

Object Detection Based Automatic Image Captioning using Deep Learning

By

18CP809:Trushna Patel

Guided by

Dr. D.G. Thakore

Dr. N. M. Patel

A Dissertation Submitted to
Birla Vishvakarma Mahavidyalaya (Engineering College), An Autonomous Institution
affiliated to Gujarat Technological University in Partial Fulfillment of the Requirements
for
the Master of Technology (*Computer Engineering*) with Specialization in *Software Engineering*

June 2020



BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
AN AUTONOMOUS INSTITUTION
Vallabh Vidyanagar – 388120
GUJARAT, INDIA

COMPLIANCE CERTIFICATE

This is to certify that the research work embodied in this dissertation entitled “***Object Detection Based Automatic Image Captioning using Deep Learning***” was carried out by **18CP809:Trushna Patel**, at Birla Vishvakarma Mahavidyalaya (Engineering College) An Autonomous Institution for partial fulfillment of Master of Technology (***Computer Engineering***) with Specialization in ***Software Engineering*** degree to be awarded by Gujarat Technological University. She has complied to the comments given by the Dissertation phase – I as well as Mid Semester Dissertation Reviewer to my / our satisfaction.

Date :

Place :

(***Trushna Patel***)

(***Dr. D. G. Thakore***)

(***Dr. N. M. Patel***)

Head, (***Computer Department***)
(***Dr. D. G. Thakore***)

Principal
(***Dr. I. N. Patel***)



BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
AN AUTONOMOUS INSTITUTION
Vallabh Vidyanagar – 388120
GUJARAT, INDIA

PAPER PUBLICATION CERTIFICATE

This is to certify that the research work embodied in this dissertation entitled “*Object Detection Based Automatic Image Captioning using Deep Learning*” was carried out by **18CP809:Trushna Patel** at Birla Vishvakarma Mahavidyalaya (Engineering College) An Autonomous Institution for partial fulfillment of Master of Technology (*Computer Engineering*) with Specialization in *Software Engineering* degree to be awarded by Gujarat Technological University. He / She has published/article entitled “*A Survey on Object Detection Based Automatic Image Captioning using Deep Learning*” accepted for publication by the *International Journal for Modern Trends in Science and Technology* at *Vallabh Vidyanagar, Anand, Gujarat, India* during/on **April 2020**.

Date :

Place :

(Trushna Patel)

(Dr. D. G. Thakore)

(Dr. N. M. Patel)

Head, (Computer Department)
(Dr. D. G. Thakore)

Principal
(Dr. I. N. Patel)



BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
AN AUTONOMOUS INSTITUTION
Vallabh Vidyanagar – 388120
GUJARAT, INDIA

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this dissertation and that neither any part of this dissertation nor the whole of the dissertation has been submitted for a degree to any other University or Institution.

I certify that, to the best of my knowledge, my dissertation does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in my dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Indian Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my dissertation and have included copies of such copyright clearances to our appendix.

I declare that this is a true copy of dissertation, including any final revisions, as approved by my dissertation review committee.

Date:

(18CP809:Trushna Patel)



BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
AN AUTONOMOUS INSTITUTION
Vallabh Vidyanagar – 388120
GUJARAT, INDIA

CERTIFICATE

This is to certify that work embodied in this dissertation entitled “*Object Detection Based Automatic Image Captioning using Deep Learning* “ was carried out by **18CP809:Trushna Patel**, at Birla Vishvakarma Mahavidyalaya (Engineering College) An Autonomous Institution for partial fulfillment of Master of Technology (*Computer Engineering*) with Specialization in *Software Engineering* degree to be awarded by Gujarat Technological University. This work has been carried out under my / our supervision meets the requirement of Gujarat Technological University.

Date :

Place :

(Dr. D. G. Thakore)

(Dr. N. M. Patel)

Head, (Computer Department)
(Dr. D. G. Thakore)

Principal
(Dr. I. N. Patel)



BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
AN AUTONOMOUS INSTITUTION
Vallabh Vidyanagar – 388120
GUJARAT, INDIA

DISSERTATION APPROVAL CERTIFICATE

This is to certify that dissertation titled “*Object Detection Based Automatic Image Captioning using Deep Learning*” was carried out by **18CP809:Trushna Patel** at Birla Vishvakarma Mahavidyalaya (Engineering College) An Autonomous Institution is approved for award of the degree of Master of Technology (*Computer Engineering*) with Specialization in *Software Engineering* by Gujarat Technological University.

Date :

Place :

Signature:			
Name:			

Examiners



BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
AN AUTONOMOUS INSTITUTION
Vallabh Vidyanagar – 388120
GUJARAT, INDIA

Abstract

Nowadays, people necessitate engendering captions for multiple reasons such as, posting an image on social media, creating news headlines from an image and many more. An Image Captioning system intends to produce captions for an image automatically instead of manually writing. It delivers a descriptive sentence for an image, that helps people to better understand the semantic meaning of an image. Image understanding is an essential technique to interpret semantic image data which can be implemented by VGG16. Image Captioning is an application for both Natural Language Processing and Computer Vision and can be achieved using either Traditional Machine Learning approach or Deep Learning approach. The necessity in performing the intended task is detecting objects and establishing relationships among objects. Feature Extraction is a technique for converting the image into a vector for further processing. The objects and image content are forwarded to the LSTM that will connect the words to produce a descriptive sentence. The work carried out by this thesis presents the implementation model for Object Detection based Image Captioning using Deep Learning. The implementation model is based on the article by J. Brownlee[1] while the use of object detection for captioning is based on X. Yin et al.[2] and also provides the comparison between the results obtained by both VGG16 and InceptionV3.

Acknowledgments

The presented thesis is based on “Object Detection Based Automatic Image Captioning using Deep Learning”. I would not be able to complete the research conducted without contributions of many people.

I bethink as an honor to accomplish the presented thesis under the guidance of Dr. Darshak G. Thakore and Dr. Narendra M. Patel. Their supervision made this thesis a success. Dr. Darshak G. Thakore and Dr. Narendra M. Patel were available to monitor my work and help me whenever I needed. I would like to acknowledge them for guiding me as a mentor and suggesting me in each and every circumstance.

I am unable to express their support in words and I would also like to appreciate the support of my parents who encouraged me in my life.

I would also thank to Birla Vishvakarma Mahavidhyalaya College of Engineering for allowing me to coordinate with the other faculties and college resources for the success of this thesis. I would be grateful to all the other people who directly or indirectly helped and encouraged me to complete this experiment.

Table of Contents

List of Figures.....	iv
List of Tables.....	vi
Chapter 1: Introduction.....	1
1.1. Introduction.....	1
1.2. Objective.....	2
1.3. Outline.....	2
Chapter 2 : Literature Survey.....	3
2.1. Template-Based Image Captioning[3].....	3
2.2. Retrieval-Based Image Captioning[3].....	3
2.3. A novel based Image Captioning[3].....	3
Chapter 3 : Method for Image Captioning.....	10
3.1. Introduction.....	10
3.2. Image Captioning Process.....	10
3.2.1. Object Detection using YOLO[21].....	12
3.2.2. Feature Extraction using a CNN Model.....	12
3.2.3. Preprocessing the training captions.....	13
3.2.4. Build Tokenizers and generate Vocabulary for both captions and the detected objects.....	13
3.2.5. Generate Model for training.....	13
3.2.6. Evaluate the model to achieve the results as BLEU[20] score and generating captions for test images.....	14
Chapter 4 : Dataset Details and Evaluation Metrics.....	16
4.1. Dataset Details.....	16
4.2. Evaluation Metrics.....	16
Chapter 5 : Implementation Results and Observation of Results.....	17
5.1. Object Detection using YOLO[34].....	18
5.2. Feature Extraction using VGG16[21].....	20
5.3. Results for VGG16[25] as CNN.....	21
5.4. Results for InceptionV3[27] as CNN.....	21
5.5. Results with Bidirectional LSTM as language model.....	21
5.6. Observation of Results.....	25
Chapter 6 : Summary.....	27
References.....	28
APPENDIX:A ABBREVIATION.....	
APPENDIX:B REVIEWS.....	

List of Figures

Figure No.	Figure Title	Page No.
3.1.	General Flow Diagram according to J. Brownlee[1] and X. Yin et al.[2].....	11
3.2.	YOLO Architecture[23].....	12
3.3.	VGG16 Architecture[26].....	12
3.4.	RNN Architecture[29].....	13
3.5.	LSTM Architecture[18].....	14
3.6.	Image Captioning Model based on J. Brownlee[1].....	15
5.1.1	Objects Detected and Bounding Boxes – 1.....	18
5.1.2	Objects and Confidence.....	18
5.1.3	Objects Detected and Bounding Boxes – 2.....	18
5.1.4	Objects and Confidence.....	18
5.1.5	Objects Detected and Bounding Boxes – 3.....	18
5.1.6	Objects and Confidence.....	18
5.1.7	Objects Detected and Bounding Boxes – 4.....	18
5.1.8	Objects and Confidence.....	18
5.1.9	Objects Detected and Bounding Boxes – 5.....	19
5.1.10	Objects and Confidence.....	19
5.1.11	Objects Detected and Bounding Boxes – 6.....	19
5.1.12	Objects and Confidence.....	19
5.1.13	Objects Detected and Bounding Boxes – 7.....	19
5.1.14	Objects and Confidence.....	19
5.1.15	Objects Detected and Bounding Boxes – 8.....	19
5.1.16	Objects and Confidence.....	19
5.1	Object Detection results using YOLO[34] implemented on Flickr8K[19] Dataset.....	19
5.2	Feature Extraction Results on Flickr8K[19] Dataset.....	19
5.2.1	Input for Feature Extraction – 1.....	19
5.2.2	Input for Feature Extraction – 2.....	19
5.2.3	Input for Feature Extraction – 3.....	19
5.2.4	Input for Feature Extraction – 4.....	19
5.2.5	Input for Feature Extraction – 5.....	19
5.2.6	Input for Feature Extraction - 6.....	19
5.2	Feature Extraction results on Flickr8K[18] Dataset.....	19
5.4.1	Model while using InceptionV3[22] based on J. Brownlee[1].....	20
5.4.2	Model while using Bidirectional LSTM[13] based on J. Brownlee[1].....	21
5.3.1.	Flickr8K[19] Input Image for VGG16[25] – 1.....	22
5.3.2.	Flickr8K[19] Input Image for VGG16[25] – 2.....	22
5.3.3.	Flickr8K[19] Input Image for VGG16[25] – 3.....	22
5.3.4.	Flickr8K[19] Input Image for VGG16[25] – 4.....	22
5.3.5.	Flickr8K[19] Input Image for VGG16[25] – 5.....	22
5.3.6.	Flickr8K[19] Input Image for VGG16[25] – 6.....	22
5.3	Captions generated when using VGG16[25] as CNN on Flickr8K[19] Dataset.....	22
5.4.2.	Flickr8K Input Image for InceptionV3[27] – 1.....	23
5.4.3.	Flickr8K Input Image for InceptionV3[27]– 2.....	23
5.4.4.	Flickr8K Input Image for InceptionV3[27]– 3.....	23
5.4.5.	Flickr8K Input Image for InceptionV3[27]– 4.....	23
5.4.6.	Flickr8K Input Image for InceptionV3[27]– 5.....	23

5.4.7.	Flickr8K Input Image for InceptionV3[27]– 6.....	23
5.4	Captions generated when using InceptionV3[27] as CNN on Flickr8K[19] Dataset.....	23
5.5.2.	Flickr8K Input Image for Bidirectional LSTM – 1.....	24
5.5.3.	Flickr8K Input Image for Bidirectional LSTM – 2.....	24
5.5.4.	Flickr8K Input Image for Bidirectional LSTM – 3.....	24
5.5.5.	Flickr8K Input Image for Bidirectional LSTM – 4.....	24
5.5.6.	Flickr8K Input Image for Bidirectional LSTM – 5.....	24
5.5.7.	Flickr8K Input Image for Bidirectional LSTM – 6.....	24
5.5	Captions generated while using Bidirectional LSTM[35] on Flickr8K[19] Dataset.....	24

List of Tables

Table No.	Title	Page No.
2.1	Literature Survey.....	6, 7
4.1	Dataset Details.....	16

Chapter 1

Introduction

1.1 Introduction

An Image Captioning task consists of describing an image in sentence form. This requires brief knowledge of computer vision and natural language processing. The challenging task in the process of caption generation is to understand the semantics, that contains the objects and other image information, and knowledge of natural language processing. The sentence generation needs the establishment of a relationship between the extracted objects. Image information can be extracted using the technique known as Feature Extraction. An image contains various information such as objects and its meaning in the context of an image.

Image Understanding is broadly divided into two categories: (1) Traditional Machine Learning techniques and (2) Deep Learning techniques[3]. In traditional machine learning techniques, features can be extracted using the Scale-Invariant Feature Transform(SIFT) and Histogram of Oriented Gradient(HOG)[3]. In deep learning techniques, the Convolutional Neural Network(CNN) is used, which is followed by Recurrent Neural Network(RNN) for generating the captions[3].

The report presents a way to achieve Image Captioning using Object Detection. The presented approach uses the YOLO(You Only Look Once) method for Object Detection. It detects an object and provides the bounding box and confidence for the detected object. Along with objects, the image data is also important for understanding image properly. Features extraction is used to extract some useful features and important image data from image. Feature Extraction is implemented using two types of Convolutional Neural Network(CNNs) viz., VGG16 and InceptionV3. The object and image feature resulted from this techniques, are provided to the Recurrent Neural Network(RNN) for the sentence generation. RNN takes them as an input and creates a variable-length description as an output. RNN generates a textual description for an image.

Image Captioning has its roots in many real-life applications[4], [5] such as:

- Image Captioning is useful for blind or low vision people that rely on audio where the text captions are converted into audio for them.
- Used for video subtitles
- Used in self-driving cars
- Used for efficient image search while browsing
- Used for interpreting newspaper articles

1.2 Objectives

- The Objective of Image Captioning is to generate meaningful sentences for the given input image
- Establish the relationship between image features and word sequences
- Study the research carried out previously in the area of Image Captioning.
- Build an Image Caption Generation Model.
- Study on the categories of Image Captioning in Deep Learning.
- Use of Object Detection for generating captions for an image

1.3 Outline

- Chapter-2 provides brief discussion on the literature. It includes the basic approaches towards Image Captioning which includes Template based, Retrieval based and Novel based Image Captioning.
- Chapter-3 provides the Image Captioning System that was implemented in this experiment with steps. It also contains the methods used to generate captions based on Object Detection.
- Chapter-4 discusses the Image Captioning Datasets and its Evaluation Metrics
- Chapter-5 provides the implementation results using both VGG16 and InceptionV3 and Bidirectional LSTM in the model. This chapter also provides the observations on the presented variety of models.
- Chapter-6: concludes the report and the experimentation work.

Chapter 2

Literature Survey

Deep learning is an improved way of achieving image captioning in terms of better results. There are three types of approaches in deep neural network-based methods to solve this problem as introduced by M. Z. Hossain et al. [3]:

1. Template-based Image Captioning.
2. Retrieval based Image Captioning
3. A novel based Image Captioning

2.1 Template-based Image Captioning[3]

The template-based approaches have fixed templates with several blank slots that are filled after detecting objects and attributes[3]. The sample of the Template-Based approach is provided by D. Hutchison et al.[6], which make us of a triplet *<object, action, scene>* of scene elements to fill the blank slots. S. Li et al.[7] discussed extraction the phrases from the detected objects, attributes and their relationships for generating captions. Though this approach provides grammatically correct captions, it does not generate variable length captions[3]. Template-based Image Captioning generates grammatically correct captions but the templates are of fixed length and hence cannot generate variable length captions[3].

2.2 Retrieval based Image Captioning[3]

Retrieval based Image Captioning requires captions that are retrieved from the existing set of captions. This approach finds visually similar images with their associated captions from the training dataset[3]. These collected captions are called candidate captions. The input image can be captioned by matching the input image and collected images. The matching images will be re-ranked based on the matched content and return the top-ranked caption as a result[3]. M. Hodosh[8] and P. Kuznetsova[9] presented the similar approach for image captioning. Though this approach generates syntactically correct captions, it does not generate semantically correct captions. This approach generates syntactically correct captions but cannot generate image specific semantically correct captions.

2.3 A novel based Image Captioning[3]

Novel based Image Captioning uses both visual and multimodal space for generating captions. It analyzes the visual content from the image and then generates captions based on the visual information. In the Novel Image Caption Generation

Approach, it analyzes the image content and then generates image captions with the help of language models[3]. This approach generates new captions for each image that are accurate and semantically correct.

N. K. Kumar et al.[10], in their research, proposed region based approach for captioning image. The Image Captioning is implemented by the objects detected using Region Based Object Detection (RODe). In the region-based approach, the image is divided into regions for Object Detection. The specific technique used by authors is Region Based CNN or R-CNN. The research work includes other techniques such as Feature Extraction and Scene Classification that make use of CNN. The RNN technique is used for the sentence generation with the help of objects and scene attributes and image feature.

C. Amritkar et al.[11] proposed a model that make use of LSTM instead of simple RNN. The research implemented Feature Extraction using the pre-trained VGG16 model. Since RNN cannot remember the previously predicted words, LSTM can be used as LSTM can remember the words for longer period of time. LSTM is a cell that stored the words till the caption is generated properly. According to this, the results observed in categories such as descriptions with errors, without errors, related and unrelated captions. These categories are due to considerations of the neighborhood of words.

P. Shah et al.[12] proposed the Show and Tell method for image captioning. Show and Tell method make use of both image recognition and neural machine translation. The proposed method is a combination of the Inception-v3 model and LSTM. Inception-v3 model is used for object recognition while, the LSTM cell is used for storing intermediate words during the captioning process. The input image is first processed by the Inception-v3 model for object recognition and produces the vector form of an image. This vector is used by LSTM in order to generate description.

F. Fang et al.[13] proposed a Word Level Attention model for Image Captioning. The model has a word-level attention layer is produced to process image features with two models for word prediction accurately. The two modes are Line Level Bidirectional spatial embedding used for feature maps and the Attention mode to extract word-level attention. It also used Bidirectional LSTM networks that process words in both the directions i.e., forward and backward. The Word Level Attention Extraction used to extract visual information to predict the next word using a softmax activation function.

D.-J. Kim et al.[14] focuses on better sentence learning using Deep Convolutional Network. A deep Fisher Kernel method is used for image feature extraction. The extracted activations are aggregated to form a Fisher Vector. Instead of LSTM, this approach uses gLSTM. Since the information of the input image is only provided in the first step, the input image information gets diluted as it proceeds. gLSTM provides a piece of additional information called 'guide', that pertains to the input image information throughout the process.

A. Poghosyan et al.[15] proposed an approach which enables the use of LSTM with Read-Only LSTM. LSTM cell is a storage mechanism that provides the ability to store the intermediate words while the read-only LSTM cell provides image features. The image content is provided only in the first step thus, the predicted word may or may not be related to the image. to relate both previous and current word, an additional unit is required in order to predict current word related to image content. The additional unit so produced is called LSTM with Read-Only Unit that results in better accuracy.

Tanti et al.[16] discussed both the types of architectures i. e, Merge Architecture and Inject Architecture. The merge architecture merges the image with final state of RNN while the inject architecture injects image into the RNN for further process. He concluded that the merge architecture performs better with state size 256 and inject architecture performs better with state size 128 on Flickr8K Dataset

Lu et al.[17] proposed Adaptive Attention model that learns adaptively when needed. They also discussed that focusing on proper words is important rather than paying attention on conjunctions, punctations etc. The model proposed performed better with visual attention.

	Paper Title	Author Name	Journal Name	Publication Year	Method Used	Advantages	Disadvantages	Remark
1	Detection and Recognition of Objects in Image Caption Generator System: A Deep Learning Approach[10]	N. Komal Kumar, D. Vigneswari, A. Mohan, K. Laxman, J. Yuvaraj	2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)	2019	R-CNN, CNN, RNN	Improve existing image caption generator system using Deep Learning Approach		Flickr8K Dataset
2	Image Caption Generation using Deep Learning Technique[11]	Chetan Amritkar, Vaishali Jabade	2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)	2018	CNN, LSTM, RNN	Reduce losses and increase efficiency for larger dataset	Due to neighborhood words for an object, captions generated may be incorrect	Flickr8k —BLEU = 0.53356, Flickr30k -- BLEU = 0.61433, MSCOCO -- Dataset BLEU = 0.61433
3	Image Captioning using Deep Neural Architectures [12]	Parth Shah, Vishvajit Bakrola, Supriya Pati	2017 International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS)	2017	Show and Tell, LSTM	Uses advancement of Object Recognition as InceptionV3 model	InceptionV3 requires large compute power	MSCOCO Dataset, BLUE score is 65.5
4	Image Captioning with Word Level Attention[13]	Fang Fang, Hanli Wang, Pengjie Tang	2018 25th IEEE International Conference on Image Processing (ICIP)	2018	CNN, RNN, LSTM, L2BE, WLAE	Comparison between various state-of-the-art methods like, NIC, m-RNN, Attention with CNN and RNN, gLTSM	Error may propagate to subsequent layers, which is solved by attention but errors cannot be fully removed	MSCOCO Dataset — BLUE-4=34.0, CIDEr=106.0

						ResNet etc		
5	Sentence Learning on Deep Convolutional Networks for Image Caption Generation [14]	Dong-Jin Kim, Donggeun Yoo, Bonggeun Sim, In So Kweon	2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)	2016	CNN, Deep Fisher Kernel, gLSTM	gLSTM provides better results compared to LSTM	Does not provide fluent sentences with proper adjectives	Flickr8K Dataset — BLUE-1=66.8
6	Long Short-Term Memory with Read-Only Unit in Neural Image Caption Generator[15]	Aghasi Poghosyan, Hakob Sarukhanyan	2017 Computer Science and Information Technologies (CSIT)	2017	CNN, RNN with LSTM, LSTM with Read-only Unit	Increases model accuracy, Generate more accurate captions		MSCOCO Dataset
7	What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator? [16]	Marc Tanti, Albert Gatt, Kenneth P. Camilleri	2017 Proceedings of the 10th International Conference on Natural Language Generation	2017	CNN, RNN using both Merge and Inject Architecture	Better results with size 256 in merge architecture and 128 in inject architecture for Flickr8K		Flickr8k, Flickr30k, MSCOCO
8	Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning [17]	J. Lu, C. Xiong, D. Parikh, and R. Socher	2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)	2017	Adaptive Attention Model, CNN, RNN	Adaptive attention allows paying attention on proper words rather than extra words, conjunctions etc.		Flickr30k: BLEU4: 0.251 MSCOCO: BLEU4: 0.332

Table 2.1. Literature Survey

According to M. Z. Hossain et al.[3], there are various types of Deep Learning methods such as:

- i. Visual Space vs. Multimodal Space
- ii. Dense Captioning vs. Captions for the whole scene
- iii. Encoder-Decoder Architecture vs. Compositional Architecture
- iv. LSTM vs. Others

According to M. Z. Hossain et al.[3], The detailed description of those variants are described in the below points

- i. In Visual Space-Based methods[3], image features and their captions are separately passed to the language decoder while in Multimodal-Based methods, a shared multimodal is learned from an image and their captions and passed to the language. The Multimodal-Based method[3] consists of a language encoder, vision part, multimodal space part and language decoder. The language encoder part contains the word features while the vision part performs the feature extraction using CNN[3]. The multimodal part maps the features with the word features and the language decoder generates captions[3].
- ii. Dense Captioning is well described by M. Z. Hossain et al.[3], uses different image regions to extract information of image objects and provide region-wise captions whereas the latter method provides a caption for the whole image, irrespective of the image regions[3]. This method generates a sentence for each object of an image and combines them to create a full image description. Dense Captioning collects region-wise information and generates descriptions for each region[3].
- iii. In Encoder-Decoder[3] based methods, it provides an end to end manner of generating captions using encoder-decoder architecture. It uses CNN for image feature extraction and then fed to RNN for the generation of words related to an image[3]. In Compositional architecture[3] based methods, image captioning is composed of several independent methods. This method integrates independent building blocks into a pipeline to generate captions[3]. It uses CNN for image understanding and then a set of candidate caption is generated. The final caption to be generated as output by re-ranking these candidate captions[3].
- iv. M. Z. Hossain et al.[3], in their paper, discussed variations of RNNs for sentence generation. LSTM is a type of RNN with a memory cell that maintains the information of an image over a long period of time. LSTM is used in sequence to the sequence learning task. LSTM contains various gates such as the input gate, the output gate, and the forget gate[18]. These gates that learns what information is relevant to keep or forget[18]. A Gated Recurrent Unit(GRU) is similar to LSTM but does not use a separate memory cell and uses less number of gates to flow the information[18]. It contains two types of gates namely, update gate and

reset gate. The Update gate takes care of what information to keep and what to throw away while the Reset gate decides how much past information to forget[18].

Chapter 3

Method for Image Captioning

3.1 Introduction

An Image Captioning system can be performed with the help of various datasets such as Flickr8K, Flickr30K, and MSCOCO. This report contains the results conducted on the Flickr8K[19] dataset. The Image Captioning system requires a Feature Extraction technique, that enables the use of a type of Convolutional Network(CNN) called VGG16. The model is evaluated with the help of Bilingual Evaluation Understudy(BLEU)[20]. The language model used for generating captions is a variation of Recurrent Neural Network(RNN) called Long Short Term Memory(LSTM).

The use of object detection for image captioning is based on the survey provided by X. Yin et al.[2] while, the model generation and training is based on the article by J. Brownlee[1]. According to X. Yin et al.[2], the model uses YOLO method for object detection along with CNN for feature extraction. The model layers used for training the implemented model is based on the study of article given by J. Brownlee[1], which make use of VGG16 and LSTM for image captioning rather than using object detection for captioning. The system implemented by this work uses VGG16 for feature extraction and LSTM as language model according to J. Brownlee[1] but with the use of object detection using YOLO according to X. Yin et al.[2]. The model implemented by this work combines the advantages of the work described by both J. Brownlee[1] and X. Yin et al.[2].

3.2 Image Captioning Process

Various steps involved in order to produce a caption from the image according to J. Brownlee[1] are:

- i. Object Detection using You Look Only Once(YOLO)
- ii. Feature Extraction using a CNN model
- iii. Preprocessing the training captions
- iv. Build Tokenizers and generate Vocabulary for both captions and the detected objects
- v. Generate Model for training
- vi. Evaluate the model to achieve the results as BLEU score and generating captions for test images

The above 5 steps are involved in the training process while the last step is for the evaluation and testing process. The complete process for training and testing the Image Caption Generating system is shown in the Figure 3.1 below. The left part depicts the training process while the right part depicts the testing process. The output of training process is an .h5py file which can be used to generate captions for test images and also for the generated model evaluation. The work flow generated in order to implement the model is based on J. Brownlee[1] and X. Yin et al.[2].

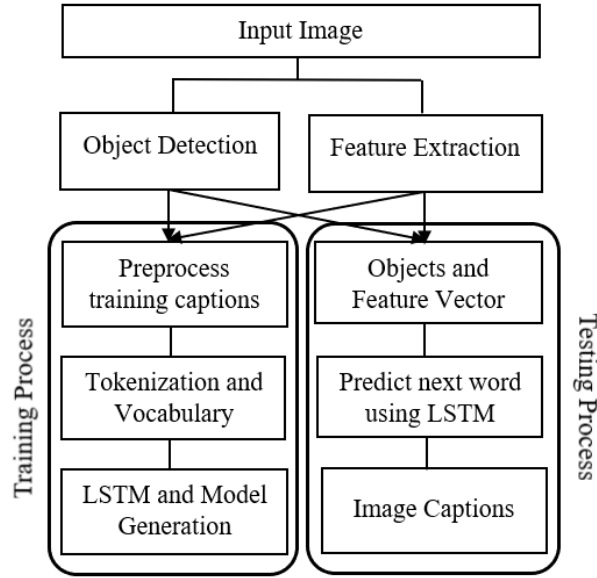


Figure 3.1 General Flow Diagram according to J. Brownlee[1] and X. Yin et al.[2]

3.2.1 Object Detection using YOLO[21]

A fast technique for Object Detection YOLO[21], is implemented in this research. It splits the input image into the $S \times S$ grid. Each grid is in charge of detecting one object. Along with object detection, each grid is also responsible for predicting a fixed number of bounding boxes[21]. A grid cell predicts B bounding boxes, a bounding box score, C conditional class probabilities and one object per grid irrespective of several bounding boxes[22]. The bounding box delivered by the grid cell consists of various parameters that properly describes the bounding box. Those parameters include X-Coordinate(x), Y-Coordinate(y), Height of box(h), Width of the box(w), and confidence[21]. The coordinates help to locate the object in the image. If the confidence is high, the probability of an object in that box increases. The class probability categorizes the object. A grid may contain many bounding boxes but the box with highest confidence and the area intersected by the bounding boxes may contain the object[21]. The YOLO architecture is depicted in the Figure 3.2 below, which states that it accepts an input image of size 448×448 and gradually decrease the size by max pooling to receive the spatial dimension of 7×7 [21]. YOLO uses two fully connected layers to get $7 \times 7 \times 2$ boundary box prediction and choose the box with high

confidence[21].

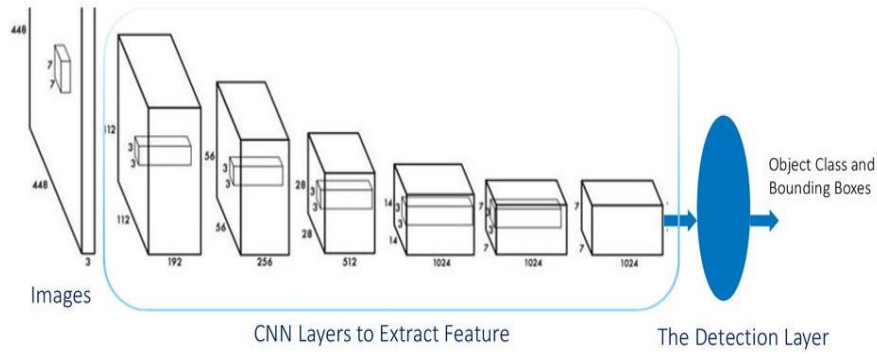


Figure 3.2 YOLO Architecture[23]

3.2.2 Feature Extraction From CNN[24]

Feature Extraction is the technique for extracting image feature from the image. An image feature is the image content in form of vector. The CNN layers includes Convolution layer, Max Pooling Layer, and Dense Layer. The last Dense Layer from the model is removed in case of using pre-trained model.

3.2.2.1 Feature Extraction using VGG-16[25]

A VGG-16 is a variant of Convolutional Neural Network with 16 layers used for image classification. VGG-16 is pre-trained on ImageNet dataset[25]. The VGG-16 architecture and its layers are described in the below figure. It accepts a 224*224 size input image and delivers 4096 sized feature vectors. The convolution layers process the image data and the max pooling layers reduce the size of image by half. Convolution layers only alters the image data while the pooling layers changes the size of image. Max Pooling includes the maximum values for further procedure[25]. Convolution layer takes 3*3 filter with stride 1 while, max pooling layer takes 2*2 filter with stride 2[25]. The layered architecture of VGG16 is shown in the Figure 3.3 which involves the use of convolution layers for processing the image data while the use of pooling layer for reducing the dimension of the image to get 4096 size vectors[25].

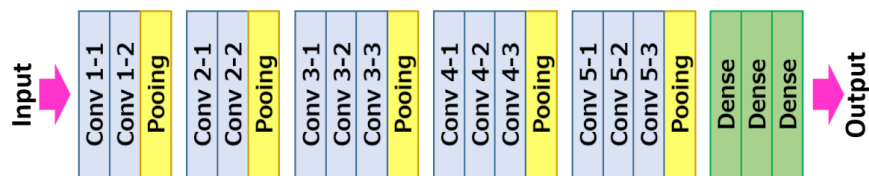


Figure 3.3 VGG16 Architecture[26]

3.2.2.2 Feature Extraction using InceptionV3[27]

Inceptionv3 is also a Feature Extraction model that outputs feature vector after removing the last layer. The model is originally trained in ImageNet Dataset for image classification. After removing last layer, the resulting feature vector is 2048 size image vector[27]. It requires the image to be resized into 299*299 image.

3.2.3 Preprocessing the training captions

Training dataset captions contain some symbols such as punctuations, white space, and apostrophe[1]. These captions require preprocessing to generate a proper word vocabulary by creating tokens for each word. This step converts each word of the dataset into token and removes the apostrophe attached to it. It also removes other punctuations and converts each letter into a lower case.

3.2.4 Build Tokenizers and generate Vocabulary for both captions and the detected objects

In order to build Vocabulary for all the words occurring in the training captions, the Tokenizer[28] class is used in Keras. The goal of this class is to convert each word into either integers, where each integer is the index for the token stored in the dictionary, or into a vector, where the coefficient of tokens is binary based on the word count in the dataset[28].

3.2.5 Generate Model for Training

3.2.5.1 Recurrent Neural Network(RNN)[29]

The basic task of RNN is to predict the next word in the sequence of words to engender a meaningful description. In RNN a neuron is provided to the network first and then current state is calculated using a combination of both current input and the previous state[29]. All the previously predicted words are assembled to generate caption. As the method uses backpropagation mechanism, the output is compared to the original output. If the error is encountered, the error is backpropagated to the network to update the weights that will help to retrieve weights that provide least error[29]. The RNN architecture is described in Figure 3.4 which describes that the previous output h_i and current input X_i is used to predict the next word as an output.

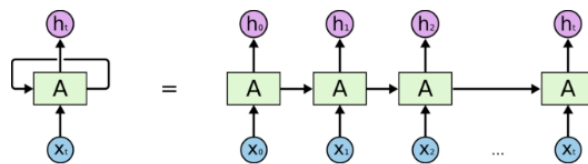


Figure 3.4 RNN Architecture[29]

3.2.5.2 Long Short Term Memory(LSTM)[18]

RNN cannot remember the previously predicted words for longer period of time hence a storage mechanism is needed to store those words. LSTM networks layer consists of a set of recurrently connected blocks known as memory[18]. It contains a cell and three units[18]: input gate, output gate and forget gate. LSTM gates regulate the flow of information into and out of the cell. The Forget[18] gate decides if the

information should be kept or throw away. The Input[18] gate which information in the cell state should be updated, where the Cell State[18] is the storage mechanism that keeps the task relevant information. The Output[18] gate decides what will be the next hidden state that can be used for evaluating the next word. Figure 3.5 shows the LSTM cell with all the described types of gates and cell state. The forget gate, input gate and output gate uses either sigmoid activation function or the tangent hypotential function to decide which information is important and which is to be removed from the cell state.

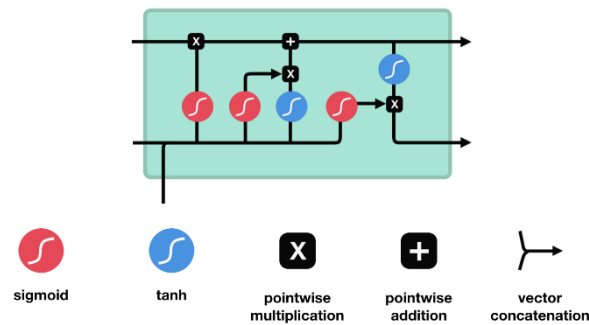


Figure 3.5 LSTM cell[18]

3.2.5.3 Model Training

For a model to be generated, all the results of the previous computation are required such as, Detected Objects, Image Features, preprocessed captions, and the vocabulary. The model represented in this report is based on from the experiment performed by J. Brownlee[1]. The model takes 3 inputs for training that are image features, detected objects, and the training captions word by word. The feature vector is 4096 size vectors generated by VGG16 while the maximum length of the detected objects is 49 and that of captions is 34. The Embedding and LSTM is used for handling the captions in training process. All the 3 inputs are merged by converting each with the same size 256. The model generated is shown in Figure 3.6.

3.2.6 Evaluate the model to achieve the results as BLEU[20] score and generating captions for test images

The model can be evaluated using many evaluation parameters such as BLEU[20], METEOR[30], and CIDEr[31]. The presented model is evaluated using BLEU score. The detailed description about those evaluation parameters are described in section 4.2

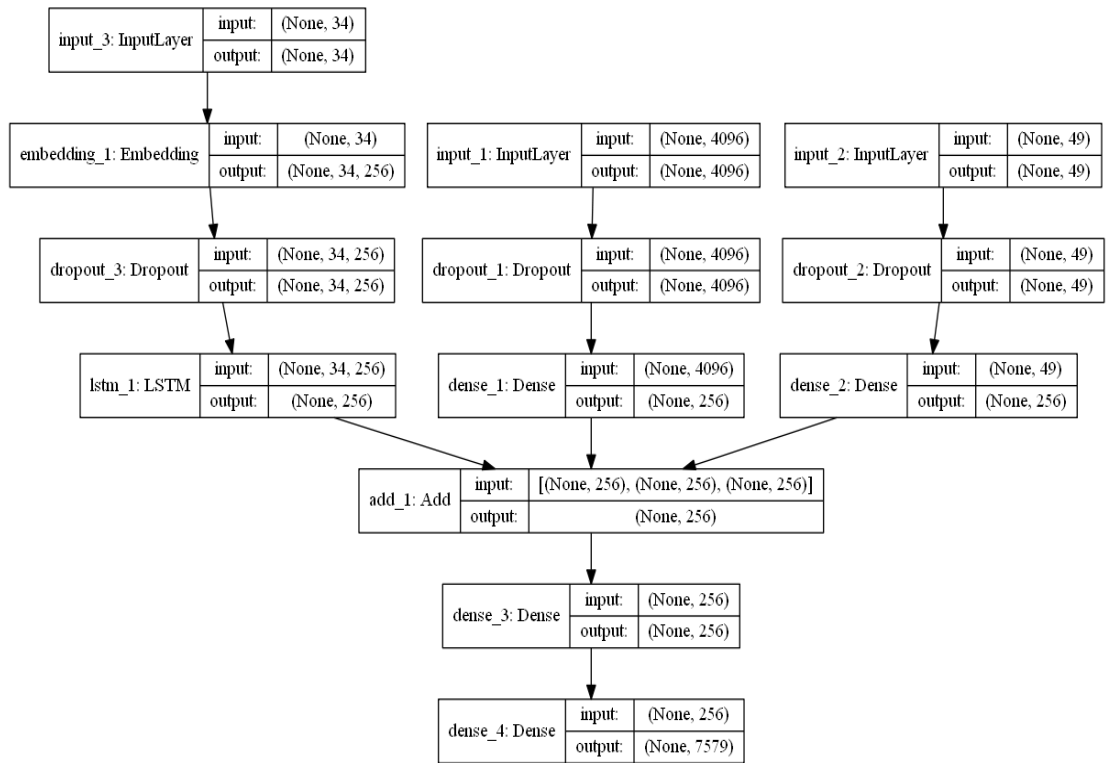


Figure 3.6 Image Captioning Model based on J. Brownlee[1]

Chapter 4

Dataset Details and Evaluation Metrics

4.1 Dataset Details

There are three datasets available for Image Captioning with variety of image types. Flickr8K Dataset[19] has 8K images with 6K training images, 1K validation images and 1K testing images. Flickr30K Dataset[32] has 30K images with 28K training images, 1K validation images and 1K testing images. MSCOCO[33] is an Object detection and Image Captioning dataset introduced by Microsoft. MSCOCO contains 328K images with 82783 training images, 40504 validation images and 40775 testing images.

Dataset Name	Size		
	Train	Valid	Test
Flickr8K[19]	6000	1000	1000
Flickr30K[32]	28000	1000	1000
MSCOCO[33]	82783	40504	40775

Table 4.1. Dataset Details

4.2 Evaluation Metrics

There are variety of evaluation metrics for Image Captioning technique to evaluate the produced captions. The metric that is mostly considered by the authors is BLEU(Bilingual Evaluation Understudy)[20], which is used to evaluate a machine-generated text. The produced text segments are compared with the set of reference text and scores for each text segment[3]. The complete evaluation can be determined by averaging that individual text. However the BLEU score is popular for machine translation, it is only suitable for short captions[20]. An alternative is METEOR(Metric for Evaluation of Translation with explicit ORdering)[30], which is similar to BLEU i.e., used to evaluate a machine-generated text. It compares the word segment with reference text. It also provides matching of synonyms of words, thus making a better correlation of sentence[3]. CIDEr(Consensus-Based Image Description Evaluation)[31] is a metric for evaluation image descriptions. It also provides a consensus between generated and human suggested descriptions[3].

Chapter 5

Implementation Results and Observation of Results

- This experiment includes the implementation of each step involved in the task of Image Captioning conducted on Flickr8K[19] Dataset.
- The subsequent sections of this chapter use Flickr8K[19] Dataset for conducting the experiment of Object Detection, Feature Extraction and Image Captioning.
- The sub-topics in this section provides the results of Object Detection, Feature Extraction and caption generation. The descriptions produced are resulted in 3 different ways viz., using VGG16[25], using InceptionV3[27] and using Bidirectional LSTM[13].

5.1 Object Detection using YOLO[34]



Figure 5.1.1 Objects Detected and Bounding Boxes - 1

pottedplant: 0.6189
person: 0.9998
pottedplant: 0.5257
sports ball: 0.6737

Figure 5.1.2 Objects and Confidence



Figure 5.1.3 Objects Detected and Bounding Boxes - 2

person: 0.9896
person: 0.9611
person: 0.9990
sports ball: 0.8350

Figure 5.1.4 Objects and Confidence



Figure 5.1.5 Objects Detected and Bounding Boxes - 3

person: 0.9854
surfboard: 0.8197

Figure 5.1.6 Objects and Confidence

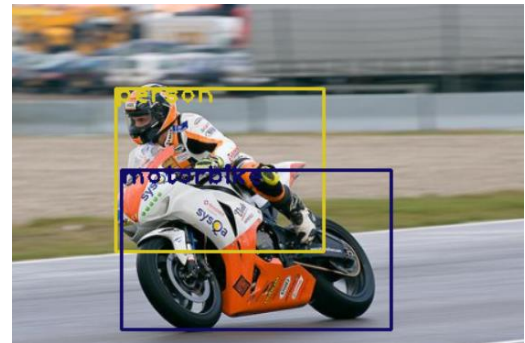


Figure 5.1.7 Objects Detected and Bounding Boxes - 4

person: 0.9990
motorbike: 0.9902

Figure 5.1.8 Objects and Confidence

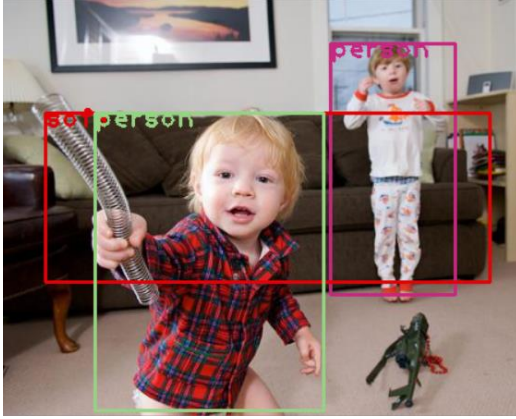


Figure 5.1.9 Objects Detected and Bounding Boxes - 5

person: 0.9601
sofa: 0.9951
person: 0.9995

Figure 5.1.10 Objects and Confidence



Figure 5.1.11 Objects Detected and Bounding Boxes - 6

dog: 0.9897
frisbee: 0.5840

Figure 5.1.12 Objects and Confidence



Figure 5.1.13 Objects Detected and Bounding Boxes - 7

person: 0.9445
snowboard: 0.9952
person: 0.7225

Figure 5.1.14 Objects and Confidence

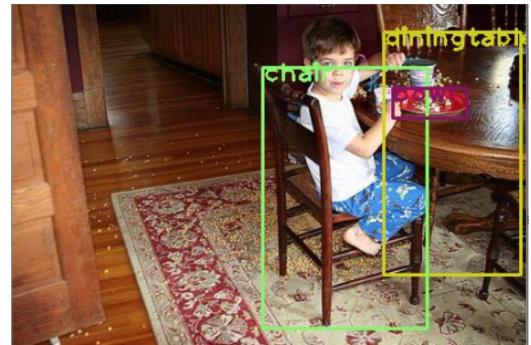


Figure 5.1.15 Objects Detected and Bounding Boxes - 8

diningtable: 0.5242
chair: 0.9942
bowl: 0.5936

Figure 5.1.16 Objects and Confidence

Figure 5.1. Object Detection results using YOLO[34] implemented on Flickr8K[19] Dataset

- YOLO is performed on Flickr8K[19] Dataset and the results shown in Table 5.1 includes the objects with bounding box and their confidence.

5.2 Feature Extraction using VGG16[21]



Figure 5.2.1 Input for Feature Extraction - 1

Feature Vector:
array([[-0. , -0. , -0. , ..., -0. ,
2.971505 , 4.6332965]], dtype=float32)



Figure 5.2.2 Input for Feature Extraction - 2

Feature Vector:
array([[3.6856923 , 1.921611 , -0. , ..., -
0. , 1.1533508 , 0.91561234]],
dtype=float32)



Figure 5.2.3 Input for Feature Extraction - 3

Feature Vector:
array([[0.597802 , -0. , -0. , ...,
0.45515984, 1.0223458 , 0.7752677]],
dtype=float32)



Figure 5.2.4 Input for Feature Extraction - 4

Feature Vector:
array([[1.2609477, -0. , 1.123865 , ..., -0.
, -0. , -0.]], dtype=float32)



Figure 5.2.5 Input for Feature Extraction - 5

Feature Vector:
array([[-0., -0., -0., ..., -0., -0., -0.]],
dtype=float32)



Figure 5.2.6 Input for Feature Extraction - 6

Feature Vector:
array([[1.064071 , 0.21511406, -0. , ...,
-0. , 0.33713078, 2.1175423]],
dtype=float32)

Figure 5.2. Feature Extraction Results on Flickr8K[19] Dataset

Table 5.2 shows some results of features extracted using VGG16[25]. The output feature vector is the 4096-size array of float type.

5.3 Results for VGG16[25] as CNN

VGG16[25] is used as CNN in the Image Captioning for Feature Extraction. The output of VGG16 is a feature vector of size 4096 after removing the last layer. The input image should be resized in 224*224 in order to meet the image size required by the model. The model architecture is depicted in Figure 3.6. Table 5.3 shows the input image from Flickr8K[19] Dataset and its corresponding caption while using VGG16[25].

5.4 Results for InceptionV3[27] as CNN

InceptionV3[27] is used as a CNN for Feature Extraction. The output of InceptionV3 is a feature vector of size 2048. The input image should be resized in 299*299 as the model requires an image of same size. The model architecture is presented in the Figure 5.4.1, which accepts a 2048-size input feature vector received from InceptionV3[27]. Table 5.4 shows the input image from Flickr8K[19] Dataset and its corresponding caption while using InceptionV3[27].

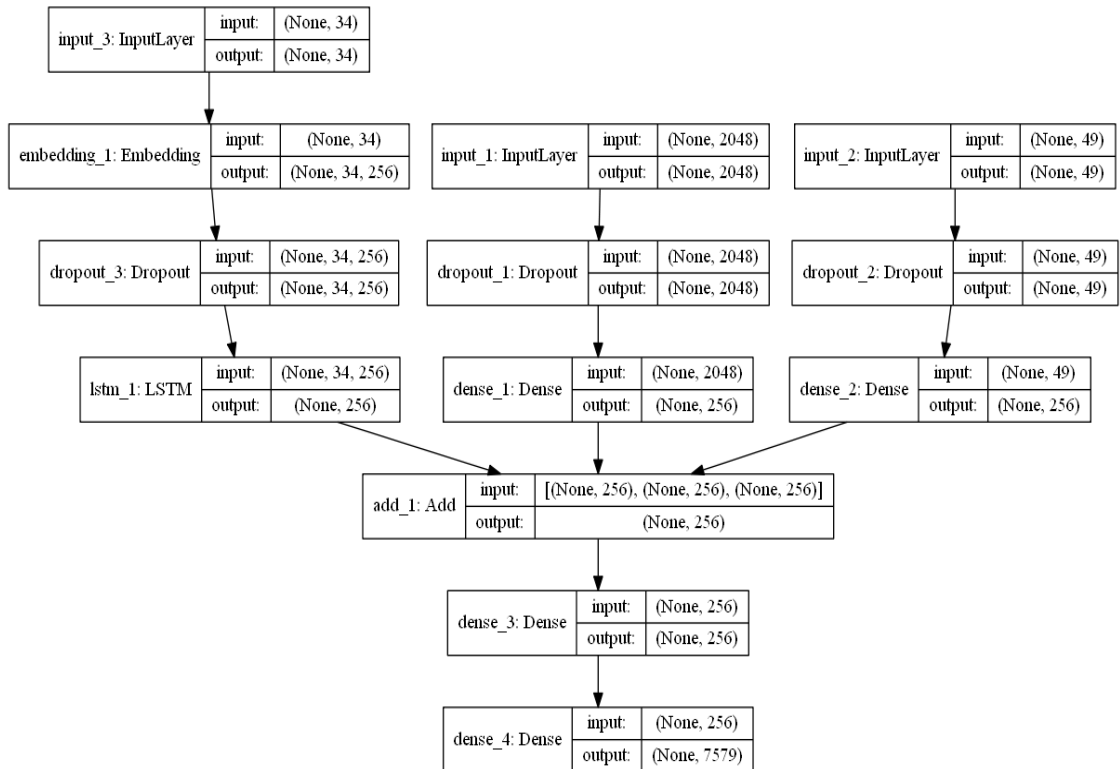


Figure 5.4.1. Model while using InceptionV3[22] based on J. Brownlee[1]

5.5 Results with Bidirectional LSTM as language model

It also used Bidirectional LSTM networks that process words in both the directions

i.e., forward and backward[13]. The model architecture using Bidirectional LSTM is shown in the Figure 5.5.1, which uses a Bidirectional LSTM to create description. Table 5.5 shows the input image from Flickr8K[19] Dataset and its corresponding caption while using Bidirectional LSTM[13].

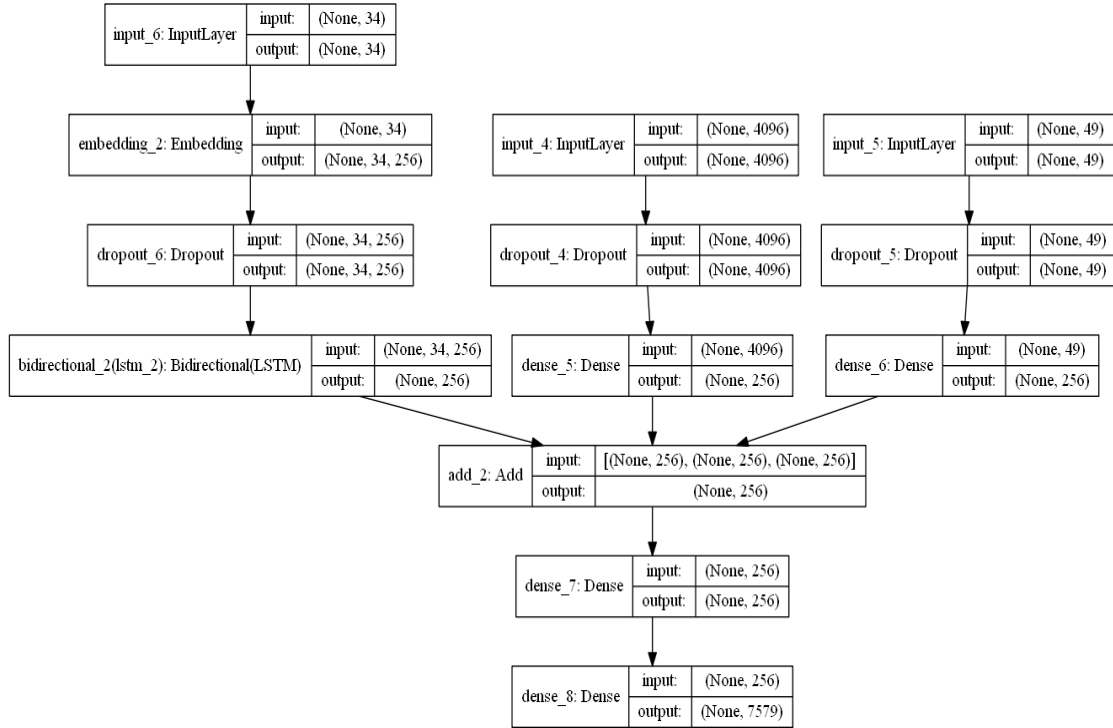


Figure 5.5.1 Model while using Bidirectional LSTM[13] based on J. Brownlee[1]



Figure 5.3.1 Flickr8K[19] Input Image
for VGG16[25] - 1

Objects: person
Caption: startseq skier is riding mountain
endseq



Figure 5.3.2 Flickr8K[19] Input Image
for VGG16[25] - 2

Objects: person
Caption: startseq the man is standing
on the grass endseq



Figure 5.3.3 Flickr8K[19] Input Image
for VGG16[25] - 3

Objects: person,person
Caption: startseq the player is playing
basketball endseq



Figure 5.3.4 Flickr8K[19] Input Image
for VGG16[25] - 4

Objects: person,surfboard
Caption: startseq the boy is playing in
the water endseq



Figure 5.3.5 Flickr8K[19] Input Image
for VGG16[25] - 5

Objects: backpack,person,backpack
Caption: startseq the skier is riding the
snow endseq

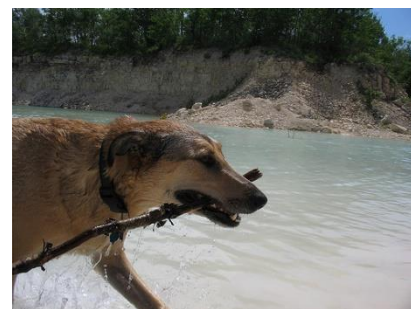


Figure 5.3.6 Flickr8K[19] Input Image
for VGG16[25] - 6

Objects: dog
Caption: startseq the dog is running in
the water endseq

Figure 5.3. Captions generated when using VGG16[25] as CNN on Flickr8K[19] Dataset



Figure 5.4.2 Flickr8K[19] Input Image for InceptionV3[27] - 1

Objects: person
Caption: startseq the man is sitting on the water endseq



Figure 5.4.3 Flickr8K[19] Input Image for InceptionV3[27] - 2

Objects: person
Caption: startseq the girl is wearing red shirt and blue shirt endseq



Figure 5.4.4 Flickr8K[19] Input Image for InceptionV3[27] - 3

Objects: person, person
Caption: startseq the boy is wearing red shirt and white shirt endseq



Figure 5.4.5 Flickr8K[19] Input Image for InceptionV3[27] - 4

Objects: person, surfboard
Caption: startseq the girl is sitting on the water endseq



Figure 5.4.6 Flickr8K[19] Input Image for InceptionV3[27] - 5

Objects: backpack, person, backpack
Caption: startseq the man is jumping over the air in the air endseq



Figure 5.4.7 Flickr8K[19] Input Image for InceptionV3[27] - 6

Objects: dog
Caption: startseq the dog is running through the water endseq

Figure 5.4. Captions generated when using InceptionV3[27] as CNN on Flickr8K[19] Dataset



Figure 5.5.2 Flickr8K[19] Input Image for Bidirectional LSTM[35] - 1

Objects: person, snowboard, person, person
Caption: startseq the skier is jumping over the snow endseq



Figure 5.5.3 Flickr8K[19] Input Image for Bidirectional LSTM[35] - 2

Objects: person, motorbike
Caption: startseq the biker is riding bike on the dirt track endseq



Figure 5.5.4 Flickr8K[19] Input Image for Bidirectional LSTM[35] - 3

Objects: person, person
Caption: startseq the young boy is playing to the ball endseq



Figure 5.5.5 Flickr8K[19] Input Image for Bidirectional LSTM[35] - 4

Objects: person, surfboard
Caption: startseq young boy is playing in the water endseq



Figure 5.5.6 Flickr8K[19] Input Image for Bidirectional LSTM[35] - 5

Objects: backpack, person, backpack
Caption: startseq man in red shirt is jumping down the snow endseq



Figure 5.5.7 Flickr8K[19] Input Image for Bidirectional LSTM[35] - 6

Objects: person
Caption: startseq man in red shirt is standing on the snow endseq

Figure 5.5. Captions generated while using Bidirectional LSTM[35] on Flickr8K[19] Dataset

5.6 Observation of results

The above implementations depict results using VGG16[25] and InceptionV3[27] as CNN in image captioning. The results were conducted on the Flickr8K[19] Dataset with 1000 test images. The model was evaluated using BLEU[20] score with both the CNN variants. The resulting BLEU score of both VGG16[25] and InceptionV3[27] are 0.3692 and 0.3572 respectively. According to the training using those CNNs, it was observed that the InceptionV3 provides approximately similar results but with a greater number of epochs than that of VGG16[25]. While using VGG16[25] model, the model generated required 7 epochs while the InceptionV3[27] requires 12 epochs to get similar results. The use of Bidirectional LSTM[35] resulted in a BLEU score of 0.333 after 12 epochs.

Chapter 6

Summary

Image Captioning is the procedure that enables machine to interpret image on form of text. As discussed in the thesis, an Image Captioning problem allows the use of two advanced technologies viz. Natural Language Processing and Computer Vision for understanding word sequences and image respectively. The basic task is to detect objects and interpret the image data. Initially, this problem was addressed by Traditional Machine Learning methods but with the advancement in research and technology, Deep Learning methods achieved popularity[3]. The presented thesis work also provides the basic information about the availability of datasets, evaluation parameters and the brief discussion of the literature. The Object Detection is performed with the help of YOLO[21] method while the features are extracted using the VGG16[25] and InceptionV3[27]. The experiment is analyzed by both VGG16[25] and InceptionV3[27] as CNN for interpreting image data. The model implemented in this report is a merge architecture where objects and the image feature are merged to pass them to the LSTM for generating description. From the experiment, it was observed that the VGG16 provides better results compare to InceptionV3. Along with providing better results, VGG16 requires few numbers of epochs while, InceptionV3 requires a greater number of epochs for achieving the similar result. Summarizing altogether, Object Detection Based Image Captioning is an automation for human as well as machine, delivering a description for an image. The illustrated experiment is implemented on the Flickr8K[19] Dataset.

REFERENCES

- [1] J. Brownlee, “How to Develop a Deep Learning Photo Caption Generator from Scratch,” *Machine Learning Mastery*, Jun. 26, 2019. <https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/> (accessed Jul. 09, 2020).
- [2] “[1707.07102] OBJ2TEXT: Generating Visually Descriptive Language from Object Layouts.” <https://arxiv.org/abs/1707.07102> (accessed Jul. 20, 2020).
- [3] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, “A Comprehensive Survey of Deep Learning for Image Captioning,” *ArXiv181004020 Cs Stat*, Oct. 2018, Accessed: Jul. 20, 2020. [Online]. Available: <http://arxiv.org/abs/1810.04020>.
- [4] G. Nishad, “Automatic Image Captioning : Building an image-caption generator from scratch !,” *Medium*, Mar. 12, 2019. <https://blog.goodaudience.com/automatic-image-captioning-building-an-image-caption-generator-from-scratch-4bdd8744bc38> (accessed Jul. 20, 2020).
- [5] “deep learning - What’s the commercial usage of ‘image captioning’?,” *Artificial Intelligence Stack Exchange*. <https://ai.stackexchange.com/questions/10114/whats-the-commercial-usage-of-image-captioning> (accessed Jul. 20, 2020).
- [6] D. Hutchison *et al.*, “Every Picture Tells a Story: Generating Sentences from Images,” in *Computer Vision – ECCV 2010*, vol. 6314, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–29.
- [7] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, “Composing Simple Image Descriptions using Web-scale N-grams,” p. 9.
- [8] M. Hodosh, P. Young, and J. Hockenmaier, “Framing Image Description as a Ranking Task Data, Models and Evaluation Metrics Extended Abstract,” p. 5.
- [9] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi, “Collective Generation of Natural Image Descriptions,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jeju Island, Korea, Jul. 2012, pp. 359–368, Accessed: Dec. 24, 2019. [Online]. Available: <https://www.aclweb.org/anthology/P12-1038>.
- [10] N. K. Kumar, D. Vigneswari, A. Mohan, K. Laxman, and J. Yuvaraj, “Detection and Recognition of Objects in Image Caption Generator System: A Deep Learning Approach,” in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, Mar. 2019, pp. 107–109, doi: 10.1109/ICACCS.2019.8728516.
- [11] C. Amritkar and V. Jabade, “Image Caption Generation Using Deep Learning Technique,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Aug. 2018, pp. 1–4, doi: 10.1109/ICCUBEA.2018.8697360.
- [12] P. Shah, V. Bakrola, and S. Pati, “Image captioning using deep neural architectures,” in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Mar. 2017, pp. 1–4, doi: 10.1109/ICIIECS.2017.8276124.
- [13] F. Fang, H. Wang, and P. Tang, “Image Captioning with Word Level Attention,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 1278–1282, doi:

10.1109/ICIP.2018.8451558.

- [14] D.-J. Kim, D. Yoo, B. Sim, and I. S. Kweon, "Sentence learning on deep convolutional networks for image Caption Generation," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Aug. 2016, pp. 246–247, doi: 10.1109/URAI.2016.7625747.
- [15] A. Poghosyan and H. Sarukhanyan, "Short-term memory with read-only unit in neural image caption generator," in *2017 Computer Science and Information Technologies (CSIT)*, Sep. 2017, pp. 162–167, doi: 10.1109/CSITechnol.2017.8312163.
- [16] M. Tanti, A. Gatt, and K. P. Camilleri, "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?," *ArXiv170802043 Cs*, Aug. 2017, Accessed: Jul. 20, 2020. [Online]. Available: <http://arxiv.org/abs/1708.02043>.
- [17] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning," *ArXiv161201887 Cs*, Jun. 2017, Accessed: Jul. 20, 2020. [Online]. Available: <http://arxiv.org/abs/1612.01887>.
- [18] M. Nguyen, "Illustrated Guide to LSTM's and GRU's: A step by step explanation," *Medium*, Jul. 10, 2019. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> (accessed Jan. 01, 2020).
- [19] "Flickr8K." <https://kaggle.com/shadabhussain/flickr8k> (accessed Nov. 25, 2019).
- [20] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," 2002, pp. 311–318.
- [21] J. Hui, "Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3," *Medium*, Aug. 27, 2019. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088 (accessed Jul. 20, 2020).
- [22] "Yolo Framework | Object Detection Using Yolo," *Analytics Vidhya*, Dec. 06, 2018. <https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framework-python/> (accessed Jul. 20, 2020).
- [23] "Understanding object detection in deep learning - The SAS Data Science Blog." <https://blogs.sas.com/content/subconsciousmusings/2018/11/19/understanding-object-detection-in-deep-learning/> (accessed Jul. 20, 2020).
- [24] Prabhu, "Understanding of Convolutional Neural Network (CNN) — Deep Learning," *Medium*, Nov. 21, 2019. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> (accessed Jul. 21, 2020).
- [25] R. Thakur, "Step by step VGG16 implementation in Keras for beginners," *Medium*, Aug. 20, 2019. <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c> (accessed Jul. 09, 2020).
- [26] "VGG16 - Convolutional Network for Classification and Detection." <https://neurohive.io/en/popular-networks/vgg16/> (accessed Jul. 20, 2020).
- [27] "Transfer Learning with InceptionV3." <https://kaggle.com/kmader/transfer-learning-with-inceptionv3> (accessed Jul. 20, 2020).

- [28] K. Team, “Keras documentation: Text data preprocessing.” <https://keras.io/api/preprocessing/text/> (accessed Jun. 12, 2020).
- [29] A. Mittal, “Understanding RNN and LSTM,” *Medium*, Oct. 12, 2019. <https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e> (accessed Apr. 08, 2020).
- [30] S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” p. 8.
- [31] R. Vedantam, C. L. Zitnick, and D. Parikh, “CIDEr: Consensus-based Image Description Evaluation,” *ArXiv14115726 Cs*, Jun. 2015, Accessed: Jul. 20, 2020. [Online]. Available: <http://arxiv.org/abs/1411.5726>.
- [32] “Image captioning.” <https://kaggle.com/hsankesara/image-captioning> (accessed Nov. 25, 2019).
- [33] “COCO - Common Objects in Context.” <http://cocodataset.org/#home> (accessed Nov. 25, 2019).
- [34] “YOLO object detection using Opencv with Python,” *Pysource*, Jun. 27, 2019. <https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/> (accessed Jul. 21, 2020).
- [35] “tf.keras.layers.Bidirectional | TensorFlow Core v2.2.0,” *TensorFlow*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Bidirectional (accessed Jul. 20, 2020).

APPENDIX A

ABBREVIATION

VGG16	Visual Geometry Group with 16 layers
LSTM	Long Short Term Memory
SIFT	Scale-Invariant Feature Transform
HOG	Histogram of Oriented Gradient
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
YOLO	You Only Look Once
RODe	Regional Object Detector
gLSTM	Guided Long Short Term Memory
RCNN	Region Based Convolutional Neural Network
L2BE	Line Level Bidirectional Embedding
WLAE	Word Level Attention Extraction
GRU	Gated Recurrent Unit
MSCOCO	Microsoft Common Objects in Context
BLEU	Bilingual Evaluation Understudy
METEOR	Metric for Evaluation of Translation with explicit ORdering
CIDEr	Consensus-Based Image Description Evaluation

Table A.1. List of Abbreviations

APPENDIX B

REVIEWS

BIRLA VISHVAKARMA MAHAVIDYALAYA
(An Autonomous Institute)
M. Tech. Computer Engineering (Software Engineering)
Dissertation-II Internal Review card

Semester: 4th AY: 2019-2020

ID No: 18CP809 Name of the Student: Trushna Patel

Title of the Dissertation: Object Detection based Image Captioning using Deep Learning

Dr. U.K. Talreja Prof. K.J. Sharma Dr. D.G. Thakore Dr. N.M. Patel
Name of Convener Name of Member Name of Guide Name of Guide

Mid Semester Review-1		Date: <u>19/12/2020</u>
No.	Comments given by DPC Members	Modification done based on Comments
1	Do more analysis based on probability.	
2	Try to use RNN for caption generation.	

(Convener Sign) (Member Sign) (Guide Sign) (Guide Sign)

Mid Semester Review-2		Date: _____
No.	Comments given by DPC Members	Modification done based on Comments













(Convener Sign) (Member Sign) (Guide Sign) (Guide Sign)

Figure B.1. Review

Document Information














Analyzed document	18CP809.pdf (D76692390)
Submitted	7/21/2020 9:08:00 AM
Submitted by	BVM Engineering College
Submitter email	mec008owner@gtu.edu.in
Similarity	9%
Analysis address	mec008owner.gtu@analysis.arkund.com

Sources included in the report

W	URL: https://www.ijmtst.com/vol6issue04.html Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://oayman1.wordpress.com/2019/07/19/image-captioning-survey/ Fetched: 7/21/2020 9:10:00 AM		2
W	URL: https://www.groundai.com/project/a-comprehensive-study-of-deep-learning-for-image- ... Fetched: 7/21/2020 9:10:00 AM		7
W	URL: https://arxiv.org/abs/1801.05568 Fetched: 7/21/2020 9:10:00 AM		2
W	URL: https://www.researchgate.net/publication/330478855_Image_Caption_Generator_with_No ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://res.mdpi.com/d_attachment/information/information-10-00354/article_deploy/ ... Fetched: 2/8/2020 2:37:08 AM		2
W	URL: https://www.researchgate.net/publication/318981488_What_is_the_Role_of_Recurrent_N ... Fetched: 2/8/2020 2:46:59 PM		3
W	URL: https://github.com/zhjohnchan/awesome-image-captioning Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://www.researchgate.net/figure/Taxonomy-for-English-image-captioning-approach ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://www.researchgate.net/publication/307747289_Show_and_tell_A_neural_image_ca ... Fetched: 12/2/2019 7:08:22 AM		1
W	URL: https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-mode ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-d ... Fetched: 7/21/2020 9:10:00 AM		1

URL: <https://mc.ai/automatic-image-captioning-building-an-image-caption-generator-from-...>



W	URL: https://medium.com/@hamzajg16/image-captioning-c835feb6026f Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://medium.com/@hamzajg16/image-captioning-c835feb6026f Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://ai.stackexchange.com/questions/10114/whats-the-commercial-usage-of-image-c ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://medium.com/mlreview/multi-modal-methods-image-captioning-from-translation- ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://openaccess.thecvf.com/content_cvpr_2016/papers/You_Image_Captioning_With_C ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://www.researchgate.net/publication/337063523_ImageToText_Image_Caption_Gener ... Fetched: 11/27/2019 6:15:25 AM		2
W	URL: https://lrec2020.lrec-conf.org/media/proceedings/Workshops/Books/SLTUCCURLbook.pdf Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://bib.irb.hr/datoteka/1005660.Deep_Image_Captioning_MIPRO2019_Final.pdf Fetched: 7/21/2020 9:10:00 AM		2
W	URL: https://www.slideshare.net/mz0502244226/image-captioning Fetched: 7/21/2020 9:10:00 AM		2
W	URL: https://www.ijedr.org/papers/IJEDR1804011.pdf Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://blogs.sas.com/content/subconsciousmusings/2018/11/19/understanding-object- ... Fetched: 7/21/2020 9:10:00 AM		1
W	URL: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn ... Fetched: 7/21/2020 9:10:00 AM		1