Automatic Image Captioning Using Deep Learning

Submitted in partial fulfilment of the requirements

For the degree of

Bachelor of Engineering

Synopsis Report

By

DHRUV SOLANKI

ROLL NO: -50

TEJAL SUPE

ROLL NO: -51

SIDDHI UNDIRKAR

ROLL NO: -55

Under the Supervision of

Prof. J.P.Patil

DEPARTMENT OF INFORMATION TECHNOLOGY
KONKAN GYANPEETH COLLEGE OF ENGINEERING,
KARJAT-410201
November 2020

Certificate

This is to certify that the project entitled Automatic Image Captioning Using Deep Learning is a bonafide work of DHRUV SOLANKI (Roll No.50), TEJAL SUPE (Roll No.51), and SIDDHI UNDIRKAR (ROLL No.55) submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of Undergraduate in DEPARTMENT OF INFORMATION TECHNOLOGY.

Supervisor/Guide
Prof. J.P. Patil
Department of Information Technology

Head of Department
Prof. J.P.Patil
(Department of Information Technology)

Principal
Dr. M.J. Lengare
(Konkan Gyanpeeth College of Engineering)

Project Report Approval

This project report Automatic Image Captioning Using Deep Learning by DHRUV SOLANKI (Roll No.50), TEJAL SUPE (Roll No.51), SIDDHI UNDIRKAR (ROLL No.55) is approved for the degree of DEPARTMENT OF INFORMATION TECHNOLOGY.

Examiners

1......................................

2.....................................

Date: -

Place: -

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data /fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature
DHRUV SOLANKI
(Roll No.50)

Signature
TEJAL SUPE
(Roll No.51)

Signature
SIDDHI UNDIRKAR
(Roll No.55)

Date:

Abstract

Image captioning means automatically generating a caption for an image. Automatically creating the description of an image using any natural language sentences is a very challenging task. It requires expertise of both image processing as well as natural language processing. Sharing photos through the internet (e.g. Instagram, Facebook, etc.), which becomes a common practice, leads to archives in the order of millions of images. Manual annotation is time consuming and extremely labour-intensive work. Thereafter, computer-assisted system are desired to lesson these difficulties by automation using machine learning. The annoted image can be used for retrieval purposes, in image processing algorithms, intelligent scanner, digital cameras, photocopiers, and printers. Our goal is to create system which will annotate images based on previously learned datasets.

Acknowledgement

Success is nourished under the combination of perfect guidance, care blessing. Acknowledgement is the best way to convey. We express deep sense of gratitude brightness to the outstanding permutations associated with success. Last few years spend in this estimated institution has molded us into condent and aspiring Engineers. We express our sense of gratitude towards our project guide Prof. J.P. Patil. It is because of his valuable guidance, analytical approach and encouragement that we could learn and work on the project. We will always cherish the great experience to work under their enthusiastic guidance. We are also grateful to our principle Dr. M.J. Lengare who not only supporting us in our project but has also encouraging for every creative activity. We extend our special thanks to all teaching and non-teaching staff, friends and well-wishers who directly or indirectly contributing for the success of our maiden mission. Finally, how can we forget our parents whose loving support and faith in us remains our prime source of inspiration. Lastly we would like to thank all those who directly and indirectly helping to complete this project. We would also like to acknowledge with much appreciation the crucial role of the staff of Information Technology Department, who gave the permission to use the all required software/hardware and the necessary material to completing to the project.

Signature
DHRUV SOLANKI
(Roll No.50)

Signature
TEJAL SUPE
(Roll No.51)

Signature
SIDDHI UNDIRKAR
(Roll No.55)

Contents

Contents

Abbreviations

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

SVM: Support Vector Machine

LSTM : Long Short Term Memory

Chapter 1
INTRODUCTION

1.1 Introduction

The explosive growth of image data leads to the research and development of Content Based Image Retrieval (CBIR) systems. CBIR systems extract and retrieve an images by their low-level features, such as color texture, and shape. However, these visual contents do not allow users to query images by semantic meanings. Image annotation systems, a solution to solve the inadequacy of CBIR systems, aim at automatically annotating image with some controlled keywords.

Image annotation refers to the tagging of images with appropriate keywords. The process of annotating images manually is time-consuming and extremely labor-intensive work. Thereafter, computer-assisted systems are desired to lessen these difficulties by automation. The automatic image annotation is targeted to assign a few relevant text keywords to an input image that reflects its visual content. Machine learning techniques are used to develop the image annotation systems to map the low-level (visual) features to high-level concepts or semantics.

There are two main approaches to Image Captioning: bottom-up and top-down. Bottom-up approach generate items observed in an image, and then attempt to combine the items identified into a caption. Top-down approach attempt to generate a semantic representation of an image that is then decoded into a caption using various architectures, such as recurrent neural networks. There are various methods and algorithms in deep neural networks through which a caption can be generated by giving an image .We are going to use top-down approach for our project. We are going to use deep convolutional neural network to generate a representation of an image that we will then feed into a Long-Short-Term Memory (LSTM) network, which will then generate caption. We are going to use two neural network algorithm which is Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) based Long Short Term Memory (LSTM). In the project CNN will be used to train the images as well as to detect the objects in the image with the help of various pre trained models and RNN based LSTM will be used to generate captions from the generated object keyword. We are going to use flask interface for detecting an object in the image

1.2 OBJECTIVES:

1. The objective of our project is to extract the features of a given image and to automatically caption the objects present in the image.
2. To build a system that will identify multiple objects from a given image.
3. To create a system that will be able to caption the objects present in the image.
4. To be able to train the model created to identify and caption the images.
5. To be able to test and evaluate the performance of the system of both the training and testing phase.

1.3 PURPOSE, SCOPE AND APPLICATION:

Purpose scope and application. The description of purpose scope and application are given below:

1.3.1 PURPOSE:

The main purpose of image annotations is to highlight or capture the targeted object in a picture to make it recognizable for machines. As large collection of data is required for technologies based on machine learning, it is crucial that the data available is highly precise in terms of correctness and its relevance with real world entities. This kind of data can be obtained by manually tagging images, this task is very tedious as well as time consuming and labor intensive. It is not practical. So, a method is required to carry the same automatically.

1.3.2 SCOPE:

1. The scope of this project is to classify the images from the different classes available which makes it easy to search an image based on a keyword.
2. The system developed in this project is such that it will add a caption to multiple objects present in an image.
3. The images after been identified and captioned will be saved into the separate database.

1.3.3 APPLICABILITY:

Image annotation system have immense areas of applicability, we list some of them that are practical interest to us:

1. Content based image retrieval.
2. Improve data-sets for training.
3. Improve image based search engines.
4. Create problem specific image data-sets.

Chapter 2
LITERATURE SURVEY

2.1 LiteratureSurvey

Literature Survey: Is the process of analysing, summarizing, organizing and presenting novel conclusions from the results of technical review of large number of recently published scholarly articles. In this chapter we survey previous research done on automatic image annotation, we have studied about following papers published by some experts.

1) IMAGE ANNOTATION USIN GSVM

(Author: Claudio Cusano, Gianluigi Ciocca, Raimondo Schettini)

In this paper, author suggested an image annotation tool using SUPPORT VECTOR MACHINE for classifying image regions in one of seven classes - sky, skin, vegetation, snow, water, ground, and buildings or as unknown. The central idea of SVM is to adjust a discriminating function so that it makes optimal use of the separability information of boundary cases. Using SVM for classification of image the results are satisfactory. The accuracy in the classification has increased for each class (more than 50% for the ground class). Some errors are still present, ground tiles in particular have often been misclassified as vegetation tiles (5,4% of cases) and vice-versa (5%), snow tiles have been misclassified as sky tiles (5,4%), and buildings tiles as ground tiles (4,5%)

2)AUTOMATICALLY ANNOTATING IMAGES WITH KEYWORDS: A REVIEW OF IMAGE ANNOTATIONSYSTEM.

(Author: Chih-Fong Tsai, Chihli Hung. (200a))

This paper briefly describes commonly used image segmentation methods includ- ing global and local feature extraction.This paper is organized as follows. Section 2 briefly describes commonly used image segmentation methods including global and local feature extraction. Section 3 covers the low-level features which are genre- ally extracted from images. Section 4 overviews the mostly used supervised learning methods for the task of image annotation. Some representative image annotation systems are also described. Section 5 presents advanced techniques by combining multiple classifiers and hybrid classifiers for image annotation. Section 6 provides a comparison of related image annotation systems in terms of their feature representtation, classifier used, classification scale.

3) LEARNING MODELS:

3.1 Probabilistic Classifiers
3.2 Artificial Neura1Networks
3.3 Support Vector Machines
3.4 Decision Trees
3.5 k-Nearest Neighbour
3.6 Tem- plate Matching

4. ADVANCED LEARNING TECHNIQUES.

All the models are compared against a dataset and results are as follows: to each image, but only 8 systems consider assigning multiple keywords to an image by using either region based or local block-based image features.

14 systems only use color features for high-level concept learning and classification under the problem scale between 2 to 50 categories. The majority, i.e. 19 systems, uses color and texture features for the problem scale between 2 to 125 categories. Only 6 systems consider other features such as shape in addition to color and texture features.

As the problem scale increases, i.e. larger numbers of categories, only Tsai et a1. 34] reports the number of (un) predictable classes since there should be some categories which are difficult to classify.

SVMs and ensemble classifiers have attracted much more attention recently. That is, 18 systems use SVMs and ensemble classifiers; k-NN, and naive Bayes are used for 7 and 8 systems respectively. Only one system uses decision trees and two uses template matching method.

4) DEEP CAUTIONING WITH MULTIMODAL RECURRENT NEURAL NETWORKS(M-RNN)

(Author: Junhua Mao. (2015))

This paper presents a m-RNN model for generating novel image captions, over bench- mark datasets ((IAPR TC-12, Flickr 8K, Flickr 30K and MS COCO). In addition, the m-RNN model is applied to retrieval tasks for retrieving images or sentences. Using this model sentence level description can be obtained. In this work, author propose a multimodal Recurrent Neural Networks (m-RNN) model to address both the task of generating novel sentences descriptions for images, and the task of image and sentence retrieval. The whole m-RNN model contains a language model part, a vision part and a multimodal part. The language model part learns a dense feature embedding for each word in the dictionary and stores the semantic temporal context in recurrent layers. The vision part contains a deep Convolutional Neural Network (CNN) which generates the image representation. Results are shown for three tasks:

1) Generating novel sentences
2) Retrieving images given a sentence and
3) Retrieving sentences given an image. Results on IAPR TC-12 are as follows: The result shows that 20.9% top-ranked retrieved sentences and 13.2% top-ranked retrieved images are ground truth. Result on FLICKR30K and MS COCO are as follows: 71%of the generated sentences for MSCOCO datasets are novel (i.e. different from training sentences).

5) AUTOMATIC IMAGE ANNOTATION USING NEURAL NETWORKS

(Author:Aanchan K Mohan, Mar wan A. Tokri.)

This paper talks about the training process to achieve automatic image annotation using neural networks. This report is organized as follows: Section 2 describes the training process in great detail and the steps followed therein, Section 3 summarizes the results obtained following our proposed approach, and Section 4 concludes this report with a discussion about future work that this project could be extended to. The Training Process, 2.1. The Training Database: The IAPR-TC 12 Benchmark image database was used for this project. This database consists of 20000 natural still inn ages (plus 20000 thumbnails). Each image is associated with a free-flowing text caption describing the image. For this project, about 17558 images were provided as a part of the training set along with image keywords(nouns) extracted from the free-flowing text, and the free-flowing text itself. 2.2 The Proposed Approach the procedure that is followed consists of doing some preprocessing on the image database to extract a dictionary of key words. 2.3 The Preprocessing created a dictionary of keywords based on their frequency of occurrence in the image database. We got rid of words which appeared to be outliers, and kept only those keywords whose frequency of occurrence was greater than 10. 2.4 Feature Extraction: The process of feature extraction involves the extraction of a 6 element vector for each block of the image after having divided the entire image into regions of blocks of size 4x4. The feature extraction for the color information is done in the Luv space. This involves in converting the image from the default RGB color space to the ,Luv color space. 2.5 Region Segmentation: The second step in training approach is to cluster the image into a set of 16 clusters that describe the image. We used the K-means clustering algorithm to perform the task of clustering to limit our output clusters to 16 clusters as author mentioned. 2.6 Artificial Neural Networks: The last step in training approach is to build and train a neural network to learn relationship between image segments and the annotation keywords. Results shows that using this method even in low end computers the process of training takes few minutes. The accuracy rate and the recall rate are both greater than 19%, which is quite high.

6) OBJECTIVE-GUIDED IMAGE ANNOTATION

(Author:Ivor Wai-Hung Tsang, Qi Mao. (2013)).

This paper talks about various methodologies to assign multiple tags for images. The rest of this paper is organized as follows. Related work is brief discussed in Section II. The objective-guided performance measures and its property analysis are given in Section III and Section IV. Section V illustrates the unified multi-label learning framework. Experimental results are shown in Section VI. To address the issue that many image annotation methods neglect optimizing the objective-guided performance measures, in this paper, we attempt to optimize a variety of objective specific measures in a unified multi-label learning framework. We present a multi-layer hierarchical structure of learning hypothesis for multi-label problems based on which a variety of loss functions with respect to objective guided measures are defined. And then, the unified learning framework is presented. Our analysis reveals that macro-averaging measures are very sensitive to infrequent keywords, microaveraging measures are time-consuming, and hamming measure is easily affected by skewed distributions. The experimental results on four image annotation datasets demon state that optimizing the objective-guided performance measure is able to improve this performance measure, especially for F1 score, which consistently shows very competitive results over three measures on all four datasets.

7) AUTOMATIC IMAGE ANNOTATION USING MULTI-OBJECT IDENTIFICATION

(Author:Yin-Fu Huang, Hsin-Yun Lu. (2010)).

In this paper author suggest in order to improve annotation accuracy, the irrele- vant areas such as background can be eliminated from an image before extracting the main features from the image, and then classifiers are built using these extracted features. The system framework consists of two phases: training phase and annotation phase. The training phase could be divided into three stages: 1) main object training, 2) background object training, and 3) object association analysis. First, a main object detection method is proposed to segment the main object from a training image. Then, the feature vectors of the main object are extracted to train its classifier. The classifier could determine what class an image belongs to. Second, we also segment training images and extract the features vectors from the background objects to train background classifiers. Finally, the main task is to perform the association analysis by using a probability model called GMM. The object association analysis is to find the associations between main objects (or image classes) and image background objects, and build the association knowledge base. The purpose is to eliminate the irrelevant model testing in the annotation phase so that annotation accuracy could be improved. In the annotation phase, we segment a test image and extract the feature vectors from a main object. The main object classifier would identify the class of the test image. Then, the system can retrieve the relevant back- grounds of the detected image class from the association knowledge base, and the relevant background classifiers would be used to detect whether these backgrounds appear in the test image. Main and background object training is done next. The following step is object association analysis. The purpose of object association analysis is to speed up the process time in the annotation phase and improve annotation accuracy. Gaussian mixture model: The GMM is an efficient method to precisely describe the sample clustering in the feature space. Through the training, proper parameters could be obtained to fit with target statistics. The final step is Annotation phase. First, we segment and detect the main object out of a test image. Then, the features of the main object are extracted to identify its class, using the built main object models. According to the identified class, we can get all related back- ground models from the association knowledge base. Next, the background objects are discriminated one by one from the remainder of the image. Then, the features of the background objects are extracted to detect the backgrounds, using these related background models. Finally, the annotation of an image, including the class and backgrounds, can be achieved. This results in a very good detection of subject and specific annotation of individual objects. This increases accuracy. Experimental results show that the final annotation of most classes achieves more than 85they also validate that the system would not annotate incorrect backgrounds in an image even if its image class implies these backgrounds in the association knowledge base.

8) AUTOAMTIC MULTI-LABLE IMAGE ANNOTATION FORSMART CITIES

(Author: Gyayak Sanghi, Nalin Kanungo. (2017)).

In this author used ML-KNN algorithm to classify specific objects from dataset. In this case objects that are seen in a city are expected as output.Several images of a landmark are extracted from flicker dataset. Every new image is compared with all the images and its nearest neighbours are determined, then the image is compared to the visually closest image of the landmark. Finally, the image is annotated with a particular label only if it matches above a particular threshold.A kernel method for multilabeled classification, which is basically a slight modification of the KNN approach discussed in a research work. Here the images are categorized according to the keywords and for each keyword the k-nearest neighbours to the input image are found.Finally, a weighted sum is taken over the samples to assign a rank to each label so obtained.A research work proposed use of maximum entropy for automatic image annotation which divides each training image into visterms or rectangular regions having a label associated to it. Then for any test image, the closest neighbours are found foreach visterm and probabilistic approach is used to assign thelabel to the test image. Methodology used is ML-KNN algorithm on own dataset which is composed of 100 training images and 50 testing images all of urban cities. Images are represented by feature vectors with 78 features each. These 78 features are com- posed of colour, edge, texture features. The values of result parameters obtained above are comparable to the value of same parameters obtained when this algorithm is applied on benchmark dataset (corel5kdataset). Using Benchmark dataset there is an improvement in the value of hamming loss and coverage parameters.

9) **AUTOMATED ANNOTATION OF NATURAL IMAGES USINGANEXTENDED ANNOTATION MODEL**

(Author: GABRIEL MIHAI, LIANA STANESCU. (2012)).

The annotation process implemented in this system is based on CMRM (Cross Media Relevance Model).The annotation model is based on object oriented approach. The paper describes the extension of an image annotation model that can be used for annotating natural images. The CMRM annotation model has proved to be very efficient by several studies. This model learns the joint probability of concepts and blobs based on a well know benchmark: SAIAPR TC-12. This benchmark contains a large-size image collection comprising diverse and realistic images, includes an annotation vocabulary having a hiearchical organization, well defined criteria for the objective segmentation and annotation of images. Because the quality of an image region and the running time of the segmentation process are two important factors for the annotation process that have been used, uses a segmentation algorithm based on a hexagonal structure which was proved to satisfy both requirements: a better quality and a smaller running time. Each new image was annotated with concepts taken from an ontology created starting from the information provided by the bench- mark: the hierarchical organization of the vocabulary and the spatial relationships between regions. For storing the information required by the annotation process it used an object oriented database called db4o. The object oriented approach has simplified the way of describing the modified version. The experimental results realized from two perspectives (annotation and retrieval) have proved that the proposed modified model produces better results than the initial model.

Experimental Results:

(1) **Annotation perspective:** In order to evaluate the annotation system author has used a testing set of 400 images that were manually annotated and not

included in the training set used for the CMRM model This set was segmented using the segmentation algorithm described above and a list of concepts having the joint probability greater than a threshold value was assigned to each image. Then the number of relevant concepts automatically assigned by the annotation system was compared against the number of concepts manually assigned by computing an accuracy value for both modules. The average accuracy value obtained for the initial model was0.46and the average accuracy value obtained for the modified model was 0.54.

(2) **Re- trieval perspective:** After computing the precision and recall values for all concepts it was computed a mean precision equal to 0.38 (0.34 obtained using the standard version) and a mean recall equal to 0.44 (0.36 obtained using the standard ver sion). It can be observed that the values corresponding to the proposed modified model are always greater than the values of the initial model.

10) A COMPARATIVE IMAGE AUTO-ANNOTATION

(Author: Mahdia Bakalem, Nadjia Benblidia. (2013)).

The image annotation process presented in this paper is: The first annotation process based on texture parameters allows extracting texture visual features. The second an notation process based on color parameters extracts visual features of color. The latest process based on fusion of texture and color parameters permits to extract visual features of texture and color at the same time. The image auto-annotation process consists of two main steps: a training step and a new image processing step. Training Step: This step consists of regrouping the similar visual regions in classes called Blobs and annotating them. 1) Visual Space Preparation. The aim of this part is to construct the blobs that represent visual space. 2) Visual-Textual Correlation. We correlate between two aspects visual and semantic in order to annotate the blobs constructed in the first one. For each blob, author used a blob annotation algorithm which permits to inherit the image keys words of regions belong to the blob by selective heritage. B. New image processing step: This step permits to annotate automatically any new image; the first task is the segmentation of a new image into regions, followed by the extraction of the visual features of each region and by the affectation of regions to the blobs defined in the previous step (training step). The new image will be annotated by the selective heritage of the key words of the blobs to which its regions belong. In order to improve the annotation of image, we conduct our experiments on the annotated image data set referred to as Corel, consists of 16.000 images annotate manually with 1 to 5 keywords. Author used 1801 images in the training process , which have been considered in order to construct the annotated blobs. For each region, a visual features vector is defined, depending on the system. First vector is defined by texture parameters.The second is defined by color parameters (RGB space: the average, the variance and the moments order 3 of each component of RGB) and the last vector is defined by fusing the parameters of texture and color. The results show that the annotation by latent space is more promising than the annotation by textual space. Author suggests improvements in the system by using special images and also by refining the training process

11) <u>A survey in Deep Learning Model for Image annotation</u>

(AUTHOR: PHYU PHYU KHAING, MAY THE YU)

This paper is the survey for image annotation that have applied deep learning model. From this paper we have come to known that image annotation are of two types: 1) sentence based annotation2) Single word annotation . Image annotation has two kind of approaches 1) top-down approach 2) bottom-up approach to success the machine translation. Top-Down approaches apply the encoder-decoder architecture (Convolution Neural as encoder and LSTM as decoder). It initially takes the images into the encoder to get the feature and the features were fed into the decoder to generate the image description. The bottom-up approaches include several separated tasks, such as identifying objects or attributes, arranging words and sentences, describing sentences using a language model to generate the image caption.Deep Learning is also a technique that learns data from image to encourage the implementation of machine learning that is the function and structure of the brain known as artificial neural network. Deep learning is also called hierarchical learning and structure of brain known as artificial neural network. From this paper we have come to known that there are various models used in image annotation such as CNN,RNN,LSTM.

CNN stands for **Convolutional Neural Network** which are mostly used for speech and image recognition. **RNN** stands for **Recurrent Neural Network** which is used to predict next word by learning the current word.

LSTM stands **Long and Short Memory** which is one type of RNN.LSTM is applied for sentence representation in image annotation and also is developed with the gates such as input gate, output gate , forget gate and cell. From this paper we have also come acrossed various dataset used in the models.

There are various kinds of datasets that applied for image detection, image classification, image recognition , and image caption generation of image . MSCOCO , FLICKR 8K and FLICKR 30K and the standard benchmark datasets that are famous and mostly used for image annotation. All the dataset has three parts 1) Training set 2) Testing set and 3) Validation set.MSCOCO dataset has various version and each version contain different Number of images. FLICKR 8K contains 8,092 images where 6,000 images to train , 1000 images to test and 1000 images to validate. Five sentences are created for each image. FLICKR 30K contains 513,644 images for scene and entity and there has been working with five sentence per image and also it has 28,000 images to train ,1000 images to test and 1000 images to validate.

Evaluation metrics which are commonly used to evaluate the accuracy and effictiveness.The popular Evaluation metrics are BLEU,ROUGE,METEOR,CIDEr and SPICE. All these methods calculate with similarity based measure between ground truth sentence and machine generated sentence.

12) Building detection and segmentation using a CNN with automatically generated training data

(AUTHOR: Xiangyu zhuo, friedrichfraundorfer, franzkurz. peter reinartz)

Deep learning-based methods usually require a large amount of training data, which is quite labor-intensive and time-demanding.In this paper, to deal with the problem in generating training data, a novel approach to generate image annotations by transferring labels from **aerial images to UAV images** and the annotation using densely connected **CRF model** with an embedded naive **Bayes classifier** is proposed.Considering the fact that aerial images usually have much larger coverage than UAV images, we seek to propagate the labels from one aerial image to multiple co-registered UAV images an the UAV point cloud.Theoretically, we simply need to annotate one aerial image manually and then transfer the labels to numerous co-registered UAV images of the same area. A new pipeline for automatic image annotation generation is proposed to achieve this goal.

This approach consists of three steps-

1. Label one or two images manually.
2. Transfer the pixel labels to multiple UAV images using UAV point cloud.
3. Refine the generated annotations with densely connected CRF models and a naïve Bayes classifier.

To validate the accuracy of automatically generated annotations, we also train a deep convolutional neural network with the automatic annotations for image segmentation and compare the performance with manual annotations.

The learning procedure is implemented under the deep learning framework **Caffe**.

19 generated annotations featuring different regions of the survey Site were selected. In order to enrich the training data, we augment the annotations by scaling and rotating, resulting in around **8208 images** with the size of 300300.We fined tune the **FCN-8s model2** with our dataset. Besides, we plug in the **CRF-RNN** 3 layer in order to achieve sharp edges at class borders.

In our case, the interesting categories are Building, Roof, Ground, Vegetation and Car, while the indistinguishable objects are labeled as Clutter and will not be involved in segmentation.

2.2) Paper Comparison :

Sr.No	Paper Name	Author Name	Description
1)	Image annotation using SVM	1) Claudio 2) Cusano 3)Gianluigi Ciocca	This paper author suggested an image annotation tool using SUPPORT VECTOR MACHINE for classifying image regions in one of seven classes - sky, skin, vegetation, snow, water, ground, and buildings-oras unknown. The central idea of SVM is to adjust a discriminating function so that it makes Optimal use of the separability information of boundary cases.

2)	Automatically Annotating Images with Keywords: A Review of Image Annotation Systems	1)Chih Fong Tsai 2) Chihli Hung (2007)	This paper briefly describes commonly used image segmentation methods including global and local feature extraction. This paper reviews 50 image annotation systems published from 1997 to 2006 in terms of their image processing and learning modules
3)	Deep Captioning with Multimodal Recurrent Neural Networks(M-RNN)	1)Junhua Mao (2015)	This paper present a m-RNN model for generating novel image captions, over benchmark datasets ((IAPR TC-12, Flickr 8K, Flickr 30K and MS COCO). In addition, the m-RNN model is applied to retrieval tasks for retrieving images or sentences.
4)	Automatic Image Annotation using Neural Networks	1)Aanchan Mohan 2)Marwan A Tokri	This paper talks about the training process to achieve automatic image annotation using neural networks.
5)	Objective-Guided ImageAnnotation	1) IvorWai-Hung Tsang 2)Qi Mao (2013)	This paper talks about various methodologies to assign multiple tags for images. Author used several methods like SVM, kNN, MML, reverse MML, IAF, etc on COREL5k dataset, and compared the through put.
6)	Automatic Image Annotation Using Multi-Object Identification	1) Yin-Fu Huang 2) Hsin-Yun Lu (2010)	In this work ,author proposed a novel method for the task of image captioning, which achieves state of the art performance across popular standard benchmarks. Different from previous work, our method combines top-down and bottom-up strategies to extract richer information from an image.
7)	Automatic Multi-Label Image Annotation for smart cities.	1Gyayak Sanghi 2)Nalin Kanungo	In this author used ML-KNN algorithm to classify specific objects from datasets. In this case Objects that are seen in a city re expected as output.

<table>
<tr><td></td><td></td><td>(2017)</td><td></td></tr>
<tr><td><b>8)</b></td><td>Automated Annotation of Natural Images Using an Extended Annotation Model.</td><td>1) Gabriel Mihai
2)Liana Stanescu
(2012)</td><td>The annotation process implemented in this system is based on CMRM (Cross Media Relevance Model). The annotation model is based on object oriented approach. The CMRM annotation model has proved to be very efficient by several studies.</td></tr>
<tr><td><b>9)</b></td><td>A Comparative Image Auto-Annotation</td><td>1)Mahdia Bakalem
2)Nadjia Benblidia
(2013)</td><td>Image annotation process presented is: The first annotation process based on texture parameters allows extracting texture visual features. The second annotation process based on color. The latest process based on fusion of texture and color parameters permits to extract visual features of texture and color at the same time.</td></tr>
<tr><td><b>10)</b></td><td>A survey in Deep Learning Model for Image annotation</td><td>1) Phyu Phyu Khaing
2) May The Yu
(2019)</td><td><table><tr><td>TECHNIQUES</td><td>DATASET</td><td>EVALUATION METRICS</td></tr><tr><td>CNN</td><td>MSCOCO</td><td>BLUE</td></tr><tr><td>RNN</td><td>FLICKR8K</td><td>METEOR</td></tr><tr><td>LSTM</td><td>FLICKR30K</td><td>SPICE</td></tr></table></td></tr>
<tr><td><b>11)</b></td><td>Building detection and segmentation using a CNN with automatically generated training data</td><td>1)Xiangyu zhuo
2)Friedrich Fraundorfer
3)Franz kurz
4)Peter reinartz
(2019)</td><td>In this paper, to deal with the problem in generating training data, a novel approach to generate image annotations by transferring labels from aerial images to UAV images and the annotation using densely connected CRF model with an embedded naive Bayes classifier is proposed.</td></tr>
</table>

Chapter 3
SURVEY OF TECHNOLOGIES

In this chapter Survey of Technologies we demonstrate our awareness and understanding of Available Technologies related to the topic of our project. Given below are the detail of all the related technologies that are necessary to complete ourproject.

Convolution is a simple mathematical operation which is fundamental to many com- mon image processing operators. Convolution provides a way of 'multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values. In an image processing context, one of the input arrays is normally just a gray level image. The second array is usually much smaller, and is also two-dimensional (although it may be just a single pixel thick), and is known as the kernel.

The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. (Note that implementations differ in what they do at the edges of images, as explained below.) Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together. Convolution can be used to implement many different operators, particularly spatial filters and feature detectors.

Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

There are Four Types of Machine Learning

a) **Supervised learning.** In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

b) **Unsupervised learning.** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. Both the data algorithms train on and the predictions or recommendations they output are predetermined.

c) **Semi-supervised learning.** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

d) **Reinforcement learning.** Reinforcement learning is typically used to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

Deep Learning :

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks **.** . Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

Examples:

1) Automated Driving: Automotive researchers are using deep learning to automatically detect objects such as stop signs and traffic lights. In addition, deep learning is used to detect pedestrians, which helps decrease accidents.
2) Medical Research: Cancer researchers are using deep learning to automatically detect cancer cells. Teams at UCLA built an advanced microscope that yields a

high-dimensional data set used to train a deep learning application to accurately identify cancer cells.

Neural Network:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.A neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.Hidden layers fine-tune the input weightings until the neuralnetwork's margin of error is minimal.

Keras

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow.It was developed to make implementing deep learning models as fast and easy as possible for research and development.It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license. Keras was developed and maintained by François Chollet, a Google engineer. Keras is relatively straightforward to install if you already have a working Python and SciPy environment.You must also have an installation of Theano or TensorFlow on your system already.

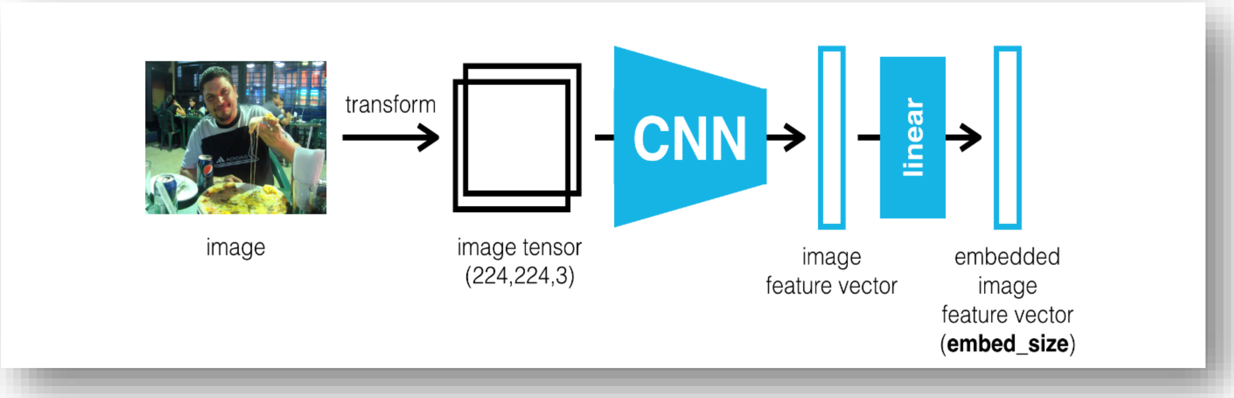
Tensorflow

TensorFlow is a framework created by Google for creating Deep Learning models. Deep Learning is a category of machine learning model. Machine Learning has enabled us to build complex applications with great accuracy. Whether it has to do with images, videos, text or even audio, Machine Learning can solve problems from a wide range. Tensorflow can be used to achieve all of these applications.

CHAPTER 4
MODELS

❖ Convolutional Neural Network

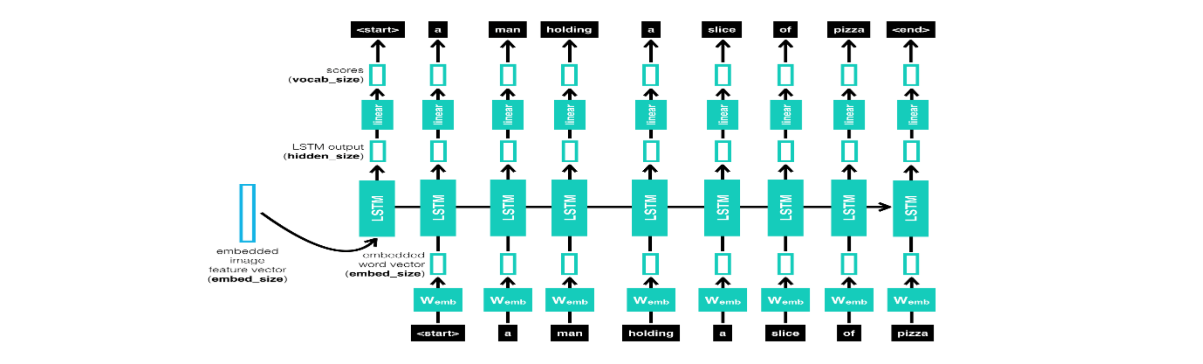
1) The convolutional neural network, or CNN for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data.
2) Convolutional neural networks do not learn a single filter; they, in fact, learn multiple features in parallel for a given input.
3) For example,
   It is common for a convolutional layer to learn from 32 to 512 filters in parallel for a given input. This gives the model 32, or even 512, different ways of extracting features from an input, or many different ways of both "learning to see" and after training, many different ways of "seeing" the input data. This diversity allows specialization, e.g. not just lines, but the specific lines seen in your specific training data.



4) CNN is often called as encoder because it encodes the content of the image into a smaller feature vector.

❖ Long short term memory

1) Recurrent neural networks (RNN) are the state of the art algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data.
2) RNN remembers things for just small durations of time, i.e. if we need the information after a small time it may be reproducible, but once a lot of words are fed in, this information gets lost somewhere. This issue can be resolved by applying a slightly tweaked version of RNNs – the Long Short-Term Memory Networks.
3) Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.



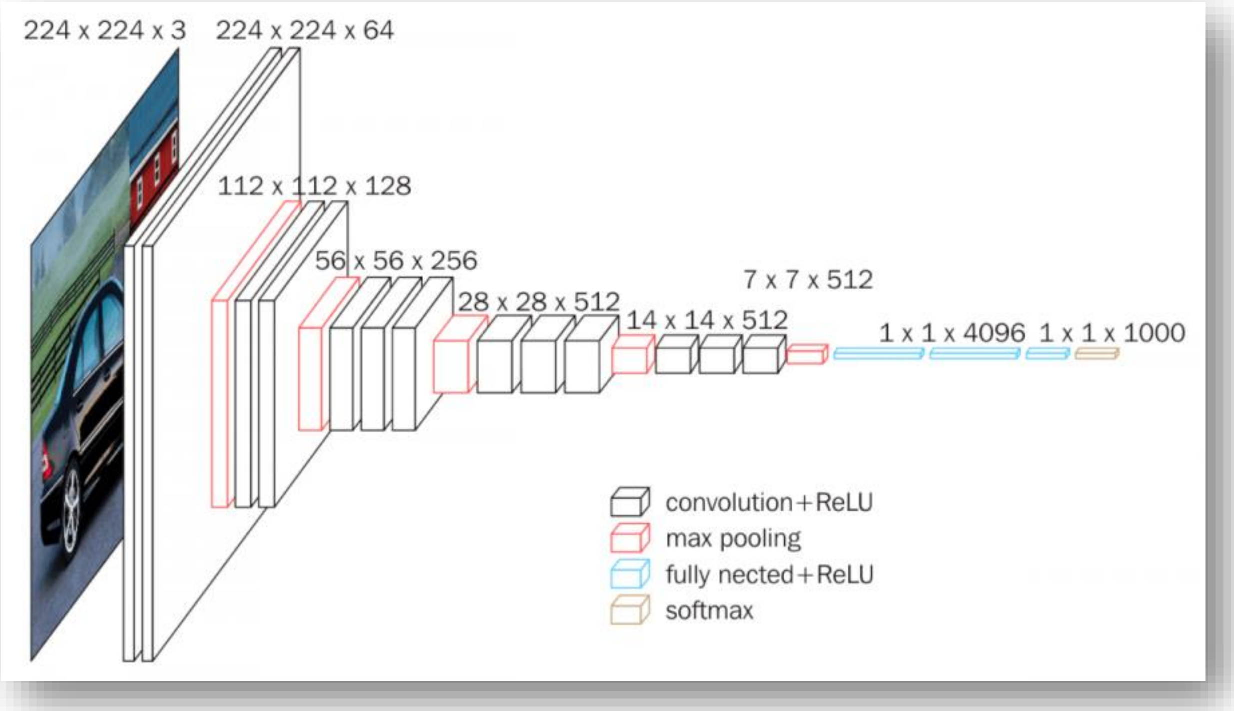
4) Lstm is called as decode because it decodes the vector and try to turn into a caption.
5) The LSTM based RNN is used to decode the process vector and turn it into a sequence of words.

❖ VGG16 MODEL

1) There are various pre-trained Convolutional Neural Network Trained model such as VGG19, VGG16, Inceptionv3, ResNet, Mobile Net etc. but in our project we are going to use VGG16 pre-trained CNN model.
2) VGG16 has 12 convolutional layers, some of which are followed by maximum pooling layers and then 4 fully connected layers and finally a 1000-way softmax classifier.
3) Input : (224,224,3)

   a) First two layer have 64 channels of 3*3 filter size and same padding.

   b) After a max pool layer of stride (2, 2) ad again 2 layer having convolutional layer of 256 and (3, 3) filter size.
4) At last there are 3 fully connected layer followed by last softmaxlayer.
5) In our project we are going to cut-off the last layer because vgg16 is use to classify the image but we don't want to classify the image but we want the internal representation of image so we are going to cut off the last layer.



Chapter 5
REQUIREMENTS AND ANALYSIS

5.1 Problem Definition

In this section we define the problem on which we are working in the project. Details are provided of the overall problem and then divided the problem in to sub-problems.

a) Problem Definition:

To build a system that will generate an automatic caption of the image by CNN , LSTM and trained VGG16 model. The image will be of the dataset provided to the system. The caption will get generate by identifying various objects in the image.

b) Sub-problem:

- To assign tags to images based on their semantic features. To build a system that can perform annotation automatically.
- To extract features using the favorable method.
- To classify multiple objects from a single image.
- To test and evaluate the performance of the system.

5.2 Requirements Specification

In this phase we define the requirements of the system. The Requirements Specification describes the things in the system and the actions that can be done on these things.

The requirements of the system are:

1) The image from the dataset and the dataset should have minimum two Phases including one for training and the second for testing.
2) A system or model to train and test the dataset.
3) High level API such as Keras and Tensor Flow.

5.3 Software and Hardware Requirements:

Hardware:

1) A computer system having a multi-core processor, minimum of 8GB RAM. Storage of minimum 500 GB and input and output peripherals.

Software:

1) Python
2) Anaconda
3) Jupyter Notebook

5.4 Evaluation Metrics:

The current study mostly uses the degree of matching between the caption sentence and the reference sentence to evaluate the pros and cons of the generation results. The commonly used methods include BLEU, METEOR, ROUGE, CIDEr, and SPICE these five measurement indicators. Among them, BLEU and METEOR are derived from machine translation, ROUGE is derived from text abstraction, and CIDEr and SPICE are specific indicators based on image captioning.

- **BLEU**

Bleu is widely used in the evaluation of image annotation results, which is based on the n-gram precision. The principle of the BLEU measure is to calculate the distance between the evaluated and the reference sentences. BLEU method tends to give the higher score when the caption is closest to the length of the reference statement.

- **ROUGE**

ROUGE is an automatic evaluation standard designed to evaluate text summarization algorithms. There are three evaluation criteria, ROUGE-N, ROUGE-L, and ROUGE-S. ROUGE-N is based on the given sentence to be evaluated, which calculates a simple n-tuple recall for all reference statements: ROUGE-L is based on the largest common sequence (LCS) calculating the recall. ROUGE-S calculates recall based on co-occurrence statistics of skip-bigram between reference text description and prediction text description.

- **CIDEr**

CIDEr is the special method which is provided for the image captioning work. It measures consensus in image captioning by performing a term frequency inverse document frequency for each n-gram. Studies have shown that the match between CIDEr and human consensus is better than other evaluation criteria.

- **METEOR**

METEOR is based on the harmonic mean of unigram precision and recall, but the weight of the recall is higher than the accuracy. It is highly relevant to human judgment and differs from the BLEU in that it is not only in the entire set, but also in the sentence and segmentation levels, and it has a high correlation with human judgment.

- **SPICE**

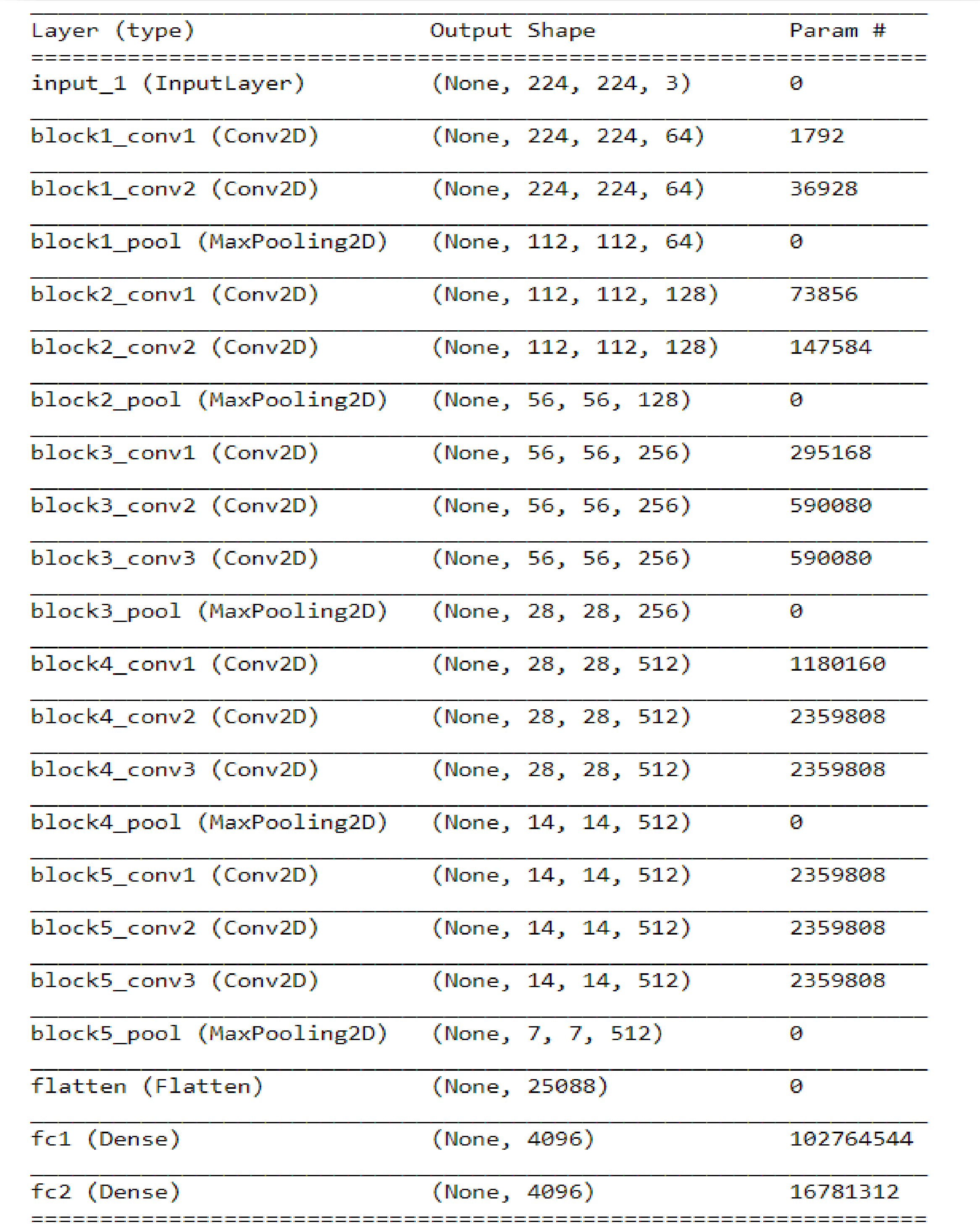
SPICE evaluates the quality of image captions by converting the generated description sentences and reference sentences into graph-based semantic representations, namely "scene graphs". The scene graphs extract lexical and syntactic information in natural language and explicitly represents the objects, attributes, and relationships contained in the image.

Chapter 6
Implementation

Code:

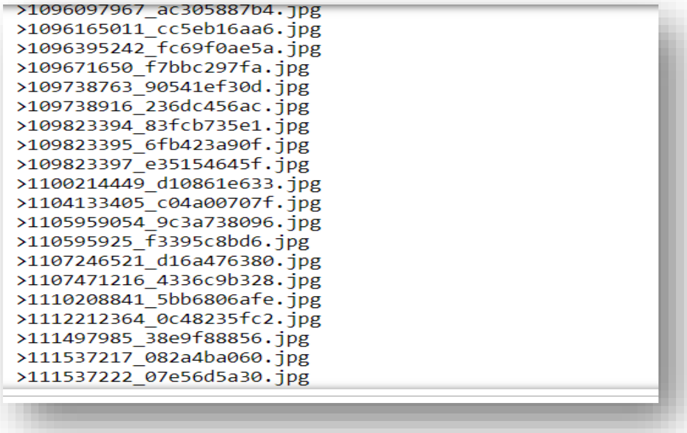
```
from os import listdir
import tensorflow as tf
from pickle import dump
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.applications.vgg16 import preprocess_input
from tensorflow.keras.models import Model

#tf.compat.v1.get_default_graph()

# extract features from each photo in the directory
def extract_features(directory):
    model = VGG16() #loading the model
    model.layers.pop() #restructure (Removing the last layer of the model)
    model = Model(inputs=model.inputs, outputs=model.layers[-1].output)
    print(model.summary()) #summarize
```

Output:



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
=================================================================
 features = dict() #extracting features from each photo
    for name in listdir(directory):
         filename = directory + '/' + name #loading the image from file
         image = load_img(filename, target_size=(224, 224))
         image = img_to_array(image)#converting image to numpy array
         image = image.reshape((1, image.shape[0],
image.shape[1],image.shape[2])) #reshaping data
         image = preprocess_input(image)
# preparing the image for the VGG model
         feature = model.predict(image, verbose =0) #get the features
         image_id = name.split('.')[0] #getting the image id
         features[image_id] = feature #storing the features
         print('>%s' %name)
    return features
directory = 'Dataset/Images'
features = extract_features(directory)
print('Extracted Features : %d' % len(features))
dump(features , open('features.pkl',"wb"))
```

Output:



```
>1096097967_ac305887b4.jpg
>1096165011_cc5eb16aa6.jpg
>1096395242_fc69f0ae5a.jpg
>109671650_f7bbc297fa.jpg
>109738763_90541ef30d.jpg
>109738916_236dc456ac.jpg
>109823394_83fcb735e1.jpg
>109823395_6fb423a90f.jpg
>109823397_e35154645f.jpg
>1100214449_d10861e633.jpg
>1104133405_c04a00707f.jpg
>1105959054_9c3a738096.jpg
>110595925_f3395c8bd6.jpg
>1107246521_d16a476380.jpg
>1107471216_4336c9b328.jpg
>1110208841_5bb6806afe.jpg
>1112212364_0c48235fc2.jpg
>111497985_38e9f88856.jpg
>111537217_082a4ba060.jpg
>111537222_07e56d5a30.jpg
```

```
# Loading the file containg all the descriptions into memory

def load_doc(filename):
    # Opening the file as read only
    file = open(filename, 'r')

    # Reading all text and storing it.
    text = file.read()

    # Closing the file
    file.close()

    return text

def photo_to_description_mapping(descriptions):

    # Dictionary to store the mapping of photo identifiers to descriptions
    description_mapping = dict()

    # Iterating through each line of the descriptions
    for line in descriptions.split('\n'):

        # Splitting the lines by white space
        words = line.split()

        # Skipping the lines with length less than 2
        if len(line)<2:
            continue
       # The first word is the image_id and the rest are the part of the
description of that image
        image_id, image_description = words[0], words[1:]

    # Retaining only the name of the image and removing the extension from
it
        image_id = image_id.split('.')[0]

  # Image_descriptions contains comma separated words of the description,
hence, converting it back to string
        image_description = ' '.join(image_description)
        # There are multiple descriptions per image,
        # hence, corresponding to every image identifier in the dictionary, there is a list of description
        # if the list does not exist then we need to create it

        if image_id not in description_mapping:
            description_mapping[image_id] = list()

        # Now storing the descriptions in the mapping
        description_mapping[image_id].append(image_description)

    return description_mapping
def clean_descriptions(description_mapping):

    # Preapring a translation table for removing all the punctuation
    table = str.maketrans(",", string.punctuation)

    # Traversing through the mapping we created
    for key, descriptions in description_mapping.items():
        for i in range(len(descriptions)):
            description = descriptions[i]
            description = description.split()

            # Converting all the words to lower case
            description = [word.lower() for word in description]

            # Removing the punctuation using the translation table we made
            description = [word.translate(table) for word in description]

            # Removing the words with length =1
            description = [word for word in description if len(word)>1]

            # Removing all words with number in them
            description = [word for word in description if word.isalpha()]

            # Converting the description back to string and overwriting in the descriptions list
            descriptions[i] = ' '.join(description)
```

```
# Converting the loaded descriptions into a vocabulary of words

def to_vocabulary(descriptions):

    # Build a list of all description strings
    all_desc = set()

    for key in descriptions.keys():
        [all_desc.update(d.split()) for d in descriptions[key]]

    return all_desc

# save descriptions to file, one per line
def save_descriptions(descriptions, filename):
    lines = list()
    for key, desc_list in descriptions.items():
        for desc in desc_list:
            lines.append(key + ' ' + desc)
    data = '\n'.join(lines)
    file = open(filename, 'w')
    file.write(data)
    file.close()

filename = 'Dataset/Textfiles/Flickr8k.token.txt'

# Loading descriptions
doc = load_doc(filename)

# Parsing descriptions
descriptions = photo_to_description_mapping(doc)
print('Loaded: %d ' % len(descriptions))

# Cleaning the descriptions
clean_descriptions(descriptions)

# Summarizing the vocabulary
vocabulary = to_vocabulary(descriptions)
print('Vocabulary Size: %d' % len(vocabulary))
```

```
# Saving to the file
save_descriptions(descriptions, 'descriptions.txt')
```

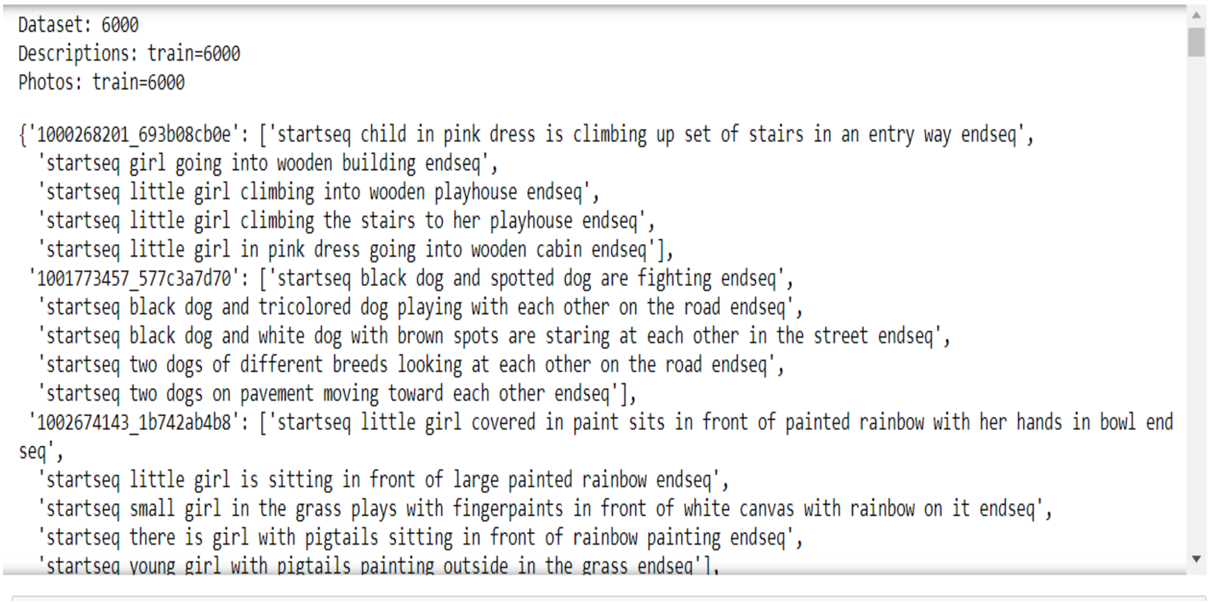
<u>Output</u> **- Loaded: 8092**

Vocabulary Size: 8763

```
from pickle import load
# Function for loading a file into memory and returning text from it
def load_file(filename):
    file = open(filename, 'r')
    text = file.read()
    file.close()
    return text

# Function for loading a pre-defined list of photo identifiers
def load_photo_identifiers(filename):

    # Loading the file containing the list of photo identifier
    file = load_file(filename)

    # Creating a list for storing the identifiers
    photos = list()

    # Traversing the file one line at a time
    for line in file.split('\n'):
        if len(line) < 1:
            continue

        # Image name contains the extension as well but we need just the name
        identifier = line.split('.')[0]

        # Adding it to the list of photos
        photos.append(identifier)

    # Returning the set of photos created
    return set(photos)
```

```
# loading the cleaned descriptions that we created earlier
# we will only be loading the descriptions of the images that we will use for training
# hence we need to pass the set of train photos that the above function will be returning

def load_clean_descriptions(filename, photos):

    #loading the cleaned description file
    file = load_file(filename)

    #creating a dictionary of descripitions for storing the photo to description mapping of train images
    descriptions = dict()

    #traversing the file line by line
    for line in file.split('\n'):
        # splitting the line at white spaces
        words = line.split()

        # the first word will be the image name and the rest will be the description of that particular image
        image_id, image_description = words[0], words[1:]

        # we want to load only those description which corresponds to the set of photos we provided as argument
        if image_id in photos:
            #creating list of description if needed
            if image_id not in descriptions:
                descriptions[image_id] = list()

            #the model we will develop will generate a caption given a photo,
            #and the caption will be generated one word at a time.
            #The sequence of previously generated words will be provided as input.
            #Therefore, we will need a ‘first word’ to kick-off the generation process
            #and a ‘last word‘ to signal the end of the caption.
            #we will use 'startseq' and 'endseq' for this purpose
        #also we have to convert image description back to string

        desc = 'startseq ' + ' '.join(image_description) + ' endseq'
        descriptions[image_id].append(desc)

    return descriptions

# function to load the photo features created using the VGG16 model
def load_photo_features(filename, photos):

    #this will load the entire features
    all_features = load(open(filename, 'rb'))

    #we are interested in loading the features of the required photos only
    features = {k: all_features[k] for k in photos}

    return features


filename = 'Dataset/Textfiles/Flickr_8k.trainImages.txt'
train = load_photo_identifiers(filename)
print('Dataset: %d' % len(train))
# descriptions
train_descriptions = load_clean_descriptions('descriptions.txt', train)
print('Descriptions: train=%d' % len(train_descriptions))
# photo features
train_features = load_photo_features('features.pkl', train)
print('Photos: train=%d' % len(train_features))
train_descriptions
```

Output:



```
Dataset: 6000
Descriptions: train=6000
Photos: train=6000

{'1000268201_693b08cb0e': ['startseq child in pink dress is climbing up set of stairs in an entry way endseq',
  'startseq girl going into wooden building endseq',
  'startseq little girl climbing into wooden playhouse endseq',
  'startseq little girl climbing the stairs to her playhouse endseq',
  'startseq little girl in pink dress going into wooden cabin endseq'],
 '1001773457_577c3a7d70': ['startseq black dog and spotted dog are fighting endseq',
  'startseq black dog and tricolored dog playing with each other on the road endseq',
  'startseq black dog and white dog with brown spots are staring at each other in the street endseq',
  'startseq two dogs of different breeds looking at each other on the road endseq',
  'startseq two dogs on pavement moving toward each other endseq'],
 '1002674143_1b742ab4b8': ['startseq little girl covered in paint sits in front of painted rainbow with her hands in bowl end
seq',
  'startseq little girl is sitting in front of large painted rainbow endseq',
  'startseq small girl in the grass plays with fingerpaints in front of white canvas with rainbow on it endseq',
  'startseq there is girl with pigtails sitting in front of rainbow painting endseq',
  'startseq young girl with pigtails painting outside in the grass endseq'],
```

Chapter 7
Conclusion

7.1 Conclusion:

After researching through various papers related to Automatic Image Annotation. We have concluded that, a system can be developed that can automatically annotate images based on their visual aspects. We will test our system against benchmark datasets and compare our results based on precision, number of tags, and efficiency of the system.

BIBLOGRAPHY:

1) Claudio Cusano, Gianluigi Ciocca. *Image* nnnoto/ion usinp*SVM*
2) Chih-Fong Tsai, Chihli Hung. *Automatically Annotating Images with Keywords:AReviewofImaǥjeAnnotation*Systems.RecentPatentsonComputerScience 2008, 1, 55-68, 1874- 4796 /08 c 2008 Bentham Science Publishers Ltd.(2007)
3) Junhua Mao *Deep Captioning with Multimodel Recurrent Neural Networks(M-RNN)*Published as a conference paper at ICLR 2015.(2015).
4) Aanchan K Mohan, Marwan A. Tokri *Automatic Image Annotation* usinp *Neural Networks*.
5) Ivor Wai-Hung Tsang, Qi Mao. *ObJ eC tive- Guided Image Annotation.* IEEE TRANS- ACTIONS ON IMAGE PROCESSING, VOL. 22, NO. 4, 1057 7149/ c. 2012 IEEE.(2013).
6) Yin-Fu Huang, Hsin-Yun Lu *Automatic Image Annotation Using Multio b jeclIdeiiIi-*/cotion. Fourth Pacific-Rim Symposium on Image and Video Techno1ogy,978-0-7695- 4285-0/10 c 2010 IEEE DOI 10.1109/PSIVT.2010.71. (2010).
7) Nalin Kanungo, Gyayak Sanghi *Automatic Multi-Label Image Annotation for Smart* Cities. IEEE Region 10 Symposium (TENSYMP), 978-1-5090-6255-3/17/c20l7IEEE. (2017).
8) Gabriel Mihai, Liana Stanescu. *Automaled Annotation* oJ *Naturel Images Using* nn *E:ctended Annotation Model.* International Journal of Computer Science and Appli- cations Technomathematics Research Foundation Vol. 9, No. 3, pp. 1 — 19.(2012)
9) Mahdia Bakalem, Nadjia Benblidia. *A Comparative* /rape *Annotation.* 978-1-4799-4796-6/13/ c 2013 IEEE. (2012).
10) www.researchgate.net/publication/332109738_A_Survey_in_Deep_Learning_Model_for_Image_Annotation.
11) www.researchgate.net/publication/328993377_Building_Detection_and_Segmentation_Using_a_CNN_with_Automatically_Generated_Training_Data.