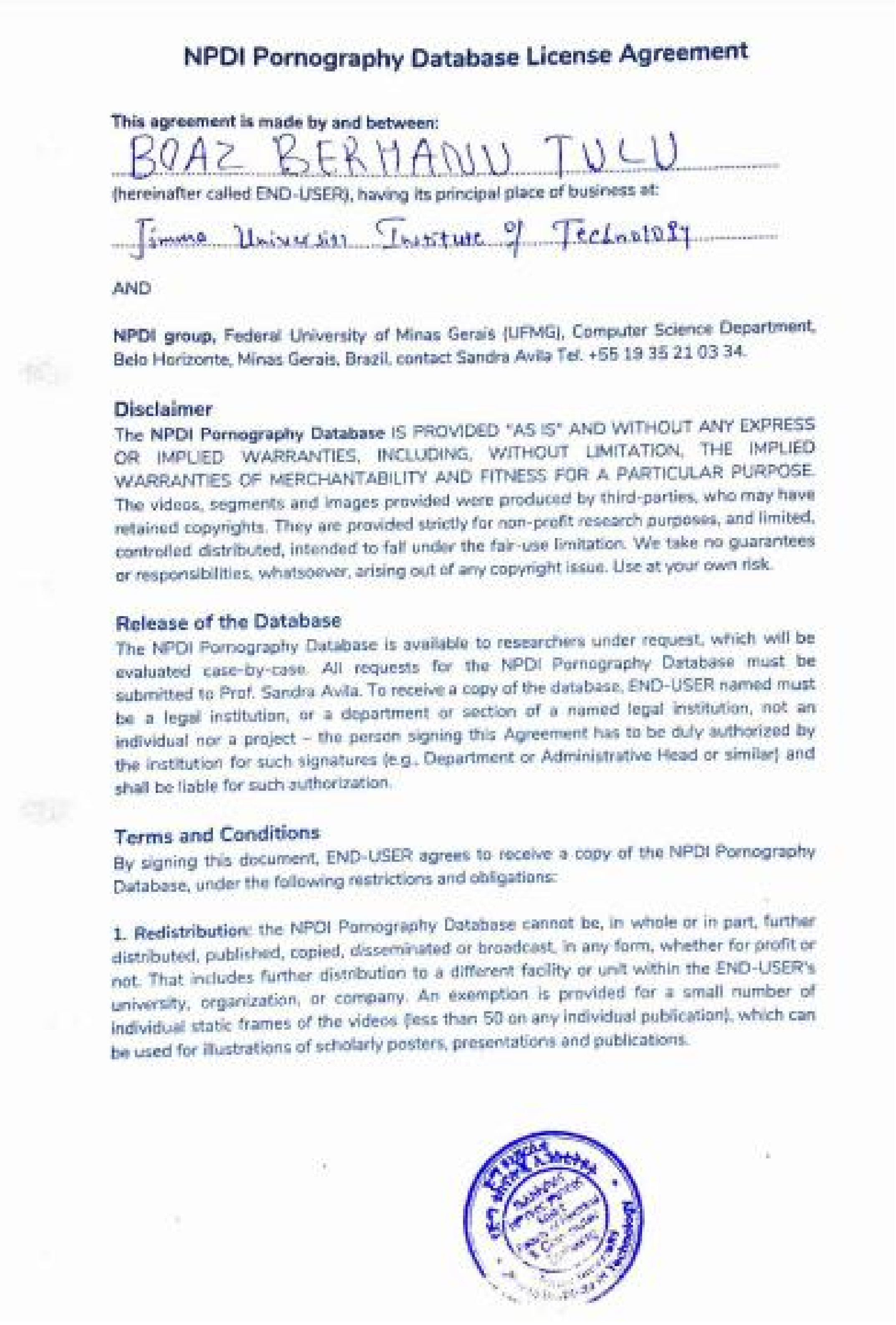
NPDI Pornography Database License Agreement
INSTRUCTIONS for COMPILATION

Please fill in, print, and sign the license agreement, scan it and send it by email to:

Prof. Dr. Sandra Avila
Institute of Computing
University of Campinas (Unicamp)
Campinas, Brazil
sandra@ic.unicamp.br

The End-User named must be a legal institution, or a department or section of a named legal institution, not an individual nor a project – the person signing this Agreement has to be duly authorized by the institution for such signatures (e.g., Department or Administrative Head or similar) and shall be liable for such authorization.

Faculty of Electrical & Computer Engineering
Jimma University
Jimma Institute of Technology



NPDI Pornography Database License Agreement

This agreement is made by and between:

BOAZ BERHANU TULU

(hereinafter called END-USER), having its principal place of business at:

Jimma University Institute of Technology

AND

NPDI group, Federal University of Minas Gerais (UFMG), Computer Science Department, Belo Horizonte, Minas Gerais, Brazil, contact Sandra Avila Tel. +55 19 35 21 03 34.

Disclaimer

The **NPDI Pornography Database** IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. The videos, segments and images provided were produced by third-parties, who may have retained copyrights. They are provided strictly for non-profit research purposes, and limited, controlled distributed, intended to fall under the fair-use limitation. We take no guarantees or responsibilities, whatsoever, arising out of any copyright issue. Use at your own risk.

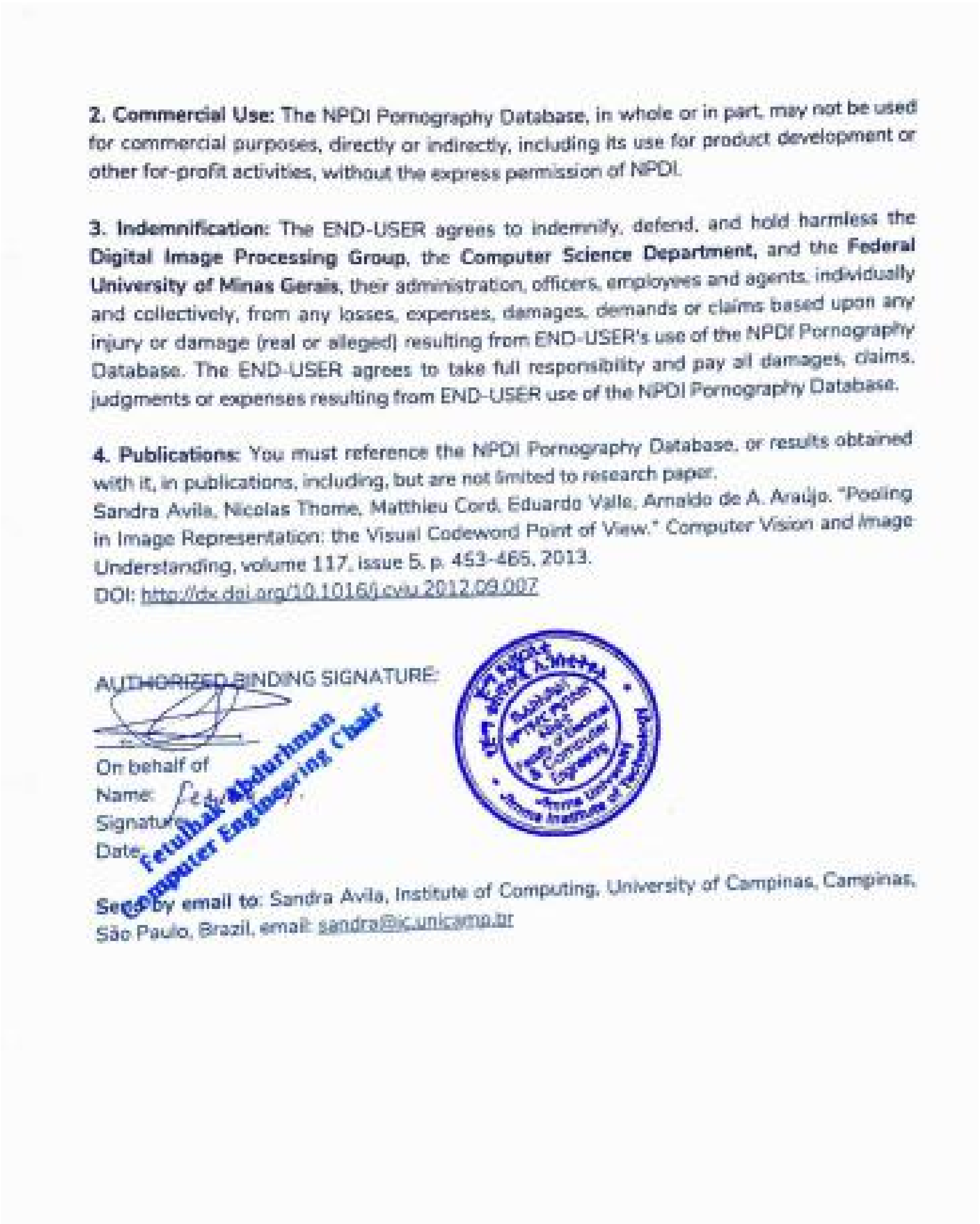
Release of the Database

The NPDI Pornography Database is available to researchers under request, which will be evaluated case-by-case. All requests for the NPDI Pornography Database must be submitted to Prof. Sandra Avila. To receive a copy of the database, END-USER named must be a legal institution, or a department or section of a named legal institution, not an individual nor a project – the person signing this Agreement has to be duly authorized by the institution for such signatures (e.g., Department or Administrative Head or similar) and shall be liable for such authorization.

Terms and Conditions

By signing this document, END-USER agrees to receive a copy of the NPDI Pornography Database, under the following restrictions and obligations:

1. Redistribution: the NPDI Pornography Database cannot be, in whole or in part, further distributed, published, copied, disseminated or broadcast, in any form, whether for profit or not. That includes further distribution to a different facility or unit within the END-USER's university, organization, or company. An exemption is provided for a small number of individual static frames of the videos (less than 50 on any individual publication), which can be used for illustrations of scholarly posters, presentations and publications.



2. Commercial Use: The NPDI Pornography Database, in whole or in part, may not be used for commercial purposes, directly or indirectly, including its use for product development or other for-profit activities, without the express permission of NPDI.

3. Indemnification: The END-USER agrees to indemnify, defend, and hold harmless the **Digital Image Processing Group**, the **Computer Science Department**, and the **Federal University of Minas Gerais**, their administration, officers, employees and agents, individually and collectively, from any losses, expenses, damages, demands or claims based upon any injury or damage (real or alleged) resulting from END-USER's use of the NPDI Pornography Database. The END-USER agrees to take full responsibility and pay all damages, claims, judgments or expenses resulting from END-USER use of the NPDI Pornography Database.

4. Publications: You must reference the NPDI Pornography Database, or results obtained with it, in publications, including, but are not limited to research paper.
Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, Arnaldo de A. Araújo. "Pooling in Image Representation: the Visual Codeword Point of View." Computer Vision and Image Understanding, volume 117, issue 5, p. 453-465, 2013.
DOI: http://dx.doi.org/10.1016/j.cviu.2012.09.007

AUTHORIZED BINDING SIGNATURE:

On behalf of
Name: Fetu
Signature:
Date:

Fetulhak Abdurhman
Computer Engineering Chair

Faculty of Electrical & Computer Engineering
Jimma University
Jimma Institute of Technology

Send by email to: Sandra Avila, Institute of Computing, University of Campinas, Campinas, São Paulo, Brazil, email: sandra@ic.unicamp.br

APPENDIX E: HYPERPARAMETER TUNING

E.1 Epoch vs Accuracy

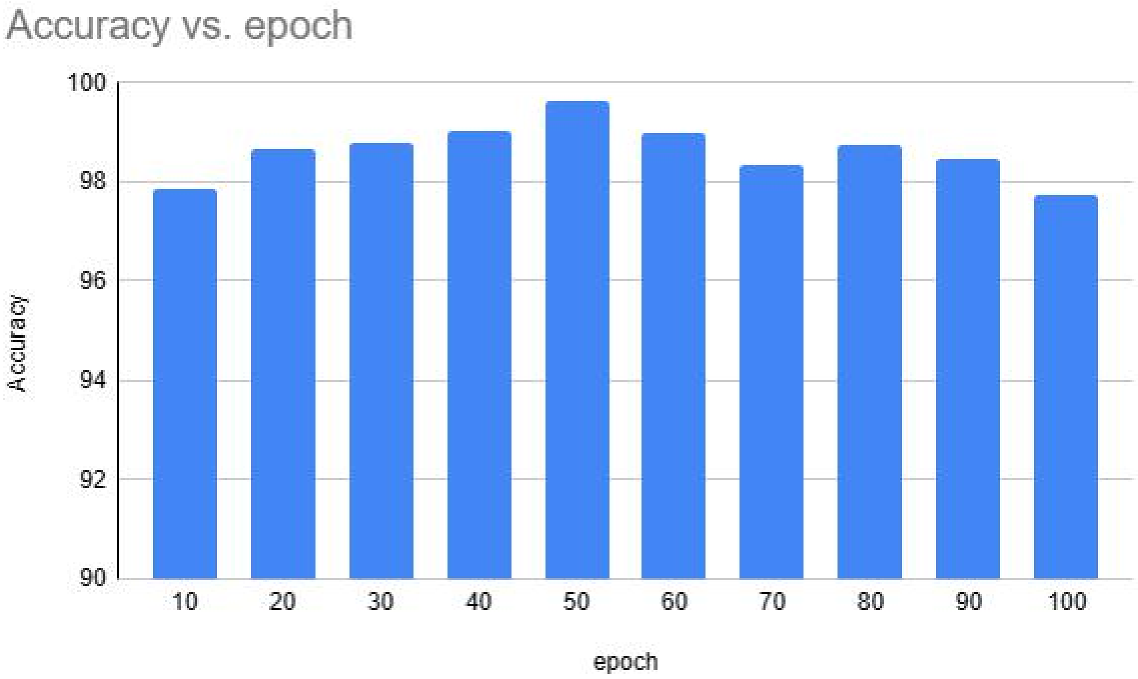


Figure 0- 1: Result with Epoch vs Accuracy

E.2 Effect of Optimizers

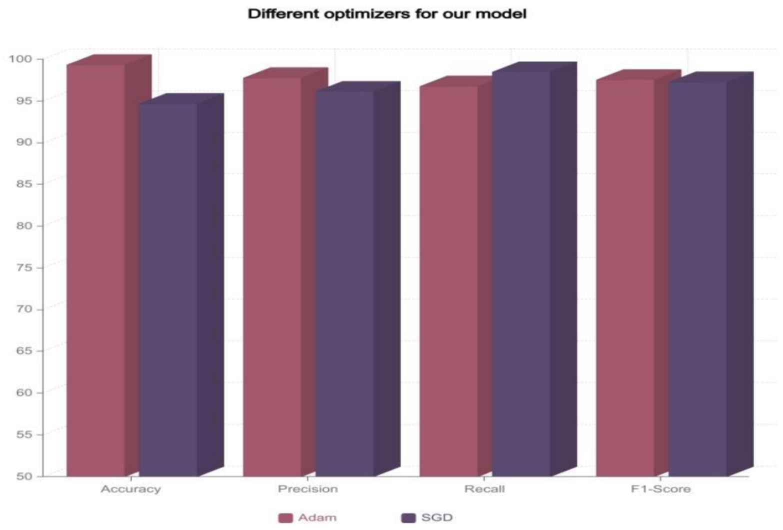


Figure 0- 2: Different optimizers for our model

E.3 Effect of Batch size

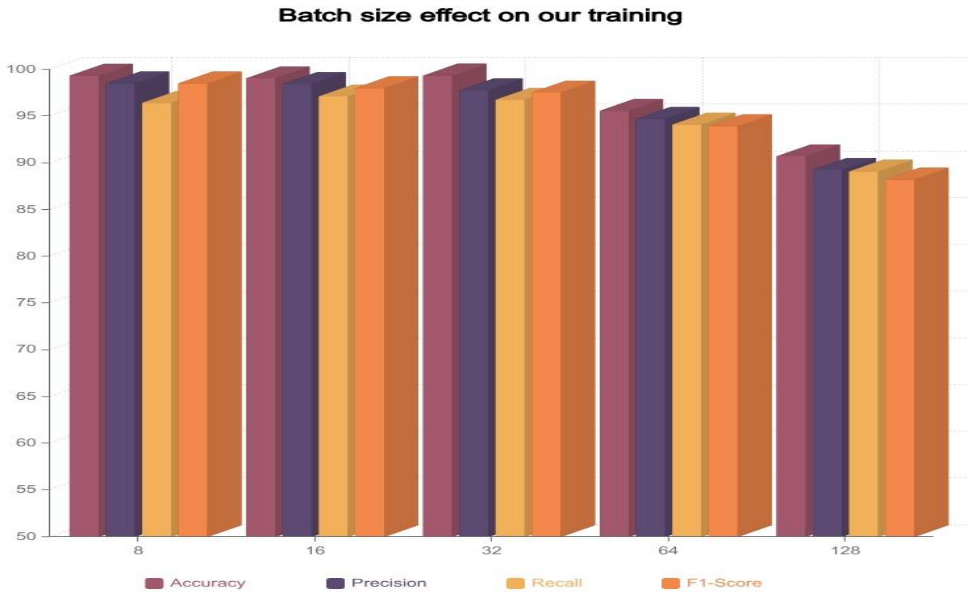


Figure 0- 3: Batch size effect on our training

E.4 Effect of learning rate

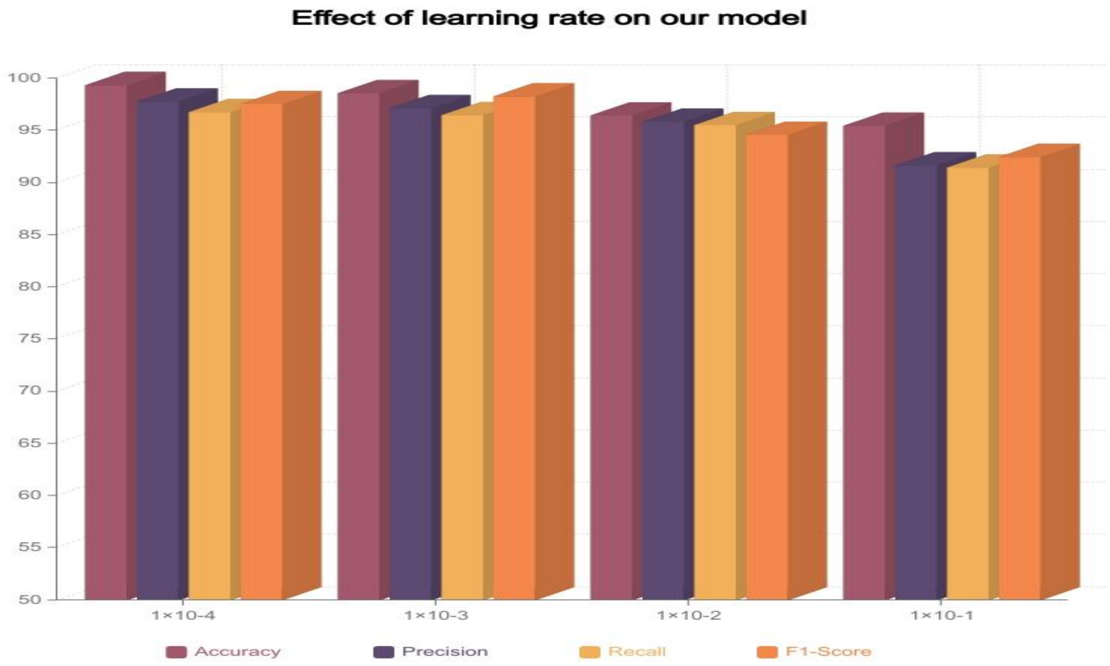


Figure 0- 4: Effect of learning rate on our model

E.5 Effect of Dropout

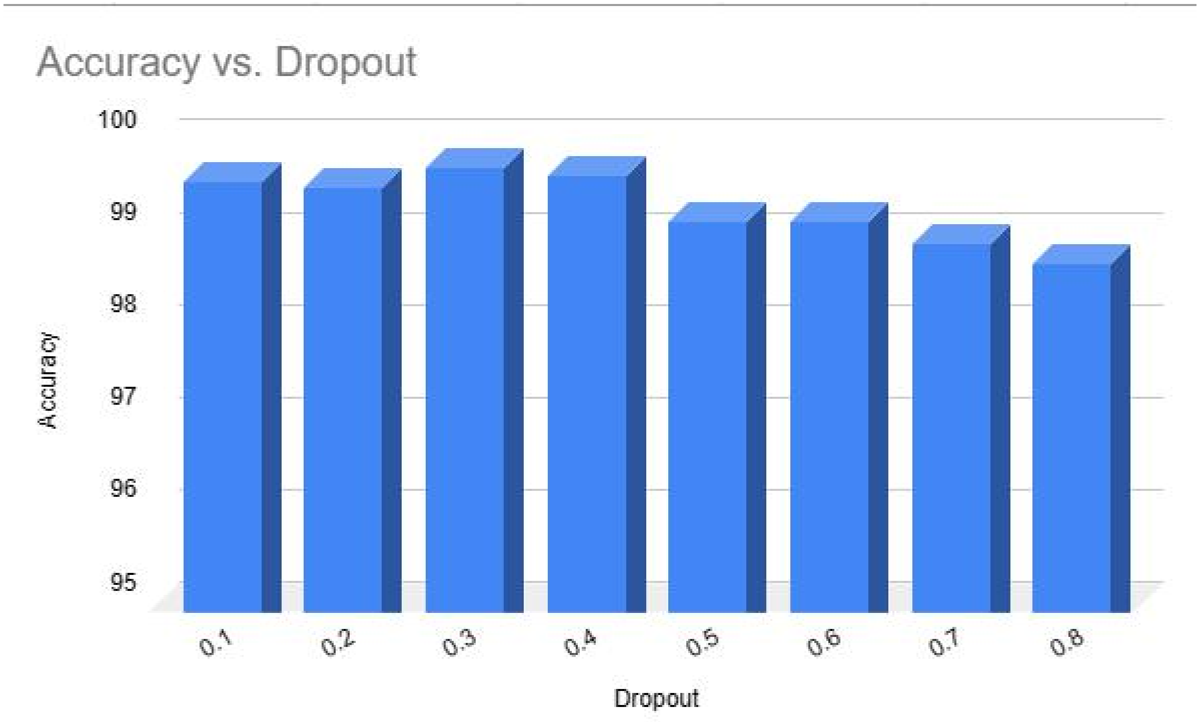
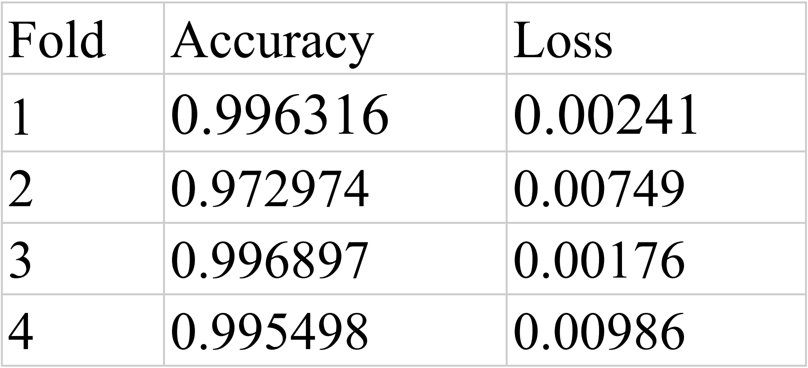


Figure 0- 5: Accuracy vs Dropout

E.6 Result for K-Fold Cross Validation



Fold	Accuracy	Loss
1	0.996316	0.00241
2	0.972974	0.00749
3	0.996897	0.00176
4	0.995498	0.00986