

INFORMATION AND CAPTION GENERATOR USING REVERSE IMAGE SEARCH

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in
Information Technology

By

Y VENKATA SAI
(REG.NO 37120085)
SALMAN
(REG NO 37120062)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119.

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Y Venkata Sai (37120085), Salman (37120062) who carried out the project entitled " Information and caption generator using Reverse Image Search " under my supervision from November 2020 to march 2021.

DR. T. SASIKALA

InternalGuide

(Name in Capital letters with signature)

Head of the Department

Submitted for Viva voce Examination held on_____

InternalExaminer

External

DECLARATION

I Y Venkata Sai (Reg. No. 37120085), Salman (Reg No 37120062) here by declare that the Project Report entitled “Information and caption generating using reverse image search” done by me under the guidance of DR.T.SASIKALA, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE :

PLACE :

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing ,Dr.R.Subhashini M.E., Ph.D., Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr.T.Sasikala., for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Information Technology who were helpful in many ways for the completion

ABSTARCT

An In-Depth Look at the Images for Finding Information using Deep learning and Reverse Image Search tells Modern Technology requires a modern solution to find the information from an image. Generating a caption from an image may require computer vision and natural language processing concepts for generating the caption in a natural language like English etc. In this paper, we tried to show how more information can be generated from an image for further analysis of making some predictions. This project is able to detect the person's identity if a person's face is visible in the image as well as all about the image as to where it has taken, resolution, and all other factors. The entire project will describe as much information as possible and use it for further analysis. For example, given a white lion image then the image may tell the output as "The white lion is a rare color mutation of the lion, specifically the Southern African lion." or if an image contains a person as an anonymous then the output will be like "Name, Address and lot more". It can give you complete information about an image, apart from the basic information and the EXIF data, it also shows other useful and in-depth data. There are some reasons that why you need to get information from a photo if a photo you saw that selling a product at a cheap price then we definitely may be interested to buy but wait that may be a scam or what if someone send you the friend request that is the real or fake person that also we have to be aware and what if you are going to meet with a wrong people if he used a fake photo last time with you and you are going to meet with the wrong guy.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGENO
	ABSTRACT	vi
	LIST OF FIGURES	Vii
1.	INTRODUCTION	1
	1.1 REVERSE IMAGE SEARCH	2
	1.2 DEEP LEARNING	2
2.	LITERATURE SURVEY	4
3.	METHODOLOGY ADOPTED	7
	3.1EXISTING SYSTEM	7
	3.2 PROPOSED APPROACH	7
	3.3 ARCHITECTURE DIAGRAM	10
	3.4 PREPROCESSING	10
	3.5 VISUAL SIMILARITY SEARCH	12
	3.6 REVERSE IMAGE SEARCH	14
	3.6.1 TRACK IMAGE USAGE	15
	3.6.2 EXPOSE SOCIAL MEDIA	15
	PROFILES AND POSTS.	
	3.6.3 DISTINGUISH UTTERLY	15
	COMPLETELY DIFFERENT OBJECTS.	
	3.6.4CHECK CREDIBILITY OF CREATIVE	15

	ART WORK.	
	3.6.5 CHECK CREDIBILITY OF CREATIVE	15
	ART WORK.	
	3.6.6 CONFIRMING IMAGES.	15
	3.6.7 CORROBORATORY PEOPLE	15
	VICTIMIZATION CATFISHING	
	TECHNOLOGY.	
	3.6.8 NOTICE APPLICABLE RECIPES	16
	3.6.9 DISCOVER CELEBRITIES NAMES.	16
	3.6.10 NOTICE ASSOCIATE IN NURSING	16
	APPLICABLE APARTMENT	
	3.7 PACKAGES USED	16
	3.7.1 Numpy	16
	3.7.2 Pandas	17
	3.7.3 Os	17
	3.7.4 Flask	17
	3.7.5 Argparse	17
	3.7.6 Json	18
	3.7.7 Requests	19
	3.7.8 PycURL	20
	3.7.9 Markup	21
	3.8.10 Random	21
4.	RESULTS AND DISCUSSION	23
5.	CONCLUSION AND FUTURE WORK	28

5.1 FUTURE WORK	28
REFERENCES	29

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	FLOW CHART	8
3.2	ARCHITECTURE DIAGRAM	10
3.3	PREPROCESSING THE ORIGINAL IMAGE	10
3.4	CHECKING SIMILAR IMAGES	12
4.1	HOME PAGE	23
4.2	LOGIN PAGE	23
4.3	PHOTO UPLOAD PAGE	24
4.4	SAMPLE OUTPUT	24
4.5	TEACHING SAMPLE	25
4.6	COOKING SAMPLE	25
4.7	COMPARISION GRAPH	26

LIST OF TABLES

TABLE NO	FIGURE NAME	PAGE NO
4.1	SEARCH ENGINES OUTPUTS FOR TWITTER	26
4.2	SEARCH ENGINES OUTPUTS FOR LINKED IN	27

CHAPTER 1

INTRODUCTION

Reverse image search is a query technique that formulates a search query based on the sample image provided to it. It works on the principle of content-based image retrieval (CBIR) which allows users to obtain results related to the content of the sample image provided by the user. It also helps to find out other derivative or manipulated versions of the sample image.

Thus, reverse image search helps to find images that most closely resemble the input image. The human mind remembers more of what it sees than it reads. Images help us visualize the information we are looking for. Thus, it is widely crowd-pleasing, and a large amount of population depends on image search for various activities which was concluded by the survey conducted by the authors. It showed that about 80% of people use reverse image search frequently for different purposes. Out of which about 70% of people did not receive expected results.

The reason is that the current system generates its output based on a single keyword search, color distribution or meta tags relating to the image. Such superficial information is not able to provide accurate results because it does not take into consideration the content of the image. This motivated us to improve and optimize the search engine. Visual knowledge fed to the computer through images can be interpreted with the help of descriptions or captions. Captions are the detailed description of images that help to distinctly locate an image from its pool. These descriptions are composed of objects represented in the image and the relationship among those objects. Using captioning, one can find results that are more accurate and apt to the user's search query. Captioning also allows us to filter out any irrelevant and disturbing results that are widely published on the Internet.

Thus, developing the said model will help users to search images that are more in sync with their query and provide results efficiently. Thus, the system presented in this paper will generate captions or descriptions for the images. This will help in optimizing the reverse image search technique by searching images based on the captions rather

than the current criterion for search. This will provide optimized results because the interpretations are universal, and scope for confusion is minimized. This will increase user engagement, popularity and improved search engine optimization of the system.

1.1 Reverse Image Search : Invert Image Search is characterised as a substance based largely image recovery question technique. All in all, Reverse Image Search implies an online crawler innovation that takes input inquiry as a result of the image document and provides a yield familiar with the image, that we've a bent to use inside the knowledge. utterly completely different sites likewise supply the workplace of Reverse Image Search like Reddit.

In the event that you simply have to be compelled to utilize the Reverse Image Search of Google, you want to jaunt the image search page, enter the address of a picture or transfer the image from your laptop computer. Another technique for Reverse Image Search is downloading the augmentation of Chrome and chamber, that empower you to accumulate a listing of things by sound on any online image. This interaction is most acceptable with those photos that unit connected to cyber web substance implies the things that there unit potential outcomes rather than equivocal photos.

1.2 Deep Learning: Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.

In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousand of their neighbours. The question here is how do we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for output value and in between, there may be lots of neurons inter connected in the hidden layer.

Architectures :

- a) **Deep Neural Network** – It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.
- b) **Deep Belief Network(DBN)** – It is a class of Deep Neural Network. It is multi-layer belief networks.
 - a. Learn a layer of features from visible units using Contrastive Divergence algorithm.
 - b. Treat activations of previously trained features as visible units and then learn features of features.
 - c. Finally, the whole DBN is trained when the learning for the final hidden layer is achieved.
- c) **Recurrent:** (perform same task for every element of a sequence) Neural Network – Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.

CHAPTER 2

LITERATURE SURVEY

Elamri and Planque [1] generated image captions using deep learning techniques. This system was developed for visually impaired people, so that they can better understand their surroundings. One of the objectives of the system is to automate caption generation of online images which will make the Web a more inviting place for visually impaired surfers. In this system, image features are extracted using convolutional neural network. Then, the dimensions of this image feature vector are reduced using principal component analysis (PCA). This resulting feature vector is then fed into the recurrent neural network model for sentence generation. The generated description of the image is in valid English. This system was capable to generate precise captions.

Yang et al. [2] presented a novel method for caption generation using pretrained vectors. The new scope covered by them is in terms of diversity of images. It uses a model that is already trained by Oxford to learn input features in addition to the new features learnt from the user-specific dataset, thus making the model more exhaustive with respect to object identification. The concept of “attention mechanism” is used in this model. It presents a distinct approach, wherein attention is concentrated on features using probability distribution of objects based on their brightness and intensity in the overall image. For sentence generation, long short-term memory (LSTM) model is used (an improvement to RNN) which uses memory to remember context of words.

Zhou et al. [3] presented a model which showed better performances than other state-of-the-art models developed until 2016. The authors talk about the use of text conditional-based attention; it aims at training the model for “captioning” rather than mere feature extraction. The scope of this model is expanded to cover detailed and intelligent feature extraction where the models try to find objects related to the object detected previously. This theory covers methods such that the model becomes intelligent enough to recognize relations between objects of an image.

Asakawa et al. [4] trained the model in unsupervised environment by remembering long-term details in memory. In order to incorporate a complete reference, a more robust system is developed using autoencoders. The method of using autoencoders and paragraph vectors was found to have better context in this methodology. This approach has an encoder and a decoder, both work recursively by using words as input and giving sentences. In this case, the intermediate sentences are generally ungrammatical and do not transit smoothly from one to other. In later iterations, it takes text from neighboring sentences rather than only the current sentence. Later, formation of sentences was recommended using a generative model which is based on a regularized version of standard autoencoder. Once the features are translated into probability distributions, the process of extracting words for sentences becomes faster. Autoencoders yielded similar results as the standard recursive models but improved autoencoders can be increasingly used for more narrative descriptions and for generation of longer passages.

Chen et al. [5] replaced the CNN part with three state-of-the-art architectures (VGGNet, AlexNet, GoogleNet). It is found that VGGNet performs best according to the BLEU score. A simplified version of Gated Recurrent Unit (GRU) (a variant of LSTM) is used as a new recurrent network which is implemented using Caffe. Caffe provides a modifiable framework for the state-of-the-art deep learning algorithms. The simplified GRU achieves comparable result when it is compared with the LSTM method. But, it has few parameters which save memory and are faster in training.

Chen et al. [6] replaced the CNN part with three state-of-the-art architectures (VGGNet, AlexNet, GoogleNet). It is found that VGGNet performs best according to the BLEU score. A simplified version of Gated Recurrent Unit (GRU) (a variant of LSTM) is used as a new recurrent network which is implemented using Caffe. Caffe provides a modifiable framework for the state-of-the-art deep learning algorithms. The simplified GRU achieves comparable result when it is compared with the LSTM method. But, it has

few parameters which save memory and are faster in training. In all works stated above, the performance is measured using BLEU metric. It measures the difference and deviation between captions generated by the computer and those given by humans.⁶²⁴ D. Kansara et al.

Literature Related to Works:

The information gained from reading all the existing work is that the captions are generated in two steps: first, feature extraction using object detection followed by sentence generation using language interpretation. Feature extraction: This is the process of extracting features and objects from images uploaded by the user. The main approach followed by all the existing systems is, initially, the image is divided into various parts for identifying components in each section. This makes it easy for the model to focus on each part and retrieve maximum information for each section. This also ensures that any important information is not left out.

Using the similar approach, Karpathy et al. [7] present a very accurate approach by detecting the minute details and even those which can be missed by any interpreter. Zhou et al. [4] used attention mechanism along with object detection through which the model tries to predict the object it has to focus. Khurana and Awasthi [8] used template matching for object detection. In this technique, templates of objects are stored and each input image object is compared with the stored templates to find out a match. Sentence generation:

In this process, captions are generated using language interpretation based on the features extracted in the previous step. In the existing systems, a recursive approach is used where sentences are generated based on Parts of Speech, i.e., using grammatical context and relevance to the previous words of sentences. A recursive method is used where each word of the sentence which is accepted by the model is followed by the next word, which is most suitable in context and grammar. Other approaches using autoencoders and probability distributions for generating sentences are also proposed by Asakawa et al.

CHAPTER 3

METHODOLOGY

3.1 Existing System:

The major drawbacks in the above systems and the approaches are: They fail to generate accurate and contextual captions for moderate-to-high complex images. They sometimes overestimate the relationships between objects by forcefully trying to focus on things which are not present in the image. The models using attention mechanism generate captions which ignore the background details due to reduced intensity of such objects but provides a concise description including all the important objects. Autoencoders are not feasible for generating concise captions; they are more suited for producing large paragraphs rather than captions. In the proposed work, we tried to overcome these issues.

3.2 Proposed System:

After a thorough literature survey, it was concluded that convolutional neural network (CNN) is the most suitable method for the purpose of feature extraction; thus, objects from the image will be extracted using CNN. In order to upgrade the object detection, a pretrained model by Oxford called as “VGGNet” is used. This will lead to improved accuracy as initial weights in the neural networks will not be random, rather will be based on a model which is already trained on many images. The system uses additional datasets on top of it, thus leading to improved accuracy. In order to generate the sentences, long short-term memory (LSTM) is used. LSTM is a form of recurrent neural network that uses an internal memory to remember its inputs.

Thus, it is appropriate for problems involving sequential data (same as in our problem). The internal working of LSTM uses three gates, the input, output and forget gate. The forget gate determines which information is important and which is not. Based upon the decision taken by the forget gate, the algorithm deletes or remembers certain information. The importance of information is assigned by the weights calculated during the training process. Thus, after receiving the input, the image is passed through the system in order to receive the most appropriate caption. Figure 1 shows the flow of information while generating the caption of an uploaded image. The overall caption generation process is divided into two

phases, namely the training phase and the execution phase. In the training phase, the required data is acquired from various sources, and then, it is gathered at one place which is later used for the training purpose.

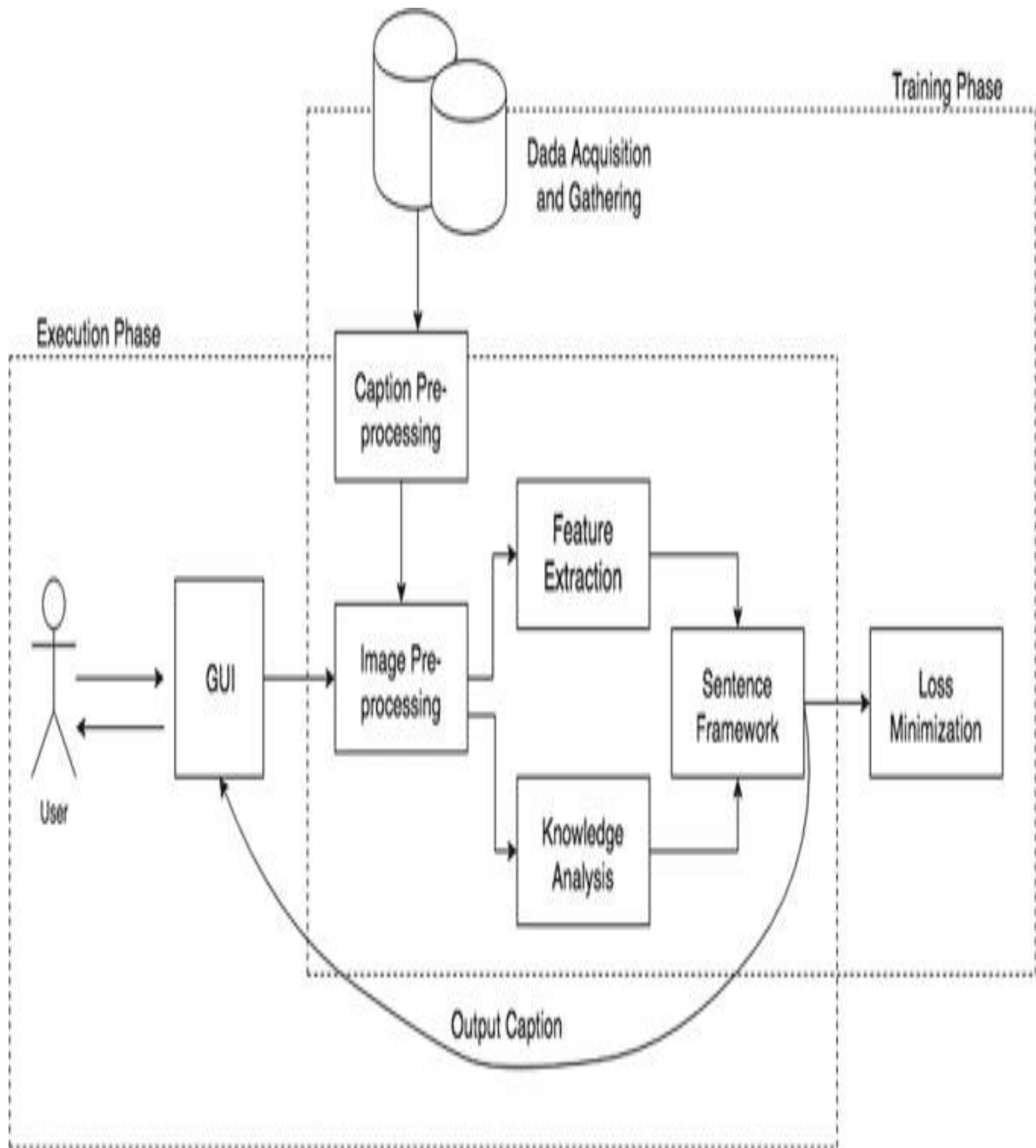


Fig 3.1 Flow Chart

It is that a pre-trained CNN model like Resnet50 can extract 2048 features from any

image (without the fully connected layers). The images in your database can be indexed based on those features (an array/vector of floating points) and then you could find the euclidean distance of the features of the query image vs all other images in your database.

When you sort based on this distance, you should get similar images with the smallest distance. There's one catch with euclidean distance based ranking of images - all of the 2048 features are given equal priority. Lets assume for simplicity sake that the "blue sky" is feature number 2000 in the vector and the "brown building" is feature number 1000. Its actually more complicated than this and almost impossible to identify which features stand for what in an image. However, we do know that they do stand for "something". With Euclidean Distance matching, the blue sky will be given equal importance with the brown building. As humans, we know by context that the building in the foreground is more important than the blue sky in the back ground.

It turns out that context awareness can also be incorporated in the image search by training a neural network to "look at" two vectors of two images and spitting out a score for similarity. This could be a supervised learning task, with features of similar images as inputs, and a synthetic ground truth as output.

I agree with Zhou et al assumption that that "Google uses some combination" of techniques and "ranks the results using a proprietary algorithm".

But what I guess the most important algorithm they use for ranking their results is based on Image compression technique using Auto-Encoders.

Let me first show you how powerful Auto-Encoders are at image compression and reconstruction. Below is a gray-scale image of a dog which is labelled as 'Original' of dimensions 32*32 pixels (total of 1024 pixels). The structure of Auto-Encoder used for compression here is [1024-256-64-8-64-256-1024], which means that a 1024 Dimension image is being compressed to just 8 Dimensions. And in the second part of the image it can be seen how well the 'Auto-Encoder' is able to reconstruct the 'Original' image. Just beside that, we can see how badly PCA algorithm reconstructs the same image after compressing the original image to 8 Dimensions (i.e. taking first 8 principal components only).

3.3 Architecture Diagram:

This diagram will explain step by step process of our project.

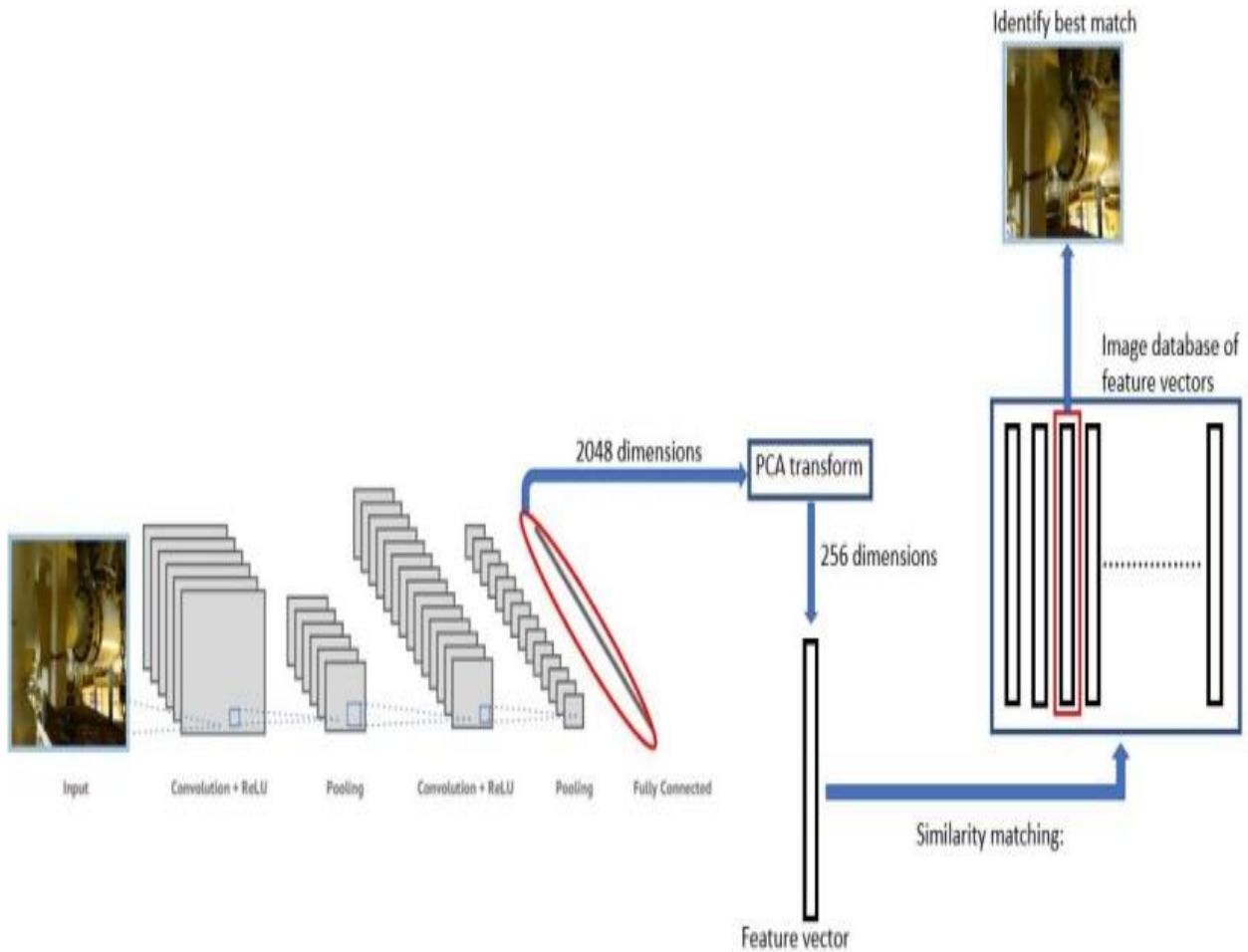
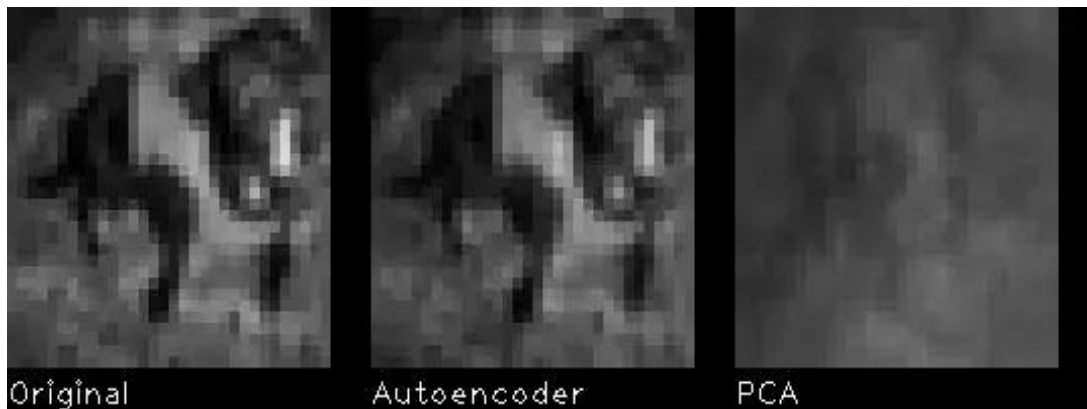


Fig 3.2 Architecture Diagram

3.4 Pre-processing:

PS: Auto-Encoder compression and PCA are NOT loss-less compression techniques.



3.3 Preprocessing the original image:

So for every image of this dimension, it can be mapped to a point in 8-D space by compressing using Auto-Encoders. And if some images are visually similar, then plotting the points of the compressed images in 8-D space shows that they all lie very close to each other (using Euclidean for distance calculations). Or in other words, we can say that, all the points that lie close to each other in the compressed 8-D space are visually similar.

An image is worth a thousand features. In this section, we play with and understand the concepts of feature extraction, primarily with the Caltech 101 dataset (131 MB, approximately 9,000 images), and then eventually with Caltech 256 (1.2 GB, approximately 30,000 images). Caltech 101, as the name suggests, consists of roughly 9,000 images in 101 categories, with about 40 to 800 images per category. It's important to note that there is a 102nd category called "BACKGROUND_Google" consisting of random images not contained in the first 101 categories, which needs to be deleted before we begin experimenting.

Now, import all of the necessary modules: Load the ResNet-50 model without the top classification layers, so we get only the bottleneck features. Then define a function that takes an image path, loads the image, resizes it to proper dimensions supported by ResNet-50, extracts the features, and then normalizes them.

The ResNet-50 model generated 2,048 features from the provided image. Each feature is a floating-point value between 0 and 1.

If your model is trained or fine tuned on a dataset that is not similar to ImageNet, redefine the "preprocess_input(img)" step accordingly. The mean values used in the function are particular to the ImageNet dataset. Each model in Keras has its own preprocessing function so make sure you are using the right one.

Now it's time to extract features for the entire dataset. First, we get all the filenames with this handy function, which recursively looks for all the image files (defined by their extensions) under a directory. Then, we provide the path to our dataset and call the function: We now define a variable that will store all of the features, go through all filenames in the dataset, extract their features, and append them to the previously defined variable.

On a CPU, this should take under an hour. On a GPU, only a few minutes.

To get a better sense of time, use the super handy tool `tqdm`, which shows a progress meter along with the speed per iteration as well as the time that has passed and expected finishing time. In Python, wrap an iterable with `tqdm`; for example, `tqdm(range(10))`. Its Jupyter Notebook variant is `tqdm_notebook`.

3.5 Visual similarity search:

Below is the set of images for similar images generated for query image and this is how to reverse image search generates the similar images and that we can use to obtain the captions.



Fig 3.4 Processing related colour images.

The given Features are stored for indexing and matching the Google mostly use this in combination of keywords to refine searches on most large collections.

And here are several approaches to this problem [8]. One approach is to compare patches of areas between two images. Although this can help find exact or near-exact images (that might have been cropped), even a slight rotation would result in dissimilarity. By storing the hashes of the patches, duplicates of an image can be found. One use case for this approach would be the identification of plagiarism in photographs.

Another naive approach is to calculate the histogram of RGB values and compare their similarities. This might help find near-similar images captured in the same environment without much change in the contents. For example, this technique is used in image deduplication software aimed at finding bursts of photographs on your hard disk, so you can select the best one and delete the rest. Of course, there is an increasing possibility of false positives as your dataset grows. Another downside to this approach is

that small changes to the color, hue, or white balance would make recognition more difficult.

A more robust traditional computer vision-based approach is to find visual features near edges using algorithms like Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB) and then compare the number of similar features that are common between the two photos. This helps you go from a generic image-level understanding to a relatively robust object-level understanding. Although this is great for images with rigid objects that have less variation like the printed sides of a box of cereal, which almost always look the same, it's less helpful for comparing deformable objects like humans and animals, which can exhibit different poses. As an example, you can see the features being displayed on the camera-based product search experience on the Amazon app. The app displays these features in the form of blue dots. When it sees a sufficient number of features, it sends them to the Amazon servers to retrieve product information.

Because our aim is to search among millions of images, what we ideally need is a way to summarize the information contained in the millions of pixels in an image into a smaller representation (of say a few thousand dimensions), and have this summarized representation be close together for similar objects and further away for dissimilar items.

Luckily, deep neural networks come to the rescue. As we saw, the CNNs take an image input and convert it into feature vectors of a thousand dimensions, which then act as input to a classifier that outputs the top identities to which the image might belong (say dog or cat). The *feature vectors* (also called *embeddings* or *bottleneck features*) are essentially a collection of a few thousand floating-point values. Going through the convolution and pooling layers in a CNN is basically an act of reduction, to filter the information contained in the image to its most important and salient constituents, which in turn form the bottleneck features. Training the CNN molds these values in such a way that items belonging to the same class have small Euclidean distance between them (or simply the square root of the sum of squares of the difference between corresponding values) and items from different classes are separated by larger distances. This is an important property that helps solve so many problems where a classifier can't be used,

especially in unsupervised problems because of a lack of adequate labeled data.

3.6 Reverse Image Search:

Google is a very hardcore engineering-oriented company. The Google run map reduces all day long. Images get mapped to features. Features get mapped to new features. Feature selection chooses subsets. Then they get hashed. Then they get quantized. Then some more learning etc.

One thing [7] about google we can say this is single most amazing and single most upsetting thing about vision research at Google. The whole process works because the engineers are top-notch, really care about writing good code, and they aren't stressed out (so they can write high-quality code quite often).

Invert Image Search is characterised as a substance based largely image recovery question technique. All in all, Reverse Image Search implies an online crawler innovation that takes input inquiry as a result of the image document and provides a yield familiar with the image, that we've a bent to use inside the knowledge. utterly completely different sites likewise supply the workplace of Reverse Image Search like Reddit.

In the event that you simply have to be compelled to utilize the Reverse Image Search of Google, you want to jaunt the image search page, enter the address of a picture or transfer the image from your laptop computer. Another technique for Reverse Image Search is downloading the augmentation of Chrome and chamber, that empower you to accumulate a listing of things by sound on any online image. This interaction is most acceptable with those photos that unit connected to cyber web substance implies the things that there unit potential outcomes rather than equivocal photos.

Advantages

3.6.1. Track Image Usage: With the assistance of Reverse Image Search, you may likely follow the photographs on cyber web, and it's the foremost ideal approach to search out the sites where the photographs unit online distributed. inside the web search tool, image making prepared contains some way that options separating valuable information by making Associate in Nursing interpretation of the image take a glance at into a complicated inquiry as against a watchword and activity specific procedure there on. The picture following a Reverse Image Search isn't restricted to net primarily based business

item footage, TV photos, and photos but to boot contain figure focuses painstakingly encoded inside the image.

3.6.2. Expose Social Media Profiles and Posts: Social media is acclaimed for making being reports and associating folks round the globe. For this purpose, once you utilize Twitter or Facebook, you may see utterly different strange phony profiles. With the assistance of Reverse Image Search, you may distinguish that one might be a phony profile which one is real.

3.6.3. Distinguish utterly completely different Objects: Science understudies utilize Reverse Image Search to acknowledge plants, creatures, and science tasks to differentiate their species, names, and completely different connected information on the net. as Associate in Nursing example: we'll snap a photograph of a canine's tree and after transfer the image of a canine or tree on the Reverse Image Search to urge the species and name of the canine or a tree.

3.6.4. Investigate unlabelled Products: With the assistance of Reverse Image Search, we'll likewise investigate the unlabelled item. for example, we'll discover the numerous information of any item whereas not legitimate names.

3.6.5. Check credibility of creative Art Work: If you are a craftsman and you want to seem at that your work isn't utilizing any web site, diary, or webbased media whereas not your authorization, at that point with the assistance of Reverse Image Search, we'll continue the realism of the work and chop-chop discover where your work seems on cyber web.

3.6.6. Confirming Images: Digital image verification through Reverse Image Search apparatuses is a vital profit in inconsistency to the assembly of a cheap image that shows counterfeit occasions. With the assistance of Reverse Image Search, we'll validate any image on the net. As we've a bent to absolutely acknowledge, their unit several phony photos that unit day by day worshipped and shared on Facebook and Instagram. Something dismal is that the image isn't a touch of the being stories and also the chances are that the image was initially distributed once the creator used it. Utilizing Reverse Image Search, it turns into the best approach to ascertain it.

3.6.7.corroboratory people victimization Catfishing Technology: With the assistance of the Reverse Image Search, we will doubtless discover the phony photos and so the phony those that use photos of another person on cyber web. On the off probability that you {simply|that you just} simply utilize long vary informal communication stages to perform Reverse Image Search on profile photos than apparently, a massive name soul is companions on Facebook World Health Organization could also be a human making a trial to kindle folks. variety of the time you also notice that these phony folks took their profile image from another shopper of Facebook or LinkedIn. an identical image is likewise accustomed send some tricks sends. At times, to catfish online new chomps, con artists oftentimes utilize the photographs of creators, entertainers, and entertainers. By utilizing Reverse Image Search, you may keep from programmers.

3.6.8. notice applicable Recipes: The name of the formula are blazing clearly with an image. With the assistance of Reverse Image Search, we'll discover the formula substance and picture on cyber web. inside the gift circumstance, you simply have to be compelled to transfer the image inside the Reverse Image Search apparatus; at that point, you will get the entire formula on cyber web.

3.6.9. Discover Celebrities' Names: By utilizing Reverse Image Search, we'll likewise discover the names of the entertainers and entertainers with the assistance of the image got from the TV comes, Magazines or Films. circle Image Search is Associate in Nursing Associate in Nursing instrument that will assist you with discovering the previous neighborhood, name, course of events history, or the birthday of any megastar that we would like.

3.6.10. notice Associate in Nursing applicable Apartment: If you want to search out the house address, you will want to lease, and if your assets skilled is declining to reveal the planet of the house. you may utilize the Reverse Image Search instrument to follow the house's cause and area.

3.7 PACKAGES USED:

Here we have explained the packages used in the project.

3.7.1. Numpy: NumPy can also be used as an efficient multi-dimensional container of

generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

It provides:

- a powerful N-dimensional array object
 - sophisticated (broadcasting) functions
 - tools for integrating C/C++ and Fortran code
 - useful linear algebra, Fourier transform, and random number capabilities
- and much more.

3.7.2. Pandas:

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis/manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

- a) Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet.
- b) Ordered and unordered (not necessarily fixed-frequency) time series data.
- c) Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels.
- d) Any other form of observational / statistical data sets. The data need not be labeled at all to be placed into a pandas data structure.
- e) The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users, DataFrame provides everything that R's data.frame provides and much more. pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

3.7.3. OS: It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

You first need to import the os module to interact with the underlying operating system. So, import it using the import os statement before using its functions.

3.7.4. Flask: Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

3.7.5.Argparse: The argparse module makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and argparse will figure out how to parse those out of sys.argv. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

ArgumentParser objects

```
class argparse.ArgumentParser(prog=None, usage=None, description=None, epilog=None, parents=[], formatter_class=argparse.HelpFormatter, prefix_chars='-', fromfile_prefix_chars=None, argument_default=None, conflict_handler='error', add_help=True, allow_abbrev=True, exit_on_error=True)
```

Create a new ArgumentParser object. All parameters should be passed as keyword arguments. Each parameter has its own more detailed description below, but in short they are:

- prog - The name of the program (default: sys.argv[0])
- usage - The string describing the program usage (default: generated from arguments added to parser)

- `description` - Text to display before the argument help (default: none)
- `epilog` - Text to display after the argument help (default: none)
- `parents` - A list of `ArgumentParser` objects whose arguments should also be included
- `formatter_class` - A class for customizing the help output
- `prefix_chars` - The set of characters that prefix optional arguments (default: '-')
- `fromfile_prefix_chars` - The set of characters that prefix files from which additional arguments should be read (default: None)
- `argument_default` - The global default value for arguments (default: None)
- `conflict_handler` - The strategy for resolving conflicting optionals (usually unnecessary)
- `add_help` - Add a `-h/--help` option to the parser (default: True).
- `allow_abbrev` - Allows long options to be abbreviated if the abbreviation is unambiguous. (default: True).
- `exit_on_error` - Determines whether or not `ArgumentParser` exits with error info when an error occurs. (default: True).

3.7.6. *Json*: Since its inception, JSON has quickly become the de facto standard for information exchange. Chances are you're here because you need to transport some data from here to there. Perhaps you're gathering information through an API or storing your data in a document database. One way or another, you're up to your neck in JSON, and you've got to Python your way out.

Luckily, this is a pretty common task, and—as with most common tasks—Python makes it almost disgustingly easy. Have no fear, fellow Pythoneers and Pythonistas. This one's gonna be a breeze!.

3.7.7. *Requests*: `Requests` allows you to send HTTP/1.1 requests extremely easily. There's no need to manually add query strings to your URLs, or to form-encode your PUT & POST data — but nowadays, just use the `json` method!.

Requests is one of the most downloaded Python package today, pulling in around 14M downloads / week— according to GitHub, Requests is currently depended upon by 500,000+ repositories. You may certainly put your trust in this code.

Supported Features & Best-PracticesRequests is ready for the demands of building robust and reliable HTTP-speaking applications, for the needs of today.

- Keep-Alive & Connection Pooling
- International Domains and URLs
- Sessions with Cookie Persistence
- Browser-style TLS/SSL Verification
- Basic & Digest Authentication
- Familiar `dict`-like Cookies
- Automatic Content Decompression and Decoding
- Multi-part File Uploads
- SOCKS Proxy Support
- Connection Timeouts
- Streaming Downloads
- Automatic honoring of `.netrc`
- Chunked HTTP Requests

3.7.8. PycURL: is a Python interface to libcurl, the multiprotocol file transfer library.

Similarly to the urllib Python module, PycURL can be used to fetch objects identified by a URL from a Python program. Beyond simple fetches however PycURL exposes most of the functionality of libcurl, including:

- Speed - libcurl is very fast and PycURL, being a thin wrapper above libcurl, is very fast as well. PycURL was benchmarked to be several times faster than requests.
- Features including multiple protocol support, SSL, authentication and proxy options. PycURL supports most of libcurl's callbacks.
- Multi and share interfaces.

- Sockets used for network operations, permitting integration of PycURL into the application's I/O loop (e.g., using Tornado).

3.7.9. Markup: MarkupPy is an intuitive, light weight, easy-to-use, customizable and pythonic HTML/XML generator. It attempts to help developers output valid HTML 4.01 as defined by the W3C specification. The set of valid and invalid elements is not hard-wired and is easily customizable. There are no third-party dependencies only the standard Python library is needed in order to use MarkupPy.

- MarkupPy works with both python 2 and 3.
- The current (2017) version is 1.12.
- The code is in the public domain, you can do absolutely whatever you want with it.
- MarkupPy is a very simple and light weight tool for quickly and painlessly putting together HTML/XML documents practically without any learning curve. If you need anything in a production environment choose something else, e.g. ElementTree.

3.7.10. Random: This module implements pseudo-random number generators for various distributions. For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement. On the real line, there are functions to compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. For generating distributions of angles, the von Mises distribution is available.

Almost all module functions depend on the basic function `random()`, which generates a random float uniformly in the semi-open range `[0.0, 1.0)`. Python uses the Mersenne Twister as the core generator. It produces 53-bit precision floats and has a period of $2^{19937}-1$. The underlying implementation in C is both fast and threadsafe. The Mersenne Twister is one of the most extensively tested random number generators in existence. However, being completely deterministic, it is not suitable for all purposes, and is completely unsuitable for cryptographic purposes.

The functions supplied by this module are actually bound methods of a hidden

instance of the `random.Random` class. You can instantiate your own instances of `Random` to get generators that don't share state. Class `Random` can also be subclassed if you want to use a different basic generator of your own devising: in that case, override the `random()`, `seed()`, `getstate()`, and `setstate()` methods. Optionally, a new generator can supply a `getrandbits()` method — this allows `randrange()` to produce selections over an arbitrarily large range. The `random` module also provides the `SystemRandom` class which uses the system function `os.urandom()` to generate random numbers from sources provided by the operating system.

CHAPTER 4

RESULTS AND DESCUSSION

This chapter describes about the results and I have explained how the results have came.



Fig 4.1 Home page

In this page we will give introduction about our project and at top right menu will be there . In that you can click your option and it will proceed you to next page.

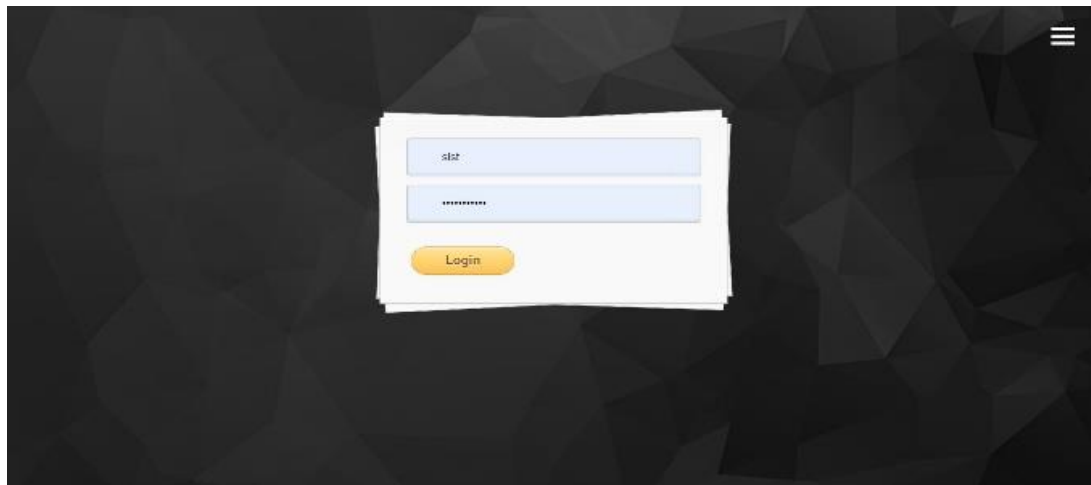


Fig 4.2 Login page

you need to login credintials and login to upload pictures.



Fig 4.3 Photo Upload page

There will be photo upload section you can upload photo and it will generate related captions for particular image. And source links of an image.

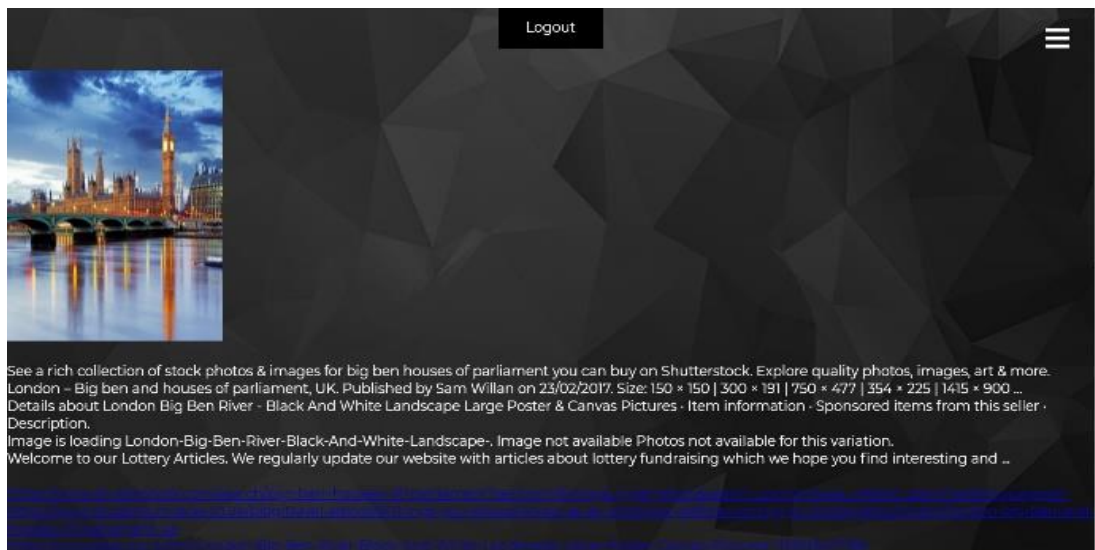


Fig 4.4 Sample output

I have uploaded London Big ben picture I will generate the caption as A rich collection of photos and images for big ben houses of parliament and source links of an image.

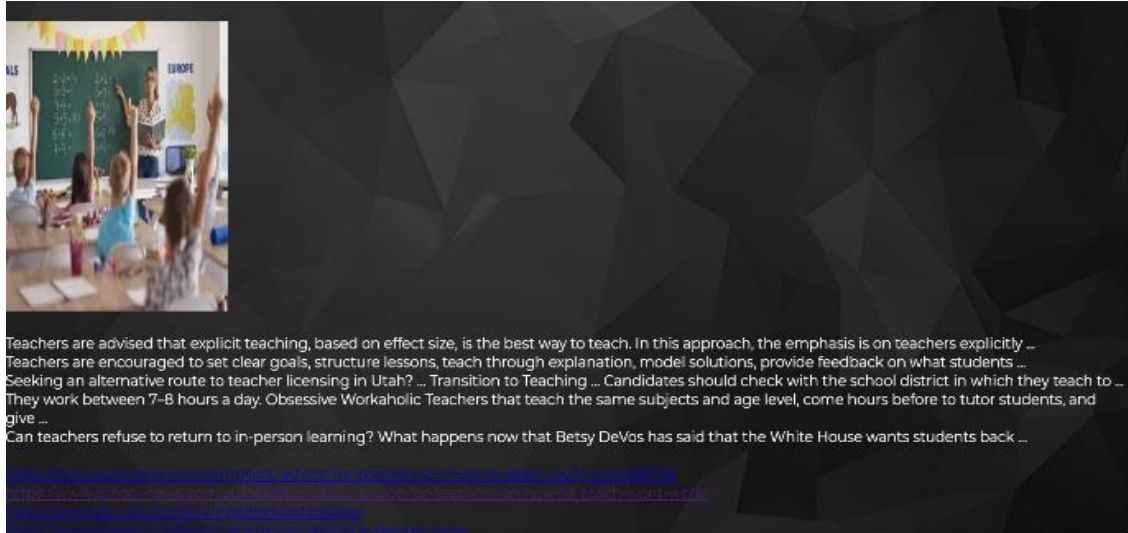


Fig 4.5 Teaching sample

Here I have uploaded the sample teaching picture it will generate caption like teachers are advised that explicit teaching is the best way to teach.

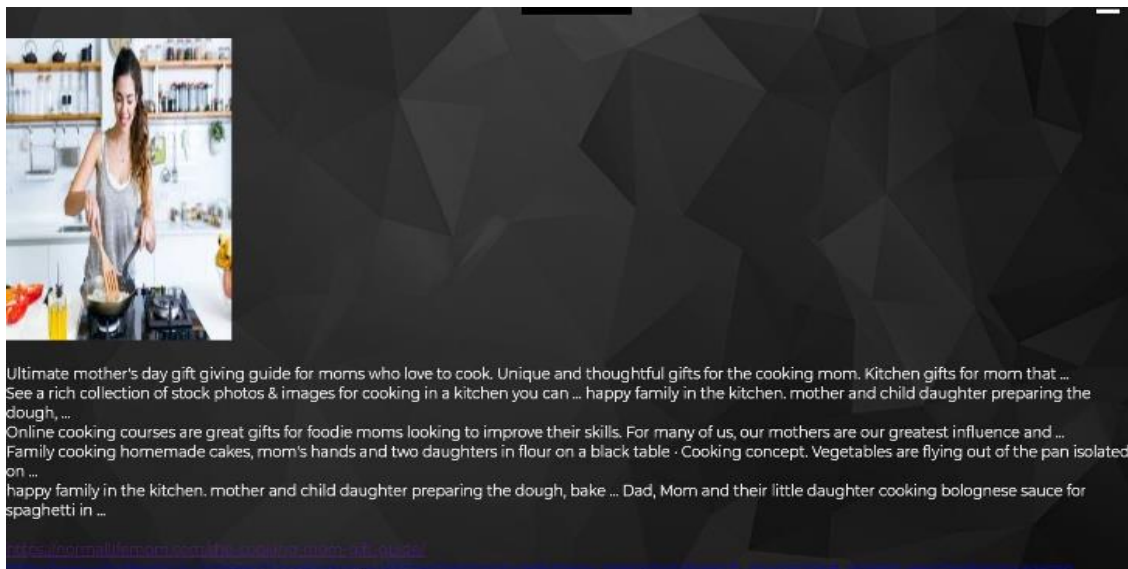


Fig 4.6 cooking sample

I have given a sample cooking picture it will generate captions like ultimate mothers day gift giving guide for mom's who love to cook .

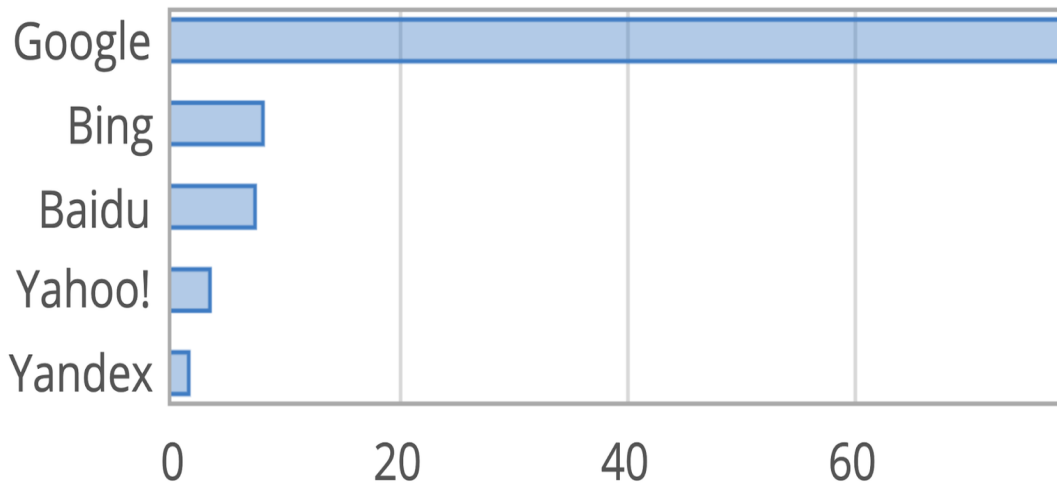


Fig 4.7 comparison graph

Google is getting 80% accuracy for image search engine than compared to other search engines.

		Twitter	Twitter	Twitter	Twitter	Twitter	Twitter	Twitter
Name	URL	Jan	Bill Gates	Neil P. Harris	Kim Kardashian	Katrina	Kaka	Hung
Google Image Search	https://images.google.com/	Y	Y	Y	Y	Y	Y	Y
TinEye	https://tineye.com	Y	Y	Y	Y	Y	Y	Y
Bing Image Search	https://www.bing.com/images/	X	Y	Y	Y	Y	Y	X
Yandex Image Search	https://www.yandex.com/images/	Y	Y	Y	Y	Y	Y	Y
Baidu Image Search	https://image.baidu.com/	X	Y	Y	Y	X	Y	X
Image Search Engine	http://image.so.com/	X	Y	X	Y	X	X	X
Sogou Search Engine	https://www.sogou.com/	X	Y	Y	Y	Y	Y	X

Table 4.1 Search engines outputs for twitter.

Google image search have found every person image than compared to other search engines.

		Linkedin	Linkedin	Linkedin	Linkedin	Linkedin	Facebook	Facebook	Facebook	Facebook
Name	URL	Alice	Iris	Kristian	Hanson	Jane	Petra	Phil	George	Irene
Google Image Search	https://images.google.com/	X	Y	Y	X	Y	X	Y	X	X
TinEye	https://tineye.com	X	X	Y	X	X	X	Y	X	X
Bing Image Search	https://www.bing.com/images/	X	X	Y	X	X	X	X	X	X
Yandex Image Search	https://www.yandex.com/images/	Y	X	Y	X	X	X	Y	X	X
Baidu Image Search	https://image.baidu.com/	X	X	X	X	X	X	X	X	X
Image Search Engine	http://image.so.com/	X	X	X	X	X	X	X	X	X
Sogou Search Engine	https://www.sogou.com/	X	X	X	X	X	X	X	X	X

Table 4.2 search engine outputs for linked in.

Finding images of people through linked in website.here also google gave better results than other search engines. That's the reason we will recommend google image search.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Thus in this proposed system we implemented to find depth information about the images given using reverse image search algorithms provided by google it will cross check all the data and also source of an image which it is available. And mainly the scam detection that is any harmful content present in the images we will generate the information about all the harm content clearly.

In the proposed model of image captioning and thus reverse image optimization, images are taken as the input from the user and efforts are made to formulate most specific and relevant description to the image. This description will be in the form of captions and given as the output to the users. These descriptions are the universal to prevent any misinterpretation. The search engine is directly queried using these descriptions so that the user gets the resultant images which most closely resemble the input image. The proposed system will help the users to find the relationships between objects in the image and information about the image.

5.1 FUTURE WORK:

In Future we can update the project to find the Anonymous identity reveal and also Unknown person detection. These two key features are very useful to take the project to high level and it will be useful to so many people.

References

1. Elamri C, de Planque T (2016) Automated neural image caption generator for visually impaired people. Department of Computer Science, Stanford University. <https://cs224d.stanford.edu/reports/mcelamri.pdf>. Accessed 18 Oct 2018
2. He X, Shi B, Bai X, Xia G-S, Zhang Z, Dong W (2016) Image caption generation with part of speech guidance. In: IEEE conference on computer vision and pattern recognition 2016, school59 Optimizing Reverse Image Search by Generating ... 631 of electronic information and communications, Huazhong University
3. Yang Z, Zhang Y-J, ur Rehman S, Huang Y (2016) Image caption with object detection and localization. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. <https://arxiv.org/pdf/1706.02430.pdf>
4. Zhou L, Xu C, Koch P, Corse JJ (2017) Image caption generation with text-conditional semantic attention. Robotics Institute, University of Michigan, Department of Computer Science, University of Rochester, Electrical and Computer Engineering, University of Michigan, 2017
5. Asakawa S, Ogata T (2017) Comparison between variational autoencoder and encoder-decoder models for short conversation. Proc Int Conf Artif Life Robot 22:639–642. <https://doi.org/10.5954/ICAROB.2017.OS1-4>
6. Chen J, Dong W, Li M (2017) Image caption generator based on deep neural networks. Department of Computer Science, University of British Columbia. https://www.cs.ubc.ca/~carenini/TEACHING/CPSC503-19/FINAL-PROJECTS-2016/image_caption_generator_final_report.pdf. Accessed 12 Nov 2018
7. Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: IEEE conference on computer vision and pattern recognition
8. Khurana K, Awasthi R (2013) Techniques for object recognition in images and multi-object detection. Int J Adv Res Comp Eng Technol (IJARCET) 2(4):13839. Xinpeng Chen, Lin Ma, Wenhao Jiang, Jian Yao, and Wei Liu. Regularizing rnns for caption generation by reconstructing the past with the present. In CVPR, 2018.

APPENDICES

A.PROJECT CODE :

```
def sr(su):
```

```
    code=dolmageSearch(su)
```

```
    return parseResults(code)
```

```
def dolmageSearch(full_url):
```

```
    # Directly passing full_url
```

```
    print("""Return the HTML page response.""")
```

```
    if python3:
```

```
        returned_code = bytesIOModule.BytesIO()
```

```
    else:
```

```
        returned_code = StringIO()
```

```
    # full_url = SEARCH_URL + image_url
```

```
    conn = pycurl.Curl()
```

```
    if python3:
```

```
        conn.setopt(conn.CAINFO, certifi.where())
```

```
    conn.setopt(conn.URL, str(full_url))
```

```

conn.setopt(conn.FOLLOWLOCATION, 1)

conn.setopt(conn.USERAGENT, 'Mozilla/5.0 (Windows NT 6.1; Win64;
x64; rv:61.0) Gecko/20100101 Firefox/61.0')

conn.setopt(conn.WRITEFUNCTION, returned_code.write)

conn.perform()

conn.close()

if python3:

    #print(returned_code.getvalue().decode('UTF-8'))

    return returned_code.getvalue().decode('UTF-8')

else:

    #print(returned_code.getvalue())

    return returned_code.getvalue()

def parseResults(code, resized=False):

    """Parse/Scrape the HTML code for the info we want."""

    soup = BeautifulSoup(code, 'html.parser')

    results = {

        'links': [],

```

```

        'descriptions': [],

        'titles': [],

        'similar_images': [],

        'best_guess': "
    }

    for div in soup.findAll('div', attrs={'class':'g'}):

        sLink = div.find('a')

        results['links'].append(sLink['href'])


    for desc in soup.findAll('span', attrs={'class':'st'}):

        results['descriptions'].append(desc.get_text())


    for title in soup.findAll('h3', attrs={'class':'LC20lb DKV0Md'}):

        results['titles'].append(title.get_text())


    for similar_image in soup.findAll('div', attrs={'class':'rg_meta'}):

        tmp = json.loads(similar_image.get_text())

        img_url = tmp['ou']

        results['similar_images'].append(img_url)

```

```
for best_guess in soup.findAll('a', attrs={'class':'fKDtNb'}):  
    results['best_guess'] = best_guess.get_text()
```

PROJECT REPORT

by Plagiarism Checker

Submission date: 26-Feb-2021 04:26AM (UTC-0500) **Submission ID:** 1518704317

File name: IT-01_Paper_publication_2_1.pdf (386.95K) **Word count:** 2059

Character count: 10507

An In-Depth Look at the Images for Finding Information using Deep learning and Reverse Image Search

Y Venkata Sai¹ , Mr. Salman², Dr. T Sasikala³

Department of Information Technology , Sathyabama Institute of Science and Technology, Chennai

Abstract : An In-Depth Look at the Images for Finding Information using Deep learning and Reverse Image Search tells Modern Technology requires a modern solution to find the information from an image. Generating a caption from an image may require computer vision and natural language processing concepts for generating the caption in a natural language like English etc. In this paper, we tried to show how more information can be generated from an image for further analysis of making some predictions. This project is able to detect the person's identity if a person's face is visible in the image as well as all about the image as to where it has taken, resolution, and all other factors. The entire project will describe as much information as possible and use it for further analysis. For example, given a white lion image then the image may tell the output as "The white lion is a rare color mutation of the lion, specifically the Southern African lion." or if an image contains a person as an anonymous then the output will be like "Name, Address and lot more". It can give you complete information about an image, apart from the basic information and the EXIF data, it also shows other useful and in-depth data. There are some reasons that why you need to get information from a photo if a photo you saw that selling a product at a cheap price then we definitely may be interested to buy but wait that may be a scam or what if someone send you the friend request that is the real or fake person that also we have to be aware and what if you are going to meet with a wrong people if he used a fake photo last time with you and you are going to meet with the wrong guy.

I. Introduction: In this project there are four modules, In the first module we implemented the finding the information from an image using reverse image search and in second module we will generate caption from an image with more accuracy and finally we are generating text from a given image.

Keywords: deep learning, CNN, Image processing, Image captioning, python.

II. Literature Survey

Elamri and Planque [1] he created this model mainly for physically impaired people so that they can clearly understand about surroundings. One of the main modules is generating captions for images in an automatic way via web which is good for impaired people. For extracting image features he used convolutional neural networks. the generated captions are in valid English. This model is more capable of generating highly precise captions.

Yang et al. [2] designed one of the best method for generating captions to an image using some good pretrained vectors. This will cover a vast diversity of images. He uses a methods that is developed by oxford university with is input features in addition to the new features provided by user specific dataset thus it will be like very highly developed for object detection.one of the main mechanism is "attention mechanism" which is used to extract features from images. Which mainly features are extracted using probability distribution.

Zhou et al. [3] he presented one of the best models till now presented on state of art concept models until 2016. The authors mainly discuss about the features of text on conditional-based attention; it , mostly focuses on training the model on generating captions rather than extracting image feature. here this model is trained mostly to detect objects which is detected previously by trained model. And this model more enough to recognize the relations between the objects of an image.

Asakawa et al. [4] trained the model in mostly unrelated environment like where the result is unknown by remembering the very long term details in memory. In order to go get a full reference the best robust system is developed using autoencoders. This methods using paragraphs and other context n auto encoders were found to have better results in methodology .this method having an encoders and decoders both will work recursively in getting by giving words as an inputs and generating sentences . Auto encoders gave best results as the standard recursive models where if improved auto encoders will be increasingly used for extracting image features. And generating longer passages as descriptions for an image

Chen et al. [5] replaced the convolutional neural networks with three architectures mainly VGGNet, AlexNet, GoogleNet . it is identified that the VGGNet is highly performed according to BLEU score . A small version of Gated Recurrent Unit is mostly used as a new recurrent network which is

implemented using Caffe. It will highly saves memory and very faster in training sessions .

III Methodology Adopted

It is that a pre-trained CNN model like Resnet50 can extract 2048 features from any image (without the fully connected layers). The images in your database can be indexed based on those features (an array/vector of floating points) and then you could find the Euclidean distance of the features of the query image vs all other images in your database.

When you sort based on this distance, you should get similar images with the smallest distance.

There's one catch with Euclidean distance based ranking of images - all of the 2048 features are given equal priority. Lets assume for simplicity sake that the "blue sky" is feature number 2000 in the vector and the "brown building" is feature number 1000. Its actually more complicated than this and almost impossible to identify which features stand for what in an image. However, we do know that they do stand for "something". With Euclidean Distance matching, the blue sky will be given equal importance with the brown building. As humans, we know by context that the building in the foreground is more important than the blue sky in the back ground.

It turns out that context awareness can also be incorporated in the image search by training a neural network to "look at" two vectors of two images and spitting out a score for similarity. This could be a supervised learning task, with features of similar images as inputs, and a synthetic ground truth as output.

I agree with Zhou et al assumption that that "Google uses some combination" of techniques and "ranks the results using a proprietary algorithm". [3]

I think most important algorithm used for ranking their results is based on Image compression technique using Auto-Encoders.

I can clearly say that Auto-Encoders are mostly image compression and reconstruction. There is a gray-scale image which is labelled as 'Original' of dimensions 32*32 pixels (total of 1024 pixels).

The structure of Auto-Encoder used for compression here is 1024-256-64-8-64-256-1024, which means that a 1024 Dimension image is being compressed to just 8 Dimensions. And in the second part of the image it can be seen how well the 'Auto-Encoder' is able to reconstruct the 'Original' image. Just beside that, we can see how badly PCA algorithm reconstructs the same image after compressing the original image to 8

Dimensions (i.e. taking first 8 principal components only).

System Architecture

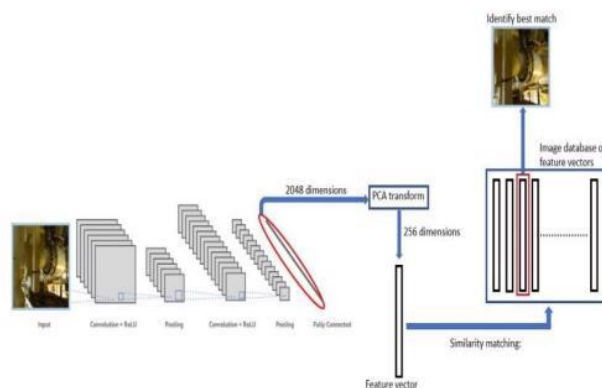


Fig.1 System architecture

1. Analysis of methods for finding identical images:

For the choice of methods for analysis and the search for identical images, it is necessary to explore the methods of indexing images; definition of a measure of similarity; methods used in the construction of detectors and descriptors. The result of the research is the basis for the design of an information reverse image search system. To obtain exactly identical, but not similar images, the method of hashing for the average value was analyzed, and for the determination of similarity dimensions, the

Hamming distance was used. The main objective of the study is to determine the threshold function, which allows asserting that the images are complete duplicates.

2.Pre-processing: PS: Auto-Encoder compression and PCA are NOT loss-less compression techniques.

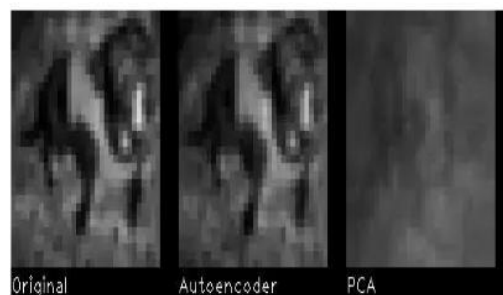


Fig 2

So for every image of this dimension, it can be mapped to a point in 8-D space by compressing using Auto-Encoders. And if some images are

visually similar, then plotting the points of the compressed images in 8-D space shows that they all lie very close to each other (using Euclidean for distance calculations).

Or in other words, we can say that, all the points that lie close to each other in the compressed 8-D space are visually similar.

3. Visual Similarity Search:

Below is the set of images for similar images generated for query image and this is how to reverse image search generates the similar images and that we can use to obtain the captions.



Fig 3

The given Features are stored for indexing and matching the Google mostly use this in combination of keywords to refine searches on most large collections.

4. Reverse Search Algorithm:

Google is a very hardcore engineering-oriented company. The Google map reduces all day long. Images get mapped to features. Features get mapped to new features. Feature selection chooses subsets. Then they get hashed. Then they get quantized. Then some more learning etc.

One thing about google we can say this is single most amazing and single most upsetting thing about vision research at Google. The whole process works because the engineers are top-notch, really care about writing good code, and they aren't stressed out (so they can write high-quality code quite often)

IV Results and Discussion

1.Home page : In this page we will give introduction about our project and at top right menu will be there . In that you can click you option and it will proceed you to next page.



Fig 4 Home page

2. Login Page : you need to login credentials and login to upload pictures.



Fig 3 Login page

3. Upload Page : There will be photo upload section you can upload photo and it will generate related captions for particular image. And source links of an image.



Fig 4 Upload page



Fig 5 Sample output

I have uploaded London Big ben picture I will generate the caption as A rich collection of photos and images for big ben houses of parliament and source links of an image.



Fig 6 Teaching sample

Here I have uploaded the sample teaching picture it will generate caption like teachers are advised that explicit teaching is the best way to teach.



Fig 7 cooking sample

I have given a sample cooking picture it will generate captions like ultimate mothers day gift giving guide for mom's who love to cook .

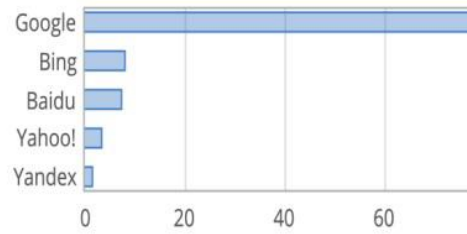


Fig 8

Google is getting 80% accuracy for image search engine than compared to other search engines.

		Twitter	Twitter	Twitter	Twitter	Twitter	Twitter	Twitter
Name	URL	Jan	Bill Gates	Niel P. Harris	Kim Kardashian	Katrina	Kala	Hung
Google Image Search	https://images.google.com/	Y	Y	Y	Y	Y	Y	Y
TrEye	https://trEye.com	Y	Y	Y	Y	Y	Y	Y
Bing Image Search	https://www.bing.com/images/	X	Y	Y	Y	Y	Y	X
Yandex Image Search	https://www.yandex.com/images/	Y	Y	Y	Y	Y	Y	Y
Baidu Image Search	https://image.baidu.com/	X	Y	Y	Y	X	Y	X
Image Search Engine	http://image.se.com/	X	Y	X	Y	X	X	X
Sogou Search Engine	https://www.sogou.com/	X	Y	Y	Y	Y	Y	X

Table 1

Google is giving clear results for every person identity we opted as search engine for images.

		LinkedIn	LinkedIn	LinkedIn	LinkedIn	LinkedIn	Facebook	Facebook	Facebook	Facebook
Name	URL	Alice	Iris	Kristian	Hanson	Jane	Petra	Phil	George	Irene
Google Image Search	https://images.google.com/	X	Y	Y	X	Y	X	Y	X	X
TrEye	https://trEye.com	X	X	Y	X	X	X	Y	X	X
Bing Image Search	https://www.bing.com/images/	X	X	Y	X	X	X	X	X	X
Yandex Image Search	https://www.yandex.com/images/	Y	X	Y	X	X	X	Y	X	X
Baidu Image Search	https://image.baidu.com/	X	X	X	X	X	X	X	X	X
Image Search Engine	http://image.se.com/	X	X	X	X	X	X	X	X	X
Sogou Search Engine	https://www.sogou.com/	X	X	X	X	X	X	X	X	X

Table 2

VI Conclusion

Thus in this proposed system we implemented to find depth information about the images given using reverse image search algorithms provided by google it will cross check all the data and also source of an image which it is available. And mailly the scam detection that is any harmful content present in the images we will generate the information about all the harm content clearly.

VII References

1. Elamri C, de Planque T proposed Automated neural image caption generator for visually impaired people. Department of CS, Stanford University.
2. He X, Shi B, Bai X, Xia G-S, Zhang Z, Dong W proposed the Image caption generation with part of speech guidance..
3. Yang Z, Zhang Y-J, Image caption with object detection and localization. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
<https://arxiv.org/pdf/1706.02430.pdf>.
4. Zhou L, Xu C, Koch P, Corse JJ Image caption generation with text-conditional semantic attention. Robotics Institute, University of Michigan, Department of Computer Science, University of Rochester, Electrical and Computer Engineering, University of Michigan, 2017.
5. Asakawa S, Ogata T (2017) Comparison between variational autoencoder and encoder-decoder models for short conversation. Proc Int Conf Artif Life Robot 22:639642. <https://doi.org/10.5954/ICAROB.2017.OS1-4>
6. Chen J, Dong W, Li M (2017) Image caption generator based on deep neural networks. Department of Computer Science, University of British Columbia.
https://www.cs.ubc.ca/~carenni/TEACHING/CPS C50319/FINALPROJECTS2016/image_caption_generator_final_report.pdf. Accessed 12 Nov 2018.

IT-01 report 2

ORIGINALITY REPORT

14	5	11	3
%	%	%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

11	
"AApdpvliacnactieodn sC",o Smppruintignegr TSecciehnncielo agnieds B aunsdiness	% Media LLC, 2020
Publication	
compleannocapisce.icu	1
Internet Source	%
Submitted to Arizona State University	1
3	%
Student Paper	
1	
4 www.ijsar.in	%
Internet Source	

Exclude quotes

On

Exclude matches

Off Exclude bibliography

On