Video Content Restriction Using Gender and Age Classification

CS492 Project

MDL15CS019	Aparna P M
MDL15CS078	Nikitha John
MDL15CS096	Saurav Jayakumar
MDL15CS100	Shilpa S

B. Tech Computer Science & Engineering

Department of Computer Engineering
Model Engineering College
Thrikkakara, Kochi 682021
Phone: +91.484.2575370
http://www.mec.ac.in
hodcs@mec.ac.in

May 2019

Model Engineering College Thrikkakara
Department of Computer Engineering

C E R T I F I C A T E

This is to certify that, this report titled ***Video Content Restriction Using Gender and Age Classification*** is a bonafide record of the work done by

MDL15CS019	**Aparna P M**
MDL15CS078	**Nikitha John**
MDL15CS096	**Saurav Jayakumar**
MDL15CS100	**Shilpa S**

Eighth Semester B. Tech. Computer Science & Engineering

students, for the course work in **CS492 Project**, which is the second part of the two semester project work, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, B. Tech. Computer Science & Engineering of **APJ Abdul Kalam Technological University**.

Guide

Reshma Prasad
Assistant Professor
Computer Engineering

Coordinator

Manilal D L
Associate Professor
Computer Engineering

Head of the Department

Manilal D L
Associate Professor
Computer Engineering

June 8, 2019

Acknowledgements

This project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

First of all, we would like to thank our esteemed Principal, Dr.Vinu Thomas, for his guidance and support in maintaining a calm environment to work in and also for providing the facilities that this work demanded.

We are highly indebted to our Project Coordinator and Head of the Department Dr. Manilal D L for his guidance, support and constant supervision throughout the duration of the work as well as for providing all the necessary information and facilities that this work demanded.

We would like to thank our Project Guide, Mrs. Reshma Prasad for her support and valuable insights and also for helping us out in correcting any mistakes that were made during the course of the work.

We offer our sincere gratitude to all our friends and peers for their support and encouragement that helped us get through the tough phases during the course of this work.

Last but not the least, we thank the Almighty God for guiding us through and enabling us to complete the work within the specified time.

Aparna P M

Nikitha John

Saurav Jayakumar

Shilpa S

Abstract

Some videos in the public domain may not violate the online video content policies, however, they are not necessarily appropriate for all audiences. Such videos may contain vulgar languages, disturbing imagery of violence, portrayal of harmful or dangerous activities, etc. In these cases, review teams may place an age restriction on the video when they are notified of the content. So usually age-restricted videos are not visible to users who are logged out, are under 18 years of age, or have Restricted Mode enabled.

But the age provided by a new user on sign-up can be falsified. This is a loophole that can be taken advantage of by many. Also, sometimes the real user can be a different person. So a way to dynamically acquire the age of the real user is required. The method proposed here is to use the device's camera to classify the gender and age of the real user.

So a good enough model that can analyze the user's age and gender accurately and quickly is important. Also a sufficiently good training data should be used to get a good classifier that is accurate enough to estimate the age and gender in real life applications. The usage of images over the internet has grown at an exponential rate and this afresh wealth of data has now enabled us to tackle computer vision problems that were previously complex to solve. But there exists a data scarcity problem that is due to the limitations of access to personal information which are private pieces of information, so we should alter the network architectures and algorithmic approaches to cope with these limitations.

The first working CNN for age and gender classification having 5 layers : 3 convolutional and 2 fully connected layers, has improved performance compared to previous non-CNN based approaches. Its performance could be improved here by adding Gabor filter to the input images.

The Gabor filter responses are extracted from the input and then image intensities are added to the Gabor output with weights. Such a network could show better performance over the former.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Proposed Project

1.1.1 Problem Statement

In recent times, the access of age-restricted content by minors both knowingly and unknowingly has gone up due to the easy access and more availability of such content. Steps to curb this using sign up data have been bypassed using loopholes such as falsified data, etc. So there is a need for a dynamic system that can verify the age of the real user who is trying to access the video. But such real-time classification also presents some problems of its own. The aging process depends on several factors such as gender, race, living habits and factors for gender determination depends on various facial attributes. The accuracy and performance of existing classifiers are not satisfactory and degrade in performance quickly and significantly, when evaluated on lower-quality images than they were trained with.

1.1.2 Proposed Solution

Video content restriction using gender and age classification is used here to dynamically capture the user's face image and then classify the age and gender based on which, access to view a age-restricted video is granted or blocked. The classifier network to be used comprises of a CNN along with Gabor filter, which is used to improve the accuracy and performance of the system. This method of estimating the age of the user in real time proves more effective than the user manually signing up with the age he provides which can easily be falsified. We shall incorporate this camera activity which captures user image and sends it the network for classification. The camera activity shall be invoked only when user tries to enable Age Restricted Content. We shall also develop an activity that records the view details and grant access status to such content.

Chapter 2

System Study Report

2.1 Literature Survey

1. Age and Gender Classification Using Wide Convolutional Neural Network and Gabor Filter[1]
Age and Gender Classification using convolutional neural networks has growing importance nowadays. Such classification without adequate accuracy cannot be used without concerns about its reliability. Hence classification methods where a high level of accuracy can be assured, are vital in current scenarios. A recognizable improvement in performance can be implemented by using Gabor filter extracts along with the input. This is in order to use some hand-crafted features in addition to the image itself as in the conventional CNN approaches. Specifically, we first extract Gabor filter responses to the input, and then add image intensities to the Gabor filter output with weights. Also, we use CNN with wider receptive field than the previous model, which also contributes to improve the performance.

2. Age/Gender Classification with Whole-Component Convolutional Neural Networks[2]
Whole-Component CNN solution contains both the whole face and the facial component networks, and a confidence analysis module to handle the classification task. It is composed of four major modules: 1) the face and facial components localization module 2) the whole face network 3) the facial component networks, and 4) the final classification module. The WC-CNN method applies the face detection and facial landmark localization techniques to input images for extracting the whole face and the facial component regions. The whole face network is composed by three convolutional and two fully connected layers. The overall classification accuracy is not very high, so the whole face network is used as the primary classifier to yield an initial classification result. Confidence analysis is then conducted to evaluate the confidence score of the initial decision.However miss classification if misalignment in facial component localization or with data constraints. This is a relatively simple CNN architecture which is easy to train and suitable for implementation in embedded systems with limited resources.

3. Multiple Hierarchical Decision on Neural Network to Predict Human Age Gender[3]
A feed forward propagation Neural Networks is constructed for human age and gender clas-

sification system for gray-scale facial images. In Feed-Forward Artificial Neural Network the data is forwarded to the next higher level for processing.The mean of each image is given as input to the neural network. In the initial hierarchy neural network classifies the faces based on the different age and gender,however there is chance for mis-detection. The Core Hierarchy is done after initial hierarchy by applying three Sigma control limits on the Neural network. The three Sigma control cover more than 90% of the population of the dataset under consideration. The mean and standard deviations are then calculated. Core hierarchy is done to reduce the mis-detection rate and improve the success rate of detection.The proposed method however fails to detect side-view faces, occluded faces and partial face image.The low complexity is suitable for real time implementations and the approach is efficient in terms of speed.

4. Face identification using Multi Task Learning[4]
The human attribute information is embedded into face representations. It combines important characteristics including very deep architecture, low dimensional representation and small filters. It consists of ten convolutional layers, five pooling layers and fully output layers that maps to different tasks. Instead of training on identity and then fine tuning by attribute, they make use of multiple loss function by multi-task learning. Multiple related tasks are learnt jointly. The also used multi-class CNN classifiers to detect the age information of face images in CASIA- WebFace and corrected some obviously wrong labels by human efforts. It uses more complicated multi-task learning with early stopping, but also learns a common features from CNN. The mentioned methods however requires input images to be well-aligned. The method decreases the needed size of the dataset and reduce the computational efforts.

5. Automatic Age Estimation Based on Facial Aging Patterns[5]
The model used to train the network here is the aging pattern, which is defined as the sequence of a particular individuals face images sorted in time order, by constructing a representative subspace. The proper aging pattern for a previously unseen face image is then determined by the projection in the subspace that can reconstruct the face image with minimum reconstruction error, while the position of the face image in that aging pattern will then indicate its age. During the age estimation process, the proper aging pattern for the test image is generated based on both the aging pattern subspace and the face image feature. In each iteration, the missing part of the personal aging pattern is first estimated by the current global aging pattern model. Then, the global model is further refined by the updated personal aging patterns. In this way, the commonality and the personality of the aging patterns are alternately utilized to learn the final subspace. The advantages of such a classification method are the ability to simulate facial aging effects and also can be applied on pose and illumination patterns. A disadvantage of this method is that it does not consider face shapes which can reduce the accuracy among children and old age group.

6. Joint Estimation of Age and Gender from Unconstrained Face Images using Lightweight Multitask CNN for Mobile Applications[6]
A Lightweight Multi-Task CNN (LMTCNN) is composed of one general convolution layer, two depth-wise separable convolution layers and two fully connected layers. It is very useful in accomplishing multiple tasks while reducing the memory cost. Here the general CNN is

decomposed into depth-wise and point-wise convolution networks. Also, to achieve both the age classication for eight age classes and the gender classication for two gender classes, two separate softmax layers are followed by the output feature map of the average pooling layer. The rst softmax layer assigns a probability for each class of the age and the other assigns a probability for each class of the gender. A major advantage of this method is that only a single CNN can be used to achieve multiple tasks which can reduce the memory requirements and hence can be realized on mobile devices. But performance of this method can be poor owing to its limitation in the number of CNNs used.

7. A Gender Classification Method Using Age Information[7]
   A gender classification method robust to age variation is suggested here by using age information and two facial features: appearance and geometry feature. Before classifying gender, the age estimation is used, we extract different feature for gender classification, based on this age group division. Gender classification uses two facial features; appearance feature and geometry feature. Facial shape feature obtained is used as global feature, and skin and wrinkle features are used as local feature. A very important advantage of such a classification is the improvement in accuracy where classification is not easy like in children or old age group. The use of age information to classify genders improves the performance significantly which increases the accuracy of classification among old age groups which is normally quite difficult. A disadvantage of such a classification is the requirement of age information along with the images dataset to train the network.

8. An Automatic Face Detection and Gender Identification from Color Images using Logistic Regression[8]
   It pre process the facial region by convolving with Gabor filters at five scales and eight orientations . Color model conversion algorithm is used. Input image is then converted to binary image. To remove small areas that have been obtained in previous stage, geometric operations, using the available filters, will be done on this area. An image is converted into 40 images with 5 scales and 8 orientations and the features are the individual Gabor filters coefficients. Processing of the image the detected face is cropped and converted to gray scale image. Image can be extracted from non -uniform background. Highest accuracy still reported with only 3.6 error is one of it's greatest advantage. Automatically detect face area from image. Filtered images should be sampled at regular intervals. Several steps of tuning required.

9. Combining Facial Dynamics With Appearance For Age Estimation[9]
   The aim of the proposed method is to estimate the age of subject by using a sequence of images that show the subject displaying a facial expression as input.It propose a fully automatic age estimation framework and uses the high-resolution UVA-NEMO Disgust Database. By using 3D volume changes via surface patches instead of landmark movements add frequency and facial asymmetry descriptors to the feature set.We use a two-level adaptive classification scheme and analyze gender-specific and spontaneity-specific effects of aging features. Using spontaneity information reduces the mean absolute error, advancing the state of the art for

facial age estimation but it can be tested on people with an age range between 8 and 76.

10. Age And Gender Classification Using Convolutional Neural Network[10]
Images are first rescaled to 256 x 256 and a crop of 227 x 227 is fed to the network. The output of the last fully connected layer is fed to a soft-max layer that assigns a probability for each class. The prediction itself is made by taking the class with the maximal probability for the given image. The network comprises of only three convolutional layers and two fully-connected layers with a small number of neurons. Smaller network design is motivated both from our desire to reduce the risk of overfitting as well as the nature of face recognition. Prediction in this technique consist of two methods. Center Crop: Feeding the network with the face image, cropped to 227 x 227 around the face center. Over-sampling: We extract five 227 x 227 pixel crop regions, four from the corners of the 256 x 256 face image, and an additional crop region from the center of the face. The network is presented with all five images, along with their horizontal reflections. Its final prediction is taken to be the average prediction value across all these variations. Overfitting is a common problem when machine learning based methods are used on such small image collections. This problem is exacerbated when considering this method due to their huge numbers of model parameters.Gender estimation mistakes frequently occur for images of babies or very young children where obvious gender attributes are not yet visible.

2.2 Proposed System

Here we propose to build an application to solve the problems encountered by identifying the user's gender and age through live input with access to the users camera. We then captures the image of the user and then, using our gender and age classifier, we determine whether the user must be granted access or denied from viewing the content. We thus restrict the wider domain and make it safer for individuals. The age and gender classification method deployed includes a CNN with Gabor filters. Here a Convolutional neural network (CNN) based architecture is proposed for age gender classification, where the images and their Gabor filter responses are used as the input. The architecture is trained to label the input images into 2 ranges of age (below and above 18) or as male and female. The approach shows improved accuracy in both age and gender classification compared to the state-of-the-art methodologies.

The attempt here is to improve the performance by adding Gabor filter to the input images in order to use some hand-crafted features in addition to the image itself as in the conventional CNN approaches. Specifically, we first extract Gabor filter responses of the input image, and then extract some features from them. Gabor filter, which is a linear filter used for texture analysis, analyzes whether there are any specific frequency content in the image in specific directions in a localized region around the point or region of analysis.

The network[1] is simple (having only 5 layers: 3 convolutional and 2 fully connected layers), and yet achieves a recognizable improvement in performance compared to previous non-CNN based approaches. Here the dataset used is Adience and UTKFace dataset. Unlike other age/gender datasets, the Adience dataset pictures are mainly in wild covering different posses, races, lightning and etc. Our system is expected to improve age and gender accuracy compared to the state-of-the art methods. We believe that the advantage of our scheme is to let the network to focus on useful features, which could improve the performance. A disadvantage of this system is that it cannot automatically learn the features from the massive training data as in the case of deep CNN which does not require Gabor filter extracts in the input.

Chapter 3

Software Requirement Specification

3.1 Introduction

3.1.1 Purpose

This SRS describes the software functional and non-functional requirements for Video Content Restriction using Gender and Age Classification. It blocks age-restricted videos by classifying the age and gender of the user in realtime. The scope of this SRS is limited to only the classification of the age and gender of the users without including the classification of videos as age-restricted or not.

3.1.2 Document Conventions

Table 3.1: Document Conventions

Abbreviation	Expansion
CNN	Convolutional Neural Network
SRS	Software Requirement Specification
FRS	Functional Requirements Specification
HCI	Human Computer Interface
DNN	Deep Neural Network
ANN	Artificial Neural Network
SDK	Software Development Kit
NDK	Native Developement Kit

.

3.1.3 Intended Audience and Reading Suggestions

This document is intended to be used by members of the project team that will implement and verify the correct functioning of the system. The rest of this SRS consists of an overall description, external interface requirements, hardware and software requirements, functional requirements and finally other requirements. The readers are suggested to read in the same order.

3.1.4 Project Scope

Video Content Restriction using Gender and Age Classification is an android application that can classify the age and gender of the user in real-time and restrict the video if necessary. It uses external data for determining whether the video has age-restricted content or not. It is developed for parents to prevent under-age children from viewing age-inappropriate videos. It is a light weight application that is activated only when an age-restricted content is played.

3.1.5 Overview of Developer's Responsibilities

The developer has to implement an android application that is light enough to be supported in all android phones and can run without consuming too much battery. The application must be able to classify the age and gender of the user in real-time without delay and with reasonable accuracy. The application can be kept running always and it will be able to check the users age and gender to restrict playback of the content if there is age-restricted content in the video.

3.2 Overall Description

3.2.1 Product Perspective

Video Content Restriction using Gender and Age Classification is a new system that is used as a parental control system to restrict age-restricted videos for under-age children. It is meant as an additional security over the existing age verification using account information which can be falsified. This system incorporates an android application that uses external data to classify videos as age-restricted or not and then classifies the user in real-time using gender and age classification.

3.2.2 Product Functions

FE-1: Capture user image in real time.
FE-2: User age and gender classification.
FE-3: Video blocking.
FE-4: Log records attempts to watch restricted content with date time and video name.

3.2.3 User Classes and Characteristics

Parents	Parents enable this application on the phone to block their childrens access to age-restricted content. Only an adult viewer (parent) can clear the log which records the information related to attempts to view restricted content.
Under-age children	Children will be blocked from accessing age-restricted video content by this application. They will be shown a restricted content alert on the phone and will immediately be backed out of the video. will not be able to clear logs.

3.2.4 Operating Environment

The system is an android application and thereby is supported only on android devices. It requires minimum SDK version of 18 along with a good internet connection. It must be able to get data from external sources to check whether a video is age-restricted or not. The hardware must be computationally powerful enough to classify the age and gender of the user in real-time without delay.

3.2.5 Design and Implementation Constraints

A hardware limitation is that the device on which the application is to run should have a well functional front camera with needed resolution.
Access to the users camera has to be requested and only on acquiring the user camera permission will the intended result be obtained.
Further the accuracy of the designed system is not completely 100% and thus there may be ambiguities related to the same.

3.2.6 User Documentation

The need for restricting videos online : https://support.google.com/youtube/answer/2802167?hl=en

3.2.7 General Constraints

Battery usage is limited as far as possible by activating the user camera only when restricted videos are requested.

3.2.8 Assumptions and Dependencies

AS1	The user has granted access to the devices camera at the time the application is installed.
AS2	The accuracy of the age estimation algorithm is assumed to be almost precise enough to categorize the age groups and to atleast precisely identify minor users distinctly.
DP1	Dataset related to the type of videos to be restricted should be available.
DP2	The speed with which access is granted depends on the processing power of the mobile hardware.

3.3 External Interface Requirements

3.3.1 User Interfaces

Front camera acts as the first and foremost interface between the software and the user. The android application has videos tagged by youtube as having violent content blocked by default. To view such videos a user must enable age restricted content by capturing image of the user with the camera. In order to access the camera in the android system, confirmation must be taken from the user before hand. So a pop up window could be used to satisfy the need.

3.3.2 Hardware Interfaces

The available operating system need to interacts with the graphics card necessarily of Nvidia Geforce 1060 or above.

3.3.3 Software Interfaces

The application communicates with the camera module to get the image of the user in real time and classify the gender and age. Based on the obtained result access to view the video is either granted or denied. A log is maintained to record the details of accesses to age restricted video content. It lists at what time, date and name of the video which was accessed.

3.3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating system for the mobile application.

3.4 Hardware and Software Requirements

3.4.1 Hardware Requirements

- CPU - Quad core or above.
- Ram - 8GB or more.
- Storage - SSD with at least 256GB of storage.
- Graphics Card - GTX 1060 6GB or above.
- Front camera of sufficient quality.
- Android - Snapdragon 835 and above.

3.4.2 Software Requirements

- Keras with TensorFlow backend
- OpenCV
- Python
- Ubuntu
- Android SDK 18 and above
- Flask

3.5 Functional Requirements

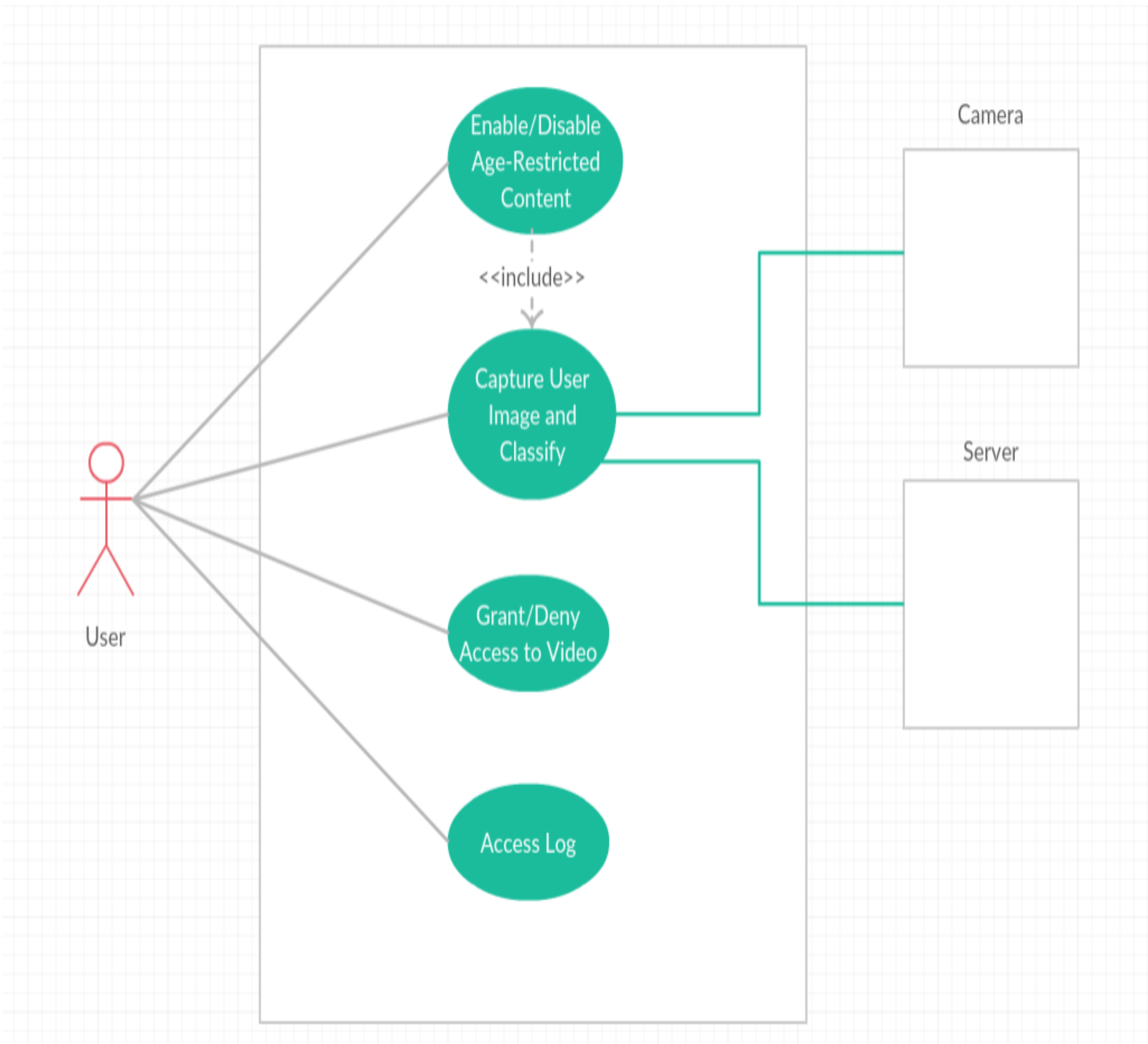


Figure 3.1: Use Case Diagram

3.5.1 Log access

Input: Request for log access.
Process: The log can be viewed by any user or cleared by a user if the toggle is enabled.
Output: Access to clear is granted or blocked. Access to view log is given.

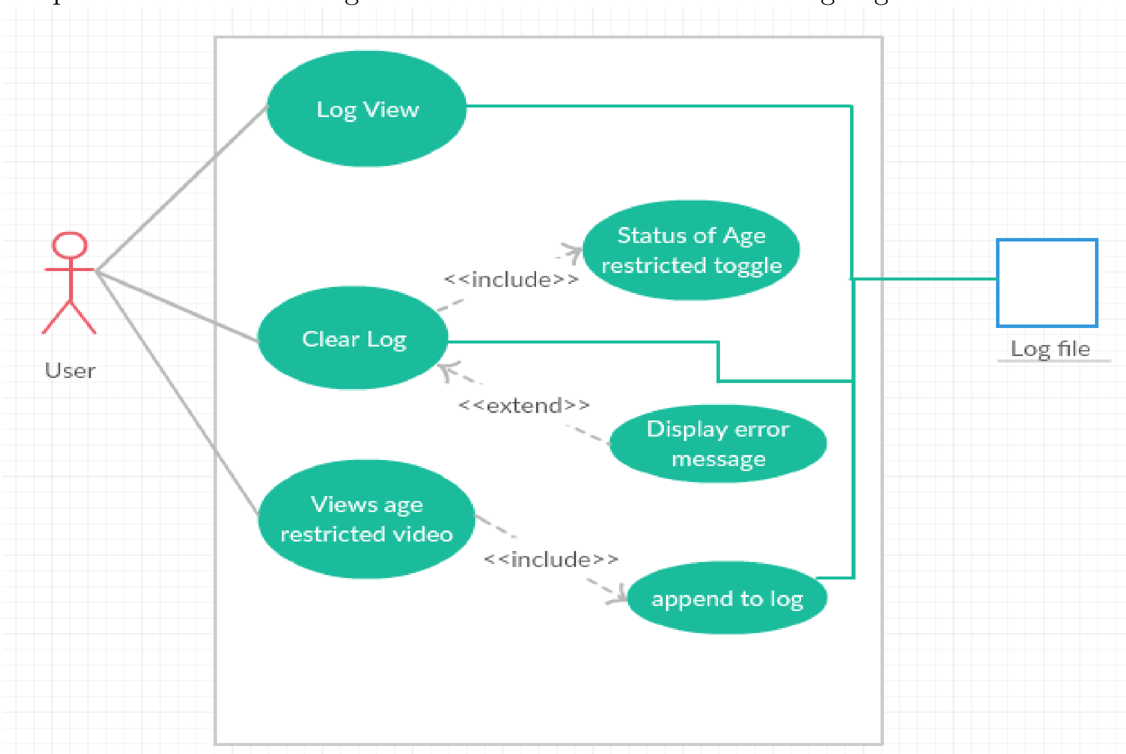


Figure 3.2: Log access and Log Clearing

3.5.2 Video requesting and content-type identification

Input: The user tries to access a video.
Process: If the video is tagged to have violent content and age restricted content tab is disabled the video is blocked or prevented from playing. Else access to the video is granted.
Output: The result is returned.

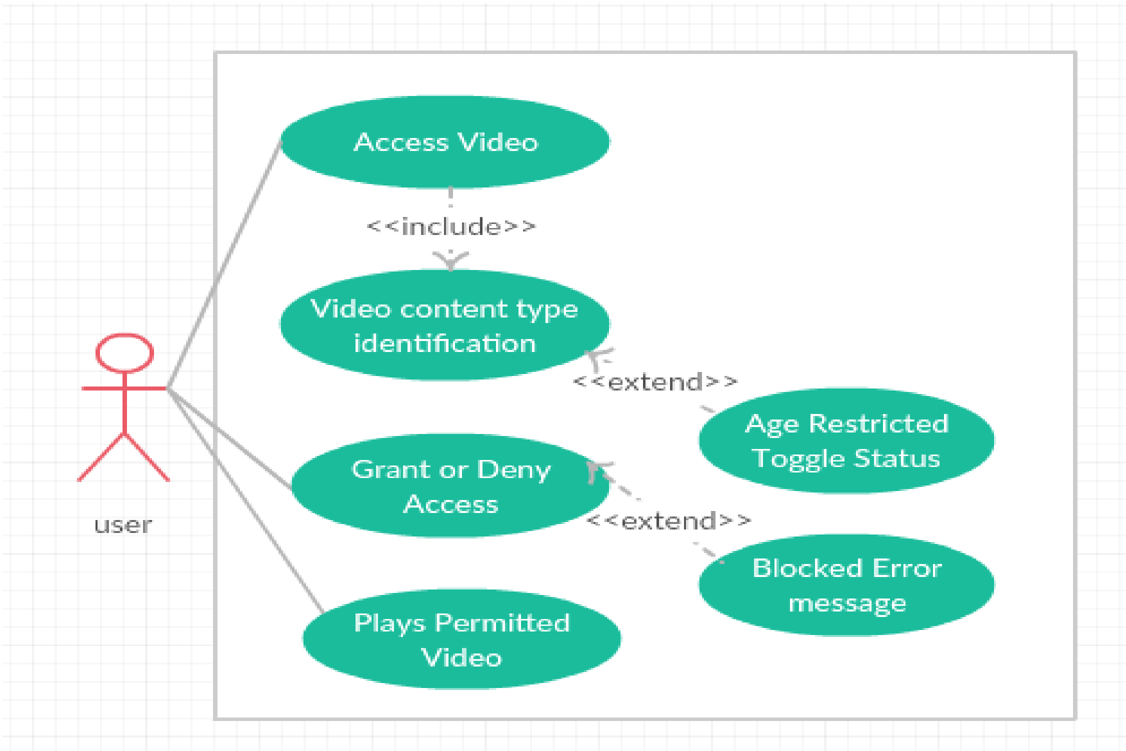


Figure 3.3: Video requesting and identification

3.5.3 Enabling age restricted content tab

Input: User tries to enable the toggle of age restricted content
Process: The front camera is accessed to get the image of the user which is then classified to check whether the user is under-age or not. If the user is a minor, the toggle is disabled when submitted else the toggle is enabled.
Output: Age restricted content toggle is enabled or disabled

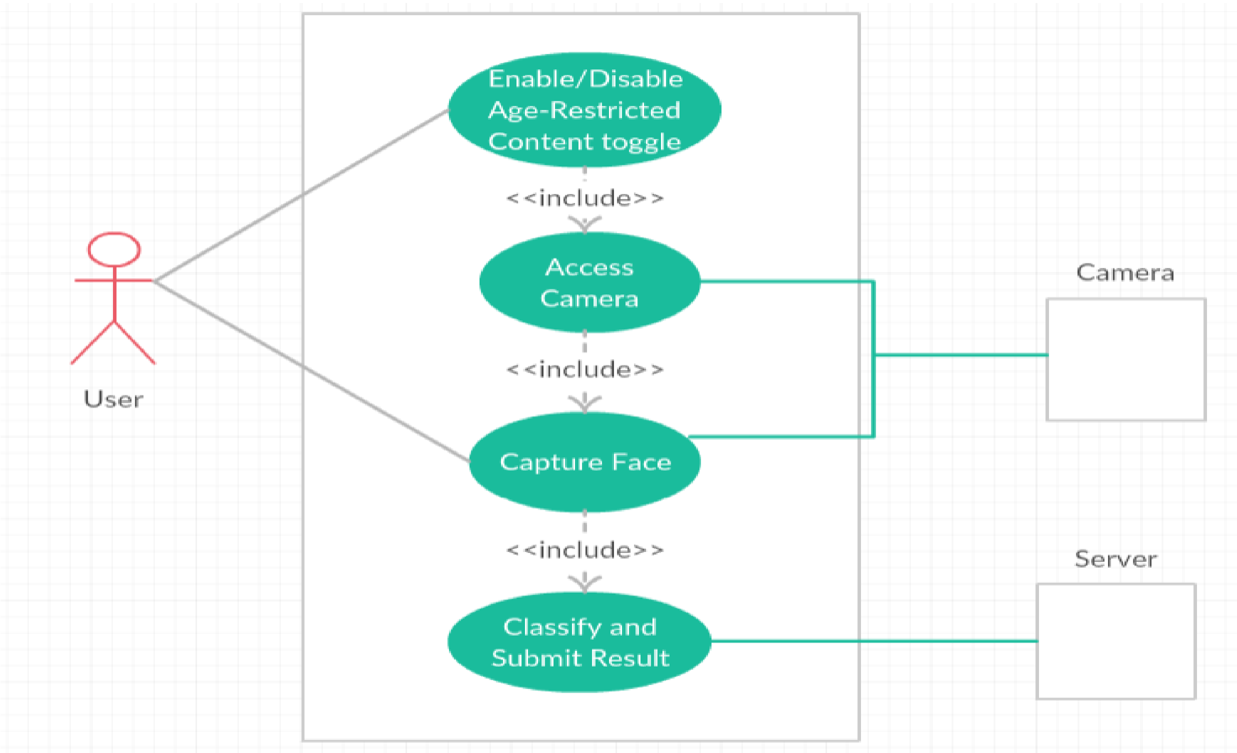


Figure 3.4: Enabling age restricted content button

3.6 Non-functional Requirements

3.6.1 Performance Requirements

Video content is restricted using Gender and Age Classification, if necessary. It prevents under-age children from viewing age-inappropriate videos.

3.6.2 Safety Requirements

The image captured from the user's camera is not stored in the network and is discarded immediately.

3.6.3 Security Requirements

Only the parent (adult user) can clear the log records.

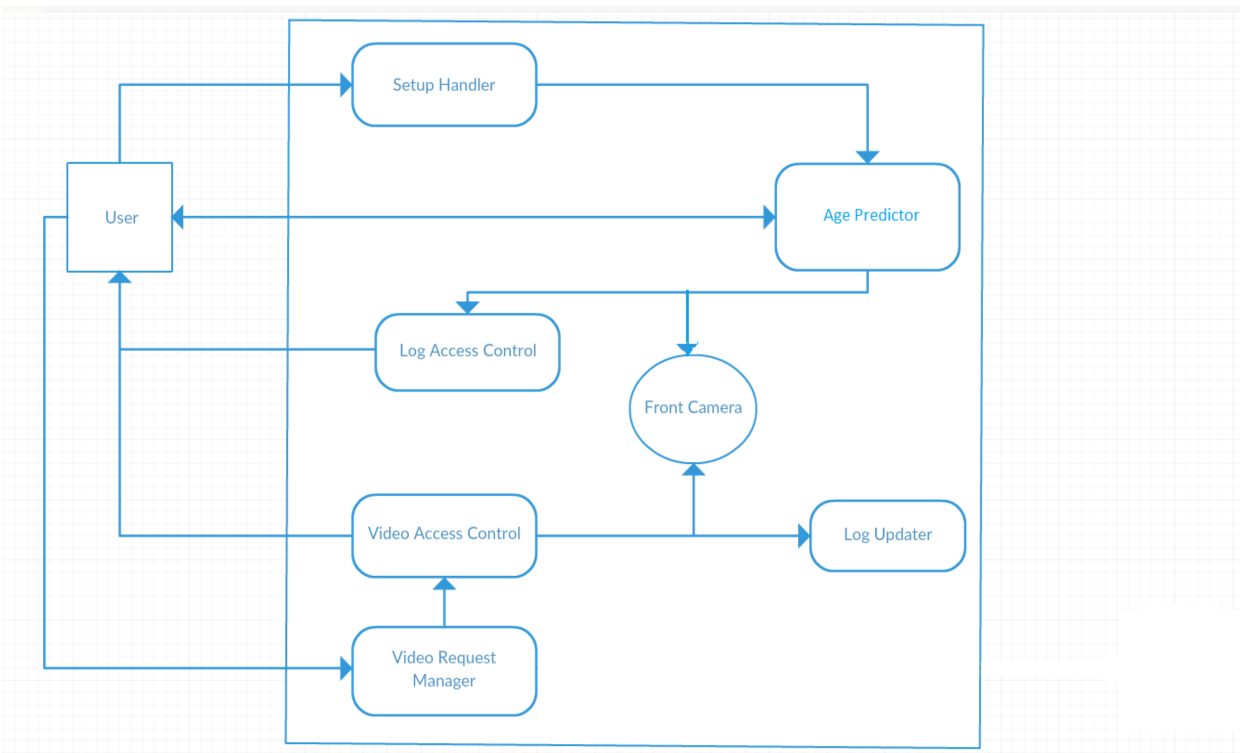
3.6.4 Software quality attributes

The UI implemented as an android application is light-weight. The application would prove to be easy to use, reliable, portable, maintainable.

Chapter 4

System Design

4.1 System Architecture



4.1.1 Log Access Control

The log can be viewed by any user and cleared only by a user who has been verified to be an adult user.

4.1.2 Log Updater

The log is updated with the recent request consisting of user details, accessed video and whether the access was granted or not.

4.1.3 User Verification

If the video is age-restricted, an image of the user is captured and age is classified. If the user is a minor, the access is restricted. Else, permitted.

4.1.4 Video Access Control

Video request manager provides the type of video accessed. Depending on whether the video accessed is age-restricted or not, the user access is permitted or denied.

4.2 Input Design

Image captured from the camera is given as input to the model. The resolution and size of the image is adjusted and further face cropping is performed to extract only the essential features.

4.3 Database Design

No database is used to store data.

4.4 Libraries and Packages Used

4.4.1 Tensorflow

TensorFlow is used for machine learning applications and is an end to end open source platform. It provides comprehensive and flexible ecosystem of tools. It also provides resources to easily build and deploy machine learning applications. It has a set of high level APIs which makes for easy debugging and immediate model iterations.

4.4.2 OpenCV

It is a library of programming functions for real time computer vision. It includes 2D and 3D feature toolkit, ego motion estimation, facial recognition system, gesture recognition, HCI, mobile robotics, motion understanding, object identification, segmentation and recognition, augmented reality and many more. It has a machine learning library that supports decision tree learning, expectation maximization algorithm, DNN. OpenCV has its primary interface in c++.

4.4.3 Numpy

It is an open source library available in Python with a high performance multidimensional arrays and matrices. It works well for matrices multiplication and multidimensional arrays. It can be integrated to c/c++ and Fortran for different applications. It can handle large amount of data than any other library and is highly memory efficient. It supports large number of mathematical operations for multi-dimensional arrays and matrices.

4.4.4 Keras

It is an open-source neural-network library used in python. It is capable of running on top of TensorFlow. It minimises the number of user actions and provides clear and actionable feedback in case of user errors. It is user-friendly and extensible. It offers consitent and simple APIs. Keras has three backend implementations. They are: the TensorFlow backend, the Theano backend and the CNTK backend. Theano is a symbolic tensor manipulation framework and CNTK is an open source toolkit for deep learning developed and Microsoft.

4.4.5 PrettyTable

It is a python library for generating simple ASCII tables. Using this the desired columns and rows to be displayed in the final output can also be chosen. PrettyTable can read data from HTML, CSV and output data in ASCII or HTML. pip3 tool is used to install PrettyTable. add_row() or add_column() methods are used to create table.

4.4.6 OkHttp3

It is a third- party library used for sending and receiving HTTP-based network requests. It is more efficient in reading and writing data than the standard Java I/O libraries by creating a shared memory pool. It supports synchronous and asynchronous calls. Synchronous calls does not support cancelling a request. Asynchronous calls supports native cancelling and cancelling with a single method call.

4.4.7 Newpipe

It has a lightweight YouTube front end used without the proprietary YouTube-API or any Google's play-services. Its features includes: searching videos, display general information about a video, download requested videos, listen to YouTube videos, search channels, watch videos from a channel, view history, subscribe to channels,queuing videos, local playlists, subtitles, watch/block age restricted material, show next/related videos.

4.5 Module Description

4.5.1 Generating the input for neural network

Generating TF-Record Data:

Data from Adience and UTKFace are read, then face is cropped from the images using OpenCV. The Metadata with image filename, age label, gender label is obtained. Age labels are converted into ranges of 0-18 or 18-100 and then saved as 0 or 1. Gender labels are male or female which are then saved as 0 or 1. This combined data is written into TF-Records with each containing data about 2000 images.

Applying Gabor filter:

Using OpenCV function generate the kernels for the gabor filter. The function cv2.getGaborKernel is used with the parameter values. Convolve each image with the filters to generate response matrices. Convert response matrix of each image to a feature vector which includes local energy and mean amplitude. This is also written along with image data into the Tfrecords.

4.5.2 Create Neural Network

Create training and testing dataset. Read age, gender, image and feature set from both. Performed the necessary pre-processing on the input. Created 2 neural network with three convolutional layers(128*5*5), two fully connected layers, dropout layer and softmax layer with inputs as gabor feature set and image.
2 checkpoints are used : EarlyStopping and ModelCheckpoint.
EarlyStopping is used to stop the training when validation loss doesnt decrease beyond a point. ModelCheckpoint is used to save the best models in h5 format with the best validation accuracy.

4.5.3 Video Player

An android based video player application, currently extended from NewPipe Application. The app provides options to get age restriction on videos, Such videos are initially blocked. An option in the settings page is used to enable age-restricted content. This toggle passes control to the Capture Face app which returns the age and gender of the user. If the user is above 18, he/she is allowed to view age-restricted content.
When such videos are opened, a log is written into the log file detailing the video opened, date, time and the access status. This log can be viewed from the settings through the Log app. The logs can also be cleared if the user is above 18 by using the value of the age-restricted content toggle.

4.5.4 User Classification

This is done using the Camera activity. The activity receives a request from the main activity to get the users age and gender. It captures the users image and then sends it over the network to a remote server. The server is setup using Flask and accepts http requests. The image received is pre-processed and then provided as input to the 2 neural networks. The resulting classification is then returned back to the app.

4.5.5 Log Management

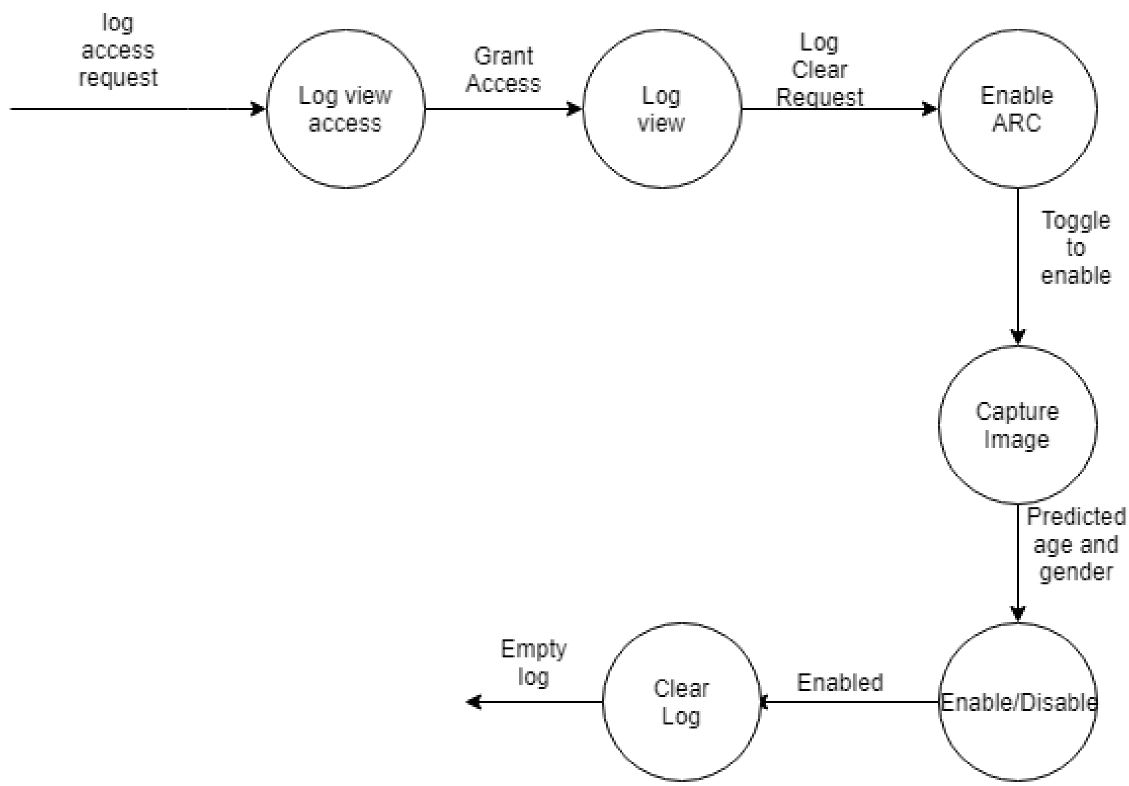
This is managed using the Log activity. It is a simple scrollable textbox app that displays the log generated.

Chapter 5

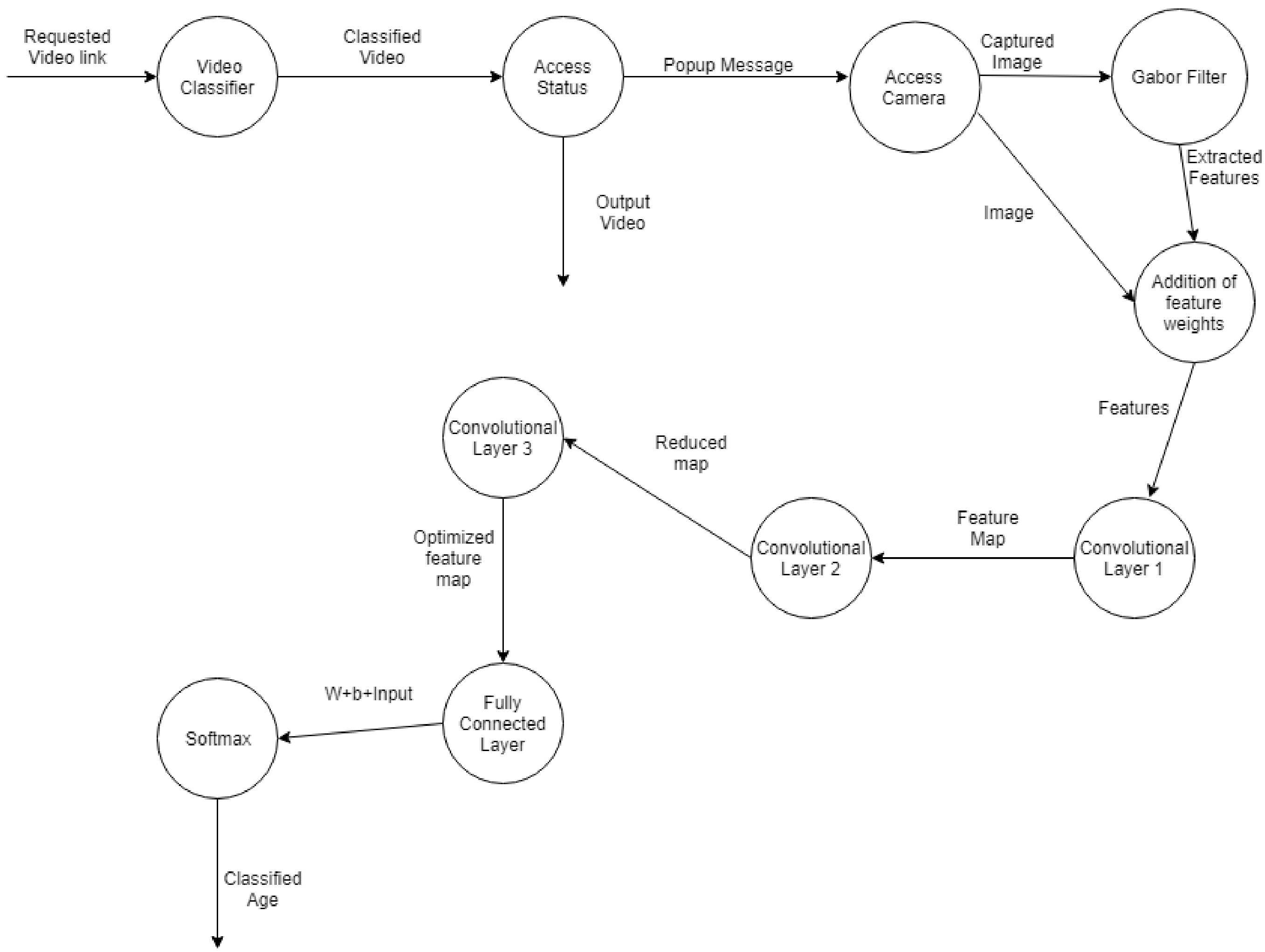
Data Flow Diagram

5.0.1 Level 0 DFD

5.0.2 Level 1 DFD



5.0.3 Level 2 DFD



Chapter 6

Implementation

6.1 Algorithms

6.1.1 Data Pre-processing

Input: Adience and UTKFace Dataset
Output: TfRecords

1. The faces from the Adience and UTKFace dataset are first extracted and then resized to 100x100.

2. Feature extraction using Gabor filter provided by OpenCV as a function is applied to all the images.

3. The responses are converted to local energy and mean amplitude which are stored as a single list.

4. The image, gabor feature set and labels are then written into TfRecords.

5. Each record contains about 2000 image data.

6.1.2 Training Algorithm

Input: TfRecords
Output: Neural network models for age and gender classification

1. Data is read from the TfRecords into the appropriate format as Numpy stacks.

2. The 2 networks are built using functional API.

3. There are 5 layers: 3 Convolutional and 2 Dense Layers.

4. Input to the 2 networks are the images, their Gabor responses and labels.

5. The batch size is set to 500 and trained for 1000 epochs.

6. Model Checkpoint and Early Stopping is used to get the model with the best validation accuracy and least validation loss.

6.1.3 Age and Gender Classifier Server

Input: User Image
Output: Age and Gender Classification label

1. The image received from the app is first face cropped and then scaled to 100x100.
2. It is then classified using the trained neural network models.
3. The age and gender result is then passed back to the app.

6.1.4 User Image Capture and classification

Input: Request for image of the user
Output: Age and Gender data

1. The image from the user is captured.
2. It is then sent to the Flask server.
3. Based on the output from the server, age and gender values are updated.

6.1.5 Log

Input: Video Details and Access Permission
Output: Updated Log

1. If the video accessed contains age restricted content, then the log file is modified.
2. The video name, date, time and access status is appended to the log file.

6.2 Development Tools

6.2.1 Android Studio

Android Studio is the official integrated development environment for Google's Android operating system. It is available for download on Windows, macOS and Linux based operating systems. An android based video player NewPipe was extended to build our application. Camera and Log activities where added to this.

6.2.2 Google Cloud Compute

Google Compute Engine is the Infrastructure as a Service component of Google Cloud Platform which is built on the global infrastructure that runs Google's search engine, Gmail, YouTube and other services. Google Compute Engine enables users to launch virtual machines on demand. GCS was used to train the neural network using GPU. It is also used to host the classifier server using Flask.

6.2.3 Ubuntu Terminal

Ubuntu is a Linux based operating system. Terminal is a command-line interpreter or shell that provides a command line user interface for Unix-like operating systems.

6.2.4 Android Phone

Smartphones are a class of mobile phones and of multi-purpose mobile computing devices. They are distinguished from feature phones by their stronger hardware capabilities and extensive mobile operating systems, which facilitate wider software, internet and multimedia functionality alongside core phone functions such as voice calls and text messaging. Essential for testing and debugging the android applications being developed.

6.2.5 Flask

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. A flask server was used to run the classifier on google cloud compute.

Chapter 7

Testing

7.1 Testing Methodologies

Software Testing Methodology is defined as strategies and testing types used to verify that the application under test meets user expectations. Test Methodologies include functional and non-functional testing to validate the Application Under Test. Each testing methodology has a defined test objective, test strategy and deliverables. We performed the following testing: Unit testing,Integration testing, System testing, Performance testing.

7.2 Unit Testing

Unit testing finds problems early in the development cycle. This includes both bugs in the programmers implementation and flaws or missing parts of the specification for the unit. The process of writing a thorough set of tests forces the author to think through inputs, outputs, and error conditions, and thus more crisply define the units desired behaviour. The cost of finding a bug before coding begins or when the code is first written is considerably lower than the cost of detecting, identifying and correcting the bug later; bugs may also cause problems for the end-users of the software. We did unit testing on the following modules:-

1. Neural Network Model Testing- The gender and age classification model that classifies a user as male or female, and age as either an adult or an underage user was tested using around 90 images labelled manually. The test results are as shown and we obtained an accuracy of 86.67% for age, and 77.78% for gender.

2. Age detection - Test on an image given as input and then from image capture via in-live video feed from the camera and predicted the age of the user.

Predicted Gender	Actual Gender	Filename
f	f	10_1_1_.jpg
f	f	11_1_1_.jpg
f	f	12_1_1_.jpg
f	f	13_1_1_.jpg
m	f	14_1_1_.jpg
m	f	15_1_1_.jpg
f	f	16_1_1_.jpg
f	f	17_1_1_.jpg
f	f	18_0_1_.jpg
f	f	19_0_1_.jpg
f	f	1_1_1_.jpg
f	f	20_1_1_.jpg
f	f	21_1_1_.jpg
m	f	22_1_1_.jpg
f	f	23_1_1_.jpg
f	f	26_1_1_.jpg
f	f	27_1_1_.jpg
f	f	28_1_1_.jpg
f	f	29_1_1_.jpg
f	f	2_1_1_.jpg
f	f	30_1_1_.jpg
m	f	31_1_1_.jpg
f	f	32_1_1_.jpg
m	f	33_1_1_.jpg
f	f	34_1_1_.jpg
f	f	35_1_1_.jpg
f	f	36_1_1_.jpg
f	f	37_1_1_.jpg
f	f	38_1_1_.jpg
f	f	39_1_1_.jpg
f	f	3_1_1_.jpg
f	f	40_1_1_.jpg
f	f	41_1_1_.jpg
f	f	42_1_1_.jpg
f	f	43_1_1_.jpg
m	m	44_1_0_.jpg
m	m	45_1_0_.jpg
m	m	46_1_0_.jpg
m	m	47_1_0_.jpg
m	m	48_1_0_.jpg
m	m	49_1_0_.jpg
f	f	4_1_1_.jpg
f	m	50_1_0_.jpg
f	m	51_1_0_.jpg
m	m	52_1_0_.jpg
m	m	53_1_0_.jpg
f	f	54_1_1_.jpg

Figure 7.2: Result -Gender Classification

Using TensorFlow backend.

Predicted Age	Actual Age	Filename
(19,100)	(19,100)	10_1_1_.jpg
(19,100)	(19,100)	11_1_1_.jpg
(19,100)	(19,100)	12_1_1_.jpg
(19,100)	(19,100)	13_1_1_.jpg
(19,100)	(19,100)	14_1_1_.jpg
(19,100)	(19,100)	15_1_1_.jpg
(19,100)	(19,100)	16_1_1_.jpg
(19,100)	(19,100)	17_1_1_.jpg
(19,100)	(0,18)	18_0_1_.jpg
(19,100)	(0,18)	19_0_1_.jpg
(19,100)	(19,100)	1_1_1_.jpg
(19,100)	(19,100)	20_1_1_.jpg
(19,100)	(19,100)	21_1_1_.jpg
(19,100)	(19,100)	22_1_1_.jpg
(19,100)	(19,100)	23_1_1_.jpg
(19,100)	(19,100)	26_1_1_.jpg
(19,100)	(19,100)	27_1_1_.jpg
(19,100)	(19,100)	28_1_1_.jpg
(19,100)	(19,100)	29_1_1_.jpg
(19,100)	(19,100)	2_1_1_.jpg
(19,100)	(19,100)	30_1_1_.jpg
(19,100)	(19,100)	31_1_1_.jpg
(19,100)	(19,100)	32_1_1_.jpg
(19,100)	(19,100)	33_1_1_.jpg
(19,100)	(19,100)	34_1_1_.jpg
(19,100)	(19,100)	35_1_1_.jpg
(19,100)	(19,100)	36_1_1_.jpg
(19,100)	(19,100)	37_1_1_.jpg
(19,100)	(19,100)	38_1_1_.jpg
(19,100)	(19,100)	39_1_1_.jpg
(0,18)	(19,100)	3_1_1_.jpg
(19,100)	(19,100)	40_1_1_.jpg
(0,18)	(19,100)	41_1_1_.jpg
(0,18)	(19,100)	42_1_1_.jpg
(19,100)	(19,100)	43_1_1_.jpg
(19,100)	(19,100)	44_1_0_.jpg
(19,100)	(19,100)	45_1_0_.jpg
(19,100)	(19,100)	46_1_0_.jpg
(19,100)	(19,100)	47_1_0_.jpg
(19,100)	(19,100)	48_1_0_.jpg
(19,100)	(19,100)	49_1_0_.jpg
(19,100)	(19,100)	4_1_1_.jpg
(19,100)	(19,100)	50_1_0_.jpg
(19,100)	(19,100)	51_1_0_.jpg
(19,100)	(19,100)	52_1_0_.jpg
(19,100)	(19,100)	53_1_0_.jpg

Figure 7.3: Result- Age Classification

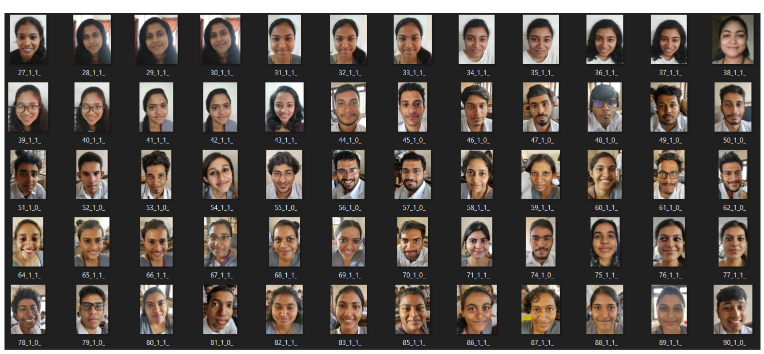


Figure 7.1: Test Dataset

Figure 7.4: Image Classified - Adult Male

Figure 7.5: Image Classified - Underage Female

Figure 7.6: Image Classified - Underage Male

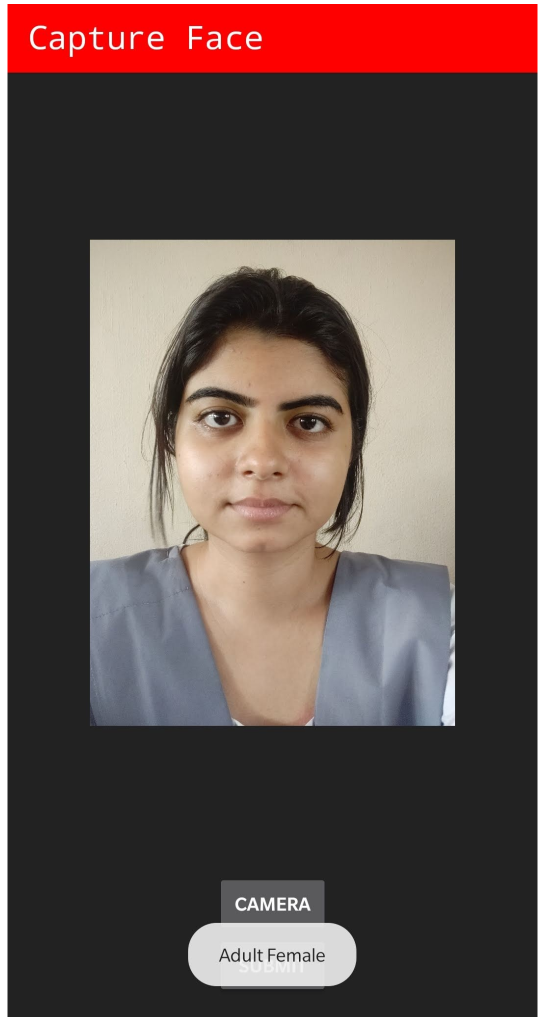


Figure 7.7: Age and Gender Detected

3. Updating log file - Test updation of date, time, name of accessed video and information of whether the video access was granted or denied in the log file.

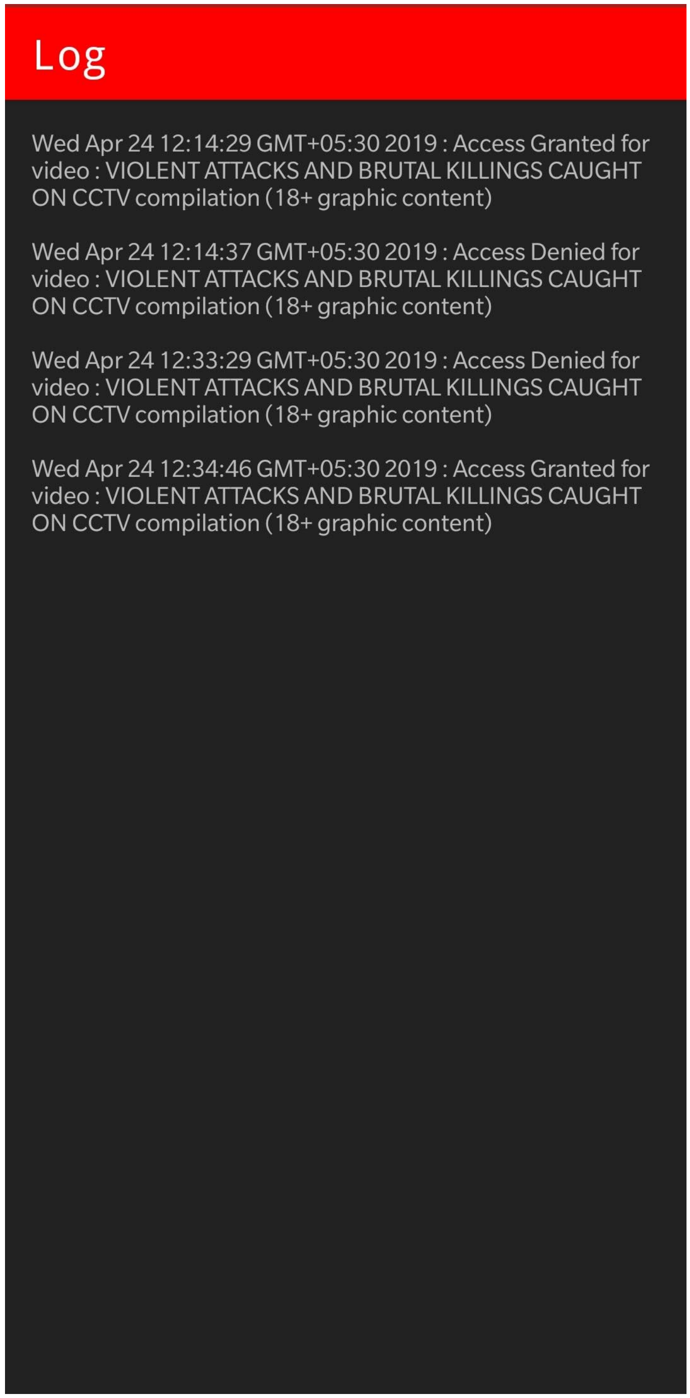


Figure 7.8: Log for Age restricted videos

4. Video Blocking- Tested automated video blocking on a test video if the user is not recognized as an adult and before enabling the age restricted content toggle button.

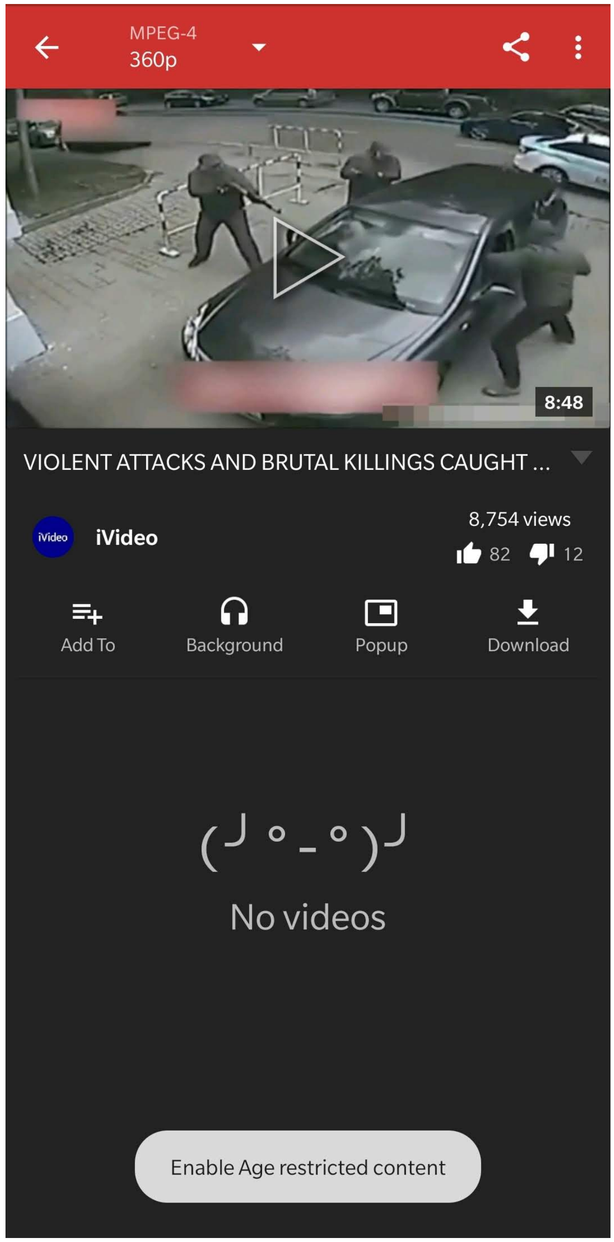


Figure 7.9: Playing 18+ video without enabling Age restricted content

7.3 Integration Testing

The purpose of integration testing is to verify functional performance and reliability requirements placed on major design items. These design items, i.e., assemblages (or group of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly. We integrated our modules one by one and tested the partially integrated system.

7.4 System Testing

System testing is performed on the entire system in the context of Functional Requirement Specifications (FRS) and/or System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond what is defined in the software/hardware requirements specification(s).
All modules were integrated at the end of integration testing and the entire system was tested to check whether the video was blocked when the age restricted content toggle was off. The efficiency of the model was tested with the images being captured and classified. The video access was granted once the toggle button was enabled.

Chapter 8

Graphical User Interface

8.1 GUI Overview

We have a graphical user interface which is an android application. It is video player application extended from NewPipe. NewPipe is a simple video player application.It has been extended to include custom activities to classify user face and log age restricted videos.

8.2 Main GUI Components

8.2.1 Video Player

The application opens with an interface similar to any of the existing video player application. It lists the most popular or trending videos on the main opening page. It also includes a search bar for the user to search a particular video as per need. The search bar shows suggestions based on the previous searches and relevant searches. The opening page has a swipe screen to display a list of subscriptions and further swiped to list the bookmarked videos.

8.2.2 Image Capture

The home page has a option to list settings download and history. The settings option further include a content tab with the Age Restricted Content toggle button. On enabling the toggle button a call to another activity to capture the image is initiated. This image is captured and posted to the server run on google cloud via the http request method. The classification model running on the server classifies the image and returns the result which appears as a toast in our application. This result can be submitted to enable or disable the toggle button.

8.2.3 Log Generation

The setting tab also contains an option to view and clear the Log being maintained. This log is locally maintained on the device. It records information regarding the date and time of access, if the access was granted or denied, and the name of the video being requested.

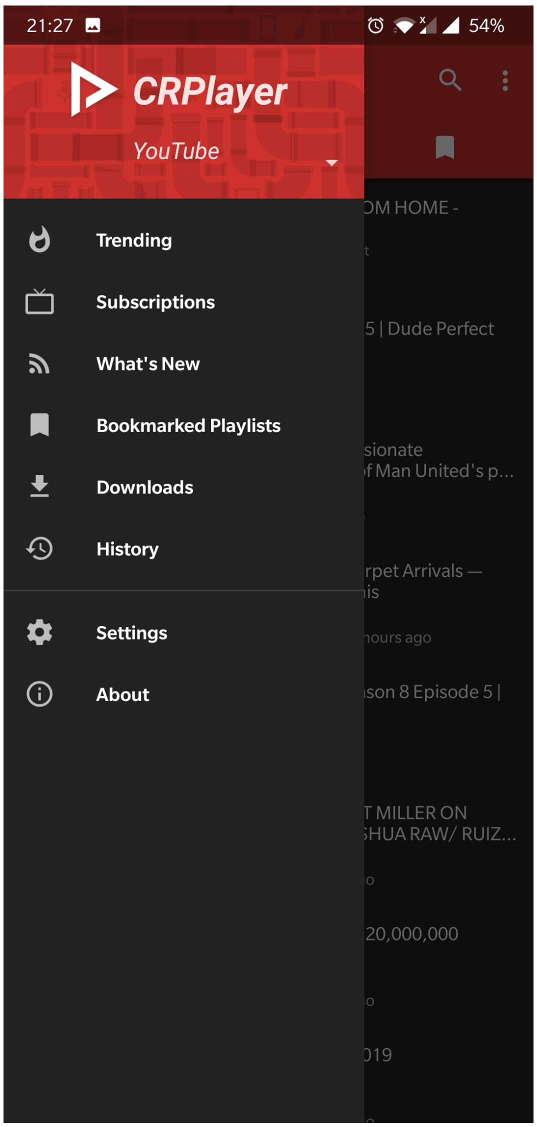


Figure 8.1: Menu

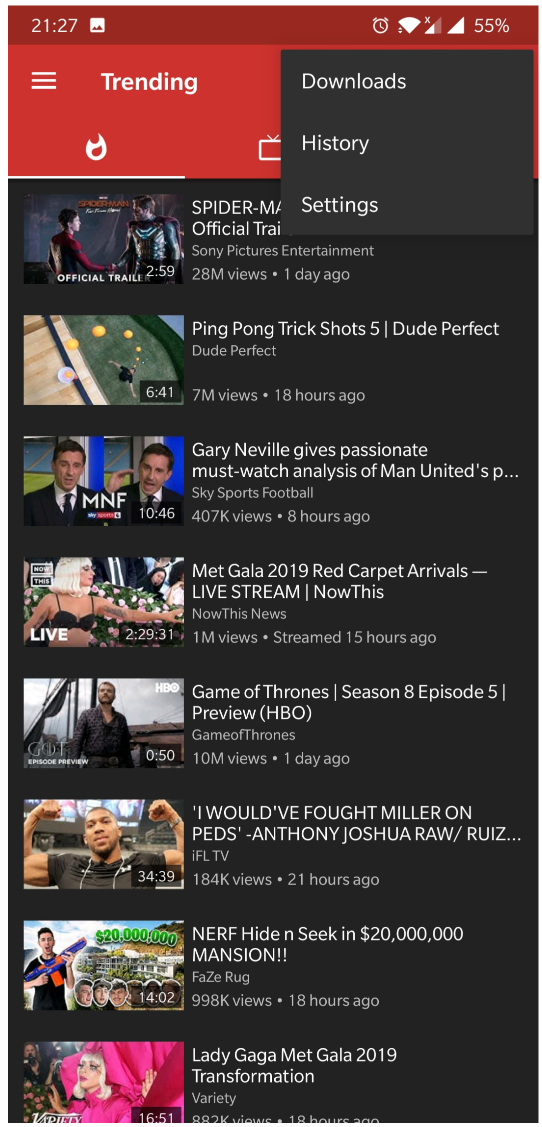


Figure 8.2: Video Player

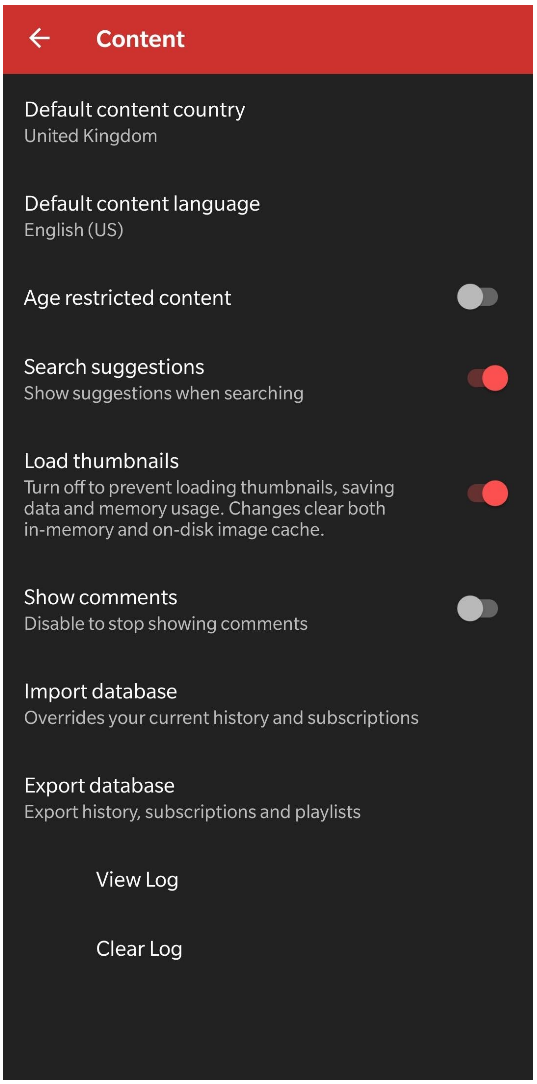


Figure 8.3: Content Settings with age restricted toggle button

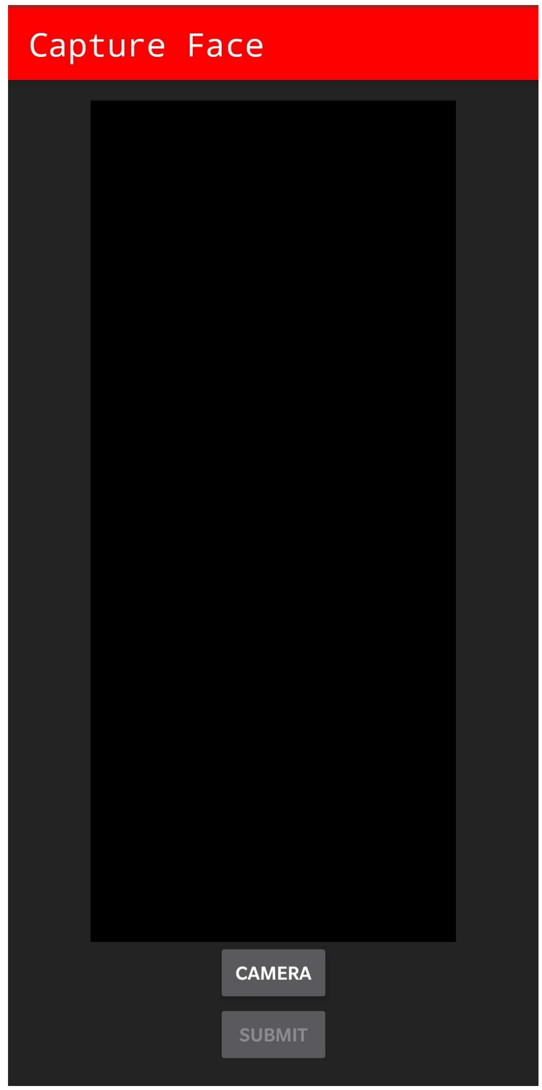


Figure 8.4: Face Capture Activity

The log only records the details when the user tries to play a blocked video. The log can be viewed by any user however it can only be cleared by an adult user, i.e, when the age restricted content button is enabled. If this toggle button is disabled and the user tries to clear the log a toast with the corresponding error message is shown.

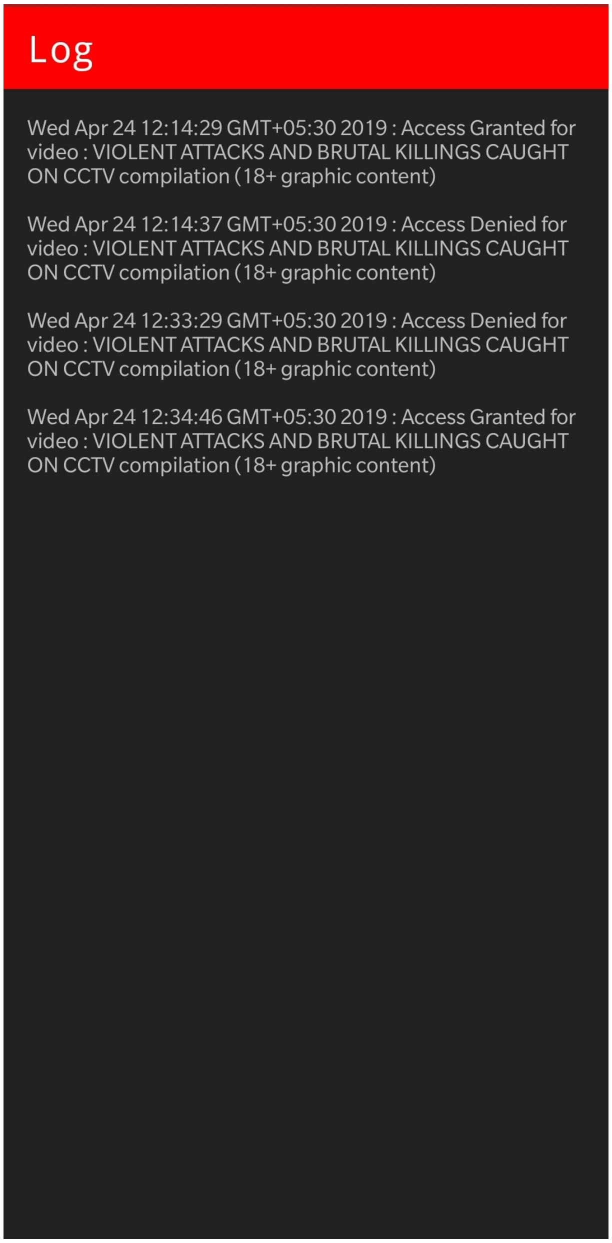


Figure 8.5: Log for Age restricted videos

Chapter 9

Results

CR Video Player application is aimed to ensures safe browsing for kids over the internet which nowadays play a major role in providing misleading content to seize young minds. Present video players have a restricted mode option that can easily be enabled or disabled by any user. We try to restrict the users who can disable this mode by incorporating a camera activity that recognizes their gender and age. Recognition is performed by creating a CNN model with a cropped face and its Gabor filter responses as input. The advantage of this proposed scheme is to let the network focus on useful features, which could improve the performance.

9.1 Gender and Age Estimation Neural Network

Images from the Adience and UTKFace dataset is cropped to get only the face. This is done to reduce the miss classification that may occur with different backgrounds. This image filename with the age label and gender label is written into TFRecords. Each record contains upto 2000 images. TFRecord also contains the feature vector obtained from response matrix of each image. The response matrix is generated by applying gabor filter to the images. Gabor filters are used in feature extraction for texture analysis. Te gabor kernel size is defined as 31, with each kernel defined with parameters sigma value 2.0, theta varies from 0 to π through $\pi/6$ rotation, lambda 2.0 and gamma 0.3, and phase offset from 0 to 2π through π.

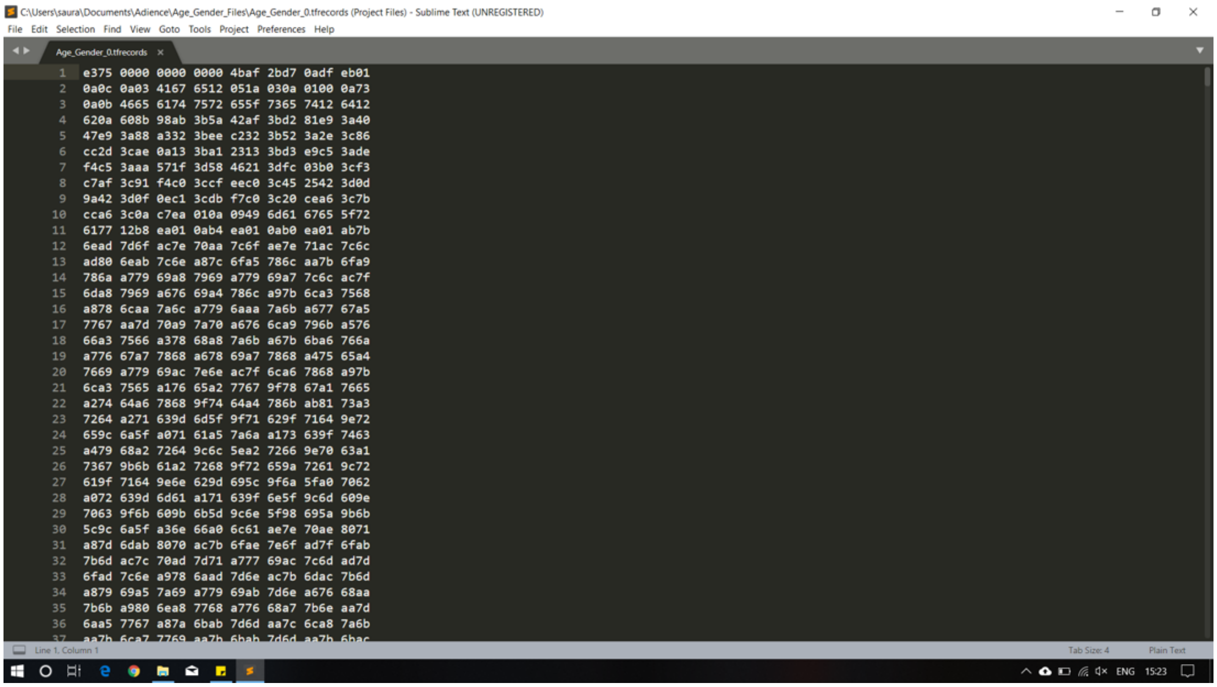


Figure 9.1: Content of TF Records

The model is created using the Functional API in Keras. The network with three convolutional layer, two fully connected layer, dropout layer and softmax layer with inputs as gabor feature set and image is created. Two checkpoints are used: Early Stopping and Model Checkpoint. Early Stopping is used to stop the training when validation loss does not decrease beyond a point. Model Checkpoint is used to save the best model in h5 format with the best validation accuracy. An accuracy of 82.5% was obtained when tested on a set of 90 images manually picked.

```
Epoch 00145: val_acc did not improve from 0.95456
Epoch 146/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0140 - acc: 0.9956 - val_loss: 0.3121 - val_acc: 0.9467

Epoch 00146: val_acc did not improve from 0.95456
Epoch 147/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0109 - acc: 0.9966 - val_loss: 0.2952 - val_acc: 0.9385

Epoch 00147: val_acc did not improve from 0.95456
Epoch 148/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0153 - acc: 0.9950 - val_loss: 0.3422 - val_acc: 0.9497

Epoch 00148: val_acc did not improve from 0.95456
Epoch 149/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0063 - acc: 0.9986 - val_loss: 0.4141 - val_acc: 0.9527

Epoch 00149: val_acc did not improve from 0.95456
Epoch 150/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0122 - acc: 0.9956 - val_loss: 0.3592 - val_acc: 0.9530

Epoch 00150: val_acc did not improve from 0.95456
Epoch 151/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0090 - acc: 0.9972 - val_loss: 0.4067 - val_acc: 0.9549

Epoch 00151: val_acc improved from 0.95456 to 0.95488, saving model to age_model.h5
Epoch 152/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0102 - acc: 0.9967 - val_loss: 0.3808 - val_acc: 0.9514

Epoch 00152: val_acc did not improve from 0.95488
Epoch 153/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0067 - acc: 0.9981 - val_loss: 0.3610 - val_acc: 0.9465

Epoch 00153: val_acc did not improve from 0.95488
Epoch 154/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0061 - acc: 0.9981 - val_loss: 0.4354 - val_acc: 0.9525

Epoch 00154: val_acc did not improve from 0.95488
Epoch 155/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0075 - acc: 0.9975 - val_loss: 0.3567 - val_acc: 0.9511

Epoch 00155: val_acc did not improve from 0.95488
Epoch 156/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0055 - acc: 0.9986 - val_loss: 0.3984 - val_acc: 0.9477

Epoch 00156: val_acc did not improve from 0.95488
Epoch 157/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0089 - acc: 0.9972 - val_loss: 0.3515 - val_acc: 0.9447
```

Figure 9.2: Saving the best Age Model

```
Epoch 214/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0057 - acc: 0.9985 - val_loss: 0.3840 - val_acc: 0.9486

Epoch 00214: val_acc did not improve from 0.95488
Epoch 215/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0070 - acc: 0.9977 - val_loss: 0.4362 - val_acc: 0.9512

Epoch 00215: val_acc did not improve from 0.95488
Epoch 00215: early stopping
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            (None, 100, 100, 3)  0
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 94, 94, 96)   14208       input_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 46, 46, 96)   0           conv2d_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 42, 42, 256)  614656      max_pooling2d_1[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 20, 20, 256)  0           conv2d_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 18, 18, 384)  885120      max_pooling2d_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, 8, 8, 384)    0           conv2d_3[0][0]
__________________________________________________________________________________________________
input_1 (InputLayer)            (None, 24)           0
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 24576)        0           max_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          6400        input_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 256)          6291712     flatten_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 512)          0           dense_1[0][0]
                                                                 dense_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 512)          0           concatenate_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 512)          262656      dropout_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 512)          0           dense_3[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 2)            1026        dropout_2[0][0]
==================================================================================================
Total params: 8,075,778
Trainable params: 8,075,778
Non-trainable params: 0
```

Figure 9.3: Age model training stopped when reduction in validation loss is negligible

```
Epoch 00164: val_acc did not improve from 0.89535
Epoch 165/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0188 - acc: 0.9933 - val_loss: 0.6336 - val_acc: 0.8903

Epoch 00165: val_acc did not improve from 0.89535
Epoch 166/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0151 - acc: 0.9941 - val_loss: 0.6459 - val_acc: 0.8928

Epoch 00166: val_acc did not improve from 0.89535
Epoch 167/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0161 - acc: 0.9940 - val_loss: 0.6691 - val_acc: 0.8962

Epoch 00167: val_acc improved from 0.89535 to 0.89616, saving model to gender_model.h5
Epoch 168/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0131 - acc: 0.9951 - val_loss: 0.7393 - val_acc: 0.8916

Epoch 00168: val_acc did not improve from 0.89616
Epoch 169/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0146 - acc: 0.9947 - val_loss: 0.6509 - val_acc: 0.8904

Epoch 00169: val_acc did not improve from 0.89616
Epoch 170/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0188 - acc: 0.9933 - val_loss: 0.6580 - val_acc: 0.8902

Epoch 00170: val_acc did not improve from 0.89616
Epoch 171/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0137 - acc: 0.9953 - val_loss: 0.6820 - val_acc: 0.8905

Epoch 00171: val_acc did not improve from 0.89616
Epoch 172/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0149 - acc: 0.9945 - val_loss: 0.7077 - val_acc: 0.8871

Epoch 00172: val_acc did not improve from 0.89616
Epoch 173/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0150 - acc: 0.9945 - val_loss: 0.7042 - val_acc: 0.8889

Epoch 00173: val_acc did not improve from 0.89616
Epoch 174/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0132 - acc: 0.9955 - val_loss: 0.6711 - val_acc: 0.8914

Epoch 00174: val_acc did not improve from 0.89616
Epoch 175/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0144 - acc: 0.9944 - val_loss: 0.6503 - val_acc: 0.8914

Epoch 00175: val_acc did not improve from 0.89616
Epoch 176/1000
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0139 - acc: 0.9946 - val_loss: 0.7152 - val_acc: 0.8894
```

Figure 9.4: Saving the best Gender Model

```
28807/28807 [==============================] - 42s 1ms/step - loss: 0.0131 - acc: 0.9951 - val_loss: 0.7109 - val_acc: 0.8930

Epoch 00218: val_acc did not improve from 0.89616
Epoch 00218: early stopping
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            (None, 100, 100, 3)  0
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 94, 94, 96)   14208       input_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 46, 46, 96)   0           conv2d_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 42, 42, 256)  614656      max_pooling2d_1[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 20, 20, 256)  0           conv2d_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 18, 18, 384)  885120      max_pooling2d_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, 8, 8, 384)    0           conv2d_3[0][0]
__________________________________________________________________________________________________
input_1 (InputLayer)            (None, 24)           0
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 24576)        0           max_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          6400        input_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 256)          6291712     flatten_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 512)          0           dense_1[0][0]
                                                                 dense_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 512)          0           concatenate_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 512)          262656      dropout_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 512)          0           dense_3[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                 (None, 2)            1026        dropout_2[0][0]
==================================================================================================
Total params: 8,075,778
Trainable params: 8,075,778
Non-trainable params: 0
__________________________________________________________________________________________________

real    154m46.426s
user    75m10.484s
sys     26m24.036s
```

Figure 9.5: Gender model training stopped when reduction in validation loss is negligible

9.2 Application

This application is a simple video player application extended from the existing NewPipe application. On trying to view an age restricted video the user receives a toast requesting to enable "Age Restricted Content". Enable the "Age Restricted Content" toggle button from "Settings". This will correspondingly open the camera activity with a button to capture the users image. After capturing a preview is generated and a toast mentioning the classified result is displayed. This result is submitted by the user and accordingly the toggle is enabled or disabled. This toggle is enabled only if the user is recognised as an adult irrespective of the gender and it is disabled if the user is underage. This method of estimating the age of the user in real time proves more effective than the user manually signing up with the age he provides which can easily be falsified.

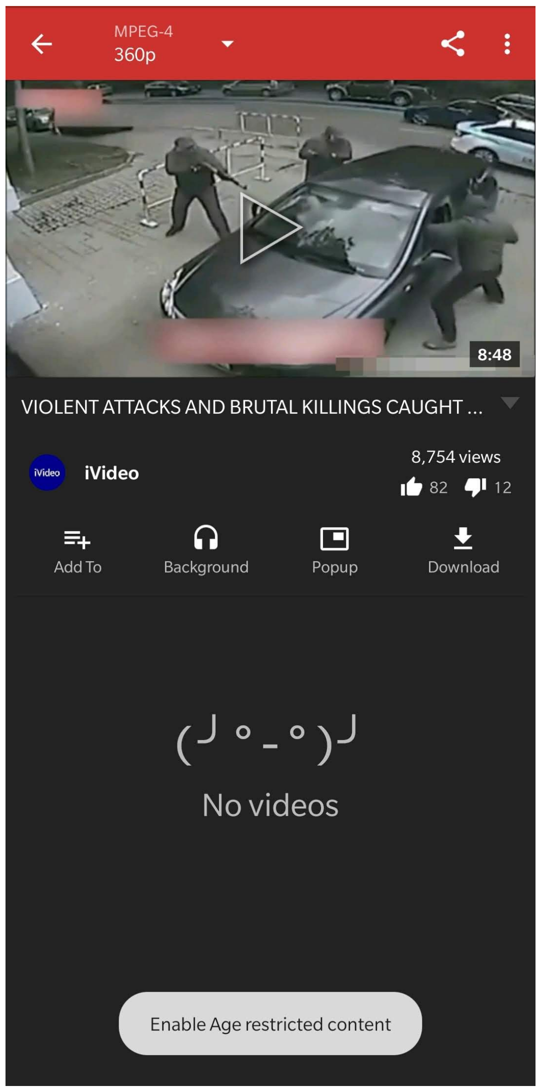


Figure 9.6: Playing 18+ video without enabling Age restricted content

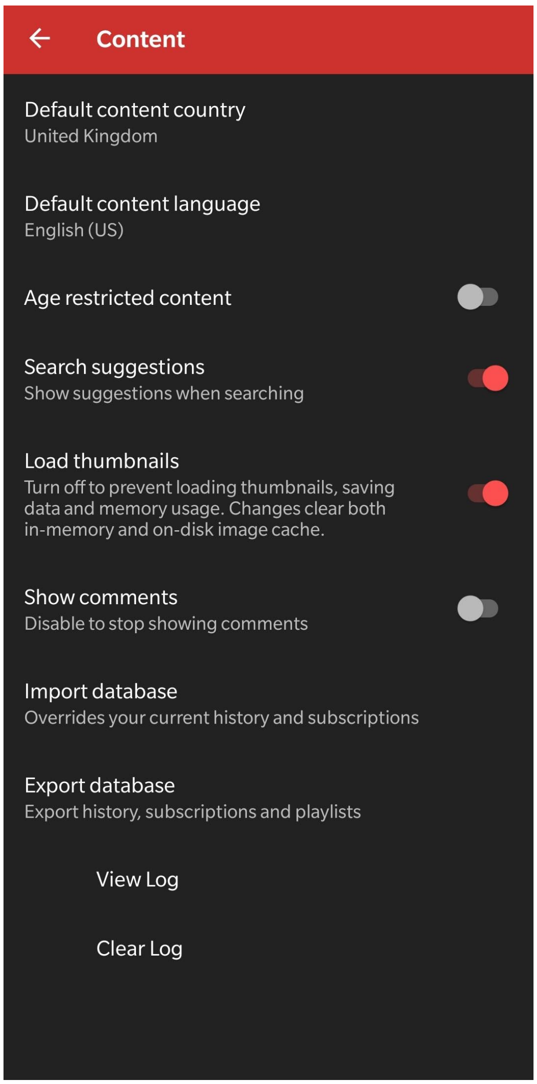


Figure 9.7: Click on the Age restricted toggle button to enable it

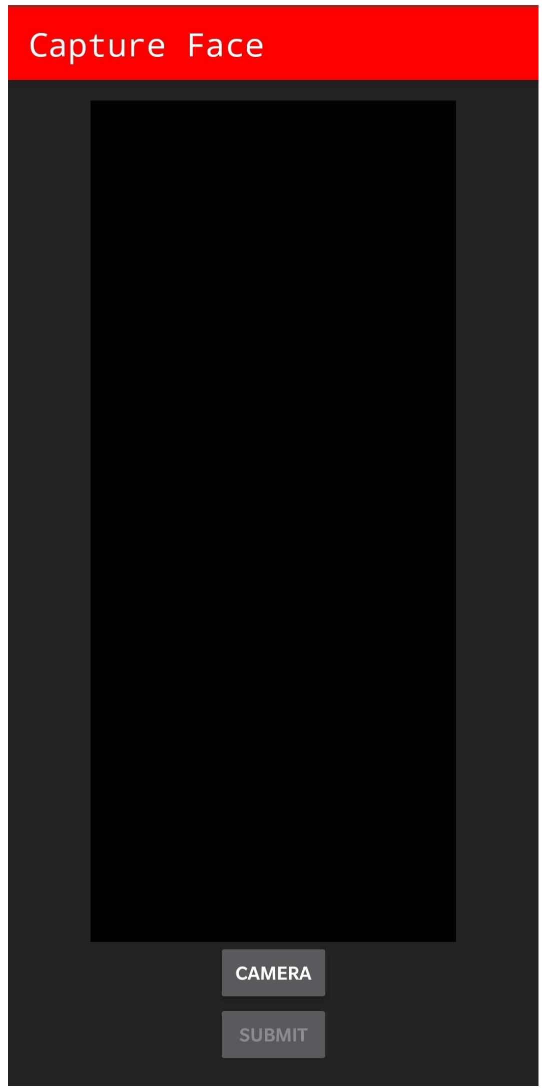


Figure 9.8: Face Capture Activity

Figure 9.9: Image Captured

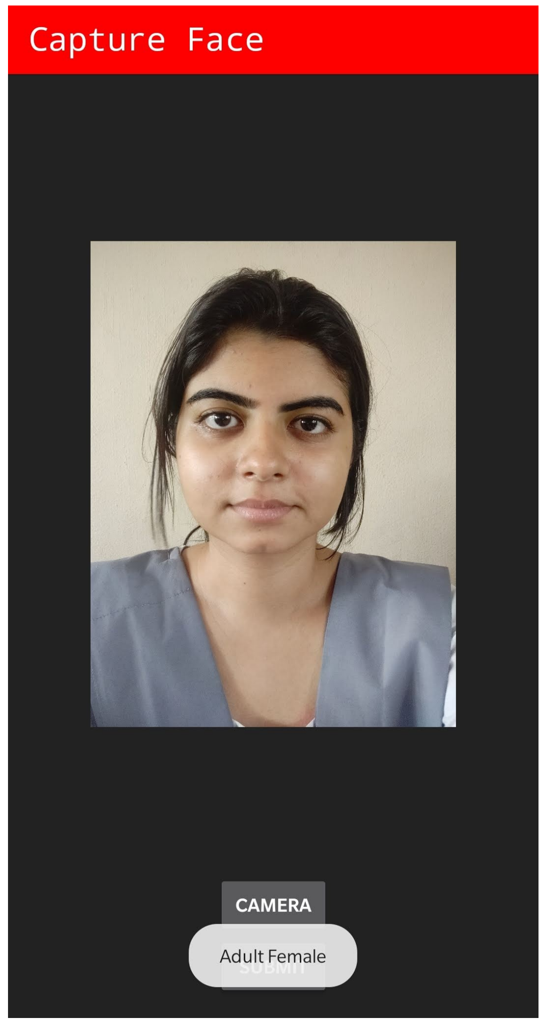


Figure 9.10: Image Classified - Adult Female

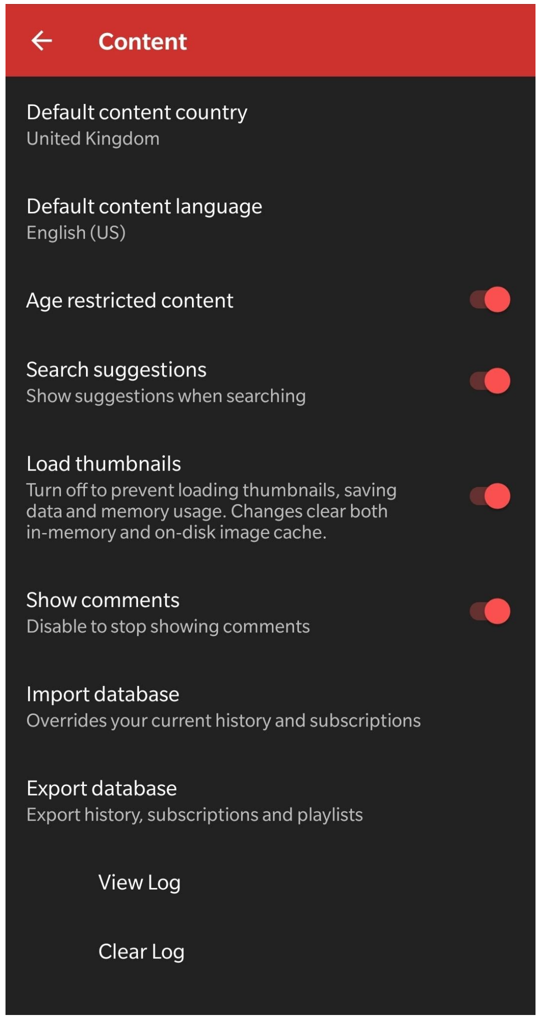


Figure 9.11: Age restricted content enabled

A log is maintained with details such as day, date, time of access, status of access and the name of the requested age restricted video. This log can be viewed by any user. If the user tries to clear the log while the "Age Restricted Content" toggle is in the disabled mode then a toast requesting to enable it is displayed. The log can thus only be cleared by an adult user.

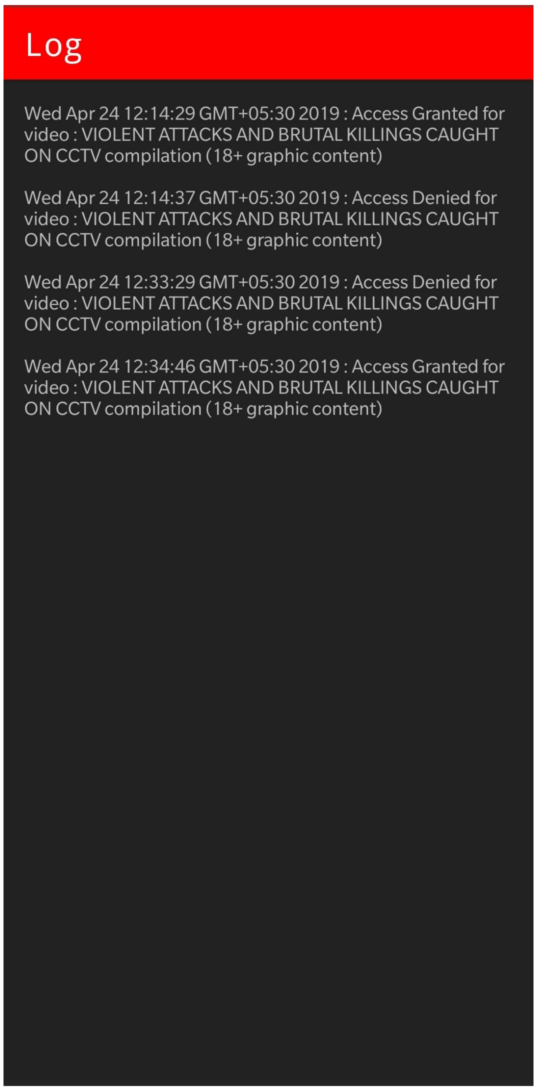


Figure 9.12: Log for Age restricted videos

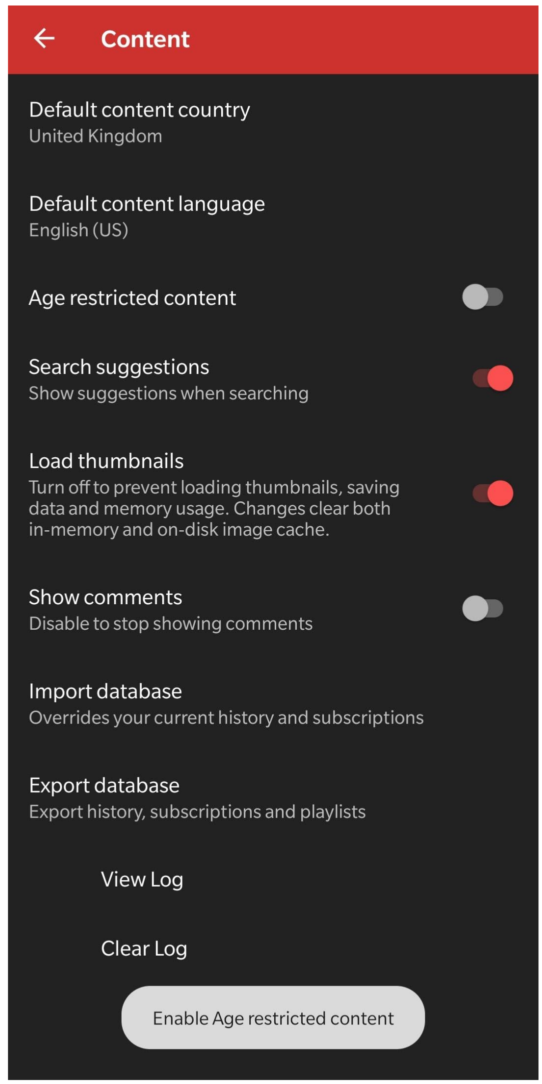


Figure 9.13: Clearing Log without enabling Age restricted content

Chapter 10

Conclusion

Present day video players have a restricted mode that can easily be toggled "on" and young viewers may fall prey to watching inappropriate content. The user manually signs up with an age which can easily be falsified. Our project is a smart video player which estimates age of user in real time and thus is more effective compared to the existing methods. It ensures that a user who has not verified his age with his image, is not granted access to view any of the age restricted content or clear the log details in the application.
A user is in safe browsing mode by default in our application, with disturbing content blocked. He/she may play such a video on request, only after their age is verified and estimated to be an adult. The user may request access by enabling the age restricted content toggle button, which initiates an image capture of the user. From the image being captured the face is cropped and age and gender of the user is estimated with the convolutional neural network model supported with Gabor filters which is run on a flask server hosted on google cloud.
A log is maintained that records the details of access to age restricted content which helps in monitoring the use of application by minors and to ensure that the access was denied to such video content. This log can be cleared by an adult user only, thus preventing minors from clearing browsing content.

Chapter 11

Future Scope

The application can be extended into the following paradigm:

- The flask server can be hosted in a production server with a scalable environment to support multiple parallel classifications in real time. This will allow multiple devices to connect and get the user's age and gender simultaneously.
- The neural network can be improved by widening the layers and improving image size on better hardware. This is so that the classification result is still received in real time.
- As we have implemented the recognition model in our video player, it can also be implemented in the popular YouTube Application.
- The age and gender recognition model can be used in a variety of applications where the need arises to distinguish between adults and children such as Smart Cars. Such cars can have an automatic locking system enabled if the user is found to be below 18 years of age.

References

[1] S. Hosseini, S. H. Lee, H. J. Kwon, H. I. Koo, and N. I. Cho, "Age and gender classification using wide convolutional neural network and gabor filter," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–3, Jan 2018.

[2] C. Huang, Y. Chen, R. Lin, and C. . J. Kuo, "Age/gender classification with whole-component convolutional neural networks (wc-cnn)," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1282–1285, Dec 2017.

[3] M. R. Dileep and A. Danti, "Multiple hierarchical decision on neural network to predict human age and gender," in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–6, Feb 2016.

[4] H. Hsieh, W. Hsu, and Y. Chen, "Multi-task learning for face identification and attribute estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2981–2985, March 2017.

[5] X. Geng, Z. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2234–2240, Dec 2007.

[6] J. Lee, Y. Chan, T. Chen, and C. Chen, "Joint estimation of age and gender from unconstrained face images using lightweight multi-task cnn for mobile applications," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 162–165, April 2018.

[7] J. Beom Ko, W. Lee, S. E. Choi, and J. Kim, "A gender classification method using age information," in *2014 International Conference on Electronics, Information and Communications (ICEIC)*, pp. 1–2, Jan 2014.

[8] M. H. Rahman, M. A. Bashar, F. H. M. Rafi, T. Rahman, and A. F. Mitul, "An automatic face detection and gender identification from color images using logistic regression," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 1–6, May 2013.

[9] H. Dibekliolu, F. Alnajar, A. Ali Salah, and T. Gevers, "Combining facial dynamics with appearance for age estimation," *IEEE Transactions on Image Processing*, vol. 24, pp. 1928–1943, June 2015.

[10] G. Levi and T. Hassncer, "Age and gender classification using convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 34–42, June 2015.