Child Safe Browser Extension: A Browser Extension to Detect Adultery and Violent Content to Make Safer Web for Children

by

Md. Alhossen
16301207
Rafika Zannat Himi
17201102
Zahid Hasan
16201028

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

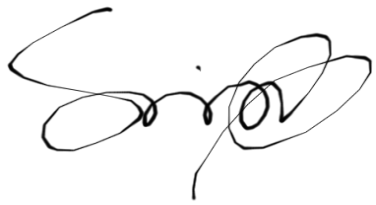
Department of Computer Science and Engineering
Brac University
May 2021

Declaration

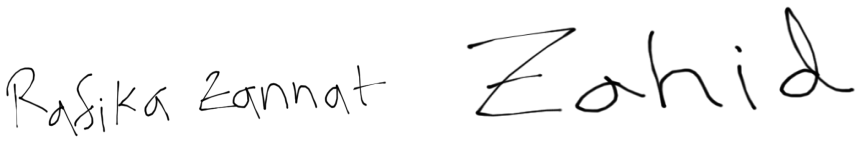
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Md. Alhossen
16301207



Rafika Zannat Himi
17201102

Zahid Hasan
16201028

Approval

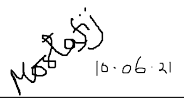
The thesis/project titled "A browser extension to detect adultery and violent content to make safer web for children" submitted by

1. Md. Alhossen (16301207)
2. Rafika Zannat Himi (17201102)
3. Zahid Hasan (16201028)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 18, 2021.

Examining Committee:

Supervisor:
(Member)

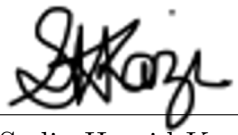


Mostafijur Rahman Akhond
Lecturer
Department of Computer Science and Engineering
Jeshore University of Science Technology

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

We, hereby declare that this thesis is based on the findings of our research. All other materials used have been properly noted in the document. This work, neither in whole nor in part, has never been submitted by anyone to any other university or institution for the award of any degree.

Abstract

The world is changing with the pace of information technology revolution and now a-days anybody can access to the internet including the children. Since birth These 21st century children are able to access to the different types of websites on the internet because of their accessible devices but not every time the internet websites are child friendly and children can view some violent and adultery images which can affect their childhood along with their mind. Furthermore, most of the people are having business from websites what is called E-commerce nowadays but the issue is that the business persons are adding ads to their websites for the profits. However, sometimes these ads can be violent and also can contain adultery images which can be viewed by a child. To stop, viewing these types of images the main way can be stopping the access of the images from a child's device which can be done by adding extension to their devices. Then parents can feel relief and children can gain their knowledge by using the great side of the internet. So, to not access the images a filter should be made but the main challenge is the making of filter which will know a specific work to maintain the extension. Also, none of the model can give 100 percent. However, when an algorithm gets heavier then it will work slowly and it will have an impact on the browser speed so to avoid this issue the model need to make a lighter algorithm. Here, to make this extension happen and efficient it will need image processing and machine learning so that it can help to make a virtual surveillance for the children and can make their mind fresh and creative. Our system will be hosted in a cloud space and will be called by the extension with the help of an API. In our system we used beautifulsoup4 for image scrapping from web page. Augmentor is for the data augmentation. For a faster machine learning PIL is our choice. And Tensorflow performed extremely well in our Convolution Neural Network.

Keywords: Machine Learning, Prediction, Scrapping, CNN, Augmentation, Tensorflow, Accuracy.

Dedication

We would like to dedicate this research work to our dearest parents. Beyond their absolute support and motivation, we could not have succeed this far.

Acknowledgement

To begin with, all praise to the Great Allah for whom our thesis have been completed without any vital interruption.
Along with, we would like to thank our honorable and conscientious supervisor Mostafijur Rahman Akhond and co-supervisor Dr. Tauhid Bin Iqbal sir for their kind support, guidance and feedback in our work. He motivated us to conduct this research and always gives his valuable time to make us understand the research works.
Furthermore, We would like to thank our respected BRAC University for providing important resources and facilities during our research phase .
To conclude, a big thank to our parents without their perpetual sup-port it may not be possible. With their mention-able support and prayer we are now on the brink of our graduation.

Table of Contents

Declaration i

Approval ii

Ethics Statement iii

Abstract iv

Dedication v

Acknowledgment vi

Table of Contents vii

List of Figures ix

List of Tables 1

1 Introduction 2
- 1.1 Research Problem . . . 3
- 1.2 Research Objective . . . 3
- 1.3 Thesis Structure . . . 3

2 Related Work 5
- 2.1 Literature Review . . . 5
- 2.2 Existing Algorithm . . . 8
  - 2.2.1 Support Vector Machine . . . 9
  - 2.2.2 Logistic regression . . . 10
  - 2.2.3 Random Forest . . . 11
  - 2.2.4 Recurrent Neural Networks . . . 12
  - 2.2.5 Convolutional Neural Network . . . 13

3 Problem Definition 15
- 3.1 Definition 1: Child Safe Browser . . . 15
- 3.2 Definition 2: Browser Suitable Machine Learning Algorithm . . . 16

4 System Model 17
- 4.1 Workflow of The System . . . 17

5 Experiment Analysis **20**
5.1 Methodologygy 20
5.1.1 Web Scrapping 21
5.2 Adult and Violence Detection Algorithm 22

6 Result Ananlysis **25**
6.1 Accuracy formula 25
6.2 Data Visualization 26
6.3 Result Explanation 28
6.3.1 Nudity Detection 28
6.3.2 Violent Image Detection 33

7 Conclusion and Future Works **39**
7.1 Conclusion 39
7.2 Future Works 39

Bibliography **41**

List of Figures

2.1 Support Vector Machine Algorithm 9
2.2 Logistic regression Algorithm 11
2.3 Random Forest Algorithm 12
2.4 RNN Algorithm 13
2.5 Convolutional Neural Network Algorithm 14

4.1 Browser Extension Architecture 17
4.2 CNN Algorithm Execution 18
4.3 Final Image Representation 19

5.1 Convolution Neural Network Architecture 20
5.2 Library Import 21
5.3 URL Validation Checker 21
5.4 Function to Scrap Web Image 22
5.5 Main Function 22
5.6 Basic Library Importing 22
5.7 Image Re-scaling 23
5.8 Path Defining to Training Image 23
5.9 CNN Core-1 24
5.10 CNN Core-2 24
5.11 Model Training 24

6.1 Bar plot data visualize 27
6.2 Nudity detection dataset sample 28
6.3 Image Reshaping code 28
6.4 Reshaped image 29
6.5 Training Result 30
6.6 Nudity Train accuracy curve 31
6.7 Nudity Model loss Curve for train data 31
6.8 Nudity Model Accuracy Curve for validation data 31
6.9 Nudity Model Accuracy Curve for validation data 32
6.10 Nudity detection original image 32
6.11 Nudity detection sample 33
6.12 Violent image original shape 33
6.13 Violent image dataset sample 34
6.14 Violence Detection Training Result 35
6.15 Violent image sample 36
6.16 Violent Model Accuracy Curve for Train data 37
6.17 Violent Model Accuracy Curve for Train data 37

6.18 Violent Model Accuracy Curve for validation data 37
6.19 Violent Model Accuracy Curve for validation data 38

List of Tables

6.1 Accuracy Table 25
6.2 Data visualization Table 27

Chapter 1

Introduction

Internet has become more and more available to every corner of the society. Anyone from anywhere can see the outside world with the help of a mobile, computer etc. In internet there are immense data, among them not all are for everyone. People search for the particular data according to their interest. But sometimes along with desired data some others content can come up which may have no relevance with the original data. Pornography and violent images can fall under this category. With the increasing rate of advertisement on every kind of pages these type of content are becoming more available to the viewers. Specially the children since they are the most vulnerable. They watch whatever content comes in front of them in any platform. Because of the increasing rate of pornography, in recent times parental control has been necessary. And to help them the demand for automatic detection of the pornography and violent image is continuously rising. Also there is threat for malware which comes along with the pornography content.

Some recent report states that there is direct relation between pornography and malware. The violent and adult images have been a big risk to the children and it is affecting their near future. So,to prevent these issues need a strong demand for specialized tools and mechanism[1] can detect the violent and adult images for children. This study is going to filter the adult and violent images by detection and supportive add-ons.These will recognize how much skin has been used and the distance between two people[2] can be calculated to detect the nudity or adult images. On the other hand, the violent images will be compared between violent and non-violent images and when it will find that that picture is not like other non-violent pictures then it will process the image to blur or block it.Internet victimization.This survey is the first comprehensive attempt to measure children's exposure[3] to violence in the home, school, and community across all age groups from birth to age 17, and the first attempt to measure the cumulative exposure to violence over the child's lifetime. The survey confirms that most of our society's children are exposed to violence in their daily lives. The ongoing system of image processing is implementing to deploy data to detect the obvious images but it has shown that it is a very heavy process. This heavy process can affect the system.

So, we want to remove this heavy deployed system to make it work faster. The processing of image to blur and block will be done by some tools to call in it.

1.1 Research Problem

In this new generation, adult images are so easy to get from anywhere by ads and from other recommendation websites. Recently, some rape cases have been aroused and everything is in front of every age's people and for that reason it associates a great misery to the children mind by watching them. Some early age kids may watch these pictures and join to abuse the victims. The violent and adult images can affect children early upbringing by being not given actual information or making them mentally weak and they can join the bad circle to fulfill their desire. The newspaper and other broadcast channel is showing the young adults and children are attached with the rape cases because of the fantasy they get from the online pornography.

Therefore, we thought with our immersive study that it is high time to stop these by creating a method of algorithm with the help of machine learning and image processing. The methodology will make a browser extension which will block or blur the images for children.

1.2 Research Objective

The proposed system determine to achieve two goals, the first one is to create a safe browser environment for children. In children's device a mode will be on to blur or block the violent and adult images to help them to grow a healthy mentality without these horrible things on their mind. Secondly, our world is now a fast pace place where the system also have to be fast. So, we decided to make a faster and relevant system by using our authentic algorithm and calculation to get rid of the violent and adult images from our children's devices. Our main contributions are present here:

1. We have proposed an image processing architecture with detection and supportive add-ons to block and blur the adult and violent images faster and authentically.

2. We have proposed a browser and android application to detect the images.

1.3 Thesis Structure

Chapter 1 - Introduction where motivation, problem statement, research objectives and contribution are discussed.

Chapter 2 - Related works where background, literature review and algorithms are discussed. In the literature review part, we reviewed the previous work on violent and nudity image detection.

Chapter 3 - Problem Definition, here it has showed two define problems of nudity and violent image delectation .

Chapter 4 - Our Proposed Approach, here it discussed about the workflow and our methodology to detect violent and nudity content.

Chapter 5 - Experimentation where we discussed about the codes used in our research.

Chapter 6 - Result Analysis and Data Visualization, where we visualized our data and analyzed the result occur-ed from our algorithms.

Chapter 7 - Conclusion and Future works where the conclusion consists of our work till now and future works include the scope of improvement.

Chapter 2

Related Work

2.1 Literature Review

The article[1] showed a method for detecting violent content in cartoon videos in order to prevent harmful psychological effects on children. Here, the authors try to achieve the basic goal of extracting general information from an picture without any specific query. In this study, local SIFT descriptors were extracted from a large set of images and then encoded using a fish vector instead of a traditional word bag representation. The projected system is an efficient way to differentiate among violent and non-violent images using discriminant kernel analysis using spectral regression. In addition, it is then used as a dimensionality lessening technique, followed by classification using the nearest neighbor classifier. SRKDA has an accuracy of 97 for this study and in the future SRKDA will be working on several deep learning features to develop the system and identify images that are not properly predicted by the violence detection system in animated videos and other images.

In [2], the author used nudity recognition and detection algorithm to filter pornographic images and videos. The algorithm will mainly detect the percentage of the skin and also depends on the color. All multimedia files were preprocessed, segmented, and filtered to analyze skin-colored pixels by processing in $\mathrm{YC}_b C_r$, space and then classifying it as skin or non-skin pixels. With pixel- wise skin detection with image processing techniques such as preprocessing, segmentation, texture filtering, and colored skin detection for pornography filtering, precision of 90.33 percentage and accuracy of 80.23 percentage were obtained. This study is basically dependent on skin color so, it will detect both nude and non-nude images based on how much skin has been shown. The future work is to use some machine learning models to detect more inappropriate contents.

In [3], the author suffered Internet damage. Survey and longevity are the first comprehensive attempts to measure exposure of child violence in homes, schools and communities across all age groups from birth to 17 years. The survey confirmed that most of the children in our society are exposed to violence in their daily lives. Over 60% of the children surveyed have been directly or indirectly exposed to violence during the past year.

The author of [4] in discussed on blocking objectionable or adult image and harm-

ful symbols. This mainly worked on the basis of image zoning. Two filters have been used here, one is adult image filter and another one is harmful symbol filter. To detect the skin statistical method was used and neural network to classify the image. Because of the better success rate the SVM is highly acceptable. And this will make the detection even easier. some machine learning models to detect more inappropriate contents. In [5], the author presented the idea of nudity detection based on image zoning. Feature extraction performed using texture and color information. Then there is SVM which was used for image classification. Highest rates of classification were attained through combining two colors based and texture-based features. Lastly the author ensures to reduce the features at the time of maintain high classification rate. In [6], a new system was introduced by the authors name MoSIFT, to spot interest points and define local features for human action acknowledgement. The spatial domain and the temporal domain are treated independently in MoSIFT. They would like to locate behaviors in frame alongside detecting them. They also introduced a bigram model to build a connection among local features to detect new global structure of actions as well as they also performed their approach to 100 hours of surveillance data on recognizing human actions in the real-world surveillance videos. In [7], authors delivered a review of some well-known approaches for pornographic and child sexual abuse (CSA) uncovering along with presenting their categorization. They found that the deep learning-based methods report state-of-the-art accuracy and meaningfully overtake all other systems based on skin detection models or image descriptor grounded pipelines. The authors felt that training more deep networks on a big dataset may perhaps sort the present state-of-the-art to more advance.

In [8], this author used a mathematical process to recognize the face image data, verbal document, written numeral picture and facial pictorial data. The method mainly focuses onto make a spectral regression to less the computational cost. Furthermore, LDA a is a popular method but can be completed each in the main input space or in the replicating kernel Hilbert space (RKHS) into which data points are plotted, then it leads to kernel discriminant analysis (KDA). When the numbers are highly nonlinear circulated, KDA can attain healthier performance than LDA. Though, calculating the projective tasks in KDA contains Eigen-decomposition of kernel matrix, which is very luxurious when a huge amount of preparation examples exists. In this paper, it presents a new algorithm for kernel discriminant analysis, named Spectral Regression Kernel Discriminant Analysis (SRKDA). KDA in the binary class circumstance has been shown to be same to normalized kernel regression with the class label as the output. The paper extends this relation to multi class case.

In [13], Based on whether domain awareness is needed or not when using them, current CNN architecture design algorithms can be divided into two separate categories. The first is the design of the "automatic + manual tuning" CNN architecture, which means that manual tuning is still needed based on the experience in designing CNN architectures. The genetic CNN approach, the hierarchical process, covers this category. The neural architecture search method, the Cartesian genetic programming method, and the metamodeling method belong to this group. It is natural that the "automatic + manual tuning" designs frequently display slightly better performance than the "automatic" designs, due to the additional benefits of manual expertise in

CNNs. When the Large-scale Evolution approach is used, the users simply perform it directly.

The reason for this article[14] was to provide researchers, editors, and followers with a set of rules that can be predicted in the process of leveraging calculated recurrence techniques. Review the tables, diagrams, graphs, and estimates to validate for a comprehensive evaluation. This article shows the preferred pattern of applying a calculated strategy along with a calculated recurrence image associated with the information set when testing the theory of study. Proposals will also be advertised for appropriate publication groups of calculated recurrence results and at least the predicted proportion of observations. The author evaluated the utilization and interpretation of the calculated recurrence in eight articles published in The Diary of Educational Research between 1990 and 2000. They found that both of the eight studies met or exceeded the proposed criteria. In this paper, the author compared[15] the classification of both Random Forest and J48 models to classify 20 flexible datasets. They brought in 20 information sets that can be accessed from the UCI repository, including outbreaks that changed from 148 to 20000. We compared the classification with that obtained from the strategy, the Random Forest and Decision Tree (J48). Classification parameters, if correctly classified, consist of F-measures, accuracy, accuracy, and review if misclassified. We investigated the pros and cons of leveraging these models for large, small information sets. Classification helps Arbitrary Woodland provide excellent results for the same number of characteristics and a wide set of information (that is, when the number of occurrences is higher), while J48 helps for a smaller set of information (less occurrences). The Breast Cancer Information Set shows the exact rate when the number of occurrences is 286 and increases by 699. In [17] the writer introduced a new technology to characterize and index violent scenes in TV dramas, movies and more. They blocked violence in the film and signed multiple "acoustic" signatures on meaningful violence identification and potential applications. They combined auditory and visual signals to identify violent scenes. But here, in addition to the visual effects similar to flashlights, the flames associated with violence are inherently accompanied by inherent acoustic effects that are interrelated in time. Investigating its audio capabilities, the author can distinguish between total fire / explosion flames with a camera flashlight. In this study, high levels of traits are very subjective and it is difficult to define the extent and extent of these traits.

The author[18] here identified the problem for image and video classification and data augmentation, along with estimating diverse methods. It is simple facts that the more data a machine learning algorithm has access to, the more actual it can be. Furthermore, when the data is of lesser quality, algorithms can really do better, providing valuable data can be extracted by the model from the main data set. In this paper authors used two diverse approaches to data augmentation. The first approach is creating augmented data before training the classifier and it can apply GANs and basic transformations to make a bigger dataset. Again, in the second approach tries to absorb augmentation through a prepended neural network. The authors also explained the work can be appropriate in more basic ways, as" style" transfer can be used to augment data in circumstances were the accessible data set is unbalanced. The authors [19] examined the part of noise injection in sta-

bility training and detect that progressive data augmentation approaches, precisely those tasks nest in managed learning also do well in semi-supervised learning. Hereafter, they approached to substitute the old-style noise injection systems with high quality data augmentation systems in order to progress consistency training. To highlight the use of improved data augmentation in steadiness training, they name their system Unsupervised Data Augmentation or UDA. They used different kind of classifications to classify the images such as, Rand-Augment for Image Classification, Back-translation for Text Classification, Word replacing with TF-IDF for Text Classification. To conclude, they said it is likely to attain great performance with inadequate labeled data. In this paper the authors[20] had recognized two original data augmentation methods for the job of hypernymy detection which have the probable to make almost limitless amounts of synthetic data. Secondly, they had demonstrated, that adding synthetic data is more actual than adding non-synthetic data in nearly all cases. In next, they released a novel standard evaluation dataset for the lexical entailment job which is not reliant on WordNet. They also have utilized Distributional Composition based Augmentation, GAN based Augmentation, WordNet based Extension, Pattern based Extension. Future work includes leveraging data augmentation for more complex models and the extension of our approach to a multilingual setup as well as domains with a more dedicated vocabulary. The authors in this paper[21] showed that AutoAugment attained excellent developments in two use cases: AutoAugment can be used directly on the dataset of attention to find the best augmentation policy and also learned policies can be moved to new datasets which can be called AutoAugment-transfer. This consequence proposes that moving data augmentation policies offers an substitute method for standard weight transfer learning. Furthermore, search algorithm facts which uses Reinforcement Learning. Also, the authors used The training of controller RNN: The controller is trained with a reward signal, that is how decent the policy is in refining the simplification of a "child model". They furthermore worked with Architecture of controller RNN and training hyperparameter. Here, the authors[22] did not aim to give a comprehensive survey of all present approaches on the other hand to briefly label the existing state-ofthe-art of methods used for a data augmentation. They also brought readers nearer to the texture transfer and style transfer approaches which permitted to yield the new images of high perceptual quality that combine the subject of a base image with the entrance of another ones. The authors offered a new expression at the style-transfer technique along with the novel application idea and applicable examples. Lastly, they discussed the compensations and drawbacks of the approaches being analyzed and existing their conclusions. Authors used Traditional transformations, Generative Adversarial Networks, Texture transfer, CNN-based methods and some usable algorithms. In addition they merged and used all of the approaches can bring enormous potential for developing data-hungry deep learning algorithms. In future the authors related to this paper will cover but will not be limited to testing the neural network competence after pre-training it with the practice of synthetic images made with the style transfer systems.

2.2 Existing Algorithm

We have used different kind of Algorithms for our research purpose. Followings are explained in this section.

2.2.1 Support Vector Machine

Support Vector Machine or SVM is a managed machine learning algorithm accomplished to do Classification [9] as well as Regression problems. SVM is applied for Classification problems in Machine Learning. The goal of the SVM algorithm is to distinct the datasets into classes to discover a greatest marginal hyperplane and is to make the most outstanding mark or judgment border that can detach n-dimensional space into classes so that we will successfully put the new data point within the redress group within the forthcoming. Hyperplane is the finest choice margin. Detaching line will be categorized with the assistance of these data points. The data points are classified by boundaries of decision like Hyperplane. Data points falling on each lateral of the hyperplane can be attributed to various classes.

SVM chooses the extraordinary points/vectors that benefit in making the hyperplane. These extraordinary cases are called as support vectors, and later algorithm is named as Support Vector Machine.

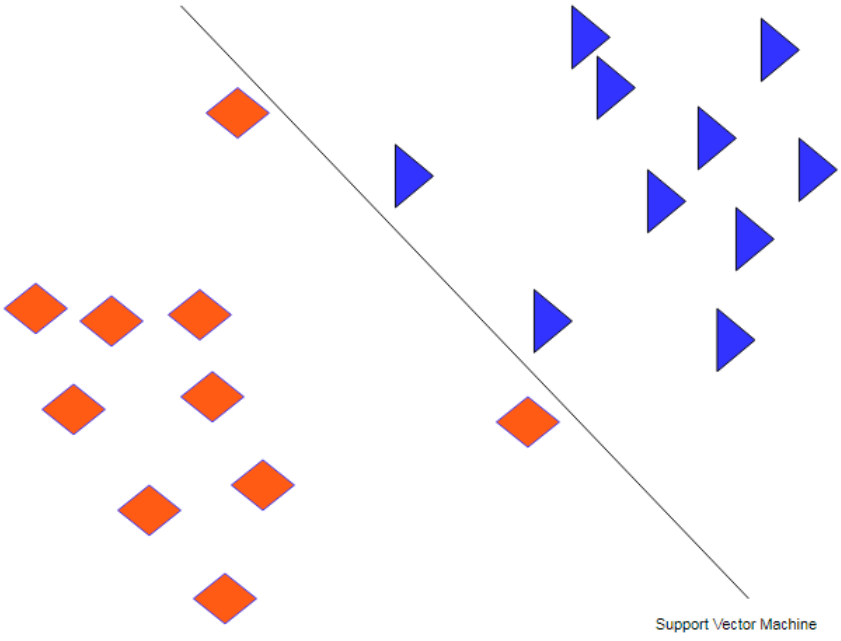


Figure 2.1: Support Vector Machine Algorithm

Pros and Cons of Support Vector Machine:

Pros:

1. When data has high dimension then a Support Vector Machine with the right settings could be the way to go and create actual valid results.

2. Support Vector Machines can be used to both predict continuous numerical values along with classify data.

3. SVMs manage data set as a whole at once along with in this sense it is not an incremental approach.

Cons:

1. Support Vector Machines are not that much efficient algorithms and again it can be quite expensive computationally to train them.

2. SVMs don't have a scalable nature and they don't work that well with mid-size or large datasets.

2.2.2 Logistic regression

Logistic regression is a classic of fashionable machine learning algorithms is under the indicated [11] learning technique. This is applied to prevent categorical sub variables involved in a given set of autonomous variables. Logistic regression is most commonly used when there are binary results in the address data. When there is a class or other class location. It can be 0 or 1, Yes or No, true or False.

Logistic regression is a method in which is used, although it is much more relative than linear regression. Linear regression is applied to solve regression problems and works on to solve classical problems. Logistic regression can be a logistic function in the form of an "S" that predicts the most extreme values instead of the tropically suitable.

Logistic Regression is an incredible machine learning algorithm because it has a volume that can classify new data using individual datasets determined by providing probabilities. It might be.
Pros and Cons of Logistic Regression:

Pros:

1. Logistic regression is easier to interpret ,implement, and very sufficient to train.

2. Logistic regression has vital accuracy for many simple data sets and it performs well when the data set is linearly separable.

3. LR can make assumption to model coefficients as indicators of feature importance.

Cons:

1. Logistic Regression is the prediction of linearity between the independent variable and the dependent variables.

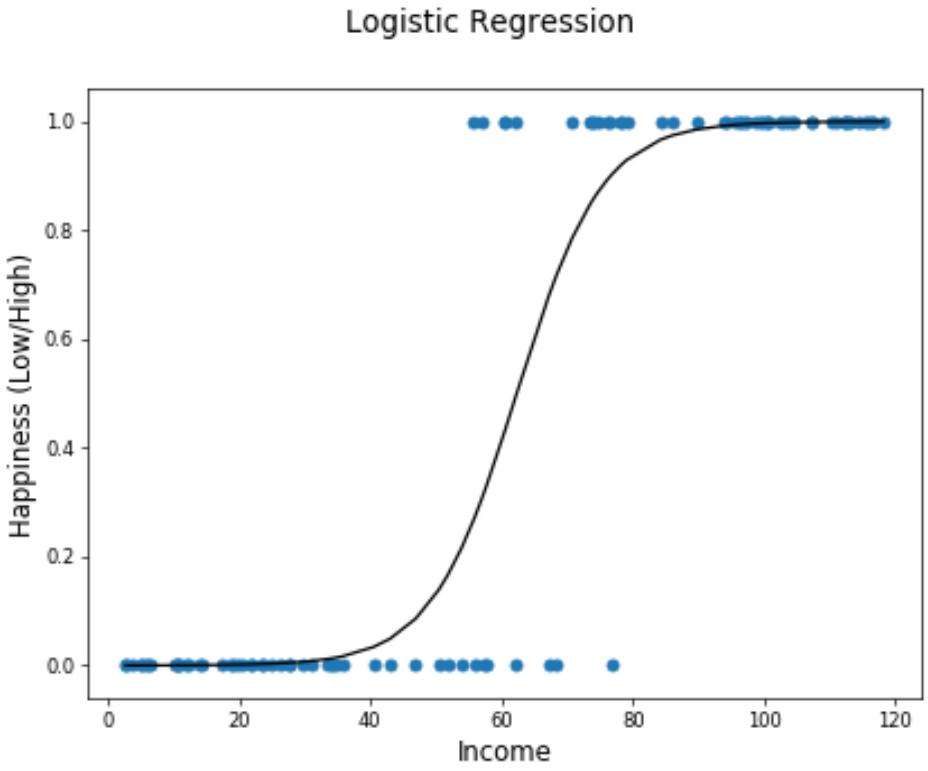


Figure 2.2: Logistic regression Algorithm

2. Logistic Regression needs average or no multicollinearity between independent variables.

2.2.3 Random Forest

Random forest is a overseen and flexible learning algorithm[12]. It is simple to operate as machine learning algorithm that produces best consequence in maximum time. It is the e leading used calculations, since of its smoothness and divergent qualities. It is the most widespread ML algorithm which fits to the supervised learning method. It is applied for both Regression and Classification difficulties in machine learning. The basic perception of Random Forest is, it is assembled learning, which may be a grip of combining diverse classifiers to measure acomplex subject and to make paces the implementation of the demonstrate. Random Forest may be a classifier that covers a number of decision trees on different subsetsof the given dataset and receipts the average to move forward the prophetic accuracy of that dataset.

Rather than relying on one decision tree, the random forest takes the forecast from each tree and based on the popular votes of estimates, and it predicts the definitive yield. Themore notable number of trees within the forest leads to advanced exactness, precision and antedates the matter of over fitting.

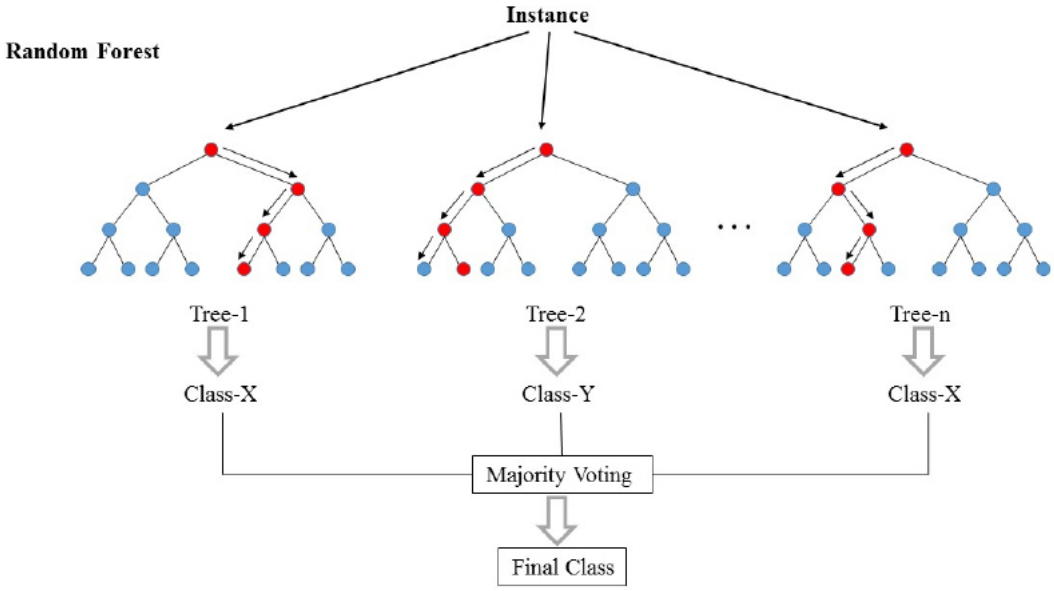


Figure 2.3: Random Forest Algorithm

Pros and Cons of Random Forest:

Pros:

1. Random forest can hold thousands of input variables and identify maximum significant variables so it is well-thought-out as one of the dimensionalities lessening system.

2. Random forest has an active system for assessing lost data and maintains accuracy when big portion of the data are missing.
has ways for balancing errors in data sets where classes are not balanced.

Cons:

1. Random forest is not a descriptive tool and it is just a predictive modeling tool.

2. In Random Forest Over fitting can happen easily.

2.2.4 Recurrent Neural Networks

RNN is a commanding and energetic type of neural network [25], and it remains the most promising algorithm used because RNN is the only algorithm with internal memory. Thanks to the RNN's internal memory, it can remember important things about the inputs it recognizes. A recursive neural network can make a deep statement of order and context compared to other algorithms.
RNNs and feed-forward neural networks can be named from the methods shown in Network Channel Information. In a feed-forward neural network, information

only changes one way from the input layer to the hidden layer to the output level. Information is modified directly over the network and does not double the node.

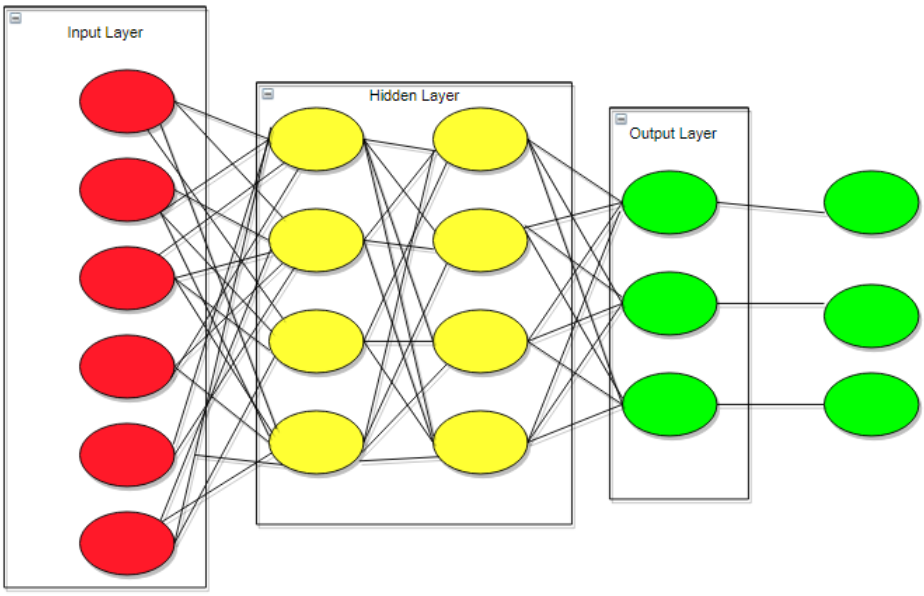


Figure 2.4: RNN Algorithm

Pros and Cons of Random Forest:

Pros:

1. RNN model is modeled to recall individual information during the time which is very supportive in any period series predictor.

2. The model size does not rise, if the input size is bigger.

3. The weights can be shared through the time stages.

Cons:

1. RNN models can be difficult during the time of its training.

2. The calculation is slow, because of its recurrent nature.

2.2.5 Convolutional Neural Network

Convolutional Neural Network or CNN is a Deep Learning algorithm[13] that can yield in an input image and it is one of the main types to do image or picture

data classification and picture salutation in neural networks. Scene labeling, objects discoveries, and confront acknowledgment, etc., are a few of the zones where convolutional neural systems are largely used. CNNs are applied for image data classification and acknowledgment since of its higher accuracy. Convolutional Neural Network takes after a broadminded demonstrate which works on building a network similar a pipe, and at long last gives out a fully-connected layer where all the neurons are accompanying to each other and the yield is handled.

Additional purpose why CNN are vastly well known is since of their architecture-the most outstanding thing is there's no necessitate of comprise withdrawal. The framework absorbs to do include extraction and the center perception of CNN is, its employments convolution of picture and channels to yield invariant highlights which are conceded on to the following layer.

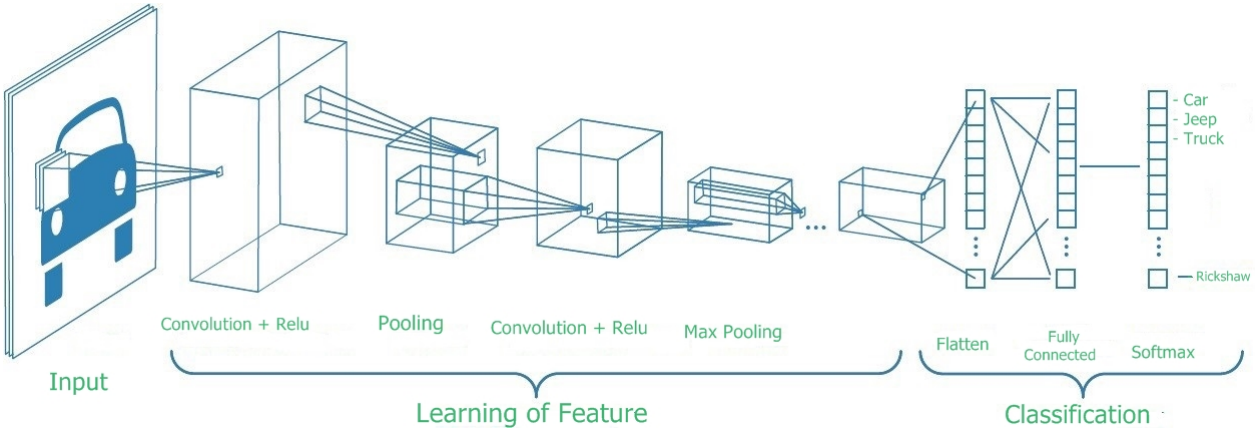


Figure 2.5: Convolutional Neural Network Algorithm

Pros and Cons of Convolutional Neural Network:

Pros:

1. Convolutional Neural Network can make learning of accurate insights and patterns from the provided data.It depends on how clean, well structured or feature engineered the data is).

2. CNNs results are more accurate than typical machine learning techniques.

3. CNN can provide better outcomes than other machine learning algorithms if that is tuned along with good amount of feeded data.

Cons:

1. Convolutional Neural Network requires more computing power.

2. CNN's performance issues happen if someone is not noticing big data tools.

Chapter 3

Problem Definition

3.1 Definition 1: Child Safe Browser

The Internet has gone from an interest to an obligation in an intensely brief period of time. When mutual with the outpouring in mobile devices like tablet computers, smartphones and so on, the technology has converted a fixture of the up-to-date way of life. With a lot of access, though, it's significant for parents to do their finest to make sure their children stay safe. Parents need to safeguard that they are not only modifiable time spent online, as well as cultivating open communication, and making sure they're on the similar page their children and adolescents are. Social media networking sites are understood as being very casual by children and they may be overstretched by their friends into linking them. The sites don't really present any threats that don't previously exist somewhere else online. The hazard is that the threats occurring in a new online atmosphere, a kid may not be acquainted with. As with most latent online risks, the complications can start through a child seeing offensive images which can contain adult images along with violent images.

In different websites they receive advertise money so it gives the chance to many adult industries to post their ads. These ads sometimes can be seen by the children which can cause unsafe environment for innocent kids.

In addition, violent images of different categorized subjects can cause same kind of issue to innocent child's mind. These images can root the problematic behavior of children. Furthermore, it also can hamper mental stability of kids and adolescent.

To solve these a safe extension to block the inevitable browsers can help the children and their parents a normal upbringing. This type extension can block adult and violent images which are really inappropriate not the images which are nude or showing skin with an avoidable matter such as a little boy running in the beach only wearing a pant. So, this can be a whole new future with safe browsing in this internet based solidity.

3.2 Definition 2: Browser Suitable Machine Learning Algorithm

Now a days Machine Learning algorithms are getting widely used in different section of the tech worlds for different purpose. Because of the invention of the browsers, users could search for any information that is need or do any kind of official or unofficial works. But after some time there was a need felt by the users that browsers can't do everything. So the idea of extensions came out to help with exceptional features. Among them machine learning step into. Which kind of changed the whole perception of the browser capabilities. Based on this feature we build our extension which runs on the machine learning. In recent times several machine learning experiments have been conducted and most of them have loading issues or they take several moments to finish their assigned job.

Since we are working with machine learning in our extension that's why we had to keep in mind these limitations. In addition we needed to make it user friendly. For that we chose Convolution neural network or CNN. There is another algorithm named RNN. But it has limitations and were less suited for our system that's why CNN has must be accepted. We need to design our algorithm in such a way that firstly it does not get slow like traditional machine learning algorithm and secondly it finishes the given job within a short time since the loading of web pages by the browser is the main priority for any user. Like if an extension spends too much time to process the images of a web page then simply the user would not be interested to use it in daily life. Furthermore to keep extension a light weighted we need to host the core CNN algorithm in a cloud space and an API should be created so that based on that single API call the pre-trained algorithm gets into work. In the end we can say that our system will be lighter weighted, user friendly and fast than traditional existing one's with Machine Learning algorithms.

Chapter 4

System Model

4.1 Workflow of The System

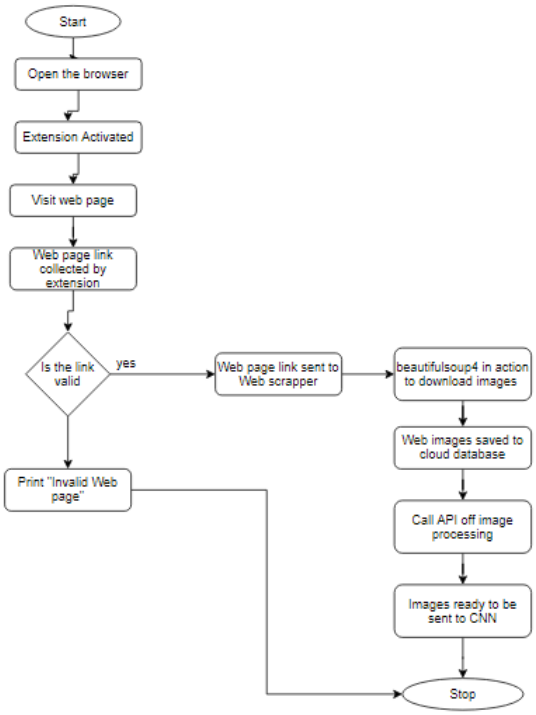


Figure 4.1: Browser Extension Architecture

To describe our model, we have divided it in some sub sections like Image collection, testing them and lastly taking action according the results. First let's see how the first section will work. Firstly the System will be activated by turning on the browser. Then, if the browser is being used by an adult and there is no need to use the system or extension then it can be turned off or the parents can turn it on when

their kids start using the internet. Alright, now after the user starts surfing the web and enters in a new web page, if the page is a valid and not a spam or advertising webpage then within a fraction of seconds by the web scrapper using beautifulsoup4 all the images of the web page will be downloaded and stored in a cloud database. Else the web page will be blocked and user will notified. After the storing of the images, the system from the extension will call the API of the CNN hosted in cloud space. After the call is done then the images will be feeded to already trained CNN model.

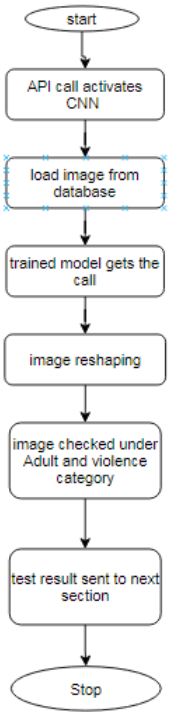


Figure 4.2: CNN Algorithm Execution

Now comes the critical part of our system. After getting the call of the API from the extension and receiving the images the Convolutional Neural Network will start testing each images. So, when the API calls the CNN, the testing model loads the images from dataset. After that it starts reshaping each images it has to test since the training has been done following a fixed shaped to make the algorithm work faster and effectively. Then the specific path of the image is sent in another function. After that the image gets a prediction value that will help in deciding in the next section of the system to whether pass the image or blur them. Lastly after getting the prediction value of the images, it will be sent to execution section for final action.

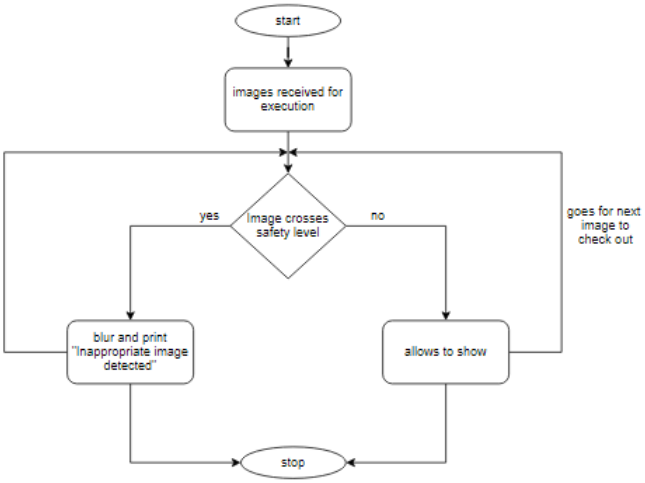


Figure 4.3: Final Image Representation

Finally let's see how the execution or final action of the system will work. So, when the predicted value of each image is received by the execution part, it will start it work by checking the value. Now, the safety value set by observing the training, will come in action. System will now check the value of each image and if it crosses the safety level the image will be blurred and then a warning message will be printed on that image. In contrast the image will be passed and the page will be loaded after all the images has been checked. And all of the actions will be done within a very short time. But it can sometime vary depending on the size of the web page.

Chapter 5

Experiment Analysis

5.1 Methodologygy

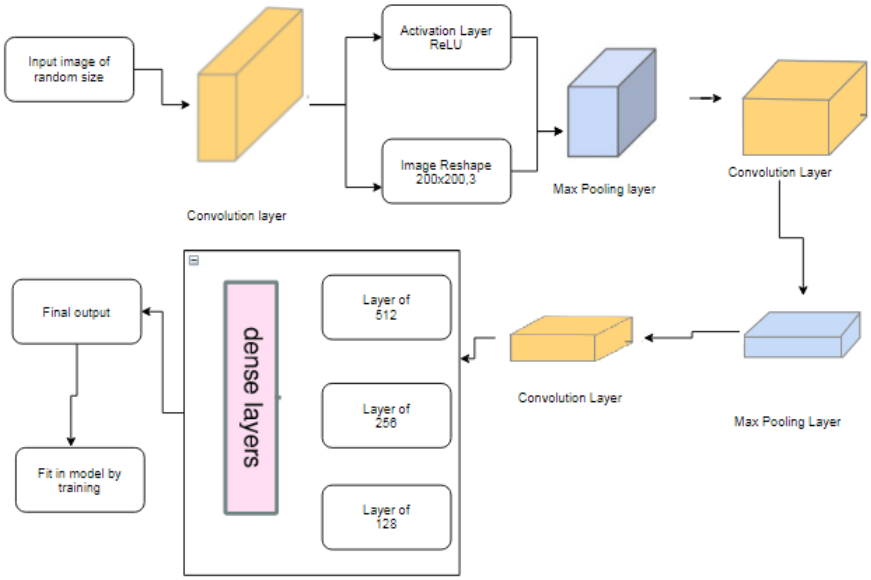


Figure 5.1: Convolution Neural Network Architecture

Every system has a core component that keeps it running and helps it to do the task as assigned. So now we'll be talking about our core components for the machine learning algorithm. For our system we have used convolutional neural network. So first let's get into the main part of our algorithm. First we initiated the path to the training images that we have divided the dataset and to train we used 70%. So if you see from the given image that whenever image first comes in random shape. So, the model first put it into a convolutional layer of (32, 3x3) after that the model sends the image to reshape in a (200 X 200,3) format and we used the Rectified

linear Unit activation function. We used to ReLU because it doesn't activate all the neurons that will be using at the same time. So when it works with one neuron the others stay put after its work finished with first layer we send it to Max pulling. We know that Max pooling helps to reduce the dimension of the feature maps That's why when the learning steps comes, Max pooling reduces the parameters and after that in the end it summarizes the features that are left with. Then again we send it to another convolutional layer of (64,3x3).

The same steps repeats and finally we sent it to the last Convolution layer. Last layer sends the results from the convolution layer to dense layer. Using the dense layer, input the data of the output of previous layer to the neurons and output from dense layer is m dimensional vector. Finally, to mention about dense layer it basically helps to change the dimension of the vector. After the output that we have to feed it to the model to train them and after all of the training we get a full pledge trained model and we are ready to test them on the rest of the database we have left. We have imported the necessary libraries like keras, cv2, tensorflow, matplotlib.

5.1.1 Web Scrapping

```
!pip install beautifulsoup4
import requests
import os
from tqdm import tqdm
from bs4 import BeautifulSoup as bs
from urllib.parse import urljoin, urlparse
```

Figure 5.2: Library Import

Next to make sure an authentic URL so that we could avoid invalid URLs, we used this part of the code as instead of a URL there can be some data that has been encoded.

```
def is_valid(url):
    """
    Checks whether `url` is a valid URL.
    """
    parsed = urlparse(url)
    return bool(parsed.netloc) and bool(parsed.scheme)
```

Figure 5.3: URL Validation Checker

Here is the function that is basically downloading the necessary files we need

```
def download(url, pathname):
    """
    Downloads a file given an URL and puts it in the folder `pathname`
    """
    # if path doesn't exist, make that path dir
    if not os.path.isdir(pathname):
        os.makedirs(pathname)
    # download the body of response by chunk, not immediately
    response = requests.get(url, stream=True)
    # get the total file size
    file_size = int(response.headers.get("Content-Length", 0))
    # get the file name
    filename = os.path.join(pathname, url.split("/")[-1])
    progress = tqdm(response.iter_content(1024), f"Downloading {filename}",
total=file_size, unit="B", unit_scale=True, unit_divisor=1024)
    with open(filename, "wb") as f:
        for data in progress:
            # write data read to the file
            f.write(data)
            # update the progress bar manually
            progress.update(len(data))
```

Figure 5.4: Function to Scrap Web Image

And lastly, we have the main function that is responsible for running this whole code

```
def main(url, path):
    # get all images
    imgs = get_all_images(url)
    for img in imgs:
        # for each image, download it
        download(img, path)
```

Figure 5.5: Main Function

5.2 Adult and Violence Detection Algorithm

First, we imported the necessary modules

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import RMSprop
import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
import cv2
import os
```

Figure 5.6: Basic Library Importing

Next, the images we have, the RGB value ranges from 0 to 255, so make it range from 0 to 1 we divided each element by 255.We did it for training a validation.

```
train = ImageDataGenerator(rescale=1/255)
validation =ImageDataGenerator(rescale= 1/255)
```

Figure 5.7: Image Re-scaling

Next, we redirected the path to the dataset we have and we also resized the image since we can't feed multiple sized images to our algorithm, that's why we resized as (200,200). Next we defined the batch size as 10 and we defined the classes. As we have only two classes for both type like nude and non-nude or violent and non-violent we given binary as the class. And so we created for both train and validation data. Since we that neural network cannot take string type input that's why it will convert the specified class as 0 and 1 for both nudity and violent.

```
train_dataset = train.flow_from_directory('/content/Traingn/train',
                                          target_size= (200,200),
                                          batch_size=10,
                                          class_mode='binary')
valid_dataset = validation.flow_from_directory('/content/Traingn/validation',
                                               target_size= (200,200),
                                               batch_size = 10,
                                               class_mode = 'binary')
```

Figure 5.8: Path Defining to Training Image

Following that, we defined our CNN(Convolutional Neural Network) with max pooling and all the layers defined as list. We used 16 filters and given the size there. Here for the activation function we use "Relu". Here the layers can have only one input an one output layer. To calculate the largest value from each layer the max pooling is used and because of this the original matrix cuts down into half. All the layers are grouped into tf.keras model.

```
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3), activation
                                    = 'relu', input_shape =(200,200,3)),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    tf.keras.layers.Conv2D(32,(3,3),activation
                                    = 'relu'),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    tf.keras.layers.Conv2D(64,(3,3), activation
                                    = 'relu'),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    tf.keras.layers.Conv2D(128,(3,3),activation
                                    = 'relu'),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(512, activation
                                    = 'relu'),
                                    tf.keras.layers.Dense(256, activation
                                    = 'relu'),
                                    tf.keras.layers.Dense(128, activation
                                    = 'relu'),
                                    tf.keras.layers.Dense(1, activation
                                    = 'sigmoid')
                                   ])
```

Figure 5.9: CNN Core-1

```
model.compile(loss= 'binary_crossentropy',
              optimizer = RMSprop(lr=0.001),
              metrics =['accuracy'])
```

Figure 5.10: CNN Core-2

Last part of this code is the training using one of the three libraries of keras that is fit. The epoch means the neural network gets trained with the data in one cycle. All of the data is used one at a time. Epoch contains both forward and backward pass and both are counted as one. At least one or more batches are considered as one epoch. We set the epoch as 5 which we have to alter for a better result.

```
model_fit = model.fit(train_dataset,
                      steps_per_epoch = 5,
                      epochs = 200,
                      validation_data= validation_dataset
                      )
```

Figure 5.11: Model Training

Chapter 6

Result Ananlysis

6.1 Accuracy formula

Accuracy is the formula in fraction that shows how much accurately the model can give the satisfactory result or the right result to show the formula now it's given below hair TP equal to positive TN equal to negative FM equal false negative and ft equal false positive.

$$\text{Accuracy} = \frac{Number\ of\ Correct\ predictions}{Number\ of\ Correct\ predictions}$$

In binary classification we can calculate the accuracy by the following formula:

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

Now the results obtained for our CNN and RNN algorithm using the above formula:

Algorithms	Accuracy
CNN	93%
RNN	84%

Table 6.1: Accuracy Table

From the above comparison we can see that the CNN algorithm gives accuracy of about 90% and the RNN algorithm gives accuracy of 84. And by comparing we can easily get that CNN algorithm is performing far better than RNN algorithm. Also earlier we mentioned about the other algorithm and they are support vector machine and random forest. Deep neural network has all the neurons interconnected in all possible way inside all the layers. Because of this, the network becomes more complex as there are more parameters which also leads to over fitting. Sometimes in a picture there might be a lot of pixels that are lying in other portion of the picture

except the main object that is the target, which might not be not be helpful to the overall image or it may not help in the proper. Because of these less contribution and to Bypass these issues convolution neural network was brought up. In this network the pixel counting happens in a more random way and it reduces the number of dimensions for the picture. Which later helps to predict the image in a similar way the fully connected network would do. Because of this network the dimensions shrinks at a huge rate and it leads to less complexity of the model because of the less parameters it has to learn. That's why it became our ultimate choice and the result will be followed in the upcoming sections.

6.2 Data Visualization

Since we are working for a big scale algorithm that requires thousands of images to be processed to get a good result we plotted them in some graphs to give a better visualization of the size of our dataset. Inside the dataset we have used some tags to identify the images in a short time.

Column 1	Column 2
EXPOSED-ANUS	EXPOSED ANUS
EXPOSED ARMPITS	Exposed Armpits; Any gender
COVERED-BELLY: Provocative, but covered Belly; Any gender	EXPOSED-BELLY Exposed Belly;Any gender
COVERED-BUTTOCKS Provocative, but covered Buttocks; Any gender	EXPOSED-BUTTOCKS Exposed Buttocks; Any gender
FACE-F Female Face	FACE-M Male Face
COVERED-FEET Covered Feet; Any gender	EXPOSED-FEET Exposed Feet; Any gender
COVERED-BREAST-F Provocative, but covered Breast; Female	EXPOSED BREAST F Exposed Breast; Female
COVERED GENITALIA F Provocative, but covered Genitalia; Female	EXPOSED GENITALIA F Exposed Genitalia; Female
EXPOSED BREAST M Exposed Breast; Male	EXPOSED GENITALIA M Exposed Genitalia; Male

Table 6.2: Data visualization Table

Other than this to keep a balance we collected thousands of normal photos and made 10000 images by implementing data augmentation. Which we categorized as safe or non-nude images. We also collected images for our violent and nonviolent purpose. For violent we collected images containing knife, blood, wound, offensive etc. images of around thousand. Then we increased its number as well using data augmentation. Lastly for nonviolent section we used data augmentation as well to fit the number of images in the CNN model so that the model recognizes the desired images with ease. This shows the percentage of the dataset among the total number of images.

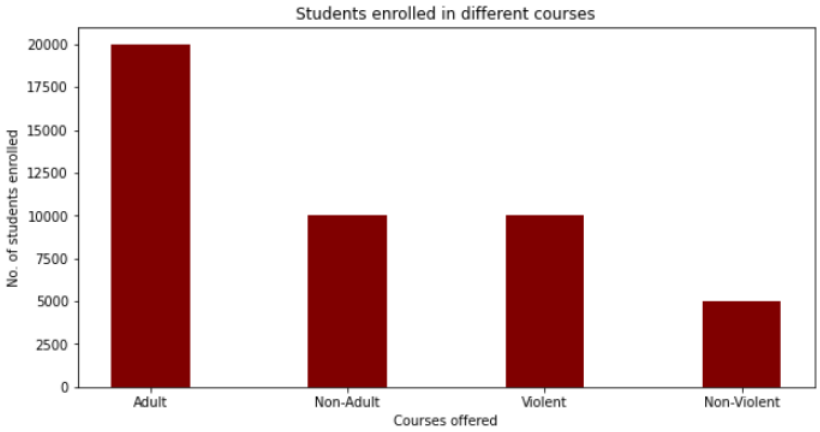


Figure 6.1: Bar plot data visualize

And this shows the exact number of images that each dataset contains.

6.3 Result Explanation

6.3.1 Nudity Detection

First let's check out how we trained our images for further testing. For example check this image to train. For that we first reshaped the image as the image processing system works very in a better way with reduced pixels images. We checked the shape of the image.

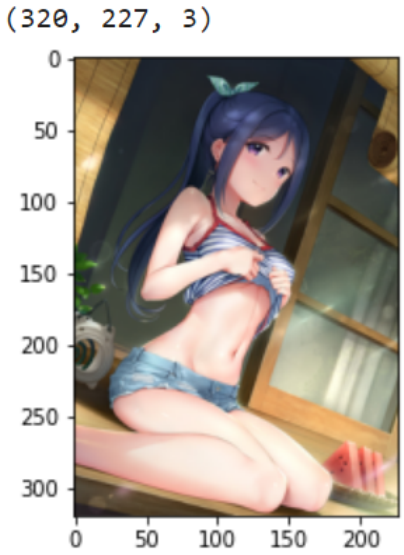


Figure 6.2: Nudity detection dataset sample

then we reshaped it, using the following code

```
f = r'/content/Vio-test'
for file in os.listdir(f):
    f_img = f+"/"+file
    img = Image.open(f_img)
    img = img.resize((200,200))
    img.save(f_img)
```

Figure 6.3: Image Reshaping code

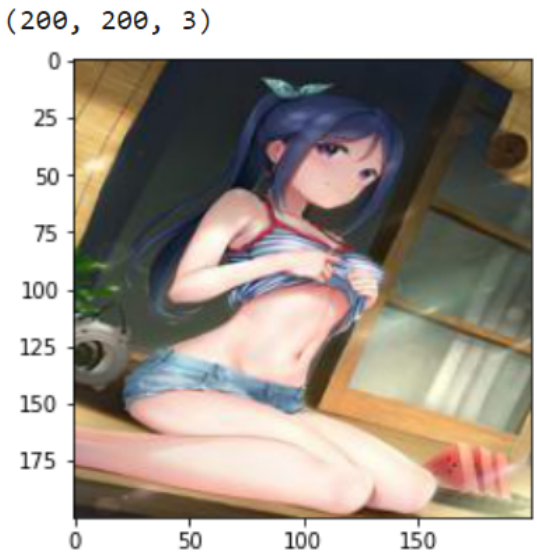


Figure 6.4: Reshaped image

We reshaped the image into 200x200 then we trained our 20000 images, here we have shown a sample of that result:

```
Epoch 1/20
5/5 [==============================] - 4s 896ms/step - loss: 0.4777 - accuracy: 0.8000 - val_loss: 0.4013 - val_accuracy: 0.8549
Epoch 2/20
5/5 [==============================] - 4s 880ms/step - loss: 0.4998 - accuracy: 0.8000 - val_loss: 0.4067 - val_accuracy: 0.8238
Epoch 3/20
5/5 [==============================] - 4s 833ms/step - loss: 0.5272 - accuracy: 0.8605 - val_loss: 0.3473 - val_accuracy: 0.8601
Epoch 4/20
5/5 [==============================] - 4s 874ms/step - loss: 0.5377 - accuracy: 0.7442 - val_loss: 0.4039 - val_accuracy: 0.8290
Epoch 5/20
5/5 [==============================] - 4s 876ms/step - loss: 0.4928 - accuracy: 0.8200 - val_loss: 0.3095 - val_accuracy: 0.8497
Epoch 6/20
5/5 [==============================] - 4s 877ms/step - loss: 0.5941 - accuracy: 0.8200 - val_loss: 0.5004 - val_accuracy: 0.7306
Epoch 7/20
5/5 [==============================] - 4s 874ms/step - loss: 0.3647 - accuracy: 0.8400 - val_loss: 0.3105 - val_accuracy: 0.8549
Epoch 8/20
5/5 [==============================] - 4s 871ms/step - loss: 0.2359 - accuracy: 0.9302 - val_loss: 0.2695 - val_accuracy: 0.8912
Epoch 9/20
5/5 [==============================] - 4s 877ms/step - loss: 0.4553 - accuracy: 0.8600 - val_loss: 0.3270 - val_accuracy: 0.8756
Epoch 10/20
5/5 [==============================] - 4s 826ms/step - loss: 0.2738 - accuracy: 0.8837 - val_loss: 0.2294 - val_accuracy: 0.9275
Epoch 11/20
5/5 [==============================] - 4s 878ms/step - loss: 0.2951 - accuracy: 0.8400 - val_loss: 0.2355 - val_accuracy: 0.9326
Epoch 12/20
5/5 [==============================] - 4s 886ms/step - loss: 0.3196 - accuracy: 0.8400 - val_loss: 0.4143 - val_accuracy: 0.8031
Epoch 13/20
5/5 [==============================] - 4s 876ms/step - loss: 0.2546 - accuracy: 0.9200 - val_loss: 0.1830 - val_accuracy: 0.9326
Epoch 14/20
5/5 [==============================] - 4s 870ms/step - loss: 0.1970 - accuracy: 0.9200 - val_loss: 0.2759 - val_accuracy: 0.8860
Epoch 15/20
5/5 [==============================] - 4s 875ms/step - loss: 0.2152 - accuracy: 0.9400 - val_loss: 0.7824 - val_accuracy: 0.6891
Epoch 16/20
5/5 [==============================] - 4s 869ms/step - loss: 0.1837 - accuracy: 0.9000 - val_loss: 0.5811 - val_accuracy: 0.7720
Epoch 17/20
5/5 [==============================] - 4s 884ms/step - loss: 0.2409 - accuracy: 0.9200 - val_loss: 0.1996 - val_accuracy: 0.9171
Epoch 18/20
5/5 [==============================] - 4s 873ms/step - loss: 0.4638 - accuracy: 0.8200 - val_loss: 0.2291 - val_accuracy: 0.9171
Epoch 19/20
5/5 [==============================] - 4s 885ms/step - loss: 0.3095 - accuracy: 0.8605 - val_loss: 0.1259 - val_accuracy: 0.9534
Epoch 20/20
5/5 [==============================] - 4s 886ms/step - loss: 0.1700 - accuracy: 0.9400 - val_loss: 0.1313 - val_accuracy: 0.9430
```

Figure 6.5: Training Result

After Training we get almost 0.9400 means that almost 94 percentages accuracy is possible to get and the loss is 0.1700 or 17 percentages. Also if we see the validation then we get 0.9403 or 94.03 percentages accuracy and we lose 0.1313 or 13.13 percentages accuracy. After this at the time of testing we used 15 epochs and alternated in different values to observe to changes.

To make it clearly understandable about the effectiveness of our model it is essential to get a visual of the end results. And we can use different techniques of existing visualization. Since python got tons of libraries that will help us to visualize our output result. We have used “Model Accuracy” and “Model Loss” differently for both training and Validation data to get a proper idea about both of them. We also get some predicted value from the test images that used. The “Model Accuracy” and “Model Loss” curve for both Nudity and Violent images our CNN model is provided below:

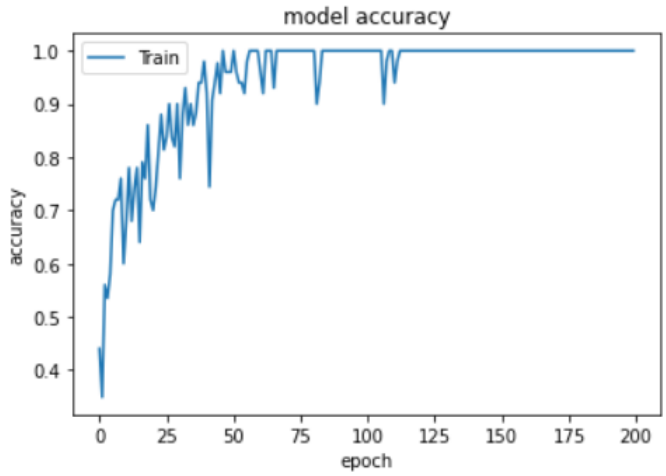


Figure 6.6: Nudity Train accuracy curve

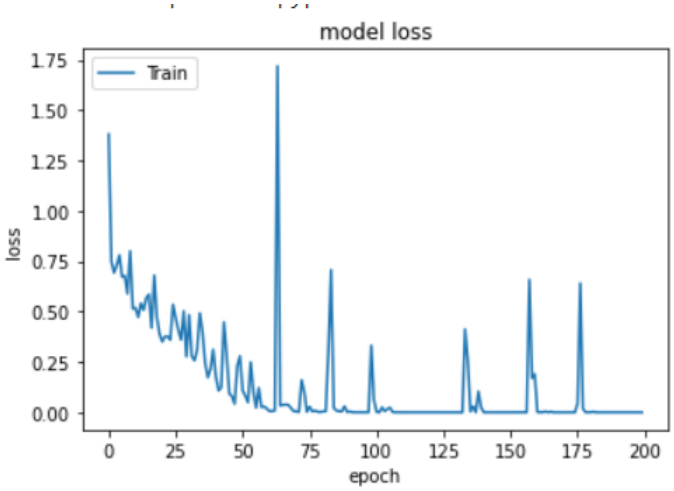


Figure 6.7: Nudity Model loss Curve for train data

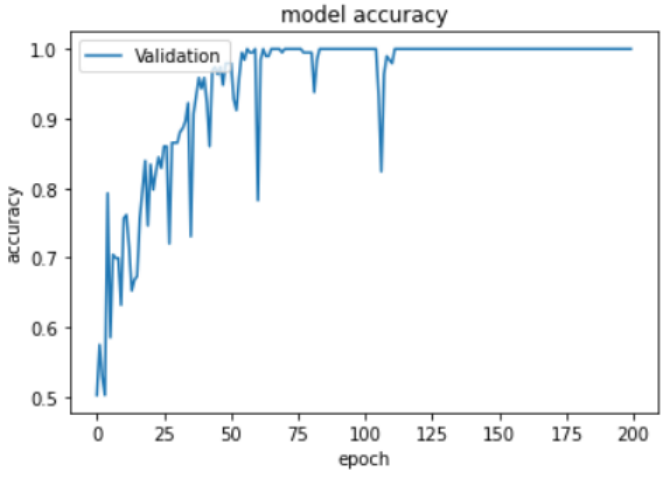


Figure 6.8: Nudity Model Accuracy Curve for validation data

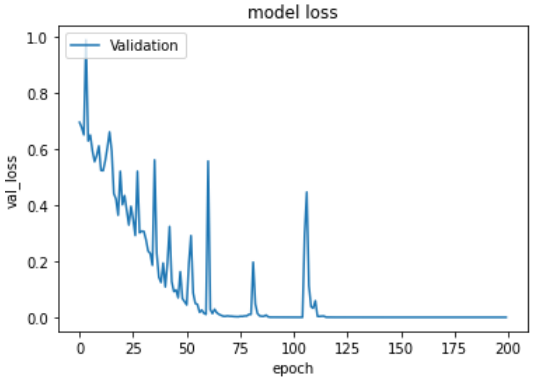


Figure 6.9: Nudity Model Accuracy Curve for validation data

Now let's test one of our sample image :

Figure 6.10: Nudity detection original image

First we reshape the image,

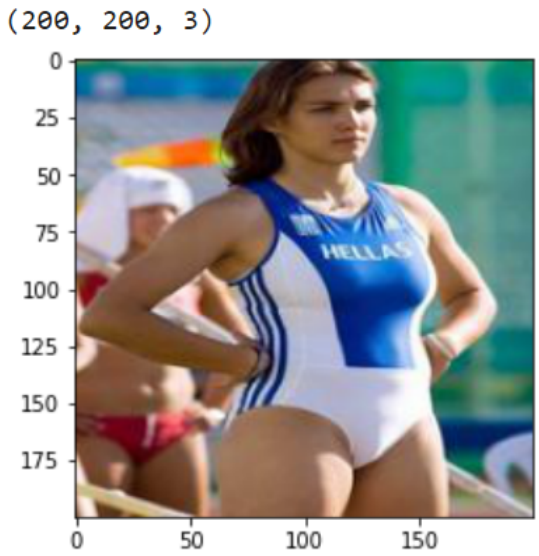


Figure 6.11: Nudity detection sample

Then we run our testing code on it by using (model-fit.predict) and here what we get Safe= 87.373648436438 percentages
Non-Safe= 12.626351563562 percentages.

Here we can see a woman on beach. And since it had chance to be get selected under non-safe category for children but as our model is designed in such a way that it detects the accurate category of an image which is suitable and can be allowed.

6.3.2 Violent Image Detection

Violent original image:

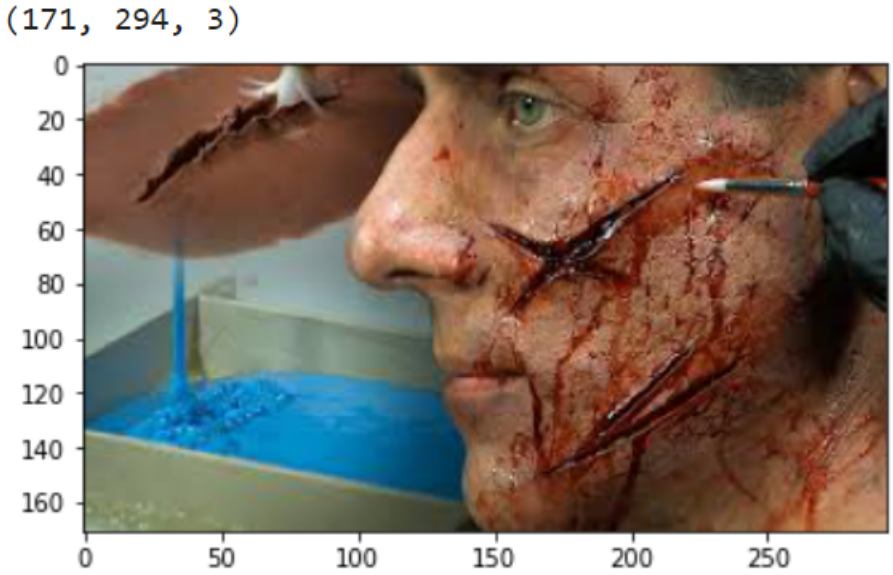


Figure 6.12: Violent image original shape

After reshaping the output is,

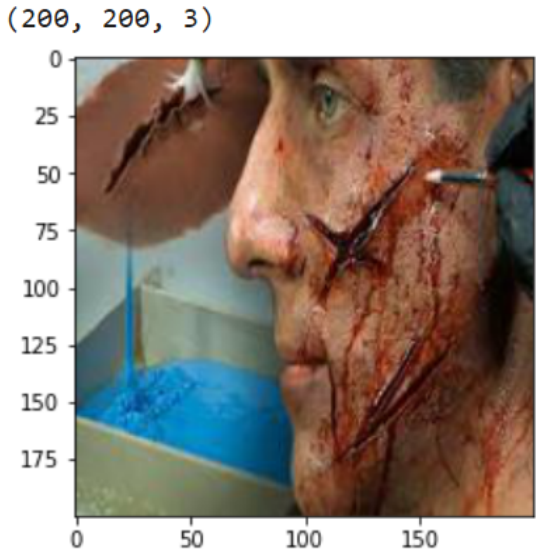


Figure 6.13: Violent image dataset sample

We reshaped all the images into 200x200 then we trained our 10000 images, here we have shown a sample of that result:

```
Epoch 1/20
5/5 [==============================] - 5s 939ms/step - loss: 0.6221 - accuracy: 0.8258 - val_loss: 0.2254 - val_accuracy: 0.9296
Epoch 2/20
5/5 [==============================] - 4s 898ms/step - loss: 0.5183 - accuracy: 0.7936 - val_loss: 0.1974 - val_accuracy: 0.8967
Epoch 3/20
5/5 [==============================] - 4s 889ms/step - loss: 0.2011 - accuracy: 0.9150 - val_loss: 0.1111 - val_accuracy: 0.9718
Epoch 4/20
5/5 [==============================] - 4s 892ms/step - loss: 0.0899 - accuracy: 0.9769 - val_loss: 0.0781 - val_accuracy: 0.9765
Epoch 5/20
5/5 [==============================] - 4s 891ms/step - loss: 0.1200 - accuracy: 0.9278 - val_loss: 0.9320 - val_accuracy: 0.7371
Epoch 6/20
5/5 [==============================] - 4s 896ms/step - loss: 0.3839 - accuracy: 0.9586 - val_loss: 0.0805 - val_accuracy: 0.9812
Epoch 7/20
5/5 [==============================] - 4s 886ms/step - loss: 0.1143 - accuracy: 0.9586 - val_loss: 0.0680 - val_accuracy: 0.9718
Epoch 8/20
5/5 [==============================] - 4s 898ms/step - loss: 0.0671 - accuracy: 0.9753 - val_loss: 0.0995 - val_accuracy: 0.9624
Epoch 9/20
5/5 [==============================] - 4s 902ms/step - loss: 0.0288 - accuracy: 0.9933 - val_loss: 0.2296 - val_accuracy: 0.9296
Epoch 10/20
5/5 [==============================] - 4s 896ms/step - loss: 0.0800 - accuracy: 0.9728 - val_loss: 0.3576 - val_accuracy: 0.8685
Epoch 11/20
5/5 [==============================] - 4s 890ms/step - loss: 0.2450 - accuracy: 0.9008 - val_loss: 0.0762 - val_accuracy: 0.9765
Epoch 12/20
5/5 [==============================] - 4s 895ms/step - loss: 0.0568 - accuracy: 0.9589 - val_loss: 0.0667 - val_accuracy: 0.9624
Epoch 13/20
5/5 [==============================] - 4s 888ms/step - loss: 0.0672 - accuracy: 0.9800 - val_loss: 0.0551 - val_accuracy: 0.9765
Epoch 14/20
5/5 [==============================] - 4s 846ms/step - loss: 0.0650 - accuracy: 0.9742 - val_loss: 0.0416 - val_accuracy: 0.9906
Epoch 15/20
5/5 [==============================] - 4s 895ms/step - loss: 0.2681 - accuracy: 0.9161 - val_loss: 0.3049 - val_accuracy: 0.8451
Epoch 16/20
5/5 [==============================] - 4s 889ms/step - loss: 0.1667 - accuracy: 0.9892 - val_loss: 0.1029 - val_accuracy: 0.9718
Epoch 17/20
5/5 [==============================] - 4s 896ms/step - loss: 0.0869 - accuracy: 0.9586 - val_loss: 0.0694 - val_accuracy: 0.9812
Epoch 18/20
5/5 [==============================] - 4s 848ms/step - loss: 0.0874 - accuracy: 0.9472 - val_loss: 0.1569 - val_accuracy: 0.9343
Epoch 19/20
5/5 [==============================] - 4s 891ms/step - loss: 0.0362 - accuracy: 1.0000 - val_loss: 0.0574 - val_accuracy: 0.9906
Epoch 20/20
5/5 [==============================] - 4s 852ms/step - loss: 0.0837 - accuracy: 0.9337 - val_loss: 0.1115 - val_accuracy: 0.9624
```

Figure 6.14: Violence Detection Training Result

Here the accuracy of 0.9337 or 93.37 percentages and we lose 0.0837 or 8.37 percentages. Also the validation accuracy is 0.9624 or 96.24 percentages and the loss is 0.115 or 11.15 percentages. After this we tested with altered epoch and observed the changed result as well.

Now Lets See Our one of sample image for the test result

Next we applied or tested our code on the image.

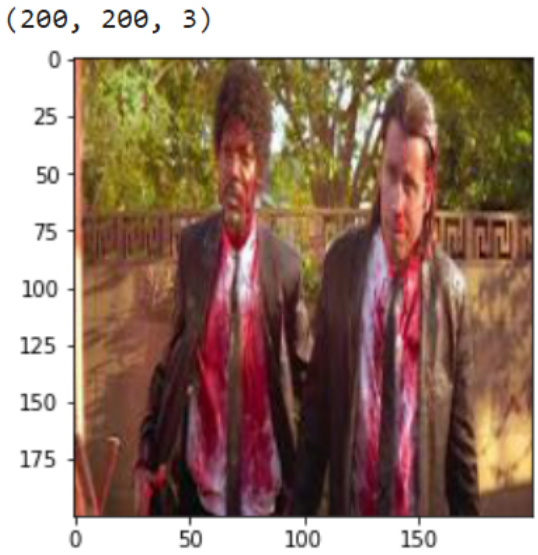


Figure 6.15: Violent image sample

Safe: 16.161513525262 percentages
Unsafe: 83.838486474738 percentages
Following the nudity images here also our model predicts the category of the image perfectly as not safe.

Similar to the Nudity detection we have used “Model Accuracy” and “Model Loss” differently for both training and Validation data to get a proper idea about both of them. We also get some predicted value from the test images that used. The “Model Accuracy” and “Model Loss” curve for both Nudity and Violent images our CNN model is provided below:

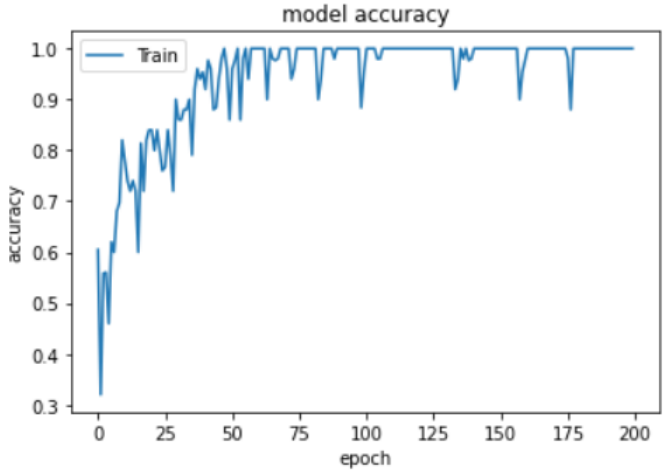


Figure 6.16: Violent Model Accuracy Curve for Train data

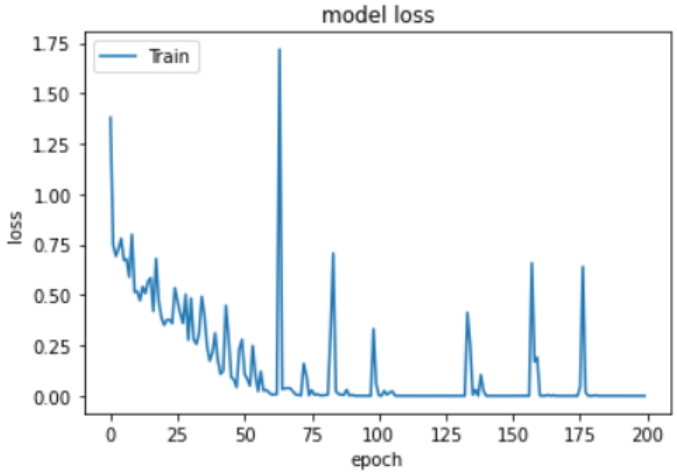


Figure 6.17: Violent Model Accuracy Curve for Train data

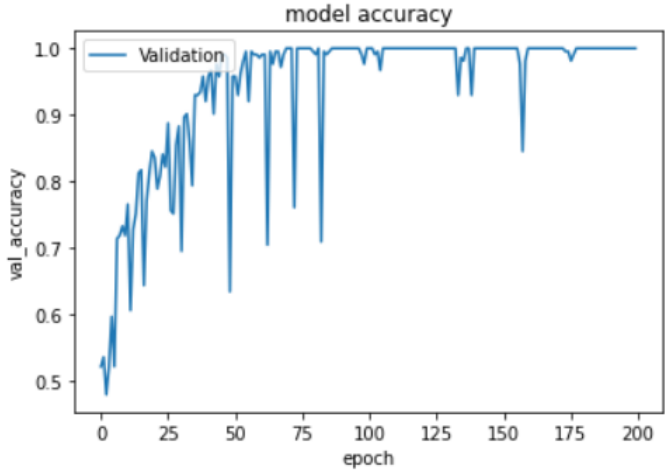


Figure 6.18: Violent Model Accuracy Curve for validation data

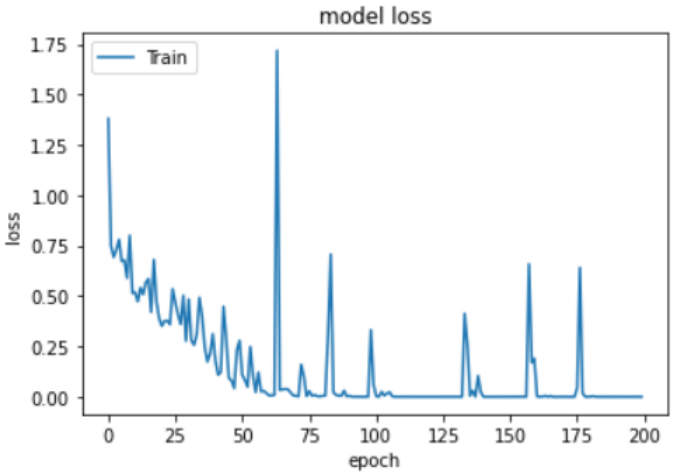


Figure 6.19: Violent Model Accuracy Curve for validation data

So this was all about our result analysis. As we can see out=r model has a very accuracy level and it happened because we kept our model very light weighted. When it is a part of the extension, by looking at the accuracy curves we can say that the detection accuracy would be at a high rate

Chapter 7

Conclusion and Future Works

7.1 Conclusion

In this paper we proposed a system that will detect and warn about the adult image and violent image to prevent the children from watching it. Parents especially always have to worry about their children in this era of open internet. We have brought our system to support them. We also want to support the children to grow up with healthy mentality and enjoy their childhood. By using the cloud hosting our system will work faster as it will have to bear less weight of the algorithms. Moreover this extension will work as a guard and the entire browser will be under a strict supervision. Whenever the user or the children starts surfing a web page, our web scrapper downloads all images, feeds it to the trained machine learning algorithm(CNN) and upon observing the result further actions will be taken. Since it is an extension parents will have an easy control over it as it can easily be enabled or disabled according to the need. In this way a lot of parents who suffers from insecurity due to the availability of mobile devices in the hand of the children. And now in this pandemic it has become even easier for the children to carry one. So to remove this insecurity our system is the most preferable.

7.2 Future Works

Keeping all of the above work aside, there are still a lot of room for improvements. We have seen that our algorithm can miss the web images in recent times. This images are not as simple as our system has been trained with. Like sometimes there is an ad in it with some product's images that are being put in the websites. And the usual image that are being trained as a nudity image don't match with this images. As a result some images does not get recognized by the System. So there we can improve that. Also the biggest improvement left for our work is the making of extension. Since our mail goal was to make a full fledged browser extension where all the things will be put it together like the database for storing images, algorithm to test it and blur it and the main extension file. Additionally the extension need to be hosted in a clod space and create an API so that by simply calling the API from our code the Image processing algorithm starts its work. Moreover, with the pace the world is changing, the system of advertisement and other image posting is changing we will be needing to keep up to date so that the percentage of the better detection increase and our customers get the best experience of the safe and fresh

web contents. Lastly, we have to improve our algorithms for videos as well. Cause it is such a thing that can contain lots of thing at the same time. As the kids watch videos a lot so working with unsafe videos that might hamper the mentality of kids will be our one of the most prioritized field. To conclude, our project have a lot of prospect and lots field to work with so that with the aim that we have started will be fulfilled.

Bibliography

[1] M. Khan, M. A. Tahir and Z. Ahmed, "Detection of Violent Content in Cartoon Videos Using Multimedia Content Detection Techniques," 2018 IEEE 21st International Multi Topic Conference (INMIC), Karachi, 2018, pp. 1-5, doi: 10.1109/INMIC.2018.8595563.

[2] M. B. Garcia, T. F. Revano, B. G. M. Habal, J. O. Contreras and J. B. R. Enriquez, "A Pornographic Image and Video Filtering Application Using Optimized Nudity Recognition and Detection Algorithm," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 2018, pp. 1-5, doi: 10.1109/HNICEM.2018.8666227.

[3] Finkelhor, D., Turner, H., Hamby, S. (2011). National Survey to Children's Exposure to Violence: Questions and Answers About the National Survey of Children's Exposure to Violence. PsycEXTRA Dataset. doi:10.1037/e691452011-001.

[4] H. Zheng, H. Liu and M. Daoudi, "Blocking objectionable images: adult images and harmful symbols," 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763), Taipei, 2004, pp. 1223-1226 Vol.2, doi: 10.1109/ICME.2004.1394442.

[5] C. Santos, E. M. dos Santos and E. Souto, "Nudity detection based on image zoning," 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, 2012, pp. 1098-1103, doi: 10.1109/ISSPA.2012.6310454

[6] M. Chen and A. Hauptmann. (2009). MoSIFT: Recognizing Human Actions in Surveillance Videos. CMU-CS-09-161

[7] A. Gangwar, E. Fidalgo, E. Alegre and V. González-Castro, "Pornography and child sexual abuse detection in image and video: A comparative evaluation," 8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017), Madrid, 2017, pp. 37-42, doi: 10.1049/ic.2017.0046

[8] D. Cai, X. He. J. Han, Speed up kernel discriminant analysis. The VLDB Journal 20, 21–33 (2011). https://doi.org/10.1007/s00778-010-0189-3.

[9] Support Vector Machine Retrieved from https://www.analyticsvidhya.com/blog/2017/09/underst
support-vector-machineexample-code

[10] Support Vector Machine Retrieved from https://stats.stackexchange.com/questions/23391/how-does-a-support-vectormachine-svm-work

[11] Logistic Regression Retrieved from https://machinelearningmastery.com/logistic-regression-for-machine-learning/

[12] Random Forest Retrieved from https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing

[13] Sun, Y., Xue, B., Zhang, M., Yen, G. G., Lv, J. (2020). Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. IEEE Transactions on Cybernetics, 50(9), 3840-3854. doi:10.1109/tcyb.2020.2983860.

[14] Peng, Joanne Lee, Kuk Ingersoll, Gary. (2002). An Introduction to Logistic Regression Analysis and Reporting. Journal of Educational Research - J EDUC RES. 96. 3-14. 10.1080/00220670209598786.

[15] Ali, Jehad Khan, Rehanullah Ahmad, Nasir Maqsood, Imran. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues(IJCSI).9.

[16] Convolutional Neural Network. (n.d.). Retrieved January 10, 2021, from https://www.sciencedirect.com/topics/engineering/convolutional-neural-network.

[17] J. Nam, M. Alghoniemy and A. H. Tewfik, "Audio-visual content-based violent scene characterization," Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269), Chicago, IL, USA, 1998, pp. 353-357 vol.1, doi: 10.1109/ICIP.1998.723496.

[18] Social media, online gaming and keeping children safe online. (2021, January 26). Nidirect. https://www.nidirect.gov.uk/articles/social-media-online-gaming-and-keeping-children-safe-online

[19] C. (2018, August 22). Why Safe Browsers Don't Work. Netsanity. https://netsanity.net/why-safe-browsers-dont-work/

[20] Wang, J., Perez, L. (n.d.). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. http://cs231n.stanford.edu/reports/2017/pdfs/300.

[21] Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., Le, Q. V. (2020, November 5). Unsupervised Data Augmentation for Consistency Training. arXiv.org. https://arxiv.org/abs/1904.12848.

[22] Kober, T., Weeds, J., Bertolini, L., Weir, D. J. (2020). Data Augmentation for Hypernymy Detection. Data Augmentation for Hypernymy Detection, abs/2005.01854. https://arxiv.org/abs/2005.01854.

[23] Cubuk, Ekin Zoph, Barret Mane, Dandelion Vasudevan, Vijay Le, Quoc. (2019). AutoAugment: Learning Augmentation Strategies From Data. 113-123. 10.1109/CVPR.2019.00020.

[24] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 International Interdisciplinary PhD Workshop (IIPhDW), 2018, pp. 117-122, doi: 10.1109/IIPHDW.2018.8388338.

[25] Donges, N. (2021, May 17). A Guide to RNN: Understanding Recurrent Neural Networks and LSTM. Built In. https://builtin.com/data-science/recurrent-neural-networks-and-lstm?fbclid=IwAR2NAZ-5MsEo7oFmI9D7M6WncrS-MJGo548FbpWeXpYaADWnwUGBr91DPk4

[26] Pedamkar, P. (2021, May 13). Recurrent Neural Networks (RNN). EDUCBA. https://www.educba.com/recurrent-neural-networks-rnn/

[27] Gikunda, A. (2021, April 23). Post author: Amos Gikunda. Pros an Cons. https://prosancons.com/education/pros-and-cons-of-logistic-regression/

[28] Rizvi, M. S. Z. (2020, October 19). Learn Image Classification on 3 Datasets using Convolutional Neural Networks (CNN). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/