International Conference on Smart Data Intelligence
(ICSMDI 2021)

Smart Gallery using Google Vision

Shalva Thakurdesai [a], Shubham Vira [b], Gouri Kanitkar [c], Dr. Jagruti Save [d]

[a,b,c,d]*Fr. Conceicao Rodriguis College of Engineering, Bandra, Mumbai 400 050, India*
shalva.td@gmail.com,shubhamvira@gmail.com, gouri.kanitkar23@gmail.com, jsave@fragnel.edu.in

A B S T R A C T

The proposed project comprises a system which helps intelligent gallery management. For this, a server, mobile interface and web interface has been made. The mobile interface uploads photos to the server and ensures privacy of photos. The server processes the photos and saves the data like size, object analysis and OCR in the database. The web interface let's user login and query the photos for different objects and OCR. The user also has option to sort photos by size or date of upload. If the user wants to clear the gallery, all the images can be deleted or downloaded at once. The user can also search for similar images by giving an image as a query alongside text queries. Technologies used include HTML, CSS, SQL, flutter ensures cross platform compatibility and PHP makes API calls and manages the backend. Google vision ensures reliable image processing.

Keywords : Image processing; Label detection; OCR; Google vision; Image query; Image annotation.

1. Introduction

Considering the current situation of the smartphones and the vast number of use cases they have in our life, memory conservation in smartphones has become an issue. Along with this, social networking and messaging applications like WhatsApp, Snapchat have resulted in collection of more spam photos every day. These spam photos mix in the gallery and become difficult to sort and use up the local storage of the device. To tackle similar issues and to manage the photos in a smartphone better, this project has been proposed. The proposed project will have a mobile application made using flutter framework [1] which will be secured with user credentials such as id and password [2]. On entering proper credentials, the user is given the power to upload photos from the local device that is the mobile phone. These images are processed and stored in the SQL database [3]. The user will be able to go on the website and manage the images stored on the server which is a php based server [4].

1.1. Current problems

In today's day and age, when data is so abundant and almost every app on our mobile is responsible in some way or the other for generating data, ever increasing price of smartphone variants with high internal memory seems inevitable. Most of the people today are on WhatsApp and especially India being the largest markets for WhatsApp a lot of data is shared on WhatsApp every day. Out of all this data being generated much of it is spam data and there needs to be a system to preserve the important data and that too smartly with intuitive user interface.

1.2. Hypothesis

Today's data crisis can be solved by intelligently dealing with the important data and getting rid of the spam data. One way to solve this problem is by uploading the important images on a server which stores the images and sort them by the contents of the images. Images contain a variety of data like landmarks, landscapes, cars planes, trains, plants, motorbikes, people, text, etc. Google Vision and Microsoft Cognitive Services proves to be cost effective and reliable or the purpose [5]. Saving these uploaded images based on the contents and letting the user search for these using the content of the images can prove to be useful.

1.3. Objectives

The objective of the proposed project is to smartly store the important media content that we get on a daily basis from the spam images and be able to access them in a smart and user-friendly manner. The users must also be able to access the images from any device as the storage is web hosted and can manage the image database on the server using a web-based interface. For doing this, following is our objective

- Create a password protected mobile application [2].
- Give the user access to phone gallery [1].

- Let the user upload images to server [4].
- Process the images on the server.
- Let the user view image database on web hosted server.
- Let the user search for images based on contents of the image.
- Let the user delete images on the server.

2. Literature Survey

Computer vision is a newly emerging field along with image processing [3] they have found their use in areas like health, robotics, automation and artificial intelligence. Image analysis involves the study of segmentation, feature extraction, and classification techniques.

Text detection is the method of locating areas in a picture wherever, text is present [16]. Text detection and classification in natural pictures is very important for several computer vision applications like optical character recognition, distinguish between human and machine inputs and spam removal. Text detection is identifying text from natural images. Basic digital image processing techniques are used to detect text from the images.

Content-based image retrieval (CBIR) methods study the different peculiar features of the image and extract features like objects, notations or labels[8]. A machine learning (ML) algorithm is used in order to process the images and extract these features. However while using these heavy machine learning algorithms, importing extensive datasets and need for a powerful CPU are major hurdles. Google Cloud Vision application programming interface (API) overcomes these two shortcomings by doing the machine learning work for the developer. Cloud Vision API is trained by Google using a vast dataset and hours of computation spent into developing a strong Machine Learning model; thus, saving computational time in obtaining image labels. It was tested with the help of a well-known dataset having 4972 figures whether this API can outperform existing ML algorithms in describing annotations. This study found the results that around Google Vision bore 42.4% correctness among 4972 images tested. In each dataset, Cloud Vision API is more effective than the ML algorithms. This paper makes a direct comparison between the CBIR performance of Google Cloud Vision API and some well known ML algorithms. After extensive experimental results, it was concluded that Cloud Vision API is quite competitive compared with other image annotation algorithms. Hence, this API could be extended to test other image datasets and be used as a benchmark method for evaluating ML algorithm performance. Most importantly, the computational effort is reduced because the proposed methods do not require training. This framework could be applied directly to fit their image recognition requirements.

Google Vision API is an application programming interface provided by Google that enables developers to understand the content of an image [5]. Due to its powerful machine learning models and a huge database of images, not only it quickly classifies images into thousands of categories, but also detects individual objects and faces within images. Google Vision features to check their usability in practical applications. like label Detection, Explicit Content Detection, Logo Detection, Landmark Detection, Optical Character Recognition, Face Detection.

3. Proposed methodology

3.1. Traditional ML/DL Algorithms

Traditional Machine Learning algorithms can be used however they come with their own downsides. ML algorithms need a vast dataset and high processing power. Because of this, the accuracy and recall may take a hit. To solve this problem, a reliable API has been used. [6][7]

3.2. Google vision

Google image recognition API uses large datasets and trained machine learning models to classifies the images into different categories like objects, tags, faces, landmark, logo, localized-object and ocr in the images and then prints the results along with the confidence value better than many ML algorithms currently used [8]. Developers can put a cap on the confidence value that suits their needs and can limit the number of categories to process iage with. Developers can use this API in C#, GO, JAVA, NODE.JS, PHP, PYTHON, RUBY [9]. This API takes an image input and gives the result as a "Batch Annotate Images Response" in the form of json.

Advantages:

- Being a google powered system, Google's Vision boasts a vast dataset which facilitates excellent object detection with considerable Accuracy [5] [8].
- The REST API response consists of image labels along with their confidence and the developer has control over the max-limit of labels as well as the threshold on the confidence [4].
- Being a google cloud service the responses are rapid with minimal latency.
- The cost of using google vision is for processing images is affordable for moderate use.
- Google vision lets the developers use vision api free of cost for first 1000 images , thus giving users freedom to test the capabilities before purchasing.

Limitations:

- Google vision has a limited performance when considering noisy images [10].
- Working with fixed dataset can show its effects when dealing with unusual objects which the dataset may lack.
- Google vision displays poor to moderate OCR performance against images which have fancy and designer fonts.

3.3. Algorithm

1. Import autoloader from vendor files, database connection details and Google Vision Client. [9]
2. Initialize the Google Vision Client.
3. Decrypt and store the images and ‘userid’ of the user from the flutter application.
4. Make a temporary image array.
5. Using foreach loop decrypt each image and store it in the uploads folder.
6. Rename the images by attaching respective users' ‘userid’ one by one.
7. Push the image into the temporary image array.
8. Start a foreach loop for temporary image array and extract date and also set the time zone.
9. In the same loop, check if the image is already uploaded by the user.
10. If the image is not uploaded, insert the image into the database.
11. Create a path for every image with name taken from image array and path from uploads and send it to Vision Client. [11]
12. Keep cap of 20 tag results maximum.
13. Extract the ‘photoid’ from photos table from database and save tags for that photo id. [12]
14. Segregate the labels and texts with ‘annotate’ function.
15. Extract score and description from each label.
16. Put a cap on the confidence rating at 80% and concatenate all tags.
17. Check if any text is present in the image.
18. If so, add it as a tag else do not add as a tag
19. Insert the tags into the database
20. And if the image is already existing in the database then send "Image already Exists" to the flutter application.

3.4. Flowchart

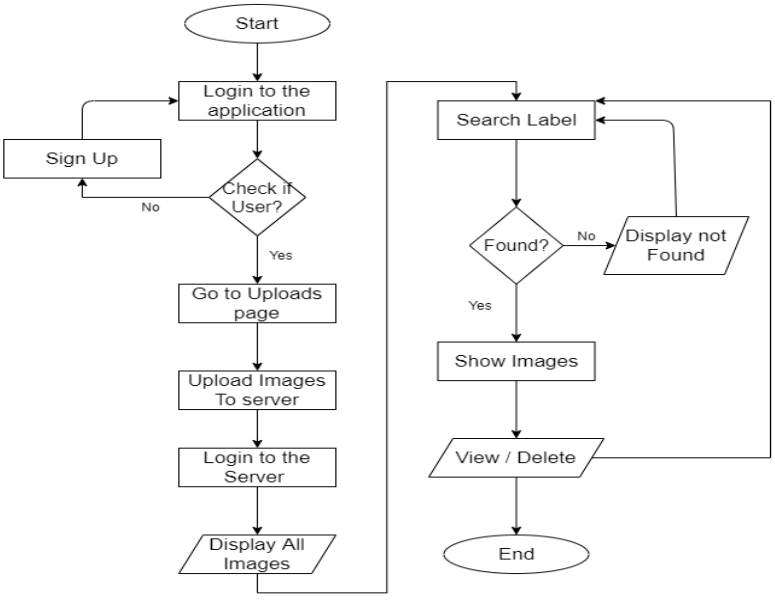

Figure 2.1 Flowchart

4. Database

Pascal’s Dataset:

The Pascal VOC Dataset [13] is a very popular dataset for building and evaluating algorithms for image classification, object detection, and segmentation. These images include 20 classes of images namely aero plane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV, etc. The images in this dataset are real not filtered for “quality” (there’s no CC tag). The images depict complex and unique scenarios, distinct poses, lighting. Even noisy images are included. We have made use of this dataset to test our system like how would our system work in with this set of complex images. Also, to check if the system could withstand some noisy images and analyse the system accordingly.

In the project we have made use of MYSQL database [3] . In the database, three main tables are made namely photo, tags and user.

User Table - The user table stores all the login information and credentials of the user signing up on the application. Every user has an autoincremented id (‘userid’) which is the primary key of the table and proves useful to fetch data from other two tables as and when needed. Login from mobile and website use the email address and the password of the user to verify from the database.

	id	email	password	first_name	last_name
Edit Copy Delete	1	svira@gmail.com	123	shubham	vira
Edit Copy Delete	2	st@gmail.com	123	st	td
Edit Copy Delete	3	shalva.td@gmail.com	123	shalva	td

Figure 3.1 User Table

Photo Table - When the user uploads photos from the mobile application, information regarding the images is stored in this table. This includes name, size and timestamp, photoid and ‘userid’. The “photoid” column autoincrements when any photo is uploaded and is the primary key of the table. Here “‘userid’” acts as the foreign key to link the user and photo table.

	photo_id	user_id	name	size	timestamp
Edit Copy Delete	1	4	4uid2007_006317.jpg	79	Mar,13,2021 04:06:54 PM
Edit Copy Delete	2	4	4uid2007_006254.jpg	131	Mar,13,2021 04:06:57 PM
Edit Copy Delete	3	4	4uid2007_006260.jpg	99	Mar,13,2021 04:06:58 PM

Figure 3.2 Photo Table

Tags Table - The tags of the uploaded photos along with photoid are stored in this table. The tags are generated with the help of Google Vision Client [9, 11] and stored as strings. Here “photoid” acts as the foreign key.

photo_id	tag
1	Bicycle, Wheel, Tire, Bus, Shorts, Train, Bicycles...
2	Train, Vehicle, Plant, Rolling stock, Nature, Trac...
3	Cat, Personal computer, Comfort, Computer, Output ...

Figure 3.3 Tags Table

5. Results

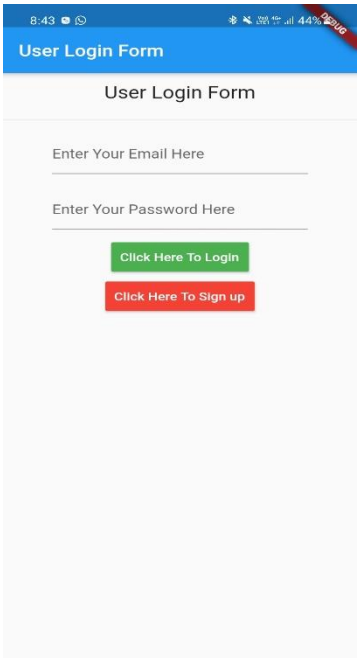

Figure 4.1 Website Login

Mobile Application takes care of login, logout and image uploading. Flutter has been used to create mobile application. User credentials are sent to the server using Http protocol and verification is done on the server side using php. Having credentials ensures privacy of the images and in case two different users upload images having same names, the images wont be interchanged.

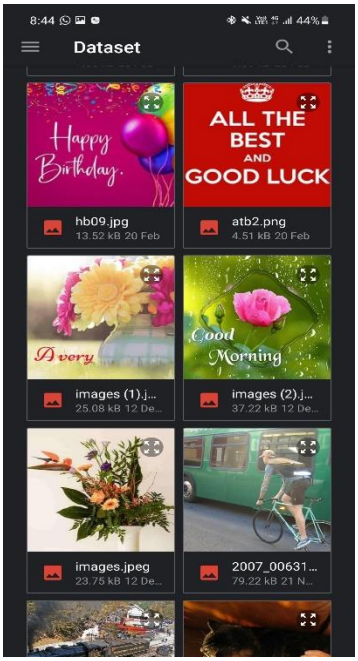

Figure 4.2 Flutter File-picker

Images are picked using file-picker module of flutter. File-picker module gives the user complete access to user's file system. Images are converted to base64encryption format before sending image name and encryption as json to the server. The image formats supported include Jpg, Jpeg, png , img .

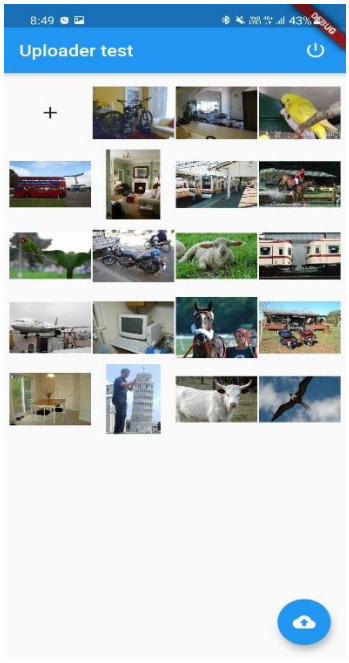

Figure 4.3 Preview of Selected Images

The size of the images and the speed at which they can be uploaded to the server depends on the speed of the server. If the server is fast and powerful enough to decode the encrypted images then the user can upload multiple high definition images taken from modern smartphone cameras. The project

currently uses Hostinger Hosting Services. The size and speed at which the server can decode images also depends on the country of use as the speeds at which the Hosting Server operates also depends on which countries server the data is being sent to.

The Image Processing server Decodes Json & retrieves name, user id and base64encrypted image. It Stores image name in the Database. It sends the image to google vision to process. It decodes response from google vision and save image data like OCR and tags returned back in database to be used for executing user queries.

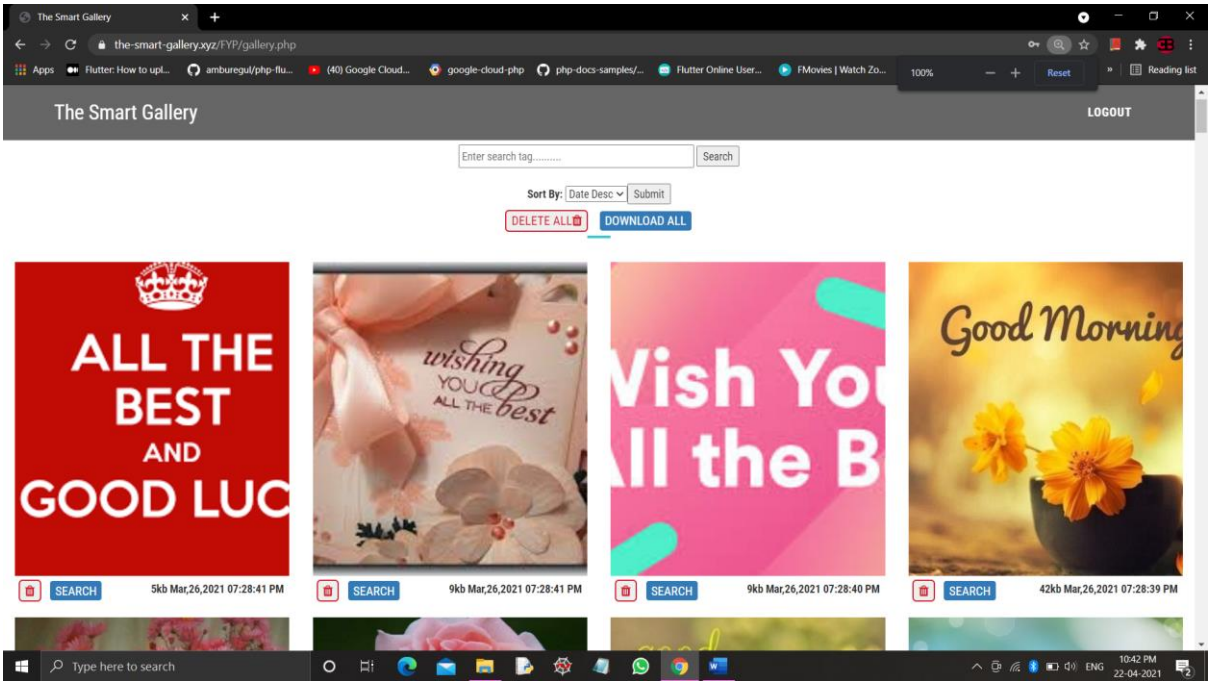

Figure 4.4 Landing page for web-interface

The landing page lets the user see the images uploaded by them from the mobile application. It lets the users perform various operations on these images such as sort, delete, download and two kinds of search namely textual search and image-based search.

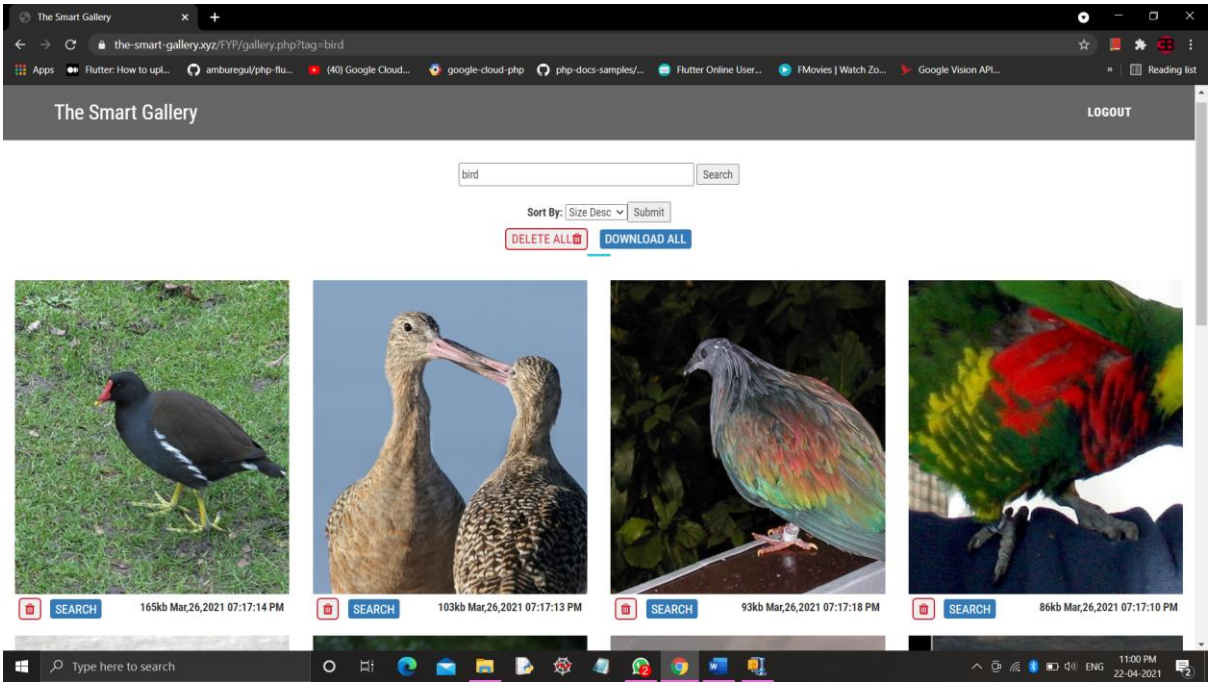

Figure 4.5 Search Result

Using a textual query, the user can search for different tags in images such as objects or textual content. The sorting method selected before also works on the images brought up by the search query. If the user wants to search for multiple tags, they can enter multiple space separated tags in the text search

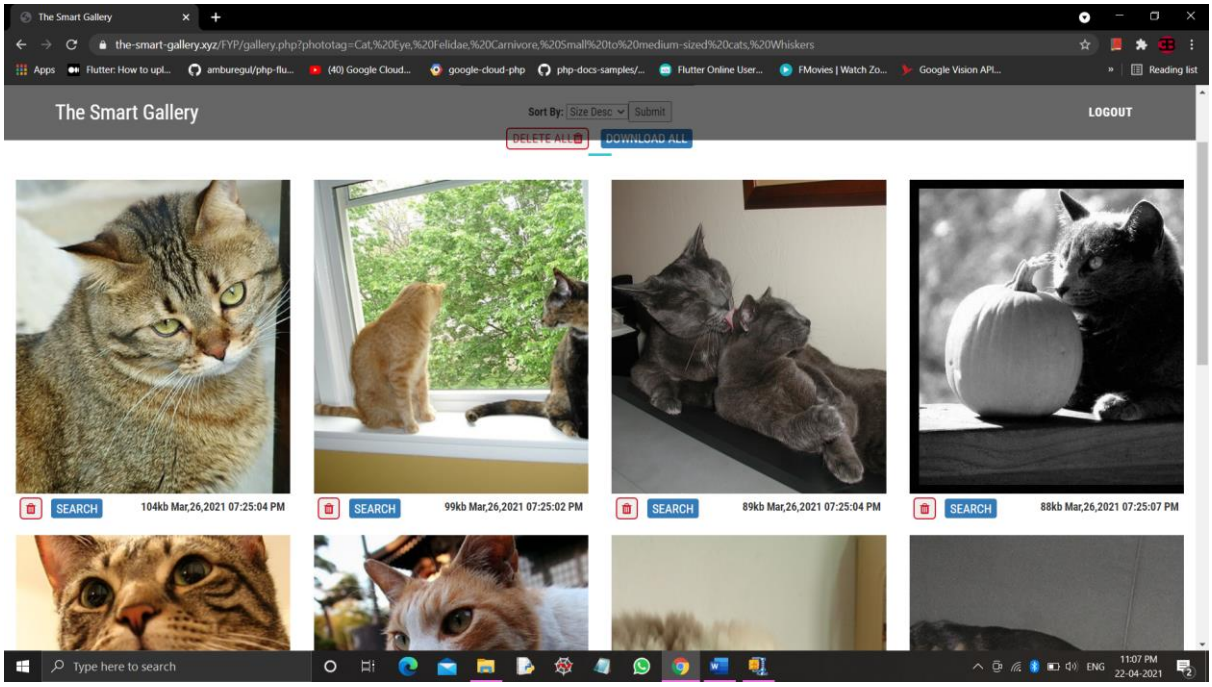

Figure 4.5 Searching using Image Query

Every image has a search button below it. On pressing the button, images similar to the selected images are brought up. This helps user visually group images to work on. After searching for the images, the user can perform operations like “delete all” which will delete all the images brought up by the search query and “download all” which will download all the images in a zip file format.

6. Results discussion

In the proposed project, an analysis has been done by us using 100 different images to check for the OCR and object detection capabilities of the project. For the analysis, different categories such as trains, cars, motorbikes, bicycles. cats, aero planes and images with texts have been taken. Individually for each category, parameters such as recall, precision, accuracy and error rate have been calculated. The average of the values for all these categories have been calculated as the final values.

Table 1 Individual Parameters for dataset

	Accuracy	Precision	Recall	Error
Train	0.99	0.91	1	0.01
Airplane	0.98	1	0.8	0.02
Bird	1	1	1	0
Car	0.8	0.33	1	0.2
Cat	0.98	0.9	0.9	0.02
Motorcycle	0.98	0.87	1	0.01
OCR	0.95	0.73	0.9	0.05

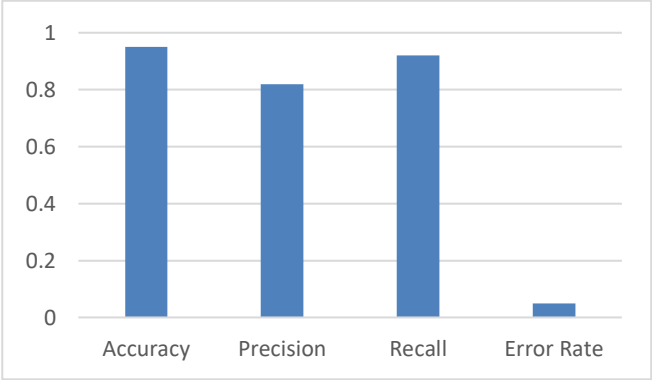

Figure 5.1 Average Parameters for entire dataset

While there is consistency in the recall and accuracy of the searched query, precision takes a hit statistically. The accuracy and recall are much better compared to traditional machine learning algorithms for OCR and object recognition [14] [15]. However, for some images precision may be low, a reason for this is poor of separation between emotions and tags from google vision, for e.g., if searched for happy, it can be considered as an emotion, a tag as well as an OCR on a birthday cake.

7. Conclusion

The original topic revolved around deleting the images from a mobile's internal storage. An attempt was made to do this on a smartphone with the help of a mobile application. However, using plain Android to do this did not seem feasible due to bandwidth issues as well as limitations of computing power and computing time of a mobile phone. To counter this issue, the decision was made to use the mobile application to simply send photos to the server and do the processing on the server side. For this purpose, a flutter application was made and Google Vision API was used as image processing tool. Even though Vision API is a very powerful tool, it has its limitations when it comes to noisy images and multiple font types. The google vison OCR can give faulty results especially when dealing with calligraphic designer fonts. However, having said this it does work almost flawlessly with any other more commonly and formally used font. We have used Pascal's dataset as it has all kinds of images for testing the system thoroughly. Also being an API, the developer has no control over what tags will be returned by Vision. The results can be made more reliable by adding an image pre-processing tool and an algorithm to process the tags and add more commonly used words as results. Even after considering these shortcomings, Vision API still performs better than most of the ML models mainly due to the limitations of their dataset as evident by the accuracy, recall and precision values which are at par with the best.

REFERENCES

[1] https://flutter.dev/docs

[2] https://flutter-examples.com/flutter-online-user-login-using-php-mysql-api/

[3] https://dev.mysql.com/doc/

[4] https://www.myphpnotes.com/post/google-vision-api-with-php

[5] Neves António J. R., Lopes Daniel Pedro Ferreira; "A Practical Study about Google Vision API "in Conference Paper (October 2016)

[6] Karthick.K, Ravindrakumar.K. B, Francis R., Ilankannan S. ; "Steps Involved in Text Recognition and Recent Research" in OCR in International Journal of Recent Technology and Engineering ,Vol-8 (Issue-1), ISSN: 2277-3878, (May 2001)

[7] Chaithanya C.P., Manohar N., Issac, Ajay Bazil; "Automatic Text Detection and Classification in Natural Images" in International Journal of Recent Technology and Engineering, Vol-7 (Issue-5S3), ISSN: 2277-3878, (February 2019)

[8] Chena Shih-Hsin, Chen Yi-Hui; "A Content Based Image Retrieval Method based on the Google Vision." In Conference Paper (February 2017)

[9] https://cloud.google.com/vision/docs/libraries#client-libraries-usage-python

[10] Pathak Akshat, Ruhela Aviral, Saroha Anshul K., Bhardwaj Anant; "Examining Robustness of Google Vision API based on performance of noisy images" in International Journal of Computer Sciences and Engineering Vol.-7, Issue-3 (March 2019)

[11] https://googleapis.github.io/google-cloud-php/#/docs/google-cloud/v0.145.0/vision/visionclient

[12] https://www.youtube.com/watch?v=PqAXE67fwu8&t=217s

[13] https://cs.stanford.edu/~roozbeh/pascal-context/#download

[14] Tang Cong, Feng Yunsong, Yang Xing, Zheng Chao, Zhou Yuanpu; "The Object Detection Based on Deep Learning" in 2017 4th International Conference on Information Science and Control Engineering

[15] Shangbang Long, Xin He, Cong Yao "Scene Text Detection and Recognition: The Deep Learning Era"