Smart Image Sharing Platform

A PROJECT REPORT

submitted by

ANKIT VADEHRA 12BCE0282

ROHAN KUMAR 12BCE0622

in partial fulfillment for the award

of the

B.Tech

degree in

Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT University, Vellore

Vellore, Tamil Nadu – 632014

School of Computing Science and Engineering

DECLARATION

We hereby declare that the project entitled **Smart Image Sharing Platform** submitted by us to the School of Computing Science and Engineering, VIT University, Vellore in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is a record of bonafide work carried out by us under the supervision of **Prof. Manoov R., Assistant Professor (Senior)**. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Ankit Vadehra
12BCE0282

Rohan Kumar
12BCE0622

School of Computing Science and Engineering

CERTIFICATE

The project report entitled **Smart Image Sharing Platform** is prepared and submitted by **Ankit Vadehra (Register No: 12BCE0282)** and **Rohan Kumar (Register No: 12BCE0622)**. It has been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** in VIT University, Vellore, India.

Prof. Manoov R
GUIDE
Assistant Professor (Senior)

Internal Examiner(s) **External Examiner(s)**

ACKNOWLEDGEMENT

The completion of this project couldn't have been possible without the dedicated, sincere efforts and guidance of various faculty and staff members of the School of Computing Science and Engineering at VIT University, Vellore. We acknowledge our gratitude to all of them. This acknowledgement however would be incomplete without specific mention as follows:

We would like to acknowledge with much appreciation the crucial role of our guide **Prof. Manoov R., Assistant Professor (Senior), SCOPE** for his encouragement and guidance in facilitating the completion of this training.

Furthermore, we wish to acknowledge our deep gratitude to **Dr. R. Senthil Kumar, Head of the Department** for his guidance. We are thankful for the staunch support throughout the project duration by **Dr. Arunkumar T., DEAN, SCOPE** that helped us complete it successfully.

We would also like to take this opportunity to appreciate VIT University, Vellore for allowing us to undertake the final semester project as a part of the B.Tech Computer Science and Engineering curriculum, which allowed us to learn a lot about research problems in the Computer Science domain.

Contents

List of Figures

Abstract

We live in a golden age of digital technology like Social Networking sites, massive knowledge base like Wikipedia and it has never been easier to have access to high quality digital media like photography, videography and audiography. Moreover, it has never been easier to produce our own digital content and share it with everyone globally. Everyday hundreds of people are turning into self-taught amateur photographers as is clear by the rise of photo sharing and their respective platforms like Imgur, Instagram, 500px, Twitter, Facebook, Deviantart. One major concern present in online photo sharing community is the presence of adult content like nudity on Image sharing platforms. This makes it very difficult for many avid photographers to join a community that has no control over the media being promoted on their website. Since it is very difficult to manually maintain the image quality online we propose an image sharing platform that uses Machine Learning models to show images someone might like, along with liking and sharing features. The system uses Deep Learning to generate tags for images uploaded and then uses different models to test for adult content and only allows uploading of clean and safe images. Apart from keeping the image quality in check we also plan to put constraints of the type of comments that can be posted for images, resulting in a smart, safe and engaging platform to share images online. The platform provides a full fledged web user interface for users to make it interesting to share images.

Index Term: *Machine Learning, Image Sharing, Spam Prevention, Deep Learning, NLP, Tag Generation, Image Captioning*

Chapter 1

Introduction

This chapter enlists the literature survey as well as the significance of our project. We go through existing systems, along with their gaps and lags, that we try to rectify with our system. We justify the use and roles of a Smart Image Sharing Platform. We build a Smart Image Sharing platform that allows both amateur and expert photographers to intuitively share photographs clicked by them.

1.1 Theoretical Background

The onset of a golden digital revolution has provided us with a wide variety of technologically advanced toys. High Resolution Digital Cameras comes under the discussed advanced toys. Today, everyone has access to such digital cameras which make it easy to share photographs online. As a result of this, we have seen an exponential rise in the number of amateur photography pages. People take pride in their work and like to share said work to a large audience of millions globally.

Social networking websites such as Facebook, Instagram and Twitter are great for global interaction but defeat the purpose of discussions ranging solely on photography giving rise to internet stalking and a lot of limitations in the form of age and privacy restrictions. These platforms are not designed for a community with laser sharp focus on photography. While photo sharing communities such as Imgur, Giphy, 9gag have spawned recently, they do not provide a holistic platform for avid photographers.

To overcome this, we aim to expand the work done by these online photo sharing communities. Our core focus is to provide a clean platform for photographers to share their work and allow the global community to converse about it. One of the major problems faced is high amounts of profanity on

such platforms due to users generating and uploading content. We aim to smartly filter out such obscene content from the platform using Image Processing techniques along with a ratings system where users can upvote or downvote content uploaded to the platform. Additionally, to filter out obscene text descriptions and comments entered by the users, we use keyword matching techniques based on a dictionary.

1.2 Motivation

The motivation behind designing a Smart Image Sharing Platform is a compelling need to finally use and have a system that is simple, intuitive and free from obscene content. Additionally, with the rise in Deep Learning methods in Computer Vision for automatic generation of tags and captions for images, the project gives us a chance to include state of the art research ideas into a real world application. There are also multiple programmatic approaches that try to detect nudity in images. We perform image processing techniques on the image like YCbCR manipulation that tries to detect the percentage of pixels containing skin in an image, and detect whether it could be considered profane or not.

Our main motivation is to help the global photography community, by allowing them to share their work in a clean environment designed specifically for them. Personally we are avid photographers and one of the main problems we have faced in online photography sharing sites is the presence of a large amount of obscene adult content, along with profane comments that hugely diminish the posters or submitters interest in sharing their work. We also find that usual sites lag in retaining user interest as a result of which people get bored of a particular site and tend to move to other locations on the Internet.

1.3 Aim of the Proposed Work

Our main aim is to design a simple and scalable website for photography enthusiasts as well as creative developers that is clean and free from profane content. To introduce and accomplish user engagement and interactivity we would add rating, commenting system and trend-ranking feature along with automated image captioning and image similarity features that are bound to increase the churn rate of every visit.

Through our project, Smart Image Sharing Platform, we wish to overcome the gaps we have found personally while sharing our digital content online.

We wish our system to be a one-stop destination for all photography enthusiasts. The main aim of the smart image sharing project is to attain the ultimate goal of designing an image sharing system that is free from profanity and has features to increase human interactivity. We know that this is not entirely possible since any automatic system can never be free from giving us false-negatives and false-positives, but we try testing and tuning different programmable methods to ultimately give us the best possible results.

1.4 Objectives of the Work

It was realised that there are multiple image sharing platforms but there is no sanity check being performed on the content being uploaded. This gave rise to the realization that it is not manually possible to scan through every image by some community moderator marking the images as spam or safe. Hence we plan to make a smart system that can realize automatically(using Machine Learning and NLP) whether the content being uploaded is clean or not. Along with this, we plan to improve communication between our system and the users, hence improving engagement. To see this in action, we design interactive features like *Image Captioning and Top Trending images.*

Our main objective is to design a smart image sharing platform. We want to enforce and define the word "smart". Our objective can be broken down into the following capabilities of our platform:

- Allow or reject submitted image, based on the content submitted. As a result of which, we hope to block adult content.
- Allow or Reject the comments posted, based on the content.
- Allow users to like or dislike the submitted image.
- Significantly calculate the trending image rating and allow the users to view trending images.
- Use deep learning to generate captions for the said images, and also show similar images to the users.

1.5 Report Organization

The scope of this document is to describe the specific working and execution of the smart image sharing platform software. This would include Hardware

requirements, Software constraints, and project assumptions. Any specific information which is part of the standards or technology used to define these requirements, constraints, and assumptions are within the scope of this document. The report starts with the Introduction and the motivation behind our work, moving to the Literature Survey along with gaps in existing systems. We provide a description of our system along with complete specifications and the requirements. Finally we provide the Results and analysis along with the test cases ending with the conclusion.

Chapter 2

Literature Survey

There are various image sharing platform present today and each one comes with their own multiple features. Along with image sharing there are various other sites that focus on ranking and rating content using different techniques. We discuss and talk about all of our modules.
We also use this chapter to explain the services we wish to implement in our project as a service, like, Image Captioning, image trend ranking and also the image processing models for adult content check.

2.1 Existing Systems

As mentioned previously in the report, photography has become much more than a hobby these days[1][2]. There is tagging locations and friends, selfies, photo-bombs and what not. One thing which is certain is that fact that image sharing has become quite famous. Take into account web applications such as Instagram, Facebook, Snapchat, 500px, Imgur, 9gag and Devianart. These are just some of the applications that are focused on providing image sharing. Unfortunately each application lags in some aspect which we aim to achieve using our system.

2.2 Gaps in Existing Systems

One area in which all these applications fail spectacularly is the capability to block adult content. None of these applications have checks on the comments and the content being uploaded, allowing profane content to be uploaded which hugely diminishes the photo uploading and sharing concept. Apart from profanity check, applications such as Facebook, Twitter and Instagram do not have features to calculate the ranking of the uploaded images. Since

they are full blown social networks with heavy profile restrictions, they are not focused on calculating the trending images. Currently Facebook is introducing automatic caption generation on images for people with deforming visibility, allowing them to understand what the content is in the uploaded digital content.
Our system aims to achieve all these gaps by providing support for image and comment profanity check, along with proper statistical image ranking generation based on the like and dislike feature. We also use state of the art Deep Learning models to provide features like Image Similarity checking and Image Caption generation.

2.3 Existing Services and their Use

Our project focuses on various services that have previously been designed as a research project and havent been used as an application system. We explain the literature survey of each and every model.

2.3.1 Image Commenting and Ranking

Many applications today use the main feature of commenting upon an image as well as logging the likes and the dislikes of an image to engage the global community as well as to promote interaction and gather the popularity of the system.

2.3.2 Profanity Check

Image profanity is a service that is mostly worked in the Image Processing community in the form of Research ideas. We wish to take two Image Processing models and incorporate a hybrid system that checks images and only allows images that are clean to be saved in our system. As Jorge A. Marcial-Basilio explains, image processing models can be used to convert the RGB space to YCbCr and then perform histogram analysis on the image and then calculate the amount of adult content pixels[3].
We also use an exact keyword matching system to rule out the profane comments. Many online systems avoid comments profanity check completely like 9Gag and imgur. Although, some sites like Facebook often show a disclaimer asking that certain comments might be sensitive in nature, hence users often have to click a button to see the sensitive comments.

2.3.3 Calculate Image Content Trend

Trending content is not a new concept. Overtime, all social networking domains have incorporated some form of image trends. Using a distinct and statistically significant model to calculate the ranking is not only important but it is immensely necessary. Many sites often destroy this part of the design, as Evan Miller describes, often wrong statistical methods are applied, that do not tend to calculate ranking in a proper manner[9]. We use a new module designed by Reddit that can correctly identify and calculate the rankings based on the number of up-votes and down-votes. We talk about the existing design systems and our system in the later chapters explaining the mathematical models.[7][8]

2.3.4 Image Captioning

This is a relatively new and an interesting research idea that uses Deep Learning to perform machine translation and computer vision task to automatically perform image annotation. O Vinyals, in his paper Show and Tell describes how we can use massive ANN to caption images[9]. We save the image captions in a database that can be easily accessed using a search query. Deep Learning uses massive Neural Nets trained on unstructured data to make sense of new data[5][6]. We provide detailed description of the service in the subsequent chapter.

Chapter 3

Overview of the Proposed System

This chapter describes the control and overall execution of the project in a systematic manner. This section mentions the control structure, sequence diagram and data flow sequence. Moreover, we explain why a particular decision to design the project in a specific manner was taken.

3.1 Introduction

A carefully designed project specification is winning half the battle. When we started with the documentation and design stage of our project, we were presented with multiple ideas, and multiple solutions to each and every problem. After checking whether the solutions were feasible, we ended up with the Requirement for our project. After obtaining the requirement, we zeroed in on the exact deliverables of our project. After this careful planning resulted in coming up with the design specification of our system that is feasible and helpful in maintaining our system in the long run.

3.2 Architecture and Module Description

This section describes the control and execution of a project in a formal manner. This section mentions the literature survey, control structure, sequence diagram and data flow sequence. This section enlists all the structure diagrams for our project.

3.2.1 Smart Image Sharing Platform Sequence

This subsection explains the steps that are taken from start to finish inside the Image Sharing System through an activity diagram. The user enters an image and the system checks whether the image is profane or not. If it is clean, then we upload the image and add it in our database. If an image is considered as offensive, it goes for a manual check by the administrator, and add the safe images. We have a trending section that calculates the lower bound of the Wilson score interval, based on the number of Likes and Dislikes. The system also checks whether the comment is clean or not, and adds the clean comments.

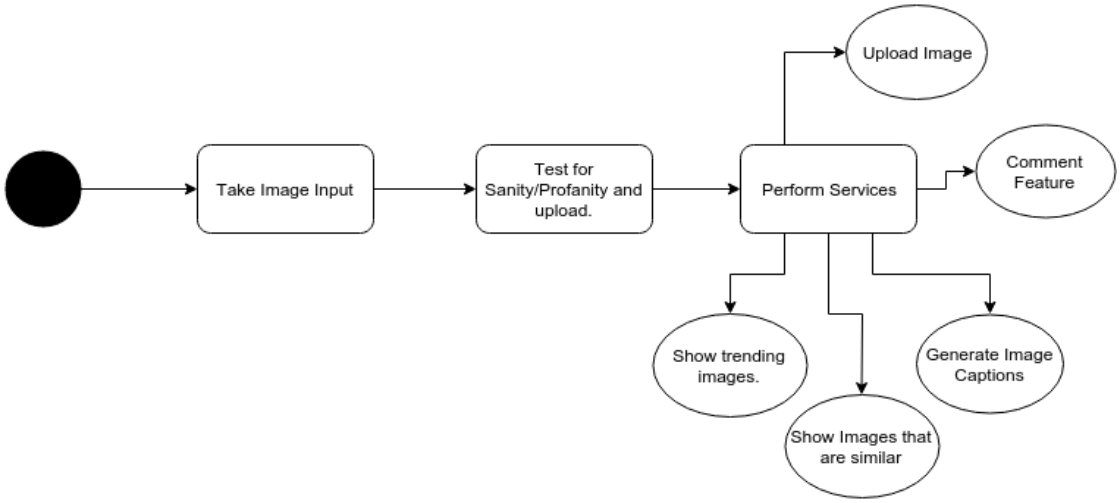

Figure 3.1: Sequence Diagram

3.2.2 Architecture Description

The architecture chosen here is the client-server approach as we have a central main server which handles all system tasks like taking appropriate steps according to user actions which are clients. The central server performs all the tasks of uploading, tag generation and providing the top trending images and similar images to client likes. The server also has the main task to check the images and comments for profanity and block wrongful content.

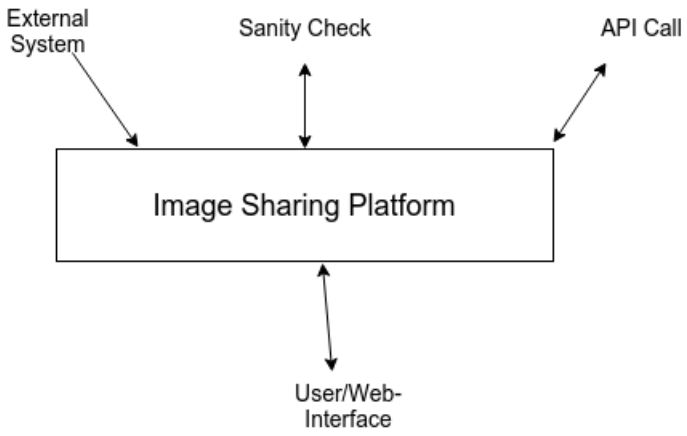

Figure 3.2: Architecture Diagram

3.2.3 System Break Down Structure

We have a centralized structure because it is highly sequential and all tasks follow the preceding process directly, and there are different routes of reaching the result hence the centralized control structure fits best for our Image sharing platform.

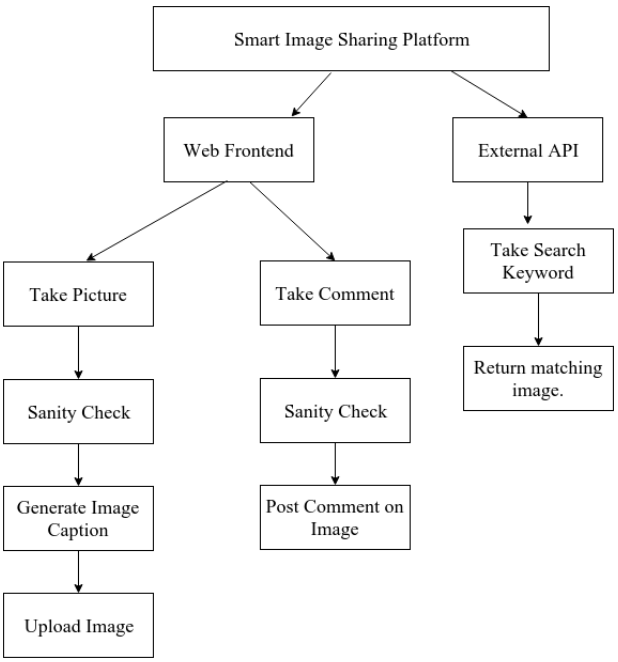

Figure 3.3: System Break Down Diagram

3.2.4 Data Flow Diagram

The data flow diagram shows how data flows from the users end, through all the processes in our backend system. We show how the data is passed in

the form of digital images, textual comments and likes/dislikes. The backend system uses the trending feature that uses the number of likes and dislikes as the input data. The image profanity check uses 2 Image processing algorithms to detect whether or not the images are clean or profane. The image caption system uses deep learning on GoogleNet and MSCOCO trained dataset to generate automatic image captions.

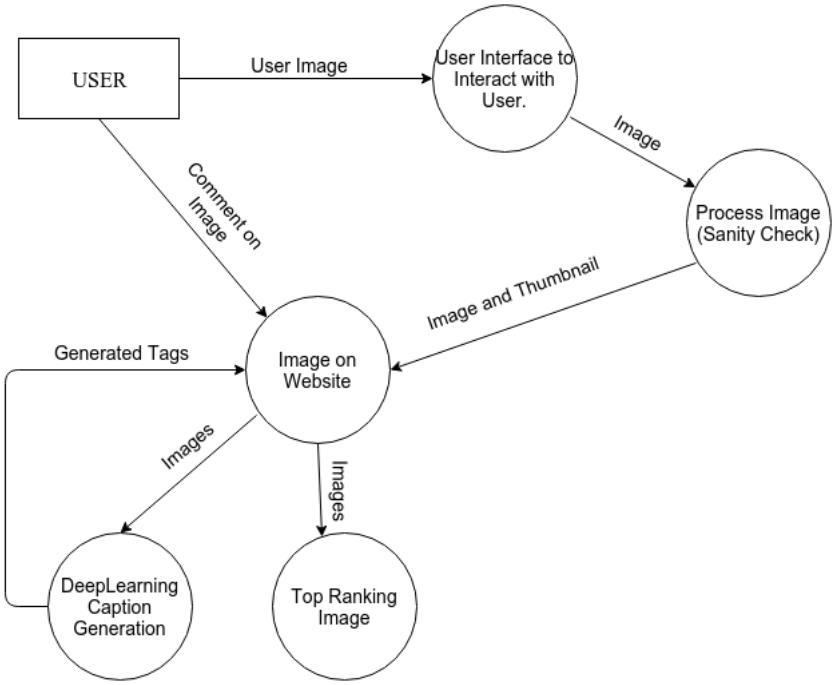

Figure 3.4: Data-Flow Diagram

3.2.5 System Analysis

3.2.5.1 Web User Interface

The Smart Image Sharing Platform is an efficient project to provide image sharing capability for multiple users along with automatic captioning and multiple sanity checks. It also provides an API to pass image URL's and return whether or not the image is safe along with automated generated captions. It can be accessed through a web interface that provides an upload section and then a board to see all uploaded images(via thumbnails). It also has a section to show the highest ranked images.

Purpose: Providing a user-interface for the system so as to easily allow uploading and viewing of images and comments.

Input: The input is an image or textual-comments.

Processing: Sanity checks on the uploaded content(input data).

Output: Uploaded image URL, if it is clean.

3.2.5.2 Image Profanity Check

The sanity check module is responsible for checking via programmatical, Image Processing methods whether an image is clean or not.

Purpose: The core of the system. Provides a way to check for image profanity. Has to be selected via multiple tests to see what works best.

Input: Uploaded Image.

Processing: Image Processing methods for sanity check.

Output: Returns the result of the profanity check on the Images.

3.2.5.3 Comments Profanity Check

Allow only clean comments to be uploaded.

Purpose: Clean comments uploading.

Input: Textual user comments.

Processing: Check the comments words against dedicated profane word list.

Output: Returns the result of the profanity check on the comments.

3.2.5.4 Top Ranked Images

Useful to allow user interaction. The Thumbs-Up/Thumbs-Down ranking is calculated using the probabilistic model of Chess-Ranking method.

Purpose: Provide ranked images in a descending order.

Input: Image ranking from the database.

Processing: Calculate the ranking.

Output: Ranking list for web-site printing.

3.2.5.5 Image Caption/Tag generation

Deep Learning model to generate captions for the uploaded image.

Purpose: Allow descriptive labels for the uploaded images.

Input: Input Image.

Processing: Convolutional Neural Networks(CNN) to generate labels.

Output: Image labels/captions/tags.

3.2.5.6 API

Providing an API module for external systems, allowing them to use our website.

Purpose: API support for external applications. SaaS.

Input: URL of an image.

Processing: Profanity Check and Tags Generation.

Output: Image clean or not along with the appropriate automatic generated tags/caption.

3.3 Proposed System Model

3.3.1 Entity Relationships

Our system ensures real-time and split second digital image obtaining and analyzing along with providing complete anonymity to the Users. Hence, we do not require the need to use any DBMS for storing and processing data. We use a simple database to store the meta-data of the file name n a small sqlite3 database.

3.3.2 Object-Oriented System Design

The Class Diagram that can be used to describe our system is:

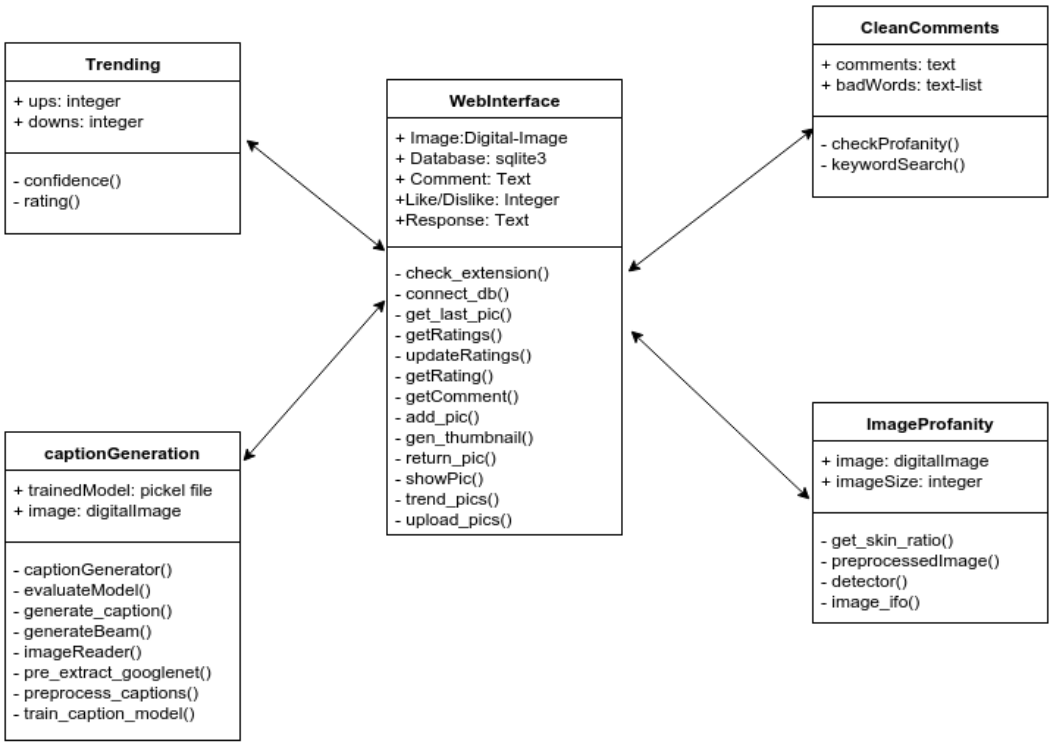

Figure 3.5: Class Diagram

3.3.2.1 Description

Our system has 5 main components/modules, namely the Web/Application User Interface, Trend Rating, Caption Generation, Clean Comments and Image Profanity. These modules are interconnected, as the Web Interface takes in the uploaded image and checks for Image Profanity on the image, uploading it only if it is clean and then performs caption generation on it. The Web Application also takes in the comments and then performs clean comment checking only uploading the comment if it is clean.

3.3.3 Usecase Design

The use case is attached for observing the major components of the system. It shows the stakeholders and the users of our system, and the functions they can perform. It is an easy way to visualize the system along with the main components it is performing.

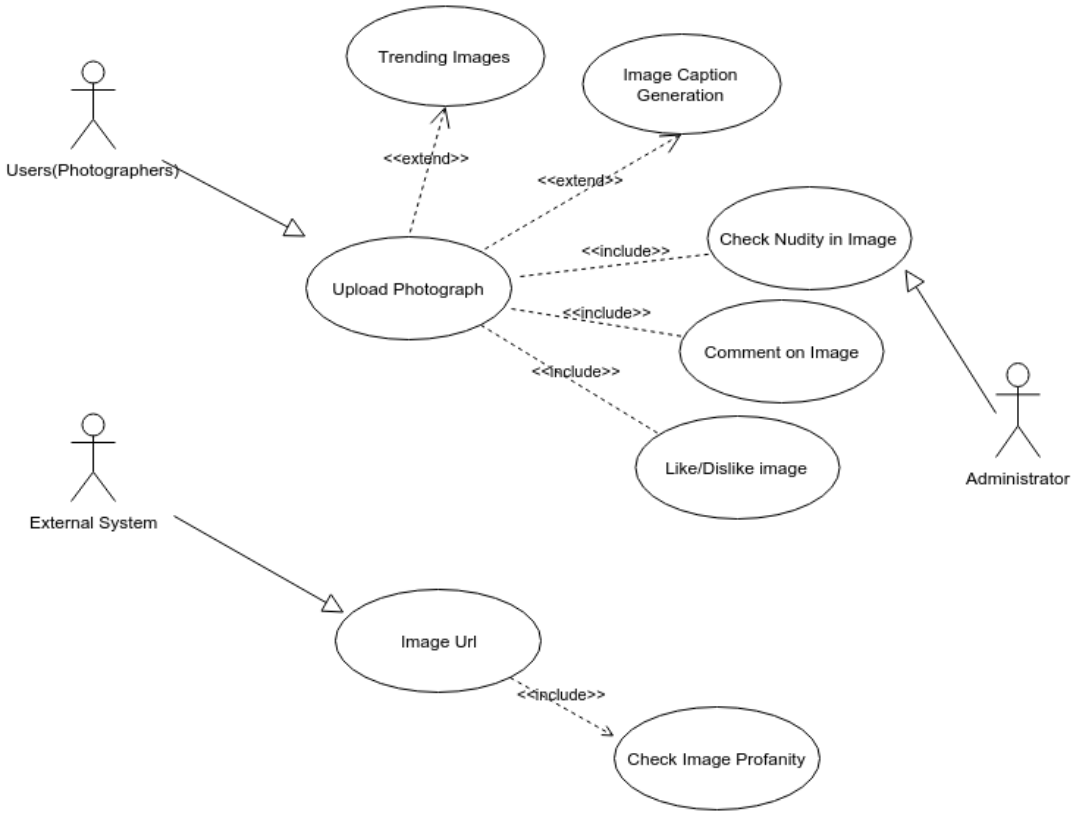

Figure 3.6: Use Case Diagram

3.3.4 Components Detailed Design

This section has all the main modules and components in our system. We explain how their interaction takes place, and how they give us the respective results.

3.3.4.1 Upload Photos

Sequence Diagram

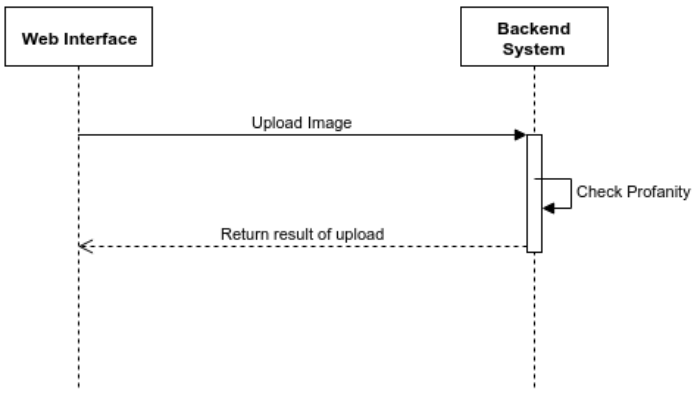

Figure 3.7: Upload Photo Swimlane Diagram

Design Description

Through the web user interface a user can enter the image they wish to upload to our system. This feature will check for the extensions of the image and then passes it to the image profanity check finally providing us with the uploaded image or an error message stating that the image violated our systems profanity check constraint.

3.3.4.2 Trending Image

Sequence Diagram

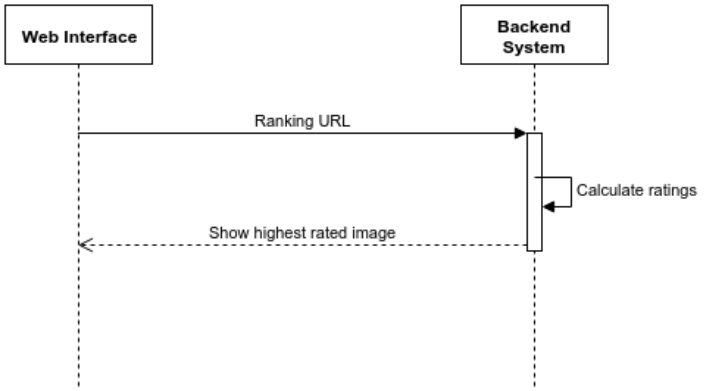

Figure 3.8: Trending Image Swimlane Diagram

Design Description

This system uses the lower bound of the Wilson score interval, that is calculated using the number of up-votes and down-votes on any particular image. After calculating the rating of the images, we return the top 25 images.

3.3.4.3 Commenting Feature

Sequence Diagram

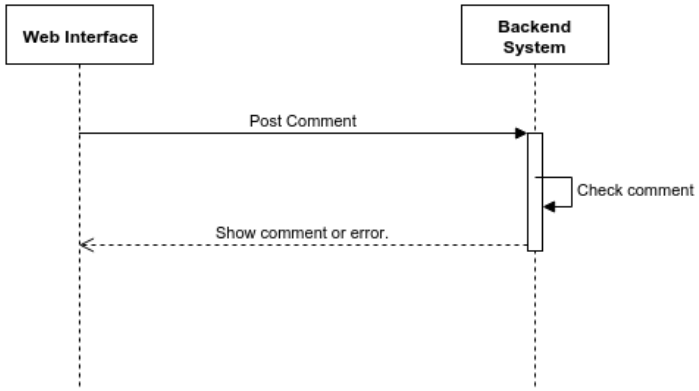

Figure 3.9: Commenting Feature Swimlane Diagram

Design Description

This module takes in the comment for a particular image, and then passes the textual comment to the clean Comment module. The module, performs absolute keyword matching form a corpus of bad words, and returns either the posted image or a message telling that the posted comment violated the profanity constraint of the posters comment.

3.3.4.4 Tag Generation

Sequence Diagram

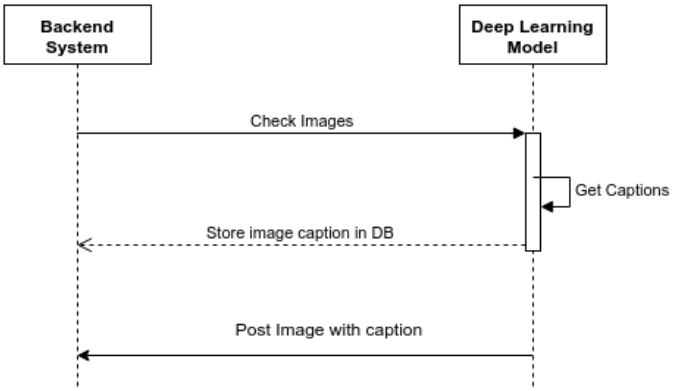

Figure 3.10: Tag Generation Swimlane Diagram

Design Description

This module uses trained corpus's on the Microsoft coco database, and the googleNet feature extraction module to return image captions. The image captions are added in the meta-data database. This is used in the search API, after passing the keyword and then returning all images that have the particular object in the image. This can be used with an external system to return image like giphly or gify.

3.3.5 Methodology Adopted

Considering our project and its various components we plan to move with an incremental approach for our system. An incremental approach would allow us to focus our energy on each individual component as a team and then in the end combine all the system to get ready with the whole project. Also, it reduces the time to ship a MVP (Minimum Viable Product) and then attach each new feature in a subsequent update. Since each update is server side, it would not require any update for the user, who can use the same system along with the new features.

This is a new online system of update that does not require hassles for the

users. New code deployments happens on our applications backend and the users are instantly presented with the updated features. We have different modules in our system, and we explain the working and the maths behind each and every module:

3.3.5.1 Web UI to Upload image

In our application, we'll be using Python, Flask and sqlite database to create a simple web application that can successfully upload images posted by photographers and the distinct users of our websites. We are living in a period where we have moved forward from manually coded and static websites. These days all social networks are dynamic websites like Facebook, Twitter and Imgur. Our website has a backend server component and we deploy our code in increments, including new features in every update.
Python Flask is a small website development tool, that takes in the Javascript and CSS scripts in their appropriate template folders and the python code that decides the content to publish content to the web, based on the different actions of the user. The website opens at the default viewing and uploading page that allows the users to upload the image and also a link that allows viewers to see the trending images. Clicking on an image allows viewers to see the image, along with the option to post comments and post likes and dislikes on it. The posted image and comment is transferred to the Sanity Checker that allows to ensure that no profane content is posted on the website for good measure, to not violate the only constraint of having a clean website.

3.3.5.2 Sanity Checker

This module is responsible for performing profanity check and allowing only clean content to be posted on our website, which is the main aim of our system. The module is broken down into 2 components:
Image Sanity
Image sanity uses image processing algorithm to detect whether an image contains nudity of any form, and block those images. This is done using a simple hybrid image processing system. The system uses a simple algorithm that checks the pixels for a range of RGB colours that can be considered as skin. We programmatically find and detect the skin tones pixel in an image - and it is our heuristic that adult images often have a lot of skin tone pixel. This heuristic may create false positives and we overcome them by allowing the site administrator to manually detect images through actual moderation of blocked content. This not only greatly reduces the work for moderators

but also gives you an easy system to block adult content. It's an excellent measure. This algorithm measures skin tones in the center of the image.
We have tested the image on multiple images and we can state that the algorithm works on light, dark and Asian complexion skin tones. It's main weaknesses with false positives are brown objects like sand and wood and of course it doesn't know the difference between "adult" and "clean" flesh (like face shots). We use a parallel algorithm to detect adult content or pornographic images by transforming from the RGB model color to the YCbCr color model, finally using the skin detection the image is segmented, and we calculate the ratio of the pixels that were detected as containing skin.

Proposed System

We show the proposed system, made up of 4 steps: transform the image from RGB to YCbCr, color model is the first step which gives us the desired output to detect for skin content. Filtering step is performed by detecting the skin pixels with the set-up threshold limit. After that we use a Quantifier to detect and count the number of skin pixels. In the final decision making stage it is decided whether the image is adult or whether it is clean.

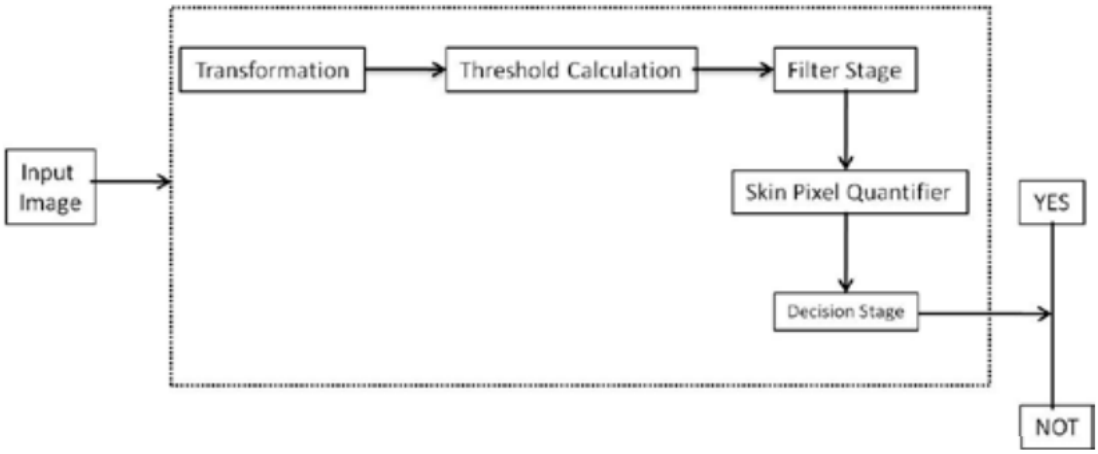

Figure 3.11: Proposed Image Processing System

To calculate the threshold of our Image Processing system, we compare the histogram of the images.

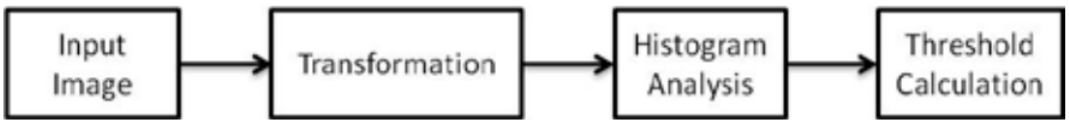

Figure 3.12: Threshold Calculation Algorithm

RGB Color Space

The RGB color space is a color model in which the primary colors are Red, Green and Blue. These colors can be added in different compositions ranging from [0,255] to give us a multiple spectrum of color ranges. The name comes from the initials of the three colors Red, Green, and Blue.

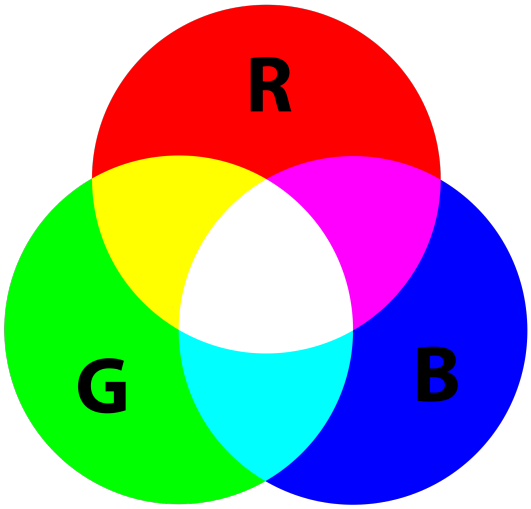

Figure 3.13: RGB Color Space

The RGB color model is primarily used in television and broadcast system to easily provide multiple color ranges. The model senses and represents images in Red, Green and Blue. RGB color model is additive as it encompasses and joins together the 3 colors together to make a final color. To come up with a color in the RGB scheme, Red, Green, and Blue lights are superimposed in the [0,255] range to get the final color. Each of the three individual color acts as a component of that color, and each can have arbitrary intensity, from fully off to fully on, in the mixture. Zero[0] intensity for each color presents us with a highly dark color, ie. black. Whereas the presence of high intensity of each color gives us a lighter shade of white. Each color in the RGB color model is categorized by the presence of each of the red, green, and blue color which can vary from zero to a final high value of 255.

YCbCr Color Space

The YCbCr color space is used in digital videography files and image processing approach among other things. In the YCbCr, luminance or brightness information is depicted by, Y, and the information regarding the color is saved as the two color component differences, Cb and Cr. Here the Cb is difference between the blue color values and a reference constant, and Cr is the difference between the red color and a reference constant. The transformation equation used to derive the YCbCr color space from the RGB color space is shown in

the following equation:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Figure 3.14: RGB to YCbCr Transformation

The corresponding skin cluster is given as

$$Y > 80$$

$$85 < Cb < 135$$

$$135 < Cr < 180$$

Where, $Y, Cb, Cr = [0, 255]$. The system has some problems to find skin of people from different races, the thresholds given above only finds people with white skin, and the second threshold segregates people from distinct regions of the world, whereas some pixels cant be represented as skin pixels.
Comparing the three thresholds

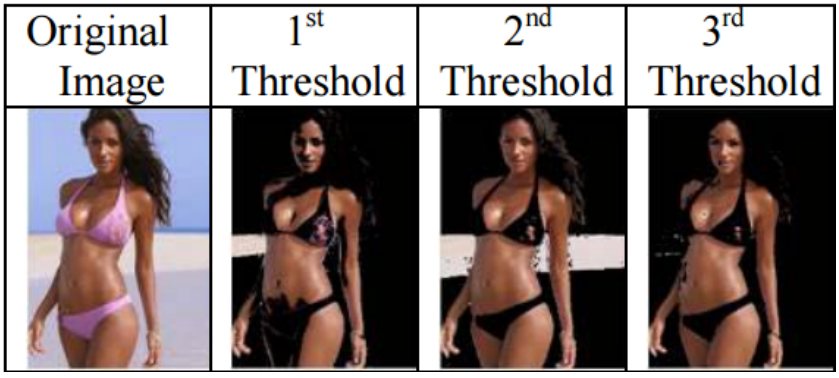

Figure 3.15: Three Threshold Levels

Detecting Skin via Color Histogram

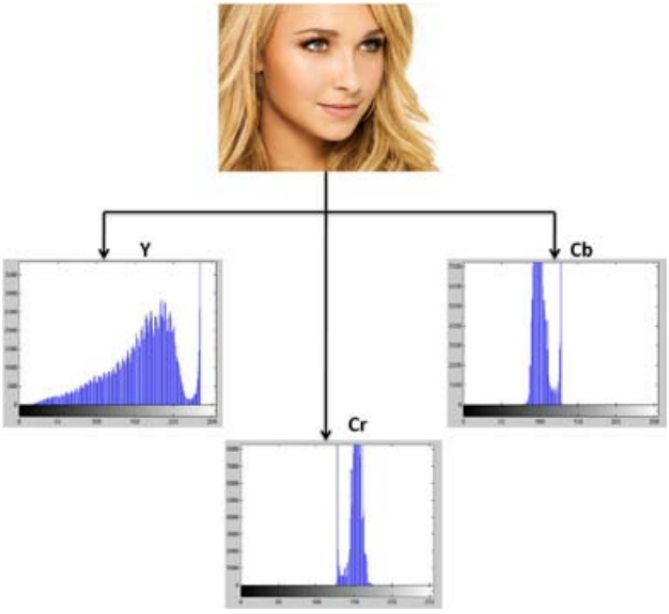

Figure 3.16: White Skin

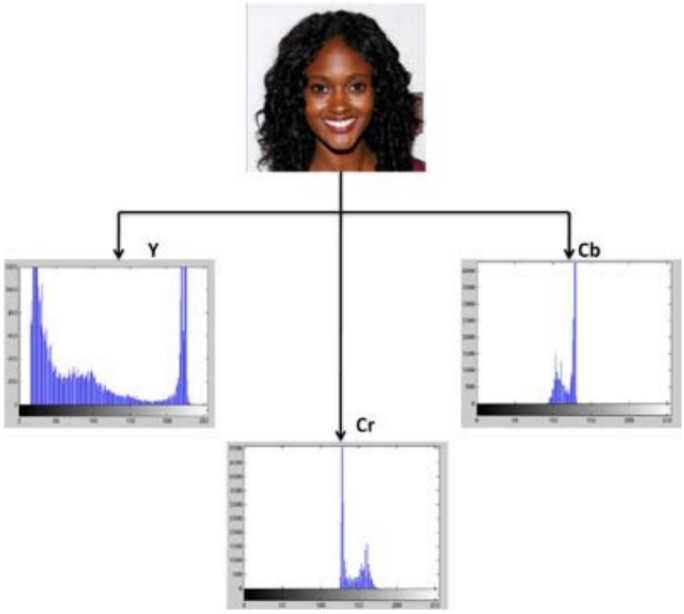

Figure 3.17: Black Skin

Image Classification

We determine if the image has adult content or not by using the, the Pixel Quantifier. It counts the number of pixels that the previous step identified as skin color, then using the classifier, the skin ratio is calculated to find whether we have adult profane content or not.

$$skin\ percentage = \frac{\#\ skin\ color\ pixels}{\#\ image\ pixels\ in\ total} x100$$

Figure 3.18: Skin Percentage Level

We consider an image as adult content by setting the skin ratio threshold as 30 or greater. In the below Table we show some depiction of the classifier when different images are presented to it in different forms:

Input Image	Images Classification		
	Color Model	Proposed Classifier	Skin Percentage
	YCbCr		60.64%
	YCbCr		56.36%
	YCbCr		2.39%

Figure 3.19: Image Classification Level

Comment Sanity

Comment Sanity is an approach that uses exact keyword search in textual content. We obtain a corpus of cuss/profane words used, and we do exact set intersection. We calculate the intersection of the 2 sets and if we obtain an element that is present in the intersection, we can safely classify the posted comment as offensive and block it.

$A = \{Set\ \ of\ \ Profane\ \ words.\}$

$B = \{Set\ \ of\ \ posted\ \ text\ \ content.\}$

$C = A \cap B$

If $C == \oslash$: Comment is Safe.

Else: Comment is Profane.

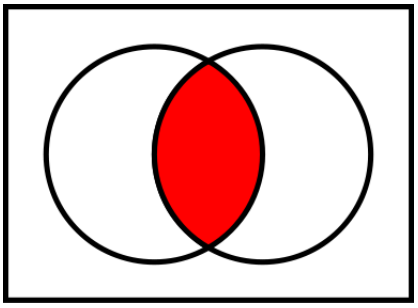

Figure 3.20: Set Intersection

3.3.5.3 Caption generation on Images

Neural Network

Neural Networks or ANNs (Artificial Neural Networks) are machine learning models based on the neurons in the human brain. They consist of layers of neurons to each other. The model is called a perceptron. Neurons are the primary computation units in an artificial neural network. They take inputs, apply an activation function and check the transformation value obtained. Output nodes are set with threshold value weights, and different values are outputted based on different values obtained. Certain activation functions allow neural nets to model complex nonlinear patterns in the data. Synapses or connections are used to exchange data between neurons. They store weights which manipulate the data from one layer to another. ANNs are very good for tasks such as image classification, NLP (Natural Language Processing), Computer Vision and Speech Processing. They have been applied in real world applications like self-driving cars and stock market prediction.

Neural nets come in many architectures such as recurrent neural networks, Boltzmann neural networks, deep neural networks and adaptive neural networks. Some may be generalized while others may be better suited at specific tasks. The most common application for neural networks is pattern recognition however they are also used for other tasks such as time series prediction, signal processing and anomaly detection.

Non-Linear Classification

The main problem the Deep Learning network tries to solve is that of non-linearity in the classifier class. Real world data is often messy and cant be solved by linear probabilistic classifiers. Deep Learning and Neural Networks use layers of massive Hidden Layers that perform transformation on the input space, providing concrete classification lines for classification. One of the main

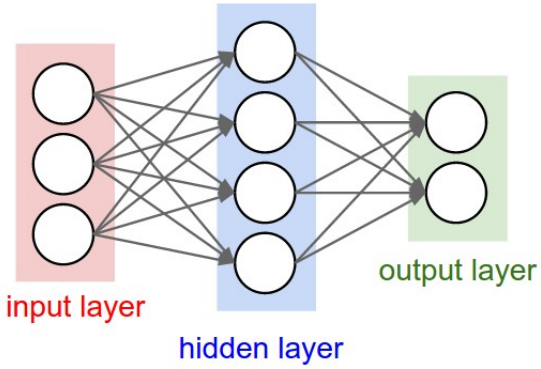

Figure 3.21: Neural Network

problem that can be explained as a non-linearity is the XOR problem.
The XOR Boolean function is a concrete example of non linearity. Depending on the input values provided $X = \{x_1, x_2....x_n\}^T$, the output can either be 0 or 1 and we can classify them in the class A or B.
The truth value for XOR is:

x_1	x_2	XOR	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B

From the following image, It is clear that no straight line exists that could separate the two classes in a linearly fashion.

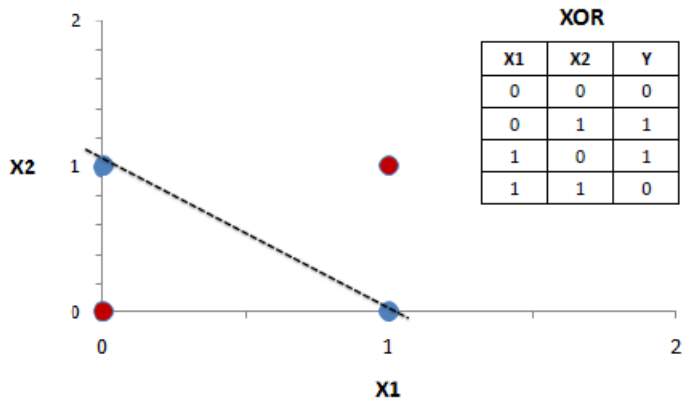

Figure 3.22: Classification Label

Hence we use Neural network(perceptron) with a hidden layer that can help us design a classifier that can help us with the non-linearity concept. The structure is designed as follows, with the appropriate weights and thresholds on each neurons:

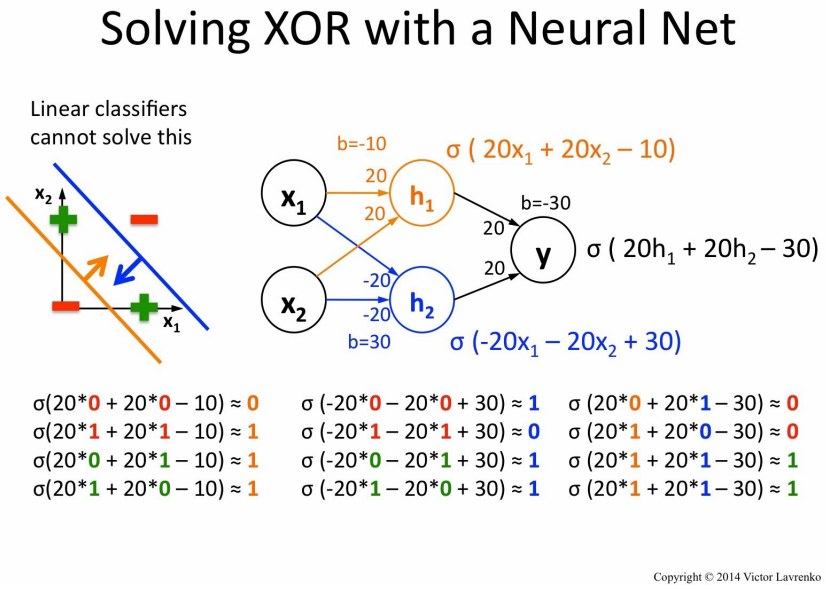

Figure 3.23: Hidden neural layer model for XOR

3.3.5.4 Image Caption System

Deep Learning systems are highly non-linear models. We pass the inserted

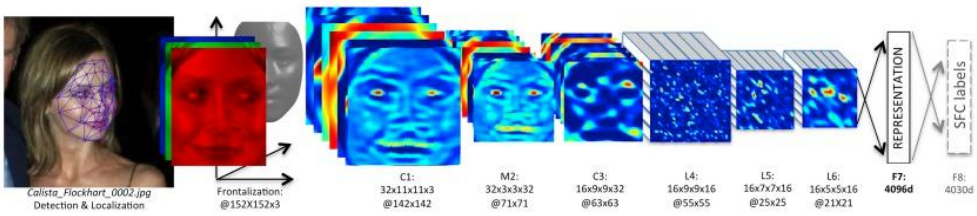

Figure 3.24: Deep Learning Feature Extraction

value through multiple transformation/hidden layers to finally get the desired transformed input, that can be classified easily.

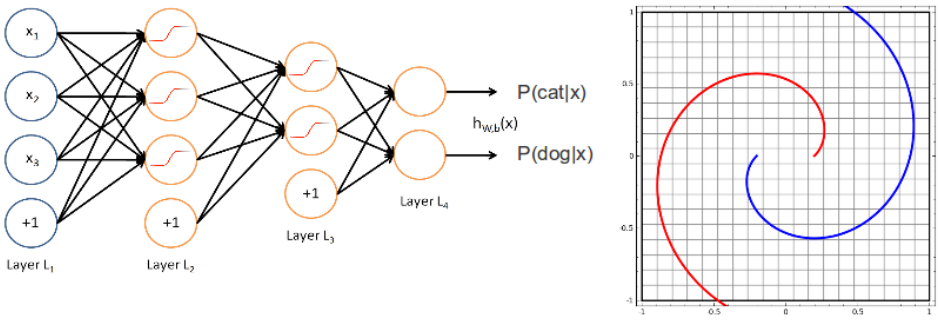

Figure 3.25: Multi-Layer Deep Learning

Deep Learning networks are able to model most of the input function by using as many hidden layers as desired. This is seen to be highly useful and immensely powerful, and given enough hidden perceptron layers, it is possible design neural network-deep learning systems to tackle intense problems.
Deep Learning systems heavily utilize Convolutional Neural Networks for image classification tasks as they tend to work better in the Computer Vision domain. CNNs allow strategic removal of edges and allows to properly utilize the input data.
We merge the multiple layers via pooling to get the desired result:

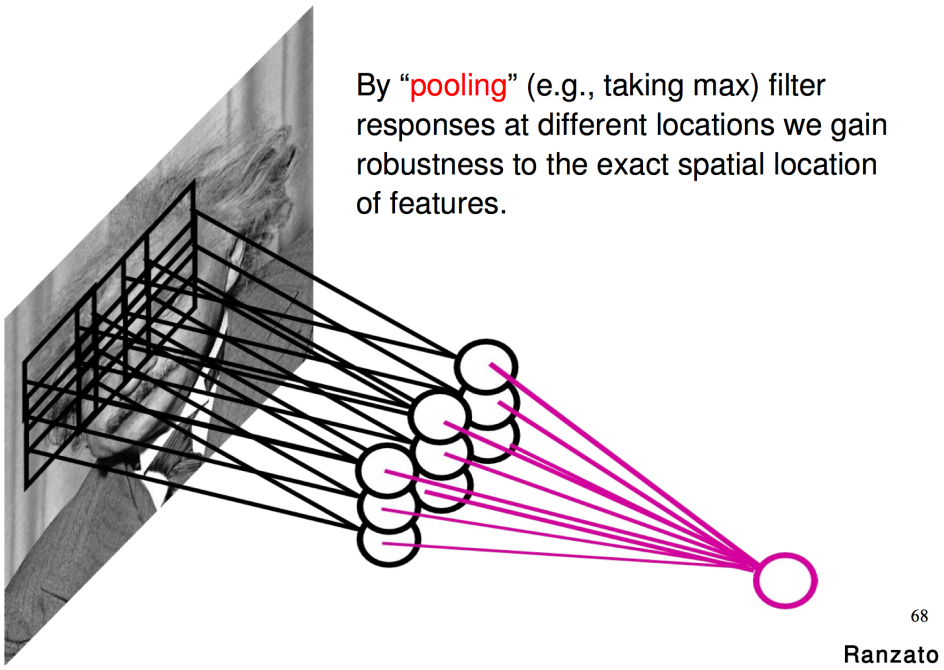

Figure 3.26: Deep Learning Layer Pooling

In the domain of computer vision they can perform the task of object identification, based on different features present in the image being worked upon. We detect multiple objects, and combine them to come up with the resulting object.

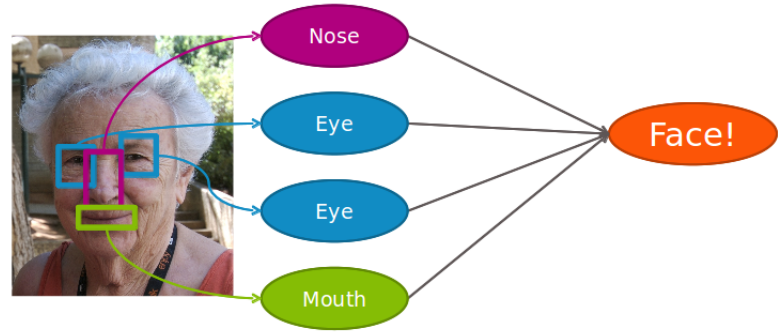

Figure 3.27: Deep Learning Object Identification

Some people who have used deep learning successfully to detect images are:

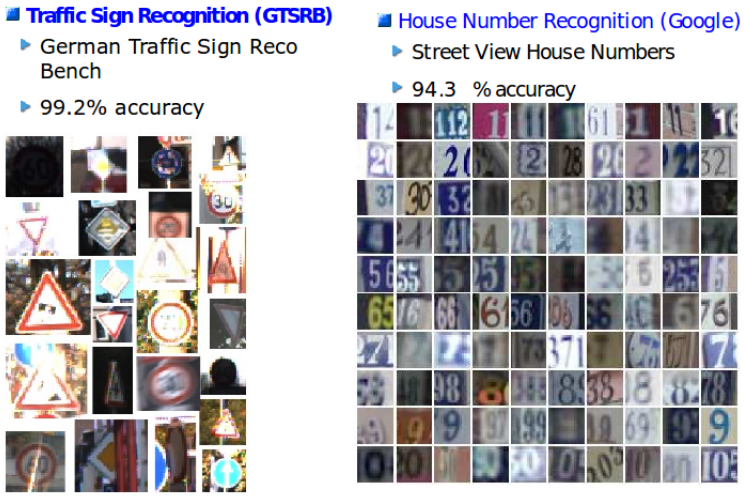

Figure 3.28: Real world cases of Deep Learning

While using traditional Image Processing models the problem that are faced are:

- Raw Pixels perform badly.
- Feature engineering is the key, which is hard.
- Feature engineering with images require domain expertise and knowledge of domain specific area.
- Deep learning is another avenue that can be used to learn the features successfully. Often image processing softwares are underdeveloped.

Deep Learning performed Computer Vision, bought new advantages to the area. The hidden layer of the architecture is responsible for performing most of the work. It helps in pushing all the classes apart so there is a clear discrimination hyperplane or region. Due to this, the hidden layers are the most important part in a deep learning architecture. We can extract the hidden layer and after performing development on that, we can remove the last layer and use the transformed feature in any model like SVM. Some advantages and disadvantages are:

Pros	**Cons**
Enables learning of features rather than hand tuning.	Computationally expensive to learn..
Impressive performance gains on different applications in multiple domains like speech and CV.	Requires hardware like GPU and so on.
-	Requires high amount of data to get good accuracy. Very hard to tune(architecture+parameter).

We use the concept of transfer learning where we take our model trained in an other application on the MSCOCO database with the GoogleNet feature extraction layer to perform machine translation and CV task for successful image caption generation.

3.3.5.5 Top Trending Images

Calculating trending content utilizes the number of up-votes and down-votes or the different ranking scheme. Different sites use different statistical measures and techniques to calculate the ranking of the system. However, generally the approaches taken by different sites are erroneous in calculating the ranking.

Present System

Some approaches taken by different sites are as follows:

WRONG SOLUTION #1:

ScoreRating = (Upvote ratings) - (Downvote ratings)

Reason for error: If we compare two scenarios:

Scenario 1	**Scenario 2**
Ups: 500 Downs: 200 Score: 480 Positive Rating: 71%	Ups: 5000 Downs: 3800 Score: 1200 Positive Rating: 57%

As can be seen from the above example, we can see that this system is highly flawed. Seeing as Scenario 2 should have higher positive rating. Social Websites that use this system, and make a mistake are: Urban Dictionary

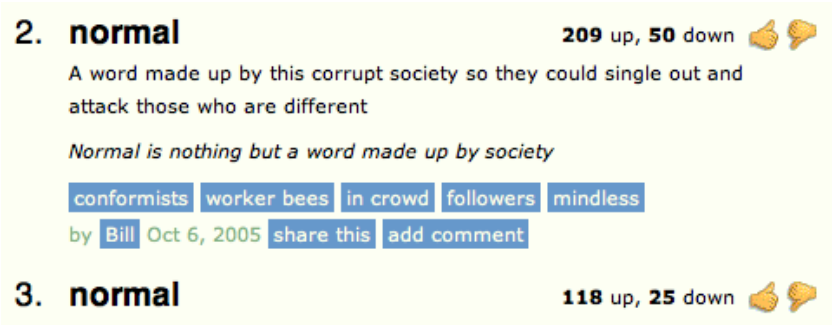

Figure 3.29: Ranking System 1

WRONG SOLUTION #2:
ScoreRating = Average Score Rating = (Upvote ratings) / (Total ratings = (Upvote+Downvote))
Reason for error: The Average score rating works perfectly if we have a lot of data with the appropriate rating attached with it, but this system works incorrectly in different situations. If we compare the following two scenarios:

Scenario 1	**Scenario 2**
Ups = 2	Ups = 100
Downs = 0	Downs = 1
AvgScore = 2/2 * 100 = 100%	AvgScore = 100/101 * 100 = 99%

As can be seen from the above two scenario, the second scenario is rated lower than the first scenario. This is obviously a false method since higher amount of up-votes should be ranked much higher on the rating scale. Social Websites that use this system, and make a mistake are: Amazon.com

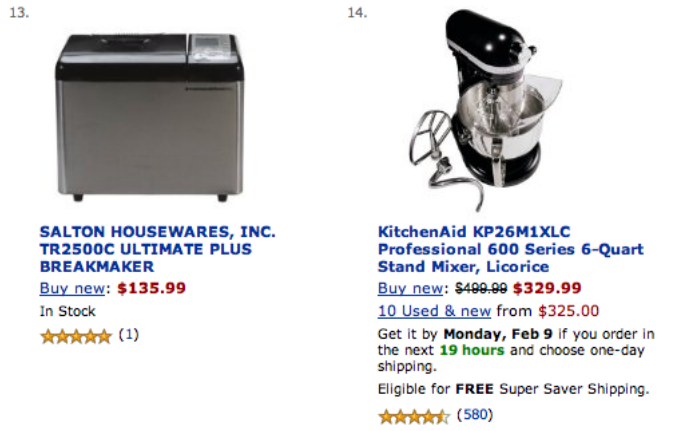

Figure 3.30: Ranking System 2

Our System and correct solution: Score = Lower bound of Wilson score.

In the Wilson Score formula the parameters represent the following variables:

$$\left(\hat{p} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2}\sqrt{[\hat{p}(1-\hat{p}) + z^2_{\alpha/2}/4n]/n}\right)/(1 + z^2_{\alpha/2}/n).$$

Figure 3.31: Wilson Score Rating

- p = NUmber of Upvotes.
- n = Total number of ratings = upvotes + downvotes.
- $z_\alpha/2$ is the $(1 - \alpha/2)$ quantile of the standard normal distribution

What the Wilson Score Means?

In the wilson score interval discussed above what happens is that, If an object(Uploaded Image) has a single upvote and no downvotes it has a 100% approval rating, but since theres not very much data, the system will keep it near the lower end of the rating list. However, if it the image has 10 likes and only 1 dislikes, the rating module tends to have enough significance confidence to place it higher than an image object with Forty likes and Twenty dislikes. And the best part is that if its erroneous (which might happen 15% of the time), it will subsequently get enough data, since the image with low data is at the top of the rating list.

The best part of the significant rating system, is that there is no effect of time on the rating system. The ranking is done solely on the basis of the number of likes and dislikes on an image.

Visualization

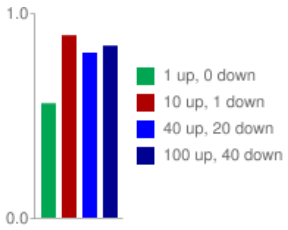

Figure 3.32: Wilson Confidence Sort Visualization

The confidence sort calculated using the Wilson score interval does not care about the quant count of how many total votes an image(object) has obtained, but only about the number of total upvotes it has gathered compared to the

total number of rating votes.
This system is the correct rating system that can and is used to rank various multiple different domains like Spam rating or others.

3.3.5.6 Likes/Dislikes Rating

Users can choose to 'like' or 'dislike' a post. This acts like an upvote/downvote system which allows the system to rank images according preferences of the users. The concept is widely successful in online communities such as reddit with a large user base and little or no moderation. The system relies on its users being the best judge of quality of the content posted. The underlying assumption is that most of the users would be able to downvote the relatively smaller number of obscene or offensive comments. This information is fed into the ranking system for generation of trending image on the website. Additionally, posts that are downvoted by a large number of users may be penalized in terms of their position or they may be completely removed from the site.

3.3.5.7 Comments Board for Images

All uploaded images have a comments board where viewers can express their views, thoughts about the images in the form of text. Since comments are a free text field, there is a chance that certain users may use inappropriate or offensive language. We have an automated stop-word checking that prevents users from using abusive language within the comments section.

Chapter 4

Proposed System Analysis and Design

In the following chapter we discuss the requirements and constraint required to bring the project to life.

4.1 Introduction

Most of the system design has already been explained previously, here we take the time to talk about the surrounding avenues of the project like requirement and the constraints in which we run the whole system. This is an important step in our project as it explains what is essentially required to run the system as well as what are the different things that can be accomplished by using our system.

4.2 Requirement Analysis

4.2.1 Functional Requirement

4.2.1.1 Product Perspective

The Smart Image Sharing system that is to be developed is not a complete information gathering system in itself. The system and its requirements are only pertaining to the functionality needed to implement a system that can successfully upload images, perform profanity check, comment and rate on the image and generate caption and trending image and display the final results, given the image to be uploaded as an information source. Since the system is not a complete information store by itself, the software is being developed as a Flask based system that takes in an image, performs profanity check

and stores it in, allowing commenting and rating features to millions of users worldwide. The system is a one stop location for all avid photographers. The uploaded images are provided to the image processing module. By doing this, the system will provide efficiently tagged and classified images, along with the appropriate captions. The users are able to rate the images, and the system can take in the rating and generate the list of trending images worldwide.

4.2.1.2 Product Features

The following is a table of the requirements that the system SHALL meet. The list of requirements was produced while working on efficient system design by the project members and proper online research, it helps in designing an efficient, portable and scalable application useful for the users in the long run.

Table of Shall Requirements

ID	Shall Requirement
1.	The user SHALL be able to choose which image they want to upload.
2.	The system SHALL be able to check whether the image is clean or not.
3.	The comment feature shall be able to take user comments, check for profanity and post clean comments.
4.	The caption system SHALL be able to read image and perform captioning on it.
5.	The trending system shall be able to calculate the popularity ratings of the images.
6.	The Search query option SHALL be able to take in the search query and provide users with all images that have the content of the search query in it.
7.	The API shall be able to take in a URL, download the image and tell whether the image is clean or profane.

4.2.1.3 User Characteristics

The following table identifies and describes the different users of the Smart Image sharing system software. The information gathered about the different users of the system helped define what the software needs to do. Also, these users are referenced in the requirements and diagrams.

Table of User Characteristics

User	Description
Photographers	The user is any person wishing to share his/her work with the global community. They can use the system to successfully upload their content and allow communication and response on the work.
Companies/Organization	Companies can use our search API to request images pertaining to a particular topic. They can use the classification API to check whether an image is clean or not.
Additional Systems	Our API's can be used by other systems to perform the aforementioned task of classification and also return images based on search.
Administrator	The admin can change the code structure, and also handle the data base. The admin is also responsible for the task of manual moderation of the uploaded images, where the can allow an image in the system if it was a false negative.

4.2.1.4 Assumptions

The following table lists the assumptions made by the requirements that define the Smart Image Sharing system.

Assumption	**Description**
Correct URL passed.	The URL should have the exact parameter passed to open the respective image features.
Image with specific extension.	The uploaded image should only be of a specific JPEG or JPG type, otherwise error message should be shown.
Image is classified correctly.	The image passed should be classified as adult or clean correctly. Hence uploaded image should be clean.
The user enters comments in english.	The comments entered should be in english as our system can only check and test for english language textual content.
The API image URL should point to an image.	Search API should have a URL that points to an image, downlaod it and classify it.
The search query should be captioned.	Some saved image should have the search query present as its description.

4.2.1.5 Dependency Description

- **Inter-Module Dependencies**

Independent Modules

The following modules are independent and do not rely on any other modules to initiate them or to provide data.

- Web-User Interface Module.

Dependent Modules
The following modules are independent and do not rely on any other modules to initiate them or to provide data.

Profanity Check This module requires the image from the web user interface, and comment from the user.

Trending Image This module requires the entered rating; likes and dislikes from the users.

Image Caption This module uses the trained pickle files to caption the images and allow users to search for images.

API Result This module takes in a URL and downloads the image and classifies it as clean or dirty. Also, it can take in the search term and present the users with all images that fit that particular search query.

• Inter-Process Dependencies
As described earlier the two main processes are the Image Classifier and the Image Caption processes. The classifier process depends on the image processing algorithm to tell whether or not an image contents profane comments. Also, the image caption uses deep learning to caption images.

4.2.1.6 Constraints

The following is a table of the design constraints that the system SHALL meet. The list of constraints was produced while working on the initial efficient system design by the project members and proper online research. The Smart Image Sharing project is an additional system designed to implement the required functionality on top of Flasks web framework. It performs captioning and classification of the images uploaded by the users, with certain design constraints on the design of the Image sharing software. The table below lists those constraints.

Table of Constraints

ID	Shall Requirement
1	The user or 3rd-Party systems using our system SHALL be able to get the desired result, without compromising the system.
2	The IMage Captioning SHALL be properly trained so as to provide as specific and accurate a caption answer as possible.
3	The system SHALL only allow the clean images to be uploaded in the system.
4	The system SHALL provide anonymity to the user while uploading their content and posting their comments and ratings.
5	The system SHALL work online as a realtime image uploading system capable of publishing comments and images.
6	The API shall take in correct parameters or else give an error or 404 page.

4.2.1.7 Performance Requirements

The following tables list the performance requirements of the Smart Image Sharing System.

Table of Performance Requirements

Performance Requirement	Description
Efficient Captions	The system should generate appropriate captions for the supplied image.
Correct Image Classification	The system should be able to classify the input image as clean or profane correctly.
Trending Rating	The system shall take the number of likes and dislikes correctly and calculate the ranking of the input images.
Image Comments	The system shall only allow clean comments to be posted, hence allowing a clean platform.

4.2.2 Non-Functional Requirement

4.2.2.1 Product Requirements

Reliability Reliability of the Smart Image Sharing Platform system is directly proportional to the classifier and captioning model and its efficiency, hence most of the time is used to train the Deep Learning model and test the Image Processing system with the correct training set and proper pre-processing and cleaning. After checking the optimization and training the model is able to obtain an efficiency of 70%.

Security The system uses complete anonymization on the users part, and hence uses A small meta-data database for data storing. The image and comment uploading is completely anonymous.

Maintainability It is easy to maintain the system as it is a simple web-based implementation with a Python Flask web framework and a Python application-backend.

Portability The benefit of designing a Web-Application is that we do not have to worry about scalability and the system back-end on the users system. Hence, we can scale the application and not worry how problematic would it be to roll- out product changes. It is very easy to port the web-system to every system and mobile computing devices.

Efficiency The system is efficient to use as it is easy to understand and also, to upload images. It is efficient in the form that it allows only clean content to be uploaded. The image captioning system has to perform well and show the correct tags. API's has to be used correctly with the correct parameters.

Usability The system can be used in the form of the web-user interface or the easy to use APIs that can be used to mine the correct data.

4.2.3 Engineering Standard Requirements

The following table lists the different standards that the Smart Image Sharing Platform project is to be in compliance with.

Table of Standard Requirement

Standard	Description
REST API	Returns a JSON of search results that has to be parsed for the content.
Correct Image extension	Only JPG/JPEG image to be uploaded for proper processing and caption generation.
Economic	The system has to be easy to use for all avid photographers.
Social	We provide a system that can only upload clean images and comments.
Political	Clean comments for all people and to be a politically correct system, without content infringement.
Ethical	The system is clean and focuses on ethicality by providing an only clean system.
Sustainability	We provide easy upgrades to the system and an incremental design for changes.

4.2.4 System Requirements

4.2.4.1 Hardware Specification

- Minimum RAM: 2GB(For Deep Learning)
- Minimum disk space: 5GB
- Processor Intel Core 2 DUO 2.4 GHz minimum.

4.2.4.2 Software Specification

- Linux/Unix Server
- Python 2.7-3.2
- NLTK
- Flask Micro-Web Framework
- Deep Learning Framework - Chainer, Matplotlib, Scipy, Numpy
- Pillow, JSON, Image Processing,
- Web Browser

Chapter 5

Results and Discussion

A self-sustained Image Sharing platform built from the ground up that allows users to interact with their pictures. Providing liking features to promote images and a comment section to allow discussions and review sharing. Apart from this the system should enforce high standard on the content being published and it should be void of any profanity like nudity and sleazy comments and words. The system also has a top trending page that show the top trending ones. Moreover we expect to start a product that users of all ages like to safely and easily promote their contents. The evaluation of our system is based on how well we achieved our functional and nonfunctional requirements. With an incremental model, we hope to achieve all the requirements.

5.1 Functional Requirement Evaluation

1. *The System should be able to upload user images.*
   Result: Our web user-interface should be able to take in the image location from a system and then check if the image is uploaded or not.

2. *The system should be able to generate URL for direct access to uploaded image.*
   Result: Show the uploaded image along with the URL generated.

3. *The system should be able to display thumbnails linked to images.*
   Result: Check thumbnail generation for multiple image uploads.

4. The system should be able to upload/attach comments on the image.
   Result: Multiple comments should be uploaded successfully and loaded to the database. Each image should load the appropriate comments for itself.

5. Uploaded images should be checked for profanity/nudity.
   Result: Only clean images should be uploaded and the profane ones blocked.

6. Comments should be clean.
   Result: Clean comments should be allowed to be loaded in the system and the database.

5.2 Non-Functional Requirement Evaluation

1. *Reliability.*
   Result: The system should be reliable and upload images and comments easily and concurrently.

2. *Maintainability.*
   Result: The system should be easily maintainable and updatable. New features can be added easily in subsequent updates.

3. *Portability.*
   Result: Should be accessible from all devices easily. Minimal dependencies.

4. *Highly Available.*
   Result: The system should give high uptime even after multiple concurrent uploads and API requests.

5.3 Brief

This document comprises of the Software Test Case for the Smart Image Sharing System. It tries to test the boundary values and see the success rate for the system. It helps in evaluating the system and to figure out the limitations and hence the future work that may be carried out in successive revisions of the system.

5.4 Test Cases

TestCase ID	1	**Test Engineer**	Ankit Vadehra
Production Module	Upload Image	**Testing Date**	22-04-2016
Product Version	1	**Testing Cycle**	2
Revision History	0	**Status**	Passing
Purpose	Check if image upload is successfull in the system.		
Assumptions	Correct Image format is provided, which is JPG/JPEG.		
Pre-Conditions	Initiate and start the Flask server. User Image: test.jpg		
Steps To Reproduce	• Initiate Flask server. • Upload the same image again from file explorer.		
Expected Results	Upload Image and display rating and comment section.		
Actual Outcome	Image Uploaded successfully. Displaying rating and comments		
Post Conditions	Show uploaded image, with rating, commenting feature.		

TestCase ID	2	**Test Engineer**	Rohan Kumar
Production Module	Image Classification	**Testing Date**	22-04-2016
Product Version	1	**Testing Cycle**	1
Revision History	0	**Status**	Failing
Purpose	Check if system can check profanity.		
Assumptions	Correct Image format is provided, which is JPG/JPEG.		
Pre-Conditions	Initiate and start the Flask server. User Image: adult.jpg		
Steps To Reproduce	• Initiate Flask server. • Upload the same image again from file explorer.		
Expected Results	Response showing error, due to adult image.		
Actual Outcome	Image Uploaded successfully. Displaying rating and comments.		
Post Conditions	Show response.		

TestCase ID	3	Test Engineer	Ankit Vadehra
Production Module	Image Classification	Testing Date	22-04-2016
Product Version	1	Testing Cycle	2
Revision History	0	Status	Passing
Purpose	Check if system can check profanity.		
Assumptions	Correct Image format is provided, which is JPG/JPEG.		
Pre-Conditions	Initiate and start the Flask server. User Image: adult2.jpg		
Steps To Reproduce	• Initiate Flask server. • Upload the same image again from file explorer.		
Expected Results	Response showing error, due to adult image.		
Actual Outcome	Response showing error, due to adult image.		
Post Conditions	Show response.		

TestCase ID	4	Test Engineer	Rohan Kumar
Production Module	Generate URL for image	Testing Date	22-04-2016
Product Version	1	Testing Cycle	1
Revision History	0	Status	Passing
Purpose	Dedicated URL generated for image.		
Assumptions	Image Upload was successful.		
Pre-Conditions	Image uploaded successfully.		
Steps To Reproduce	• Open Image. • Post comment in english.		
Expected Results	Redirect to Uploaded image dedicated URL.		
Actual Outcome	Redirect to Uploaded image dedicated URL.		
Post Conditions	Show image.		

TestCase ID	5	**Test Engineer**	Ankit Vadehra
Production Module	Comment Profanity	**Testing Date**	22-04-2016
Product Version	1	**Testing Cycle**	1
Revision History	0	**Status**	Passing
Purpose	Check if profane comments are blocked.		
Assumptions	Image was uploaded. English comment.		
Pre-Conditions	Post comment = "This image sucks. F**k O*"		
Steps To Reproduce	• Initiate Flask server. • Post textual comment.		
Expected Results	Response showing error, due to profane comment.		
Actual Outcome	Response showing error, due to profane comment.		
Post Conditions	Show response.		

TestCase ID	6	**Test Engineer**	Rohan Kumar
Production Module	Comment Profanity	**Testing Date**	23-04-2016
Product Version	1	**Testing Cycle**	2
Revision History	0	**Status**	Passing
Purpose	Clean comments are uploaded.		
Assumptions	Image was uploaded. English comment.		
Pre-Conditions	Post comment = "This image is so funny."		
Steps To Reproduce	• Initiate Flask server. • Post textual comment.		
Expected Results	Accept Comment.		
Actual Outcome	Accept Comment.		
Post Conditions	Show the image with posted comment.		

TestCase ID	7	**Test Engineer**	Ankit Vadehra
Production Module	Generate captions.	**Testing Date**	23-04-2016
Product Version	1	**Testing Cycle**	1
Revision History	0	**Status**	Passing
Purpose	Generate Captions.		
Assumptions	Image was uploaded.		
Pre-Conditions	Digital Image: "cats.jpg"		
Steps To Reproduce	• Initiate caption script. • Generate captions.		
Expected Results	Caption containing cats and room.		
Actual Outcome	A cat is sitting on a chair in a room.		
Post Conditions	Show image with captions.		

TestCase ID	8	**Test Engineer**	Rohan Kumar
Production Module	Search API	**Testing Date**	23-04-2016
Product Version	1	**Testing Cycle**	1
Revision History	0	**Status**	Passing
Purpose	Return relevant images.		
Assumptions	Image was uploaded with query.		
Pre-Conditions	Query = "Cat"		
Steps To Reproduce	• Initiate Flask server. • Pass search parameter.		
Expected Results	Images with cats in them.		
Actual Outcome	Images with cats in them.		
Post Conditions	Relevant images.		

TestCase ID	9	**Test Engineer**	Ankit Vadehra
Production Module	Scheduling script	**Testing Date**	23-04-2016
Product Version	1	**Testing Cycle**	1
Revision History	0	**Status**	Passing
Purpose	Schedule CRON script.		
Assumptions	Server is running. Modules installed.		
Pre-Conditions	Everything is designed.		
Steps To Reproduce	• Initiate Flask server. • Schedule caption via CRON.		
Expected Results	Working		
Actual Outcome	Working		
Post Conditions	All systems are working.		

TestCase ID	10	**Test Engineer**	Rohan Kumar
Production Module	Load Stability	**Testing Date**	23-04-2016
Product Version	1	**Testing Cycle**	1
Revision History	0	**Status**	Passing
Purpose	Check multi user load.		
Assumptions	Sites working. Users are posting images,		
Pre-Conditions	Everything is working.		
Steps To Reproduce	• Initiate Flask server. • Schedule caption via CRON.		
Expected Results	Can handle the load.		
Actual Outcome	Handles the load.		
Post Conditions	All systems are working.		

5.5 Software Test Report

Hence we are able to decipher that the software is highly accurate and can provide us with successful results and pass a lot of conditions except certain image classification and caption generation, which is an obvious limitation, that can be perfected with more and more training but never reach 100% accuracy. Linguistic Variations in the posted comments can also cause certain discrepancies. Right now we have support for only English language and profanity check for English language too. High accuracy is possible only when the data-sets correspond. Since there are so many ways of writing the same thing, so many antonyms and synonyms it is not possible to account for all the possible types of inputs. Not mistaking the huge range of short-abbreviations that people use. All this can cause wrong classifications. Image Processing models have a certain lacking when it comes to real-time classification. This is obvious because they are not referencing the object in the image, rather just the pixel content. We can use machine learning models to recognize models in images and then perform classification which would give much higher results.

Chapter 6

Conclusions and Future Work

Thus we are able to design a fully connected image sharing system that can take photos from the users and upload them. The system performs image processing tasks to ensure that only clean images are uploaded to the system. We also provide an interaction system that can post clean comments on the images. Also, the ratings have to be done by the users. The User ratings are taken and a statistical formula called the lower bound of the Wilson score is applied. The top rated images are shown in the trending section.
We also design an API section that can take in image URL and return whether or not the image is clean and also, show the caption generated for the particular image. The second API takes in a search query and only shows the images that are able to fit in the particular query.
The future work for the particular project contains developing other features like image similarity and developing the image processing algorithms further to allow even more sophisticated image classification.

References

[1] Imgur, http://imgur.com/

[2] Pinterest, https://in.pinterest.com/

[3] **Jorge A. Marcial-Basilio**, Detection of Pornographic Digital Images, http://www.naun.org/multimedia/NAUN/computers/20-462.pdf

[4] **O Vinyals**, Show and Tell: A Neural Image Caption Generator, http://arxiv.org/abs/1411.4555

[5] **TJ Torres**, Introduction to Chainer: Neural Network in Python, http://multithreaded.stitchfix.com/blog/2015/12/09/intro-to-chainer/

[6] **T Satoshi**, Image caption by CNN and LSTM, http://t-satoshi.blogspot.in/2015/12/image-caption-generation-by-cnn-and-lstm.html

[7] **Amir Salihefendic**, How Reddit Ranking Algorithm works, https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9#.lim7eor8u

[8] **Randal Munroe**, Reddits new comment sorting system, http://www.redditblog.com/2009/10/reddits-new-comment-sorting-system.html

[9] **Evan Miller**, How not to sort by average rating, http://www.evanmiller.org/how-not-to-sort-by-average-rating.html