Spring 5-20-2019

Image Retrieval Using Image Captioning

Nivetha Vijayaraju
San Jose State University

Image Retrieval Using Image Captioning

A Thesis

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Nivetha Vijayaraju

May 2019

The Designated Project Committee Approves the Project Titled

Image Retrieval Using Image Captioning

By

Nivetha Vijayaraju

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

Spring 2019

Dr. Robert Chun, Department of Computer Science

Dr. Katerina Potika, Department of Computer Science

Dr. Thomas Austin, Department of Computer Science

Abstract

The rapid growth in the availability of the Internet and smartphones have resulted in the increase in usage of social media in recent years. This increased usage has thereby resulted in the exponential growth of digital images which are available. Therefore, image retrieval systems play a major role in fetching images relevant to the query provided by the users. These systems should also be able to handle the massive growth of data and take advantage of the emerging technologies, like deep learning and image captioning. This report aims at understanding the purpose of image retrieval and various research held in image retrieval in the past. This report will also analyze various gaps in the past research and it will state the role of image captioning in these systems. Additionally, this report proposes a new methodology using image captioning to retrieve images and presents the results of this method, along with comparing the results with past research.

Keywords – Image retrieval, deep learning, image captioning

Acknowledgments

I would like to thank my advisor Dr. Robert Chun for his continued support and providing me with the guidance necessary to work on this project. I would also like to thank my advisor for teaching me the core skills needed to succeed and reviewing this research topic. I would also like to thank my committee members Dr. Katerina Potika and Dr. Thomas Austin for their suggestions and support.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Image retrieval is the process of retrieving images relevant to a query from large database systems. The rapid growth in internet and social media platforms like Flickr, Instagram, etc. has resulted in the massive growth of images available online. Hence, to get images of interest, image retrieval systems play a major role. This is a challenging problem as the system should understand the subcomponents of an image and understand the complete context of the image. An ideal image retrieval system should display images which are more relevant to the query.

Image retrieval by a textual query is being used in most of the image search systems. The search of images using textual query primarily depends on metadata of images. Images with metadata similar to the textual query are displayed as results. The above methodology relies on humans to annotate images. Desired results might not be obtained if there is an error in human annotations of metadata or if the metadata doesn’t define the context behind an image.

Some image retrieval systems also pass a query as an image to get similar images. This is one of the challenging problems as it is hard to get the context of the input query, and retrieving images with similar context is difficult. Current image retrieval systems use features such as color, shape, etc. to find similar images. The results of these systems are not as close to the query as expected. This is because many images with similar color

and shape may be totally irrelevant and such irrelevant images can be obtained in the results. Hence, research to address these problems in image retrieval is necessary.

Current research in this domain includes image retrieval from annotated images and Content-Based Image Retrieval (CBIR). In CBIR, features of images such as color, shape, etc. are used to search images. Research in image retrieval by image captioning has increased due to the advancement in neural networks and processing power in recent years. Captions are mostly generated using deep learning by making use of Convolutional Neural Networks (CNN) to detect features from images and Recurrent Neural Networks (RNN) to generate captions for the detected features.

This literature survey and project focuses on exploring various methodologies used in image retrieval research and answers these questions: What are the various methodologies used in image retrieval in the past research? What are some gaps in the existing research? How can image captioning facilitate image retrieval? This survey uses references from published papers and conference journals to answer these questions. It also provides a new methodology in image retrieval by using image captioning methodologies and evaluates it against the past experiments in image retrieval.

CHAPTER 2

Background

Image captioning is the process of generating text for images. This process includes object detection in images followed by generation of corresponding descriptions. The advancement in technologies in the recent days in Artificial Intelligence and Computer Vision such as processing power and large Image datasets have facilitated the research in Image Captioning. Natural Language Processing is used along with Computer Vision to generate captions. Image captioning relies on both images and languages to develop a model. Image Captioning is predominantly used in image search applications, robotics, social networks and helps in conveying information to visually challenged people.

Now, research in image captioning has increased due to the advancement in neural networks and processing power. Initially, image captioning started with object detection in images. Captions were generated with the help of datasets using the nearest neighbor algorithm. Recently, captions are generated using deep learning.

2.1 Features in Images

Features determine the property of any data in the dataset. A dataset of images does not have features explicitly and hence should be extracted from the images. Based on the problem, machine learning models and computational limitations, features are chosen to predict the class of data.

2.1.1 Histogram

Histogram of an image denotes the frequency of pixels belonging to a different color or intensity ranges in an image. The histogram can be a color histogram if it denotes the distribution of tonal colors in an image. In an intensity histogram, the frequency of pixels in different levels of intensity is extracted. Intensity histogram is used in monochromatic images.

A histogram is represented as a bar graph where X-axis denotes the different intensity or color tones in an image and Y-axis denotes the number of pixels in the image with a particular tone. These different tonal ranges and their corresponding pixel values are taken as features in image processing problems.

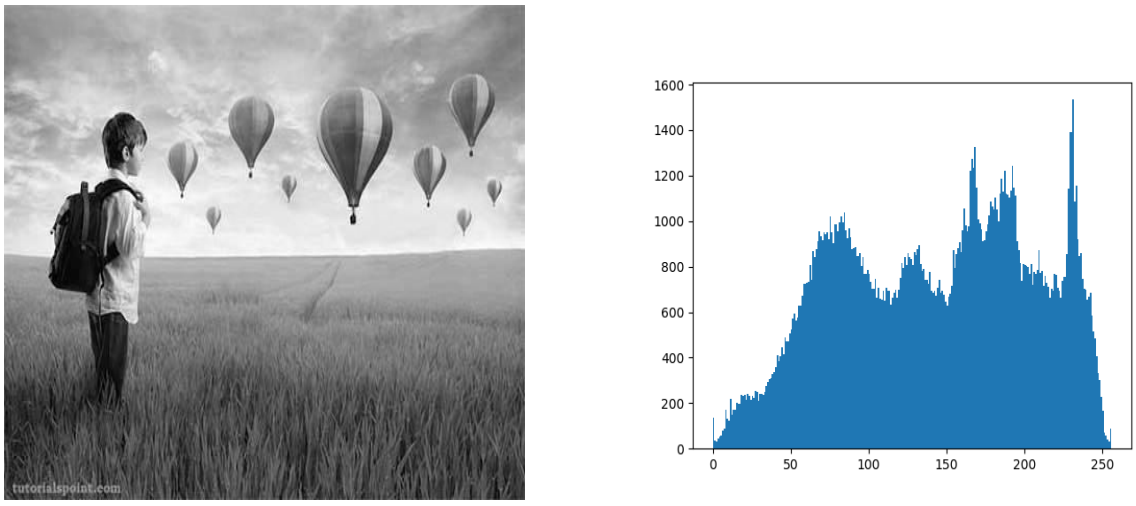

Figure 1: Histogram of an image [26]

2.1.2 Histogram of Oriented Gradients

The Histogram of Oriented Gradients (HOG) makes use of the location of objects in an image and its shape and are represented as oriented gradients. These oriented gradients are determined from the shape of an object and also provides the edge directions. In this method, the image is divided into cells, and pixels within these cells are represented as HOG. The final HOG descriptor concatenates all the results from local cells.

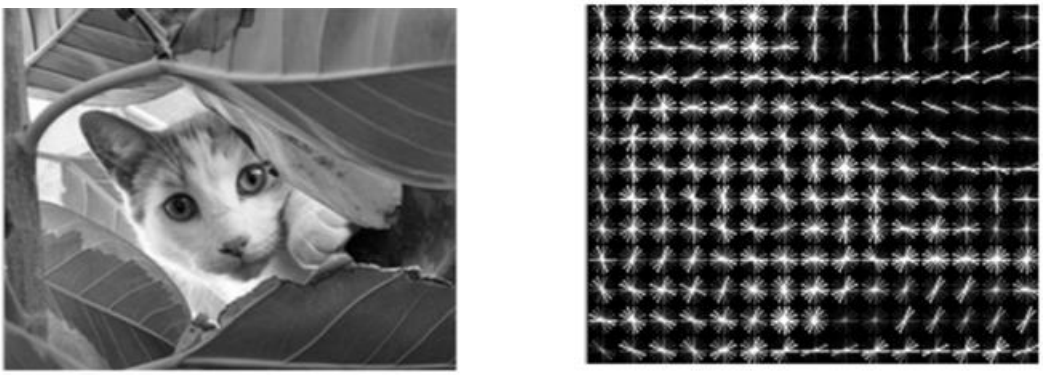

Figure 2: Image and its HOG descriptor

This algorithm follows the following steps to return features:

1. Computation of gradients
2. Cell histogram creation
3. Normalization of gradient strengths
4. Block Normalization
5. Object Recognition

2.1.3 Edges

Edges of an image are found by the sudden change of brightness or color in an image. Edge detection helps in various object detection problems. The edge detector helps in finding the boundaries of objects in an image. This provides features which are of more importance and relevance to the problem thereby saving computational time.

Edge detection methods can be grouped into two categories: search-based and zero-crossing. In a search-based approach, the edge strength in the image is computed by taking a first-order derivative, and then by finding the local maxima, the direction of the edges is found. In the zero-crossing method, zero-crossing of edges are found by taking a second order derivative computed for edge detection. Smoothing is applied on the images as part of pre-processing in edge detection and Gaussian smoothing approach is used predominantly.

Figure 3: Image and its Canny edges

Among various edge detection algorithms, Canny edge detector acts as the benchmark for other edge detection algorithms. In this approach, an optimal smoothing filter is derived by considering detection, localization and minimization of responses to an edge. In this approach, edge points are found in pre-smoothing filters by finding the local maximum in the gradient direction.

2.2 Object Detection

Object detection is the process of detecting or locating objects in an image. Any image captioning system depends on the principle of object detection to generate captions. Object detection using machine learning makes use of features extracted from one of the approaches mentioned above or any other feature extraction methodology from images. These features are then fed into a machine learning model like Support Vector Machine (SVM) to perform classification. In object detection using deep learning, features are not specified and perform complete object detection typically using Convolutional Neural Networks (CNN). Some of the deep learning algorithms used in object detection are RCNN, YOLO, SSD, etc.

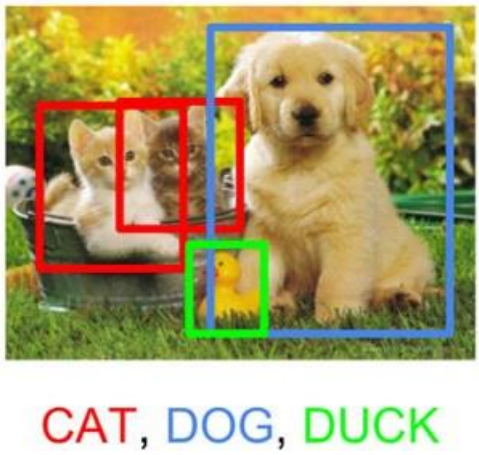

Figure 4: Object detection in an image [27]

2.3 Image Segmentation

Image segmentation is the process of segmenting an image into smaller regions and classifying those regions to an object. Segmentation makes an image easier to analyze as the image is represented in a more meaningful form. This process typically assigns a label to every pixel in the object, and all the pixels having the same label have the same characteristics which can help in boundary detection and object detection.

Figure 5: Segmentation in an image [27]

Some of the common approaches used in image segmentation are:

1. Thresholding
2. Clustering
3. Motion and interactive segmentation

4. Region-growing methods
5. Partial differential equation-based methods
6. Graph partitioning methods

2.4 Deep Learning in Image Captioning

Most of the image captioning processes use deep learning models to generate captions. Image captioning system typically has two steps:

1. Feature Extraction
2. Caption Generation

The above steps are implemented using neural networks. Features are extracted from images using Convolutional Neural Networks (CNN) and captions are generated from features using Recurrent Neural Networks (RNN).

2.4.1 Neural Networks

Neural networks are used in computing and Artificial Intelligence which is predominantly inspired by biological neural networks. They act as a framework which learns from continuous data and does not have an algorithm to predict the output. In image detection problems, the neural network learns from a training data of images and uses this knowledge to classify a new image. Hence, a neural network can work in image classification problems without any prior knowledge.

An Artificial Neural Network (ANN) consists of nodes or artificial neurons inspired by biological neurons and edges inspired from synapses. Each edge transfers signals between neurons, and each neuron emits whenever the signal is greater than a given threshold. These edges have weights, and with continuous learning, the weights among

the edges are recomputed. These weights determine the strength of the signal passing through an edge.

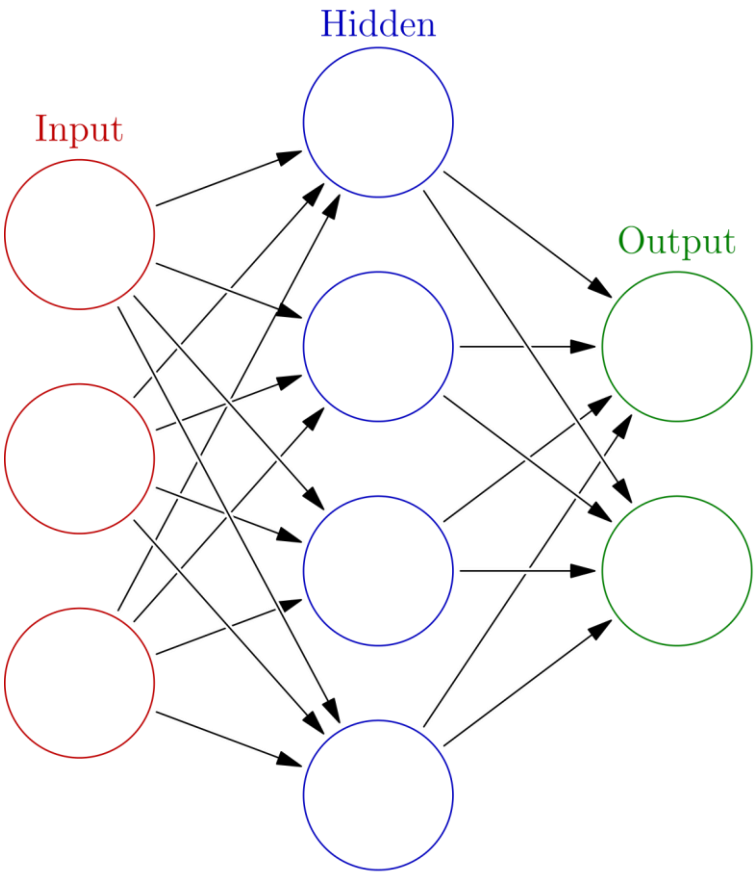

Figure 6: Artificial Neural Network [28]

Artificial neurons are separated as layers. ANN consists of an input and output layer and multiple hidden layers. Signals are passed into the input layer and after many internal traversals return the result in the outer layer. Neural networks are used in many applications such as computer vision, medical diagnosis, games, speech recognition, facial recognition and machine translation.

2.4.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep, feed-forward neural network which is mostly used in image processing applications. CNNs require little preprocessing as it learns the features by itself from the given data. A CNN consists of an input layer, an output layer and multiple hidden layers. The hidden layers of a CNN include convolutional layers, pooling layers, fully connected layers and normalization layers.

1. Convolutional layer

   This is the first layer which extracts features from images. It performs a mathematical operation taking the image and filter as input and returns a Feature map as output. This layer helps in detecting edges and performs blurring and sharpening operations using filters.

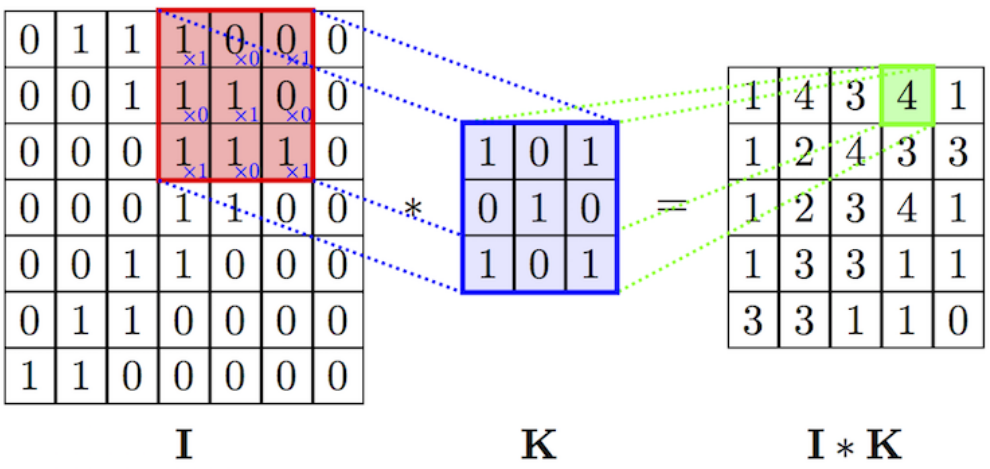

Figure 7: An example of a convolutional layer [29]

2. Pooling layer

   Pooling layer is used when the images are large as it reduces the number of parameters in an image. Spatial pooling reduces the number of dimensions in a map by retaining the most important information. This can be of different types:

   - Max Pooling
   - Average Pooling
   - Sum Pooling

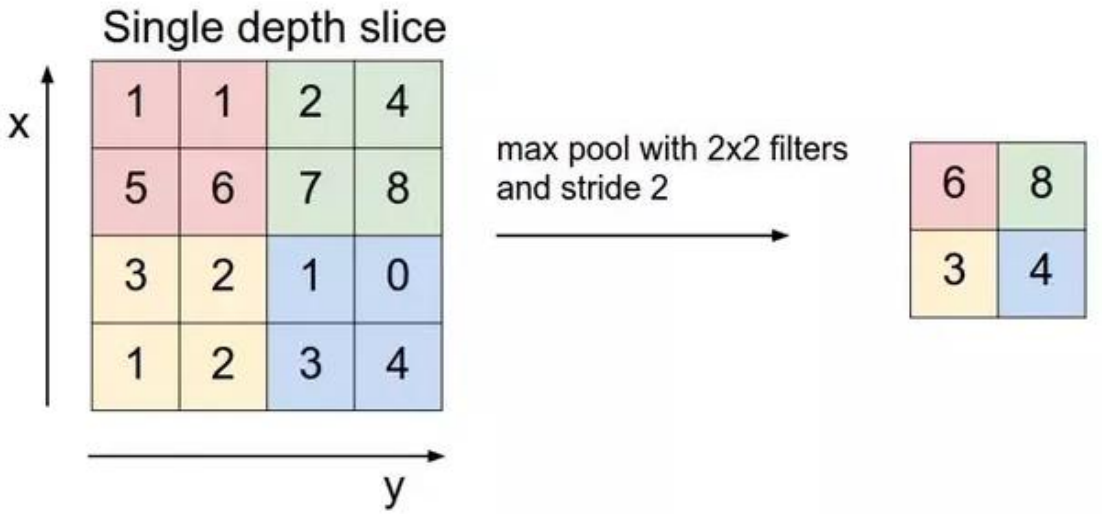

Figure 8: An example of Pooling Layer [30]

3. Fully connected layer

   Matrix is flattened into a vector and is passed into a fully connected network. Features are passed into this which creates a model to determine the weights of the edges. In the end, we have an activation function as a softmax or sigmoid function to classify the images.

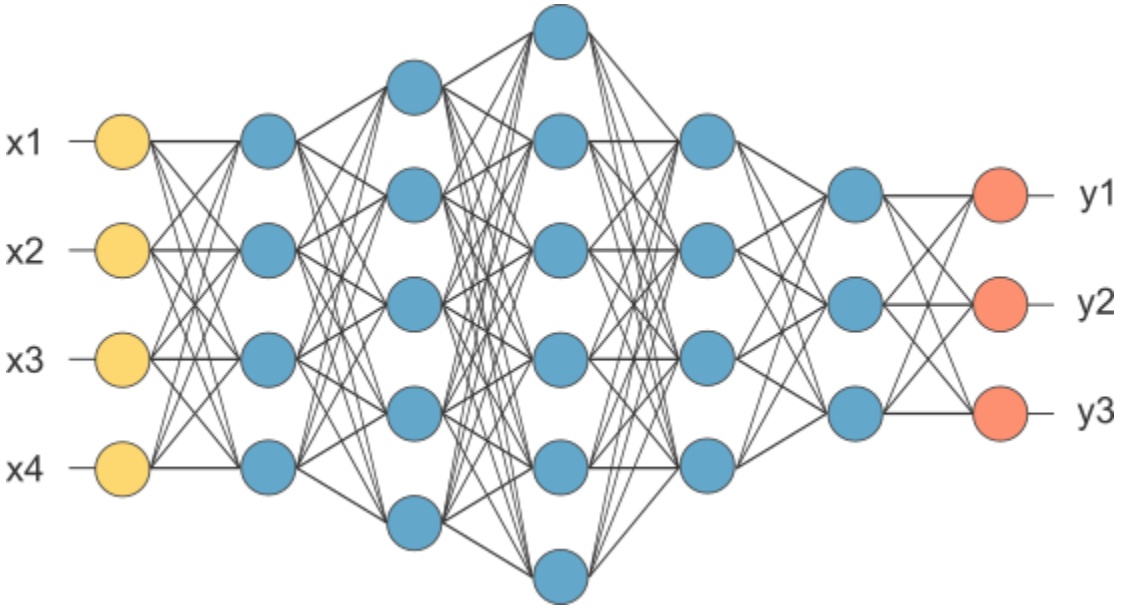

Figure 9: An example of a Fully connected layer [30]

4. Normalization layer

   This layer helps in normalizing the outputs from a previous layer by changing the output values between 0 to 1. This layer helps in making the model to train faster. This layer can also make some activation functions work for a longer time as activation functions like sigmoid loses its gradient value soon without normalization layer. Hence, this corresponds to higher learning rate of the model.

2.4.3 Recurrent Neural Networks

A Recurrent Neural Network (RNN) has nodes connected in the form of a directed graph like a sequence. This facilitates RNN to handle series operations like time sequence, handwriting recognition and speech sequence. RNNs can remember important information about the input received and hence enables them to predict the next element in a sequence. In RNN, the information loops between the nodes. RNNs predict results by considering the current input and also uses the previous inputs before making a decision.

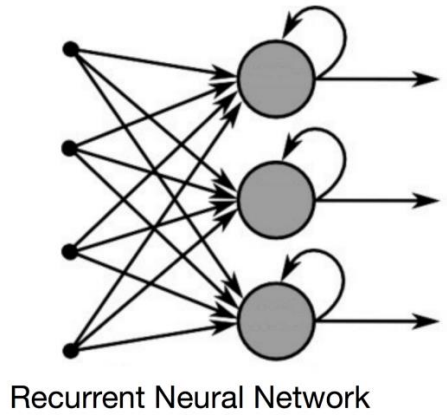

Figure 10: Sample Recurrent Neural Network [31]

RNN usually has a short-term memory and hence cannot handle very long sequences. Long Short-Term Memory (LSTM) network extends RNN which extends the memory of RNN. Therefore, LSTM can be employed in problems with sequences having long gaps. LSTMs can remember the previous inputs for a long duration as it stores all those data into a memory. In image captioning problems, captions are generated from image features using RNN along with LSTM.

CHAPTER 3

Related Work

3.1 Types of Image Retrieval

Image retrieval is primarily done by extracting features from images or by tagging images with keywords. Techniques such as hashing, Content-Based Image Retrieval (CBIR), Sketch-Based Image Retrieval (SBIR), etc. are employed in retrieving images. In some cases, these techniques also employ deep learning to get better results.

3.1.1 Hashing Methods

Hashing has been employed in many image retrieval problems in recent years due to its computational efficiency and speed. Hashing converts the features extracted from images into binary codes and these binary codes are used to search the images.

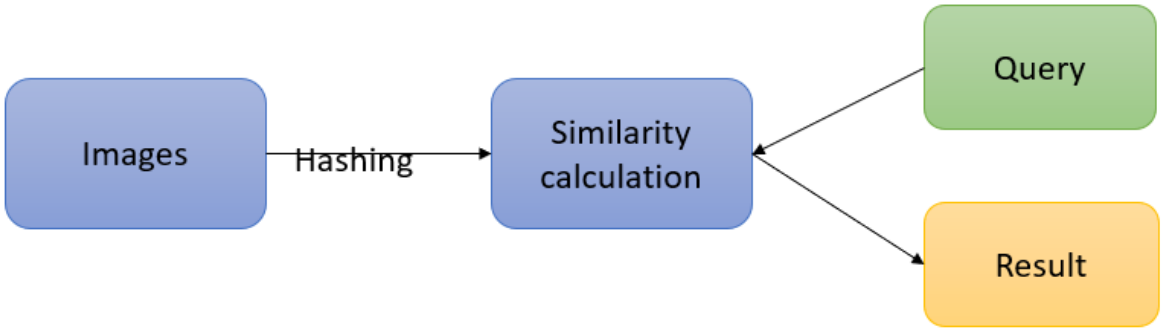

Figure 11: Hashing Based Image Retrieval

In [1], features are extracted from images using Hierarchical Recurrent Neural Network (HRNN); Hashing is used on these features and the resulting binary codes are used in image retrieval. Jin et al. [2] has also extracted features as done by Lu et al. [1]

and has performed similarity calculation among all the images in the dataset and based on the score, hash codes are generated for each image. Some hashing techniques use location and object detection to generate hashing. Location of an object in images is used by Huang et al. [3] to generate hash codes by using a filter called automatic mask. Like [1], Huang et al. [3] make use of features to generate hash code. Instead of extracting multiple features, [3] uses the location of objects in images as the feature and generates hash code.

3.1.2 Content-Based Image Retrieval

CBIR makes use of contents of the image such as objects instead of annotation or tagging techniques to search images. Features such as color, texture, shape, etc. are used in CBIR.

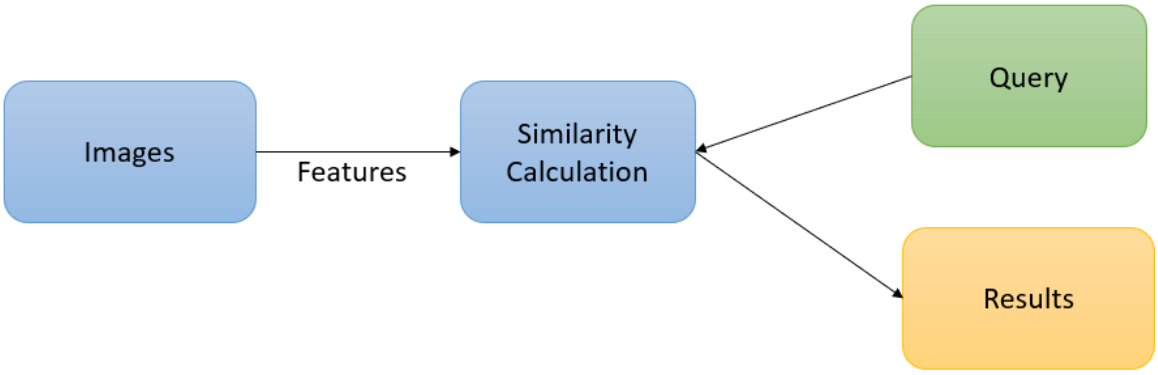

Figure 12: Content-Based Image Retrieval

A database table is created for the images in the dataset by using color, shape, texture as features and images are retrieved using similarity calculation between the

query and images in the dataset by Sreedevi et al. [4]. In [5], the above CBIR techniques are used on a region of interest. In the input query, the region of interest is mentioned, and similarity calculation is performed only on the given region of interest in the image to retrieve similar images. In [11], Correlated Primary Visual Texton Histogram Features (CPV-THF) are found in images and this feature is used to distinguish among other images in the dataset. CPV-THF integrates the visual content and semantic features of the images by using color, texture and histogram of the images.

Even though CBIR provides good results, it takes more time due to the heavy computations required to extract features from images. Hence, to lessen the computational time, Bing et al. [6] have proposed a parallel image retrieval system using the Browser/Server (B/S) mode to extract features and to perform the similarity calculation.

3.1.3 Sketch-Based Image Retrieval

SBIR makes use of edges and contours of images to generate sketches. These sketches are used as features to retrieve images. In [7], sketches are generated for the images in the dataset. These sketches are used as keys in an inverted index data structure and images containing the sketches are used as values, and with this mechanism, images are retrieved. Xu et al. [8], have incorporated the same methodology proposed by [7], but in addition to that, they have made use of continuous learning in the image retrieval system. In [7], a new set of images cannot be added in the database but in [8], we can add new images whenever needed as the system generates sketches dynamically and stores it in the database. Though [8] is computationally expensive compared to [7], the results are good in [8] due to continuous learning.

Figure 13: Sketch-Based Image Retrieval [32]

3.1.4 Image Retrieval by Deep Learning

Many image retrieval systems use deep learning to get more features out of images. In [9], features are extracted from different regions of the image in patches using Convolutional Neural Networks (CNN). The distance between the query and dataset is calculated by using the Hamming distance between the query and the generated patches to get similar images. Instead of retrieving features in patches, deep features are extracted in [10] using CNN to facilitate image retrieval. Deep feature extraction takes more computational time compared to feature extraction in patches due to the large size of images, but these deep features are used in discriminating the differences among various images in the dataset which is not possible in [9].

3.1.5. Other Image Retrieval Methods

In [12], local and global features of images are extracted, and tag information of images is also used in discriminating images. In [13], Cross Regional Matching (CRM)

techniques are employed to compare image similarity among various locations and directions in the dataset. As there are many approaches used in image retrieval on various datasets, many times the best approach depends on the problem and the dataset, and hence researchers should use the model based on the problem and not entirely rely on the past results.

3.2 Approaches in Image Captioning

Image captioning problem primarily depends on two important steps: Feature extraction from images and Caption Generation.

3.2.1 Attributes for Image Captioning

Wang et al.[24] use Convolutional Neural Networks (CNN) for feature extraction. In the training data, the skeleton sentences and corresponding attributes are extracted. Two Long Short-Term Memory (LSTM) units are used where one is used to generate skeleton sentences from the image and the other one is used to extract attributes from the image. For any given test data, a skeleton sentence is generated first, and then after attribute extraction, attributes are added to the existing skeleton sentence. The final sentence contains the skeleton sentence along with the attributes.

Similar to [24], [15] also uses attributes to generate captions. In [15], CNN pretrained on single label images is used and this is fine-tuned with a multi-label image dataset. For a given test image, the attributes are generated as multiple regions and fed into the combined CNN which is aggregated with max pooling which can generate multi-label captions for the given test image.

3.2.2 Localized Features in Image Captioning

Some image captioning approaches use sub-blocks of images as features and generate captions for those features and generate captions for the image. A similar approach is used in [14] where the image is segmented hierarchically, and each segment is passed as a feature into a CNN which is pretrained for object recognition. This method uses a visual encoder which encodes the visual features and passes this into a scene specific decoder which is used for generating text. There is also a scene vector extractor which extracts the global context of the image. This scene vector facilitates text generation from the decoder to align with the global context of the image.

3.2.3 Probabilistic approach

Image captioning with probabilistic approach assigns a probability for each caption for a given image and the sentence with the highest probability to an image is returned as a caption. Dai et al.[16] use a probabilistic approach in which each image in the training data is assigned the highest probability to its corresponding caption and low probability values to captions of other images in the dataset. Whenever a test image is passed, the new model should give a higher probability of positive pairs of image and caption and lower probability value to negative pairs.

Vedantam et al.[18] have generated context-aware captions using a probabilistic approach. Here they have used emitter suppressor beam search algorithm on RNNs to generate captions. First, a caption is generated describing an image, and with a discriminator class trained on the model, common tags or words in the caption and the discriminator class are suppressed. Based on the initially emitted caption and suppressed caption, a new set of captions with discriminating tags are generated for the image.

Similarly, in [19] probabilistic approach is used for the various semantic concepts present in an image instead of captions. For a given image, the probability of semantic concepts or tags in an image are generated using a CNN and these tags are passed as parameters into LSTM where captions are generated based on the probabilities of individual semantic concept in the image. Semantic Compositional Networks (SCN) are extended along with the weight matrix of LSTM to make them tag-dependent to generate captions.

3.2.4 Stylish caption generation

Other than the usual caption generation, certain approaches generate captions in a stylish manner. Chuang et al.[17] have proposed a StyleNet framework which is aggregated with three LSTM networks. Features from images are extracted using CNN and they are passed into LSTM to generate captions. In this approach, they use a special LSTM called Factored-LSTM. All the three LSTMs share the same parameters but differ in using the style specific matrix such as factual or romantic or humorous style. Based on the LSTMs, the corresponding captions are generated.

3.2.5 Dense caption generation

In image captioning, generating captions with dense descriptions have been growing in recent times. Yang et al.[21] use a faster R-CNN in the first stage through which regions are extracted. In the second stage, using the features of the region, detection score and bounding boxes of the regions are generated and fed into LSTM to generate captions for each region. The LSTM generates one word at a time and uses the current word prediction to generate the next word.

Similar to [21], Krause et al.[22] generates dense captions but by using a hierarchical approach. As in the previous approach, regions are detected in images using CNN but after this, a region proposal network is also used which detects regions of interest. These region features are projected to a pooled vector where the image is converted to a compact vector and fed into a Hierarchical Recurrent Network which comprises a sentence RNN and a word RNN. Sentence RNN is used to find the number of sentences to be generated for an image and word RNN uses these values to generate sentences.

3.2.6 Image Captioning by parallel training

Some image captioning networks use multiple training approaches in different layers of a neural network to generate captions. Venugopalan et al.[20] have proposed a Novel Object Captioner (NOC) which can learn from multiple sources and was intended to detect objects which were not present in the training data. This model was simultaneously trained on a single labeled dataset (ImageNet) and an image captioning dataset (MSCOCO) and used a language model LSTM. Image loss, text loss and image-text loss are computed using these models and captions with low loss values are generated for a given test image.

Liu et al. [23] also train the model using two different data: a single label dataset and an image captioning data in a model using CNN and RNN. There is an interface of semantic concepts in between CNN and RNN which allows for parallel training of the model with the unary model consisting of single label data and a relational model consisting of image captions. Based on the experiments by Liu et al. the intermediate

layer improves the performance of the model compared to other CNN RNN image captioning models as it almost decouples the interactions between CNN and RNN.

CHAPTER 4

Proposal

My project focuses on implementing an image retrieval system using image captioning. Deep learning will be used to implement image captioning. Image captioning algorithm should typically include object detection, feature generation from images and caption generation from features.

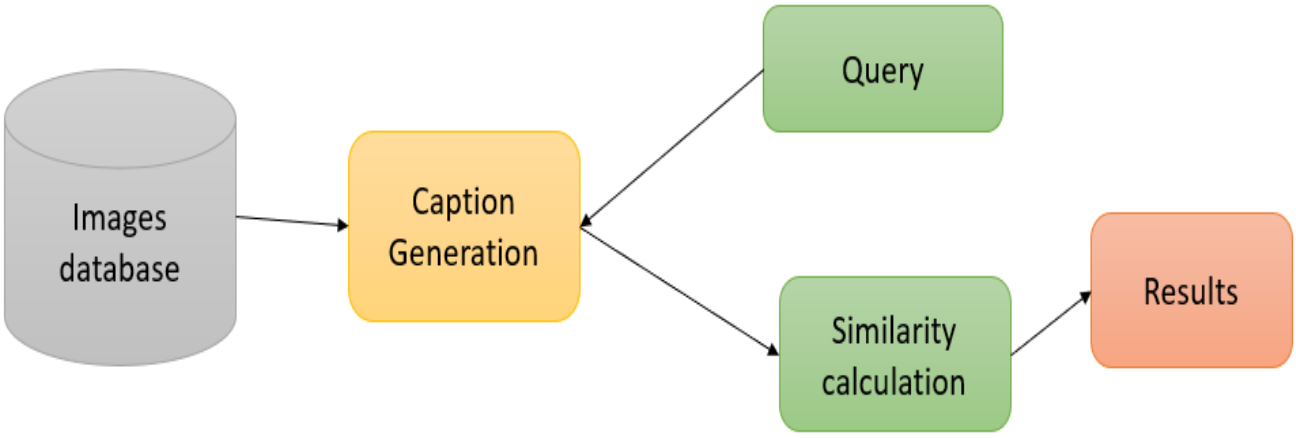

Figure 14: Image Retrieval Using Image Captioning

4.1 Flow of Implementation

This project can be broadly divided into two segments. The first part of the project requires the development of an image captioning system which can generate captions for any image passed into the system. This system can be used to generate captions for all the images present in the dataset and for the image query.

The second part requires a similarity calculation methodology between the captions of the query image and captions of the dataset images. This should return the closest and most similar images for the given query from the dataset.

4.2 Image Captioning System

An image captioning system should have the ability to predict the most probable caption for a given image. In order to achieve this, the system is trained with image captioning datasets. Convolutional Neural Networks (CNN) is used to extract features from images and these features are mapped to the corresponding captions while training. The algorithm is implemented as follows:

1. The objects in an image are detected using CNN.
2. The features of an image are also generated using CNN.
3. Captions are generated using the features generated by CNN. RNN along with LSTM is used to generate captions for the images.
4. CNN pretrained with image captioning datasets like Flickr8k[33] will be used.
5. The above steps are done on both image dataset and visual query to generate captions.
6. Similarity calculation is done on word features generated from the query as well as image dataset.
7. The images containing captions with highest similarity score to the query are returned as results.

By following the above-mentioned steps, images can be retrieved using image captioning.

CHAPTER 5

Data Preparation

This project requires two types of datasets. The first dataset should be able to train the image captioning model and the second dataset is used for image retrieval. Flickr8k dataset [33] is used for training the image captioning model and Wang's database [34] is used for image retrieval.

5.1 Flickr8k Dataset

Flickr8k dataset [33] is a public benchmark dataset for image to sentence description. This dataset consists of 8000 images with five captions for each image. These images are extracted from diverse groups in Flickr website. Each caption provides a clear description of entities and events present in the image. The dataset depicts a variety of events and scenarios and doesn't include images containing well-known people and places which makes the dataset more generic. The dataset has 6000 images in training dataset, 1000 images in development dataset and 1000 images in test dataset.

Features of the dataset making it suitable for this project are:

- Multiple captions mapped for a single image makes the model generic and avoids overfitting of the model.
- Diverse category of training images can make the image captioning model to work for multiple categories of images and hence can make the model more robust.

A blonde horse and a blonde girl in a black sweatshirt are staring at a fire in a barrel .
A girl and her horse stand by a fire .
A girl holding a horse 's lead behind a fire .
A man , and girl and two horses are near a contained fire .
Two people and two horses watching a fire .

Figure 15: Sample Flickr8k image and its captions [33]

5.2 Wang's Database

Wang's database [34] contains 1000 images belonging to 10 classes. Each class has 100 images. This database is used to test and evaluate the image retrieval system of this project which thereby can enable the comparison of the results of other image retrieval systems.

The classes in Wang's database are:

- African people
- Elephants

- Beaches
- Mountains
- Horses
- Food
- Flowers
- Buses
- Buildings
- Dinosaurs

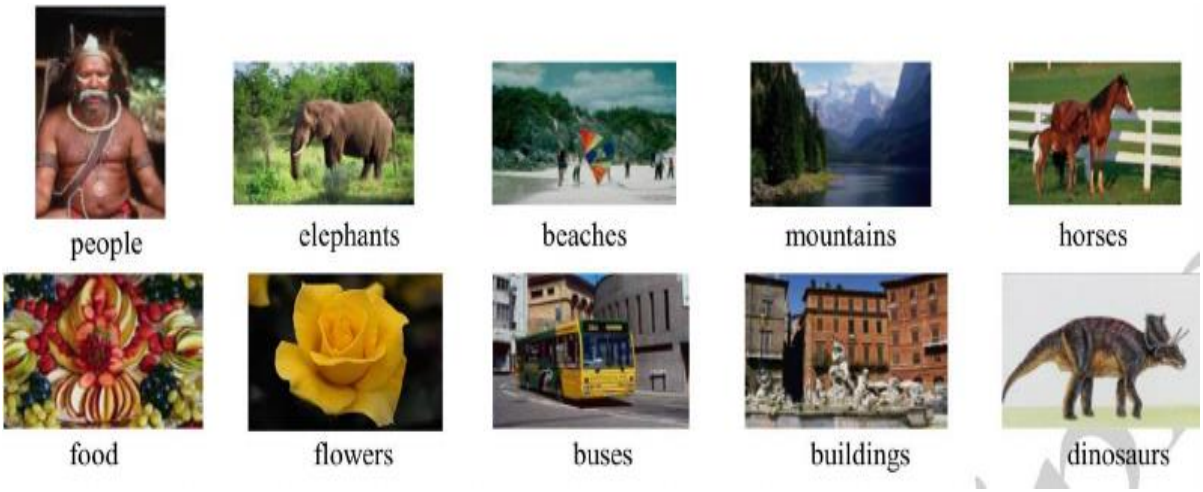

Figure 16: Sample images in Wang's database [35]

5.3 Image Data Preparation

The image should be converted to suitable features so that they can be trained into a deep learning model. Feature extraction is a mandatory step to train any image in deep learning model. The features are extracted using Convolutional Neural Network (CNN) with Visual Geometry Group (VGG-16) model. This model also won ImageNet Large Scale Visual Recognition Challenge in 2015 to classify the images into one among the 1000 classes given in the challenge. Hence, this model is ideal to use for this project as image captioning requires identification of images.

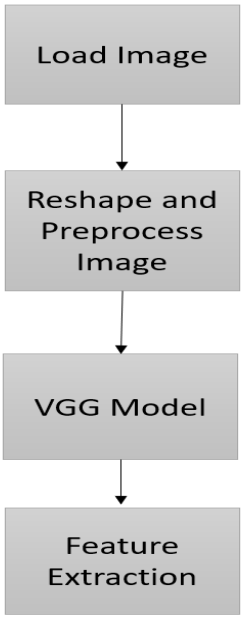

Figure 17: Feature Extraction in images using VGG

In VGG-16, there are 16 weight layers in the network and the deeper number of layers help in better feature extraction from images. The VGG-16 network uses 3*3 convolutional layers making its architecture simple and uses max pooling layer in between to reduce volume size of the image. The last layer of the image which predicts the

classification is removed and the internal representation of image just before classification is returned as feature. The dimension of the input image should be 224*224 and this model extracts features of the image and returns a 1-dimensional 4096 element vector.

5.4 Caption Data Preparation

Flickr8k dataset contains multiple descriptions described for a single image. In the data preparation phase, each image id is taken as key and its corresponding captions are stored as values in a dictionary.

5.4.1 Data cleaning

In order to make the text dataset work in machine learning or deep learning models, raw text should be converted to a usable format. The following text cleaning steps are done before using it for the project:

- Removal of punctuations.
- Removal of numbers.
- Removal of single length words.
- Conversion of uppercase to lowercase characters.

Stop words are not removed from the text data as it will hinder the generation of a grammatically complete caption which is needed for this project. Table 1 shows samples of captions after data cleaning.

Original Captions	Captions after Data cleaning
Two people are at the edge of a lake, facing the water and the city skyline.	two people are at the edge of lake facing the water and the city skyline
A little girl rides in a child 's swing.	little girl rides in child swing
Two boys posing in blue shirts and khaki shorts.	two boys posing in blue shirts and khaki shorts

Table 1: Data cleaning of captions

CHAPTER 6

Image Captioning Model

Deep learning model is defined to generate captions for the images. This model should fit on the training data set. The preprocessed Flickr8k dataset including images and captions are loaded into the model.

6.1 Loading data

The images and captions present in the training data are loaded into the model and the performance of the model is evaluated based on the results of the development dataset. Flickr8k dataset contains identifiers of the images of training data and development data separately as Flickr8k.trainImages.txt and Flickr8k.devImages.txt respectively. These files are used to identify the training data and development data respectively and helps in fetching the relevant images and captions from the dataset. Along with the dataset, features of the image are also loaded into the model.

6.1.1 Encoding text data

The text data should be converted to numbers before passing into the model and this process is called encoding. This is done in all machine learning problems as numerical features are required for the mathematical computations performed on the model. The words in the text caption are split and each word is converted to a number and then fed into the model.

Each word in the caption is then passed to the model one by one along with the corresponding image. Initially, the image is passed into the model along with the first word and it is mapped to generate the corresponding second word. Then the first two words

are passed and mapped to generate the third word of the caption. This process is repeated until the last word and for all the captions.

Input Sequence	Next word
firstword	brown
firstword, brown	dog
firstword, brown, dog	plays
firstword, brown, dog, plays	with
firstword, brown, dog, plays, with	the
firstword, brown, dog, plays, with, the	hose
firstword, brown, dog, plays, with, the, hose	lastword

Figure 18: Training phase of image and caption data

The captions in the dataset are converted into input-output pairs of data for training the model as shown in Figure 18. Two input arrays are passed into the model: one for

passing features of image and the other one for passing text data in encoded format. The output of the model is the encoded next word of the sequence.

Later, when the model is used to generate descriptions for an image, the words generated in the previous sequence are concatenated and fed into the model to generate the next word. The word “firstword” signals the start of the caption and “lastword” signals the end of the caption.

6.1.2 Word prediction

The encoded words in numbers are passed into a word embedding layer in the model. This layer helps in clustering words with similar meanings together in the text data’s vocabulary. The features of the image are fed into the model which enables predicting the output and returns a probability value of 0 or 1. The actual next word in the caption is set to a probability of 1 and the remaining words present in the vocabulary are set to a probability of 0 corresponding to the given image.

CHAPTER 7

Model Definition

7.1 Model Architecture

A merge-model architecture defined in [36] is used in this project to create an image caption generator. In this model, the encoded features of an image are used along with the encoded text data to generate the next word in the caption. In this approach, RNN is used only to encode text data and is not dependent on the features of the image. After the captions have been encoded, those features are then merged with the image vector in another multimodal layer which comes after the RNN encoding layer. This architecture model provides the advantage of feeding preprocessed text data to the model instead of raw data.

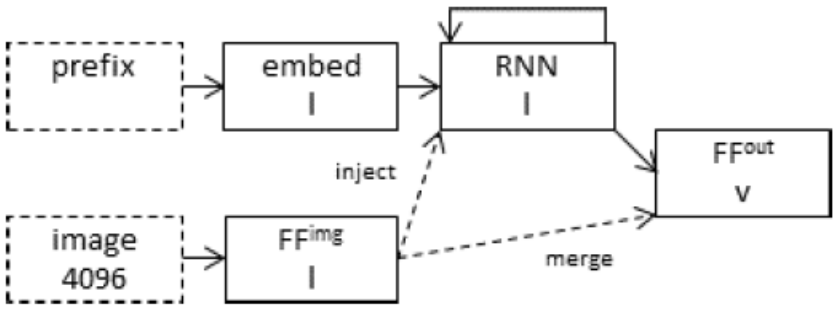

Figure 19: Merge architecture for Image Captioning [36]

The model has three major blocks:

- Image feature extractor
- Text processor
- Output predictor

7.1.1 Image Feature Extractor

The feature extractor needs an image vector of 4096 elements. The model uses VGG-16 pretrained on ImageNet dataset where the features of the image are extracted just before the last layer of classification. Another dense layer is added and converted to get a vector of length 256. Regularization is incorporated by using 50% dropout rate to avoid overfitting.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 34)	0	
input_1 (InputLayer)	(None, 4096)	0	
embedding_1 (Embedding)	(None, 34, 256)	1940224	input_2[0][0]
dropout_1 (Dropout)	(None, 4096)	0	input_1[0][0]
dropout_2 (Dropout)	(None, 34, 256)	0	embedding_1[0][0]
dense_1 (Dense)	(None, 256)	1048832	dropout_1[0][0]
lstm_1 (LSTM)	(None, 256)	525312	dropout_2[0][0]
add_1 (Add)	(None, 256)	0	dense_1[0][0] lstm_1[0][0]
dense_2 (Dense)	(None, 256)	65792	add_1[0][0]
dense_3 (Dense)	(None, 7579)	1947803	dense_2[0][0]

Total params: 5,527,963
Trainable params: 5,527,963
Non-trainable params: 0

Figure 20: Summary of the model

7.1.2 Text Processor

This layer contains the word embedding layer to encode text data. Long Short Term-Memory (LSTM) in RNN is added with 256 memory units. This unit also outputs a vector

of length 256. Similar to image feature extractor, 50% dropout rate is used to avoid overfitting.

7.1.3 Output Predictor

Output vector from both the image feature extractor and the text processor are of same length (256) and a decoder merges both the vectors using an addition operation. This is then fed into two dense layers. The first layer is of length 256 and the second layer makes a prediction of the most probable next word in the caption. This layer uses softmax activation function to predict the most probable next word in the vocabulary.

7.2 Fitting the Model

After building the model, the model is fit using the training dataset. The model is made to run for 20 epochs and the best model is chosen among the 20 epochs by computing loss function on Flickr8k development dataset. The model with the lowest loss function of 3.145 is chosen for generating captions.

CHAPTER 8

Evaluation of Image Captioning Model

The model is evaluated by examining the captions generated for the test dataset. For each photo in the test dataset, captions are generated and compared against the actual captions for the image.

8.1 BLEU Score Evaluation

Bilingual Evaluation Understudy or BLEU score is used to evaluate the descriptions generated from translation and other Natural Language Processing (NLP) applications. The n-grams of the generated captions are compared with the n-grams of the actual reference captions and a score is assigned between 0 to 1. Good scores are higher and close to 1. The image captions generated in the model are evaluated for the test dataset using BLEU score. The model achieved a BLEU-1 score of 0.59 which is close to the BLEU score of human translation of 0.69. BLEU-1 score represents the score of 1-gram and BLEU-n score represents the BLEU score of n-grams. Figure 21 shows the BLEU score obtained for a sample image in Flickr8k dataset using this model. Figure 22 shows the average BLEU score obtained for the images in Flickr8k test dataset.

Original captions
boat sail past the rise sun
sailboat in the water
sailboat in the water at sunrise
ship sail in blue sea
old ship sail at sunset

Generated Caption	sailboat in blue sea

BLEU-1	1.0
BLEU-2	1.0
BLEU-3	0.8
BLEU-4	0

Figure 21: Sample BLEU score results

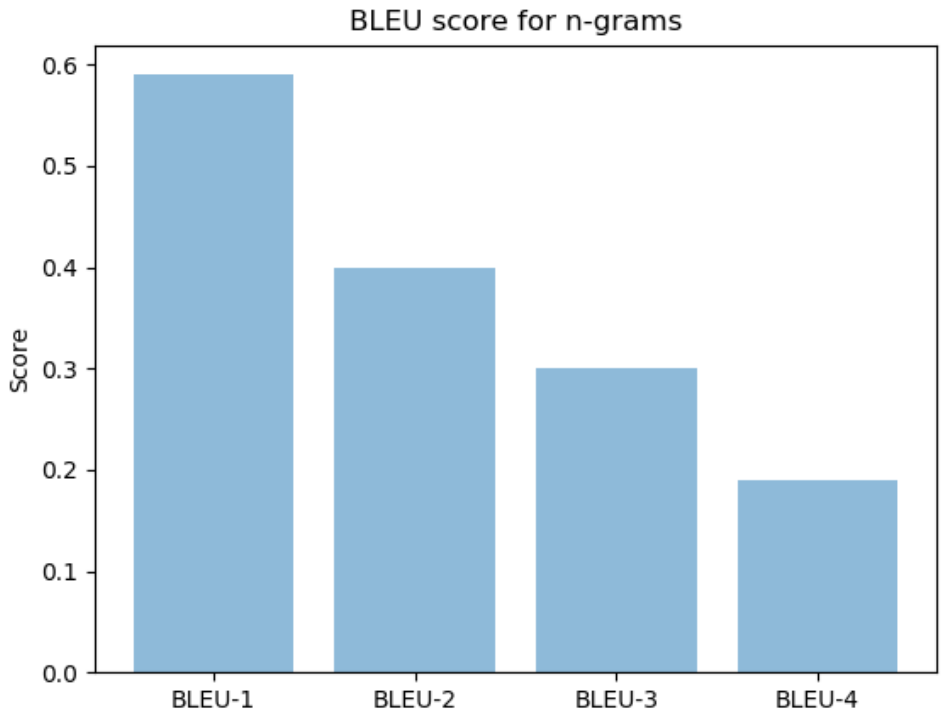

Figure 22: BLEU score of the model

CHAPTER 9

Image Retrieval Using Image Captioning

Image retrieval is the process of retrieving images from a large database of digital images. Wang's database is used to test and evaluate image retrieval results.

9.1 Caption Generation

The Wang's database is split into training and test dataset in 90% and 10% ratio. The first step in image retrieval in this project is to generate captions for all the images in the training dataset. Each image in the training data is passed into the image captioning model and captions are generated for each image present in the training data. These captions are stored by mapping it to the corresponding image identifier.

9.2 Image Retrieval

An image present in the test data of Wang's database is passed as search query. The model should return the images which are present in the same class of the test image from the database. Captions are generated for the query image. The relevant images related to the query are chosen by calculating text similarity between the captions of the query image and captions of the test dataset. The images with captions having highest text similarity with the query image are returned as results using machine learning algorithms like k-Nearest Neighbors or k-NN algorithm.

9.2.1 Text Similarity Calculation

The captions of the images in the database are converted to features so that they can be run on machine learning algorithms. The captions are vectorized using tf-idf

vectorization which is called as term frequency-inverse document frequency. This is a statistic or measurement which can set the most important word of a document to the highest value and the least important word to the lowest value.

- Term frequency is the frequency of a word in a document.
- Inverse document frequency denotes the significance of a word in a document. A word present in multiple documents has a lower value than a word present in fewer documents in the dataset.
- Tf-idf is the product of term frequency and inverse document frequency.
- Tf-idf is calculated by the formula

  Tf-idf = f * log(N/n)

  Where f is the frequency of a word in the document,

  N is the total number of documents in the corpus and

  n is the number of documents containing the word.

Once the text features are vectorized, similarity among the data is calculated using k-NN algorithm.

- The value k is decided for the algorithm in the beginning.
- For each value in the vectorized training data, distance between the query image and the training data image is calculated.
- Distance is calculated by using distance metric Euclidean distance.
- The training data values are sorted based on distance value in ascending order.
- The vectors of top k values are chosen and returned as results with the corresponding images.

Figure 23: Results for the query image at the top belonging to horse class in Wang's database

Figure 24: Results for query image at top belonging to bus class in Wang's database

CHAPTER 10

Evaluation of Image Retrieval

10.1 Precision

Precision is a statistical measure used to measure the fraction of number of relevant images retrieved in the query. Precision is measured by the formula

$$P(k) = n/k$$

where k is the number of retrieved images and n is the number of images belonging to the same class of the query image.

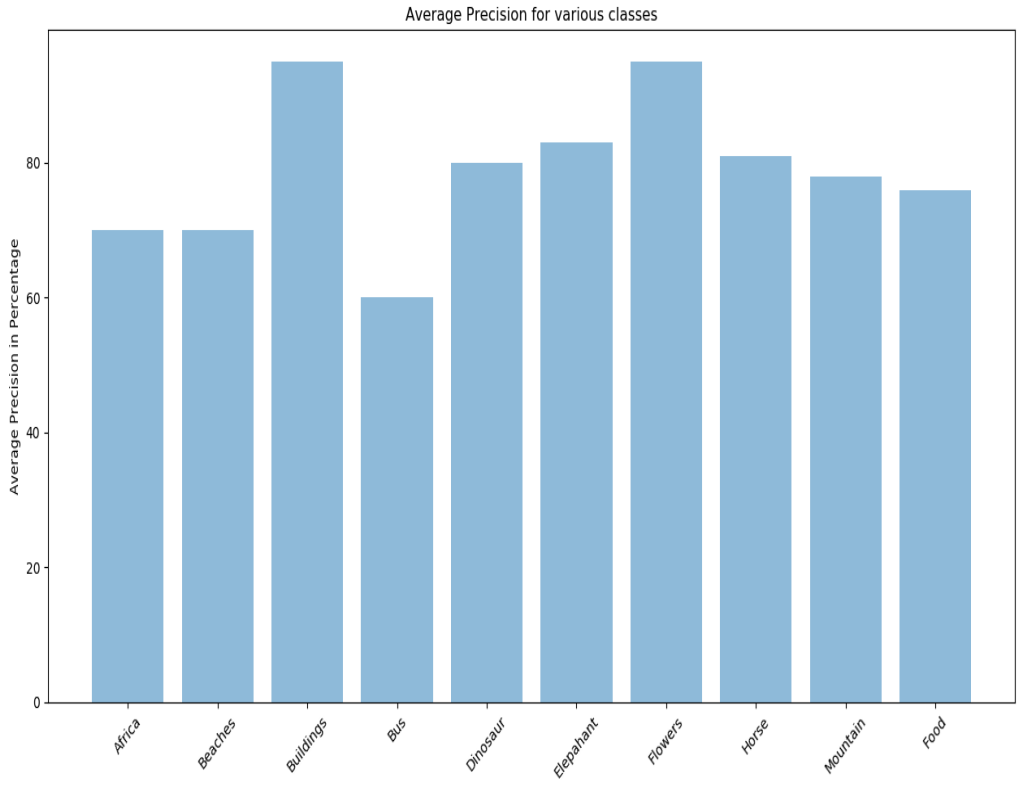

Figure 25: Average Precision for each class

Category	Previous work		Our new Proposed method
	Color Histogram + Gabor transform [38]	Color + Texture [37]	
Africa	**80**	73	70
Beaches	50	37	**70**
Building	47	56	**95**
Bus	52	**87**	60
Dinosaur	**100**	97	80
Elephants	62	67	**83**
Flowers	80	87	**95**
Horse	**91**	85	81
Mountain	28	33	**78**
Food	63	66	**76**
Average	**65**	**69**	**78**

Table 2: Average precision for k=10 of Retrieved images

Figure 25 shows the average precision obtained for all the classes when evaluating the image retrieval results. The average precision of all the classes is 78% and the average precision for each class varies from 60% to 95%. The highest average precision of 95% is obtained for classes buildings and flowers. The above results were obtained for

k=10. This method has a precision higher than the previous experiments in [37] and [38] of 68.78% and 64.76% respectively for image retrieval in Wang's database for k=10.

Table 2 compares the precision results of the images retrieved for k = 10 for experiments done in [37] and [38]. The table also shows the average precision of each class present in Wang's database. When analyzing the results, the proposed method of image retrieval using image captioning shows higher results compared to the previous experiments.

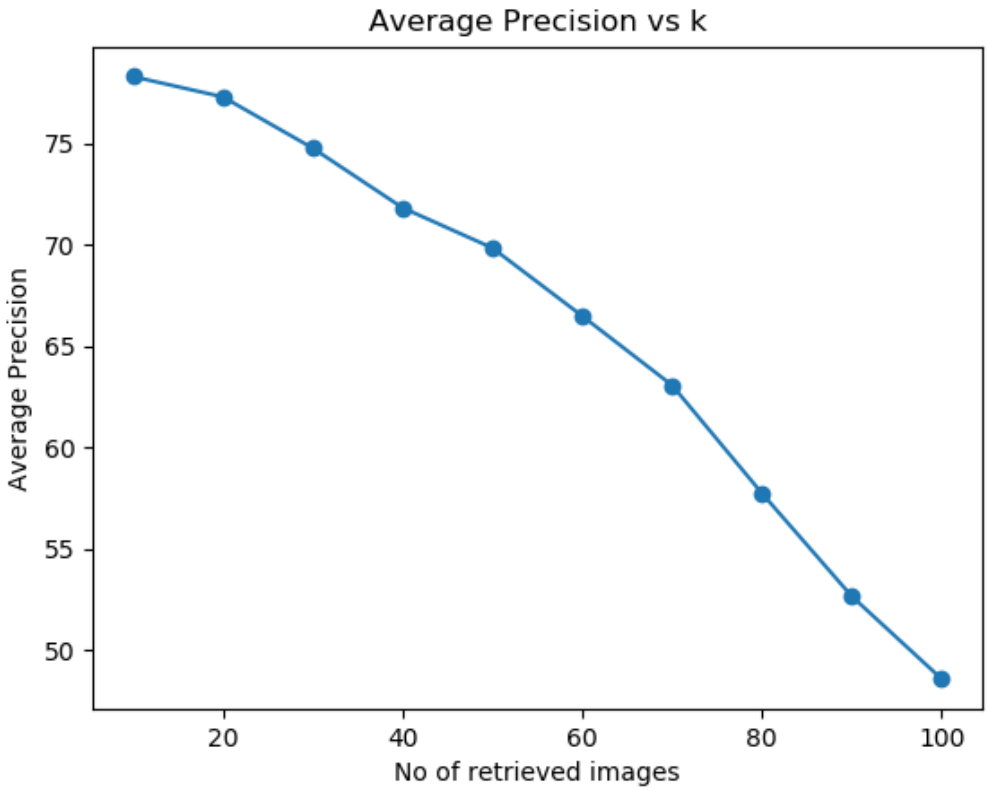

Figure 26: Average precision for values of k from 1 to 100

Figure 26 shows the average precision values of image retrieval results from k = 10 to k = 100. There is a steady decline in precision with the number of images retrieved which is common in all image retrieval algorithms.

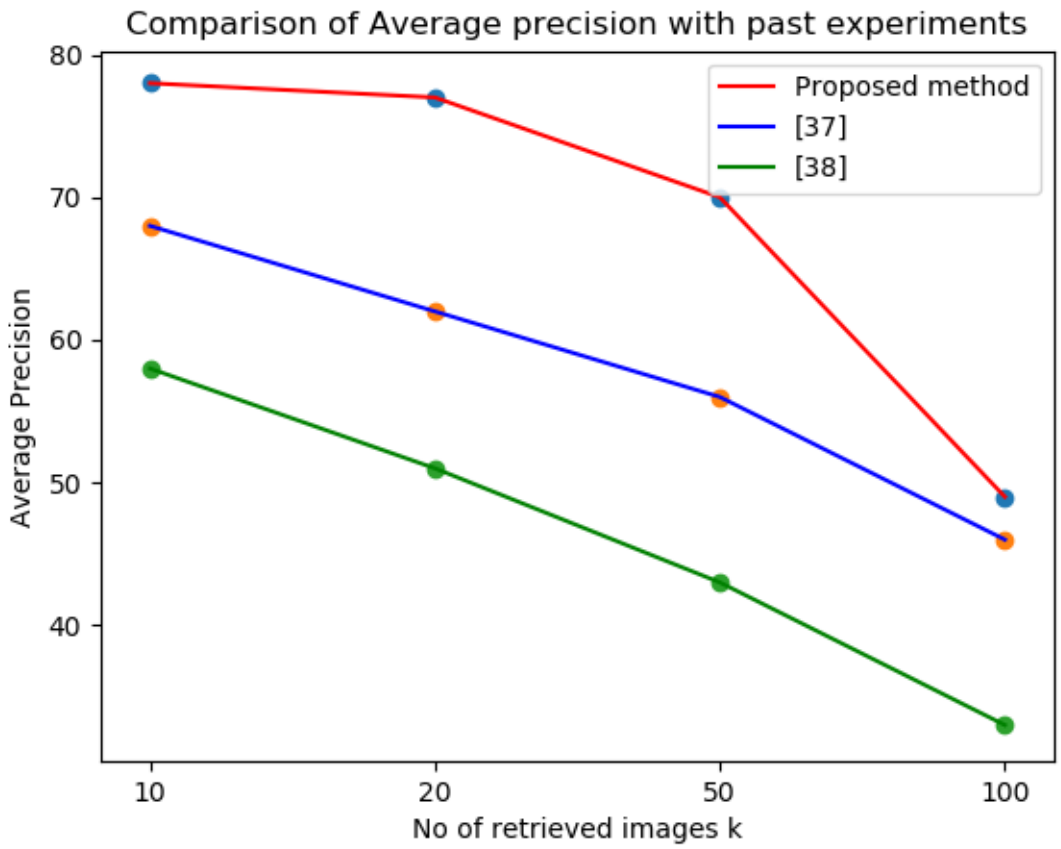

Figure 27: Comparison of average precision of various experiments

Figure 27 compares the average precision of the proposed method with the experiments in [37] and [38]. Average precision by this method is clearly high compared to the methods in [37] and [38] and the precision of the proposed method remains high even for increased values of k.

10.2 Recall

Recall is the fraction of number of relevant images retrieved in a query to the number of relevant images in the database. In this experiment, recall is calculated by the formula,

$$R(k) = n/k$$

where n is the number of relevant images retrieved,

k is the number of relevant images in the database

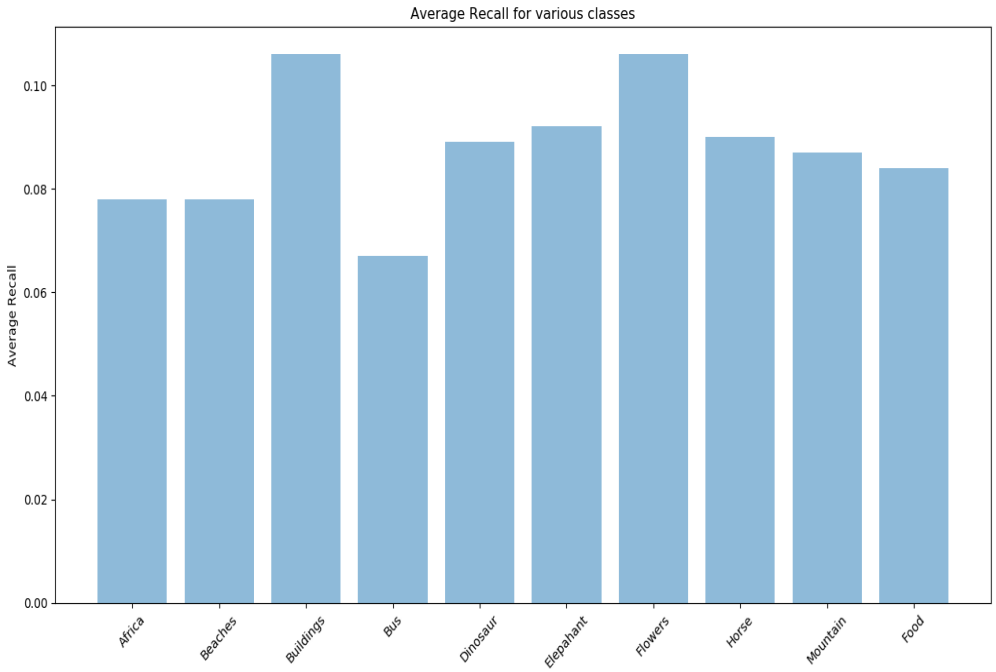

Figure 28: Average Recall for each class when n=10

Figure 28 shows the average recall of each class in Wang's database when the number of images retrieved is 10. In this experiment, value of N is 90 as 90% of the images were used in training data and hence in each class, 90 images should be present in training data. The maximum possible value of recall for any class in this experiment is

0.11 as even if all the images retrieved are correct for n= 10 and N= 90, the corresponding recall will be 0.11. Two classes have recall more than 0.1 and even the lowest recall value among all the classes is greater than 0.06. These results show that the model performs very well in almost all the classes.

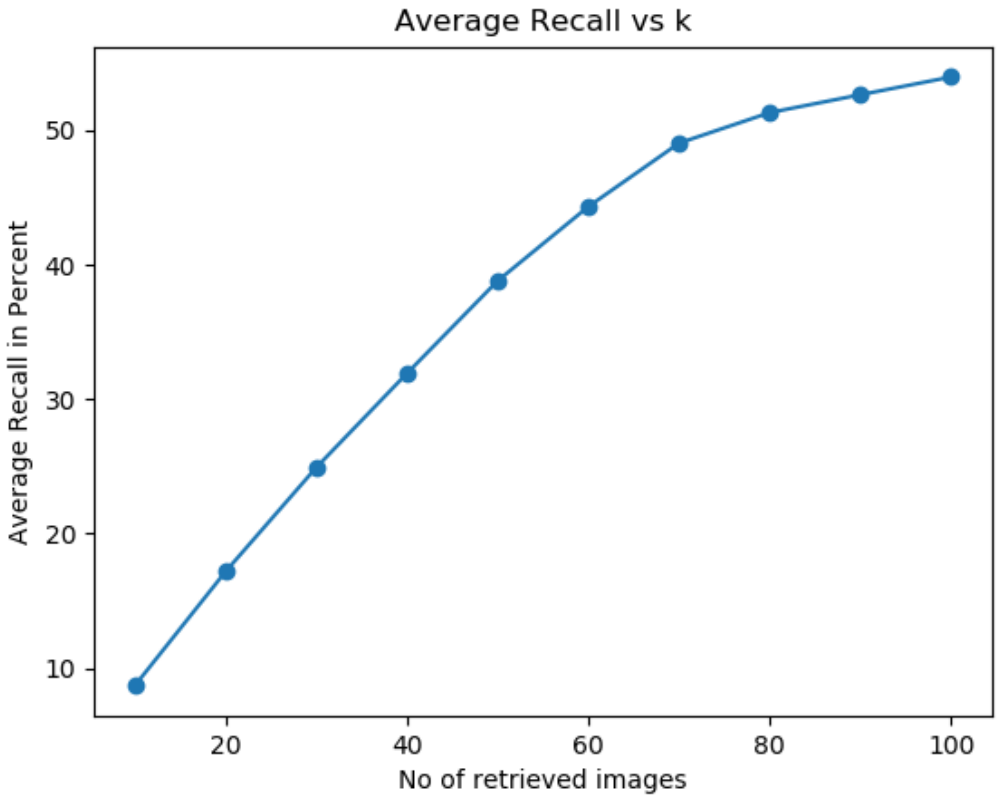

Figure 29: Average recall for values of k from 10 to 100

The recall value increases as the number of retrieved images increases due to increase in numerator value. The maximum possible recall value for k=10 is 11% and the maximum possible recall value for k=20 is 22%. Similarly, the possible recall values

increase as value of k increases. Figure 29 shows the average recall in percentage for different number of retrieved images from k=10 to k=100 for the given model.

Figure 30 compares precision and its corresponding recall values. Generally, a precision recall graph with curve above the diagonal of the graph shows the performance of the model to be good. In Figure 30, the curve lies well above the main diagonal and hence this implies that the model performance is good in terms of precision and recall.

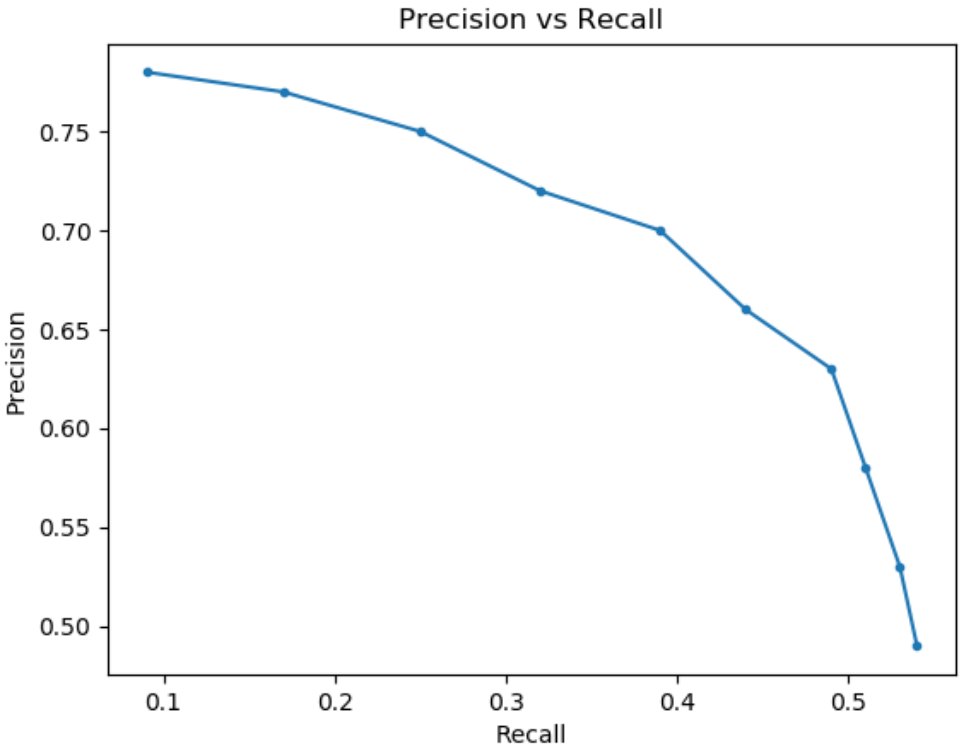

Figure 30: Precision vs Recall graph

10.3 F1 score

F1 score is a statistical measure which evaluates the performance of a model combining both precision and accuracy. F1 score is calculated by the formula

F1 = 2 * Precision * Recall / Precision + Recall

Figure 31 shows the corresponding F1 score for increasing value of k from 10 to 100. For k=10, the maximum possible value of F1 score is 0.2 in this experiment and for k=90, the maximum possible value of F1 score is 1. The actual F1 score obtained is 0.16 for k =10 and 0.52 for k=90. These values imply that the performance of the model is good by evaluating using F1 scores. As found in all image retrieval experiments, the F1 score of the model decreases as the value of k increases.

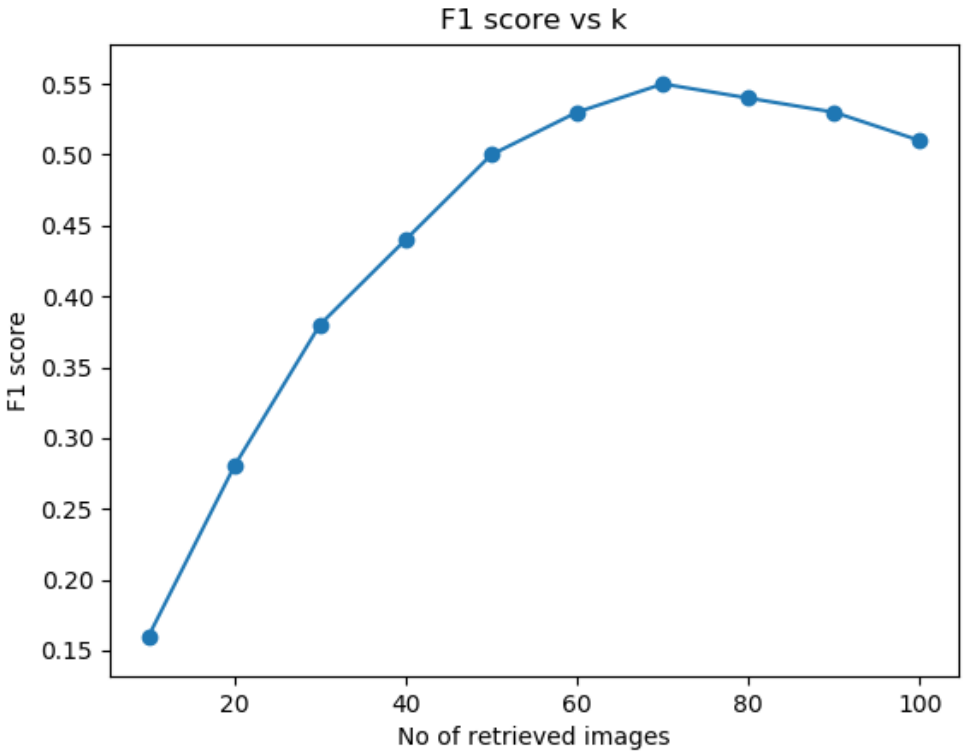

Figure 31: F1 score vs k

Chapter 11

Conclusion and Future Work

11.1 Conclusion

Image retrieval using image captioning has a higher average precision than other methodologies used in [37] and [38]. The proposed method clearly shows a higher precision in identifying classes such as beach, buildings, elephants, flowers, mountains, and food compared to the other methods. Classes like Africa and Dinosaur seem to be better identified in other methodologies. This might be attributed to lack of images belonging to similar category in Flickr8k dataset. The lowest precision value in identifying a class bus is 60% which is higher than the lowest precision of [37] and [38] of 33% and 28% respectively. The precision-recall graph also shows the curve to be above the diagonal of the graph which clearly indicates the model is good. Based on these parameters, this model performs clearly better in image retrieval based on precision compared to the previous experiments.

11.2 Future work

Image retrieval has become an important problem in recent days due to the exponential growth of images in social media and the internet. This report discusses the various research in image retrieval used in the past and it also highlights the various techniques and methodology used in the research.

As feature extraction and similarity calculation in images are challenging in this domain, there is a tremendous scope of possible research in the future. Current image retrieval systems use similarity calculation by making use of features such as color, tags,

histogram, etc. There cannot be completely accurate results as these methodologies do not depend on the context of the image. Hence, a complete research in image retrieval making use of context of the images such as image captioning will facilitate to solve this problem in the future.

This project can be further enhanced in future to improve the identification of classes which has a lower precision by training it with more image captioning datasets. This methodology can also be combined with previous image retrieval methods such as histogram, shapes, etc. and can be checked if the image retrieval results get better.

REFERENCES

[1] X. Lu, Y. Chen and X. Li, "Hierarchical Recurrent Neural Hashing for Image Retrieval With Hierarchical Convolutional Features," in *IEEE Trans. on Image Process.*, vol. 27, no. 1, pp. 106-120, Jan. 2018.

[2] L. Jin, K. Li, H. Hu, G. Qi and J. Tang, "Semantic Neighbor Graph Hashing for Multimodal Retrieval," in *IEEE Trans. on Image Process.*, vol. 27, no. 3, pp. 1405-1417, March 2018.

[3] C. Huang, S. Yang, Y. Pan and H. Lai, "Object-Location-Aware Hashing for Multi-Label Image Retrieval via Automatic Mask Learning," in *IEEE Trans. on Image Process.*, vol. 27, no. 9, pp. 4490-4502, Sept. 2018.

[4] S. Sreedevi and S. Sebastian, "Content based image retrieval based on Database revision," *2012 Intl. Conf. Mach. Vis. Image Proc. (MVIP)*, Taipei, 2012, pp. 29-32.

[5] E. R. Vimina and K. Poulose Jacob, "Image retrieval using colour and texture features of Regions Of Interest," *2012 Intl. Conf. Inform. Retrieval Knowl. Manag.*, Kuala Lumpur, 2012, pp. 240-243.

[6] Zhou Bing and Yang Xin-xin, "A content-based parallel image retrieval system," *2010 Intl. Conf. Comp. Design. App.*, Qinhuangdao, 2010, pp. V1-332-V1-336.

[7] J. Wang *et al*., "MindCamera: Interactive Sketch-Based Image Retrieval and Synthesis," in *IEEE Access*, vol. 6, pp. 3765-3773, 2018.

[8] D. Xu, X. Alameda-Pineda, J. Song, E. Ricci and N. Sebe, "Cross-Paced Representation Learning With Partial Curricula for Sketch-Based Image Retrieval," in *IEEE Trans. on Image Process.*, vol. 27, no. 9, pp. 4410-4421, Sept. 2018.

[9] J. Yang, J. Liang, H. Shen, K. Wang, P. L. Rosin and M. Yang, "Dynamic Match Kernel With Deep Convolutional Features for Image Retrieval," in *IEEE Trans. on Image Process.*, vol. 27, no. 11, pp. 5288-5302, Nov. 2018.

[10] K. Song, F. Li, F. Long, J. Wang and Q. Ling, "Discriminative Deep Feature Learning for Semantic-Based Image Retrieval," in *IEEE Access*, vol. 6, pp. 44268-44280, 2018.

[11] A. Raza, H. Dawood, H. Dawood, S. Shabbir, R. Mehboob and A. Banjar, "Correlated Primary Visual Texton Histogram Features for Content Base Image Retrieval," in *IEEE Access*, vol. 6, pp. 46595-46616, 2018.

[12] Y. Wang, L. Zhu, X. Qian and J. Han, "Joint Hypergraph Learning for Tag-Based Image Retrieval," in *IEEE Trans. on Image Process.*, vol. 27, no. 9, pp. 4437-4451, Sept. 2018.

[13] Z. Gao, L. Wang and L. Zhou, "A Probabilistic Approach to Cross-Region Matching-Based Image Retrieval," in *IEEE Trans. on Image Process.*, vol. 28, no. 3, pp. 1191-1204, March 2019.

[14] K. Fu, J. Jin, R. Cui, F. Sha and C. Zhang, "Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2321-2334, 1 Dec. 2017.

[15] Q. Wu, C. Shen, P. Wang, A. Dick and A. v. d. Hengel, "Image Captioning and Visual Question Answering Based on Attributes and External Knowledge," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1367-1381, 1 June 2018.

[16] Bo Dai and Dahua Lin, “Contrastive learning for image captioning,” in *Advances in Neural Information Processing Systems 30*, pages 898–907. Curran Associates,Inc., 2017.

[17] C. Gan, Z. Gan, X. He, J. Gao and L. Deng, "StyleNet: Generating Attractive Visual Captions with Styles," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 955-964.

[18] R. Vedantam, S. Bengio, K. Murphy, D. Parikh and G. Chechik, "Context-Aware Captions from Context-Agnostic Supervision," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 1070-1079.

[19] Z. Gan *et al*., "Semantic Compositional Networks for Visual Captioning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 1141-1150.

[20] S. Venugopalan, L. A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell and K. Saenko, "Captioning Images with Diverse Objects," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 1170-1178.

[21] L. Yang, K. Tang, J. Yang and L. Li, "Dense Captioning with Joint Inference and Visual Context," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 1978-1987.

[22] J. Krause, J. Johnson, R. Krishna and L. Fei-Fei, "A Hierarchical Approach for Generating Descriptive Image Paragraphs," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 3337-3345.

[23] F. Liu, T. Xiang, T. M. Hospedales, W. Yang and C. Sun, "Semantic Regularisation for Recurrent Image Annotation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 4160-4168.

[24] Y. Wang, Z. Lin, X. Shen, S. Cohen and G. W. Cottrell, "Skeleton Key: Image Captioning by Skeleton-Attribute Decomposition," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 7378-7387.

[25] X. Wei, Y. Qi, J. Liu and F. Liu, "Image retrieval by dense caption reasoning," *2017 IEEE Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL, 2017, pp. 1-4.

[26] “OpenCV - Colored Images to GrayScale” [Online]. Available: https://www.tutorialspoint.com/opencv/opencv_colored_images_to_grayscale.htm.

[27] L. Hulstaert, “A Beginner's Guide to Object Detection”. 2018 [Online]. Available: https://www.datacamp.com/community/tutorials/object-detection-guide.

[28] Wikipedia contributors, “Artificial neural network”. 2018 [Online]. Available: https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=871961422.

[29] D. Gilleman, “Convolutional Network (CNN)”. [Online]. Available: http://www.deeplearningessentials.science/convolutionalNetwork/.

[30] R. Prabhu, “Understanding of Convolutional Neural Network (CNN)— Deep Learning”. [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[31] N. Donges, “Recurrent Neural Networks and LSTM”. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5

[32] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, “Sketch-based shape retrieval”. *ACM Trans. Graph. (Proc. SIG-GRAPH),* 31(4):31:1–31:10, 2012.

[33] M. Hodosh, P. Young, J. Hockenmaier, "Framing image description as a ranking task: Data models and evaluation metrics". *Journal of Artificial Intelligence Research*, pp. 853-899, 2013.

[34] J. Z. Wang, "Wang's Image Database". [Online]. Available: http://wang.ist.psu.edu/.

[35] H. Du, “Effectiveness of Image Features and Similarity Measures in Cluster-based Approaches for Content-based Image Retrieval - Scientific Figure on ResearchGate”. [Online]. Available: https://www.researchgate.net/figure/Some-example-images-from-WANG-database_fig2_264003334.

[36] M. Tanti, A. Gatt, and K. Camilleri, “Where to put the image in an image caption generator,” *Natural Language Engineering*, vol. 24, no. 3, pp. 467–489, 2018.

[37] E. R. Vimina and K. Poulose Jacob, "Image retrieval using colour and texture features of Regions of Interest," *2012 International Conference on Information Retrieval & Knowledge Management*, Kuala Lumpur, 2012, pp. 240-243.

[38] S. Murala, A. B. Gonde and R. P. Maheshwari, "Color and Texture Features for Image Indexing and Retrieval," *2009 IEEE International Advance Computing Conference*, Patiala, 2009, pp. 1411-1416.