AUTOMATED DETECTION OF VISUAL CONTENTS FOR FILM CENSORSHIP

by

HOR SUI LYN

1161300122

Session 2019/2020

The project report is prepared for
Faculty of Engineering
Multimedia University
in partial fulfilment for
Bachelor of Engineering (Hons) Electronics
majoring in Computer

FACULTY OF ENGINEERING
MULTIMEDIA UNIVERSITY
February 2020

© 2016 Universiti Telekom Sdn. Bhd. ALL RIGHTS RESERVED.

Copyright of this report belongs to Universiti Telekom Sdn. Bhd. as qualified by Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialisation Policy. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Universiti Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

DECLARATION

I hereby declare that this work has been done by myself and no portion of the work contained in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

I also declare that pursuant to the provisions of the Copyright Act 1987, I have not engaged in any unauthorised act of copying or reproducing or attempt to copy / reproduce or cause to copy / reproduce or permit the copying / reproducing or the sharing and / or downloading of any copyrighted material or an attempt to do so whether by use of the University's facilities or outside networks / facilities whether in hard copy or soft copy format, of any material protected under the provisions of sections 3 and 7 of the Act whether for payment or otherwise save as specifically provided for therein. This shall include but not be limited to any lecture notes, course packs, thesis, text books, exam questions, any works of authorship fixed in any tangible medium of expression whether provided by the University or otherwise.

I hereby further declare that in the event of any infringement of the provisions of the Act whether knowingly or unknowingly the University shall not be liable for the same in any manner whatsoever and undertakes to indemnify and keep indemnified the University against all such claims and actions.

Signature: _____

Name:

Student ID:

Date:

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude towards my supervisor, Dr Sarina binti Mansor, for her guidance, motivation, assistance, patience and enthusiasm. She paved the way for me to work on this project, guiding me on the tasks to be done after each milestone was reached. She also shared her knowledge and insights that were related to the topic, aiding me in understanding the terminologies in more depth. She boosted my confidence, allowing me to brave through the journey of completing this project.

Next, I would also like to thank my mentor, Nouar AlDahoul. Being a PhD student researching on deep learning techniques, she answered to my questions whenever I was unsure of the validity of my generated results. She also helped me in building up the dataset used in this project and gave suggestions on ways to implement the system. Furthermore, a special thanks is sent to Haziq Imran bin Hanip who answered to my questions regarding this project.

Moreover, the Faculty of Engineering of Multimedia University is to be thanked for providing me the golden opportunity to work on this project. I was able to gain knowledge on deep learning, which was new to me as it was not part of my curriculum.

Last but not least, I am thankful of my family and friends who supported me and answered to my questions while I was working on this project. Without them, it would be hard for me to deal with the stress alone.

ABSTRACT

Content filtering is gaining popularity due to easy exposure of explicit visual contents to the public. Excessive exposure of inappropriate visual contents can cause devastating effects such as the growth of improper mindset and rise of societal issues such as free sex, child abandonment and rape cases. At present, most of the broadcasting media sites are hiring censorship editors to label graphic contents manually. Nevertheless, the efficiency is limited by factors such as the attention span of humans and the training required for the editors. This project proposes an automated detection system to filter out inappropriate contents in films. Frames of films can be fed as input into a Convolutional Neural Network (CNN), allowing classification of the inputs into two groups: porn or non-porn. Three CNN architectures: MobileNet, Visual Geometry Group-19 (VGG-19), Residual Network-50 Version 2 (ResNet50_V2), and two classifiers: CNN and Support Vector Machine (SVM) were used to explore the combination that produce the best result. Accuracies achieved by all combinations were above 80%. The best accuracy was 92.80% obtained using fine-tuned ResNet50_V2 as feature extractor and SVM as classifier. Transfer learning and SVM have improved the CNN model by approximately 10%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACORDE	Adult Content Recognition with Deep Neural Networks
AGNet	Ensemble-ConvNet classifier
ANet	AlexNet-based classifier
API	Application Programming Interface
CNN	Convolutional Neural Network
FYP	Final Year Project
GNet	GoogLeNet-based classifier
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IDE	Integrated Development Environment
KNN	K-Nearest Neighbour
LSTM	Long Short-Term Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
SVM	Support Vector Machine
VGG	Visual Geometry Group

CHAPTER 1 INTRODUCTION

1.1 Overview

Tight regulations are exercised in Malaysia regarding censorship of visual contents that involve pornography and nudity on screen. As a consequence, media groups have to censor or remove sensitive visual contents from being broadcasted to prevent violation of the law and penalties that follow such violation. The focus of this project is placed on visual contents as it is the most salient form of content. A concrete evidence of this statement is the nature of humans to learn by imitating or mimicking actions of others before they can even understand the meaning behind the said actions. Specifically, the type of visual content targeted in this project is pornographic related graphics that appears in films. Definition of the term "pornography" is subjective and may differ from one person to another. In this report, the definition of pornography is "any sexually explicit material with the aim of sexual arousal or fantasy", which originated from [1].

With the encouragement of open-mindedness in all aspects of life and society and the booming consumption of Internet, violence, profanity, drug abuse, nudity and other controversial contents that are unconstrained and freely accessible can easily be exposed to the public through media and social media with or without the viewers' intentions. As adolescents and children may not have the suitable mentality to differentiate the rights and wrongs, it is possible for them to have inaccurate information ingrained in their young minds when exposed to inappropriate visual contents. This can alter their mindsets into thinking that such inappropriate contents are normal, which may lure the curious ones into diving into the abyss of pornography or sexual addiction, and trick the naïve ones into becoming blackmail victims of nude photos, or other similar yet disastrous cases such as making illogical choices of committing sexual assault crimes. The fact that such distortion of healthy sexual development occurs in adolescents is supported by the findings of a review on impact of internet pornography on teenagers [2]. As a result, it is critical for the

society to avoid the arise of such unfavourable circumstances from the bud of the problem.

In order to reduce, if not eliminate, the occurrence of such circumstances, preventive measures are to be taken by the moderators of channels that can expose sexual explicit visual contents to the public. Censorships or filters need to be applied to contents deemed to be inappropriate in the ethical sense. The censors applied serve to block viewers from the direct exposure to the specific contents, protecting the public from the possible influence of the censored content.

1.2 Problem Statements

Traditionally, censorship editors are hired to filter materials to be broadcasted or published manually. The need for such jobs is because genres or tags can no longer be used to segregate films that need to be censored as any type of film may include the inappropriate visual contents targeted in this project. Nevertheless, the editors have to go through the films one by one manually in order to identify films containing banned contents. Such tedious routine increases the difficulty of manual film censorship, and the productivity is limited by humans' attention span. In addition to that, the editors need to be briefed and trained to know the ever-changing filtering standards set by the respective authorities, which reduces the efficiency of the censorship system significantly.

A solution to this problem is to automate the detection and filtering task, allowing computers to take over humans' job. By utilising automated detection of inappropriate visual contents, properties of each key frame extracted from the films will be analysed. In this case, challenge exists due to the wide range of colour and texture of the exposed skin area in the images, especially in the frames of anime episodes.

1.3 Project Objectives

The objectives of this project are:

1. To obtain and study pornographic data set from existing data set
2. To study various image processing techniques to detect explicit scene from video
3. To evaluate various Convolutional Neural Network (CNN) features and their encoding strategies

1.4 Project Scope

The visual contents that are deemed to be inappropriate and should be detected in this project are those that contain nudity of private part(s) and/or those that intend to stimulate sexual arousal or excitement. The former includes display of human's buttocks and external genital organs, which include but not limited to penis and scrotum for males, and areola, nipple and vulva for females. However, breastfeeding is an exception in this project. Although female's breast (areola) may be displayed when she is feeding a baby, it is considered appropriate and this scene is allowed to be shown and thus should not be detected as content that should be censored.

As the visual contents that are to be assessed originate from films or videos, video frames, which are essentially images, will be extracted, processed and checked with the algorithm to determine whether each of them contains any negative content. If the answer to that is positive, further action to manipulate it is required such that the said content is hidden from view. However, only the detection process is covered within the scope of this project. Filtering of images containing adult contents will not be implemented on the video files. In other words, output of the algorithm produced as a result of this final year project (FYP) will not be the censored version of the original video input. Instead, frames of the video files fed into the system will each be labelled as "Porn" or "Non Porn" based on the output of the algorithm.

Subsequently, the video files will each be assigned a label depending on whether the percentage of frames with “Porn” label exceed the threshold set by the user.

The emergence of deep learning techniques, which use neural networks to simulate human-like decision making process, in the field of artificial intelligence introduced the possibilities of automated visual recognition performed by machines. By representing an image as a matrix of pixels, computers are able to understand context of the image by decomposing it into features of different simplicity levels. Deep learning allows computers to pick and extract features from images automatically, without any human intervention, provided the system is fed with a large volume of data in the training stage. An instance of a deep learning model is illustrated in **Figure 1.1**. Aside from input to and output from the model, other layers in the model are “hidden layers”, each of which extracts specific abstract feature(s) from the image, with increasing abstract level according to the depth of model [3]. Among the many techniques, CNN specialises in automatic pattern and object detections to the extent that it is able to outperform the performance of humans in terms of accuracy and speed in such tasks. Therefore, CNN is selected to be applied in this project to design a system to detect inappropriate visual contents for the purpose of censorship.

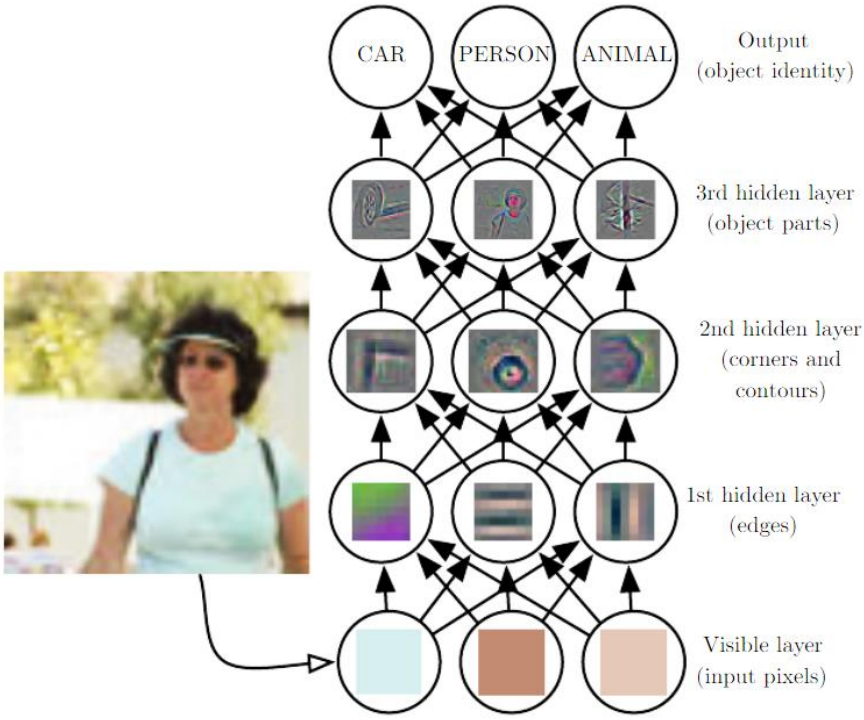

Figure 1.1: Instance of a Deep Learning Model [3]

The specifications of the platform selected to run the proposed model are as follow:

- AMD Ryzen 5 2600 Six-Core Processor @ 3.40GHz
- Windows 10 Pro
- 16.0GB DDR4 at 2666MHz
- NVIDIA GeForce GTX 1660

1.5 Report Outline

An introduction to the topic and title of this FYP project, including the reasons for choosing this title, severity of the problems if they are not addressed, objectives and scope of this project, are placed in **Chapter 1**. A brief idea on the contents of this report is also included in same chapter. Next, the findings of researches done on past attempts carried out to solve similar problems by other researchers are presented in **Chapter 2**. The methods proposed by those researchers in their respective published papers are analysed to uncover the possible advantages and limitations. Moreover, the architecture and details of the system to be implemented will be discussed in **Chapter 3**. Furthermore, in **Chapter 4**, results obtained from the implementation of the system, using the algorithms or methods mentioned in **Chapter 3**, shall be presented and analysed, together with screenshots of the implemented software. Lastly, the conclusion of this report and the areas that can be focused to improve the system in the future will be incorporated in the last chapter, **Chapter 5**.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

In this chapter, existing solutions that solved the problems similar to this project were researched to keep up-to-date with the algorithms and techniques that can be utilised in this project. Since videos are essentially sequences of images, or video frames, published papers that proposed methods for detection of pornographic contents in images and videos were studied to learn the feasibility of applying the said techniques in solving the current problem.

From the published papers, it was discovered that digital image processing techniques, Support Vector Machine (SVM) classifier, deep learning and motion information extracted from moving images were deployed in the existing solutions proposed by other researchers. Each of the chosen existing solutions will be discussed from **Section 2.2** to **Section 2.4** according to the technique employed (digital image processing, SVM classifier, deep learning) and chronological order of the publishing of the research paper. This will be followed by a short conclusion in **Section 2.5**.

2.2 Digital Image Processing Techniques

2.2.1 Implementation of K-NN Based on Histogram at Image Recognition for Pornography Detection

Nuraisha et al [4] utilised digital image processing techniques (from pre-processing, segmentation, representation and description to recognition) to recognise images with explicit contents. In brief, values from histogram generated based on distribution of skin pixels in images were utilised to determine whether a positive reaction was detected based on K-Nearest Neighbour (KNN) algorithm. The flowchart of the system is summarised in **Figure 2.1**.

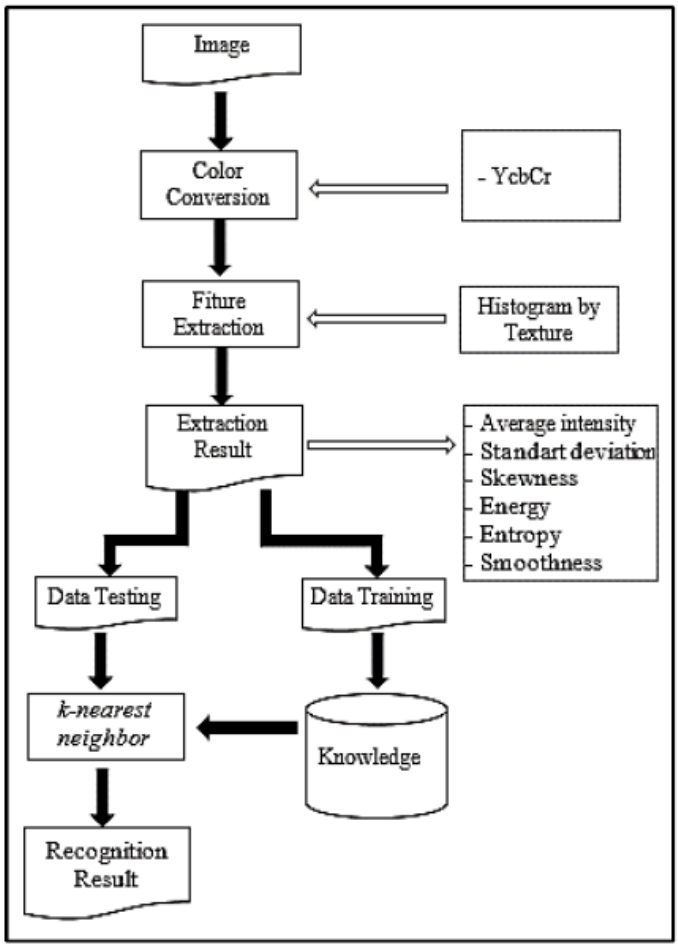

Figure 2.1: Flowchart of the System Utilising Histogram-based K-NN [4]

Images in the red, green, blue (RGB) colour space were first pre-processed by normalising them and then converting them into the YCbCr colour space which represents images more similar to human eye perception where sensitivity towards colour exceeds that of brightness [5]. Next, the pre-processed images underwent colour segmentation by setting the threshold on only the chroma components, Cb and Cr, in order to segregate the skin regions. The segmented images were transformed into binary images by changing skin pixels into 0 pixels (black) and the rest into 1 pixels (white), as shown in **Figure 2.2**, in order to allow the plotting of histogram describing the spread of intensity values for each image. Features or properties of each image were then extracted from the histogram by computing the skewness, smoothness, energy, entropy, average intensity and standard deviation. Similarities between properties of images were exploited to identify whether an explicit image was recognised. This was accomplished by using the KNN algorithm, where the

distance between a query data and the learning data was computed using the Euclidean distance formula.

Figure 2.2: Binary Image Generated After Segmentation [4]

The accuracy rate achieved by the proposed system was 90% when the dataset (60 images) used were images obtained using Google search engine. For this system, a very small number of images was required for training (2 images for each of the categories). Since this system used an algorithm based on the distribution of exposed skin region, pornographic images with distributions of exposed skin areas that are different from those of the images used for training will generate false negative results. In other words, usage of this system is limited to cases where distributions of exposed skin region are similar to those of the images used for training.

2.2.2 A Pornographic Image and Video Filtering Application Using Optimised Nudity Recognition and Detection Algorithm

Garcia et al [6] also approached the detection of pornographic visual contents using digital image processing techniques with rule-based classification in the year 2018. The system flowchart is illustrated in **Figure 2.3**.

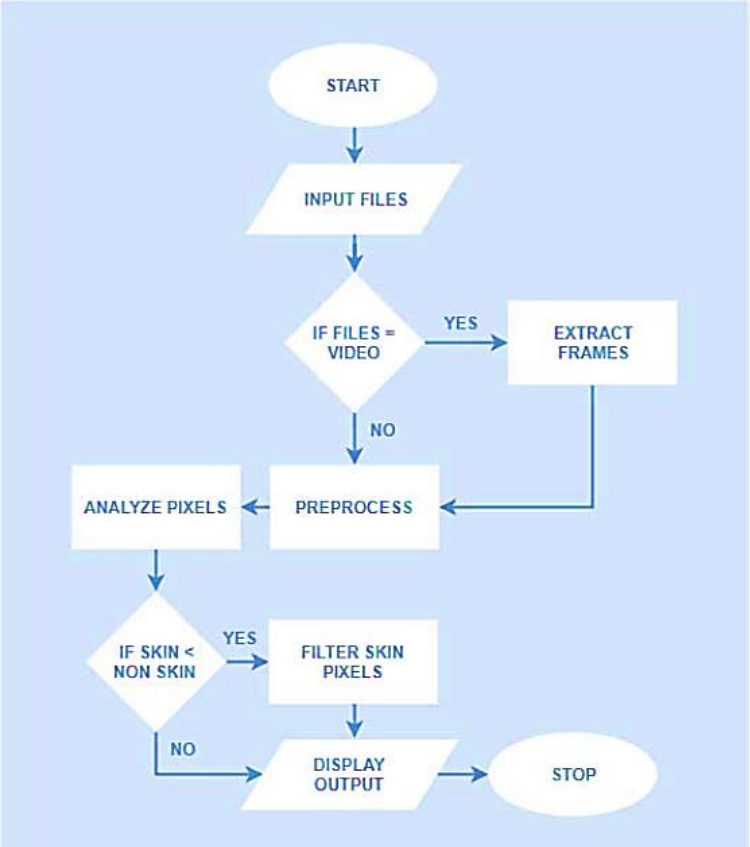

Figure 2.3: Flowchart for Nudity Filtering Process [6]

In their application, the images and extracted frames of videos were pre-processed to highlight the details and segmented to extract the skin areas. Filters were then applied on each pixel in the YCbCr colour space to separate and extract the skin areas, which were those that fulfill the mean baseline or a static threshold value. Of all the available colour models, YCbCr colour model was utilised to describe the colours mathematically due to the fact that it is superior to the other colour models [7]. It also exploits human eye properties such as higher sensitivity to light intensity than colour [5]. Texture filter was used to exclude falsely labeled skin pixels from the extracted skin regions. The next process involved removal of the non-skin pixels from the original images and video frames, as shown in **Figure 2.4**, followed by the classification of the images and frames into one of the two possible classes: pornographic or non-pornographic, according to rules that involved probabilities of skin pixels across the image and in each region or cluster, and average intensity of segmented skin areas.

Figure 2.4: Segmentation of Coloured-skin [6]

The accuracy rate of this model was 80.23% when applied on the supplied dataset which consisted of images and video frames. This was acceptable as the pixel-by-pixel segmentation scheme used was simple and easy to implement. Other than that, it was stated in the paper that this application was capable of detecting skin regions of different colours in various lighting conditions, which formed one of its strong points.

It was mentioned in the paper that this application was built on the assumption that pornographic related contents involve nudity, or in other words, large areas of skin being exposed. However, this statement is not entirely true, hence high false positive rate may be encountered when a large number of images which contain high skin pixel percentage is input into the application, for instance, in close up arm, torso or face images. Although this paper addressed this issue by setting rules for classification, many conditional expressions needed to be included in the algorithm in order to minimize the false positive rate, which is tedious and causes the application to be prone to errors.

2.3 SVM Classifiers

2.3.1 Combining Multiple SVM Classifiers for Adult Image Recognition

Zhao and Cai [8] implemented a system in 2010 to detect adult images using several SVM classifiers. Six different categories were set up for different types of pornographic images, some with exposed private parts and some without, as they claimed that it was difficult to have a detector capable of detecting all the cases at that time. **Figure 2.5** shows the overview of the proposed system.

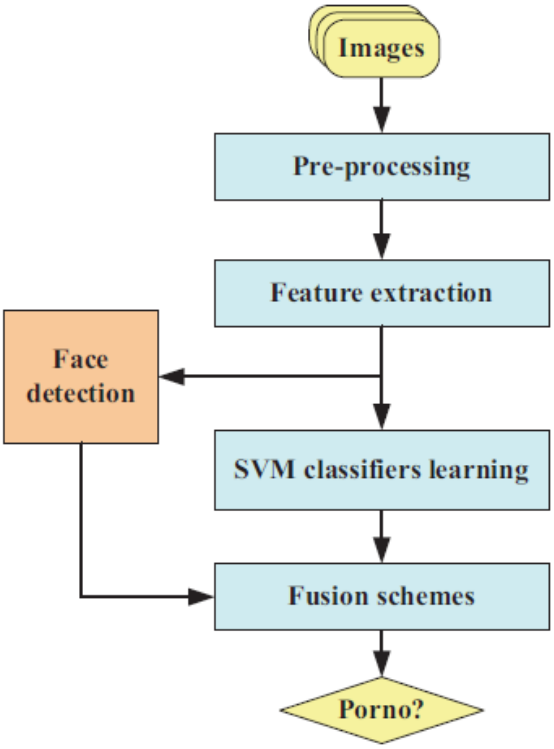

Figure 2.5: Overview of the Proposed System that Uses SVM Classifiers [8]

The pre-processing phase involved applying a filter to smoothen the input images and subsampling the resulting images. Extraction of visual features (colour, skin, edge, texture and key-point features) was done at the local, regional and global levels. Different features and feature extraction methods were used for different categories, some of them are shown in **Table 2.1**. All the extracted features were then fed into SVM classifiers, one for each class, to undergo "one-against-the-rest" training process using 5-fold cross-validation via supervised learning. Subsequently, the

results generated from the different classifiers were combined. Face detection algorithm was included to exclude close-up face images from being categorised as adult image when the pre-determined threshold for exposed skin area was exceeded, boosting the precision of the system.

Table 2.1: Features Selected to Represent the Pornographic Categories [8]

Pornographic Category	Selected Features
Hardcore	HSV correlogram, Gabor, SIFT, skin, RGB moment
Nude	HSV correlogram, skin, LBP, RGB moment
Blowjob	HSV correlogram, Gabor, LBP, skin, RGB moment
Breast	HSV correlogram, Gabor, LBP, EDH, skin, RGB moment
Underparts	HSV correlogram, Gabor, LBP, EDH, SIFT, skin, RGB moment

The accuracy of the system was not stated in the paper but it was specified that a true positive rate of 87.68% and a false negative rate of 14.17% was achieved by this system when it was tested using 50,000 images retrieved from the Internet.

A drawback of this framework is the need to specify the features to be extracted for each category depending on the properties of images of that category. In addition to this, the inclusion of face detection algorithm disregards close-up pornographic images from the positive results, generating false negatives.

2.4 Deep Learning

2.4.1 Applying Deep Learning to Classify Pornographic Images and Videos

In 2015, Moustafa [9] designed a deep learning classifier named Ensemble-ConvNet classifier (AGNet) to perform classification of pornographic images and videos.

AGNet was a product of fusion of modified CNN architectures called AlexNet-based classifier (ANet) and GoogLeNet-based classifier (GNet). Weights obtained from classification of ImageNet dataset were imported as initial weights for each of the classifiers during the training stage. Only the last layer of AlexNet model (for ANet) [10] and GoogLeNet model (for GNet) [11] were trained using the new dataset to allow the network to accommodate to the targeted visual contents. The architectures of ANet and GNet are shown in **Figure 2.6** and **Figure 2.7** respectively.

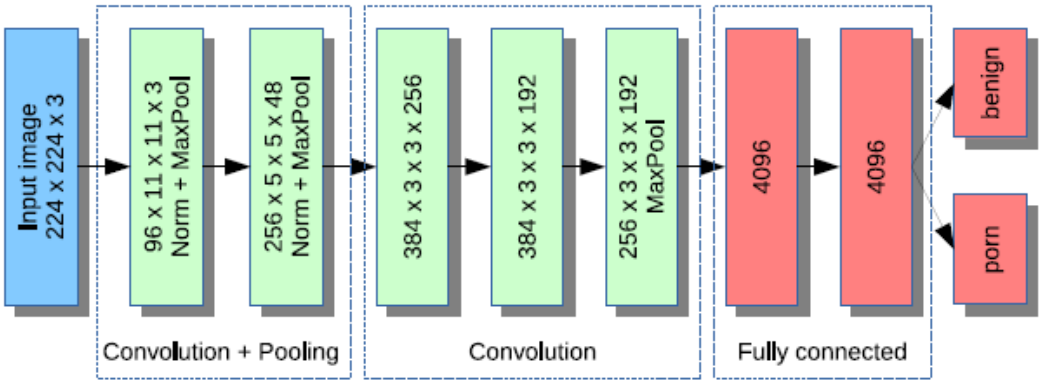

Figure 2.6: ANet Architecture [9]

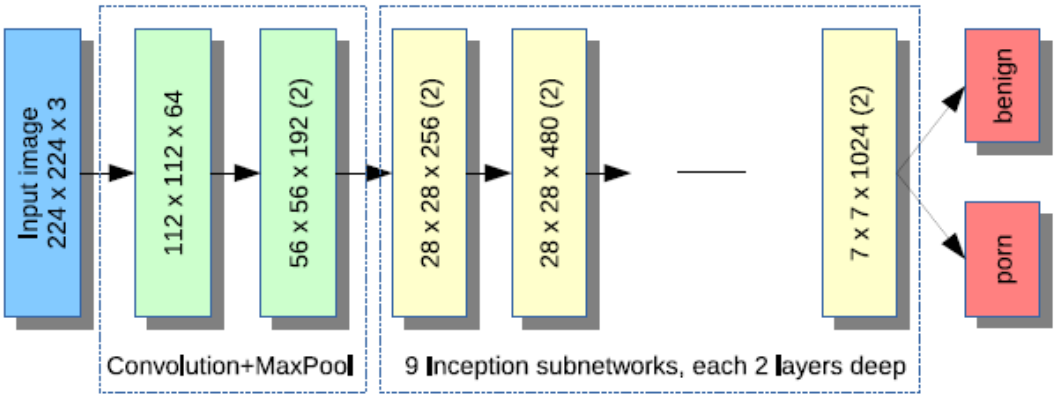

Figure 2.7: GNet Architecture [9]

AGNet was built based on the assumption that ANet and GNet produce different classification errors. In AGNet, after being pre-processed (image rescaling, subtraction with the mean image and division of image into smaller windows) [10], images and video key frames would be classified using ANet and GNet separately. The scores generated by each of the classifiers were then averaged and compared with a predetermined threshold value, which was set as 50% by default. For the case of videos, classification was done using majority voting on all key frames or by adding the scores for all key frames of a video and taking the class with the maximum total score if majority voting results in a draw. The design of the proposed system is shown in **Figure 2.8**. Another variation of AGNet, named AGbNet, performed the image and video key frame classification by comparing the maximum score between the ANet and GNet generated scores with a threshold value.

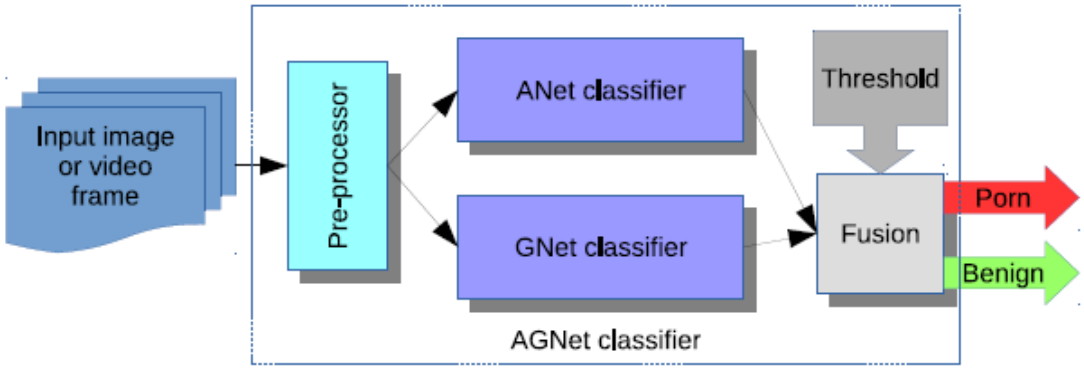

Figure 2.8: AGNet Architecture [9]

For video classification, AGNet managed to achieve an accuracy of 93.8% when applied on NPDI benchmark dataset while AGbNet obtained 94.1% when tested on the same dataset. This approach, however, disregards cases where ANet and GNet classifications generate similar errors. Another limitation of this approach is the higher requirement for computational power of system if ANet and GNet run in parallel.

2.4.2 Pornographic Image Detection Utilising Deep Convolutional Neural Networks

In the following year (2016), Nian et al [12] brought up a system which used deep CNN as a pornographic image detector. Transfer learning was applied on the CNN model and adjustments to training data was done by utilising results obtained from analysis of network validation process. The architecture of the proposed system is illustrated in **Figure 2.9**.

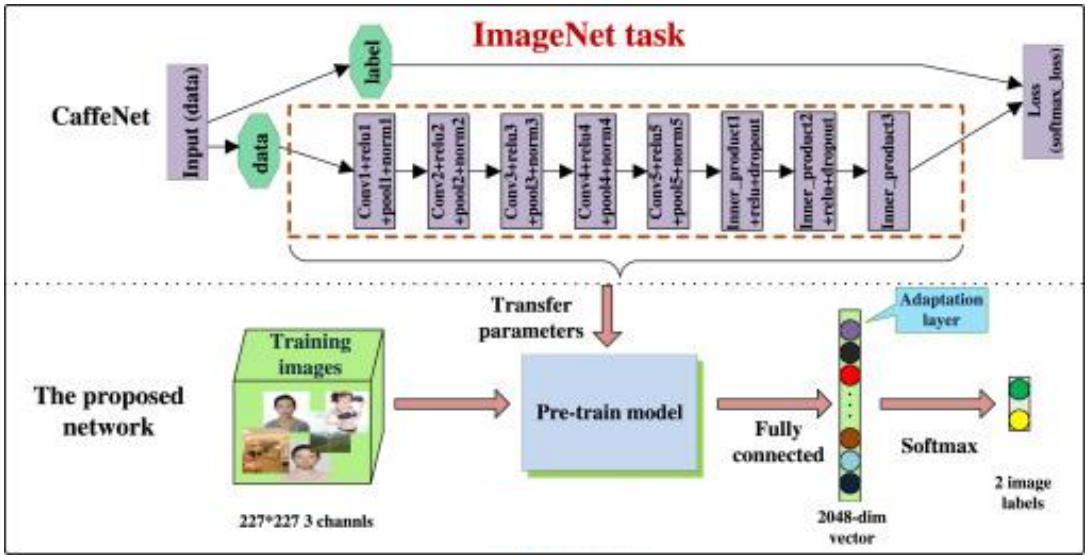

Figure 2.9: Architecture of the Proposed System in [12]

From the top part of **Figure 2.9**, a low complexity CNN architecture named CaffeNet was selected to learn the pre-trained parameters as the ImageNet classification was performed by the selected network. The learnt parameters were then transferred to the pre-trained network as initial values in the proposed system.

From the bottom part of **Figure 2.9**, RGB input images were resized such that the smallest side was scaled to 227 to allow the extraction of ten 227 x 227 subimages or image patches to ensure most of the adult content appear in square subimages. Then, manipulation of data was done via data augmentation method to allow the proposed system to be less affected by factors such as lighting condition, colour, blurring of images and orientation of objects, thus increasing robustness of the system. Subsequently, the resulting images were input into the pre-trained model. Unlike the

implementation in [13] where the transferred parameters were frozen or fixed, here, they were trainable in order to retrieve better mid-level features suited for the targeted dataset.

In order to improve the results obtained from the transfer learning process, adaptation of the training set was performed using the auxiliary training set by finding the relationship between the detection result on validation set and training set distribution. This process of fine-tuning the system was repeated until a convergence on test dataset accuracy occurs. The flowchart of the system is depicted in **Figure 2.10**.

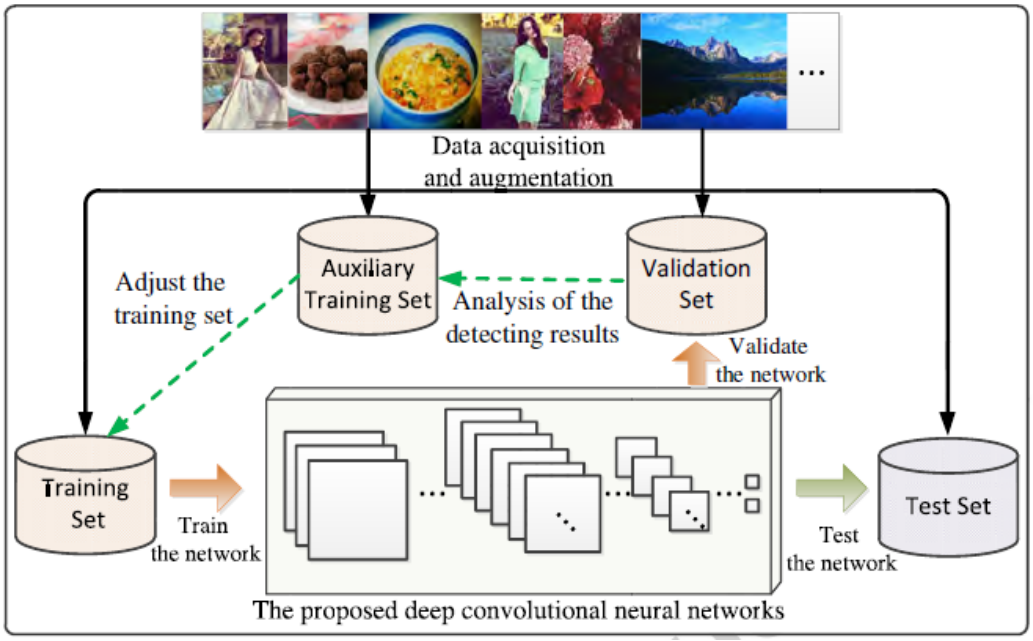

Figure 2.10: Flowchart of the Proposed System [12]

To compensate for the increased complexity, fixed-point algorithm was utilised to speed up the computation time. All floating point parameters were taken only up to four digits after the decimal point and converted to integers by multiplying a suitable factor. As a result, the computation time can be four times faster by compromising only 0.86% of accuracy. The overall accuracy of this system was 98.6% when it was used on the test data.

2.4.3 Adult Image and Video Recognition by a Deep Multicontext Network and Fine-to-coarse Strategy

In 2017, Ou et al [14] investigated usage of fine-to-coarse strategy to implement a deep multicontext network to recognise adult contents in images and key frames extracted from videos. Their system (DMCNet) was deployed in the open source platform CAFFE [15]. The architecture of this system is depicted in **Figure 2.11**.

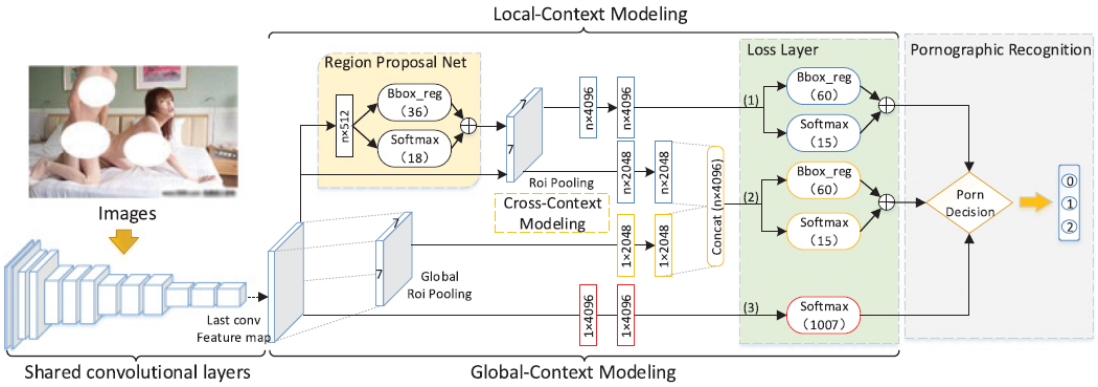

Figure 2.11: Architecture of DMCNet [14]

Contexts or features in both global and local perspectives were taken into account separately in different modules: Deep Faster R-CNN-based [16] local context modeling for local object detection (upper branch of **Figure 2.11**) and deep CNN-based modeling for global context on full frame images (lower branch of **Figure 2.11**). Minor errors of the local context were utilised to rectify the judgement of the global context that was mostly accurate using cross-context modeling (middle branch of **Figure 2.11**). Convolutional layers with parameters initialised using ImageNet classification task and fine-tuned on the targeted dataset were shared among the three previously mentioned branches. The final decision was made based on classification results produced by each of the branches, taking complementarity and differences among information of different contexts into account and filtering illegal samples by deploying a hierarchical algorithm for selection.

The fine-to-coarse strategy used in this system referred to the classification of objects in each input image into fine-grain categories, in which each fine-grain category was

then grouped into coarse-grained categories, as can be seen in **Figure 2.12**. The advantage of this strategy, as opposed to the conventional coarse-to-fine method, was the acceleration of object recognition and detection due to the limited number of fine-grained categories, which could be particularly useful when large datasets were involved. Hence, it was employed in all global, local and cross-context modules.

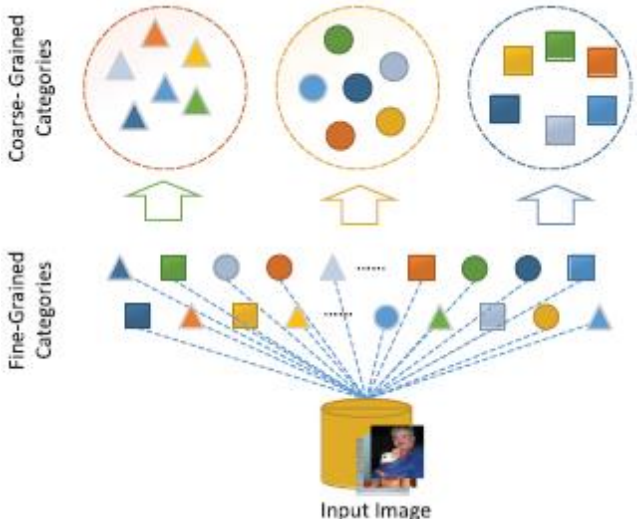

Figure 2.12: Fine-to-Coarse Strategy [14]

The dataset used was grouped into two types: L3 with three-level evaluation pattern and L2 with two-level evaluation pattern. In L3 dataset, images were grouped into three different classes: normal images, adult images and images with contents unsuitable for children such as swimwear, underwear and beauty legs. On the other hand, L2 dataset combined the second and third classes of L3 dataset into one, so it consisted of two different categories. The proposed DMCNet was able to achieve accuracies of 99.1% and 97.8% when L2 and L3 datasets were used respectively. Despite the excellent performance, the complexity of this system is rather high, thus requiring a longer development time. Furthermore, the overall accuracy of system is dependent on the degree which the information of the global context is affected by the local context.

2.4.4 Convolutional Neural Network for Pornographic Images Classification

Agastya et al [17] designed and developed an application which used a CNN architecture named Visual Geometry Group-16 (VGG-16) [18] to perform classification of images with adult contents in 2018. The said application was implemented using Python programming language and Keras framework. The schematic of VGG-16 layers is shown in **Figure 2.13**.

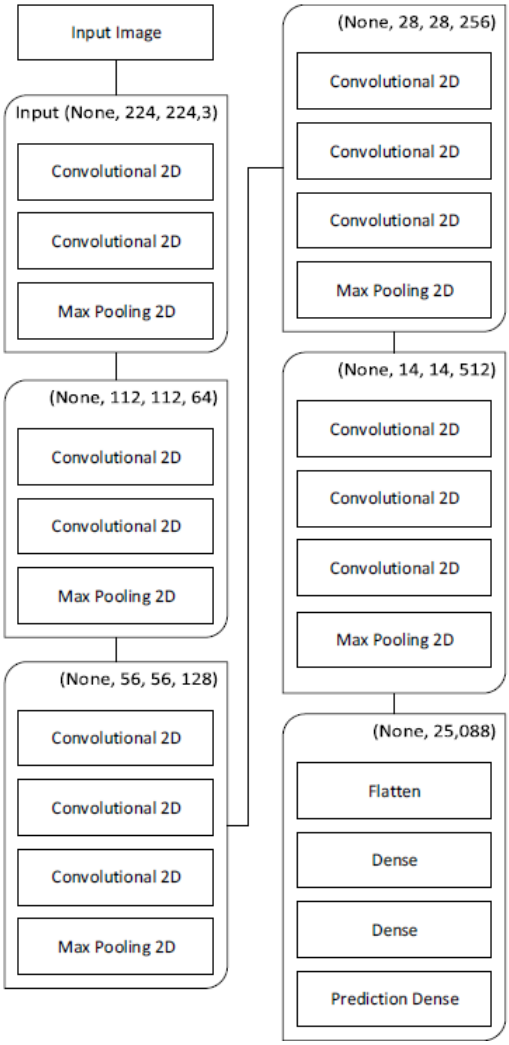

Figure 2.13: Schematic of VGG-16 [17]

This paper studied the effects of transfer learning and k-fold cross-validation in CNN training. Transfer learning was performed in the fully-connected layer of the last block of the selected architecture, leaving connection weights of all the previous layers to be frozen or untrainable via backpropagation. 5-fold cross-validation was

carried out on the dataset, where images in the dataset were separated into 5 groups, and rounds of training were performed on the 5 groups where one of the five groups would take turn to be used for testing instead of training. The overall accuracy was computed by averaging the five accuracy values obtained. The popular pornographic dataset, NPDI, was utilised to confirm the accuracy of the application. As a result, the training accuracy achieved by the application was 95.4% and the testing accuracy was 93.8%.

An interesting observation highlighted in the paper is shown in **Figure 2.14** where testing accuracy does not improve as epoch increases. It was mentioned that such phenomenon is likely due to the imbalance in difficulty of data categorisation in the current fold compared to other folds.

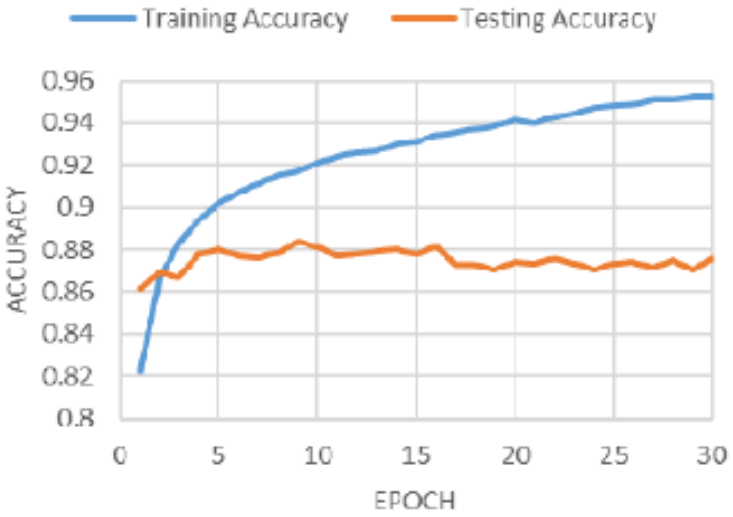

Figure 2.14: One of the Accuracy Graphs Generated [17]

2.4.5 A Deep Network for Pornographic Image Recognition Based on Feature Visualisation Analysis

In the same year (2018), Ying et al [19] developed a system capable of recognising images with explicit contents on MXNET. Their proposed solution included a CNN model fine-tuned in the conventional way using transfer learning [20] and improved using results of analysing the visualisation of features produced by the fine-tuned model.

The pre-trained models used were VGG-16 [18] and CaffeNet, with initial weights from ImageNet classification task. Weights of the last three fully connected layers were not loaded to allow the model to learn features specific to the targeted dataset which can increase the models' accuracies. Once training of the models completed, features extracted from each layer of the models were converted to images and displayed in order to analyse the said features and decide on the types of data augmentation to be applied in order to improve the results.

Two methods were used to visualise the features extracted from the layers of CNN, the first was to display the forward output directly, and the second was to de-convolute the feature before outputting it as an image. **Figure 2.15** shows a simple illustration of the deconvolution visualisation network's architecture. A sample result of outputs of feature visualisation using the second method is shown in **Figure 2.16**.

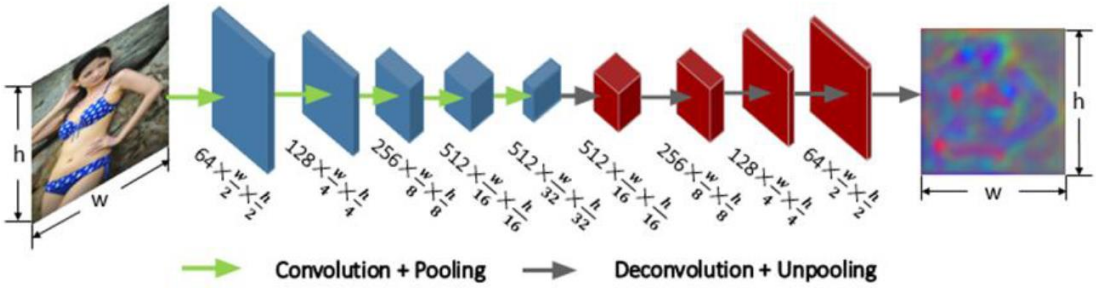

Figure 2.15: Architecture of the Feature Deconvolution Visualisation Network [19]

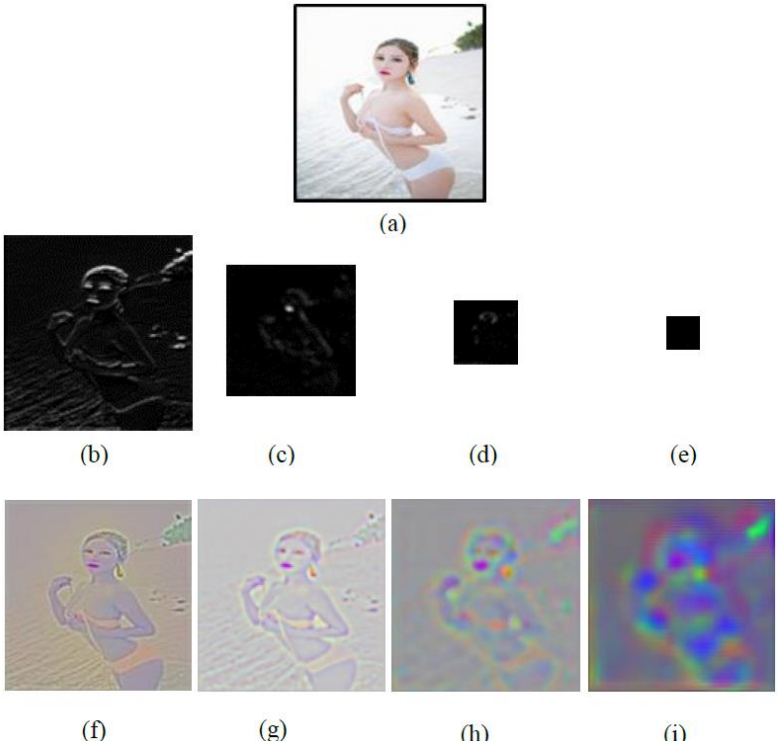

Figure 2.16: Feature Maps Produced Using Feature Deconvolution Visualisation Network (a) input image, (b-e) some forward visualisation feature maps from first to forth convolution layer, (f-i) backward visualisation feature maps of images b to e [19]

Overall, the highest accuracy achieved by this solution was 94.7%, an increase of 2.6% compared to the highest accuracy achieved without feature visualisation analysis performed on the selected CNN models (92.1% by VGG16 with the last two fully connected layers fine-tuned). A clear disadvantage of this approach is the human intervention required to decide the data augmentation method such that the accuracy will improve and not worsen.

It was highlighted in this paper that accuracy of the system relies on the distribution of data. For instance, if the distribution of adult images of females with short hair is high, the proposed solution may take “short hair” as an indicator for pornographic images.

2.4.6 Adult Content Detection in Videos with Convolutional and Recurrent Neural Networks

Wehrmann et al [21] designed a model named Adult Content Recognition with Deep Neural Networks (ACORDE) to tackle the same problem as defined in **Section 1.2**. The architecture of ACORDE is shown in **Figure 2.17**. Like most of the papers in **Section 2.4**, a pre-trained CNN model was utilised to extract features from key frames of videos. Multiple cropping technique was used during the feature extraction stage. Subsequently, a recurrent neural network (RNN) technique called long short-term memory (LSTM) was used for sequence learning to analyse video frames in their playing order. The final prediction of classes of videos was performed using majority voting scheme.

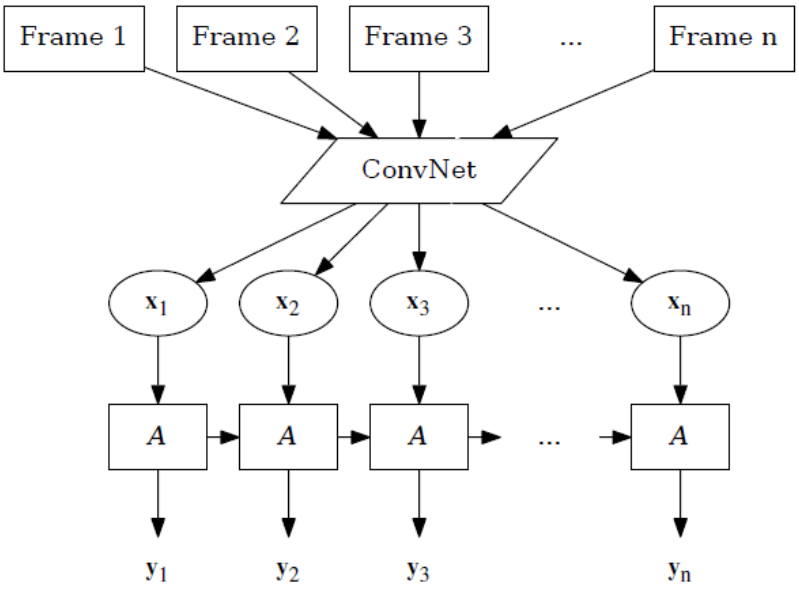

Figure 2.17: Architecture of ACORDE [21]

Different CNN architecture was used in each variant of ACORDE. The variants of ACORDE and the CNN architectures used are shown in **Table 2.2**.

Table 2.2: ACORDE Variants and Their CNN Architectures [21]

Variant	CNN Architecture
ACORDE-GN	GoogLeNet
ACORDE-50	ResNet-50
ACORDE-101	ResNet-101
ACORDE-152	ResNet-152

The highest video classification accuracy of this method was 95.6% by ACORDE-101 model, that used Residual Network-101 (ResNet-101) for the ConvNet block and 10 crops for feature extraction, when tested on the NPDI dataset. The pre-trained CNN model used in each variant of ACORDE were not trained using the pornographic dataset like how the LSTM were trained to learn the time-varying relationships among frames. The usage of LSTM made this system particularly useful in real-time applications.

From the findings, it was stated that videos with breastfeeding scenes had higher misclassification rate as women's breast were exposed. This is also one of the challenges faced in this project that should be addressed.

2.4.7 Video Pornography Detection through Deep Learning Techniques and Motion Information

Inspired by the effort of Simonyan and Ziesserman [22], Perez et al [23] proposed a system to detect pornography in video files using static as well as motion information in video files in the year 2017.

GoogLeNet [11], a CNN technique that was also the winner of ImageNet 2014 Challenge [24], was fine-tuned to the targeted problem using network weights learnt on the ImageNet images as initial weight values. The spatial features (static information) were extracted and classified using the selected CNN model. On the contrary, the temporal features (motion information) were extracted using two different techniques: optical flow displacement field method and MPEG motion vectors. The former computed the magnitude of gradient and the motion's direction while the latter used MPEG codec to measure the movement of macroblocks in video frames and represents the motion using motion vectors, which were used as feature vectors. Classification of motion information was, again, done by the CNN. An advantage of using the second method to extract motion information compared to the first is that motion vectors are available when the video is being decoded. An instance of displacement field generated from two sequential video frames, with each of the gradient magnitudes decomposed into horizontal and vertical components, is depicted in **Figure 2.18** while an example of motion vector of a macroblock is shown in **Figure 2.19**.

Figure 2.18: Optical Flow Displament Field Based Continuous Frames [23]

Figure 2.19: Motion Vector of a Macroblock [23]

Following the extraction of the static and motion data independently, fusion of the data and classification were done using several methods as can be seen in **Figure 2.20**: (a) Early fusion: the features extracted were combined and processed by another GoogLeNet, that was not fine-tuned, before being fed into a classifier, (b) Mid-level fusion: the features extracted were concatenated and then input into a classifier, and (c) Late fusion: the static and motion information underwent decision-making process separately and the average of the resulting classification scores was utilised for the classification. The final video classification was done based on majority voting.

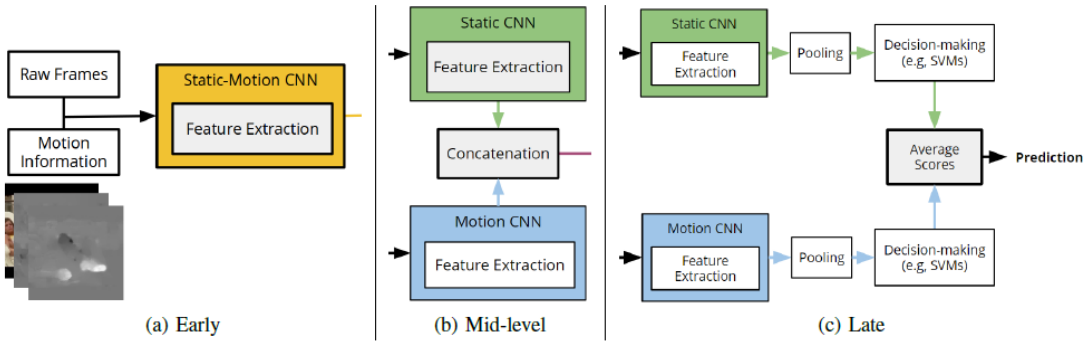

Figure 2.20: Fusion Stages [23]

This system was able to achieve accuracies of 96.4% using late fusion with any of the aforementioned motion information retrieval methods, and 97.9% using mid-level or late fusion with optical flow technique when tested with Pornography-2k [25] and Pornography-800 [26] datasets respectively.

Like [21], time-varying patterns were also utilised for the classification of videos, but the method used in this paper is inferior as this system will have a lower performance in real-time classification tasks.

2.5 Summary

Implementation of systems to detect inappropriate visual contents using digital image processing techniques mostly rely on assumptions, which restricts the context and affects the accuracy of the system. The same condition applied for the system that used SVM classifiers to achieve the same goal. Moreover, for the latter case, the means to extract features of the frames of videos were not fixed for different categories, as pointed out in **Section 2.3.1**. Hence, the frames need to be scrutinized before deciding the type of feature that is suitable to extracted and utilised to differentiate the frames or to categorise them according to the classes.

As there is a wide variety of visual contents that are considered as pornographic, where some involve exposure of naked body while some do not, some involve close-up shot of face and others do not, and the list goes on, decision for selection of feature of pornographic visual contents will be difficult to make. This is one of the reasons that pushed deep learning to gain its popularity.

Deep learning involves the training of system by allowing it to learn by examples without human intervention. Unlike the first two techniques, features to be extracted to allow classification of data are automatically decided and the extraction process is automatically performed if deep learning approach is used. As long as the example dataset is unbiased and covers a wide range of instances of the classes, results generated using the trained system will be satisfactory. A higher accuracy may be yielded through this approach which is easier to be implemented compared to the previous methods, thus the reason for the choice of using deep learning in this project.

All of the papers discussed in **Section 2.4** involved usage of CNN in their proposed systems, with most of them implementing transfer learning to adjust the

classification task to be suitable for the targeted dataset. Therefore, the effect of transfer learning on CNN is to be studied and analysed in this FYP. In addition to that, this project also aims to investigate the better CNN architecture among MobileNet, visual geometry group (VGG-19) and residual network (ResNet50_V2, and the better classifier between CNN and SVM.

It was observed that k-fold cross-validation method was employed in papers discussed in **Section 2.3.1** and **Section 2.4.4** that performed the classification on images. However, this method was decided not be employed in this experiment as it was deemed to be not practical. Continuous images from the same video may be distributed to different folds, affecting the accuracy of the system as similar images are present in both the training and validation folds. Instead, the distribution of images to the training and validation set was to be done manually such that images from the same video will be used for either training or validation but not both.

CHAPTER 3 DETAILS OF THE DESIGN

3.1 Introduction

In this project, CNN, a deep learning technique, is to be implemented to recognise inappropriate visual contents in videos. Several methods to include CNN in the proposed system will be experimented and the CNN model that gives the best results for each method and the method with the best performance are to be decided. The detailed implementation of the methods shall be described in this chapter.

The overall system architecture of the proposed system can be found in **Section 3.2**. Preparation of data to train and validate the results of proposed system is described in **Section 3.3**. The CNN architectures used will be discussed briefly in **Section 3.4**, and the techniques that will be included in the training and tuning of the methods to improve the networks' performances are described in **Section 3.5**. Furthermore, different methods to utilise CNN in the proposed system are presented in **Section 3.6**. The design of Graphical User Interface (GUI) will be presented and discussed in **Section 3.7**.

3.2 System Architecture

The overall system architecture of the proposed system is illustrated in **Figure 3.1**.

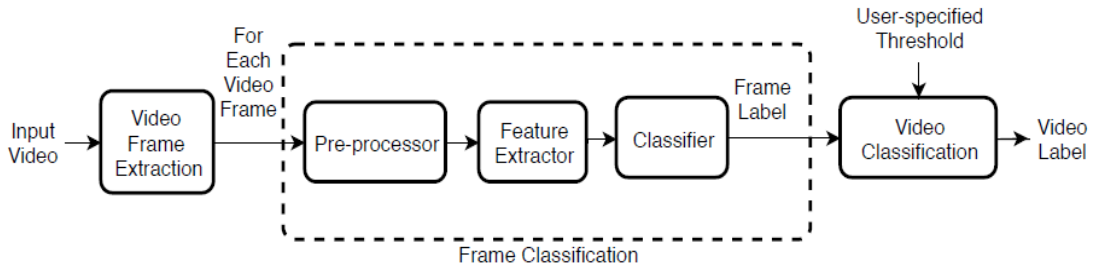

Figure 3.1: System Architecture of Proposed System

CNN is mostly used to process data with grid-like topology [3]. It is especially effective when applied in image classification tasks [27]. Hence, key frames from the input video files will be extracted to be used as inputs to the system. Each of the frames will be pre-processed such that each RGB pixel value will be normalised to a value in the range of 0 to 1 (by dividing the original pixel value of each channel by the maximum possible value of 255) or -1 to 1 (by subtracting each pixel value with the mean RGB pixel value). The resulting image will be fed into a feature extractor, which is essentially a CNN model, so that their contents can be "understood" by computers after travelling through the model. Subsequently, the classifier will place a label for each frame, indicating whether that particular frame consists of inappropriate visual content.

After all key frames from the same video file have gone through these processes and obtained their respective frame labels of either 0 ("Non Porn") or 1 ("Porn"), classification of the whole video file is to be performed by checking the ratio of "Porn" frames in the video with the threshold value specified by user to obtain the video's label. If the former is greater than or equal to the latter, then the video will be labeled as "Porn", otherwise it will be assigned the "Non Porn" label. For instance, for a video file that has 10 key frames extracted, if the user-specified threshold is 50.0% and 6 out of the total number of key frames have "Porn" labels (60.0%), then that "Porn" label will be assigned to that video.

The programming language chosen to develop the proposed system is Python [28], as it has many open-source libraries and a large community, and it allows a GUI to be created for the software. Majority of the source code related to CNN are to be written using Application Programming Interfaces (APIs) in the open-source **keras** library [29], with the plotting of graphs involving the **matplotlib** library [30]. **Sklearn** open-source library [31] will be imported to allow the implementation of SVM as classifiers and to allow classification report and confusion matrix to be generated. For the GUI, the **tkinter** library [32] will be imported. The **cv2** library [33] will be imported for the sake of extracting frames from the videos. All the source codes will be written using Spyder 3.3.6 [34] with Python 3.7.4 [28] in Anaconda 3 integrated development environment (IDE) [35].

3.3 Data Preparation

A total of 804 videos (402 are "Non Porn" and 402 are "Porn") and 727 images obtained from the Internet (all are "Porn"), with various settings (anime and real world), angles, background, human skin colour and zoom levels, will be used for the training and validation of the proposed system. The amount of images and video files for each category is summarised in **Table 3.1**.

Table 3.1: Amount of Data for Each Category

	Porn	**Non Porn**	**Total**
Video	402	402	804
Video Frames	5,519	14,258	19,777
Images Retrieved from Internet	727	-	727
Total Images	6,246	14,258	20,504

Since input to the classification part of the system should be an image, key frames from the videos are to be extracted before they are fed into the system. For the training of CNN involved in the proposed system, supervised learning is to be executed. Therefore, labels of 0 ("Non Porn") or 1 ("Porn") will be assigned to each of the images and video frames belonging to the training dataset before the training process took place [36]. All images to be input into the proposed system are in RGB colour model as colour is one of the features that can help in the current classification problem.

The data are to be split into two groups with similar distribution of probability: training dataset and validation dataset. Training dataset, consisting of a collection of

examples, will be fitted into the CNN model for it to learn the context of images [37]. On the other hand, validation dataset, or development set, will be used to adjust the weights of the CNN model and to check the performance of the model [38].

Around 80% of the data of each class (11,872 images for "Non Porn" and 4,531 images for "Porn") are allocated for training purpose and the rest (2,386 images for "Non Porn" and 988 images for "Porn") are reserved for validation use. As mentioned in **Section 2.5**, the splitting of data will be done manually.

3.4 CNN Architectures

CNN is a neural network that includes the mathematical process of convolution in two or more of its layers [3]. The convolution operation is like sliding of a filter or kernel over the input, with the adjustable filter size, stride and padding for filter boundary values. **Figure 3.2(a)** below shows an example of convolution operation with 7x7 input with stride of 1, generating output with dimension of 5x5 while **Figure 3.2(b)** shows another instance with the same input but with stride of 2, generating output with dimension of 3x3. In both cases, a 3x3 filter was used.

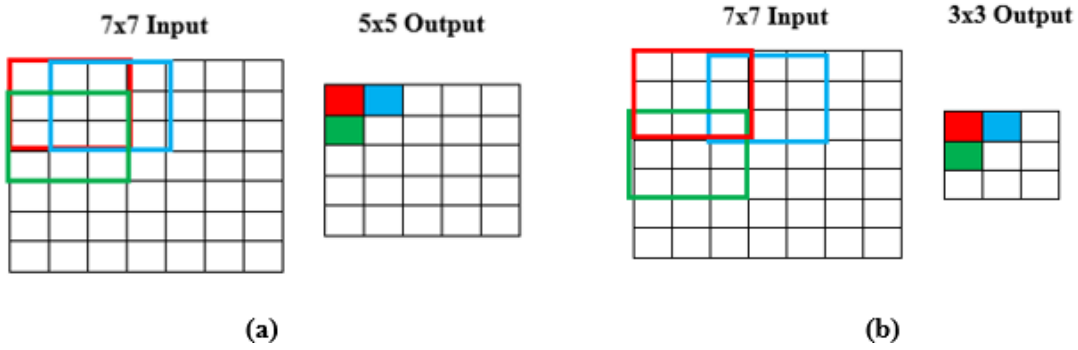

Figure 3.2: Convolution Operation

Different CNN architectures will be used for the "Feature Extractor" and "Classifier" blocks in **Figure 3.1**, depending on the method of CNN implementation that will be described in **Section 3.5**. Each of the architectures selected to be implemented in this system will be discussed briefly in this section.

3.4.1 **MobileNet** [39]

Designed for applications for mobile and embedded system, MobileNet is a lightweight CNN with 30 layers, achieved by utilisation of depthwise separable convolutions. In this model, single convolution is performed on each colour channel instead of on the flattened three colour channel values. Images of dimension 224x224 with 3 colour channels are taken in as inputs. The architecture of this CNN is shown in **Figure 3.3**.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 3.3: MobileNet Architecture [39]

3.4.2 **Visual Geometry Group-19 (VGG-19)** [18]

VGG-19 is a CNN architecture with 19 weight layers, as depicted in **Figure 3.4**, which takes in input RGB image of size 224x224. Inputs to this CNN architecture are pre-processed by performing subtraction of each pixel value with the average RGB value calculated using the training dataset.

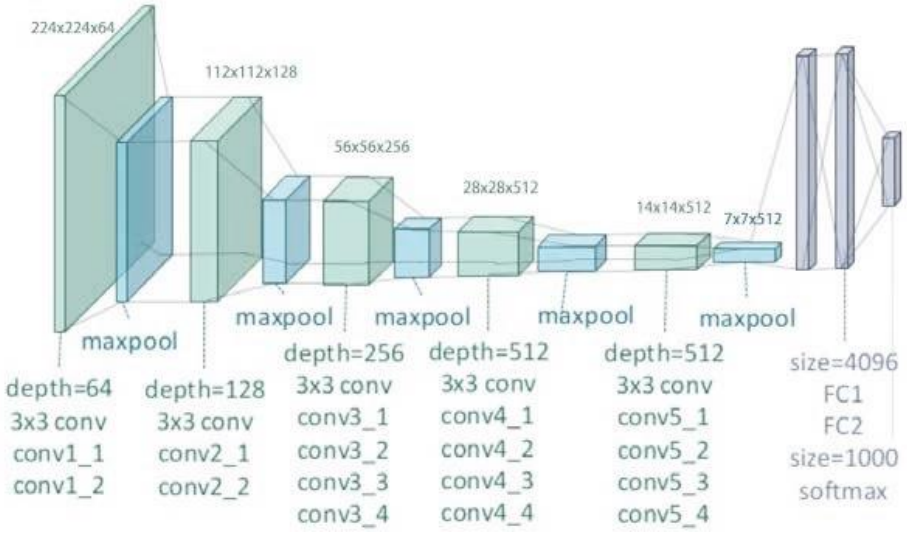

Figure 3.4: VGG-19 Architecture [40]

3.4.3 Residual Network-50 Version 2 (ResNet50_V2) [41]

Evolving from ResNet50, ResNet50_V2 involves "skip connections" and batch normalisation or after-addition activation in its 50 layers, suggesting that propagation of forward and backward signals can occur directly from one block to another block[41]. Besides the change in the Residual Unit as illustrated in **Figure 3.5**, architecture of the CNN is similar to that of its predecessor.

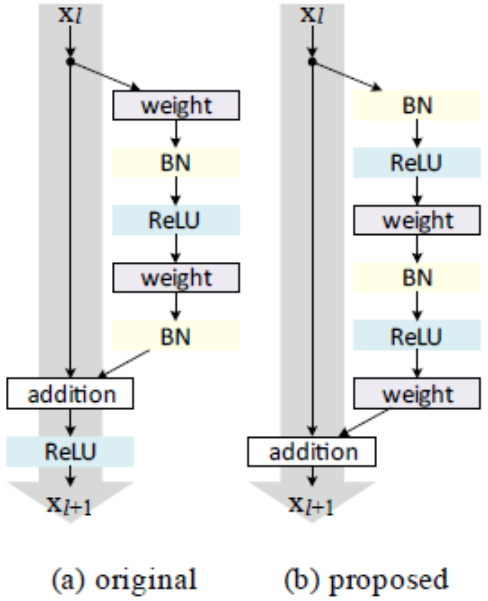

Figure 3.5: Original and Proposed Residual Units [41]

3.5 Techniques Used for CNN Implementation

It is undeniable that selection of a suitable CNN architecture is important but training of the CNN architecture such that it is dedicated towards the domain of interest is equally important. Tweaking of connection weights or variables of the CNN will be performed using the different techniques discussed later in this section in order to improve the network performance and its ability to generalise.

3.5.1 Transfer Learning

Generally, transfer learning allows learning of a new task (target task) based on previously learnt tasks (source task) through knowledge transfer [42]. It eases and accelerates training of CNN as the CNN does not need to be trained from scratch and less data is required for training. An illustration of learning processes of traditional learning and transfer learning are shown in **Figure 3.6**.

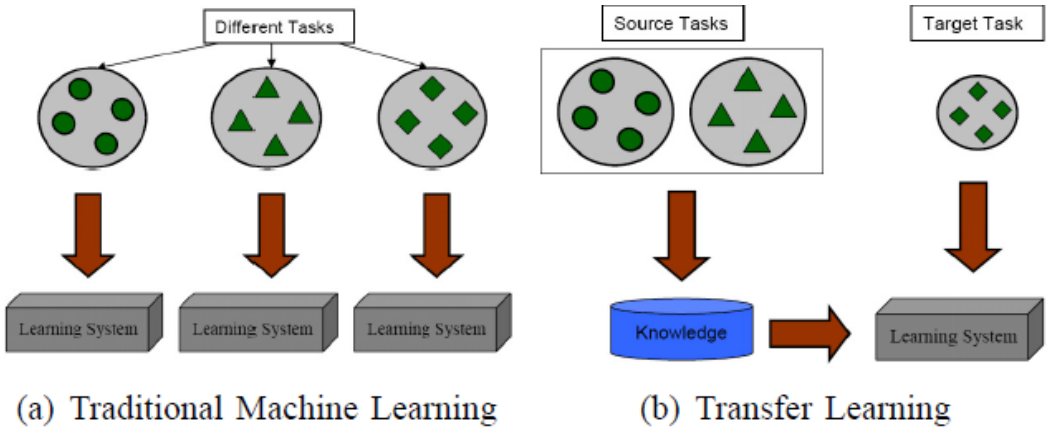

Figure 3.6: Learning Processes of Traditional Learning and Transfer Learning [43]

Pre-trained CNN architectures with weights initialised to values obtained from classification task on the large ImageNet dataset [10] are used as the starting point for all the methods employed in this project. Projection of lower dimension or coarse features like colours, edges and basic shapes in the early or bottom layers of CNN architectures are similar for most, if not all, CNN applications. Hence, the weights in the bottom layers of convolutional base do not need to be trainable via backpropagation and can be fixed [44]. This would guarantee a higher starting

accuracy and more predictable behavior of the CNN instead of having random initial weights, as depicted in **Figure 3.7**. However, the same cannot be said for the more abstract or more complex features such as human eye, mouth and nose that are detected in the top convolutional layers of CNN. Different objects or features in higher dimension are to be recognised for different problems or dataset to allow the CNN to be fine-tuned to the targeted dataset [20]. Therefore, the top layers of CNN should be unfreezed to allow their connection weights to be trainable. A graphical representation of application of pre-trained CNN sharing the same convolutional base is shown in **Figure 3.8**.

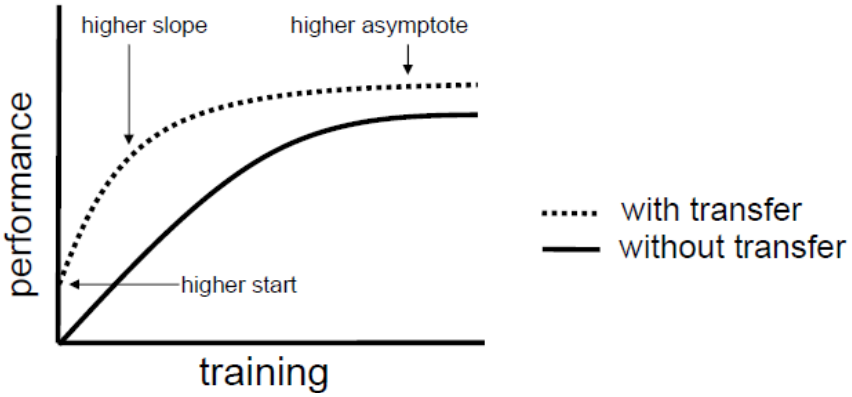

Figure 3.7: Performance of Transfer Learning vs Traditional Learning [42]

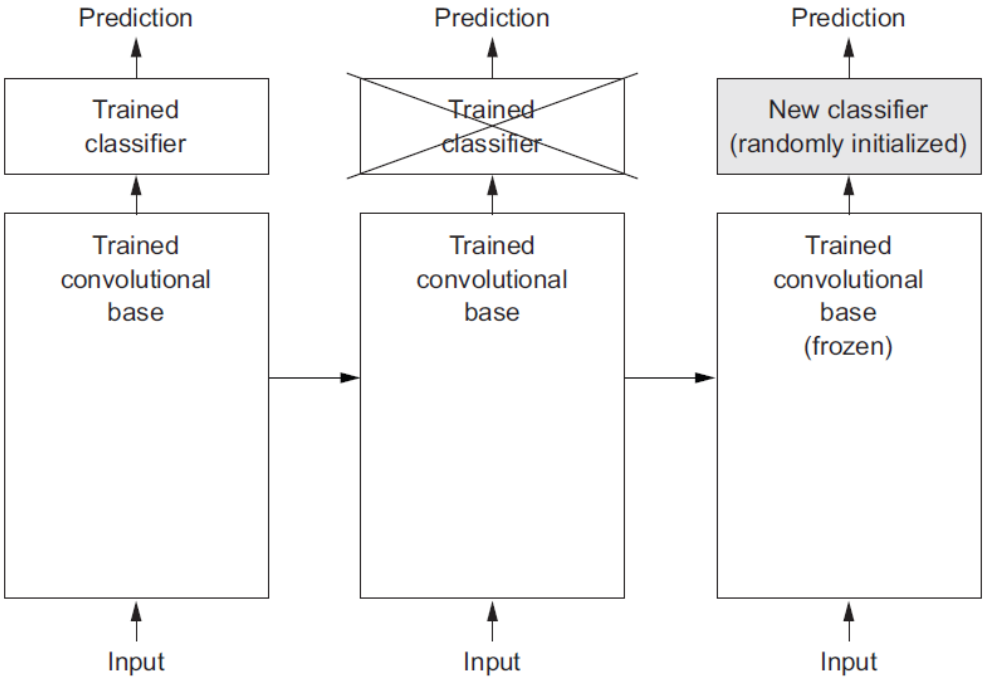

Figure 3.8: Application of Pre-trained CNN Model [44]

For cases where the fine-tuned CNN acts as the classifier in addition to being the feature extractor, the fully-connected layers at the top of the pre-trained CNN are to be replaced with new fully-connected layers so that the desired classification can take place. The final classifier, which is also the last fully-connected layer of the CNN, would be using a "softmax" activation function with *n* neurons, where *n* is the number of classes or possible outputs, to convert all the components to a value between 0 and 1 such that sum of all the components would be 1 [3].

3.5.2 Non-linear ReLU Activation

The activation function to be applied in the CNN is the non-saturating and non-linear Rectified Linear Unit (ReLU), which follows **Equation 3.1** [45]. Basically, output will be zero if input is less than zero, otherwise output will simply be the input value. This function allows training of CNN to be performed faster than the equivalent saturating neurons [10].

$$f(x) = \max(0, x) \tag{3.1}$$

where x = input,

f(x) = output

3.5.3 Data Augmentation

An effective method to reduce overfitting on the input image data and allow better generalisation is to add more data artificially via data augmentation [42]. Basically, when each image data is input to the CNN for training process, it undergoes random translation, transformation, rotation, shearing, zooming, horizontal and vertical flipping, and shifting of brightness value with limits specified in the source code, so that the trained model can recognise modified versions of the original inputs, hence increasing its robustness. **Figure 3.9** illustrates some of the randomly generated augmented data.

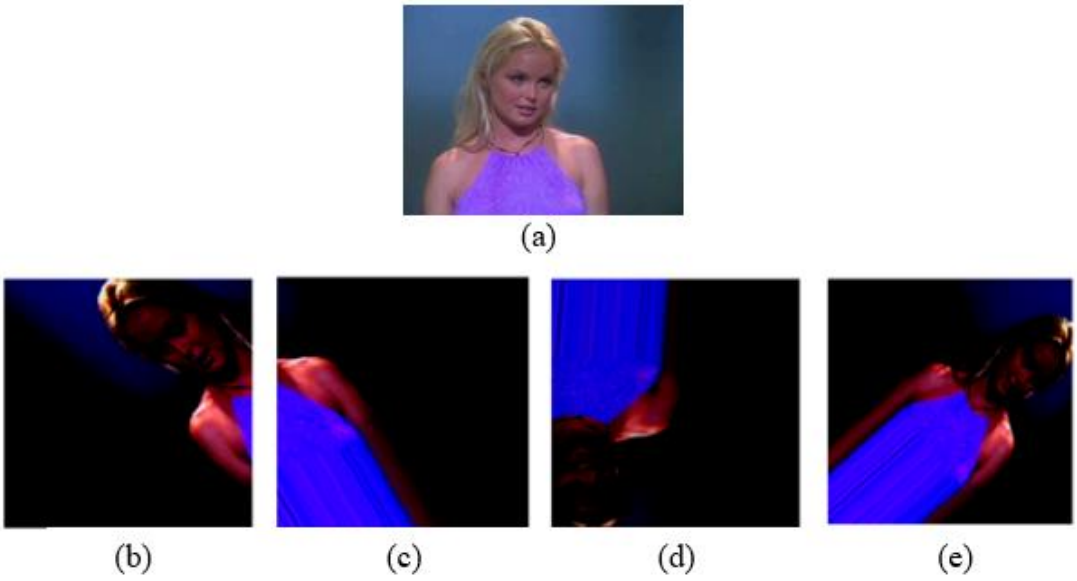

Figure 3.9: Augmented Data

(a) original image, (b-e) augmented versions of (a)

In the proposed system, data augmentation will be performed in real-time on the next batch of input images while the current batch of augmented data is used for training by the graphics processing unit (GPU), hence removing the need for additional storage space to store the augmented data [10]. Note that data augmentation will not be applied on the validation dataset.

3.5.4 Dropout Layer

Another popular mean to minimize overfitting is by adding dropout layers after the fully-connected layers that form part of the classifier. Dropout layer reduces the complex co-adaptations of training data [46] by removing neurons from the network so that they will not affect the forward pass and backward propagation processes of the CNN [10]. Note that dropout layers do not take effect in validation process of the proposed system [47]. A comparison between neural networks with and without applying dropout is shown in **Figure 3.10**.

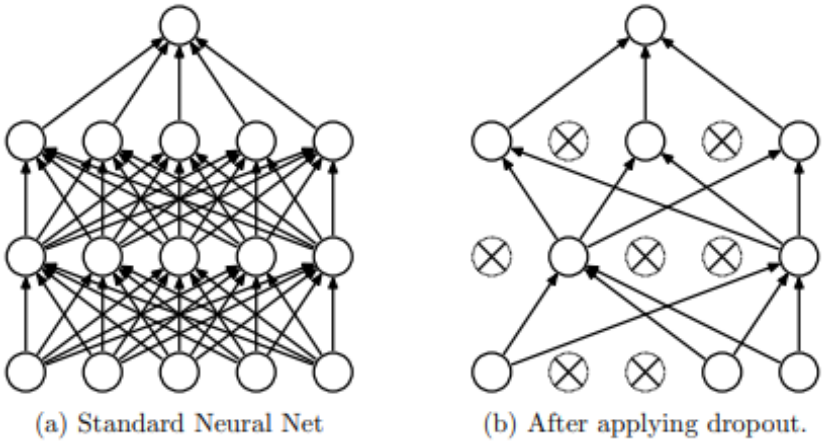

Figure 3.10: Dropout Neural Network Model [47]

3.5.5 Reduction of Network Capacity

The act of limiting the capacity of network reduces the number of trainable parameters in the network. With limited resources, the network will not be able to map to the training dataset as easily, thus reducing overfitting [44]. This can be accomplished by having a small number of neurons to be present in the classifier fully-connected layers. **Figure 3.11** shows that the curve for the smaller model rises later than the original model. This indicates that the smaller model overfits later than the original model, proving the fact that a network with larger capacity tends to overfit more easily than a smaller network although it is more capable of modelling the training dataset.

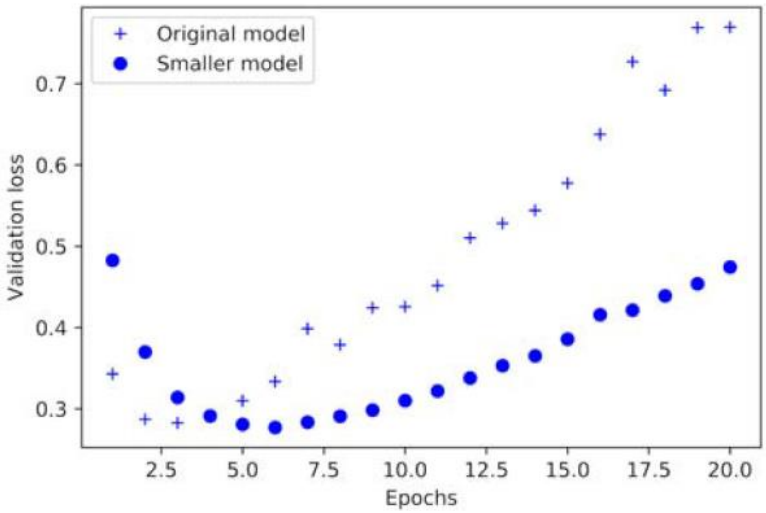

Figure 3.11: Effect of Model Capacity on Validation Loss [44]

3.5.6 Kernel Regulariser

This regularisation technique works in a way such that a cost is added to the CNN loss function. As a result, complexity of network can be constrained as the weights are forced to have only small values, causing weight values to distribute more evenly or regularly.

There are two types of cost associated with this regularisation technique. For L1 regularisation, the cost to be added is proportional to the weight coefficients' absolute value [44]. On the other hand, for L2 regularisation or weight decay, the cost to be added is proportional to the weight coefficients' squared value [48].

3.6 Methods of CNN Implementation

The ways of implementing CNN in this project and the details of training the CNN will be explained in the following parts. For each of the methods, some of the techniques explained in **Section 3.5** will be applied to improve the CNN's performance. A summary of the methods to be implemented in the proposed system is shown in **Table 3.2**.

Table 3.2: Summary of Methods to Implement the Proposed System

Method	Feature Extractor	Classifier
1	Traditional CNN	CNN
2	Fine-tuned CNN	CNN
3	Traditional CNN (from Method 1)	SVM
4	Fine-tuned CNN (from Method 2)	SVM

For methods that use CNN as classifier, two fully-connected layers with ReLU activation function and kernel regularisation function, and another fully-connected layer with softmax activation function will be added to the top of the imported network to act as the classifier. After each of the fully-connected layer with ReLU activation function is added, a dropout layer is added to remove effects of some of the neurons in the CNN so as to simplify the network. Note that the final softmax layer will have two neurons, outputting probabilities for each of the possible output of the proposed system. **Figure 3.12** shows a flowchart for the design of CNN classifier. Most of the CNN hyperparameters (learning rate, number of epochs, batch size et cetera) will be fixed, leaving only certain types of values that can be manipulated, such as number of neurons available in the added fully-connected layers in the classifier layers and the number of layers to be fine-tuned when transfer learning is applied.

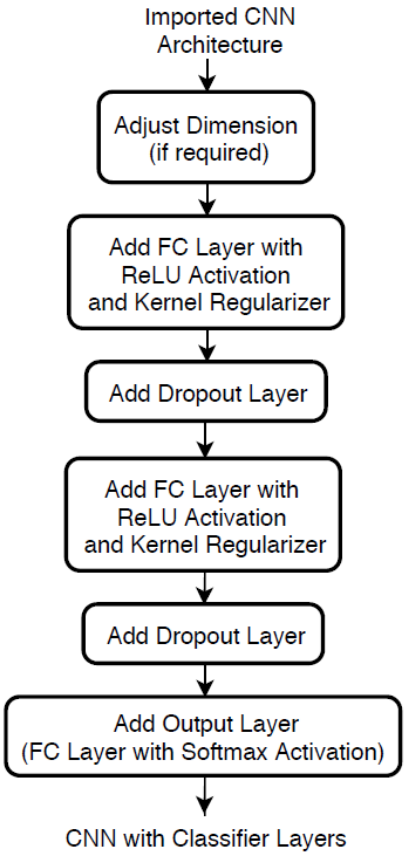

Figure 3.12: Flowchart of Design of CNN Classifier

Another type of classifier to be implemented in this project is SVM [49]. SVM tries to find the optimal decision boundary or hyperplane between points belonging to two different classes in order to solve a classification problem [44], as shown in **Figure 3.13**.

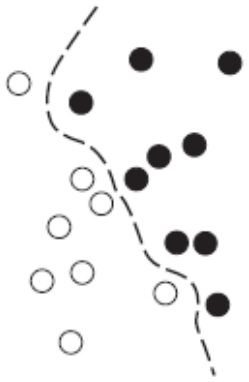

Figure 3.13: Decision Boundary Formed by SVM

The flowchart of training and validation of classification processes is illustrated in **Figure 3.14**. Since softmax activation function is used in the last layer of CNN classifier and a classification problem is to be solved, loss function involved in the

compilation of CNN model is categorical cross entropy loss (or softmax loss). In classification problem, accuracy is the only metric to be monitored. RMSprop optimiser will be employed in the all the methods.

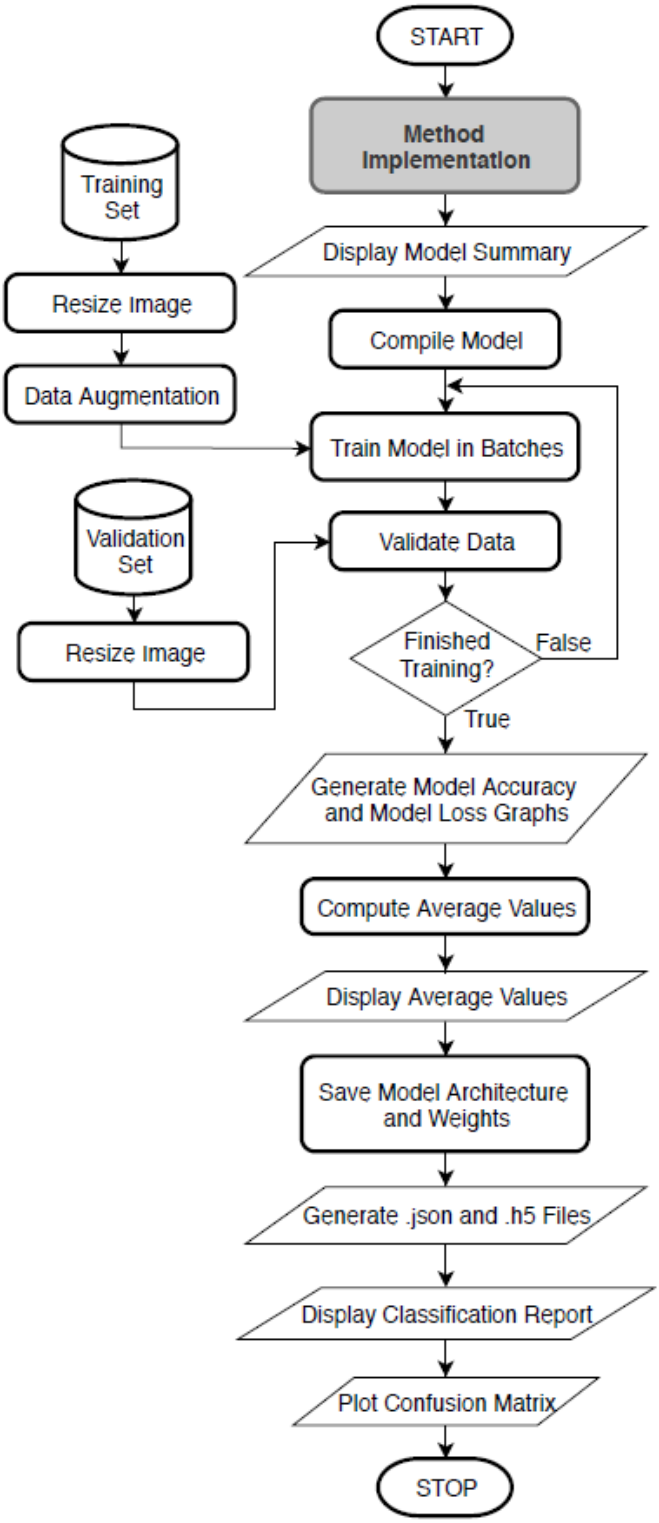

Figure 3.14: Flowchart of Training and Validation of Classification Processes

3.6.1 Method 1: Traditional CNN as Feature Extractor and Classifier

This method plays the role of a control experiment, setting the baseline for CNN performance in detection of pornographic visual contents. It is expected to have the worst performance among all the methods to be implemented.

A CNN architecture (up to the convolutional layers) acting as the feature extractor will be imported with random initial weights. Then, the procedures in **Figure 3.12** are carried out to enable the CNN to have classifier layers so that it can perform classification of the inputs. All the convolutional layers in the selected CNN (base model) will be trained to adjust the connection weights according to the targeted dataset. Three combinations of numbers of neurons will be experimented: (64, 8), (96, 24) and (128, 36), where the numbers enclosed in the parentheses denotes the number of neurons present in the first and second added fully-connected layer of the classifier respectively.

3.6.2 Method 2: Fine-tuned CNN as Feature Extractor and Classifier

The effect of transfer learning will be examined through the implementation of this method. Like Method 1, a CNN architecture (up to the convolutional base) acting as the feature extractor will be imported. However, unlike Method 1, it will be imported with initial weights from ImageNet classification task so that transfer learning is performed. Again, procedures in **Figure 3.12** will be performed to enable the pre-trained CNN to act as the classifier aside from feature extractor. For each CNN architecture, the combination of number of neurons in the classifier which gave the best result (from Method 1) will be used. This time, only the number of layers with trainable weights are manipulated.

3.6.3 Method 3: Traditional CNN as Feature Extractor and SVM as Classifier

This method is implemented to study the effect of using SVM classifier coupled with a basic CNN (without application of transfer learning) as feature extractor to perform

pornographic image classification. Feature extraction will be performed by a traditional CNN, which is a CNN architecture (up to the convolutional layers) imported with random initial weights, similar to that of Method 1. In fact, the trained traditional CNN that yielded the best result from Method 1 is to be used as the feature extractor for each of the CNN architectures. The classification of inputs, however, is to be done using SVM.

3.6.4 Method 4: Fine-tuned CNN as Feature Extractor and SVM as Classifier

This method resembles a mixture of Method 2 and Method 3 as both transfer learning and SVM classifier are employed to classify images with adult contents. For each CNN architecture, the fine-tuned CNN with the best outcome obtained from Method 2 will be used for feature projection while the classifier will be an SVM.

3.7 Design of GUI

In order to allow user interaction and increase flexibility of the system, a GUI illustrated in **Figure 3.15** is designed for the system. Since the algorithm or flow of program is to be implemented using Python programming language [28], the **tkinter** library [32] will be utilised to allow the GUI to be manifested. The CNN architecture and the combination of manipulated variable which gives the best result for each method will be used in the GUI to be implemented.

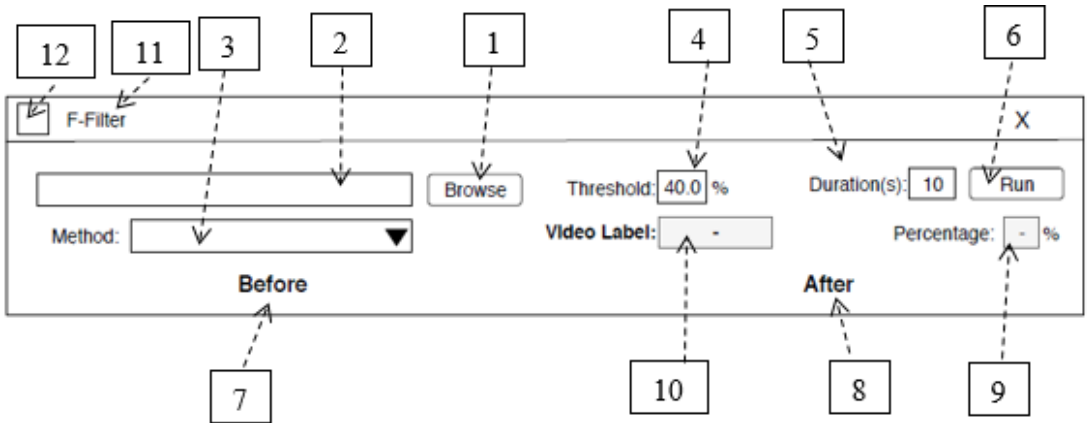

Figure 3.15: GUI Design

Details or functions of the numbered items in **Figure 3.15** are explained below:

1. Allows user to browse the video file to be processed using a file dialog.
2. Shows directory path of selected video file to be processed. It is updated automatically each time a file is selected using (1).
3. Allows selection of method which determines the feature extractor and classifier used to detect inappropriate visual content present in the selected video file. The default method will be the method with the best performance.
4. Allows user to set percentage of key frames with inappropriate content allowed to be present in the selected video file. The default value is 40.0%. Value entered into this field must be between 0 to 100, exclusive of the boundary values as the former filters all key frames irrespective of whether they contain pornographic content while the latter will show uncensored contents in almost all cases.
5. Allows user to specify the duration or time interval (in seconds) for extraction of consecutive key frames. The default value is 10 seconds.
6. Starts the detection. It should be enabled after specifying the details required for the detection process.
7. The work area for the original key frames extracted from the video file to be displayed after the detection process has completed. The original key frames will be shown below the “Before” label.
8. The work area for the processed key frames to be displayed. Key frames classified as “Non Porn” will simply be displayed while those classified as “Porn” will be blackened and displayed as a black frame. Note that the original video file will not be modified, only the frames displayed in the GUI will be changed. Like (8), The processed key frames will be shown below the “After” label.
9. Displays the percentage of “Porn” key frames in the selected video file after every successful run.
10. Displays the label assigned to the selected video file after every successful run. As mentioned in **Section 3.2**, labelling of video depends on values in (4) and (9).

11. Name of the program. “F-Filter” stands for frame filtering.
12. Icon of the program.

Note that the design of this GUI is of the alpha version. Once the algorithm and program is deemed to be matured, the GUI design may be changed.

CHAPTER 4 DATA PRESENTATION AND DISCUSSION OF FINDINGS

4.1 Introduction

The training of CNN models was performed from Method 1 to Method 4 in ascending order. Results obtained from implementation of the system are presented in **Section 4.2**. Discussion of the results will be placed in **Section 4.3**. Moreover, screenshots of the GUI of the implemented software will be included in **Section 4.4**.

4.2 Data Presentation

Due to the large hyperparameter space, testing of each hyperparameter value is computationally expensive [50]. Therefore, the hyperparameters used were decided prior to conducting this project without applying any hyperparameter optimisation method [51].

A low learning rate of 0.0001 was used for all the CNN training so that the degree of change can be limited. Batch sizes, or number of samples per batch, of 32 and 16 were used on the training and validation dataset respectively to save memory space. For all the CNN training involved, only 20 epochs, or iterations over the whole training dataset, were run to reveal the initial trends. The initial trends are sufficient to show whether transfer learning allowed CNN to perform better in the starting stage of training so as to reduce the training time required. Besides that, since performance of CNN was bound to saturate at a certain point, inference can be made from the steepness or trend of the curves. The only optimiser used was RMSprop, which was sufficient for most problems, according to [44]. Dropout rates were fixed at 0.5 or 50%, which means 50% of the inputs of the particular layer were excluded from each update cycle. Only L1 regulariser was used as the kernel regularizer such that 0.005 times of each weight coefficient value were be added to the total network loss [44]. ReLU was the activation function used in all layers except the output layer, which uses softmax function to allow the classification to be done.

Confusion matrix is a visualisation that includes the breakdown of values for accuracy calculation. The general equation for calculation of accuracy is shown in **Equation 4.1**. In all the implemented methods, confusion matrices were generated to aid user in understanding the classification errors made. The interpretation of confusion matrix is showed in **Table 4.1**. As this algorithm is designed to detect pornographic images, detection of adult content is perceived as positive result. On the contrary, negative outcome means no inappropriate visual content is detected. The terms "true" and "false" mean correct and incorrect classifications respectively.

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

where TP = True Positive,

TN = True Negative,

FP = False Positive,

FN = False Negative

Table 4.1: Interpretation of Confusion Matrix

True Negative (TN): "Non Porn" image labelled correctly	**False Positive (FP)**: "Non Porn" image labelled as "Porn"
False Negative (FN): "Porn" image labelled as "Non Porn"	**True Positive (TP)**: "Porn" image labelled correctly

Precision, recall and F1 score are performance metrics suitable to be employed for CNN models. The classification report API of **sklearn** library will be utilised to compute these aforementioned metrics. Precision is the percentage of relevant results used in the experiment out of all the results irrespective of whether they are categorised correctly. It can be generalised using **Equation 4.2**. On the other hand, recall is the percentage of relevant results classified correctly using a specified

algorithm. The general equation for recall is shown in **Equation 4.3** [52]. F1 score is the weighted average of both the precision and recall. When this score approaches 1, the model is perceived as having a good performance, and the opposite if it approaches 0. The weights for precision and recall are the same in the calculation, as shown in **Equation 4.4** [31]. Besides these three performance metrics, another parameter that will be presented in the classification report generated by **sklearn** [31] is support, which indicates the number of data used for each class.

$$\text{Precision (Porn)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Precision (Non Porn)} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{4.2}$$

where TP = True Positive,
TN = True Negative,
FP = False Positive,
FN = False Negative

$$\text{Recall (Porn)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$\text{Recall (Non Porn)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.3}$$

where TP = True Positive,
TN = True Negative,
FP = False Positive,
FN = False Negative

$$\text{F1} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

where Precision = Percentage of relevant results irrespective whether they are classified correctly
Recall = Percentage of relevant results that are classified correctly

4.2.1 Method 1: Traditional CNN as Feature Extractor and Classifier

Table 4.2 shows the results obtained from implementation of Method 1 using the CNN architectures mentioned in **Section 3.4**, where the last column is Validation Accuracy denoted as Val Accuracy. **Figure 4.1** to **Figure 4.18** show the model accuracy and loss graphs, confusion matrices and classification reports generated for each neuron number combination of each chosen CNN architecture using **sklearn** library [31].

With reference to **Table 4.1**, the number of TP, TN, FP and FN in the results of each CNN configuration can be identified from the respective confusion matrices. For instance, in **Figure 4.2**, the number of TP is 812, number of TN is 2103, number of FP is 283 and number of FN is 176 in the implementation of Method 1 with MobileNet with 64 and 8 neurons in the added fully-connected layers as the CNN model. From the same figure, information of precision (92% for "Non Porn" and 74% for "Porn"), recall rate (88% for "Non Porn" and "82%" for "Porn"), F1 score (90% for "Non Porn" and 78% for "Porn") and support (2386 for "Non Porn" and 988 for "Porn") for each class can be extracted from the classification report on the left.

Table 4.2: Results Obtained from Implementation of Method 1

CNN Model	Number of Trainable Parameter	Number of Neuron	Elapsed Time (hh:mm: ss)	Model Accuracy and Loss Graphs / Classification Report and Confusion Matrix	Val Accuracy (%)
MobileNet	3,273,114	(64, 8)	02:00:24	**Figure 4.1** / **Figure 4.2**	86.40
	3,307,754	(96, 24)	02:01:14	**Figure 4.3 / Figure 4.4**	**86.93**
	3,342,894	(128, 36)	02:02:53	**Figure 4.5** / **Figure 4.6**	86.43
VGG-19	21,630,618	(64, 8)	02:01:22	**Figure 4.7 / Figure 4.8**	**84.65**
	22,435,306	(96, 24)	02:02:08	**Figure 4.9** / **Figure 4.10**	82.13
	23,240,494	(128, 36)	02:03:07	**Figure 4.11** / **Figure 4.12**	83.91
ResNet50_ V2	29,942,490	(64, 8)	02:06:42	**Figure 4.13** / **Figure 4.14**	84.88
	33,155,626	(96, 24)	02:07:36	**Figure 4.15** / **Figure 4.16**	86.54
	36,369,262	(128, 36)	02:08:02	**Figure 4.17** / **Figure 4.18**	**86.72**

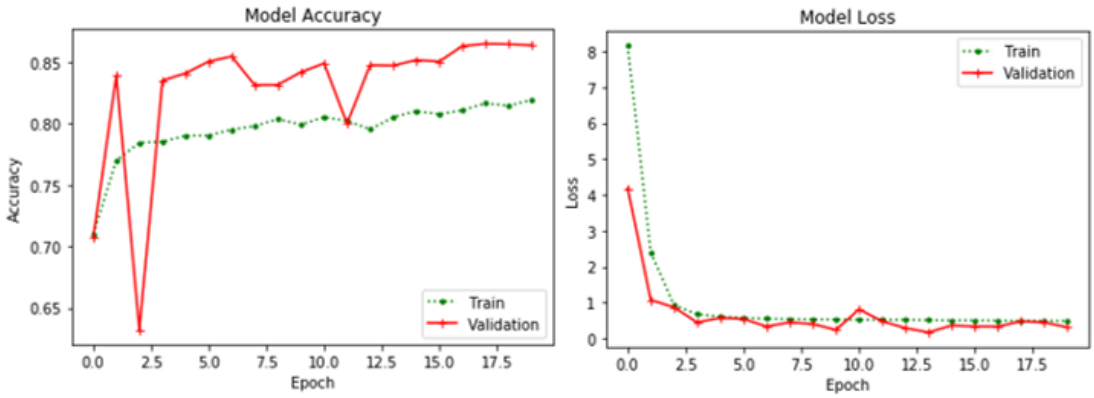

Figure 4.1: MobileNet, (64, 8) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.92	0.88	0.90	2386
vPorn	0.74	0.82	0.78	988
accuracy			0.86	3374
macro avg	0.83	0.85	0.84	3374
weighted avg	0.87	0.86	0.87	3374

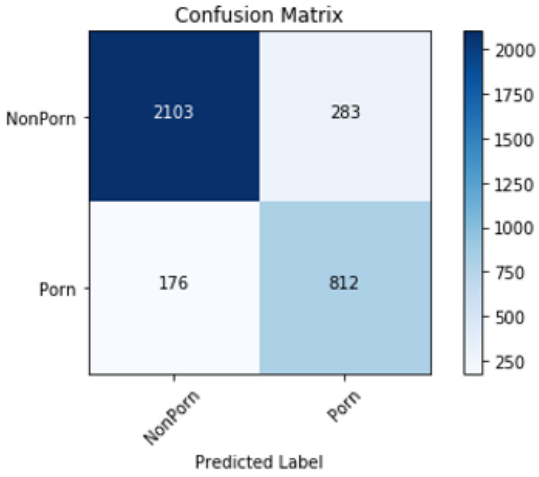

Figure 4.2: MobileNet (64, 8) Classification Report and Confusion Matrix [Method 1]

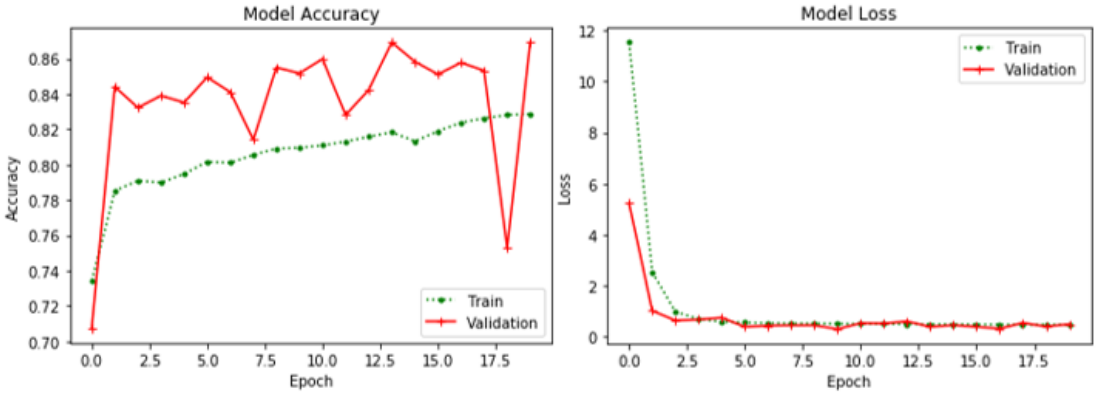

Figure 4.3: MobileNet, (96, 24) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.92	0.89	0.91	2386
vPorn	0.76	0.81	0.78	988
accuracy			0.87	3374
macro avg	0.84	0.85	0.84	3374
weighted avg	0.87	0.87	0.87	3374

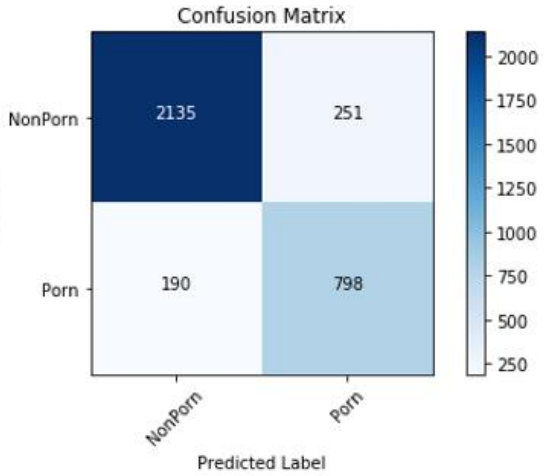

Figure 4.4: MobileNet (96, 24) Classification Report and Confusion Matrix [Method 1]

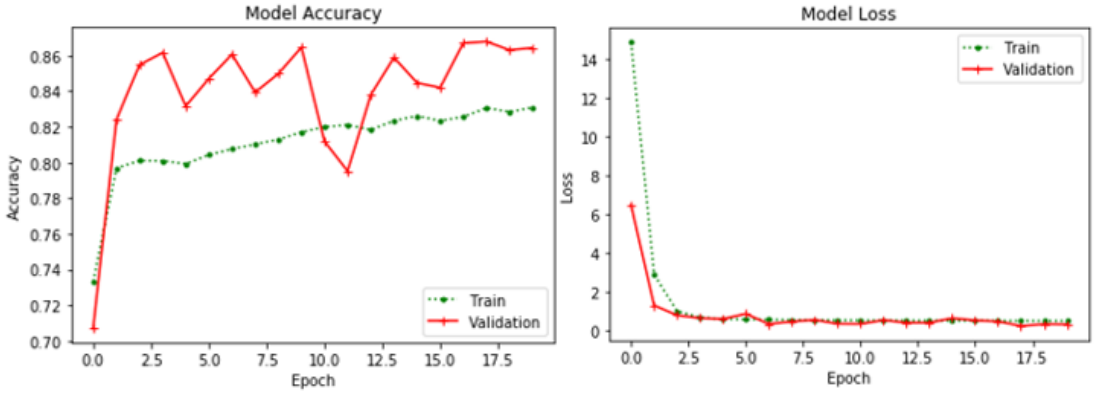

Figure 4.5: MobileNet, (128, 36) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.89	0.92	0.91	2386
vPorn	0.79	0.73	0.76	988
accuracy			0.86	3374
macro avg	0.84	0.83	0.83	3374
weighted avg	0.86	0.86	0.86	3374

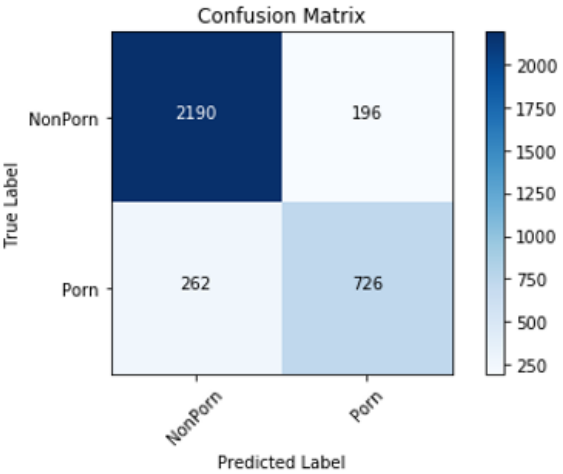

Figure 4.6: MobileNet (128, 36) Classification Report and Confusion Matrix [Method 1]

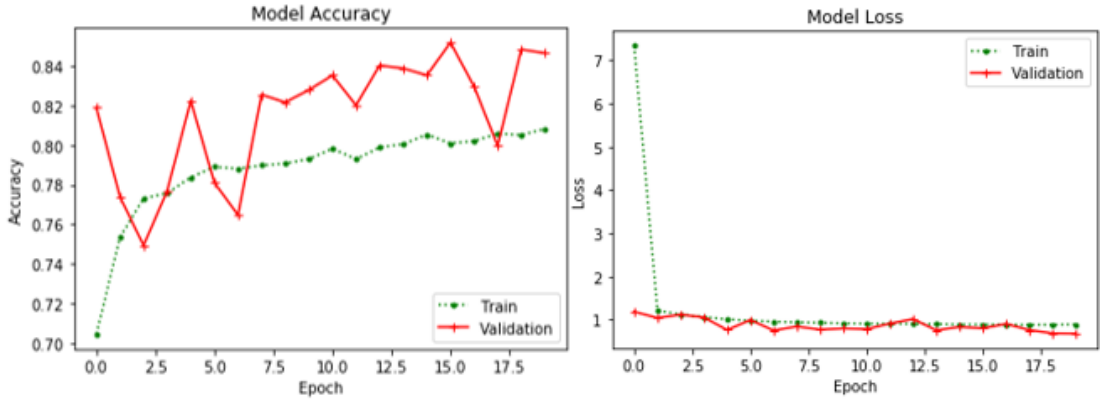

Figure 4.7: VGG-19, (64, 8) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.87	0.92	0.89	2386
vPorn	0.77	0.67	0.72	988
accuracy			0.85	3374
macro avg	0.82	0.80	0.81	3374
weighted avg	0.84	0.85	0.84	3374

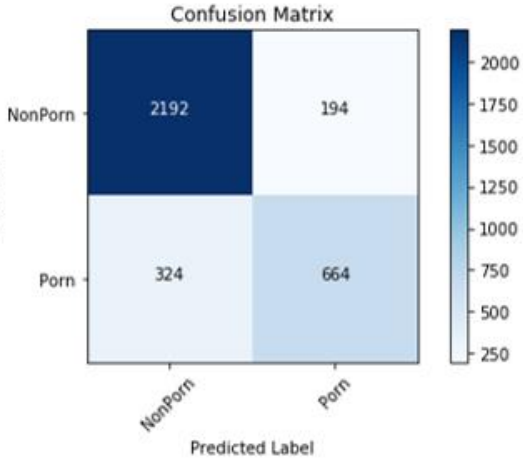

Figure 4.8: VGG-19 (64, 8) Classification Report and Confusion Matrix [Method 1]

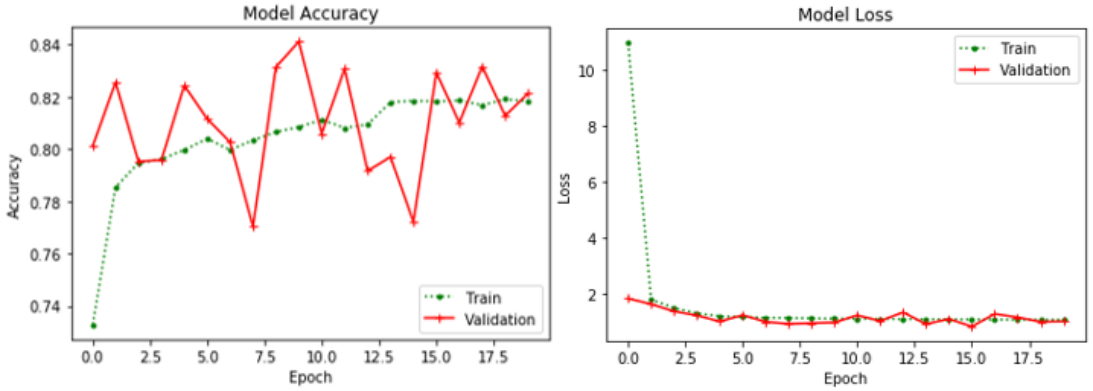

Figure 4.9: VGG-19, (96, 24) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.92	0.82	0.87	2386
vPorn	0.65	0.83	0.73	988
accuracy			0.82	3374
macro avg	0.79	0.82	0.80	3374
weighted avg	0.84	0.82	0.83	3374

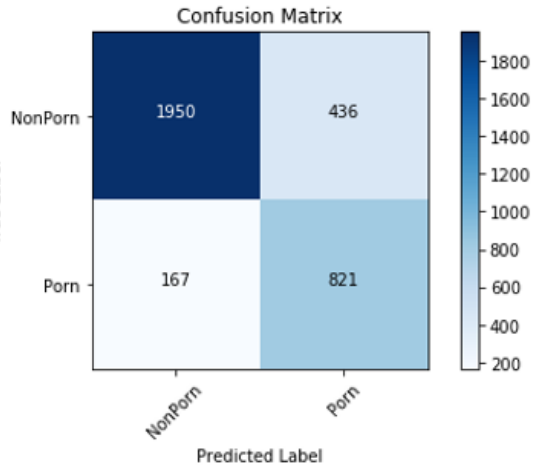

Figure 4.10: VGG-19 (96, 24) Classification Report and Confusion Matrix [Method 1]

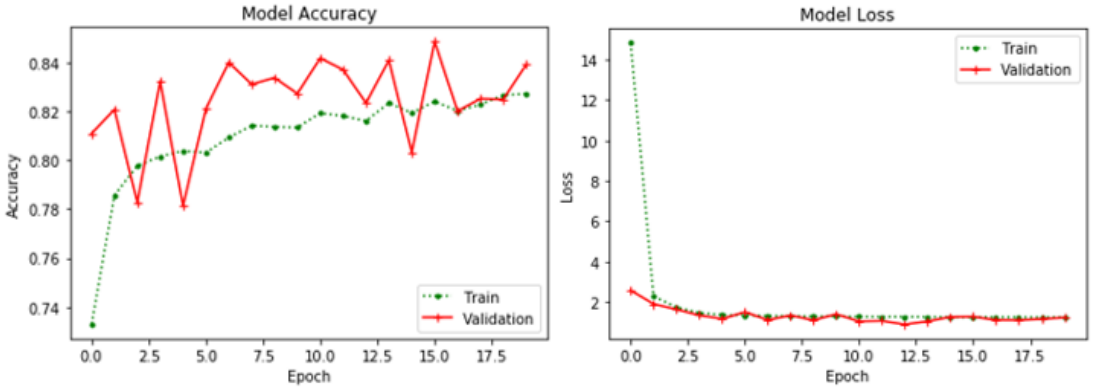

Figure 4.11: VGG-19, (128, 36) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.85	0.93	0.89	2386
vPorn	0.79	0.61	0.69	988
accuracy			0.84	3374
macro avg	0.82	0.77	0.79	3374
weighted avg	0.83	0.84	0.83	3374

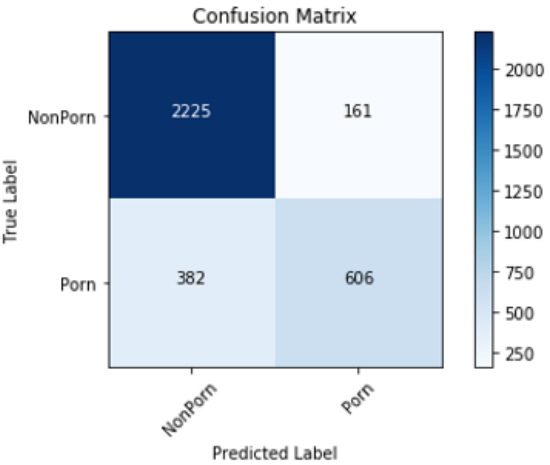

Figure 4.12: VGG-19 (128, 36) Classification Report and Confusion Matrix [Method 1]

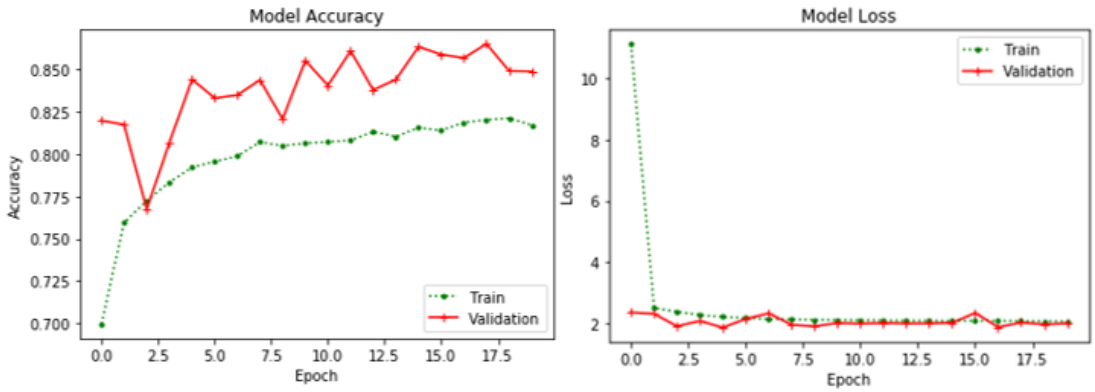

Figure 4.13: ResNet50_V2, (64, 8) Model Accuracy and Loss Graphs [Method 1]

	precision	recall	f1-score	support
vNonPorn	0.89	0.90	0.89	2386
vPorn	0.75	0.72	0.74	988
accuracy			0.85	3374
macro avg	0.82	0.81	0.81	3374
weighted avg	0.85	0.85	0.85	3374

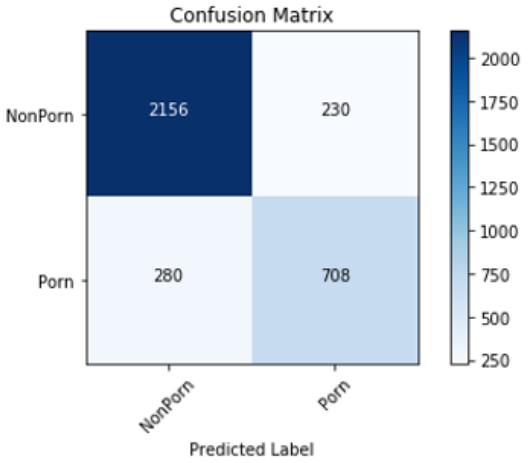

Figure 4.14: ResNet50_V2 (64, 8) Classification Report and Confusion Matrix [Method 1]

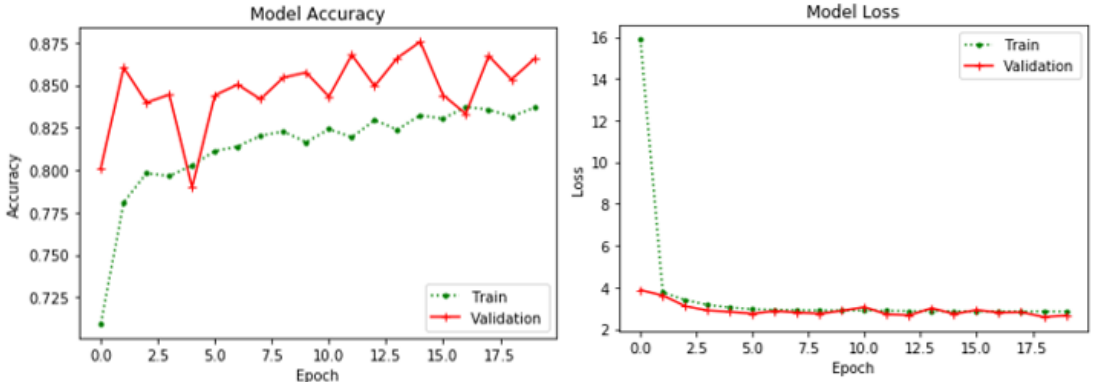

Figure 4.15: ResNet50_V2, (96, 24) Model Accuracy and Loss Graphs [Method 1]

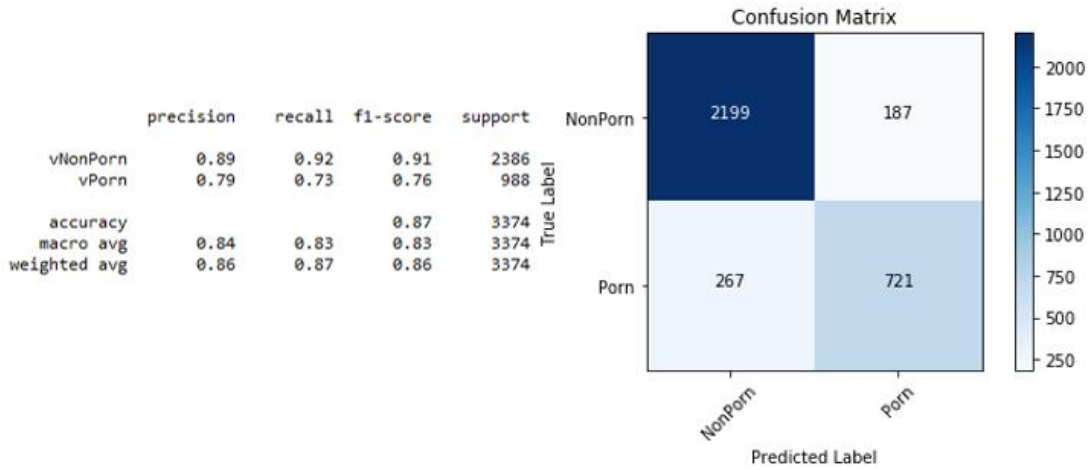

Figure 4.16: ResNet50_V2 (96, 24) Classification Report and Confusion Matrix [Method 1]

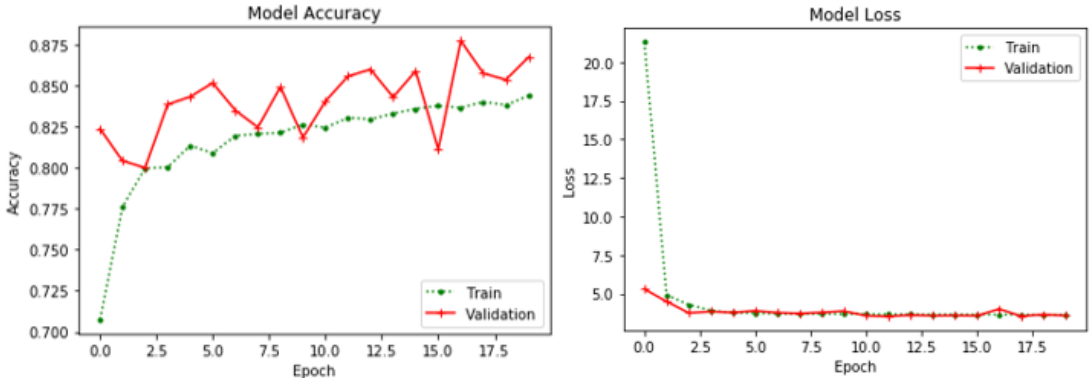

Figure 4.17: ResNet50_V2, (128, 36) Model Accuracy and Loss Graphs [Method 1]

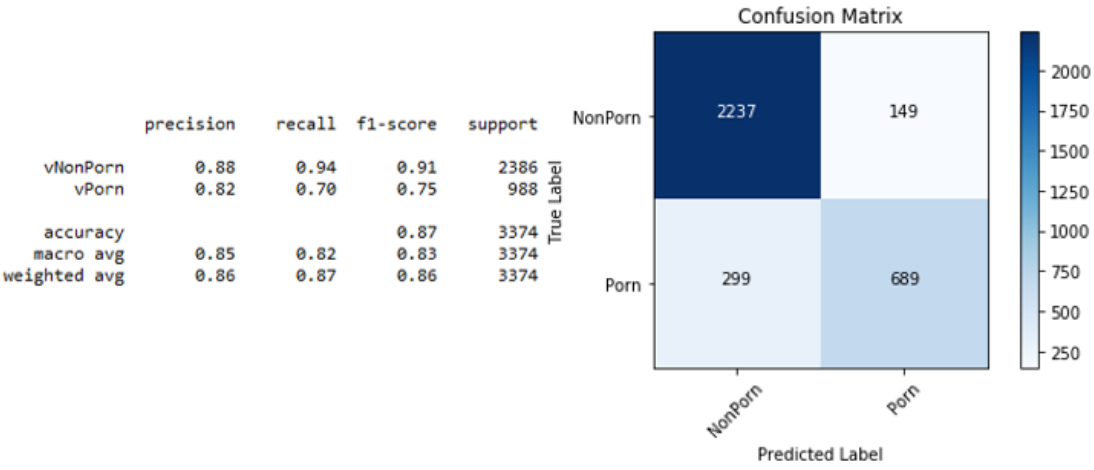

Figure 4.18: ResNet50_V2 (128, 36) Classification Report and Confusion Matrix [Method 1]

4.2.2 Method 2: Fine-tuned CNN as Feature Extractor and Classifier

The best combination of number of neurons obtained from Method 1 was noted and used to train each of the CNN architectures. Specifically, the combinations of (96, 24) for MobileNet, (64, 8) for VGG-19 and (128, 36) for ResNet50_V2 were reused in Method 2. This time, the parameter that was tuned was the number of layers towards the end (top) of the convolutional layers for each CNN architecture that were re-trained. The results obtained from implementation of Method 2 are recorded in **Table 4.3**, where the last column is Validation Accuracy denoted as Val Accuracy and the Number of Layer column refers to Number of Trainable Layer towards the top of the CNN convolutional base. **Figure 4.19** to **Figure 4.36** show the model accuracy and loss graphs, confusion matrices and classification reports generated using **sklearn** library [31] for the selected CNN architectures using Method 2.

Table 4.3: Results Obtained from Implementation of Method 2

CNN Model and Number of Neuron	Number of Trainable Parameter	Number of Layer	Elapsed Time (hh:mm:ss)	Model Accuracy and Loss Graphs / Classification Report and Confusion Matrix	Val Accuracy (%)
MobileNet (96, 24)	1,689,002	10	01:10:02	**Figure 4.19 / Figure 4.20**	88.77
	1,963,434	20	01:10:32	**Figure 4.21 / Figure 4.22**	90.78
	2,496,426	30	01:11:36	**Figure 4.23** / **Figure 4.24**	**91.82**
VGG-19 (64, 8)	11,045,466	5	01:10:43	**Figure 4.25** / **Figure 4.26**	**91.79**
	15,765,082	8	01:11:36	**Figure 4.27** / **Figure 4.28**	88.53
	19,305,050	10	01:12:26	**Figure 4.29** / **Figure 4.30**	87.20
ResNet50 _V2 (128, 36)	20,729,582	20	01:13:54	**Figure 4.31** / **Figure 4.32**	**92.68**
	27,293,422	30	01:14:12	**Figure 4.33** / **Figure 4.34**	91.35
	29,202,158	50	01:15:07	**Figure 4.35** / **Figure 4.36**	89.72

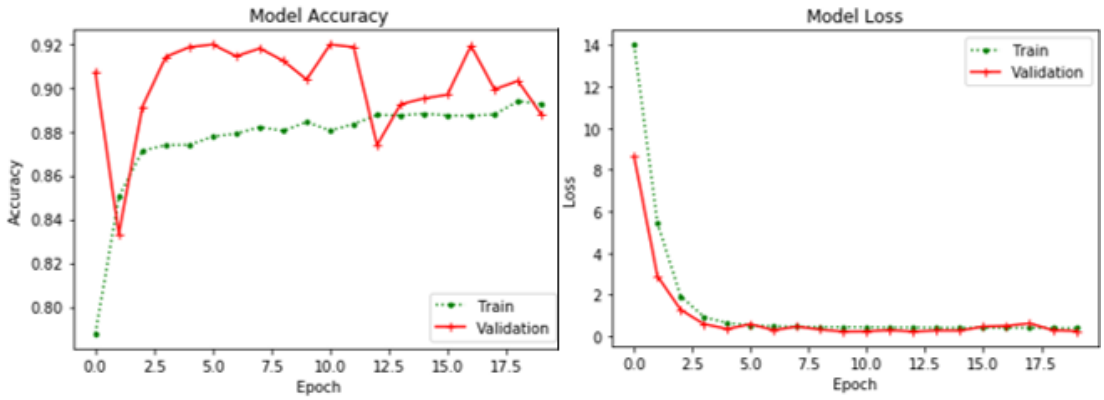

Figure 4.19: MobileNet (10 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.87	0.99	0.93	2386
vPorn	0.96	0.65	0.77	988
accuracy			0.89	3374
macro avg	0.91	0.82	0.85	3374
weighted avg	0.90	0.89	0.88	3374

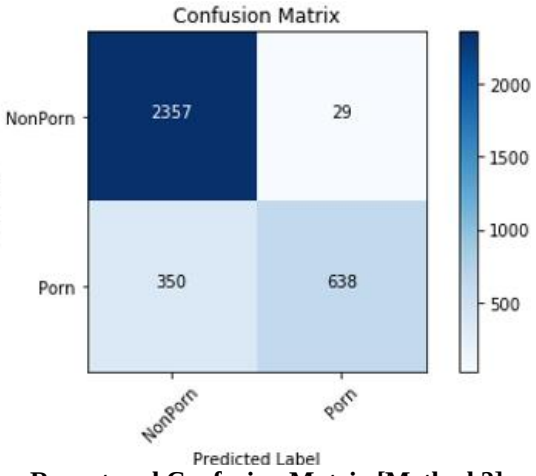

Figure 4.20: MobileNet (10 Layers) Classification Report and Confusion Matrix [Method 2]

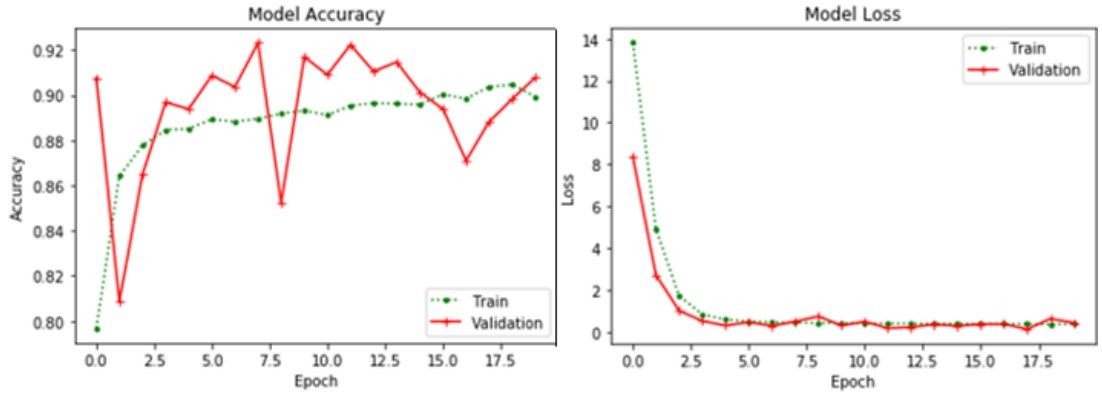

Figure 4.21: MobileNet (20 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.90	0.98	0.94	2386
vPorn	0.93	0.74	0.82	988
accuracy			0.91	3374
macro avg	0.92	0.86	0.88	3374
weighted avg	0.91	0.91	0.90	3374

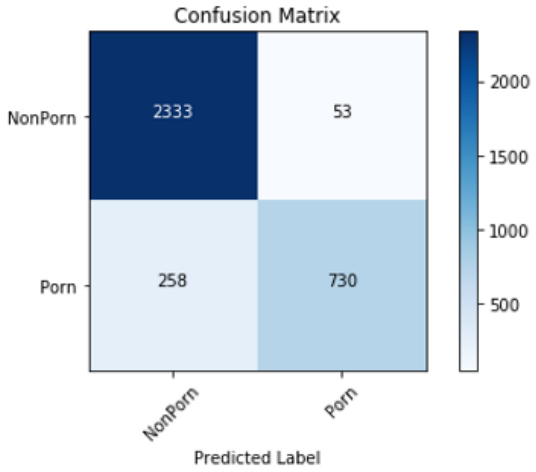

Figure 4.22: MobileNet (20 Layers) Classification Report and Confusion Matrix [Method 2]

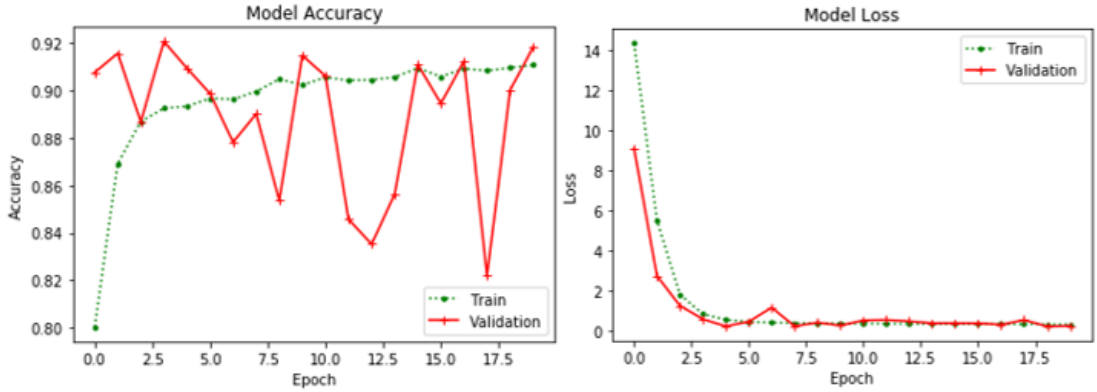

Figure 4.23: MobileNet (30 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.91	0.98	0.94	2386
vPorn	0.95	0.76	0.85	988
accuracy			0.92	3374
macro avg	0.93	0.87	0.89	3374
weighted avg	0.92	0.92	0.92	3374

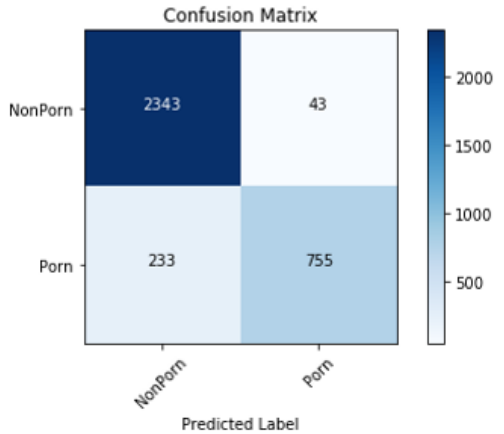

Figure 4.24: MobileNet (30 Layers) Classification Report and Confusion Matrix [Method 2]

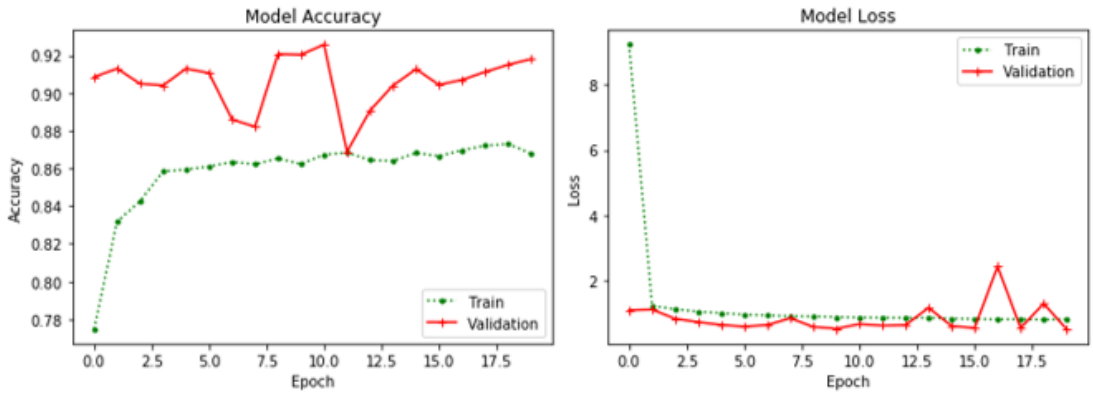

Figure 4.25: VGG-19 (5 Layers) Model Accuracy and Loss Graphs [Method 2]

vNonPorn	0.93	0.96	0.94	2386
vPorn	0.89	0.83	0.86	988
accuracy			0.92	3374
macro avg	0.91	0.89	0.90	3374
weighted avg	0.92	0.92	0.92	3374

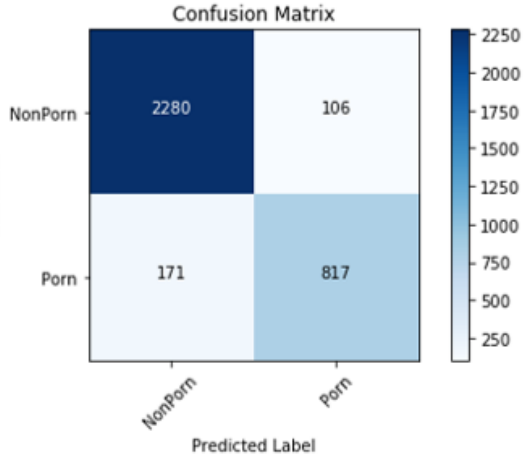

Figure 4.26: VGG-19 (5 Layers) Classification Report and Confusion Matrix [Method 2]

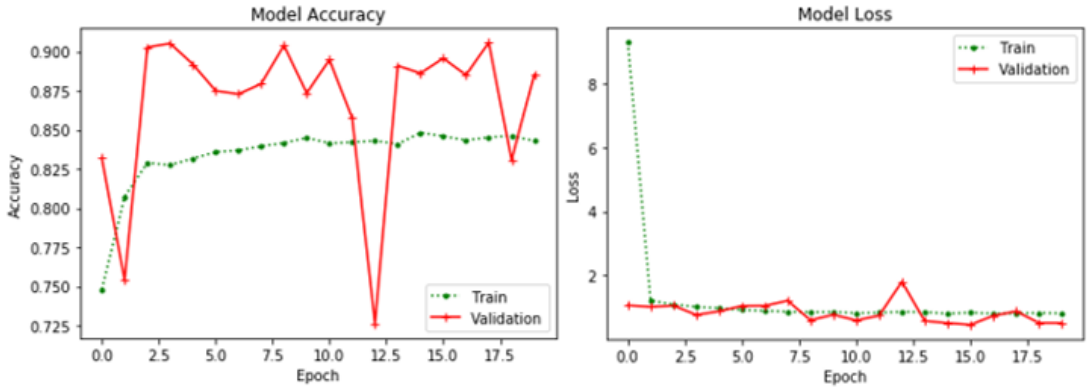

Figure 4.27: VGG-19 (8 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.95	0.88	0.92	2386
vPorn	0.76	0.89	0.82	988
accuracy			0.89	3374
macro avg	0.86	0.89	0.87	3374
weighted avg	0.89	0.89	0.89	3374

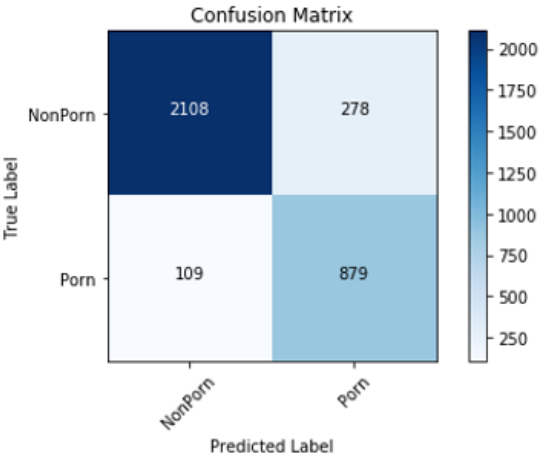

Figure 4.28: VGG-19 (8 Layers) Classification Report and Confusion Matrix [Method 2]

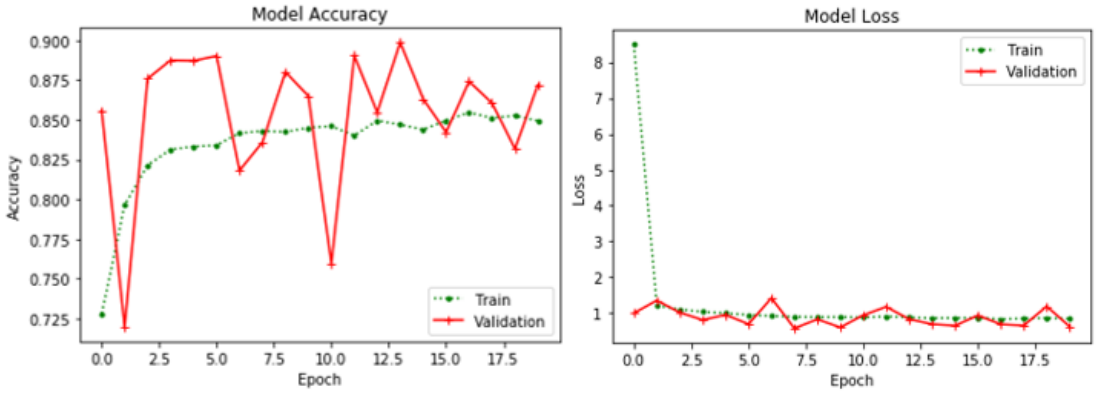

Figure 4.29: VGG-19 (10 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.95	0.87	0.91	2386
vPorn	0.73	0.88	0.80	988
accuracy			0.87	3374
macro avg	0.84	0.88	0.85	3374
weighted avg	0.88	0.87	0.88	3374

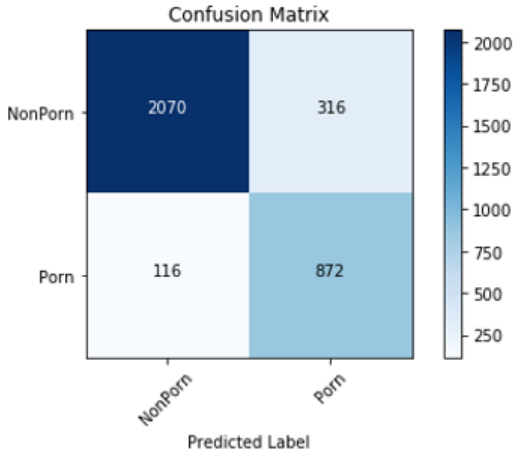

Figure 4.30: VGG-19 (10 Layers) Classification Report and Confusion Matrix [Method 2]

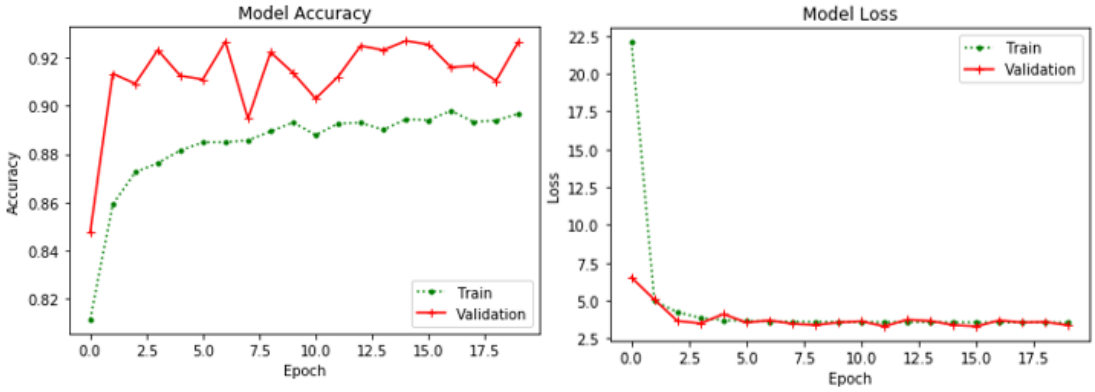

Figure 4.31: ResNet50_V2 (20 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.93	0.97	0.95	2386
vPorn	0.92	0.82	0.87	988
accuracy			0.93	3374
macro avg	0.92	0.90	0.91	3374
weighted avg	0.93	0.93	0.93	3374

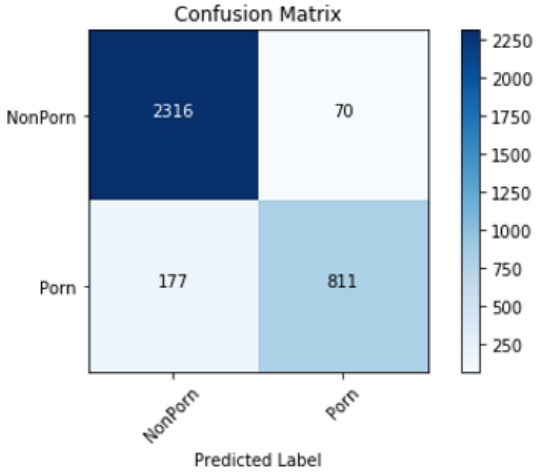

Figure 4.32: ResNet50_V2 (20 Layers) Classification Report and Confusion Matrix [Method 2]

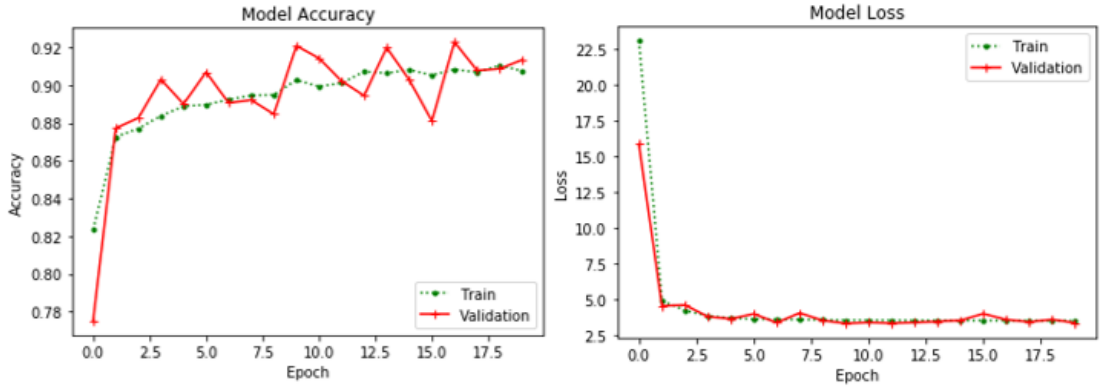

Figure 4.33: ResNet50_V2 (30 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.90	0.98	0.94	2386
vPorn	0.95	0.75	0.83	988
accuracy			0.91	3374
macro avg	0.92	0.86	0.89	3374
weighted avg	0.92	0.91	0.91	3374

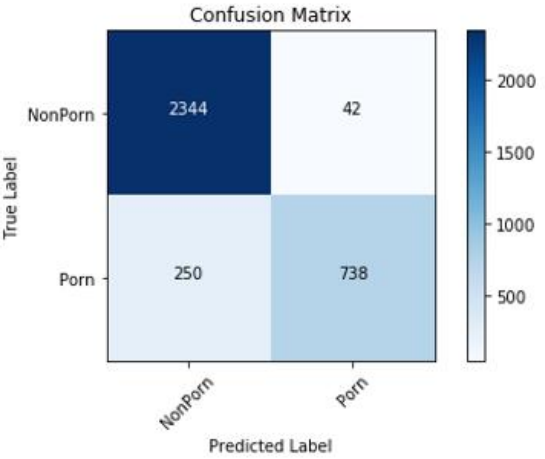

Figure 4.34: ResNet50_V2 (30 Layers) Classification Report and Confusion Matrix [Method 2]

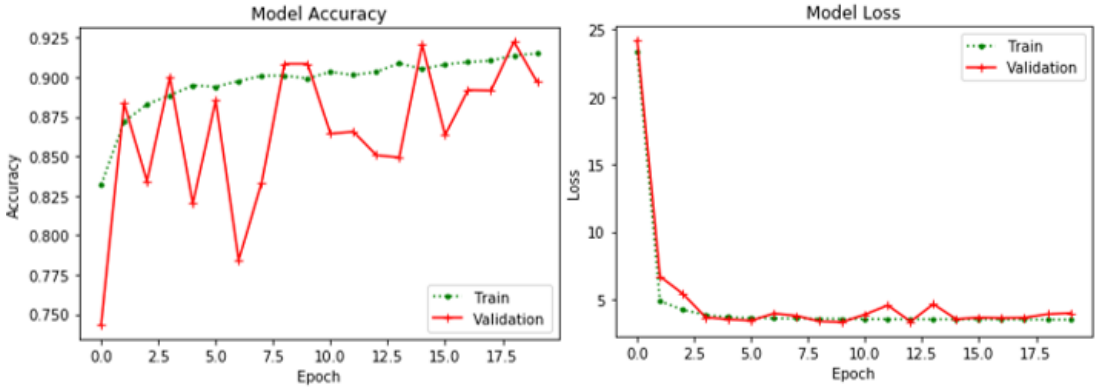

Figure 4.35: ResNet50_V2 (50 Layers) Model Accuracy and Loss Graphs [Method 2]

	precision	recall	f1-score	support
vNonPorn	0.88	0.99	0.93	2386
vPorn	0.95	0.68	0.80	988
accuracy			0.90	3374
macro avg	0.92	0.83	0.86	3374
weighted avg	0.90	0.90	0.89	3374

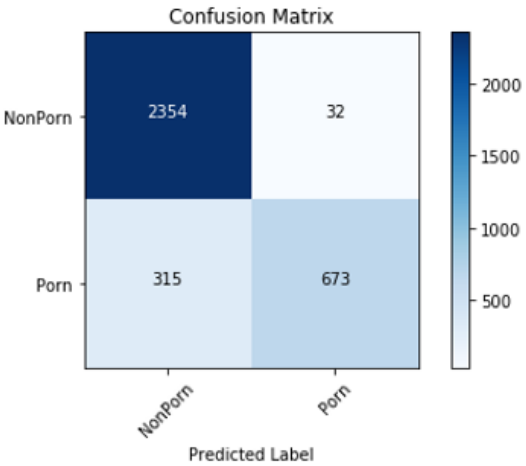

Figure 4.36: ResNet50_V2 (50 Layers) Classification Report and Confusion Matrix [Method 2]

4.2.3 Method 3: Traditional CNN as Feature Extractor and SVM as Classifier

In this method, SVM classifiers replaced the role of CNN as classifiers in Method 1. The trained model with the combination of number of neurons that produced the best results using Method 1 for each CNN architecture was loaded. Due to memory limitation, instead of the output of convolutional base, the output of the first added dropout layer was fed as input to the SVM to allow classification to be performed. The results obtained from implementation of Method 3 are recorded in **Table 4.4**, where the last column is Validation Accuracy denoted as Val Accuracy and the first and second columns under Number of Feature is for Training Dataset and Validation Dataset respectively. **Figure 4.37** to **Figure 4.39** show the confusion matrices and classification reports generated using **sklearn** library [31] for the CNN architectures involved.

Table 4.4: Results Obtained from Implementation of Method 3

CNN Model	Number of Feature		Number of Neuron	Elapsed Time (mm:ss)	Classification Report and Confusion Matrix	Val Accuracy (%)
	Train	Val				
MobileNet	1,574,688	323,904	(96, 24)	01:15	**Figure 4.37**	**87.05**
VGG-19	1,049,792	215,936	(64, 8)	02:14	**Figure 4.38**	84.91
ResNet50 _V2	2,099,584	431,872	(128, 36)	02:21	**Figure 4.39**	86.84

	precision	recall	f1-score	support
vNonPorn	0.89	0.93	0.91	2386
vPorn	0.81	0.73	0.77	988
accuracy			0.87	3374
macro avg	0.85	0.83	0.84	3374
weighted avg	0.87	0.87	0.87	3374

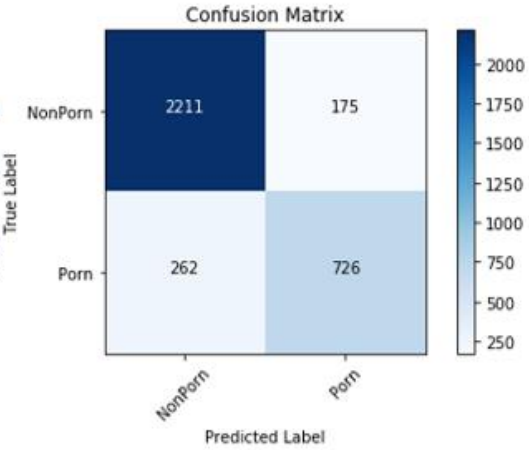

Figure 4.37: MobileNet (96, 24) Classification Report and Confusion Matrix [Method 3]

	precision	recall	f1-score	support
vNonPorn	0.87	0.93	0.90	2386
vPorn	0.80	0.65	0.72	988
accuracy			0.85	3374
macro avg	0.83	0.79	0.81	3374
weighted avg	0.85	0.85	0.84	3374

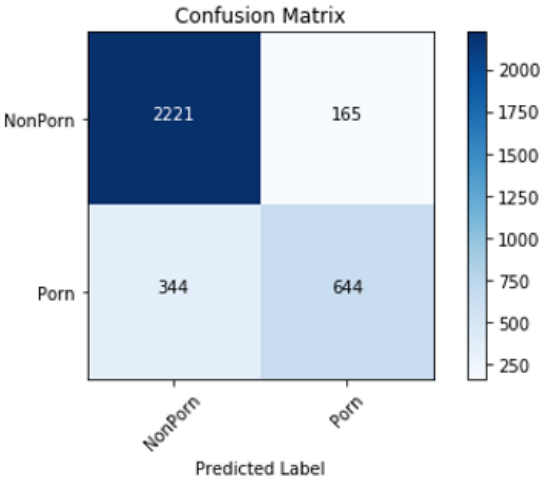

Figure 4.38: VGG-19 (64, 8) Classification Report and Confusion Matrix [Method 3]

	precision	recall	f1-score	support
vNonPorn	0.89	0.93	0.91	2386
vPorn	0.81	0.71	0.76	988
accuracy			0.87	3374
macro avg	0.85	0.82	0.84	3374
weighted avg	0.87	0.87	0.87	3374

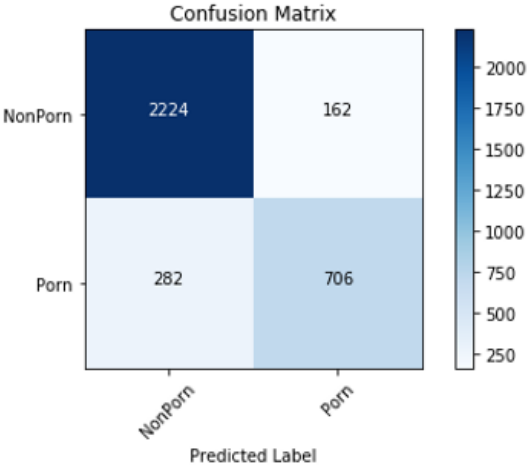

Figure 4.39: ResNet50_V2 (128, 36) Classification Report and Confusion Matrix [Method 3]

4.2.4 Method 4: Fine-tuned CNN as Feature Extractor and SVM as Classifier

Similar to Method 3, trained or fine-tuned CNNs that produced the best result for each CNN architecture using Method 2 was loaded to perform feature extraction. Subsequently, the SVM classifier was fed with output of the first added dropout layer as input for it to perform classification. This decision of using output of first added dropout layer instead of the output of the convolutional base as input into the SVM was done due to the memory space limitation.

The results obtained from implementation of Method 4 are recorded in **Table 4.5**, where the last column is Validation Accuracy denoted as Val Accuracy and the first and second columns under Number of Feature is for Training Dataset and Validation Dataset respectively. **Figure 4.40** to **Figure 4.42** show the confusion matrices and classification reports generated using **sklearn** library [31] for the selected CNN architectures using Method 4.

Table 4.5: Results Obtained from Implementation of Method 4

CNN Model and Number of Neuron	Number of Feature		Number of Trainable Layer	Elapsed Time (mm:ss)	Classification Report and Confusion Matrix	Val Accuracy (%)
	Train	Val				
MobileNet (96, 24)	1,574,688	323,904	30	01:08	**Figure 4.40**	92.77
VGG-19 (64, 8)	1,049,792	215,936	5	02:11	**Figure 4.41**	90.57
ResNet50_V2 (128, 36)	2,099,584	431,872	20	02:15	**Figure 4.42**	**92.80**

	precision	recall	f1-score	support
vNonPorn	0.93	0.97	0.95	2386
vPorn	0.92	0.83	0.87	988
accuracy			0.93	3374
macro avg	0.92	0.90	0.91	3374
weighted avg	0.93	0.93	0.93	3374

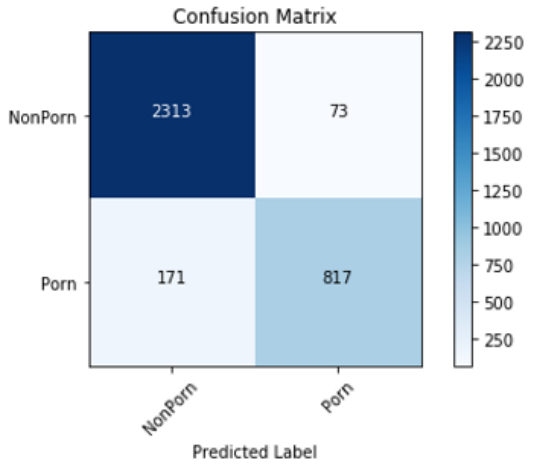

Figure 4.40: MobileNet (96, 24) (30 Layers) Classification Report and Confusion Matrix [Method 4]

	precision	recall	f1-score	support
vNonPorn	0.90	0.97	0.94	2386
vPorn	0.91	0.75	0.82	988
accuracy			0.91	3374
macro avg	0.91	0.86	0.88	3374
weighted avg	0.91	0.91	0.90	3374

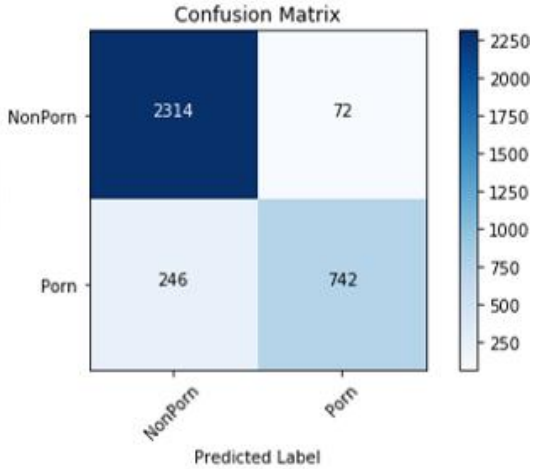

Figure 4.41: VGG-19 (64, 8) (5 Layers) Classification Report and Confusion Matrix [Method 4]

	precision	recall	f1-score	support
vNonPorn	0.94	0.96	0.95	2386
vPorn	0.90	0.85	0.87	988
accuracy			0.93	3374
macro avg	0.92	0.90	0.91	3374
weighted avg	0.93	0.93	0.93	3374

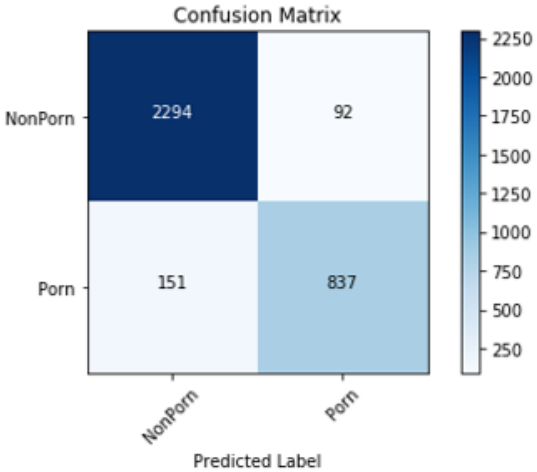

Figure 4.42: ResNet50_V2 (128, 36) (20 Layers) Classification Report and Confusion Matrix [Method 4]

4.3 Discussion of Findings

4.3.1 Comparison between Methods

Table 4.6, **Table 4.7** and **Table 4.8** compares the highest validation accuracies achieved by each of the methods explained in **Section 3.6** for the different CNN architectures. A significant outcome that can be extracted from the tables is validation accuracies obtained using Method 1 are the lowest regardless of the CNN model employed. For MobileNet and ResNet50_V2, validation accuracies achieved using Method 4 are the highest, followed by Method 2, Method 3 and Method 1. On the other hand, for VGG-19, highest validation accuracy is obtained using Method 2, followed by Method 4, Method 3 and Method 1. Interestingly, differences between accuracies obtained using Method 4 and Method 2 is in the range of 1.5%. The same goes for the differences between accuracies obtained using Method 1 and Method 3.

Table 4.6: Comparison among Usages of MobileNet with (96, 24) Neurons

Method	Validation Accuracy (%)	Difference Between Validation Accuracy with Baseline (%)
1 (Baseline)	86.93	-
2	91.82	4.89
3	87.05	0.12
4	**92.77**	**5.84**

Table 4.7: Comparison among Usages of VGG-19 with (64, 8) Neurons

Method	Validation Accuracy (%)	Difference Between Validation Accuracy with Baseline (%)
1 (Baseline)	84.65	-
2	**91.79**	**7.14**
3	84.91	0.26
4	90.57	5.92

Table 4.8: Comparison among Usages of ResNet50_V2 with (128, 36) Neurons

Method	Validation Accuracy (%)	Difference Between Validation Accuracy with Baseline (%)
1 (Baseline)	86.72	-
2	92.68	5.96
3	86.84	0.12
4	**92.80**	**6.08**

4.3.2 Observations from Tabulated Results

It is observed from **Table 4.2** that the number of trainable parameter increased as the number of neurons in the fully-connected classifier layers, which also denotes the output size of the layer, increased for each of the CNN architecture used. This correlation may be justified using the relationship stated in **Equation 4.2** that is used to calculate the number of parameter in densely-connected layers [29]. The upward trend in number of trainable parameter down the table column in **Table 4.3** is due to the increasing number of layers set to be trainable such that the CNN model can be fine-tuned to the target task. Additionally, it is apparent that the numbers of trainable parameter in Method 2 are significantly less than those in Method 1. The reason for this is connection weights of bottom layers of CNN models used in Method 2 were fixed to the initial weights imported from ImageNet classification task. On the contrary, connection weights of bottom CNN layers in Method 1 had to be updated via backpropagation as the training progresses to allow specific low-level and mid-level features to be recognised.

$$\text{Number of parameter} = \text{output size} \times (\text{input size} + 1) \qquad (4.2)$$

where output size = number of neuron specified in the densely-connected layer,
input size = output size of the previous layer

Data in **Table 4.4** and **Table 4.5** suggest that the number of neuron in the last feature extractor layer, which its output is fed into the classifier, may be related to the number of extracted features. Therefore, it is possible for the same number of features to be extracted for different CNN architectures as long as the numbers of neuron in the last feature extractor layer are the same.

Closer inspection of the tabulated data in **Section 4.2** shows the time required to train the CNN (in Method 1 and Method 2) and to apply the trained CNN to extract features (in Method 3 and Method 4) increases as the number of trainable parameters increases for each CNN architecture across all the implemented methods.

Interestingly, despite the significantly smaller amount of trainable parameters available in MobileNet (approximately 3.3 million in Method 1 and 2 million in Method 2) compared to VGG-19 (approximately 22 million in Method 1 and 15 million in Method 2), the time required to train both of these CNN architectures were almost the same. This discrepancy may be attributed to the larger amount of hidden layers in the MobileNet network compared to the VGG-19 network.

4.3.3 Hyperparameters

In the process of searching for the optimal method to apply CNN model in solving the problem specified in **Section 1.2**, the two hyperparameters that were slightly manipulated are the number of neurons in the classifier (Method 1) and number of layers towards the top of CNN model set to be trainable (Method 2). What is interesting about the data in **Table 4.2** that no particular pattern (increasing or decreasing trend) for the number of neurons in classifier can be detected in order to obtain a better network performance. Similarly, data in **Table 4.3** show that there is no specific way (increase or decrease) to manipulate the number of trainable layer in the pre-trained CNN architectures in order to improve the results. This implies that the optimal CNN solution cannot be deduced from performing simple experiments as there is still limitless possibilities for combinations of all the CNN hyperparameters. On the contrary, hyperparameters of CNN models should be tuned in a systematic and effective manner in order to allow improvement of network performance. This is an important issue for future research.

4.3.4 Transfer Learning

From outcomes of Method 1 and Method 2, it is clearly seen that the facts related to transfer learning mentioned in **Section 3.5.1** are true. Model accuracy graphs of Method 2 demonstrate higher starting values of accuracy than those of Method 1. Higher training and validation accuracies are also obtained in Method 2 as compared to Method 1. From the steepness or trend of curves in the graphs in **Section 4.2**, there is potential for the networks to improve if more epochs are run. However, it is likely

that results obtained when transfer learning is applied will still be superior to those without the usage of transfer learning.

Surprisingly, the differences in validation accuracy obtained with and without transfer learning are not very significant (in the range of 10%). Besides the un-optimised CNN used, it seems possible for the relationship between source task and target task to be a reason for this phenomenon. As mentioned in [42], the source task and its relationship with the target task affects the effectiveness of transfer method. In this case, detection of sexually explicit visual content is not part of the source ImageNet classification task, which may be an explanation for this observation.

4.3.5 Model Accuracy and Loss Graphs

From the model accuracy graphs in **Section 4.2**, it is observed that most of the validation curves fluctuate and reach accuracy values higher than that of the training curves. There are several possible explanations for such results. Firstly, techniques of dropout and kernel regularisation that were performed during CNN training were not performed during the validation process. Next, points of validation curve were decided after each epoch had finished running while those of training curve were determined during each epoch to allow backpropagation of weights. Another explanation that may justify this phenomenon is the incorrect application of data augmentation method such that the augmented training data are still biased and do not cover specific properties that are present in the validation data. It is also possible that the difference in complexity of the datasets contributed to this phenomenon. The validation dataset may be easier to be classified than the training dataset. This is quite a common phenomenon faced by many researchers and users of machine learning algorithms such as CNN and recurrent neural networks (RNNs).

It can be safely stated that the CNN models are not overfitted or underfitted but had a good fit due to the shapes of the relatively smooth curves obtained for the model loss graphs (a sharp fall followed by intertwining of training and validation curves that do

not rise gradually). Therefore, the trained models are utilised in spite of the aforementioned vision flaw.

4.3.6 Classification Reports and Confusion Matrices

Classification reports and confusion matrices presented in figures of **Section 4.2** show that most of the accuracies, precisions and recall rates of "Non Porn" images and video frames are higher than those of the "Porn" category. This inconsistency may be due to more data belonging to the "Non Porn" class than the "Porn" class as shown in **Table 3.1**. Another possible explanation is features of data belonging to the "Porn" class are not definite and evident, especially with the inclusion of "Non Porn" data that are difficult to be categorised such as those related to sumo wrestling, breast feeding, beach wears and people undressing.

4.3.7 SVM versus CNN as Classifier

Findings in **Section 4.2** show that usage of SVM as classifier yielded better accuracy values than CNN (in the range of 2%) in all but one case. This is likely because inputs to the SVM were not outputs obtained from the convolutional base but rather from the first dropout layer added to the trained model, which forms part of the classifier. The reason for making such decision was due to the limitation of memory space. Another possible explanation for this is the trained CNNs used were not the optimum ones as the hyperparameters that can affect the outcomes were randomly assigned. No specific strategy was used to tune them and only a small amount of experiments was conducted to test different combinations of hyperparameter values. On the contrary, SVM is a matured classifier that can be implemented easily as compared to CNN.

4.3.8 Automated Detection of Inappropriate Visual Content

The objective to have an algorithm that is capable of detecting adult contents in video files is achieved with the implemented system. However, since the current algorithm only takes in images as input, the (key) frames from video files to be processed need to be extracted and placed in a folder with its directory path sent into the code so that it can be accessed by the algorithm. These are achieved by having a GUI, that will be shown in **Section 4.4**, to allow user interaction and checking of the product.

To distinguish the proposed system with those that were built based on assumptions [4, 6, 9], distributions of exposed skin in the frames do not matter as long as the training dataset is not biased. In fact, no assumption needs to be made provided the training dataset covers a wide range of data with different settings with similar distribution. Unlike [6], no conditional expression is needed for the implemented algorithm. A quality that made the proposed system better than [8] is the removal of requirement for human intervention in deciding the feature to be extracted due to the usage of CNN.

Despite the advantages, the proposed system is inferior to [21] as it will not work well in real-time environment. It is also inferior to methods of [12] and [14] as no adjustment is made to improve results obtained using the CNN model.

4.3.9 Best Combination

Based on the findings listed in **Section 4.2**, the CNN models, including the CNN architecture and its parameters, that produced the most desirable results for each method are shown in **Table 4.9**. Overall, the best accuracy performance was 92.80% achieved using the (128, 36) as the combination of number of neuron in fine-tuned ResNet50_V2 and SVM as classifier (Method 4).

Table 4.9: Compilation of CNN Model with Best Results

Method	CNN Model
Method 1: Traditional CNN as Feature Extractor and Classifier	MobileNet with (96, 24) neurons
Method 2: Fine-tuned CNN as Feature Extractor and Classifier	ResNet50_V2 with (128, 36) neurons and last 20 layers in convolutional base trainable
Method 3: Traditional CNN as Feature Extractor and SVM as Classifier	MobileNet with (96, 24) neurons
Method 4: Fine-tuned CNN as Feature Extractor and SVM as Classifier	ResNet50_V2 with (128, 36) neurons and last 20 layers in convolutional base trainable

4.4 GUI

A software named F-Filter, which stands for "Frame Filter", with the GUI explained in **Section 3.7** was written to perform automated detection of pornographic visual contents in video frames using the methods in **Table 4.9**. The GUI of this software allows interaction through a user-friendly interface.

The appearance of F-Filter upon launching the application is displayed in **Figure 4.43**. Method 4 with the best performance is selected to be the default method. **Figure 4.44** shows the interface before a successful adult content detection on a selected MP4 video file. The number of frames and the sizes of displayed images in the GUI differs according to the duration value entered by user. The greater the number of key frames, the smaller the images displayed.

Figure 4.43: F-Filter upon Launching the Software

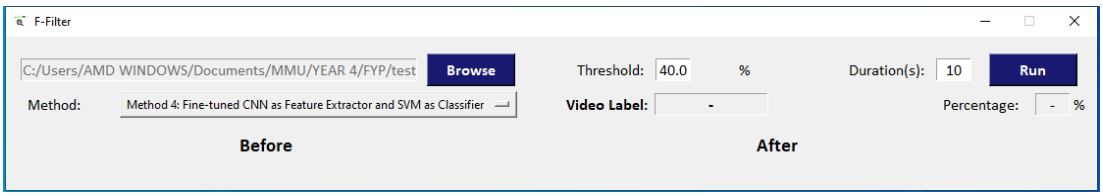

Figure 4.44: F-Filter After Setup, Right Before Clicking the "Run" Button

Figure 4.45 was generated by running a video containing pornographic contents using the default method, Method 4. The threshold was set to be 10% and key frames were extracted from the video every 20 seconds. The frames under the "Before" label are the original frames of the video while frames under the "After" label are the processed frames. The original frames that contain pornographic contents are pointed out in the figure, the rest of the frames are non-pornographic. Some frames were blackened or converted to black frames after processing (under "After" label), indicating the underlying algorithm (best CNN model combination obtained using Method 4) interpreted those frames as pornographic. The video was assigned the "Porn" label as the percentage of "Porn" frames was greater than the threshold. **Figure 4.46** to **Figure 4.48** display the interfaces after running the software on the same video file as input using the other methods.

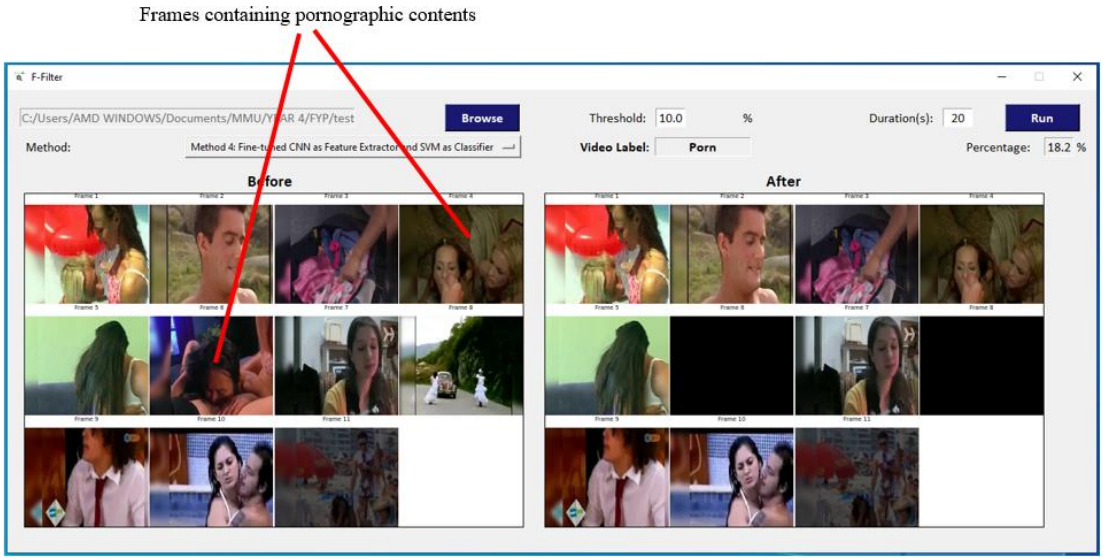

Figure 4.45: F-Filter After a Successful Run on a Video [Method 4]

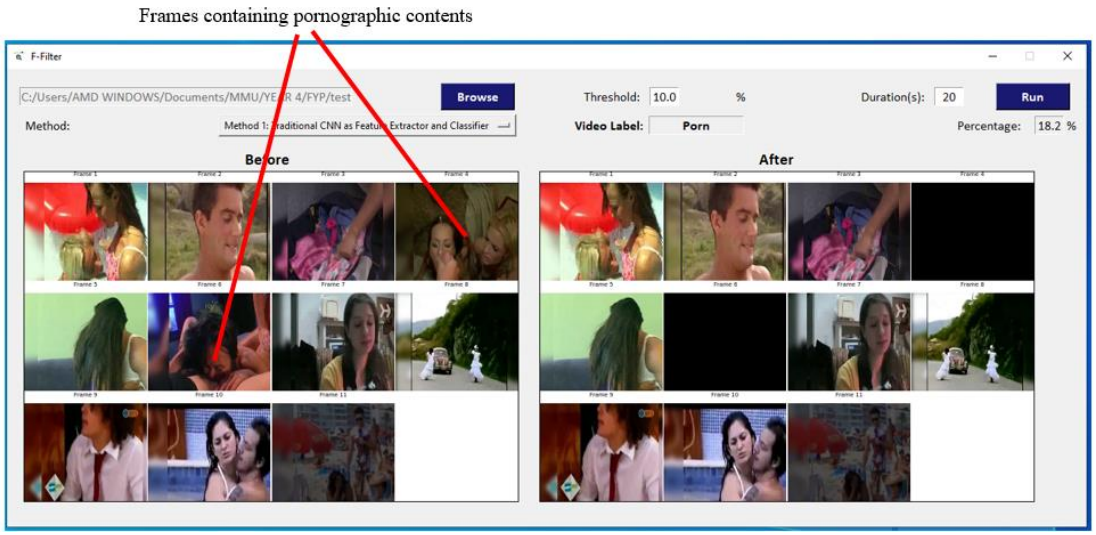

Figure 4.46: F-Filter After a Successful Run on a Video [Method 1]

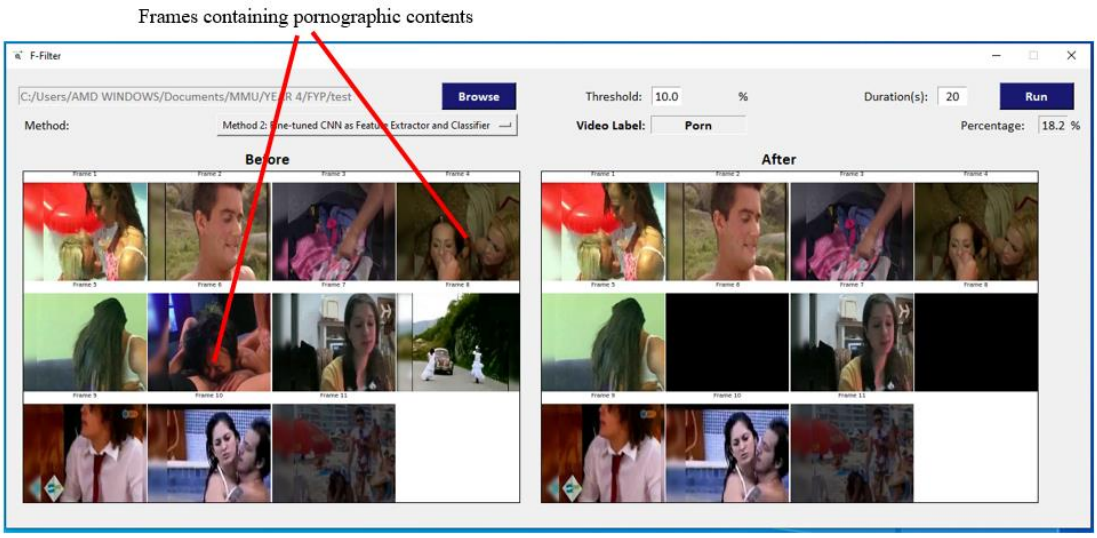

Figure 4.47: F-Filter After a Successful Run on a Video [Method 2]

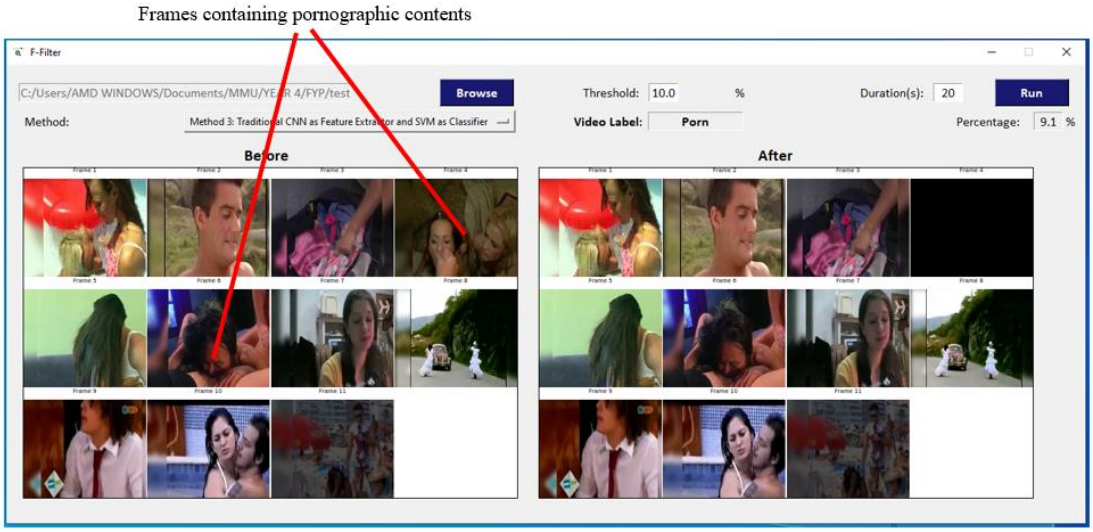

Figure 4.48: F-Filter After a Successful Run on a Video [Method 3]

Additionally, **Figure 4.49** to **Figure 4.52**, and **Figure 4.53** to **Figure 4.56** depict results obtained after running F-Filter with a pornographic anime video and a breastfeeding video as input respectively. The former is supposed to be given the "Porn" label while the latter should be assigned the "Non Porn" label. For the first video, the duration was set to 60 seconds due to the length of video and threshold was set to 30% due to the increased difficulty in classification. As for the second video, the duration was set as 20 seconds and the threshold was set to 40%. In addition, the original frames without pornographic content are pointed out when the input is the pornographic anime video as it is mainly made up of pornographic

frames. Conversely, all frames in the breastfeeding video did not contain pornographic content.

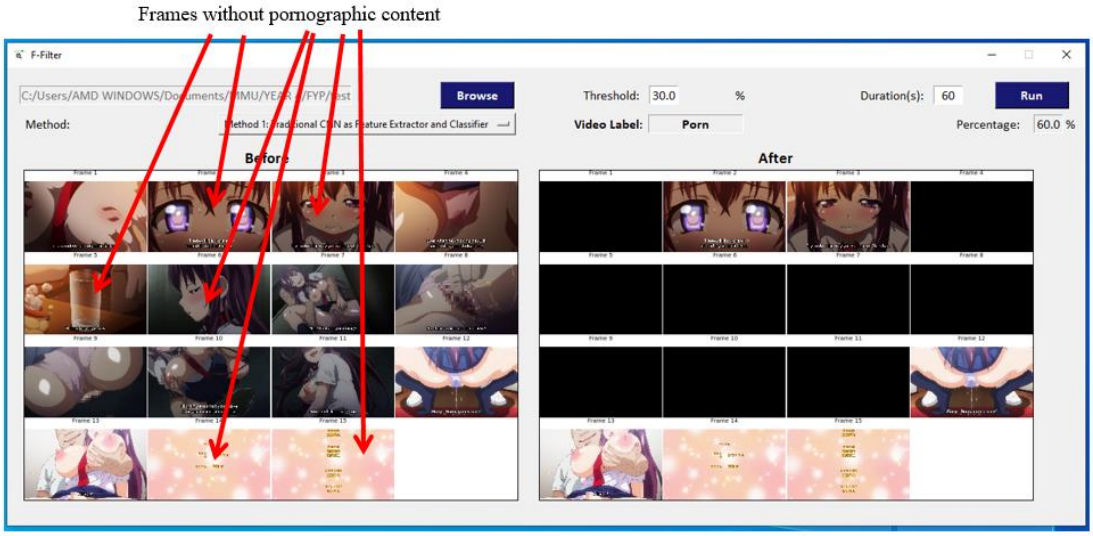

Figure 4.49: F-Filter After a Successful Run on a Pornographic Anime Video [Method 1]

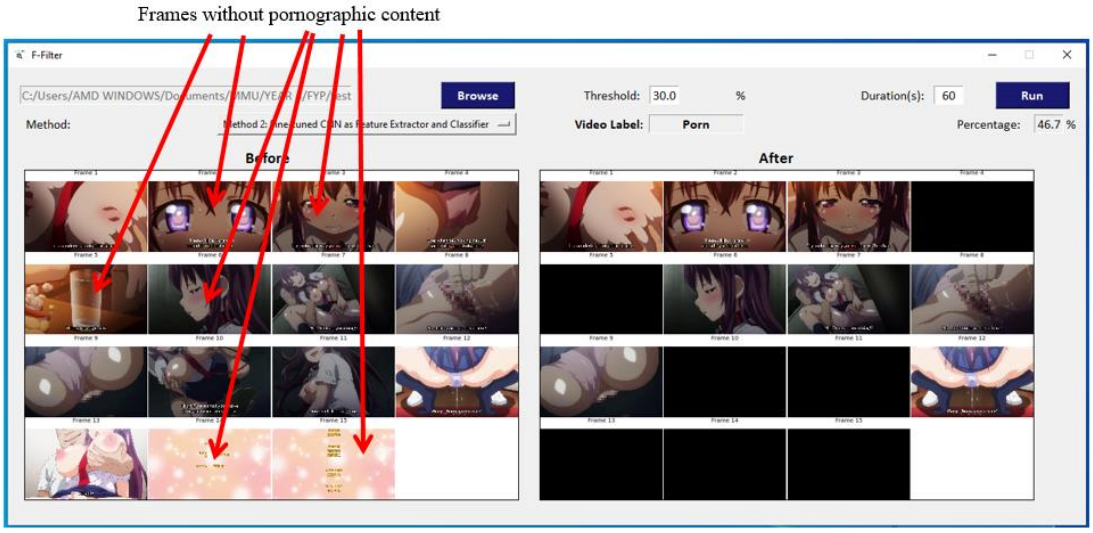

Figure 4.50: F-Filter After a Successful Run on a Pornographic Anime Video [Method 2]

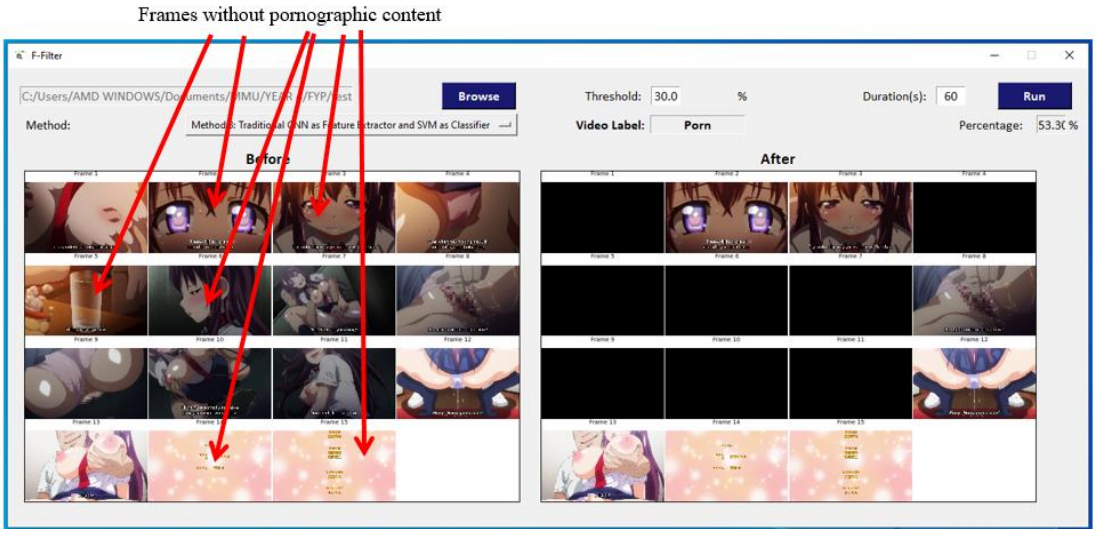

Figure 4.51: F-Filter After a Successful Run on a Pornographic Anime Video [Method 3]

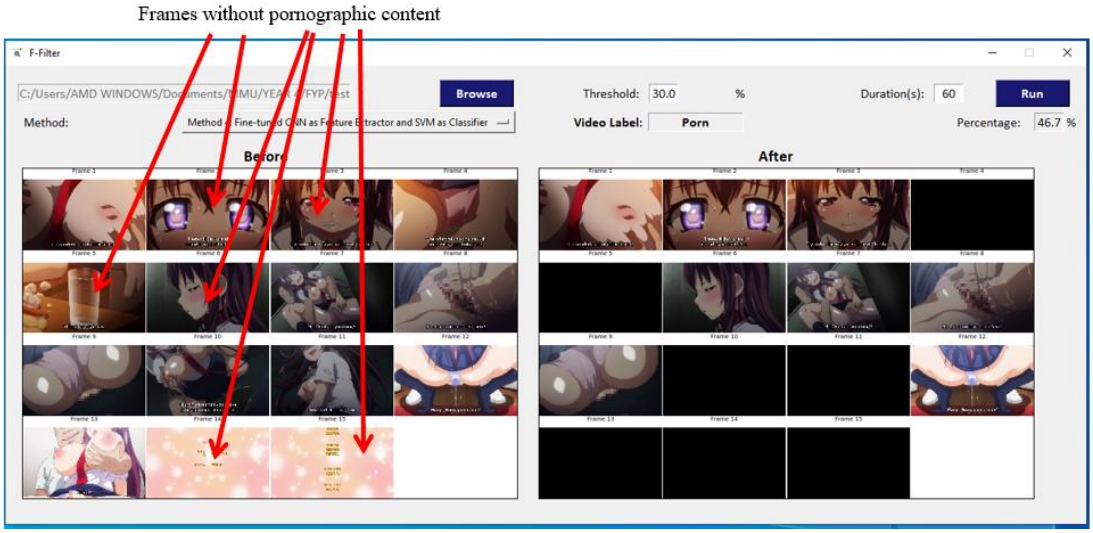

Figure 4.52: F-Filter After a Successful Run on a Pornographic Anime Video [Method 4]

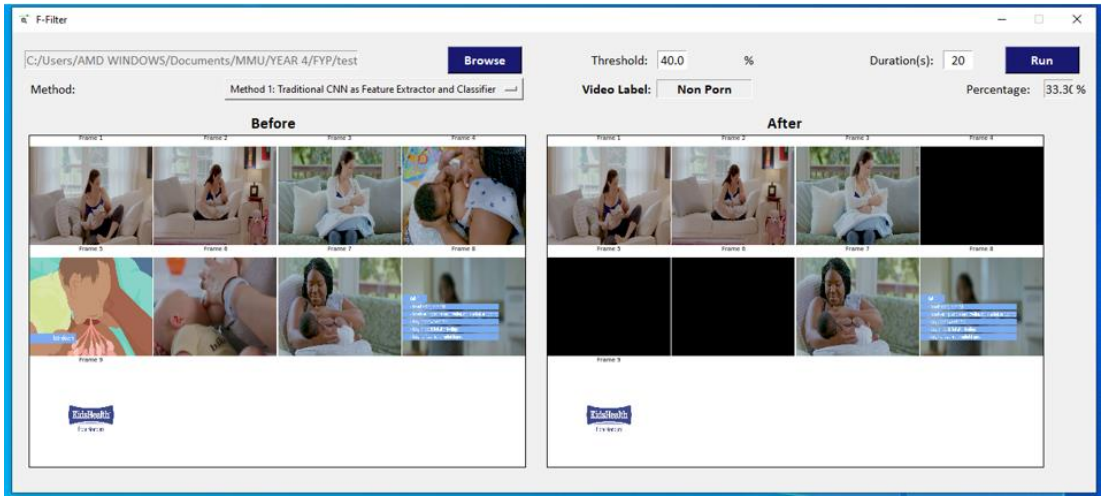

Figure 4.53: F-Filter After a Successful Run on a Breastfeeding Video [Method 1]

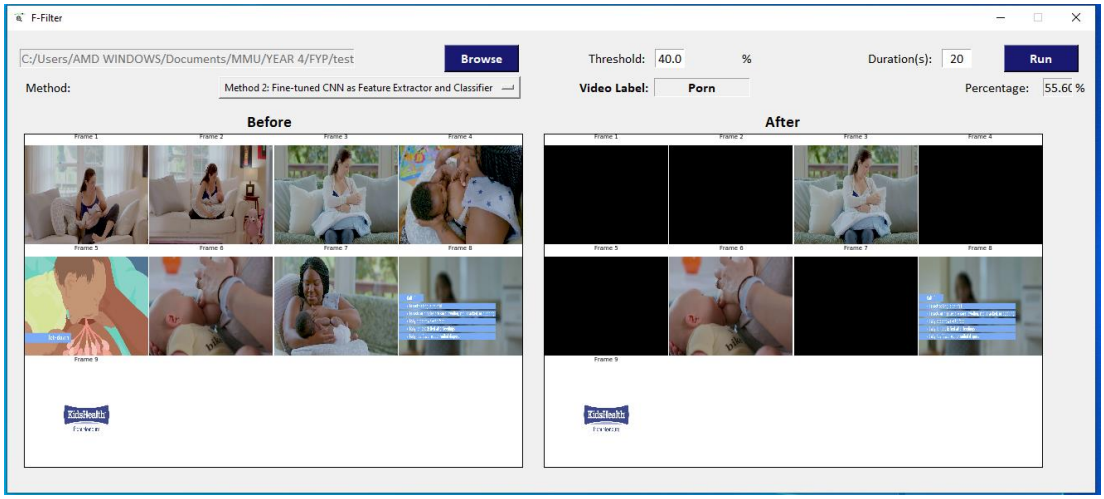

Figure 4.54: F-Filter After a Successful Run on a Breastfeeding Video [Method 2]

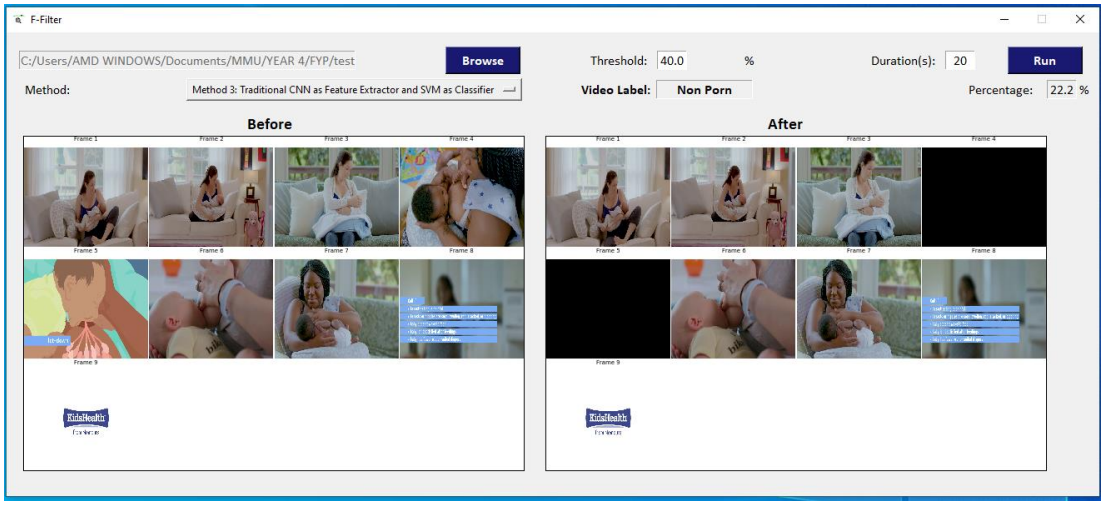

Figure 4.55: F-Filter After a Successful Run on a Breastfeeding Video [Method 3]

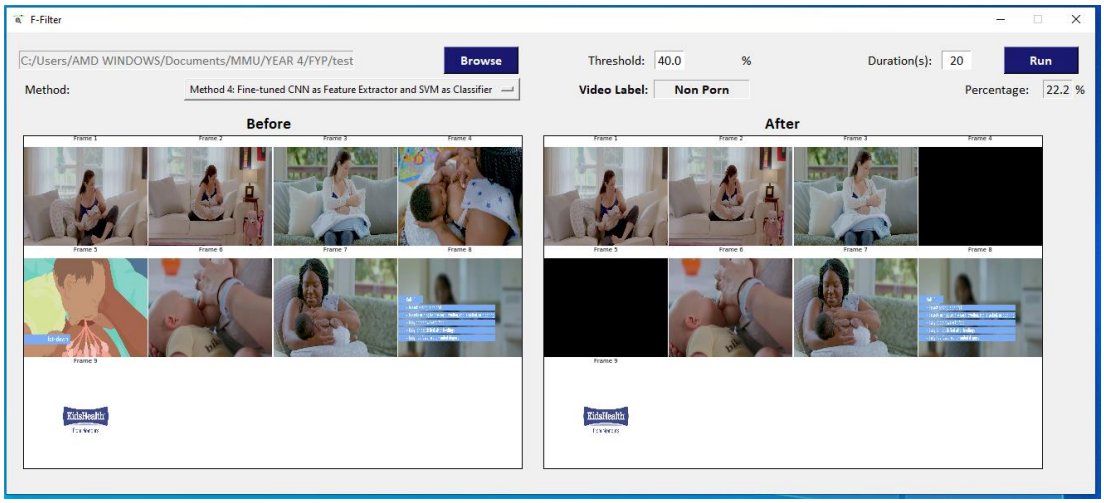

Figure 4.56: F-Filter After a Successful Run on a Breastfeeding Video [Method 4]

Besides that, error checks were included for the user-specified threshold, duration and directory path of the video file to be processed. The error message in **Figure 4.57** will be displayed if user left the directory path blank, which occurs when they fail to select the video file to be processed.

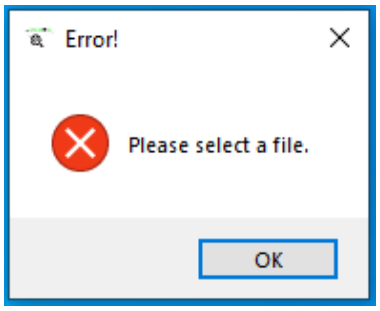

Figure 4.57: Error Message for Empty Directory Path

Figure 4.58 shows the error message that will pop up if user entered an invalid threshold (less than or equal to 0, or more than or equal to 100) or if they left it empty.

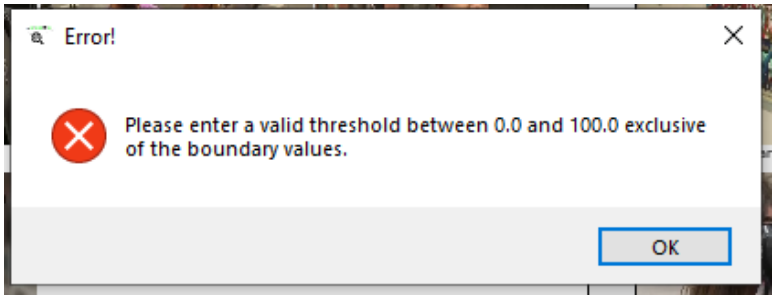

Figure 4.58: Error Message for Invalid Threshold

The error message shown in **Figure 4.59** will be displayed if the user-specified duration is invalid or if user did not specify the duration.

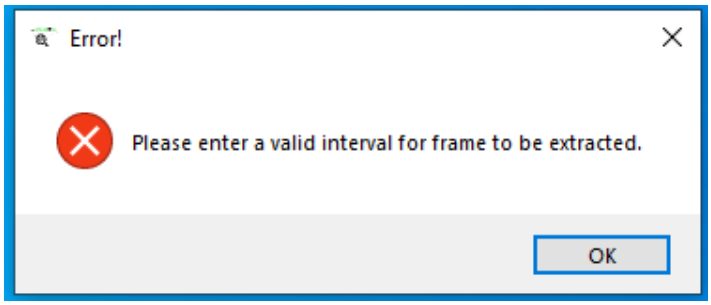

Figure 4.59: Error Message for Invalid Duration

CHAPTER 5 CONCLUSIONS

5.1 Summary and Conclusions

This report presented an automated detection of pornographic images from films using deep learning approach via CNN. Four methods were implemented to evaluate the effectiveness of CNN models. Basic CNN model with random initial weights was used as feature extractor as well as classifier in Method 1. Transfer learning technique was explored in Method 2 using initial weights imported from ImageNet classification task and fine-tuning was performed on the last few CNN convolutional layers. The effects of replacing CNN with SVM as classifier were tested in Method 3. Lastly, a combination of Method 2 and Method 3 was experimented in Method 4 with the inclusion of both the transfer learning technique in training the CNN as feature extractor and usage of SVM as classifier.

Deep learning applied using CNN allowed automated detection of inappropriate visual content in films by feeding a wide range of examples during the training process. Once the training is done, inputs to be filtered can be fed into the system from a specific a folder. As demonstrated in the project, transfer learning allowed training of CNN to be performed within a shorter amount of time with a higher start. However, effectiveness of such transfer method varies according to the degree of similarity between the source task, where initial weights are imported from, and the target task.

Overall, the best CNN model with a validation accuracy of 92.80% was obtained by applying Method 4 which uses SVM as classifier. The CNN model utilised was fine-tuned ResNet50_V2 CNN architecture, with (128, 36) as the combination of number of neuron and 20 layers towards the classifier set as trainable. This study strengthens the idea that transfer learning can improve CNN performance. Notwithstanding the limitations of this project, SVM classifier performed better than CNN classifier.

5.2 Areas of Future Research

This research has thrown up many questions in need of further investigation and confirmation. As the experiments in this project involved usage of hyperparameters which were decided randomly, further research might explore the effects of hyperparameter tuning methods in optimising CNN model to improve its performance. Considerably more work will need to be done to determine the optimiser that can yield better performance than RMSprop optimiser utilised in this project. Another natural progression of this work is to apply the adjustment to training data explored in [12] to analyse the possible improvements that can be brought by this technique.

A fruitful research area for future work is the utilisation of other deep learning techniques like LSTM. Usage of this RNN technique allows sequence learning to be performed aside from learning of spatial data, which may be beneficial to detection problem in films. In addition, filtering of contents of videos such that output of the system is the censored version of original videos is possible with application of this method.

REFERENCES

[1] M. Short, L. Black, A. Smith, C. Wetterneck and D. Wells, "A Review of Internet Pornography Use Research: Methodology and Content from the Past 10 Years", *Cyberpsychology, Behavior, and Social Networking*, vol. 15, no. 1, pp. 13-23, 2012. Available: 10.1089/cyber.2010.0477.

[2] E. Owens, R. Behun, J. Manning and R. Reid, "The Impact of Internet Pornography on Adolescents: A Review of the Research", *Sexual Addiction & Compulsivity*, vol. 19, no. 1-2, pp. 99-122, 2012. Available: 10.1080/10720162.2012.660431.

[3] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*. http://www.deeplearningbook.org: MIT Press, 2016.

[4] S. Nuraisha, F. Pratama, A. Budianita and M. Soeleman, "Implementation of K-NN based on histogram at image recognition for pornography detection", in *2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, 2017.

[5] P. Ganesan, V. Rajini, B. Sathish, V. Kalist and S. Khamar Basha, "Satellite image segmentation based on YCbCr color space", *Indian Journal of Science and Technology*, vol. 8, no. 1, p. 35, 2015. Available: http://indjst.org/index.php/indjst/article/view/51281/46256.

[6] M. Garcia, T. Revano, B. Habal, J. Contreras and J. Enriquez, "A pornographic image and video filtering application using optimized nudity recognition and detection algorithm", in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Baguio City, 2018, pp. 1-5.

[7] M. Shin, K. Chang and L. Tsap, "Does colorspace transformation make any difference on skin detection?", in *Sixth IEEE Workshop on Applications of*

Computer Vision, 2002. (WACV 2002). Proceedings., Orlando, FL, 2002, pp. 275-279.

[8] Z. Zhao and A. Cai, "Combining multiple SVM classifiers for adult image recognition", in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, 2010, pp. 149-153.

[9] M. Moustafa, "Applying deep learning to classify pornographic images and videos", in *7th Pacific-Rim Symposium on Image and Video Technology (PSIVT 2015)*, Auckland, 2015.

[10] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017. Available: 10.1145/3065386.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[12] F. Nian, T. Li, Y. Wang, M. Xu and J. Wu, "Pornographic image detection utilizing deep convolutional neural networks", *Neurocomputing*, vol. 210, pp. 283-293, 2016. Available: 10.1016/j.neucom.2015.09.135.

[13] M. Oquab, L. Bottou, I. Laptev and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks", in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 2014.

[14] X. Ou, H. Ling, H. Yu, P. Li, F. Zou and S. Liu, "Adult image and video recognition by a deep multicontext network and fine-to-coarse strategy", *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, pp. 1-25, 2017. Available: 10.1145/3057733.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, "Caffe: Convolutional architecture for fast feature embedding", in *Proceedings of the 2014 ACM International Conference on Multimedia*, 2014, pp. 675-678.

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

[17] I. Agastya, A. Setyanto, Kusrini and D. Handayani, "Convolutional neural network for pornographic images classification", in *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, Subang Jaya, 2018.

[18] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", in *International Conference on Learning Representations*, 2015.

[19] Z. Ying, P. Shi, D. Pan, H. Yang and M. Hou, "A deep network for pornographic image recognition based on feature visualization analysis", in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, 2018, pp. 212-216.

[20] J. Yosinski, J. Clune, Y. Bengio and H. Lipson, "How transferable are features in deep neural networks?", in *Advances in Neural Information Processing Systems*, 2014.

[21] J. Wehrmann, G. Simões, R. Barros and V. Cavalcante, "Adult content detection in videos with convolutional and recurrent neural networks", *Neurocomputing*, vol. 272, pp. 432-438, 2018. Available: 10.1016/j.neucom.2017.07.012.

[22] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos", in *Advances in neural information processing systems*, 2014, pp. 568-576.

[23] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, "Video pornography detection through deep learning techniques and motion information", *Neurocomputing*, vol. 230, pp. 279-293, 2017. Available: 10.1016/j.neucom.2016.12.017.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015. Available: 10.1007/s11263-015-0816-y.

[25] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, "Pornography classification: The hidden clues in video space–time", *Forensic Science International*, vol. 268, pp. 46-61, 2016. Available: 10.1016/j.forsciint.2016.09.010.

[26] S. Avila, N. Thome, M. Cord, E. Valle and A. de A. Araújo, "Pooling in image representation: The visual codeword point of view", *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453-465, 2013. Available: 10.1016/j.cviu.2012.09.007.

[27] D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, 2012.

[28] "Welcome to Python.org", *Python.org*, 2020. [Online]. Available: https://www.python.org/. [Accessed: 16- Jan- 2020].

[29] "Home - Keras Documentation", *Keras.io*, 2020. [Online]. Available: https://keras.io/. [Accessed: 16- Jan- 2020].

[30] "Matplotlib: Python plotting — Matplotlib 3.1.2 documentation", *Matplotlib.org*, 2020. [Online]. Available: https://matplotlib.org/. [Accessed: 16- Jan- 2020].

[31] "scikit-learn: machine learning in Python — scikit-learn 0.22.1 documentation", *Scikit-learn.org*, 2020. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 16- Jan- 2020].

[32] "Graphical User Interfaces with Tk — Python 3.8.1 documentation", *Docs.python.org*, 2020. [Online]. Available: https://docs.python.org/3/library/tk.html. [Accessed: 16- Jan- 2020].

[33] "OpenCV", *Opencv.org*, 2020. [Online]. Available: https://opencv.org/. [Accessed: 16- Jan- 2020].

[34] S. Team, "Spyder Website", *Spyder-ide.org*, 2020. [Online]. Available: https://www.spyder-ide.org/. [Accessed: 16- Jan- 2020].

[35] "Anaconda | The World's Most Popular Data Science Platform", *Anaconda*, 2020. [Online]. Available: https://www.anaconda.com/. [Accessed: 16- Jan- 2020].

[36] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[37] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996, p. 354.

[38] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995, p. 372.

[39] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Adreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *arXiv preprint arXiv:1704.04861*, 2017.

[40] Y. Zheng, C. Yang and A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography", in *Computational Imaging III*, 2018.

[41] K. He, X. Zhang, S. Ren and J. Sun, "Identity Mappings in Deep Residual Networks", in *European conference on computer vision*, 2016, pp. 630-645.

[42] L. Torrey and J. Shavlik, "Transfer Learning", in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, E. Soria, J. Martin, R.Magdalena, M. Martinez and A. Serrano, Ed. IGI Global, 2009, pp. 242-264.

[43] S. Pan and Q. Yang, "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010. Available: 10.1109/tkde.2009.191.

[44] F. Chollet, *Deep Learning with Python*, 1st ed. Manning Publications, 2017, pp. 104-110.

[45] "Activations - Keras Documentation", *Keras.io*, 2020. [Online]. Available: https://keras.io/activations/. [Accessed: 17- Jan- 2020].

[46] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", arXiv preprint arXiv:1207.0580, 2012.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014. [Accessed 26 January 2020].

[48] Q. Zheng, M. Yang, J. Yang, Q. Zhang and X. Zhang, "Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process", *IEEE Access*, vol. 6, pp. 15844-15869, 2018. Available: 10.1109/access.2018.2810849.

[49] C. Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Accessed 20 January 2020].

[50] E. Bochinski, T. Senst and T. Sikora, "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms", in *2017 IEEE International Conference on Image Processing*, Beijing, 2017.

[51] N. Aszemi and P. Dominic, "Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms", *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019. Available: 10.14569/ijacsa.2019.0100638.

[52] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves", in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233-240.