A Dissertation Report

On

"Caption Generation from Image"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

MASTER OF ENGINEERING (COMPUTER ENGINEERING)

Submitted By

Palwe Aditi Vijaykumar
Exam No:

Under the guidance of

Dr. Sankirti Shiravale

'येथे बहुतांचे हित'

DEPARTMENT OF COMPUTER ENGINEERING

Marathwada Mitra Mandal's

College of Engineering, Pune

Accredited with 'A' grade by NAAC, Recipient of "Best College Award 2019" by SPPU,

Accredited Mechanical and Electrical Departments by NBA

Savitribai Phule Pune University

DEPARTMENT OF COMPUTER ENGINEERING

Marathwada Mitra Mandal's
COLLEGE OF ENGINEERING, PUNE

Accredited with 'A' grade by NAAC, Recipient of "Best College Award 2019" by SPPU,
Accredited Mechanical and Electrical Departments by NBA 2020-2021

CERTIFICATE

This is to certify that the Dissertation entitled

"Caption Generation from Image"

Submitted by Ms. Aditi Vijaykumar Palwe Exam No:

is a bonafide work carried out by her under the supervision of Dr. Sankirti Shiravale and it is submitted towards the fulfilment of the requirement of Savitribai Phule, Pune University for the award of the degree of Master of Engineering .

Dr. Sankirti Shiravale
Internal Guide

Dr. K.S Thakre
Head of Department
(Computer Engineering)

Dr. V.N Gohakar
Principal, Marathwada Mitra Mandal's
COLLEGE OF ENGINEERING
Karvenagar Pune-52.

Seal of the College

Date:

Place: Marathwada Mitra Mandal's, College of Engineering, Pune

Examination Approval Sheet

CERTIFICATE

This is to certify that the project report entitled

“Caption Generation from Image ”

Submitted by,

Name: Ms. Aditi Vijaykumar Palwe Exam No.:

is approved for the degree of Master of Engineering (Computer Engineering) in Savitribai Phule Pune University, Pune at Marathwada Mitra Mandals, College of Engineering, Pune.

Internal Examiner: **External Examiner:**

Seal of the College:

Date:

Place: Marathwada Mitra Mandal’s, College of Engineering, Pune

Certificate by Guide

This is to certify that Ms. Aditi Vijaykumar Palwe has completed the dissertation work under my guidance and supervision and that, I have verified the work for its originality in documentation, problem statement, implementation and results presented in the dissertation. Any reproduction of other necessary work is with the prior permission and has given due ownership and included in the references.

Place : MMCOE, Pune-52
Date :

Dr. Sankirti Shiravale
Internal Guide
Department of Computer Engineering
MMCOE, PUNE-52

Acknowledgement

First of all, I would like to express my deep sense of gratitude to my Project guide **Dr. Sankirti Shiravale,** Computer Engineering Department, MMCOE, Pune for her guidance, inspiration and constructive suggestion that helpful for me to prepare this Dissertation.

I would like to express grateful thanks to **Dr. K. S. Thakre** Head, Computer Engineering Department, MMCOE, Pune and **Dr. Sankirti Shiravale** and **Dr. Swati Shekhapure**, ME-Coordinator, Computer Engineering Department, Pune for their constant support and supervision whenever required during my Project work.

I express my sincere thanks to **Dr. V. N. Gohokar** , MMCOE, Pune, for encouragement and creating healthy environments for all of us to learn innovative things.

I am also grateful to all staff members of the Computer Engineering Department who helped me directly or indirectly during this course of work. And last thanks to my family members for their unwavering encouragement. I also thank all my friends for being a constant source of support.

Ms. Aditi Vijaykumar Palwe,
Exam Seat No.:
Department of Computer Engineering,
MMCOE, PUNE

Contents

List of Figures

List of Tables

List Of Publications

List of Conferences:

1. Aditi Palwe*, Sankirti Shiravale, “Image Caption Generation using CNN and RNN Models” CPGCON 2021, May 2021.

List of Journals:

1. Aditi Palwe*, Sankirti Shiravale & Swati Shekapure. (2022). Image Captioning using Efficient Net. Journal of Optoelectronics Laser, 41(7), 1259–1270. July 2022.

Abstract

Image Captioning intends to produce a sound and thorough portrayal that sums up the contents of a picture. The description generator uses the encoder decoder deep learning model that elaborates the image in text format. Describing the details of an image is one of the challenging and explored areas of Artificial Intelligence. Traditional approaches cannot handle the complexity and difficulties of image captioning as well as deep learning-based approaches. In this paper we proposed EfficientNetB3 deep learning framework for caption generation on complex objects. First, we go through several current image captioning approaches, with an emphasis on deep-learning-based approaches and how they are used for pre-processing. Then CNN module implementation for feature extraction and GRU has been used for generating the caption for respective images. The various cosine similarity and N-Gram feature extraction techniques have been used for generating the blue score for entire testing dataset and show the effectiveness of proposed system.

Key terms*:* *Convolutional Neural Network, GRU-Gated Recurrent Unit, image caption generation, Natural Language processing.*

Synopsis

Dissertation Title: Caption Generation from Image .

Student Name: Aditi Vijaykumar Palwe

Exam No:

Project Guide: Dr. Sankirti Shiravale.

Branch: Computer Engineering [Master of Engineering (ME)]

College Name: Marathwada Mitra Mandal's College of Engineering, Pune

Domain: Machine learning

Keywords: Convolutional Neural Network, GRU-Gated Recurrent Unit, image caption generation, Natural Language processing

Abstract : Convolutional neural-nets i.e., CNN are usually pre trained on a particular picture dataset, which uproots visual attributes from those pictures. This helps the defined model to learn on new data in a much swift way. Elaborating an image in a text format with genuine grammatical syntax is still a demanding problem in the field of computer vision and natural language processing. Comprehending a scene, which combines the expertise of computer vision with NLP, involves image caption, which automatically generates ethical English language descriptions based on the information seen in a frame.

Privacy; Multi-Keyword; Query; Public Key Encryption; Access Control, Information Retrieval.

Using natural languages to automatically describe the content of pictures has a lot of promise. It may, for example, aid visually challenged individuals in comprehending the content of online pictures. It may also offer more accurate and concise images information in situations likes social media image sharing or video surveillance systems. The technique produces image captions that are typically semantically informative and grammatically accurate by acquiring information from image and caption pairings. Natural languages are used by humans to describe

scenes. Machine vision systems, on the other hand, characterize the scene by capturing an image that is a two-dimensional array. The concept is to combine the image and captions into one area and then learn a mapping from the image to the words.

Captioning images is a long process if done by people physically, it also contributes errors generated by humans which is very significant if the generated description is used in sensitive applications like medical or space research to name few. From computer vision point of view the content within the images is very much valuable. They help a machine understand and perform accordingly. Image captioning has various applications such as recommendations in editing applications, usage in virtual assistants, for image indexing, for visually impaired persons, for social media, and several other natural language processing applications. Through various ways of deep learning solution to the process of image annotation has been achieved at a greater level and has contributed to various fields. Through deep learning people have come up with various innovative ideas to tackle this application. From those results it has been demonstrated that deep learning models are able to achieve optimum results in the field of caption generation problems. To generate quality level image descriptions, it is important to understand not only what the objects within the image are but also the relationship that exists between them. Today the encoder – decoder models are considered one of the state-of-the-art models for image captioning [4].

Objectives

- We develop a model, for generating stylistically interesting and semantically relevant image captions by learning from a large corpus of stylized text without aligned images.
- To design and develop a hybrid deep learning algorithms to generate effective captions using combination of CNN with EfficientNetB3 and Attention Model.
- To generate the effective captions using GRU and greedy approach from module testing and validate with ground truth captions to validate the algorithms accuracy.
- To explore and validate the efficiency of proposed model with various existing systems and show the effectiveness in real time environment

Motivation

- Various systems has used reinforcement learning based captioning generation frameworks.
- Numerous ensembles learning system also democrats the caption generation using supervised learning approach.
- Up-Down [2], Ensemble [3], CNN-GAN and RNN-GAN [1] has used in various existing systems which illustrates low accuracy.

Goals

- Image caption generation for random objects.
- Implement a system on Flicker 8k-9k or MSCOCO dataset
- Implementation of EfficientNetB3 model for feature extraction and attention model and GAN for caption generation.
- Implement a NLP techniques for text processing
- Apply greedy approach for final captioning

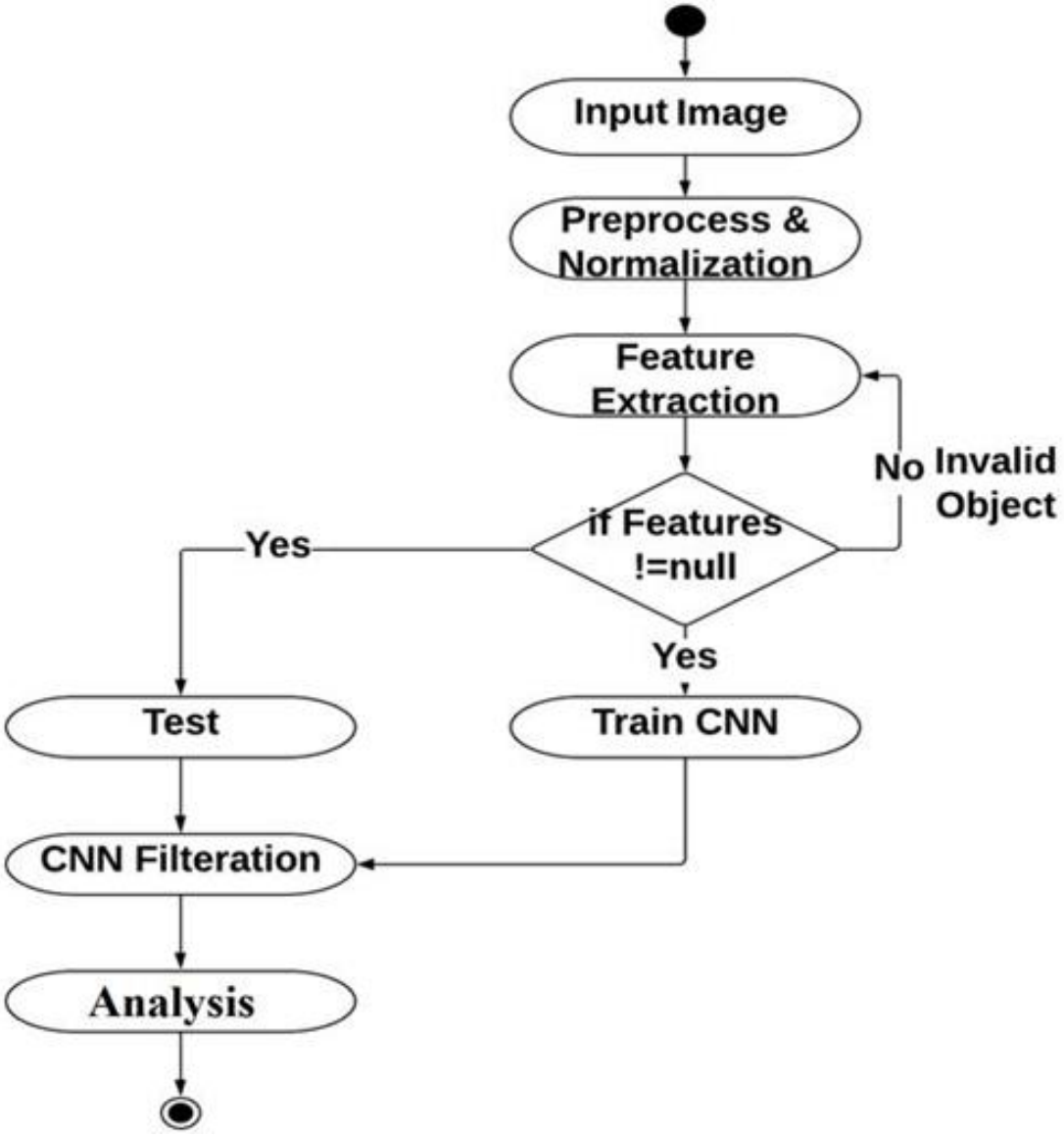
Project Scope

Deep learning methods have demonstrated advanced results on caption generation problems. What is most impressive about these methods is that one end-to-end model is often defined to predict a caption, given a photograph, rather than requiring sophisticated data preparation or a pipeline of specifically designed models. Deep learning has attracted a lot of attention because it's particularly good at a kind of learning that has the potential to be very useful for real-world applications. The ability to find out from unlabeled or unstructured data is a huge benefit for those curious about real-world applications. This system most effectives in below areas.

- Generate caption for random image and display the information
- Video captioning and detect the objects in entire video

- Provide automatic supportive tagging or captioning for image and video for social media posts.

Flow chart of System



Implementation of System

Preprocessing and Normalization: The data collection has done from synthetic repository called MSCOCOC. The data might be imbalance with dimension thus it needs to balance before proceeding to CNN. In preprocessing we define fix set of dimensions of image (300*300) and eliminate those objects which are not readable for misclassified. The cross validation has also

done in this similar phase for training and testing. (e.g., 5-fold, 10 fold, 15 fold etc.). In the dataset each image having caption as ground truth caption, and that used for analysis. Due to the large size of the dataset, data processing was done on the host computer and the results were saved as pickle files before being sent to the cloud for model training.

Lower case conversion: The dataset texts include words with various letter cases, which presents a challenge for the model since the same words with different capitalizations would be treated differently, resulting in an increase in problem vocabulary and, as a result, complexity. As a result, to prevent this, the whole text must be converted to lower case.

Punctuation removal: Because the goal of this study is to create descriptive sentences for pictures without using punctuation, the existence of punctuation adds complexity to the issue, which is beyond the scope of this study.

Number removal: The presence of numerical data in the texts is a challenge to the model since it enhances language; thus, it should be eliminated.

Indicate start and end sequence: To signal the beginning and finish of the prediction sequence to the model, the word tokens <start> and<end> are inserted at the beginning and end of each phrase.

Tokenization: The clean text is broken down into component terms, and a dictionary with the full vocabulary is produced for word to index and index to word matching.

Vectorisation: Before learning word indicative vectors from these sequencing, the words in the text data are transcribed using unique numerical interpretations from the word to index dictionary, transforming the cleaned sentence specifications to numerical sequences. Lesser sentences are stretched to the length of the largest sentence sequence to compensate for variable sentence lengths.

CNN using EfficientNetB3: The EfficientNetB3 module has used for CNN, to extract the features and selection as well. The below code snippet demonstrates the building of CNN with EfficientNetB3. The global image feature vectors for the frames are retrieved from Keras applications using the final fully connected layer of the EfficientNetB3 CNN. Before extraction, the pictures are compressed to 300x300 arrays, and the pixels are normalized to a scale of 0 to

255 to suit the EfficientNetB3 model input. A pretrained model is utilized for the picture scene features. To match the input shape of the model, the images are compressed to 224x224 and normalized on a scale of 0 to 255. Both networks output feature vectors are saved as pickle files for simple uploading for validation testing.

Convolution, batch normalization, and ReLU functions are combined in the function D. Convolution is the multiplication of image and kernel matrices element by element. In each mini-batch, batch normalization involves moving inputs with a zero mean and unit variance. The ReLU function is then used to transform the elements with negative values to zero.

Attention Model and GRU: attention model basically work for selection of features that received from CNN. The selection features eliminate the redundant and non-essential features. The attention model has divided into four different phases these are describes in below,

Encoder: The pre-trained Inception model has already done the visual encoding; thus the encoder is extremely straightforward. It has a Linear layer that receives the pre-encoded gives the view and transmits them to the Decoder.

Sequence Decoder: This is a GRU-based recurrent network. Once passing through a Hidden layer, the captions are sent in as input.

Attention: The Attention module assists the Decoder in focusing on the most important portion of the picture for producing each word in the output sequence.

Sentence Generator: This module is made up of a few Linear layers. It takes the Decoder's output and generates a frequency for each word in the lexicon, as well as for each place in the anticipated sequence.

Literature Survey

No	Technique	Dataset	Extracted Features	Research Gap
1	x-Means clustering algorithms and Neuro Fuzzy algorithm [1]	GSM operation data, 24,900 customers 22 attributes Turkey dataset	Some value-added services and some values added services	System reflects good accuracy on structured dataset only.
2	Naïve Bayes, Decision Tree[2]	European operator 106,405 customers 112 attributes	Contract, usage pattern patterns, and calls pattern	High error rate to detect actual churn due to redundant features.
3	Neural network, Regression [3]	Unknown 129,892 customers 113 attributes	Demographic, Value added, usage pattern	Heterogeneous dataset tedious to handle in similar patterns environments.
4	Neural network, Regression [4]	Unknown, 169 customers 10 attributes	Demographic, Billing data, usage pattern, customer relationship	High space complexity generates in each layer
5	Stepwise variable selection partial least squares [5]	Cell2Cell Dataset 100,000 customers 171 attributes	Behavioral information, Customer care and demographics	Redundant features should be generating high error rate.
6	Artificial Neural Network [6]	ML Dataset of UCI 2,427 user's information with 20 attributes	Demographics, Usage pattern, Value added services	It works only define statically parameters.
7	Binomial logistic	Iranian telco	Demographic, call	Language influence

	regression model [7]	operator 3150 customers 15 attributes	usage pattern, customer care service	should be generate irrelevant features vector.
8	Generalized additive models (GAM) [8]	Belgian 134, 120 customers 27 attributes	Demographic Usage patter, bill and payment	High error rate during unknown text prediction.
9	Logistic regression Decision tree [9]	Polish mobile operator 122098 customers 1381 attributes	Demographic, call data records, customer care services	Its works only synthetic data only and high data reduction rate.
10	Decision tree as well as machine learning [10] algorithms has used.	Cell2Cell Dataset 100,000 customers 171 attributes	Behavioral data, of customer care and feature information	Behaviors information generate the churn possibility sometime it generate false ratio.

Table 1.1 Literature Survey

Technical Keywords

- Convolutional Neural Network,
- GRU-Gated Recurrent Unit
- image caption generation
- Natural Language processing.

Chapter 1

INTRODUCTION

1.1 Introduction

Convolutional neural nets i.e., CNN are usually pre trained on a particular picture dataset, which uproots visual attributes from those pictures. This helps the defined model to learn on new data in a much swift way. Elaborating an image in a text format with genuine grammatical syntax is still a demanding problem in the field of computer vision and natural language processing. Comprehending a scene, which combines the expertise of computer vision with NLP, involves image caption, which automatically generates ethical English language descriptions based on the information seen in a frame.

Using natural languages to automatically describe the content of pictures has a lot of promise. It may, for example, aid visually challenged individuals in comprehending the content of online pictures. It may also offer more accurate and concise images information in situations likes social media image sharing or video surveillance systems. The technique produces image captions that are typically semantically informative and grammatically accurate by acquiring information from image and caption pairings. Natural languages are used by humans to describe scenes. Machine vision systems, on the other hand, characterize the scene by capturing an image that is a two-dimensional array. The concept is to combine the image and captions into one area and then learn a mapping from the image to the words.

Captioning images is a long process if done by people physically, it also contributes errors generated by humans which is very significant if the generated description is used in sensitive applications like medical or space research to name few. From computer vision point of view the content within the images is very much valuable. They help a machine understand and perform accordingly. Image captioning has various applications such as recommendations in editing applications, usage in virtual assistants, for image indexing, for visually impaired persons, for social media, and several other natural language processing applications. Through various ways of deep learning solution to the process of image annotation has been achieved at a greater level and has contributed to various fields. Through deep learning people have come up with various

innovative ideas to tackle this application. From those results it has been demonstrated that deep learning models are able to achieve optimum results in the field of caption generation problems. To generate quality level image descriptions, it is important to understand not only what the objects within the image are but also the relationship that exists between them. Today the encoder – decoder models are considered one of the state-of-the-art models for image captioning [4].

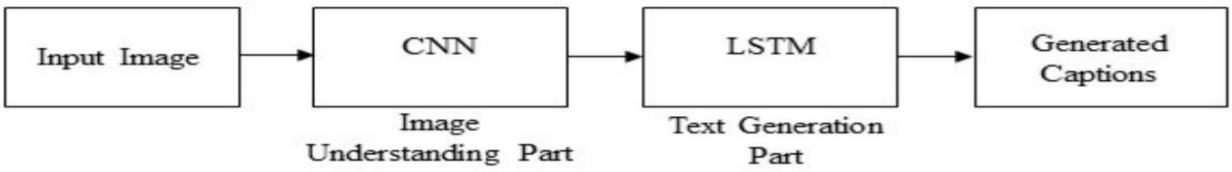


Figure 1.1 Encoder Decoder generalized architecture

In an object, a lot of information is saved. On social networking sites, including celestial objects, enormous image data can be generated each day, but this is an up with the fast thing. It takes time to annotate photographs of humanity, and the likelihood of error is greater. Deep learning models are used to accurately compile such images, thereby eliminating the manual corrections done.[16] This would significantly reduce human failure and also the efforts by removing any need for human involvement. The development of image annotations has numerous real benefits, varying from supporting the mentally challenged to helping the automated, cost-saving marking of images shared online every day, guidelines for processing software, useful for smart devices, image encoding, visually disabled people, social networking sites, and so many other natural literatures.

Models like Convolutional Neural Network have played an important role in this picture. Here, we try to show and highlight different methods for image feature extraction and how it will be used for caption creation. There has been a lot of study in the field of data science for advancing image caption generation model. Natural language processing plays an important role for creating a description, which is to the mark and has semantics to go with. Recurrent Neural Network (RNN) is the preferred network in description creation as RNN is basically used for sequence generation. Researchers have put in lot of effort in creating enormous magnitude datasets. Some of the famous datasets are the MSCOCO which is been provided by Microsoft. Other well-known and benchmark datasets are the Flickr 8K, Flickr 30K, PASCAL and some others [5].

Through this paper we have highlighted successful approaches that focus on deep learning to generate image descriptions. They have proved to be successful and have evolved with time. Evaluation of the overall model with the description generated is evaluated using evaluation metrics like the CIDEr, BLEU, METEOR, and other metrics.

This review begins as follows in Section 2 we have introduced the Image Captioning Methods and some recent methods of used for Image Captioning. In Section 3 we highlight different evaluation metrics that assess the performance of the designed model. Section 4 shows result on different ideas implemented through attention mechanism. Section 5 shows required Datasets and its related information. Also, we have focused on the software and hardware platforms needed for implementing the mode, which is followed by conclusion in the last 6th section.

1.2 Image Captioning Methods

The encoder part receives the image as input and generates a feature vector of high dimensions. The image goes through the layers of the Encoder model that converts the image to feature vector. The system does not understand anything but 0's and 1's and thus the image is converted to 0s and 1s. The image vector created is the form of binary format. The encoder is a group of convolutions that uses filters and uses pooling to help the system to focus on certain regions of the image. The encoded features from the encoder are given as input to the decoder. The decoder takes this high dimension features and generates a semantic segmentation mask. It is a process of recurrent units where it predicts an output at each stage. It accepts a hidden state and produces an output and its own hidden state. Various activation functions are used in the encoder-decoder model as per the application of the user.

Figure 1.2 System may output: There is laptop and a monitor on the desk

Encoders:

Convolutional and Recurrent networks are the head and tail of the encoder decoder model. The convolution neural network works on the two-dimensional image data and can be used in other dimensional data too.

- VGG: The model is used in Feature extraction for images. It has two versions viz. the VGG16 and VGG19, each of these models have 16 and 19 layers. These layers are stacked on top of each other. The architecture mainly consists of the convolution which than passes though the relu activation and then the pooling, at last flatten the vector value which is passed to the decoders. VGG faces issues with gradient decent as you go deeper in the model.

- ResNet: ResNets are like successors of VGG. They are deeper and better in their applications. ResNets have majorly these versions viz. ResNet 34,50,101,152. ResNets have proven to be useful and overcome the issue of vanishing gradient. The technique of skip connection is convenient it allows an alternate path for the gradient that helps the network from the gradient gradually approaching a very small value. Skip connections in deep architectures, in the computer program, skip a level and feed the performance about one layer also as layer known.

- Inception: The framework of Inception is planned in such a manner that it functions well, even when under strict machine and storage time restraints. It uses a significantly lower set of variables and gives our text recognition tasks a recommended efficiency as seen in the outcomes [5]

Decoders:

Recurrent neural networks like the Long Short-Term Memory (LSTM) and GRU (Gated Recurrent Unit) are used to decode the vectors from the encoder. Pattern generation is implemented through the LSTM and GRU models. They have been widely used in speech recognition, NLP, and other areas.

- LSTM: From the point of generating cinematic intervals, the Long Short Term Memory model is developed, consisting of an internal mechanism called gates that regulates the

information flow. The gates decide what data to hold and to discard, which provides the benefit of passing the necessary data to predict the sequence chain. This processes information that passes on data as it transmits forward. It comprised of three windows, namely the gateway of input, gate of output and gateway of forget.

- GRU: The GRU is the latest Recurrent Neural Networks technology and is very close to the LSTM. The Gated Recurrent Device has got rid of both the cell state and is transmitting information using the secret state. There are also actually two ports, an include some and an upgrade entrance. These gates control the flow of information within the model. These gates determine what data to maintain which information to discard.

1.3 Attention Model

Attention models are focused on two areas viz, soft attention and the hard attention. Both these models have their pros. In case of soft attention, the entire image is considered but certain areas are more focused or attended to more. Whereas the hard attention covers a subset of the image and focuses that part.[2] This model requires comparatively less memory and computation power as it covers a subset of the image, whereas the soft attention model requires more memory and computational power. As the decoder will use the previous hidden state, previously generated word and a context vector. The attention model uses the context vector which is formulated by the convolution neural network. The weights calculated using the attention function can be done using the soft or hard attention mechanism. Attention can be broadly interpreted as a vector of importance weights. The visualization of the attention weights clearly demonstrates which regions of the image the model is paying attention to so as to output a certain word. It eliminates the vanishing gradient problem, as they provide direct connections between the encoder states and the decoder.

1.4 Evaluation Metrics

Bleu

Bilingual Evaluation Understudy Score or BLEU is proposed by Kishore et al[3] BLEU is used to evaluate how close is the generated text as compared to the expected text. It is the most well-known measurement for assessing a produced sentence to a reference sentence based on n-grams

where n ranges from 1 to 4. Its value ranges from 0 to 1 where 0 refers to the completely mismatch and vice-a-versa. The text generated is referenced with the dataset text.

ROGUE

It is one of the methods for the Actually remember Interim manager for Gisting Examination to enhance the accuracy of produced text. By compared the summary produced by the system to an outline four, it assesses. Precision and Recall is measured to set a good significance level.

There really are countless distinctive rouge-specific tools available such as ROUGE-1,2, ROUGE-W, ROUGESU. ROUGE-SU and ROUGH-2 evaluation provides better results for small synopses and ROUGH-1 and ROUGE-W are better for multi - document evaluation. Calculation of multi-document text classification is also limited.

METEOR

The GRU is the latest Recurrent Neural Networks technology and is very close to the LSTM. The Recurrent Neural Device has did get rid of both the convolution layer and is transmitting information using the secret state. There are also actually two ports, an include some and an upgrade entrance.

CIDEr:

Consensus-based Image Description Evaluation is one of the metrics that was developed specially for image and video description. It qualifies a target examination of machine produced approach dependent on their human-resemblance, without settling on grammar, saliency, etc. irrespective of each other.

1.5 Motivation

- Various systems has used reinforcement learning based captioning generation frameworks.
- Numerous ensembles learning system also democrats the caption generation using supervised learning approach.

- Up-Down [2], Ensemble [3], CNN-GAN and RNN-GAN [1] has used in various existing systems which illustrates low accuracy.

1.6 Goals

- Image caption generation for random objects.
- Implement a system on Flicker 8k-9k or MSCOCO dataset
- Implementation of EfficientNetB3 model for feature extraction and attention model and GAN for caption generation.
- Implement a NLP techniques for text processing
- Apply greedy approach for final captioning

Chapter 2

LITERATURE SURVEY

2.1 Literature Survey

Retrieval-based image captioning is one of the most popular techniques used in early work. Retrieval-based techniques generate a caption for a query image by obtaining one or more sentences from a pre-specified sentence pool. The produced caption may be a statement that already exists or one that is made out of the recovered sentences. Let's start by looking at the line of study that utilises recovered phrases as image captions.

Convolutional Neural Network models, according to [1] have played a significant influence in this image. Here, we attempt to demonstrate and highlight several techniques for image feature extraction, as well as how they will be used to caption generation. In the area of data science, there has been a lot of research towards improving image caption generating models. Natural language processing is crucial in producing a description that is accurate and has meaning. The favored network in description construction is the recurrent neural network (RNN), which is mostly utilized for sequence generation. Researchers have put in a lot of time and effort to create massive databases. The MSCOCO dataset, which is supplied by Microsoft, is one of the most well-known datasets. The Flickr 8K, Flickr 30K, PASCAL, and a few more are additional well-known and benchmark datasets.

Captioning pictures, according to [2] is the act of creating descriptive information about visual objects, image metadata, or things that exist in a picture. The material inside the pictures is very useful from the perspective of computer vision. They aid a machine's comprehension and performance. Image captioning has a variety of uses, including editing software suggestions, virtual assistants, image indexing, accessibility for visually impaired people, social networking, and a variety of other natural language processing applications. The process of image annotation has been accomplished at a higher level and has contributed to different areas via different methods of deep learning solution. People have come up with a variety of creative ways to approach this application using deep learning. It has been shown that deep learning models are capable of achieving optimal outcomes in the area of caption generating issues based on these

findings. It's critical to comprehend not just what the items in the image are, but also how they relate to one another, in order to produce high-quality image descriptions.

Encoder–decoder models are now regarded one of the most advanced image captioning methods. A lot of information is stored in an item. Huge amounts of image data may be produced each day on social networking sites, including astronomical objects, but this is an up with the quick thing. Annotating pictures of people takes longer, and the chances of making a mistake are higher. Deep learning models are utilized to construct such pictures correctly, removing the need for human adjustments [3]. By eliminating the requirement for human participation, this would substantially decrease human failure and effort. The development of image annotations has numerous real-world benefits, ranging from assisting the mentally challenged to assisting the automated, cost-effective marking of images shared online every day, guidelines for processing software, useful for smart devices, image encoding, visually disabled people, social networking sites, and a variety of other natural literature.

Mason and Charniak utilize visual similarity to obtain a collection of captioned pictures for a query image [4] to mitigate the effects of noisy visual estimates in techniques that rely on image retrieval for image captioning. They then estimate a word probability density conditioned on the query image using the captions of the retrieved pictures. The term probability density is used to assess existing captions in order to choose the one with the highest score as the query's caption. This technique has implicitly assumed that there is always a phrase that is relevant to a query picture. In reality, this assumption is seldom accurate. Instead of directly utilizing returned sentences as descriptions of query pictures, recovered sentences are used to construct a new description for a query image in another line of retrieval-based research.

Li et al. utilize visual models to extract semantic information from pictures, including objects, characteristics, and spatial connections [5]. Then, for encoding recognition results, they construct a triplet of the type adj1, obj1, prep, adj2, obj2. To provide a description for the triplet, web-scale n-gram data is used to conduct phrase selection, which may give frequency counts of potential n-gram sequences. This allows for the collection of potential phrases that may make up the triplet. Following that, phrase fusion is used to utilize dynamic programming to discover the most suitable collection of phrases to serve as the query image's description.

Mitchell et al. use computer vision algorithms to analyze and describe images using triplets of objects, activities, and spatial connections [6]. Then, based on the visual recognition findings, they construct image description as a tree-generating process. The authors select the image contents to describe by grouping and ranking object nouns. Then, for object nouns, sub-trees are constructed, which are then utilized to construct complete trees. Finally, a trigram language model is utilized to choose a string from the complete trees produced as the image's description.

Template-based image captioning may provide syntactically accurate sentences, and the descriptions produced by these techniques are typically more appropriate to the image contents than retrieval-based descriptions. Template-based approaches, on the other hand, have certain drawbacks. Because description creation in the template-based architecture is tightly limited to image contents identified by visual models, there are generally limits on coverage, originality, and complexity of produced sentences due to the generally small number of visual models accessible. Furthermore, as compared to human-written captions, utilizing strict templates like sentence core structures would make produced descriptions seem less natural. Another kind of technique that is frequently employed in early image captioning work is template-based. Image captions are produced using template-based techniques using a syntactically and semantically restricted approach. In order to produce a description for an image using a template-based approach, a collection of visual ideas must typically be identified first. The identified visual ideas are then linked to form a phrase using sentence templates, particular language grammar rules, or combinatorial optimization techniques [7].

Long-term dependencies are known to be challenging for recurrent neural networks to learn. To address this flaw in image captioning, Chen and Zitnick suggest that a visual representation of an image be dynamically built while a caption is produced, allowing long-term visual ideas to be recalled throughout the process [8]. To this aim, a collection of latent variables Ut1 is added to convey the visual interpretation of previously produced words Wt1. The likelihood of producing the word wt using these latent factors is shown below:

$$P(wt,V \mid Wt-1,Ut-1) = P(wt \mid V,Wt-1,Ut-1)P(V \mid Wt-1,Ut-1),$$

Wt1 indicates produced words (w1, ..., wt1), and V indicates observed visual characteristics. The authors implement the aforementioned concept by incorporating a recurrent visual hidden layer u

into Recurrent Neural Networks. The recurrent layer u is useful for predicting the next word wt as well as reconstructing visual characteristics V from previous words Wt1.

A completely convolutional localization network design is used in dense captioning [9], which consists of a convolutional network, a dense localization layer, and an LSTM [20] language model. The dense localization layer takes an image and analyses it in a single, efficient forward pass, inferring a set of areas of interest in the picture. As a result, unlike Fast R-CNN or a complete network of Faster R-CNN, it does not need external region suggestions. The localization layer's working concept is similar to that of Faster R-CNN.

Another dense captioning technique suggested by Yang et al. [10] may overcome these challenges. First, it deals with an inference process that is based on the area's visual characteristics as well as the anticipated captions for that area. This allows the model to identify a single suitable location for the bounding box. Second, they use context fusion to combine context information with visual characteristics of the relevant area in order to give a comprehensive semantic description.

Vinyals et al. [11] proposed the Neural Image Caption Generator technique (NIC). This technique uses a CNN for image representations and an LSTM for image caption generation. The output of the final hidden activations of CNN is provided as an input to the LSTM decoder in this particular CNN, which uses a novel batch normalization technique. This LSTM can retain track of things that have been described in text before. The maximum likelihood estimation method is used to train NIC. The initial state of an LSTM contains image information. According to the current time step and the prior concealed state, the following words are produced. This procedure continues until the sentence's end fragment is reached. Because image data is only supplied at the beginning of the process, it may have vanishing gradient issues. The importance of the words produced at the start diminishes as well.

Wang et al. [12] presented a deep bidirectional LSTM-based approach for producing semantically and contextually rich image captions. A CNN and two distinct LSTM networks are included in the proposed design. It learns long-term visual-language interactions by combining previous and future context knowledge.

The major problem with sequential models is that they often provide overgeneralized expressions that lack information seen in the input picture. To address this issue, Tian et al. [13] proposed a hierarchical framework for image captioning that investigates both the compositionality and the sequential nature of natural language by selectively attending to different modules corresponding to unique attributes of each object detected in an input image in order to include specific descriptions such as counts and color. Experiments on the MSCOCO dataset revealed that the suggested model outperforms state-of-the-art models across many assessment criteria, while also providing visually interpretable findings.

As a result, in this paper, we examine the relationship between model complexity, as measured by the total number of parameters, and the effectiveness of different CNN architectures on feature extraction for Image Caption Generation using popular CNN architectures that have been used for Object Recognition tasks. We use two widely used Image Caption Generation frameworks: (a) the Neural Image Caption (NIC) Generator introduced in [14] and (b) the Soft Attention based Image Caption Generation presented in [15]. We found that the performance of Image Caption Generation changes when various CNN architectures are used, and that it is not directly linked to model complexity or CNN performance on object identification tasks. We test several versions of ResNet [16] with varying depths (number of layers in the CNN) and complexity: ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, where the numerical component of the name stands for the number of layers in the CNN (such as 18 layers in ResNet18 and so on).

The system replaces linear multiplication with convolution, then combines it with global video feature maps before computing the channel weight of each frame's feature maps using channel attention. To learn spatial-temporal assignment parameters, the weighted feature maps are input into parameter sharing gated recurrent convolutional units (SGRU-RCN, a version of gated recurrent convolutional units) [17], which may be used as weights for aggregating local descriptors to specific cluster centres. Different convolutional recurrent neural networks may readily implement the suggested channel soft attention.

Traditional deep learning-based video captioning models include encoder-decoder architecture and gated recurrent neural networks (RNN), such as long short-term memory (LSTM) or gated recurrent unit (GRU). Each input video sequence is encoded into semantic representations and

delivered to a decoder, which generates video captions. These methods, however, have the following drawbacks:

1. They are unable to provide natural captions for lengthy films including a variety of events.

2. They don't grasp what's going on in the context.

The traditional RNN is insufficient for encoding information in lengthy video sequences with long-term dependencies. Lengthy-term dependence is still an unresolved problem for long sequential data, despite the fact that LSTM or GRU partly handle it. Furthermore, traditional captioning algorithms seldom preserve the contextual information included in a video sequence. Traditional video captioning models only operate for short video clips with basic scenes due to memory limitations in RNN models, and thus are not suitable to lengthy movies with many complex occurrences.

The DNC (differentiable neural computer) evolved from the Turing machine. Turing machines are abstract contemporary computer architectures that demonstrate that with enough external memory and methods, any computation is feasible [19]. The Neural Turing Computer (NTM) was suggested by Google Deep Mind, a system that uses neural networks and external memory to construct a differentiable Turing machine, and an enhanced version of the NTM model, DNC, was presented in a Nature article in 2016.

This technique, however, is restricted to video snippets with brief, static backdrops. With the success of neural machine translation (NMT), the sequence-to-sequence model, an LSTM-based encoder-decoder structure, is now being used for video captioning [20]. They use the pre-trained CNN to get semantic representations of video frames, which they feed into the LSTM encoder to get the final hidden states. The loss function of the LSTM decoder is then optimized for one-step forward prediction to produce following words.

2.2 Summary of the Literature

We identify some research gap based on entire literature review that are mention in below

- Low accuracy with CNN-GAN and RNN-GAN
- The ensemble framework cant able to detect all aspects or object consists in images, it effect on training, these experiments illustrates this module can detect only large objects.
- Ensemble cannot detect new more features than RNN-GAN and CNN-GAN, it only collect unique features and detect as ensemble results.
- Some color objects aren't be detect by both algorithms as well as ensemble model in image.

No	Technique	Dataset	Extracted Features	Research Gap
1	x-Means clustering algorithms and Neuro Fuzzy algorithm [1]	GSM operation data, 24,900 customers 22 attributes Turkey dataset	Some value-added services and some values added services	System reflects good accuracy on structured dataset only.
2	Naïve Bayes, Decision Tree[2]	European operator 106,405 customers 112 attributes	Contract, usage pattern patterns, and calls pattern	High error rate to detect actual churn due to redundant features.
3	Neural network, Regression [3]	Unknown 129,892 customers 113 attributes	Demographic, Value added, usage pattern	Heterogeneous dataset tedious to handle in similar patterns environments.
4	Neural network, Regression [4]	Unknown, 169 customers 10 attributes	Demographic, Billing data, usage pattern, customer relationship	High space complexity generates in each layer

No	Technique	Dataset	Extracted Features	Research Gap
5	Stepwise variable selection partial least squares [5]	Cell2Cell Dataset 100,000 customers 171 attributes	Behavioral information, Customer care and demographics	Redundant features should be generating high error rate.
6	Artificial Neural Network [6]	ML Dataset of UCI 2,427 user's information with 20 attributes	Demographics, Usage pattern, Value added services	It works only define statically parameters.
7	Binomial logistic regression model [7]	Iranian telco operator 3150 customers 15 attributes	Demographic, call usage pattern, customer care service	Language influence should be generate irrelevant features vector.
8	Generalized additive models (GAM) [8]	Belgian 134, 120 customers 27 attributes	Demographic Usage patter, bill and payment	High error rate during unknown text prediction.
9	Logistic regression Decision tree [9]	Polish mobile operator 122098 customers 1381 attributes	Demographic, call data records, customer care services	Its works only synthetic data only and high data reduction rate.
10	Decision tree as well as machine learning [10] algorithms has used.	Cell2Cell Dataset 100,000 customers 171 attributes	Behavioral data, of customer care and feature information	Behaviors information generate the churn possibility sometime it generate false ratio.

Chapter 3

PROPOSED WORK

3.1 Objectives and Challenges

- We developed a model for generating stylistically interesting and semantically relevant image captions by learning from a large corpus of stylized text without aligned images.
- To design and develop a hybrid deep learning algorithm to generate effective captions using combination of CNN with EfficientNetB3 and Attention Model.
- To generate effective captions using GRU and greedy approach from module testing and validate with ground truth captions to validate the algorithms accuracy.
- To explore and validate the efficiency of proposed model with various existing systems and show the effectiveness in real time environment

3.2 Project Scope

Deep learning methods have demonstrated advanced results on caption generation problems. What is most impressive about these methods is that one end-to-end model is often defined to predict a caption, given a photograph, rather than requiring sophisticated data preparation or a pipeline of specifically designed models. Deep learning has attracted a lot of attention because it's particularly good at a kind of learning that has the potential to be very useful for real-world applications. The ability to find out from unlabeled or unstructured data is a huge benefit for those curious about real-world applications. This system is most effective in below areas.

- Generate caption for random image and display the information
- Video captioning and detect the objects in entire video
- Provide automatic supportive tagging or captioning for images and video for social media posts.

3.3 Problem Statement

Generate Caption from Image with maximum accuracy.

3.4 Proposed Algorithm

We proposed a system with hybrid deep learning algorithm, (CNN + Attention Model + GRU) for encoding and decoding for and extract the captaining during the module training and testing. The approach we evaluate for supervised as well as supervised learning approach.

Training Process

Input: Training dataset Train-Data [], Many activation functions [], Threshold Th

Output: Extracted Features Feature set[] for a trained module that has been finished.

Step 1: Set the data input block d[], the activation function, and the epoch size.

Step 2: Features-pkl ← Feature-Extraction (d[])

Step 3: Feature-set [] ← optimized (Features-pkl)

Step 4: Return Feature-set []

Testing Process

Input: Test Dataset which contains various test instances TestDB-Lits [], Train dataset which is built by training phase TrainDB-Lits [] , Threshold Th**.**

Output: HashMap < class label, Similarity Weight > all instances which weight violates the threshold score.

Step 1: For each testing records as given below equation

$$testFeature(k) = \sum_{m=1}^{n} (.\ featureSet[A[i] \dots \dots A[n] \leftarrow \text{TestDBLits}\ )$$

Step 2: Create feature vector from $testFeature(m)$ using below function.

$$\text{Extracted\_FeatureSetx [t.\ldots n]} = \sum_{x=1}^{n}(t) \leftarrow testFeature\ (k)$$

Extracted_FeatureSetx[t] holds the extracted feature of each instance for testing dataset.

Step 3: For each train instances as using below function

$$trainFeature(l) = \sum_{m=1}^{n} (.\ featureSet[A[i] \ldots\ldots A[n] \leftarrow \text{TrainDBList})$$

Step 4: Generate new feature vector from $trainFeature(m)$ using below function

$$\text{Extracted\_FeatureSet\_Y[t.\ldots n]} = \sum_{x=1}^{n}(t) \leftarrow TrainFeature\ (l)$$

Extracted_FeatureSet_Y[t] holds the extracted feature of each instance for training dataset.

Step 5: Now evaluate each test records with entire training dataset

$$weight = calcSim\ (\text{FeatureSetx} \mid\mid \sum_{i=1}^{n} \text{FeatureSety[y]})$$

Step 6: Return Weight

Chapter 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 Hardware and Software requirements

Hardware Requirements

- Processor: - Intel Pentium 4 or above
- Memory: - 2 GB or above
- Hard Disk: - 500gb

Software requirements

Technologies and tools used in Policy system project are as follows Technology used:

Front End

- Google chrome 96.0.46
- Tool: Python on words
- Programming Language: Python 2.7 onwards and HTML

4.2 Analysis of Input and expected Output Related to Project

- Input to the System: The input of the system is the CSV file where the customer twitted data on twitter or any social media account.
- Output of the System**:** It predict the class of sentiment like joy, fear, Love, sadness, surprise and anger etc.

4.3 UML Diagrams

The Unified Modeling Language (UML) gives a standard way to write a system model covering the conceptual ideas. It can be used for modeling a system independent of a platform language. It is a graphical language for visualizing, specifying, constructing and documenting information about software intensive system.

4.3.1 Use-Case Diagram

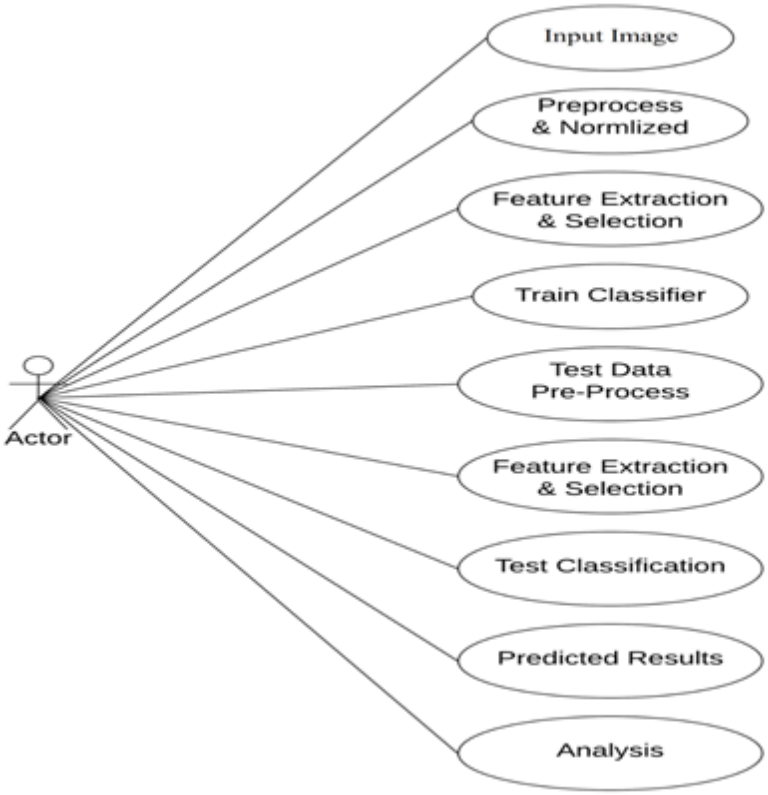


Figure 4.1 Use case Diagram

4.3.2 Class Diagram

A class diagram in the Unified Modelling Language (UML) is a sort of static structure diagram that portrays the structure of a framework by demonstrating the framework's classes, their characteristics, operations (or techniques), and the connections among objects.

The class diagram is the primary building piece of protest situated modelling. It is utilized for general theoretical modelling of the precise of the application, and for point by point modelling making an interpretation of the models into programming code. Class diagrams can likewise be utilized for information modelling. The classes in a class diagram speak to both the primary components, interactions in the application, and the classes to be modified

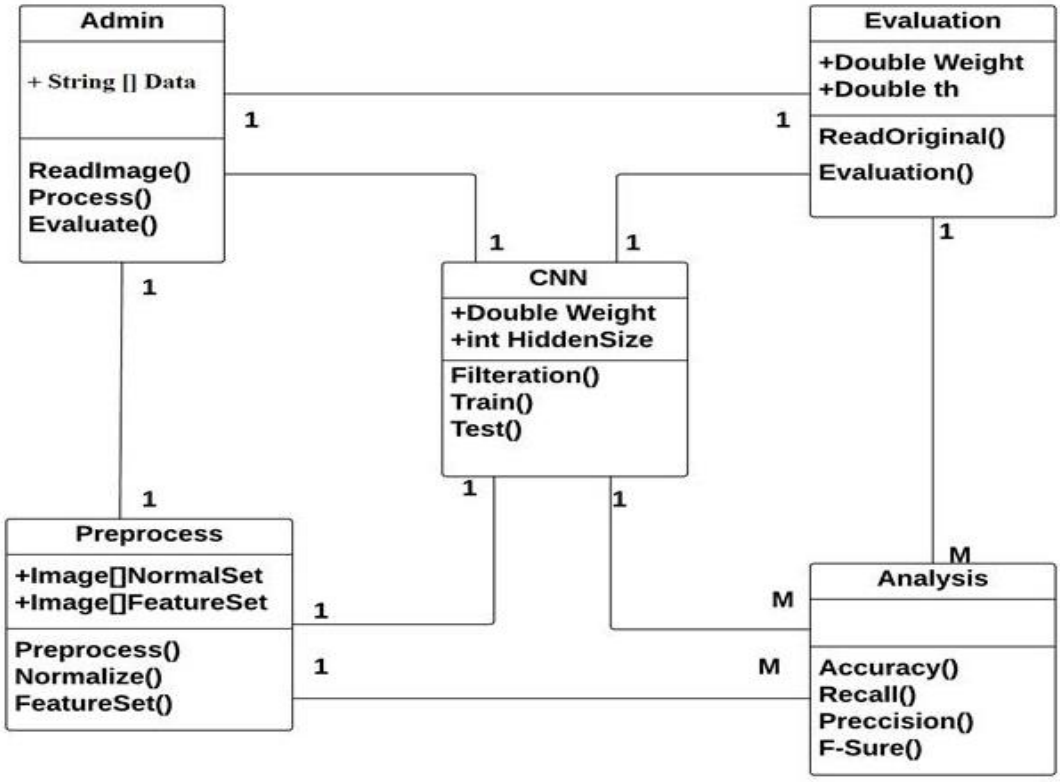


Figure 4.2 Class Diagram

4.3.3 Sequence Diagram

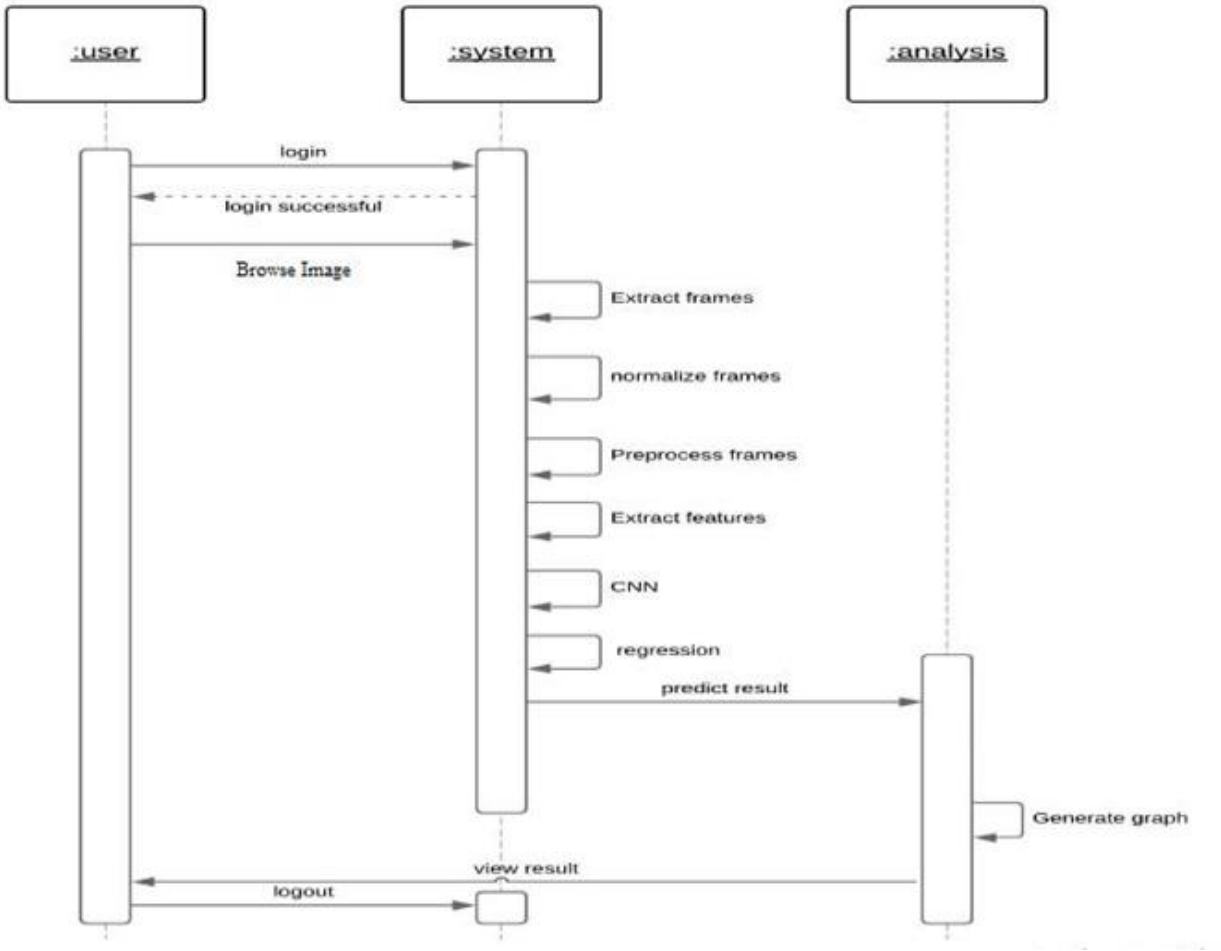


Figure 4.3 Sequence Diagram

4.3.4 Activity Diagram

Activity diagram is a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The flow can be sequential, branched or concurrent. Here we have two activity diagrams, one for the user and the other for the system. The purpose of activity diagrams is to capture the dynamic behavior of the system. These are drawn as follows

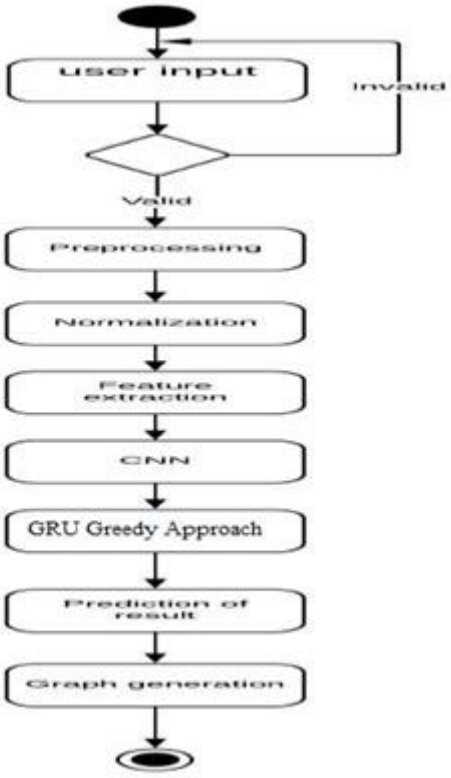


Figure 4.4 Activity Diagram for User

Chapter 5

PROPOSED SYSTEM OVERVIEW AND IMPLEMENTATION

5.1 System Architecture

We utilize a CNN + GRU to take an image as input and output a caption. An “encoder” RNN maps the source sentence (which is of variable length) and transforms it into a fixed-length vector representation, which in turn is used as the initial hidden state of a “decoder” GRU which generates the final meaningful sentence as a prediction. The below Figure 4.1 depicts the proposed system architecture for caption generation using deep learning model. In below section we describe each phase of system to generate the final outcome of validation set.

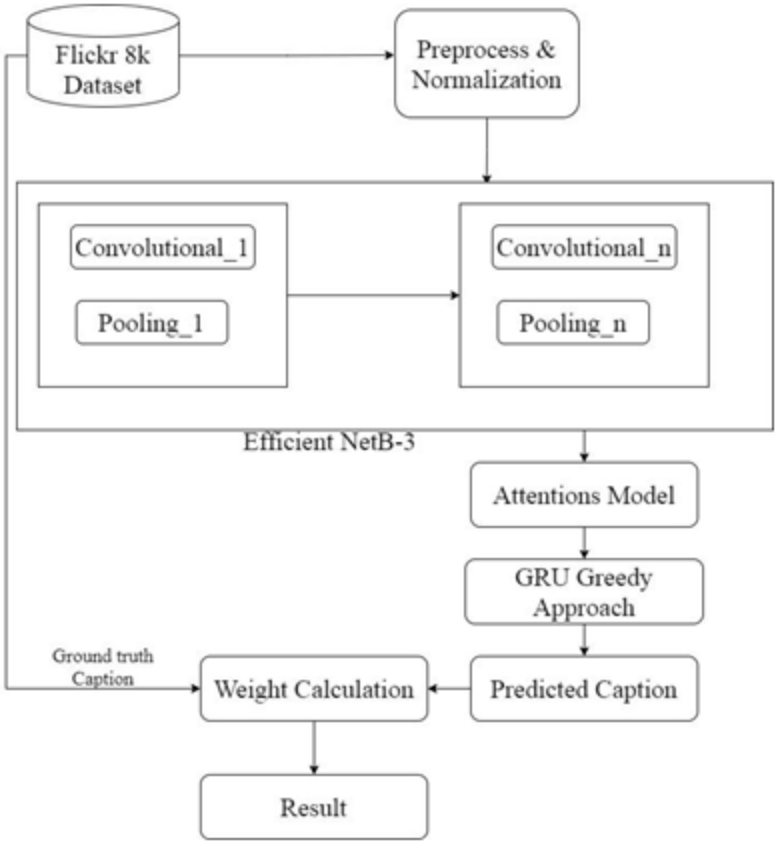


Figure 5.1 Proposed system architecture

Preprocessing and Normalization: The data collection has done from synthetic repository called MSCOCOC. The data might be imbalance with dimension thus it needs to balance before

proceeding to CNN. In preprocessing we define fix set of dimensions of image (300*300) and eliminate those objects which are not readable for misclassified. The cross validation has also done in this similar phase for training and testing. (e.g., 5-fold, 10-fold, 15-fold etc.). In the dataset each image having caption as ground truth caption, and that used for analysis. Due to the large size of the dataset, data processing was done on the host computer and the results were saved as pickle files before being sent to the cloud for model training.

Lower case conversion: The dataset texts include words with various letter cases, which presents a challenge for the model since the same words with different capitalizations would be treated differently, resulting in an increase in problem vocabulary and, as a result, complexity. As a result, to prevent this, the whole text must be converted to lower case.

Punctuation removal: Because the goal of this study is to create descriptive sentences for pictures without using punctuation, the existence of punctuation adds complexity to the issue, which is beyond the scope of this study.

Number removal: The presence of numerical data in the texts is a challenge to the model since it enhances language; thus, it should be eliminated.

Indicate start and end sequence: To signal the beginning and finish of the prediction sequence to the model, the word tokens <start> and<end> are inserted at the beginning and end of each phrase.

Tokenization: The clean text is broken down into component terms, and a dictionary with the full vocabulary is produced for word to index and index to word matching.

Vectorisation: Before learning word indicative vectors from these sequencing, the words in the text data are transcribed using unique numerical interpretations from the word to index dictionary, transforming the cleaned sentence specifications to numerical sequences. Lesser sentences are stretched to the length of the largest sentence sequence to compensate for variable sentence lengths.

CNN using EfficientNetB3: The EfficientNetB3 module has used for CNN, to extract the features and selection as well. The below code snippet demonstrates the building of CNN with

EfficientNetB3. The global image feature vectors for the frames are retrieved from Keras applications using the final fully connected layer of the EfficientNetB3 CNN. Before extraction, the pictures are compressed to 300x300 arrays, and the pixels are normalized to a scale of 0 to 255 to suit the EfficientNetB3 model input. A pretrained model is utilized for the picture scene features. To match the input shape of the model, the images are compressed to 224x224 and normalized on a scale of 0 to 255. Both networks output feature vectors are saved as pickle files for simple uploading for validation testing.

Convolution, batch normalization, and ReLU functions are combined in the function D. Convolution is the multiplication of image and kernel matrices element by element. In each mini-batch, batch normalization involves moving inputs with a zero mean and unit variance. The ReLU function is then used to transform the elements with negative values to zero.

Attention Model and GRU: attention model basically work for selection of features that received from CNN. The selection features eliminate the redundant and non-essential features. The attention model has divided into four different phases these are describes in below,

Encoder: The pre-trained Inception model has already done the visual encoding; thus the encoder is extremely straightforward. It has a Linear layer that receives the pre-encoded gives the view and transmits them to the Decoder.

Sequence Decoder: This is a GRU-based recurrent network. Once passing through a Hidden layer, the captions are sent in as input.

Attention: The Attention module assists the Decoder in focusing on the most important portion of the picture for producing each word in the output sequence.

Sentence Generator: This module is made up of a few Linear layers. It takes the Decoder's output and generates a frequency for each word in the lexicon, as well as for each place in the anticipated sequence.

A recurrent GRU can extract a lot of information from a picture, both temporally and spatially. However, semantic information is required to enhance the features obtained from the encoder in order to produce semantically correct sentences. The Greedy method is utilized in our approach

to generate semantic vectors, and a semantic compositional network is used as the decoder. Furthermore, a dual learning method is employed to save the semantic vector's information throughout training, allowing the semantic information in forward flow to be effectively used.

In GRU we applied the greedy approach for caption generation in both training testing. In both module generated caption has evaluated with ground truth sentence for validation and generate the 4 BELU scores and cosine similarity weight for confusion matrix. In result section we depicts the various ground text features extraction methods has used for generating the BELU score and similarity weight.

5.2 Mathematical Model

A System has represented by 5-different phases, each phase works with own dependency System $S = (Q, \sum, \delta, q_0, F)$ where −

- **Q** is a finite set of states.
- $\sum$ is a finite set of symbols called the alphabet.
- **Δ** is the transition function where $\delta : Q \times \sum \rightarrow Q$
- $\mathbf{q_0}$ is the initial state from where any input is processed ($q_0 \in Q$).
- **F** is a set of final state/states of Q ($F \subseteq Q$).

Q = initial transactional data with image caption

∑= {feature extraction, feature selection technique, store individual feature}

Δ = detect the face all validation process

q0 = Initial transaction T[0]

F = {training dataset, testing dataset}

State =According to achieved weight system recommend the actual predict result. Deep Convolutional Neural Network (DCNN)

Input: Test Dataset which contains various test instances TestDBLits [], Train dataset which is build by training phase TrainDBLits[] , Threshold Th.

Output: HashMap <class_label, SimilarityWeight> all instances which weight violates the threshold score.

Step 1: For each read each test instances using below equation

$$testFeature(m) = \sum_{m=1}^{n} (.\ featureSet[A[i] \dots \dots . . A[n] \leftarrow \text{TestDBLits})$$

Step 2 : extract each feature as a hot vector or input neuron from $testFeature(m)$ using below equation.

$$\text{Extracted\_FeatureSetx [t……n]} = \sum_{x=1}^{n}(\text{t}) \leftarrow testFeature\ (\text{m})$$

Extracted_FeatureSetx[t] contains the feature vector of respective domain

Step 3: create the number of convolutional

For each read each train instances using below equation

$$trainFeature(m) = \sum_{m=1}^{n} (.\ featureSet[A[i] \dots \dots . . A[n] \leftarrow \text{TrainDBList})$$

Step 4 : extract each feature as a hot vector or input neuron from $testFeature(m)$ using below equation.

$$\text{Extracted\_FeatureSety[t……n]} = \sum_{x=1}^{n}(\text{t}) \leftarrow testFeature\ (\text{m})$$

Extracted_FeatureSetx[t] contains the feature vector of respective domain.

Step 5 : Now map each test feature set to all respective training feature set

$$weight = calcSim\ (\text{FeatureSetx} \,||\, \sum_{i=1}^{n} \text{FeatureSety[y]})$$

The $Train\_Feature[]$ and $Test\_Feature[]$ both requires as input for test classifier when generate similarity score between two input objects. These are two separate attributes which represents the training and testing instance respectively. The Th is the denominator that used for selection of each epoch layer result. The $T[j]$ denotes jth attributes of testing instance while the $T[k]$ depicts kth train attribute information. By using feature selection method, we extract some features from both instances and forward to similarity measurement function which is described in step 5. The number of validates instances by threshold is the dense optimized results by CNN

5.3 Data Set.

1.Train dataset

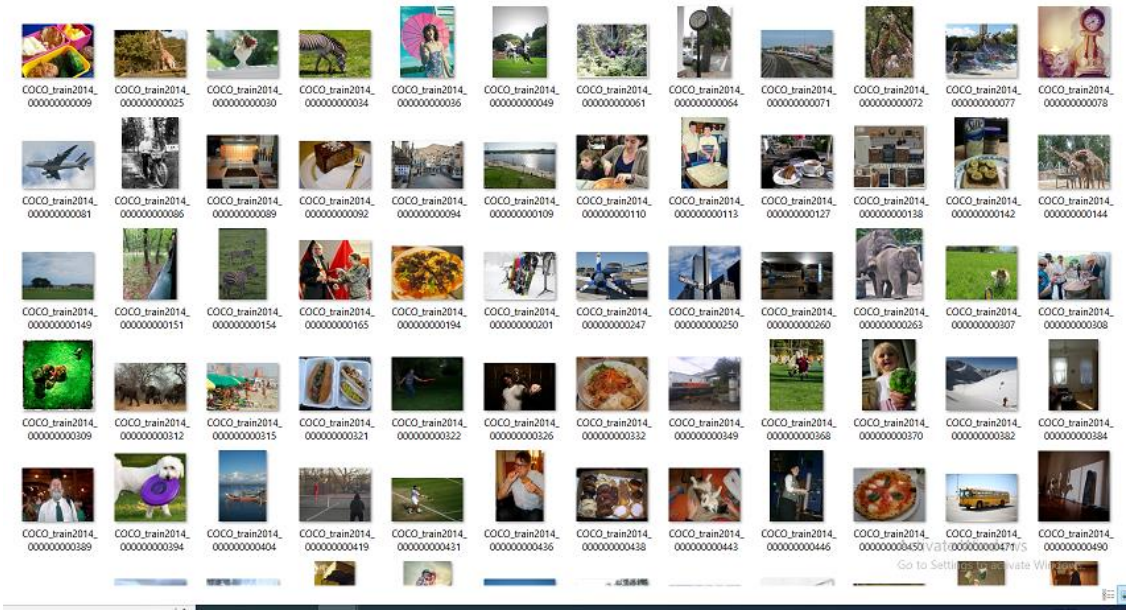


Figure 5.2 Image for Train Data set

Figure 5.2 shows some samples images from the dataset with names.

2.Test Dataset

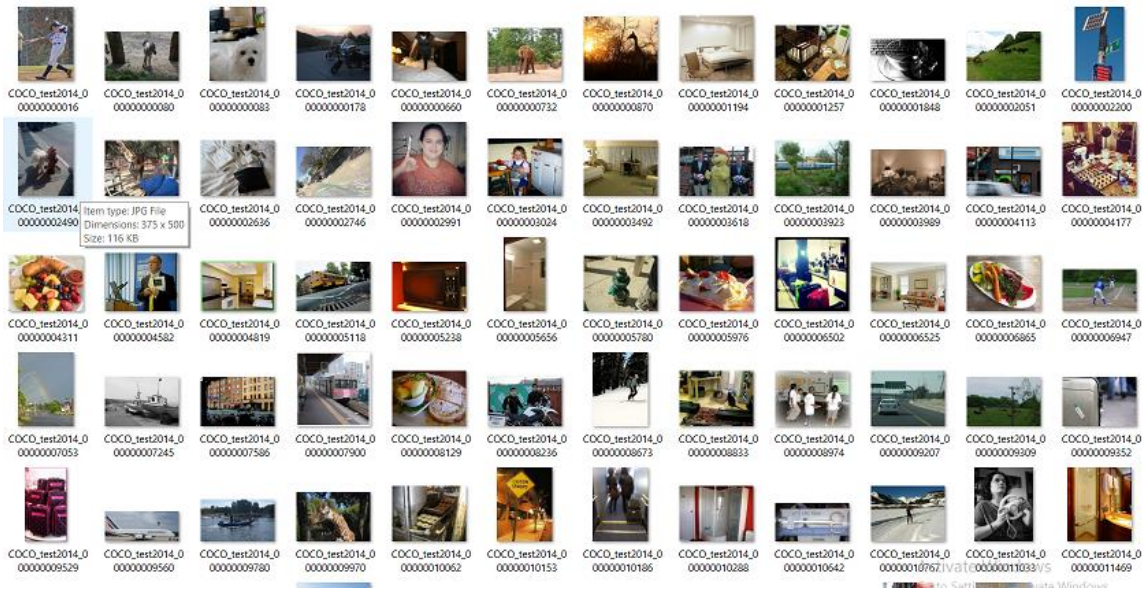


Figure 5.3: Image for Test Data set

Figure 5.3 shows the some sample images from Test dataset with names.

3. Annotation

Figure 5.4 Image for Annotations

Figure 5.4 shows the some sample annotations from dataset .

4. Code Snippet:

```
from __future__ import absolute_import, division, print_function, unicode_literals

import tensorflow as tf
from tensorflow.keras import activations, layers, losses, optimizers

# You'll generate plots of attention in order to see which parts of an image
# our model focuses on during captioning
import matplotlib.pyplot as plt
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
from matplotlib.figure import Figure

# Scikit-learn includes many helpful utilities
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle

import re
import numpy as np
import os
import time
import json
from glob import glob
from PIL import Image
import pickle
from tqdm import tqdm
import efficientnet.tfkeras as efn
'''## Download and prepare the MS-COCO dataset

You will use the [MS-COCO dataset](http://cocodataset.org/#home) to train our model. The dataset contains over 82,000 images, each of which has at least 5 different caption
annotations. Due to computational considerations we will only consider the 71,973 captions that have 8 or fewer words, and the corresponding 48,659 corresponding images. Indeed we
limit this further to the 20,000 images with the most captions, a total of 43,314 captions. The code below downloads and extracts the dataset automatically.

**Caution: large download ahead**. The 20,000 images, is about ~3GB file.
'''
annotation_file = 'annotations/captions_train2014.json'
PATH = 'train2014/'
##annotation_zip = tf.keras.utils.get_file(
##    'captions.zip',
##    cache_subdir=os.path.abspath('.'),
##    #origin='http://images.cocodataset.org/annotations/annotations_trainval2014.zip',
##    origin="https://www.uio.no/studier/emner/matnat/its/TEK5040/h19/data/captions_trainval2014_8_20000.zip?vrtxPreviewUnpublished",
```

Figure 5.5 Sample Image for Code -1

```
# limit to the first NUM_EXAMPLES from the shuffled set
NUM_EXAMPLES = 1000
trainval_captions = captions[:NUM_EXAMPLES]
img_name_vector = img_name_vector[:NUM_EXAMPLES]

print("Number of total captions: %d" % len(all_captions))
print("Number of captions used for training and validation: %d" % len(trainval_captions))

image_height = 224
image_width = 224
target_size = (224, 224,3)
def load_image(image_path):
    img = tf.io.read_file(image_path)
    img = tf.image.decode_jpeg(img, channels=3)
    img = tf.image.resize(img, (image_height, image_width))
    img = tf.keras.applications.nasnet.preprocess_input(img)

    return img, image_path

image_model=efn.EfficientNetB3(
        weights='noisy-student',
#         weights='imagenet',
        input_shape=target_size, include_top=False)
new_input = image_model.input
hidden_layer = image_model.layers[-1].output

#image_features_extract_model = tf.keras.Model(new_input, hidden_layer)
image_features_extract_model = efn.EfficientNetB3(
        weights='noisy-student',
#         weights='imagenet',
        input_shape=target_size, include_top=False)

# whether to force recompute of features even if exist in cache, useful if e.g.
# changing model to compute features from
FORCE_FEATURE_COMPUTE = False

# Get unique images
encode_train = sorted(set(img_name_vector))
if not FORCE_FEATURE_COMPUTE:
```

ython file | length : 24,337 lines : 622 | Ln : 112 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Figure 5.6 Sample Image for Code – 2

Figure 5.5 and Figure 5.6 are showing the some sample for the code which is written in Python .

Chapter 6

RESULT AND ANALYSIS

The datasets used are benchmark datasets that are used to assess the system's performance. Flickr8k, Flickr30k are all massively preferred benchmark datasets for the application of image captioning. The dataset COCO it is an acronym for common objects in context, and it is a dataset created by Microsoft. It is massive in size and covers over 91 kinds of photos, with over 164k photographs in total. For training sets of 330K photos and 1.5 million object instances, MS COCO uses iconic images. The Flickr8k holds 8000 photos, whereas the Flickr30k has roughly 30,000. Each image within these datasets is labeled or mapped with five captions. These libeling's can also help with system activities. Based on previous experiments around1000 photographs are utilized for testing and validation purpose in both Flickr datasets. Transfer learning can be used to implement these models. Models are trained on high-configuration GPUs that can be found online as well as those on the system we have used GTX 1650. The training period varies depending on the GPU; for such applications, systems with larger Gigabyte GPUs, such as 4-16 GB, are desirable, and software such as Python is required for the tensorflow libraries, with IDEs for python such as Jupyter Notebook or PyCharm or Visual Studio Code with libraries such as the tensorflow, numpy, keras, pandas and few others are required.

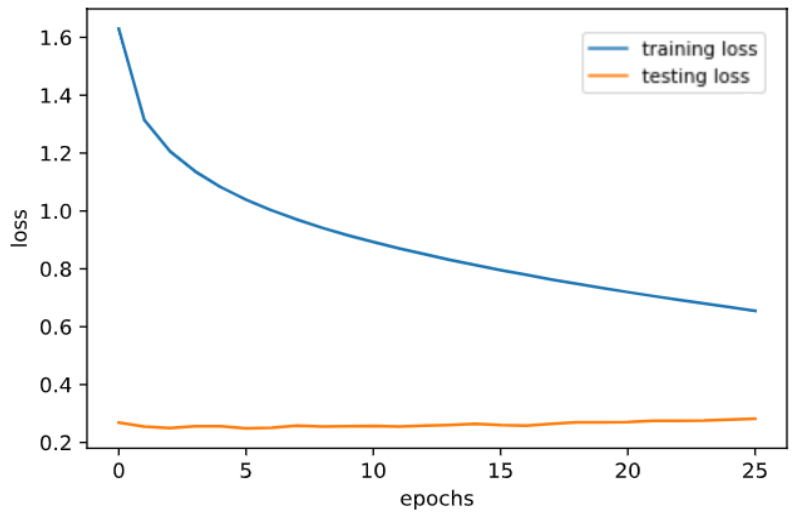


Figure 6.1: Model performance on training

The above Fig. 6.1 describes training and testing model performance with proposed EfficientNetB3 on Flickr8k dataset. With a count of 6000 pictures validated in the training and a 1000 for testing set. When system deals with high epoch values it reduces the overall loss in training while it slightly increases or constant for validation.

Figure 6.2: Model performance testing with various feature selection techniques on test image 1

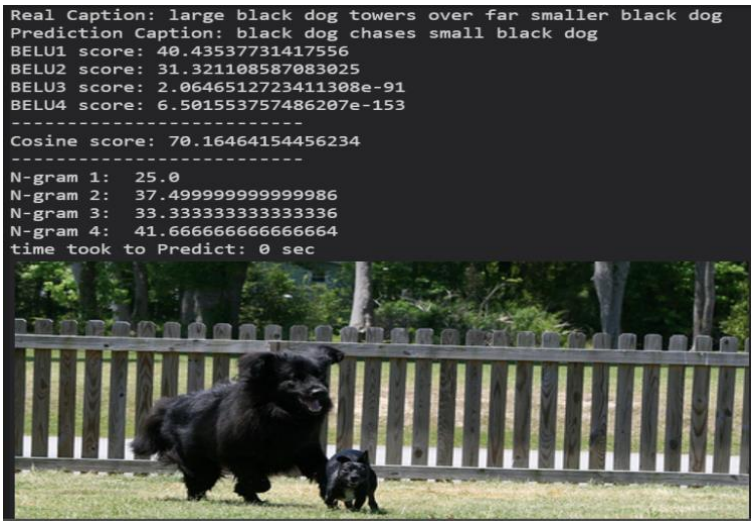


Figure 6.3: Model performance testing with various feature selection techniques on test image 2

The Fig 6.2 and Figure 6.3 describes the caption generation in validation phase for two different images. The generated caption has evaluated with both ground truth sentence and evaluated with

4 BELU functions and Bigrams methods with cosine similarity algorithm. According to both results we conclude the default cosine similarity produces higher accuracy over the other methods.

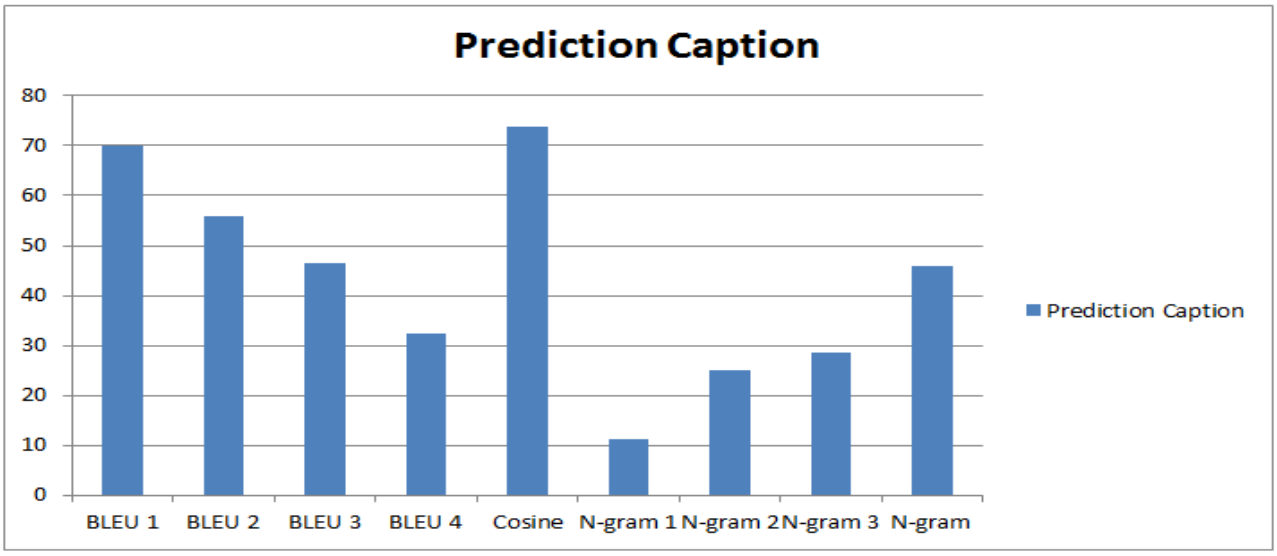


Figure 6.4: Prediction accuracy for caption generate with different techniques on test image

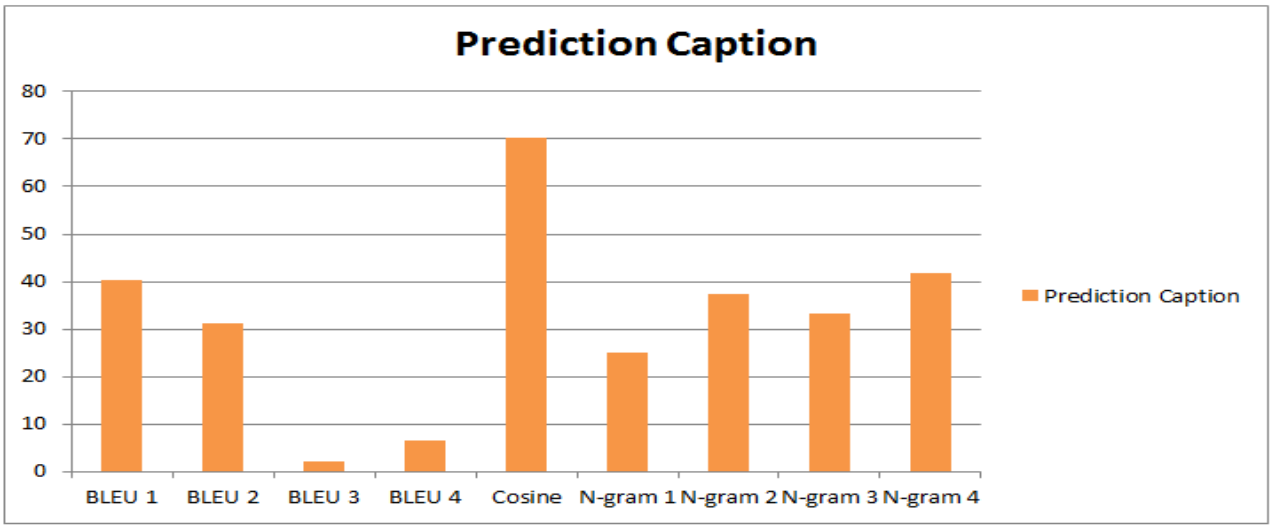


Figure 6.5: Prediction accuracy for caption generate with different techniques on test image

The above Fig. 6.4 and 6.5 all weighted accuracy for both objects evaluated in Figure 3 and 4. BELU-3 and N-Gram generates lower results than other techniques thus cosine similarity produces high caption generation accuracy for system. It produces around 75% accuracy for cosine similarity algorithm. In another experiment we have done comparative analysis with CNN-LSTM and CNN-Bi-LSTM [21] as existing system for caption generation. The below Table 1 demonstrates similarly.

Method	BLEU-1	BLEU-2	BLEU-2	BLEU-2	ROUGE-L	Cosine
CNN-LSTM [21]	46.7	31.32	19.4	9	25.7	NA
CNN-Bi-LSTM [21]	55.00	34.9	24.8	13.1	27.9	NA
Proposed	**70.00**	**55.77**	**46.47**	**32.46**	NA	**73.78**

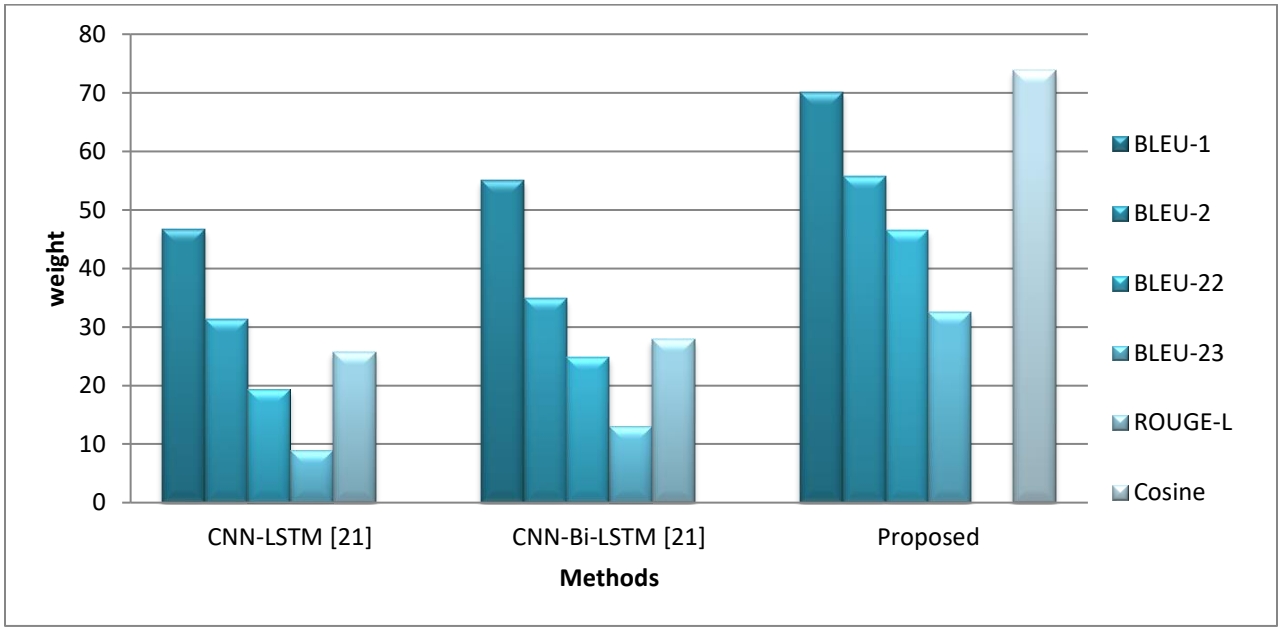


Figure 6.6 : Experiment analysis and comparative analysis of system

According to this Figure 6.6 we conclude our system predicts superior results in terms of all performance parameters.

Chapter 7

SCHEDULE OF WORK

Phase-1 Literature Survey

Phase-2 Study of exiting techniques for strengths and weaknesses

Phase-3 GUI design and database design

Phase-4 Implementation and algorithm design

Phase-5 Testing of hypothesis and performance analysis of system

Phase-6 Validate the proposed techniques

Table 7.1 System Implementation Plan

Sr. No	Task Name	Begin date	End date	Remark
1	Searching, discussion and finalize on project Idea	15-Aug-21	28-Aug-21	Done
2	Study of Base Paper	29-Aug-21	18-Sep-21	Done
3	Understanding project need and prerequisites	19-Sep-21	20-Oct-21	Done
4	Literature Survey	21-Oct-21	06-Nov-21	Done
5	Technical Paper-1Writing	12-Nov-21	02-Dec-21	Done
6	Project planning and designing	03-Dec-21	21-Jan-22	Done
7	Mathematical modeling	04-Feb-22	04-Feb-22	Done
8	Developing project stage-1 documentation	05-Feb-22	13-Feb-22	Done
9	Attending and presenting paper in 1st conference	25-Feb-22	08-May-22	Done
10	DS -1 presentation and Report Submission	09-May-22	10-May-22	Done
11	Implementation	12-May-22	30-Jun-22	Done

12	Demonstration of project	01-Jul-22	07-Jul-22	Done
13	Technical Paper-2 Writing	08-Jul-22	01-Aug-22	Done
14	Testing of hypothesis and performance analysis of system.	01-Aug-22	15-Aug-22	Done
15	Project report writing	15-Aug-22	30-Aug-22	Done

Chapter 8

CONCLUSION

We show how deep learning techniques are used to generate image captions in this study. We employed CNN models with GRU to obtain an accurate description of the picture. The ideas of the generalized Encoder – Decoder model have been discussed. Its application, as well as the attention model and how it was applied in various techniques. We also discussed several datasets that can be used to evaluate a system by using them as benchmark datasets. Finally, we discussed assessment metrics and how to use them to evaluate our model. This research focused on the image captioning task, which is a fundamental problem in artificial intelligence. For extraction, the CNN Efficientnet B3 module was employed, and GRU was used to create captions for the images. To obtain the blue score for the complete testing dataset and demonstrate the performance of the proposed system, several cosine similarity and N-Gram feature extraction algorithms were applied.

Future work

Automatic image captioning is a relatively new task, thanks to the efforts made by researchers in this field, great progress has been made. In our opinion there is still much room to improve the performance of image captioning. First, with the fast development of deep neural networks, employing more powerful network structures as language models and/or visual models will undoubtedly improve the performance of image description generation. Second, because images are consisted of objects distributed in space, while image captions are sequences of words, investigation on presence and order of visual concepts in image captions are important for image captioning

9.REFERENCES

[1] Shreyasi Charu, S.P.Mishra, Tapan Gandhi.: Vision to Language: Captioning Images using Deep Learning, 2020 International Conference on Artificial Intelligence and Signal Processing (AISP).

[2] Viktar Atliha , Dmitrij Seˇsok.: Comparison of VGG and ResNet used as Encoders for Image Captioning, 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)

[3] Fang Fang, Hanli Wang, Pengjie Tang.: Image Captioning with word level attention, 2018 25th IEEE International Conference on Image Processing (ICIP).

[4] R. Mason, E. Charniak, Nonparametric method for data driven image captioning, in: Proceedings of the Fifty Second Annual Meeting of the Association for Computational Linguistics, 2014.

[5] S. Li, G. Kulkarni, T.L. Berg, A.C. Berg, Y. Choi, Composing simple image descriptions using web-scale n-grams, in: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, 2011.

[6] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. Berg, T. Berg, H. Daume, Midge: Generating image descriptions from computer vision detections, in: Proceedings of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics, 2012

[7] USHIKU, Yoshitaka; HARADA, Tatsuya; KUNIYOSHI, Yasuo. Efficient image annotation for automatic sentence generation. In: Proceedings of the 20th ACM international conference on Multimedia. 2012. p. 549-558.

[8] X. Chen, C. Zitnick, Mind’s eye: a recurrent visual representation for image caption generation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2422– 2431.

[9]. Andrej Karpathy Justin Johnson and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. Pro- ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4565–4574, 2016

[10]. Linjie Yang, Kevin Tang, Jianchao Yang, and Li-Jia Li. Dense captioning with joint inference and visual context. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1978–1987, 2016

[11]. Oriol Vinyals ,Alexander Toshev, ,Samy Bengio ,and Dimitru Erhan. Show and tell: A neural image caption generator. Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3156–3164, 2015.

[12]. Kyunghyun Cho, Bart Van Merrinboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. Association for Computational Lin- guistics, pages 103–111, 2014.

[13]. Junjiao Tian and Jean Oh. Image captioning with compositional neural module networks. Proceedings of the Twenty-Eighth Interna- tional Joint Conference on Artificial Intelligence (IJCAI-19), pages 3576–3584, 2019.

[14] Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. ”Show, attend and tell: Neural image caption generation with visual attention.” In International conference on machine learning, pp. 2048-2057. 2015.

[15] Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. ”Show and tell: Lessons learned from the 2015 mscoco image captioning challenge.” IEEE transactions on pattern analysis and machine intelligence 39, no. 4 (2016): 652-663.

[16] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ”Deep residual learning for image recognition.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[17] Xu, Y.; Han, Y.; Hong, R.; Tian, Q. Sequential Video VLAD: Training the Aggregation Locally and Temporally. IEEE Trans. Image Process. 2018, 27, 4933–4944.

[18] Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef]

[19] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for tatistical machine translation. arXiv 2014, arXiv:1406.1078.

[20] Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112

[21] Alharbi, A.S.M.; de Doncker, E. Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. Cogn. Syst. Res. 2019, 54, 50–61.

[22] Kraus, M.; Feuerriegel, S. Sentiment analysis based on rhetorical structure theory: Learning deep neural networks from discourse trees. Expert Syst. Appl. 2019, 118, 65–79.

[23] Do, H.H.; Prasad, P.; Maag, A.; Alsadoon, A.J. Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review. Expert Syst. Appl. 2019, 118, 272–299.

[24] Abid, F.; Alam, M.; Yasir, M.; Li, C.J. Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter. Future Gener. Comput. Syst. 2019, 95, 292–308.

[25] Yang, C.; Zhang, H.; Jiang, B.; Li, K.J. Aspect-based sentiment analysis with alternating contention networks. Inf. Process. Manag. 2019, 56, 463–478.

[26] Wu, C.; Wu, F.; Wu, S.; Yuan, Z.; Liu, J.; Huang, Y. Semi-supervised dimensional sentiment analysis with variation auto encoder. Knowl. Based Syst. 2019, 165, 30–39

APPENDIX

CPGCON 2021 Paper:

Image Caption Generation using CNN and RNN Models

Ms.Aditi Palwe
Department of Computer Engineering
College of Engineering
Pune,India
aditipalwe1@gmail.com

Prof.Sankirti Shiravale
Department of Computer Engineering
College of Engineering
Pune,India
sankirtishiravale@mmcoe.edu.in

***Abstract*—Image based web crawler is the way toward looking through data by utilizing related images. The tremendous assets of images are accessible on the web in that a large number of the images are contain as with named and without named caption. The users are needed to look through the images relying upon their necessities. In that a significant number of the users can't recover the pertinent images as a result of their unpredicted appropriate inscription on their images. Our task is to generate an automatic caption for the images based on the image content. To produce an image caption, firstly, the content of the image should be fully understood; and then the semantic information contained in the image should be described using a phrase or statement that conforms to certain grammatical rules. Thus, it requires techniques from both computer vision and natural language processing to connect the two different media forms together, which is highly challenging. The paper targets producing mechanized inscriptions by learning the contents of the image. At present images are clarified with human intercession and it turns out to be almost unthinkable task for tremendous databases. The picture information base is given as contribution to a deep neural network Convolutional Neural Network encoder for creating caption which extricates the highlights and subtleties out of our image and Recurrent Neural Network decoder is utilized to interpret the highlights and articles given by our image to acquire consecutive, meaningful description of the image.**

***Index Terms*—Deep Learning, part of speech, image captioning, multi-task learning**

I. Introduction

A. Overview

To facilitate researches in areas such as cross-modal retrieval and the assistance of visually impaired people image captioning which aims to link image with language has become a hot research topic. An image captioning model needs to not only recognize the salient objects, their attributes, and object relationships in an image, but also organize these types of information into a syntactically and semantically correct sentence. With the advances of Neural Machine Translation, recent captioning models generally adopt the encoder decoder framework to "translate" an image into a sentence, and promising results have been achieved. In recent years, researchers have made significant advances in some areas of computer vision understanding, such as image classification, feature classification, object detection and recognition, scene recognition, action recognition, etc. However, having a computer automatically generate natural language descriptions for an image remains a difficult and challenging task. This task connects the two quite different media forms, requiring that computers not only have a correct and comprehensive understanding of the visual content in the image, but also use human language to combine and organize the semantics of the image. The subtasks of image captioning, i.e., identifying semantic elements such as visual objects, object attributes, scenes, are inherently challenging, and organizing words and phrases to express these identified information adds even more difficulty to the entire task.

B. Motivation

Image captioning serves as a bridge to link computer vision and natural language processing. It has attracted considerable attention in the artificial intelligence field. It aims at generating descriptions for input images using natural language. Image captioning is practically useful in applications such as human-machine interaction and content based image retrieval, and in systems helping visually impaired people to perceive the world

C. Objectives

- To recognize objects and Features in the image.
- To generate a fluent description using natural language processing.
- To improve accuracy using deep learning.

II. Review of Literature

J. Lu et al: In this paper, author propose a novel versatile consideration model with a visual sentinel. At each time step, our model concludes whether to take care of the picture (and provided that this is true, to which districts) or to the visual sentinel. The model concludes whether to take care of the picture and where all together to extricate significant data for consecutive wordage. Author test his strategy on the COCO picture subtitling 2015 test dataset and Flickr30K.

P. Anderson et al: In this work, authors propose a joined base up and topdown consideration component that empowers thoughtfulness regarding be determined at the degree of

items and other striking image areas. This is the normal reason for thoughtfulness regarding be thought of. Inside our methodology, the base up system (in light of Faster R-CNN) proposes picture districts, each with a related element vector, while the top-down component decides highlight weightings. Applying this way to deal with picture inscribing, outcomes on the MSCOCO test worker set up another best in class for the assignment, accomplishing CIDEr/SPICE/BLEU-4 scores of 117.9, 21.5 and 36.9, individually.

L. Chen et al: In this paper, Author present a novel convolutional neural organization named SCA-CNN that joins Spatial and Channel wise Attentions in a CNN. In the undertaking of picture inscribing, SCA-CNN progressively regulates the sentence age setting in multi-layer highlight maps, encoding where (i.e., mindful spatial areas at different layers) and what (i.e., mindful channels) the visual consideration is. Authors assess the proposed SCA-CNN design on three benchmark picture subtitling datasets: Flickr8K, Flickr30K, and MSCOCO. It is reliably seen that SCA-CNN fundamentally beats best in class visual consideration based picture inscribing techniques.

T. Yao et al: In this paper, authors present Long Short-Term Memory with Attributes (LSTM-A) a novel engineering that coordinates ascribes into the effective Convolutional Neural Networks (CNNs) additionally Recurrent Neural Networks (RNNs) picture subtitling system, via preparing them in a start to finish way. Especially, the learning of characteristics is fortified by coordinating between property relationships into Multiple Instance Learning (MIL). To consolidate credits into subtitling, Author develop variations of designs by taking care of picture portrayals and properties into RNNs in various manners to investigate the shared yet additionally fluffy connection between them. Broad analyses are led on COCO image subtitling dataset and our system shows clear upgrades when contrasted with cutting edge profound models.

X. Yang et al: Author propose Scene Graph Auto-Encoder (SGAE) that consolidates the language inductive inclination into the encoder decoder image subtitling structure for more human-like subtitles. Instinctively, we people utilize the inductive inclination to make collocations and logical deduction in talk. For instance, when we see the connection ”individual on bicycle”, it is normal to supplant ”on” with ”ride” and surmise ”individual riding bicycle on a street” even the ”street” isn’t clear. In this way, misusing such inclination as a language earlier is required to help the regular encoder-decoder models more outlandish overfit to the dataset predisposition and spotlight on thinking.

M. Cornia et al: In this work, Author propose an image subtitling approach in which a generative intermittent neural organization can zero in on various pieces of the information image during the age of the inscription, by abusing the molding given by a saliency forecast model on which parts of the picture are remarkable and which are logical. Authors show, through broad quantitative and subjective tests for enormous scope datasets, that our model accomplishes better execution with deference than subtitling baselines with and without saliency and to various best in class approaches consolidating saliency and subtitling.

M. Yang et al: In this paper, author present ”MLADIC”, a novel Multitask Learning Algorithm for cross-Domain Image Subtitling. MLADIC is a perform various tasks framework that all the while upgrades two coupled targets through a double learning component: image inscribing and text-to-picture combination, with the expectation that by utilizing the relationship of the two double undertakings, we can upgrade the picture inscribing execution in the target area. Solidly, the picture inscribing task is prepared with an encoder-decoder model (i.e., CNN-LSTM) to create printed depictions of the info pictures. The picture blend task utilizes the contingent generative ill-disposed organization (CGAN) to integrate conceivable pictures dependent on text depictions.

X. Xiao et al: Author propose novel Deep Hierarchical Encoder-Decoder Network (DHEDN) is proposed for picture inscribing, where a profound progressive structure is investigated to isolate the elements of encoder and decoder. This model is able to do productively applying the portrayal limit of profound organizations to intertwine significant level semantics of vision and language in creating inscriptions. In particular, visual portrayals in high degrees of deliberation are at the same time considered, and every one of these levels is related to one LSTM. The base most LSTM is applied as the encoder of printed inputs. The use of the center layer in encoder-decoder is to upgrade the interpreting capacity of top-most LSTM. Moreover, contingent upon the presentation of semantic upgrade module of picture highlight and dispersion consolidate module of text include, variations of structures of our model are built to investigate the effects and shared collaborations among the visual portrayal, literary portrayals and the yield of the center LSTM layer. Especially, the system is preparing under a fortification learning technique to address the presentation predisposition issue between the preparation and the testing by the arrangement slope enhancement.

J. H. Tan et al: Late works in image subtitling have demonstrated very promising crude execution. In any case, we understand that the majority of these encoder-decoded style networks with consideration don’t scale normally to huge jargon size, making them hard to utilize on implanted framework with restricted equipment assets. This is on the grounds that the size of word and yield inserting networks develop relatively with the size of jargon, antagonistically influencing the conservativeness of these organizations. To address this impediment, this paper presents a shiny new thought in the space of picture inscribing. That is, author tackles the issue of conservativeness of picture

inscribing models which is heretofore unexplored. Proposed model, named COMIC, accomplishes tantamount outcomes in five basic assessment measurements with state-of-the-workmanship approaches on both of the MS-COCO and InstaPIC1.1M datasets.

X. Li et al: In this paper, author propose a structure dependent on scene charts for picture inscribing. Scene charts contain plentiful organized data since they portray object elements in pictures as well as present pairwise connections. To use both visual highlights and semantic information in organized scene charts, we extricate CNN highlights from the jumping box counterbalances of article elements for visual portrayals, and concentrate semantic relationship highlights from significantly increases (e.g., man riding bicycle) for semantic portrayals. After acquiring these highlights, we acquaint a various leveled attention based module with learn discriminative highlights for word age at each time step. The test results on benchmark datasets show the predominance of our strategy contrasted and a few cutting edge strategies.

Z. Zhang et al: this paper proposes another model dependent on the Fully Convolutional Network (FCN)- LSTM system, which can create a consideration map at a finegrained lattice astute goal. Additionally, the visual component of every network cell is contributed simply by the chief article. By embracing the matrix shrewd marks (i.e., semantic division), the visual portrayals of various framework cells are associated to one another. With the capacity to go to huge territory "stuff", our strategy can additionally sum up an extra semantic setting from semantic marks. This technique can give thorough setting data to the language LSTM decoder. In this way, a component of fine-grained and semantic-guided visual consideration is made, which can precisely interface the significant visual data with each semantic significance inside the content. Shown by three trials including both subjective also, quantitative examinations, our model can produce inscriptions of high caliber, explicitly significant levels of precision, culmination, what's more, variety.

M. Tanti et al: In this paper, author empirically show that it is not especially detrimental to performance whether one architecture is used or another. The merge architecture does have practical advantages, as conditioning by merging allows the RNN's hidden state vector to shrink in size by up to four times. Our results suggest that the visual and linguistic modalities for caption generation need not be jointly encoded by the RNN as that yields large, memory-intensive models with few tangible advantages in performance; rather, the multimodal integration should be delayed to a subsequent stage.

III. Proposed Methodology

A. *Methodology to solve the task*

The task of image captioning can be divided into two modules logically – one is an image based model – which extracts the features and nuances out of our image, and the other is a language based model – which translates the features and objects given by our image based model to a natural sentence.

For our image based model (viz encoder) – we usually rely on a Convolutional Neural Network model. And for our language based model (viz decoder) – we rely on a Recurrent Neural Network.

B. *Architecture*

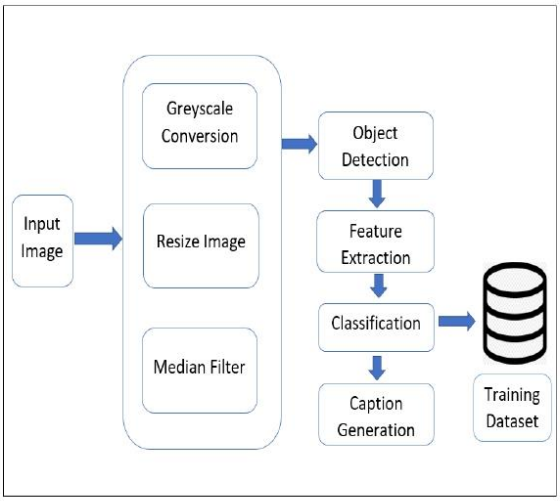


Fig. 1. Proposed System Architecture

Algorithm:

1. Input Image:
Here we will upload the Input Image.
2. Image Pre-processing:
In this step we will applying the image pre-processing methods like gray scale conversion, image noise removal.
3. Image Feature Extraction:
In this step we will applying the image object and edge detection methods to extract the image features from image.
4. Image Classification:
In this step we will applying the image classification methods
5. Result:
In this step will show the final generate caption result

convolutions neural network (CNN) to extract the visual features, and uses a recurrent neural network (RNN) to translate this data into text

C. Algorithms

1. CNN(Convolution neural network) Convolution Layer

Convolution is the first layer to extract features from an input image (image). Convolution preserves the relationship between pixels by learning image features using small squares of input data. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters i.e. identity filter, edge detection, sharpen, box blur and Gaussian blur filter.

Pooling Layer

Pooling layers would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains important information.

Fully Connected Layer

In this layer Feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model.

Softmax Classifier

Finally, we have an activation function such as softmax or sigmoid to classify the outputs.

2. Text pre-processing This is used to process the descriptions given in the data set. In this task, every annotated caption is converted into vector. This can be done using some pre-trained models like Word2Vec, Embedding, etc.

3. Recurrent neural network Recurrent Neural Network(RNN) are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.
Steps:

Suppose there is a deeper network with one input layer, three hidden layers and one output layer. Then like other neural networks, each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for second hidden layer and (w3, b3) for third hidden layer. This means that each of these layers are independent of each other, i.e. they do not memorize the previous outputs.

- A single time step of the input is provided to the network.
- Then calculate its current state using set of current input and the previous state.
- The current ht becomes ht-1 for the next time step.
- One can go as many time steps according to the problem and join the information from all the previous states.
- Once all the time steps are completed the final current state is used to calculate the output.
- The output is then compared to the actual output i.e the target output and the error is generated.
- The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained.

Formula for calculating current state:

$$h_t = \int (h_t - 1, x_t)$$

where,
ht=current state
ht-1=Previous state
xt= Input state
Formula for applying Activation function:

$$h_t = activation(w_h h h_t - 1 + w_x h x_t)$$

where,
whh= Weight at recurrent neuron
wxh= Weight at input neuron

Formula for calculating output:

$$y_t = w_h y h_t$$

where,
yt=Output
why=Weight at output layer

D. Sample Image Caption generation

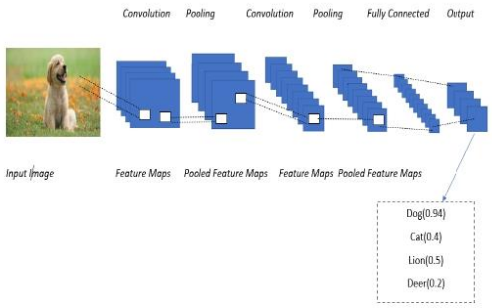


Fig. 2. Caption Generation Process

In Fig 2, we can see by using CNN we extract the features and objects of the image and by using RNN we can map this caption to the the database and give the output which has greater vector in this case suggested output is Dog(0.94).

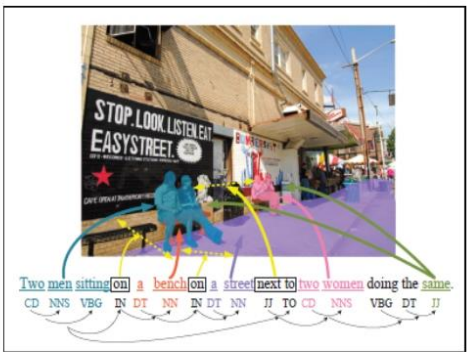


Fig. 3. Caption Generation with object and features extraction

In fig 3, Image Object defection is done with different color and then it converted to the Features and according the features caption is suggested.

E. Dataset

We evaluate the effectiveness of the proposed method on two standard image captioning datasets, Flickr30k and COCO caption dataset. Each image has five reference captions. Each dataset is split into 3 parts: training set, validation set, and test set. For the Flickr30k dataset, its training set contains 29783 images, and the validation set and test set contain 1000 pictures respectively. For the COCO caption dataset, We follow the Karpathy splits, so that the training set contains 113,287 images, and the validation set and test set each contains 5000 images.
Dataset consist of images and its 5 relevant caption. PFB Examples

Fig. 4. Dataset Sample image 1

1.Several men in hard hats are operating a giant pulley system .
2.Workers look down from up above on a piece of equipment .
3.Two men working on a machine wearing hard hats .
4.Four men on top of a tall structure .
5.Three men on a large rig .

Fig. 5. Dataset Sample image 2

Fig. 6. Dataset Sample image 3

IV. Conclusion

In this paper, we propose a novel deep neural network(NDNN) model to improve the image captioning methods. The NDNN explores the spatio-temporal relationship in the visual attention and learns the attention transmission mechanism through a tailored LSTM model, where the matrix-form memory cell stores and propagates visual attention, and the output gate is reconstructed to filter the attention values. Combined with the language model, both of the generated words and the visual attention areas obtain memory in the space. We embed the NDNN model in three classical attention-based image captioning frameworks, and adequate experimental results on the MS COCO and Flicker dataset demonstrate the superiority of the proposed NDNN.

References

[1] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3242–3250.
[2] P. Anderson et al., "Bottom-up and top-down attention for image captioning and visual question answering," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6077–6086.
[3] L. Chen et al., "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5659–5667.
[4] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 4904–4912.
[5] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 10685–10694.

[6] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Paying more attention to saliency: Image captioning with saliency and context attention," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 14, no. 2, p. 48, 2018.
[7] M. Yang, W. Zhao, W. Xu, Y. Feng, Z. Zhao, X. Chen, and K. Lei, "Multitask learning for cross-domain image captioning," IEEE Transactions on Multimedia, vol. 21, no. 4, pp. 1047–1061, 2018.
[8] X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan, "Deep hierarchical encoder-decoder network for image captioning," IEEE Transactions on Multimedia, 2019.
[9] J. H. Tan, C. S. Chan, and J. H. Chuah, "Comic: Towards a compact image captioning model with attention," IEEE Transactions on Multimedia, 2019.
[10] X. Li and S. Jiang, "Know more say less: Image captioning based on scene graphs," IEEE Transactions on Multimedia, 2019.
[11] Z. Zhang, Q. Wu, Y. Wang, and F. Chen, "High-quality image captioning with fine-grained and semantic-guided visual attention," IEEE Transactions on Multimedia, vol. 21, no. 7, pp. 1681–1693, 2018.
[12] M. Tanti, A. Gatt, and K. P. Camilleri, "Where to put the image in an image caption generator," Natural Language Engineering, vol. 24, no. 3, pp. 467–489, 2018.

CPGCON 2021 Certificate:

Sinhgad Institutes

Sinhgad Technical Education Society's

SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING, PUNE

DEPARTMENT OF COMPUTER ENGINEERING

In Association with

BOARD OF STUDIES IN COMPUTER ENGINEERING

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

Savitribai Phule Pune University Pune

cPGCON-2021

10TH POST GRADUATE CONFERENCE FOR COMPUTER ENGINEERING

CERTIFICATE OF APPRECIATION

THIS IS TO CERTIFY THAT

Aditi Palwe

HAS PRESENTED THE PAPER TITLED

"Image Caption Generation using CNN and RNN Models"

IN "cPGCON-2021" HELD ON 7TH & 8TH MAY 2021 AT SKNCOE, PUNE

Dr. V. S. Deshmukh, Coordinator, cPGCON-2021

Dr. V. V. Kimbahune, Coordinator, cPGCON-2021

Dr. Parikshit N. Mahalle, Convener, cPGCON-2021, Member BoS Comp Engg, SPPU, Pune

Dr. Varsha H. Patil, Chairman, BoS Comp Engg, SPPU, Pune

Dr. K. R. Borole, Patron, Vice- Principal, SKNCOE, Pune

Dr. A. V. Deshpande, Patron, Director STES & Principal SKNCOE, Pune

Assessment sheet of CPGCON 2021

Sinhgad Institutes

SINHGAD TECHNICAL EDUCATION SOCIETY'S
SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING, PUNE-411041
DEPARTMENT OF COMPUTER ENGINEERING
IN ASSOCIATION WITH
BOARD OF STUDIES IN COMPUTER ENGINEERING
SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
" cPGCON - 2021 "
10TH POST GRADUATE CONFERENCE FOR COMPUTER ENGINEERING
May 7th - 8 th, 2021

Savitribai Phule Pune University

Evaluation Sheet

Date:7th May 2021

Paper ID : 155	**Session No :**		**Date : 7 May 21**
Name of the student : Aditi Palwe			**College Code:**
Title of Paper : Image caption generation using CNN and RNN models			
Name of Session Chair : Ranjit Gawande			
Evaluation by Session Chair			
Sr. No	**Title**	**Max. Marks**	**Marks by Session chair**
21.	**Relevance of Title, Abstract & Keyword**	**5**	**3**
22.	**Literature Survey**	**10**	**7**
23.	**Implementation Details** **g. Software requirement specification(5)** **h. Mathematical modeling & Design(5)** **i. Implementation Status(10)**	**20**	**12**
24.	**Algorithms – Measures & Metrics** **k. Performance measures used(5)** **l. Result tables(10)** **m. Comparison with similar systems(5)** **n. Efficiency calculations(5)** **o. Outcome & Success definition of work(5)**	**30**	**23**
25.	**Concluding Remarks (Result Discussion, Conclusion & Future scope)**	**5**	**3**
26.	**References (Journal/Conference/Recent)**	**5**	**4**
27.	**Format- Organization of Paper, Clarity & Linguistic quality**	**5**	**3**
28.	**Organization of content & Presentation skills**	**5**	**4**
29.	**Questions & Answers**	**5**	**4**
30.	**Contribution of Authors: To society at large/Technology/Research(Publication or Patents)/Interdisciplinary approach/Use of modern architecture & tools**	**10**	**6**
Total		**100**	69
Remarks/Suggestion: Scope for adding impacting features of image to improve accuracy of labeling			

Performance Satisfactory / Non–Satisfactory: Satisfactory

(Name and Sign) Ranjit Gawande
(Name and Sign)
Session Chair 1

Session Chair 2

JOURNAL OF OPTOELECTRONICS LASER ISSN: 1005-0086

Volume 41 Issue 7, 2022

Image Captioning using Efficient Net

Aditi Palwe[*1], Sankirti Shirvale[2] & Swati Shekapure[3]

*[*1]ME Student, Computer Engineering Department Marathwada Mitra Mandal's College of Engineering, Pune, Maharashtra, India*

[2]Assistant Professor, Computer Engineering Department Marathwada Mitra Mandal's College of Engineering, Pune, Maharashtra, India

[3]Assistant Professor, Computer Engineering Department, Marathwada Mitra Mandal's College of Engineering, Pune, Maharashtra, India

ABSTRACT

Image Captioning intends to produce a sound and thorough portrayal that sums up the contents of a picture. The description generator uses the encoder decoder deep learning model that elaborates the image in text format. Describing the details of an image is one of the challenging and explored areas of Artificial Intelligence. Traditional approaches cannot handle the complexity and difficulties of image captioning as well as deep learning-based approaches. In this paper we propose EfficientNetB3 deep learning framework for caption generation on complex objects. First, we go through several current image captioning approaches, with an emphasis on deep-learning-based approaches and how they are used for pre-processing. Then CNN module implementation for feature extraction and GRU has been used for generating the caption for respective images. The various cosine similarity and N-Gram feature extraction techniques have been used for generating the blue score for the entire testing dataset and show the effectiveness of the proposed system.

Keywords: Convolutional Neural Network, GRU-Gated Recurrent Unit, image caption generation, Natural Language processing

I. INTRODUCTION

Convolutional neural nets i.e., CNN are usually pre trained on a particular picture dataset, which uproots visual attributes from those pictures. This helps the defined model to learn on new data in a much swift way. Elaborating an image in a text format with genuine grammatical syntax is still a demanding problem in the field of computer vision and natural language processing. Comprehending a scene, which combines the expertise of computer vision with NLP, involves image caption, which automatically generates ethical English language descriptions based on the information seen in a frame.

Using natural languages to automatically describe the content of pictures has a lot of promise. It may, for example, aid visually challenged individuals in comprehending the content of online pictures. It may also offer more accurate and concise images information in situations like social media image sharing or video surveillance systems.

The technique produces image captions that are typically semantically informative and grammatically accurate by acquiring information from image and caption pairings. Natural languages are used by humans to describe scenes. Machine vision systems, on the other hand, characterize the scene by capturing an image that is a two-dimensional array. The concept is to combine the image and captions into one area and then learn a mapping from the image to the words.

II. LITERATURE SURVEY

Retrieval-based image captioning is one of the most popular techniques used in early work. Retrieval-based techniques generate a caption for a query image by obtaining one or more sentences from a pre-specified sentence pool. The produced caption may be a statement that already exists or one that is made out of the recovered sentences. Let's start by looking at the line of study that utilizes recovered phrases as image captions.

Convolutional Neural Network models, according to [1] have played a significant influence in this image. Here, we attempt to demonstrate and highlight several techniques for image feature extraction, as well as how they will be used for caption generation. In the area of data science, there has been a lot of research towards improving image caption generating models. Natural language processing is crucial in producing a description that is accurate and has meaning. The favored network in description construction is the recurrent neural network (RNN), which is mostly utilized for sequence generation. Researchers have put in a lot of time and effort to create massive databases. The MSCOCO dataset, which is supplied by Microsoft, is one of the most well-known datasets. The Flickr 8K, Flickr 30K, PASCAL, and a few more are additional well-known and benchmark datasets.

Captioning pictures, according to [2] is the act of creating descriptive information about visual objects, image metadata, or things that exist in a picture. The material inside the pictures is very useful from the perspective of computer vision. They aid a machine's comprehension and performance. Image captioning has a variety of uses, including editing software suggestions, virtual assistants, image indexing, accessibility for visually impaired people, social networking, and a variety of other natural language processing applications. The process of image annotation has been accomplished at a higher level and has contributed to different areas via different methods of deep learning solution. People have come up with a variety of creative ways to approach this application using deep learning. It has been shown that deep learning models are capable of achieving optimal outcomes in the area of caption generating issues based on these findings. It's critical to comprehend not just what the items in the image are, but also how they relate to one another, in order to produce high-quality image descriptions. Encoder–decoder models are now regarded as one of the most advanced image captioning methods.

A lot of information is stored in an item. Huge amounts of image data may be produced each day on social networking sites, including astronomical objects, but this is an up with the quick thing. Annotating pictures of people takes longer, and the chances of making a mistake are higher. Deep learning models are utilized to construct such pictures correctly, removing the need for human adjustments [3]. By eliminating the requirement for human participation, this would substantially decrease human failure and effort. The development of image annotations has numerous real-world benefits, ranging from assisting the mentally challenged to assisting the automated, cost-effective marking of images shared online every day, guidelines for processing software, useful for smart devices, image encoding, visually disabled people, social networking sites, and a variety of other natural literature.

Mason and Charniak utilize visual similarity to obtain a collection of captioned pictures for a query image [4] to mitigate the effects of noisy visual estimates in techniques that rely on image retrieval for image captioning. They then estimate a word probability density conditioned on the query image using the captions of the retrieved pictures. The term probability density is used to assess existing captions in order to choose the one with the highest score as the query's caption. This technique has implicitly assumed that there is always a phrase that is relevant to a query picture. In reality, this assumption is seldom accurate. Instead of directly utilizing returned sentences as descriptions of query pictures, recovered sentences are used to construct a new description for a query image in another line of retrieval-based research.

Li et al. utilize visual models to extract semantic information from pictures, including objects, characteristics, and spatial connections [5]. Then, for encoding recognition results, they construct a triplet of the type adj1, obj1, prep, adj2, obj2. To provide a description for the triplet, web-scale n-gram data is used to conduct phrase selection, which may give frequency counts of potential n-gram sequences. This allows for the collection of potential phrases that may make up the triplet. Following that, phrase fusion is used to utilize dynamic programming to discover the most suitable collection of phrases to serve as the query image's description.

Mitchell et al. use computer vision algorithms to analyze and describe images using triplets of objects, activities, and spatial connections [6]. Then, based on the visual recognition findings, they construct image description as a tree-generating process. The authors select the image contents to describe by grouping and ranking object nouns. Then, for object nouns, sub-trees are constructed, which are then utilized to construct complete trees. Finally, a trigram language model is used to choose a string from the complete trees produced as the image's description.

Template-based image captioning may provide syntactically accurate sentences, and the descriptions produced by these techniques are typically more appropriate to the image contents than retrieval-based descriptions. Template-based approaches, on the other hand, have certain drawbacks. Because description creation in the template-based architecture is tightly limited to image contents identified by visual models, there are generally limits on coverage, originality, and complexity of produced sentences due to the generally small number of visual models accessible. Furthermore, as compared to human-written captions, utilizing strict templates like sentence core structures would make produced descriptions seem less natural. Another kind of technique that is frequently employed in early image captioning work is template-based. Image captions are produced using template-based techniques using a syntactically and semantically restricted approach. In order to produce a description for an image using a template-based approach, a collection of visual ideas must typically be identified first. The identified visual ideas are then linked to form a phrase using sentence templates, particular language grammar rules, or combinatorial optimization techniques [7].

Long-term dependencies are known to be challenging for recurrent neural networks to learn. To address this flaw in image captioning, Chen and Zitnick suggest that a visual representation of an image be dynamically built while a caption is produced, allowing long-term visual ideas to be recalled throughout the process [8]. To this aim, a collection

of latent variables Ut1 is added to convey the visual interpretation of previously produced words Wt1. The likelihood of producing the word wt using these latent factors is shown below:

$$P(wt,V \mid Wt-1,Ut-1) = P(wt \mid V,Wt-1,Ut-1)P(V \mid Wt-1,Ut-1),$$

Wt1 indicates produced words (w1,..., wt1), and V indicates observed visual characteristics. The authors implement the aforementioned concept by incorporating a recurrent visual hidden layer u into Recurrent Neural Networks. The recurrent layer u is useful for predicting the next word wt as well as reconstructing visual characteristics V from previous words Wt1.
A completely convolutional localization network design is used in dense captioning [9], which consists of a convolutional network, a dense localization layer, and an LSTM [20] language model. The dense localization layer takes an image and analyses it in a single, efficient forward pass, inferring a set of areas of interest in the picture. As a result, unlike Fast R-CNN or a complete network of Faster R-CNN, it does not need external region suggestions. The localization layer's working concept is similar to that of Faster R-CNN.

Another dense captioning technique suggested by Yang et al. [10] may overcome these challenges. First, it deals with an inference process that is based on the area's visual characteristics as well as the anticipated captions for that area. This allows the model to identify a single suitable location for the bounding box. Second, they use context fusion to combine context information with visual characteristics of the relevant area in order to give a comprehensive semantic description.

Vinyals et al. [11] proposed the Neural Image Caption Generator technique (NIC). This technique uses a CNN for image representations and an LSTM for image caption generation. The output of the final hidden activations of CNN is provided as an input to the LSTM decoder in this particular CNN, which uses a novel batch normalization technique. This LSTM can retain track of things that have been described in text before. The maximum likelihood estimation method is used to train NIC. The initial state of an LSTM contains image information. According to the current time step and the prior concealed state, the following words are produced. This procedure continues until the sentence's end fragment is reached. Because image data is only supplied at the beginning of the process, it may have vanishing gradient issues. The importance of the words produced at the start diminishes as well.

Wang et al. [12] presented a deep bidirectional LSTM-based approach for producing semantically and contextually rich image captions. A CNN and two distinct LSTM networks are included in the proposed design. It learns long-term visual-language interactions by combining previous and future context knowledge.

The major problem with sequential models is that they often provide overgeneralized expressions that lack information seen in the input picture. To address this issue, Tian et al. [13] proposed a hierarchical framework for image captioning that investigates both the compositionality and the sequential nature of natural language by selectively attending to different modules corresponding to unique attributes of each object detected in an input image in order to include specific descriptions such as counts and color. Experiments on the MSCOCO dataset revealed that the suggested model outperforms state-of-the-art models across many assessment criteria, while also providing visually interpretable findings.

As a result, in this paper, we examine the relationship between model complexity, as measured by the total number of parameters, and the effectiveness of different CNN architectures on feature extraction for Image Caption Generation using popular CNN architectures that have been used for Object Recognition tasks. We use two widely used Image Caption Generation frameworks: (a) the Neural Image Caption (NIC) Generator introduced in [14] and (b) the Soft Attention based Image Caption Generation presented in [15]. We found that the performance of Image Caption Generation changes when various CNN architectures are used, and that it is not directly linked to model complexity or CNN performance on object identification tasks. We test several versions of ResNet [16] with varying depths (number of layers in the CNN) and complexity: ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, where the numerical component of the name stands for the number of layers in the CNN (such as 18 layers in ResNet18 and so on).

The system replaces linear multiplication with convolution, then combines it with global video feature maps before computing the channel weight of each frame's feature maps using channel attention. To learn spatial-temporal assignment parameters, the weighted feature maps are input into parameter sharing gated recurrent convolutional units (SGRU-RCN, a version of gated recurrent convolutional units) [17], which may be used as weights for aggregating local descriptors to specific cluster centers. Different convolutional recurrent neural networks may readily implement the suggested channel soft attention.

Traditional deep learning-based video captioning models include encoder-decoder architecture and gated recurrent neural networks (RNN), such as long short-term memory (LSTM) or gated recurrent unit (GRU). Each input video sequence is encoded into semantic representations and delivered to a decoder, which generates video captions.

These methods, however, have the following drawbacks:
1. They are unable to provide natural captions for lengthy films including a variety of events.
2. They don't grasp what's going on in the context.

The traditional RNN is insufficient for encoding information in lengthy video sequences with long-term dependencies. Lengthy-term dependence is still an unresolved problem for long sequential data, despite the fact that LSTM or GRU partly handle it. Furthermore, traditional captioning algorithms seldom preserve the contextual information included in a video sequence. Traditional video captioning models only operate for short video clips with basic scenes due to memory limitations in RNN models, and thus are not suitable for lengthy movies with many complex occurrences.
The DNC (differentiable neural computer) evolved from the Turing machine. Turing machines are abstract contemporary computer architectures that demonstrate that with enough external memory and methods, any computation is feasible [19]. The Neural Turing Computer (NTM) was suggested by Google Deep Mind, a system that uses neural networks and external memory to construct a differentiable Turing machine, and an enhanced version of the NTM model, DNC, was presented in a Nature article in 2016.

This technique, however, is restricted to video snippets with brief, static backdrops. With the success of neural machine translation (NMT), the sequence-to-sequence model, an LSTM-based encoder-decoder structure, is now being used for video captioning [20]. They use the pre-trained CNN to get semantic representations of video frames, which they feed into the LSTM encoder to get the final hidden states. The loss function of the LSTM decoder is then optimized for one-step forward prediction to produce following words.

Table 1: Review of literature

No	Technique	Dataset	Extracted Features	Research Gap
1	x-Means clustering algorithms and Neuro Fuzzy algorithm [1]	GSM operation data, 24,900 customers 22 attributes Turkey dataset	Some value-added services and some values added services	System reflects good accuracy on structured dataset only.
2	Naïve Bayes, Decision Tree[2]	European operator 106,405 customers 112 attributes	Contract, usage pattern patterns, and calls pattern	High error rate to detect actual churn due to redundant features.
3	Neural network, Regression [3]	Unknown 129,892 customers 113 attributes	Demographic, Value added, usage pattern	Heterogeneous dataset is tedious to handle in similar patterns.
4	Neural network, Regression [4]	Unknown, 169 customers 10 attributes	Demographic, Billing data, usage pattern, customer relationship	High space complexity generate in each layer
5	Stepwise variable selection partial least squares [5]	Cell2Cell Dataset 100,000 customers 171 attributes	Behavioral information, Customer care and demographics	Redundant features should be generating a high error rate.
6	Artificial Neural Network [6]	ML Dataset of UCI 2,427 user's information with 20 attributes	Demographics, Usage pattern, Value added services	It works only by defining statically parameters.
7	Binomial logistic regression model [7]	Iranian telco operator 3150 customers 15 attributes	Demographic, call usage pattern, customer care service	Language influence should generate irrelevant features.

8	Generalized additive models (GAM) [8]	Belgian 134, 120 customers 27 attributes	Demographic Usage pattern, bill and payment	High error rate during unknown text prediction.
9	Logistic regression Decision tree [9]	Polish mobile operator 122098 customers 1381 attributes	Demographic, call data records, customer care services	It works with synthetic data only and has a high data reduction rate.
10	Decision tree as well as machine learning [10] algorithms have been used.	Cell2Cell Dataset 100,000 customers 171 attributes	Behavioral data, of customer care and feature information	Behavior information generates the churn possibility, sometimes it generates false ratios.

III. PROPOSED METHODOLOGY

Most image captioning methods include picture feature extraction, object recognition, and caption synthesis components, as described in the introduction. These models boost performance by analyzing the whole picture or selected picture areas. Those characteristics, on the other hand, are irrelevant for verbs, which have no significant relationship with visual characteristics. If a model is trained to produce the word "walk" from an image of a person walking with open legs, it will likewise produce the word "walk" from an image of a person swimming with similarly open legs. Despite the fact that the proper answer is the word "swim," this is the case. Another example is that a model may find it challenging to produce the words "standing" or "performing gymnastics" from a picture of a person standing inside and performing gymnastics with their hands raised. As a result, in this situation, a standard captioning algorithm is unable to create a verb and instead produces the caption "person in a room." As a result, a model [4] with a motion estimation component has been suggested. When learning the relationship between picture characteristics and verbs is challenging, this model can learn the relationship between motion features and verbs. However, as mentioned in the introduction, background motion elements are used. This flaw may result in two significant problems. To begin, captions mostly explain the contents of the image's objects. As a result, backdrop characteristics aren't as important as they formerly were. When more characteristics in the backdrop are estimated than in the object regions, the captioning model includes more background characteristics, which may have a detrimental effect on caption creation. Second, the motion estimate accuracy is not always great. There are some problems in several of the motion features. As a result, utilizing all of the motion characteristics may lead to additional chances to utilize the incorrect features, lowering caption production accuracy. To address this problem, we only utilize motion characteristics in the image's object area. In the first instance, the caption model may analyze key picture areas for picture captioning explicitly. In the latter instance, the likelihood of the caption model using the incorrect characteristics is reduced.

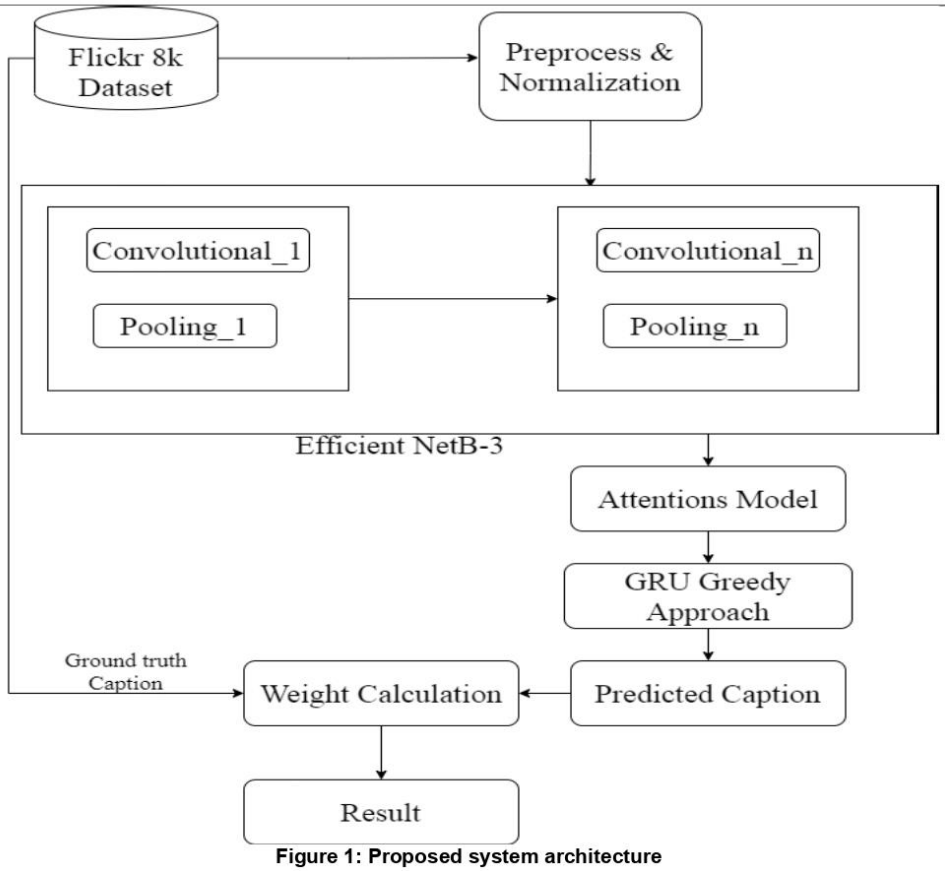


Figure 1: Proposed system architecture

The above Figure 1 depicts the proposed system architecture for caption generation using deep learning models. In the section below we describe each phase of the system to generate the final outcome of the validation set.

Preprocessing and Normalization: The data collection has been done from a synthetic repository called MSCOCOC. The data might be imbalanced with dimension thus it needs to balance before proceeding to CNN. In preprocessing we define fix set of dimensions of image (300*300) and eliminate those objects which are not readable for misclassification. Cross validation has also been done in this similar phase for training and testing. (e.g. 5 fold, 10 fold, 15 fold etc.). In the dataset each image has a caption as ground truth caption, and that is used for analysis. Due to the large size of the dataset, data processing was done on the host computer and the results were saved as pickle files before being sent to the cloud for model training.

Lower case conversion: The dataset texts include words with various letter cases, which presents a challenge for the model since the same words with different capitalizations would be treated differently, resulting in an increase in problem vocabulary and, as a result, complexity. As a result, to prevent this, the whole text must be converted to lowercase..

Punctuation removal: Because the goal of this study is to create descriptive sentences for pictures without using punctuation, the existence of punctuation adds complexity to the issue, which is beyond the scope of this study.

Number removal: The presence of numerical data in the texts is a challenge to the model since it enhances language; thus, it should be eliminated..

Indicate start and end sequence: To signal the beginning and finish of the prediction sequence to the model, the word tokens <start> and<end> are inserted at the beginning and end of each phrase.

Tokenization: The clean text is broken down into component terms, and a dictionary with the full vocabulary is produced for word to index and index to word matching.

Vectorisation: Before learning word indicative vectors from these sequencing, the words in the text data are transcribed using unique numerical interpretations from the word to index dictionary, transforming the cleaned sentence specifications to numerical sequences. Lesser sentences are stretched to the length of the largest sentence sequence to compensate for variable sentence lengths.

CNN using EfficientNetB3: The EfficientNetB3 module has been used for CNN, to extract the features and selection as well. The below code snippet demonstrates the building of CNN with EfficientNetB3. The global image feature vectors for the frames are retrieved from Keras applications using the final fully connected layer of the EfficientNetB3 CNN. Before extraction, the pictures are compressed to 300x300 arrays and the pixels are normalized to a scale of 0 to 255 to suit the EfficientNetB3 model input. A pretrained model is utilized for the picture scene features. To match the input shape of the model, the images are compressed to 224x224 and normalized on a scale of 0 to 255. Both networks output feature vectors are saved as pickle files for simple uploading for validation testing.

Convolution, batch normalization, and ReLU functions are combined in the function D. Convolution is the multiplication of image and kernel matrices element by element. In each mini-batch, batch normalization involves moving inputs with a zero mean and unit variance. The ReLU function is then used to transform the elements with negative values to zero.

$$CNN(I) = D_i\ ([I, f_1, f_2, f_3\ , \ldots., f_{i-1})]$$

Where I denotes the image and f_1 , f_2 and f_{i-1} denote the first, second, and I-1th layer features, respectively. The selected features are forward to the attention layer for generating the captions with GRU.

Attention Model and GRU: attention model basically work for selection of features that received from CNN. The selection features eliminate the redundant and non-essential features. The attention model has divide into four different phases these are described in below,

Encoder: The pre-trained Inception model has already done the visual encoding, thus the encoder is extremely straightforward. It has a Linear layer that receives the pre-encoded views and transmits them to the Decoder.

Sequence Decoder: This is a GRU-based recurrent network. Once passing through a Hidden layer, the captions are sent in as input.

Attention: The Attention module assists the Decoder in focusing on the most important portion of the picture for producing each word in the output sequence.

Sentence Generator: This module is made up of a few Linear layers. It takes the Decoder's output and generates a frequency for each word in the lexicon, as well as for each place in the anticipated sequence.
A recurrent GRU can extract a lot of information from a picture, both temporally and spatially. However, semantic information is required to enhance the features obtained from the encoder in order to produce semantically correct sentences. The Greedy method is utilized in our approach to generate semantic vectors, and a semantic compositional network is used as the decoder. Furthermore, a dual learning method is employed to save the semantic vector's information throughout training, allowing the semantic information in forward flow to be effectively used.

In GRU we applied the greedy approach for caption generation in both training testing. In both module generated captions are evaluated with ground truth sentences for validation and generate the 4 BELU scores and cosine similarity weight for confusion matrix. In the result section we depict the various ground text features extraction methods used for generating the BELU score and similarity weight.

Mathematica Model
Deep Convolutional Neural Network (DCNN)
Input: Test Dataset which contains various test instances TestDBLits [], Train dataset which is built by training phase TrainDBLits[] , Threshold Th.
Output: HashMap <class_label, SimilarityWeight> all instances whose weight violates the threshold score.

Step 1: For each read each test instances using below equation

$$testFeature(m) = \sum_{m=1}^{n} \quad (featureSet[A[i] \ldots\ldots A[n] \quad TestDBLits)$$

Step 2 : extract each feature as a hot vector or input neuron from $testFeature(m)$ using the below equation.

Extracted_FeatureSetx[t……n] = $\sum_{x=1}^{n}$ (t) □ $testFeature$ (m)

Extracted_FeatureSetx[t] contains the feature vector of respective domain

Step 3: create the number of convolutional

For each read each train instances using below equation

$$trainFeature(m) = \sum_{m=1}^{n} \quad (featureSet[A[i] \ldots\ldots A[n] \quad TrainDBList)$$

Step 4 : extract each feature as a hot vector or input neuron from $testFeature(m)$ using the below equation.

Extracted_FeatureSetx[t……n] = $\sum_{x=1}^{n}$ (t) □ $testFeature$ (m)

Extracted_FeatureSetx[t] contains the feature vector of the respective domain.

Step 5 : Now map each test feature set to all respective training feature set

$$weight = calcSim\ (FeatureSetx\ ||\ \sum_{i=1}^{n} \quad FeatureSety[y])$$

The $Train\_Feature[]$ and $Test\_Feature[]$ both required as input for the test classifier when generating similarity score between two input objects. These are two separate attributes which represent the training and testing instance respectively. The Th is the denominator that is used for selection of each epoch layer result. The $T[j]$ denotes j^{th} attributes of testing instance while the $T[k]$ depicts k^{th} train attribute information. By using the feature selection method, we extract some features from both instances and forward to a similarity measurement function which is described in step 5. The number of validated instances by threshold is the dense optimized results by CNN.

IV. RESULTS AND DISCUSSIONS

The datasets used are benchmark datasets that are used to assess the system's performance. Flickr8k, Flickr30k are all massively preferred benchmark datasets for the application of image captioning. The dataset COCO is an acronym for common objects in context, and it is a dataset created by Microsoft. It is massive in size and covers over 91 kinds of photos, with over 164k photographs in total. For training sets of 330K photos and 1.5 million object instances, MS COCO uses iconic images. The Flickr8k holds 8000 photos, whereas the Flickr30k has roughly 30,000. Each image within these datasets is labeled or mapped with five captions. These libeling's can also help with system activities. Based on previous experiments around1000 photographs are utilized for testing and validation purposes in both Flickr datasets. Transfer learning can be used to implement these models. Models are trained on high-configuration GPUs that can be found online as well as those on the system we have used, the GTX 1650. The training period varies depending on the GPU; for such applications, systems with larger Gigabyte GPUs, such as 4-16 GB, are desirable, and software such as Python is required for the tensorflow libraries, with IDEs for python such as Jupyter Notebook or PyCharm or Visual Studio Code with libraries such as the tensorflow, numpy, keras, pandas and few others are required.

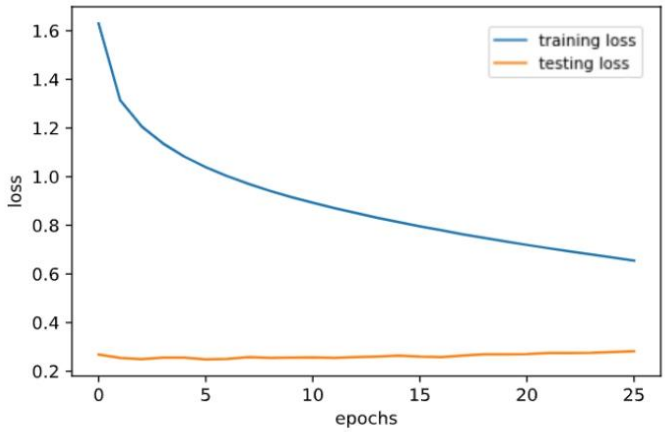


Figure 2: Model performance on training

The above Figure 2 describes training and testing model performance with proposed EfficientNetB3 on Flickr8k dataset. With a count of 6000 pictures validated in the training and a 1000 for testing set. When system deals with high epoch values it reduces the overall loss in training while it slightly increases or constant for validation.

Figure 3: Model performance testing with various feature selection techniques on test image 1

Figure 4: Model performance testing with various feature selection techniques on test image 2

Figure 3 and Figure 4 describe the caption generation in the validation phase for two different images. The generated caption has been evaluated with both ground truth sentences and evaluated with 4 BELU functions and Bigram's methods with cosine similarity algorithm. According to both results we conclude the default cosine similarity produces higher accuracy over the other methods.

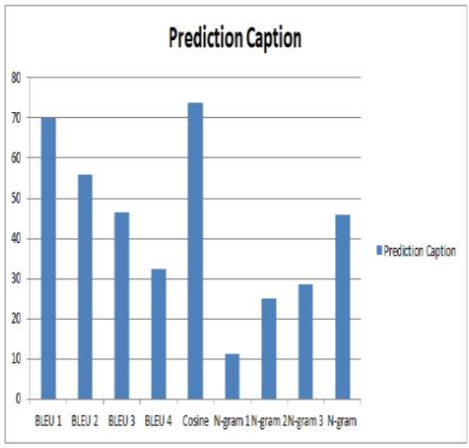


Figure 5: Prediction accuracy for caption generate with different techniques on test image

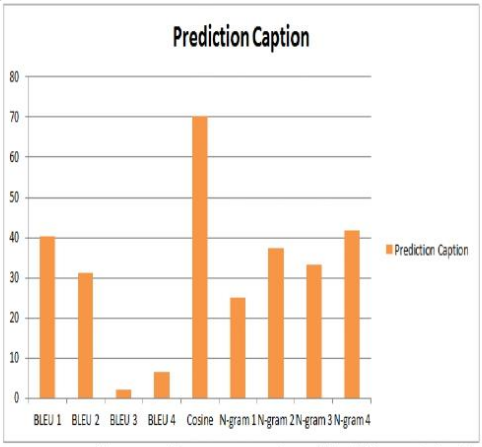


Figure 6: Prediction accuracy for caption generate with different techniques on test image

The above Figure 5 and 6 all weighted accuracy for both objects evaluated in Figure 3 and 4. BELU-3 and N-Gram generate lower results than other techniques thus cosine similarity produces high caption generation accuracy for the system. It produces around 75% accuracy for cosine similarity algorithms. In another experiment we have done comparative analysis with CNN-LSTM and CNN-Bi-LSTM [21] as existing systems for caption generation. The below Table 1 demonstrates similarly.

Method	BLEU-1	BLEU-2	BLEU-2	BLEU-2	ROUGE-L	Cosine
CNN-LSTM [21]	46.7	31.32	19.4	9	25.7	NA
CNN-Bi-LSTM [21]	55.00	34.9	24.8	13.1	27.9	NA
Proposed	**70.00**	**55.77**	**46.47**	**32.46**	NA	**73.78**

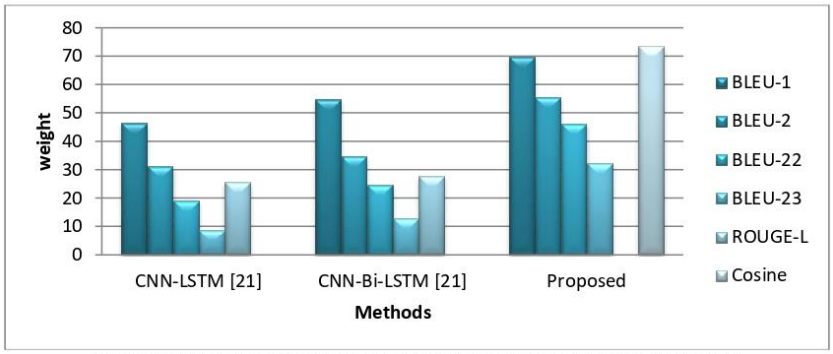


Figure 7 : Experiment analysis and comparative analysis of system

According to this Figure 7 we conclude our system predicts superior results in terms of all performance parameters.

V. CONCLUSION

We show how deep learning techniques are used to generate image captions in this study. We employed CNN models with GRU to obtain an accurate description of the picture. The ideas of the generalized Encoder – Decoder model have been discussed. Its application, as well as the attention model and how it was applied in various techniques. We also discussed several datasets that can be used to evaluate a system by using them as benchmark datasets. Finally, we discussed assessment metrics and how to use them to evaluate our model. This research focused on the image captioning task, which is a fundamental problem in artificial intelligence. For extraction, the CNN Efficient Net B3 module was employed, and GRU was used to create captions for the images. To obtain the blue score for the complete testing dataset and demonstrate the performance of the proposed system, several cosine similarity and N-Gram feature extraction algorithms were applied.

REFERENCES

[1] Karahoca, Adem, and Dilek Karahoca. "GSM churn management by using fuzzy c-means clustering and adaptive neuro fuzzy inference system." Expert Systems with Applications 38.3 (2011): 1814-1822.
[2] Kirui, Clement, et al. "Predicting customer churn in the mobile telephony industry using probabilistic classifiers in data mining." International Journal of Computer Science Issues (IJCSI) 10.2 Part 1 (2013): 165.
[3] Ballings, Michel, and Dirk Van den Poel. "Customer event history for churn prediction: How long is long enough?." Expert Systems with Applications 39.18 (2012): 13517-13522.
[4] Ismail, Mohammad Ridwan, et al. "A multi-layer perceptron approach for customer churn prediction." International Journal of Multimedia and Ubiquitous Engineering 10.7 (2015): 213-222.
[5] Lee, Hyeseon, et al. "Mining churning behaviors and developing retention strategies based on a partial least squares (PLS) model." Decision Support Systems 52.1 (2011): 207-216.
[6] Burez D, den Poel V. Handling class imbalance in customer churn prediction. Expert Syst Appl. 2009;36(3):4626–36.
[7] Brandusoiu I, Toderean G, Ha B. Methods for churn prediction in the prepaid mobile telecommunications industry. In: International conference on communications. 2016. p. 97–100.
[8] He Y, He Z, Zhang D. A study on prediction of customer churn in fixed communication network based on data mining. In: Sixth international conference on fuzzy systems and knowledge discovery, vol. 1. 2009. p. 92–4.
[9] Idris A, Khan A, Lee YS. Genetic programming and adaboosting based churn prediction for telecom. In: IEEE international conference on systems, man, and cybernetics (SMC). 2012. p. 1328–32.
[10] Huang F, Zhu M, Yuan K, Deng EO. Telco churn prediction with big data. In: ACM SIGMOD international conference on management of data. 2015. p .607–18.
[11] Oriol Vinyals ,Alexander Toshev, ,Samy Bengio ,and Dimitru Erhan. Show and tell: A neural image caption generator. Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3156–3164, 2015.
[12] Kyunghyun Cho, Bart Van Merrinboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. Association for Computational Lin- guistics, pages 103–111, 2014.
[13] Junjiao Tian and Jean Oh. Image captioning with compositional neural module networks. Proceedings of the Twenty-Eighth Interna- tional Joint Conference on Artificial Intelligence (IJCAI-19), pages 3576–3584, 2019.
[14] Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. "Show, attend and tell: Neural image caption generation with visual attention." In International conference on machine learning, pp. 2048-2057. 2015.
[15] Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge." IEEE transactions on pattern analysis and machine intelligence 39, no. 4 (2016): 652-663.
[16] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
[17] Xu, Y.; Han, Y.; Hong, R.; Tian, Q. Sequential Video VLAD: Training the Aggregation Locally and Temporally. IEEE Trans. Image Process. 2018, 27, 4933–4944.
[18] Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef]
[19] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for tatistical machine translation. arXiv 2014, arXiv:1406.1078.
[20] Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112
[21] Agughalam D, Pathak P, Stynes P. Bidirectional LSTM approach to image captioning with scene features. InThirteenth International Conference on Digital Image Processing (ICDIP 2021) 2021 Jun 30 (Vol. 11878, p. 118780B). International Society for Optics and Photonics.