Scene Understanding

Image Description In Natural Language

By

Imran Khurram

00000172237

Supervisor

Dr. Muhammad Moazam Fraz

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science

Approval

It is certified that the contents and form of the thesis entitled "Scene Understanding: Image Description in Natural Language" submitted by Imran Khurram have been found satisfactory for the requirement of the degree.

Advisor:

Dr. Muhammad Moazam Fraz

Signature: ____________________

Date: ____________________

Committee Member 1:

Dr. Asad Waqar Malik

Signature: ____________________

Date: ____________________

Committee Member 2:

Dr. Muhammad Shahzad

Signature: ____________________

Date: ____________________

Committee Member 3:

Dr. Omar Arif

Signature: ____________________

Date: ____________________

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Imran Khurram

Signature: ____________________

Acknowledgment

I pay my gratitude to Allah almighty for blessing me a lot and without His guidance I couldn't complete this task. I am thankful to Dr. Muhammad Moazam Fraz for guiding and encouraging me to complete my thesis. His timely and efficient contributions helped me to shape the thesis into its final form and I express my gratefulness for his sincere supervision all the way.

I am also thankful to Department of Computing, and the teachers for providing me with an academic base, which enabled me to complete this thesis.

I'd like to thank my parents who have provided me every comfort of life and their utmost affection and support made me reach where I am today.

Publications Arise from the Thesis

- British Machine Vision Conference (BMVC).
- Pattern Recognition Journal

Table of Contents

List of Abbreviation

RPN	Region Proposal Network
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory Network
RNN	Recurrent Neural Network
IAPR	International Association of Pattern Recognition
TC-12	Technical Committee 12
MSCOCO	Microsoft Common Objects in Context
R-CNN	Regions with Convolutional Neural Network
RoI	Regions of Interest
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent

List of Tables

List of Figures

Abstract

Automatic image captioning is an active and highly challenging research problem in computer vision aiming to understand and describe the contents of the scene in human understandable language. Existing solutions for image captioning are based on holistic approaches where the whole image is described at once, potentially losing the important aspects of the scene. To enable, more detailed captioning, we propose Dense CaptionNet, a deep region-based modular image captioning architecture, which extracts and describes each region of the image individually to include more details of the scene in the overall caption. The proposed architecture consists of three main modules to describe the image objects. The first one generates region descriptions which not only includes objects but object relationships as well and the second one generates the attributes related to those objects. The textual descriptions generated by these two modules are fused in a text file to provide as input for the subsequent sentence generation module which works as an encoder-decoder framework to merge and form a single meaningful and grammatically correct sentence which is detailed enough to describe the whole scene in a better way. The proposed architecture is trained using Visual Genome, IAPR TC-12 and MSCOCO datasets and tested on un-seen set of IAPR TC-12 dataset because of detailed nature of its descriptions. The trained architecture out-performs the existing state-of-the-art techniques e.g., Neural Talk and Show, Attend and Tell, using standard evaluation metrics especially on complex scenes.

Keywords: *image captioning, computer vision, deep learning, dense image captioning, RNN, visual captioning, scene understanding*

Chapter 1

Introduction

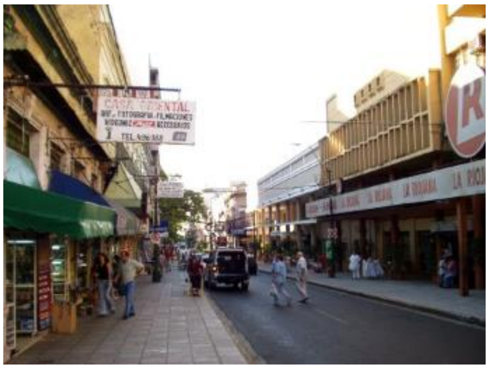
This chapter provides the opening and general information of the research to provide a clear understanding about this thesis. It also covers the problem statement along with solution statement.

1.1 Scene Description:

Human beings have a significant sense of vision which enables them to see around the world. It includes a complete visual pathway which transfers the visual image captured by the retina to the brain through the optic nerve. The image on the retina gets faded very quickly, but the eyes have the ability to transfer the old image before getting the new image at the rate of approx. 30 times per second. The rate enables to provide continuous vision. There is a small round area on the retina where the optic nerve meets it, it is called blind spot as an object's projection on this spot makes the object invisible. The ability of a human to recognize the scene is in the visual cortex that lies inside occipital lobe near the back of the head. The visual cortex has the another incredible ability to fill the gap in the vision produced by the blind spot.

Making a computer/machine to mimic the sense of vision is called computer vision which includes methods for understanding and interpreting real world data and to produce useful information from it. This gives computer/machine the ability to answer and communicate with us in human understandable language. Automatic image captioning is the process to automatically allocate text description to the digital image by analyzing and understanding the contents of the image. A better caption is a one which describes maximum possible objects of the scene so that the description becomes more and more useful.

A scene can be described in multiples lines i.e. a paragraph (see Figure 1). These descriptive paragraphs are very useful to develop an understanding of the image contents. Using these descriptive paragraphs have certain short comings e.g. alternative text of the images on the web needs only one-line description. Similarly, searching similar images on the web can be enhanced by providing metadata including one-line description of the input image. There are many other advantages of one-line descriptions in real life scenarios.



One-Line Caption: a city street filled with lots of traffic

Descriptive Paragraph: two lane street with large shops on the right and smaller shops on the left; people are walking on the sidewalk, some are crossing the street; cars are parked along the left side of the street as well;

Figure 1. One-Line Caption vs. Descriptive Paragraph

1.2 Real World Applications:

Automatic Image captioning can be used to describe any scene in words so it can be used in robotics to make intelligent decisions by including contextual awareness of the scene [1]. It can be used in surveillance to provide smart security using CCTV cameras and alarming systems [2, 3]. Moreover, it can be used by visually impaired people to understand things [4], to understand what is going on around them and to take decisions accordingly. A device containing camera can be assembled such that it can be used whenever a blind person wants to see what's going on around him/her. It can also help them to understand the contents of pictures and images on the web.

Automatic sentence generation can be used in sentence-based image search [5] as a prerequisite. Content based image retrieval (CBIR) works by including several things related to the image among which metadata such as description of the image also plays important role if available. Automatic post creation and hash tag creation [6] can also be done by using this kind of technique. Going a little diverse, natural language description can be used in sequential vision to language generation to make visual storytelling an easy task. It can also be used in visual questioning answering where one can ask question about visually available objects in the scene and model will automatically answer those questions from generated description.

Not only this, natural language description has applications in videos/movies too e.g. linguistic descriptions of the movies can allow blind people to understand and follow the movie along with their partners.

1.3 Scene Description Process:

Traditional methods work by using sentence templates [7, 8] in which there are several hard-coded visual concepts which are filled in the fixed templates to form image caption. These types of captions are short and simple because of which they are not true depiction of the scene. They also limit the variations in the text.

With the availability of large data and computational resources, deep neural networks (which have more than 2 hidden layers) have become an excellent topic of research. Many deep architectures have emerged as top performers in computer vision tasks e.g. image segmentation [9, 10], image classification and recognition [11], etc. In particular, Convolutional Neural Networks (CNNs) have been used for feature extraction in such tasks. CNNs have layers just like ordinary neural network but they perform series of convolution operations on the image while passing from all the layers. CNNs have the capability to extract majority of the visible features which will help in many image processing tasks. Another network that gained popularity is Recurrent Neural Network (RNN) [12]. RNNs have the special ability to pass the value of the hidden state from the current time step to the next time step so that the value does not lost. The special retaining power of the RNN makes it suitable for sequence generation tasks where it is necessary to remember previous values. RNNs are being used in Optical Character Recognition (OCR) systems [13], text generation systems [14] and image captioning systems [15, 16]. Taking into account the usefulness and applications of such deep architectures, there has been a paradigm shift in techniques solving the automatic image captioning problem.

The deep learning procedure to solve this problem involves making understanding of the visual scene using CNNs and language processing model based on RNNs to describe that scene in natural language [15, 16].

The process is based on two parts: understanding phase and describing phase. Understanding phase uses Convolutional Neural Networks which are simply feedforward neural networks having hierarchy of convolutional layers which learn the image features after iterating several times from sample data. CNN can produce image features in a fixed size vector representation which can be further used for description generation. Describing phase uses Recurrent Neural Networks which enables to generate text sequences. RNNs have vanishing gradient and exploding gradient problems which means the value generated on previous layers either become too small or too large so that it makes the parameters to learn very difficult. This problem keeps on becoming big as the number of layers' increases. To generate sequences of text while keeping this problem in consideration, the type of RNN used can be Long Short Term Memory Network also known as LSTM. LSTM has memory cells which enables it to handle vanishing and exploding gradient problem very efficiently. The pipeline can be following:

1. Alignment Model - Language words are mapped to a space along corresponding to the features in training.

2. Generation Model

    a. Encoder – extracting image features to make scene understanding.

    b. Decoder – generating sequence of words by matching features with words in the mapped space.

A simple depiction of how encoder-decoder framework works for image captioning task is shown in Figure 2.

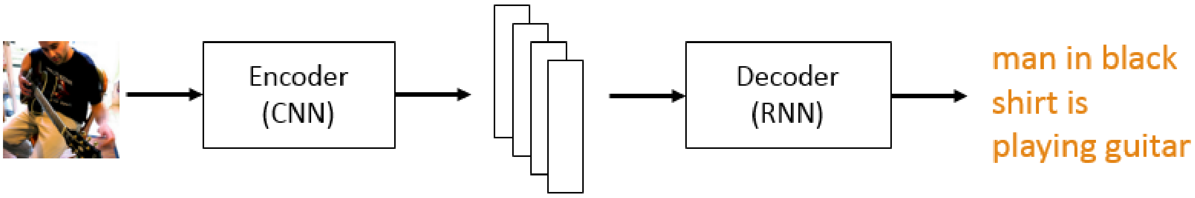


Figure 2. General image captioning mechanism.

Input is an image I. Model is trained to generate sequence of words i.e. $\{S_1, S_2, \ldots\}$ by maximizing probability likelihood $p(S \mid I)$ so that only those words are generated whose likelihood to match the image features is maximum.

1.4 Challenges:

The task is so easy for us (humans) that we can describe any scene by just having a glimpse of it and that all because of the brain, but it is a difficult and challenging task for a machine to understand and describe a scene as it has to find not only the objects but also the relationships between objects, attributes and activities and that too from an image which is merely an array of numbers representing colors at each pixel. These problems are in the understanding phase of the image. There are other problems in describing phase too like keeping the correct grammar and relationships between objects e.g. "boy ride a skateboard", not "skateboard ride a boy" and that too from index of each word. Sequences of words are represented by index number so to represent an image with a natural language description, salient features of the image are annotated to sequence of index numbers.

Most of the research done having state of the art results, matches the features with indexes in the space to create sentences but still it's a challenge to create a sentence which describes most of the objects of the scene in a dense manner e.g. the description generated for a scene having many objects like shown in the Figure 3 fails to describe the scene completely because it is very difficult to find and annotate relationships on nearly all objects.

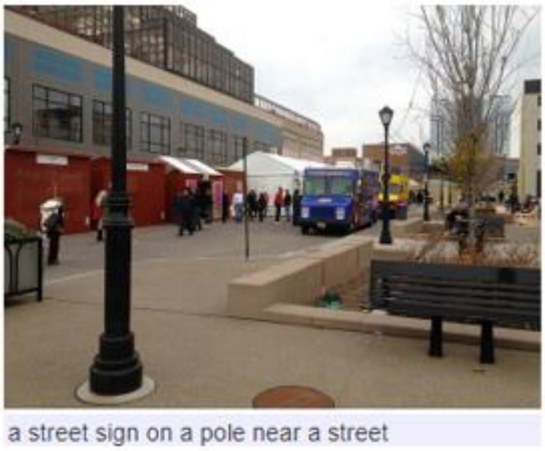


Figure 3. Less descriptive caption generated by image captioning system

The existing approaches takes whole image into consideration while creating the caption. They extract features of the whole image and then generate the caption of those features. Individual objects are not considered separately before caption generation which do not allow the in-depth description generation. To detect each object of the scene and include that in the final description is the most challenging task to be handled. Not only the objects and their relationships, the attributes of each object should also be considered so that appearance of individual objects should also be presented in the final output. As of now, these types of contextual information have not been employed while solving the problem of image captioning.

1.5 Thesis Contribution

This research proposes a novel approach for image captioning whose significant contributions are:

- Object based image captioning architecture which can also be used for region description generation instead of full image captioning.
- Sentence generation module which is used to join small sentences to one large sentence. As yet, no other tool is available to join sentences.
- The proposed architecture has been evaluated on IAPR TC-12 dataset [17]. Empirical results demonstrate that the architecture out-performed existing state-of-the-art image captioning methodologies in all evaluation metrics.

1.6 Thesis Organization

The thesis is organized as follows. **Error! Reference source not found.** delivers a detailed l iterature review about the relevant state-of-the-art image captioning methodologies. Chapter

3presents proposed architecture consisting of three major modules i.e. RPN based region extraction to detect and extract regions of interest, RNN based region description generation and attribute generation blocks and the sentence generation block based on RNN encoder-decoder framework. Chapter 4discusses the available and generated datasets. Chapter 5 presents the results obtained on these datasets and detailed analysis of the system performance. At the end, the Chapter 6provides conclusion and the future research directions.

Chapter 2

Literature Review

Automatic image captioning is being long studied task in computer vision. Captions are being generated by some image processing and natural language methods. Neural networks in machine learning were invented long before but in recent years, with the large amount of data and high processing power being available, machine learning methods surpass the capability of those methods which use combination of image processing and natural language. Based on this fact, the approaches of automatic image captioning can be divided into two types:

1. Sentence templates based methods
2. Machine learning based methods

2.1 Sentence templates

Sentence template based methods use combination of image processing and natural language processing techniques [7, 8]. These methods are based on the fact that how human describe something they see in the world. Such information includes: 1) information that how the world is described visually and 2) data about how human construct the natural language sentence to explain that visually described thing. This data is used as a training data for the system to learn how the construction should be.

The methodology implied by one of these techniques uses Conditional Random Field (CRF). Input image is processed with object detectors to find the candidate objects. Each of these objects is then processed with attribute detectors/classifiers to find their attributes. After that these same objects are then processed with prepositional relationship functions to find relationships between objects. Finally, CRF is constructed which have unary image representations and text based representations got from large corpora of documents. Labelling is generated from the graph and then sentences are generated using that labelling.

Major problem with these techniques is that they rely on document corpora or some kind of templates and visual concepts that are hard coded. These hard coded things make such techniques very limited and very hard to make variations.

These techniques are also imposing limit on a complex scene to be described in a simple and single sentence.

2.2 Machine Learning Methods

Machine learning techniques were invented a long before but they came into light when there is availability of high computing power at a low rate as well as large amount of data for training. With machine learning came into more practice, automatic image description using machine learning was also worked on.

Methodology which made the basis for all such techniques uses Convolutional Neural Network (CNN) as Encoder and Recurrent Neural Network (RNN) as Decoder [15].

Neural networks are biologically inspired networks and convolutional neural network is a type of neural network which is deep and feed forward and is used specifically for visual images. These are based on the property of convolution that convolving a window/kernel on the image gives different desired results based on your window/kernel. CNNs are a made by a hierarchy of convolutional and other layers. CNNs usually contains fully connected layers at the end of the network. The representations computed by CNN are as follows:

$$\text{if} \qquad y = Wx + b \tag{2.1}$$

$$\text{then} \qquad r = W(CNN(I) + b) \tag{2.2}$$

where r is the representation of the image, I is the input image, b is the bias and W is the weight matrix having dimension of $h \times 4096$ and h is embedding space height in which the image is going to be embedded.

Recurrent neural networks are created to overcome the limitation of simple neural networks which are unable to remember any value. Recurrent neural networks have loop connection used to get the value of previous timestamp. RNNs are used in sequence generation and similar tasks. RNN have varnishing/exploding gradient problem. Vanishing gradient problem is that a value is too small then after each occurrence it got included to the next small value and the overall value gets too small on each iteration, similarly exploding gradient means the value is too large that it keeps on becoming large with each iteration. The problem is very severe that it makes hindrance in the training of the network. Long short term memory (LSTM) networks [12] very introduced to cater this problem which contains memory cells and different kind of gates to allow or prohibit the passage of values easily. For the task of automatic image captioning, LSTM is a good choice of RNN for description generation.

Task of automatic image captioning uses CNNs to get the image features i.e. representation of input image in compact and precise form whereas it uses RNNs to describe these image features in word representations as shown in Figure 4.

Model is trained to construct description which is grammatically correct and has maximum probability given input image I so as to maximize $p(Description \mid I;\theta)$. θ represents model parameters which we can tuned to maximize the probability of generated description given input image I.

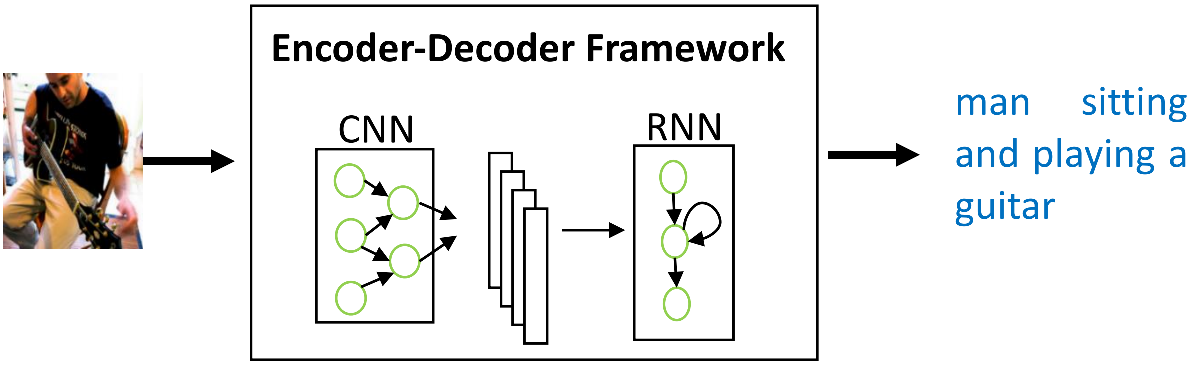


Figure 4. Encoder-decoder framework for image captioning.

2.2.1 Attention Mechanism

To keep the descriptions more accurate, visual attention mechanism [18] was employed for object validation. Visual attention is something that allow to focus on one particular area of image. As an example, consider how human put their gaze on anything making that particular thing focused while keeping the rest of the scene blurred allow them to focus on the details of one thing at a time only, visual attention works in similar way. In Figure 5, graphical representation of the mechanism of visual attention is shown. Object "frisbee" is highlighted in the image showing how the mechanism is putting visual attention on the image.

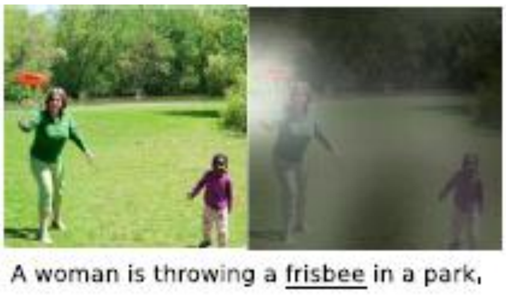


Figure 5. Visualization of Visual attention [18]

Architecture for attention mechanism contains a glimpse sensor. Glimpse sensor works just like our eye. It takes a glance at the image by extracting and taking the particular location of the image in different scales. Then resizing it to the same resolution as shown in Figure 6.

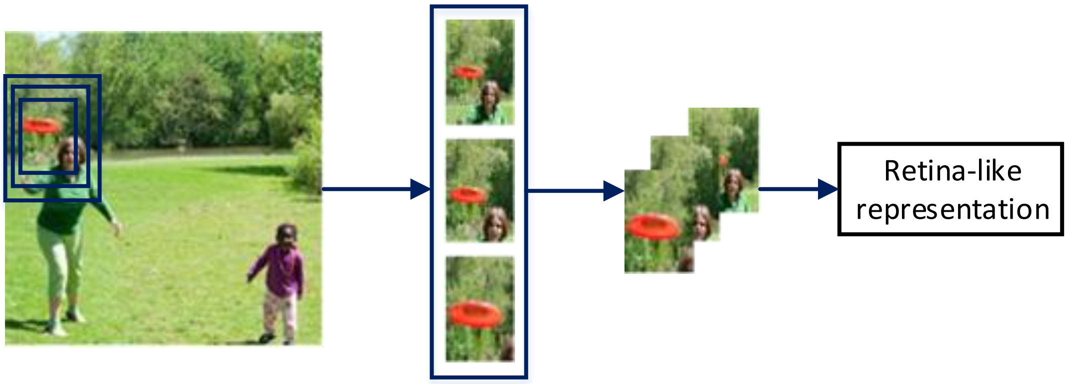


Figure 6. Working of glimpse network.

Glimpse network contains glimpse sensor to get the retina like representation and combine it with glimpse location by using hidden layers and ReLU. The final output of this network is a vector representation of retina representation and its location.

After that Location network uses this vector and tries to calculate and predict the location on where to put the next gaze. Another network called activation network is used to predict the output e.g. a digit (if its MNIST dataset) which is used as reward point. In this way the whole architecture can be trained by using this reward point as a lose.

2.2.2 Review Vectors

Encoder-decoder framework performed very well on image captioning task where encoder encodes the image to a representative vector and decoder generates language sequence for that representative vector but it was a little difficult to encode all the necessary information in a single static representation i.e. a vector so attention mechanism was introduced which made the framework attentive encoder-decoder to make important aspects dynamically be considered. There were still two main problems which needs to be focused i.e. attention mechanism works in sequential way and do not have the capability of global modelling. And the second problem is that certain researches shown that some kind of discriminative supervision is beneficial for automatic image captioning task but there was no known way to include some discriminative supervision into the above described framework. Only supervision that encoder-decoder framework has is generative supervision which aims to maximize the conditional probability described previously. Keeping in view these issues, review network [19] was introduced for image captioning task.

The review network implements a number of review phases with what is called an attention component on the hidden states of encoder which means reviewing all the information that is encoded by the encoder at all stages, and its output is a thought vector after each phase; the generated thought vectors are then passed as an input to the attention component in the decoder. This thought vector is a global representative of all the encoded information and is thus provides a supervision. Thought vector v can be described as:

$$v_t = f_t(H, v_{t-1}) \tag{2.3}$$

where t is the review phase, H is the hidden states, f is the LSTM which includes attention component at review phase t.

2.2.3 Adaptive attention

Attention mechanism performed very well and adapted extensively for automatic image captioning task where regions are highlighted against each word that is generated in a spatial map. The issue here is that attention is not need to be generated and regions are not needed to be highlighted for each word that is generated e.g. “to”, “on”, top” these words do not need to be visualized. The attention generated for such words can be misguiding for caption generation.

Adaptive attention [20] is an encoder-decoder framework which decides whether to generate attention for the word or not. When attention is not generated, it will only rely on language model. The mechanism includes a “visual sentinel” vector which is generated by the LSTM in place of individual hidden state. The vector provides a fallback option to the decoder. There is another unit called sentinel gate, which chooses how much data the decoder required to get directly from the image in contrast to depending on the visual sentinel vector when producing the next word of the caption. For example, the model learns to produce attention on the input image more when producing words like “white”, “animal”, “blue” and “sign”, and depends more on the visual sentinel vector when producing words like “bottom”, “to” and “top”. The decoder LSTM which provides the visual sentinel vector can be shown as:

$$g_t = \alpha(W_i i_t + W_h h_{t-1} + b) \tag{2.4}$$

$$s_t = g_t \odot \tanh(m_t) \tag{2.5}$$

where W_i and W_h are the weights which are learned during training, h is the hidden state of RNN, i is the input given to the RNN(i.e. LSTM) at time stamp t, b is the bias, α is the sigmoid. $\odot$ is the element wise dot product, m_t is the memory cell and g_t is the gate applied on this memory cell.

2.2.4 Multimodal attention

Most of the work before this contains encoder decoder framework having CNN as encoder and RNN as decoder in which CNN has the responsibility of converting the input image into one single feature representation. Multimodal attention mechanism changes this idea of single feature representation.

The real power of the RNN is not utilized before as CNN encodes the visual information of the input image only at the initial stage so the information of only one-time step is available for the whole RNN and the RNN suffers from imbalance of the source and target language as visual information is available of only one-time step whereas language information i.e. words are available for each time step. It also weakens the RNN's memory.

Some approaches recommend to convert objects of the input image into feature vectors, some recommend object-attribute pairs and some recommend to put the gaze on salient regions and generate feature representation.

Multimodal approach [21] converts sequence of objects to feature representation at many time steps. This approach enhances the feature representation power by leveraging the capability of object detection mechanisms so as to encode more feature information. Another benefit of this mechanism is that it balances both sides of the language i.e. source and target. Architecture is shown in the figure.

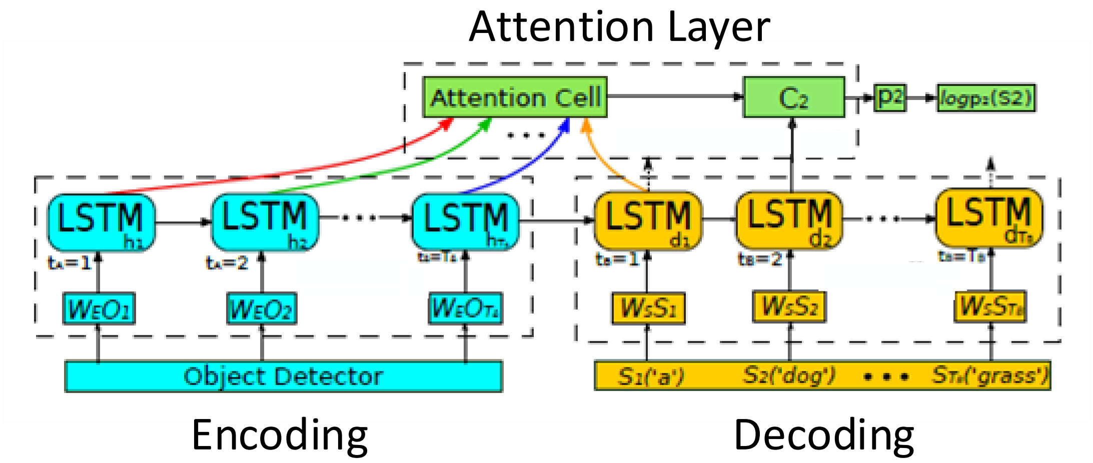


Figure 7. Attention mechanism in Multimodel attentive tranlator [21]

The architecture is such that one object at a time is encoded at each time step and one word at a time is generated at decoding side. The most important module in the architecture is the attention layer which is something different than the visual attention model. The attention layer considers hidden states of the encoder in all time steps to generate one word (at each time step) at the decoder. Advantage of this layer is some objects arrive at early time steps but their language represented word should be generated at the end of the sentence e.g. target sentence is “dog sitting on the table” and object “table” is detected first but in the resulted sentence “table” should be put at the end so the attention layer comes helpful there. Attention layer generate every word at each time step by first looking at all the objects so it will not put “table” at the start.

The model was tested with and without attention layer and the results have shown that addition of the attention layer enhances the caption generation capability.

2.2.5 Personalized Captioning

Apart from general descriptions, automatic image captioning is used for many different tasks e.g. personalized captioning [6]. Social media is used widely all around the world. People capture their personal photos and put on social media by writing captions of the photos in their own personal writing style and vocabulary. Capturing photo requires only one click or tap on the screen but writing the description of that photo explaining the context of the scene, sentiments of the persons in the photo and hashtags to upload it on social media require more time, effort and mental energy. To generate captions of their personal photos, words and writing style should be copied. Personalization issues are handled such that image description is generated taking into consideration user’s used vocabulary in previous documents. This task involves two things: hashtag and associated text generation. Generated text can also contain emoji to show the sentiments. To achieve success in personalized captioning, a new architecture was proposed which is called context sequence memory network (CSMN) [6].

2.2.5.1 Context sequence memory network (CSMN)

CSMN has three major contributions. First, memory is used as a context repository which stores prior knowledge. Multiple types of information are stored in context repository including diverse type of writing styles of the users and huge range of post topics. Second, the model is designed to store all the generated words in a sequential manner. This mechanism acts like attention layer to generate each word by considering all previously generated words. This mechanism also does not let the RNN suffer from vanishing gradient problem as most of the architectures which use previous knowledge suffer from vanishing gradient problem. Third, the model uses CNN to jointly represent nearby memory slots so that context information can be included for better encoding.

Input to the CSMN is a query image I of a particular user and the output is a text sequence $S = \{S1, S2, ...\}$ describing the input image where S_i are the hash tags and the words in the description.

2.2.5.2 Context Memory

Context memory contains three types of data. i) image features for representing input image. ii) user's personalized style for frequently used words by calculating TF-IDF weight from previous posts and iii) word memory of previously generated word in the sequence.

2.2.5.3 Personalized Dataset

Because of the personalization issue, dataset is the most important part. The experiment is done by creating own dataset collected from images from social media, the one specifically used was Instagram. Images with their respective captions and hashtags are stored corresponding to users. A large user data was collected and stored having wide range of topics and personal data. Data is pre-processed in the form of dictionary of most frequent words and dictionary of most frequent hashtags.

2.2.6 Policy and Value Networks

Most of the work in image captioning problem using deep learning is based on encoder-decoder framework. Apart from this, some work has been done using reinforcement learning based methodology. Ren et al. [22] has proposed a reward based architecture consisting of two neural networks: Policy Network and Value Network. Policy network generates the words of the description and value network generates the reward for those generated words based on which the model makes itself better on training.

2.2.6.1 Policy Network

The basic task of policy network is to provide local guidance by generating score of predicting the next word based on already generated words and the image.

Policy network consists of a CNN which is used for feature representation of the input image. An RNN which is used to generate word tokens. It is similar to basic encoder-decoder framework for image captioning. Input image and currently generated words are combinedly called the current state and mathematically represented as:

$$c_t = \{I, w_1, ..., w_t\} \tag{2.6}$$

where c is the current state, I is the input image and w_t is the word generated at time step t.

This current state is used by policy network to give probability of the actions to be taken i.e. to generate next word.

2.2.6.2 Value Network

The task of value network is to provide global guidance and to evaluate all the possible scenarios that can occur following the current state.

Value network contains a CNN to form a visual understanding of the input image, and RNN to form a textual representation of the generated word tokens so far. Both of these generated representations are fed into a Multi-Layer Perceptron (MLP) which generates the prediction of the reward.

2.2.7 Actor-Critic Sequence Training

This proposed approach [23] works on the same principle of actor-critic reinforcement learning. Actor is responsible to generate image descriptions while the purpose of critic is to evaluate and generate per-token reward. This generated reward is used to training the actor. The actor will continue to produce word tokens based on its probability distribution.

Existing architectures do not extract the object level semantic information and therefore lacks in generating the dense descriptions. Incorporating object detection module in image captioning may help in obtaining detailed captions as e.g., done in [24]

2.2.8 Region Description Generation

Objects should be localized and describe in order to describe maximum possible details of the scene. This is the main idea of DenseCap formed by Johnson et al. [24]. They created an architecture Fully Convolutional Localization Network (FCLN) which do not describe the whole scene, instead it describes the parts of the images in separate caption each. It first finds the regions with highest possibility of objects present in them, then it generates descriptions of each region separately. It contains the same CNN and RNN modules, the significant addition is the Localization Layer which is inserted between CNN and RNN to extract regions from the CNN formed feature map of the whole image. Extracted Regions are passed to RNN to generate caption for each region. Main modules of DenseCap are described in following sections.

2.2.8.1 Convolutional Network

VGG-16 is used as CNN to extract the features of the whole image. Its architecture is simple consisting of 13 layers with same 3×3 filters, 2×2 max pooling layers having stride of 2.

2.2.8.2 Localization Layer

Localization layer is the main invention in this architecture. It receives the full feature map of the image and gives three things as output i.e. region bounding boxes, region objectness scores and region features. This layer employees a part of Region Proposal Network (RPN) [25] which was introduced in Faster R-CNN [25]. Faster R-CNN used RoI pooling to extract feature map of each region from the whole image feature map. The problem with RoI pooling is that gradients can only be back propagated to feature map and not to region coordinates. The problem is solved by DenseCap by removing RoI pooling and inserting bilinear interpolation allowing gradients to back propagate to region coordinates of the predicted region proposals.

2.2.8.3 Recognition Network

It is a two layer fully connected network used to flattened the features into one matrix for each region. Recognition network also uses dropout as regularization technique.

2.2.8.4 RNN for language generation

The flattened features in the form of matrix is fed to Recurrent Neural Network (RNN). Long Short Term Memory network (LSTM) is widely used for sequence generation as RNN in caption generation task. LSTM generates a starting token, word sequences and the ending token. The hidden layers used by DenseCap for language generation RNN are 512.

DenseCap used Visual Genome [26] dataset for training of the whole end-to-end system. Visual Genome is a big dataset containing region descriptions, attributes, relationships and visual question answers. Region descriptions are the only requirement for the training of DenseCap.

Chapter 3

Methodology

The focus of this thesis is to consider the object level semantics of the image while creating the overall caption of the image so that the resultant caption is longer and more detailed. For this purpose, the proposed architecture (see Figure 8) first extracts the regions. Regions are something that contains objects and some relationships between those objects. Extracted regions are then passed to two types of modules working on object level details. First is region description generation module which uses the extracted regions as small images to generate captions of those small images separately. Second is attribute generation module which uses extracted regions as objects and finds their object attributes so that those objects can be described in more depth. Both of these modules use Long Short Term Memory (LSTM) network as RNN. Following the work of Andrej et al. [15], both of these modules work on the principle of alignment and generation models. Alignments between the visual data and textual data are made by using training dataset by the Alignment model. Then Generation model generate textual representation by matching visual features of the test dataset in that learned alignment space. The outputs of these two modules are region descriptions and object attributes. These two things are then passed to the last module for proper sentence creation. Sentence creation module contains two LSTMs and an attention mechanism. The final output will be a decent detailed caption.

A worth mentioning challenge here is that the region description generation and attribute generation should be done end-to-end. To elaborate, region descriptions can be generated just like simple conventional encoder-decoder framework but in our case we have split image into regions. To train this module end-to-end we need to modify the RPN so that it supports back propagation and make the end-to-end training possible.

The pipeline of the architecture is: 1) Region extractor, 2) RNN for natural language text generation 3) Sentence Generation. Region extractor using RPN extracts object regions present in the input image and their probabilities of having objects in them i.e. objectness scores. Using these outputs, top scoring regions are selected for further processing. Two textual representations are generated for these regions using RNNs i.e. i) region descriptions which also shows object relationships, ii) object attributes. Both of these textual representations plays significant role in describing the parts of the input image. Both are joined by using sentence generation module.

Sentence generation is done using encoder-decoder framework where both of encoder and decoder are RNNs. First RNN transforms the texts into vector representation i.e. numbers and second RNN is used for sequence generation to form a complete detailed sentence out of it.

We will explore the details of all the modules in the following sub-sections.

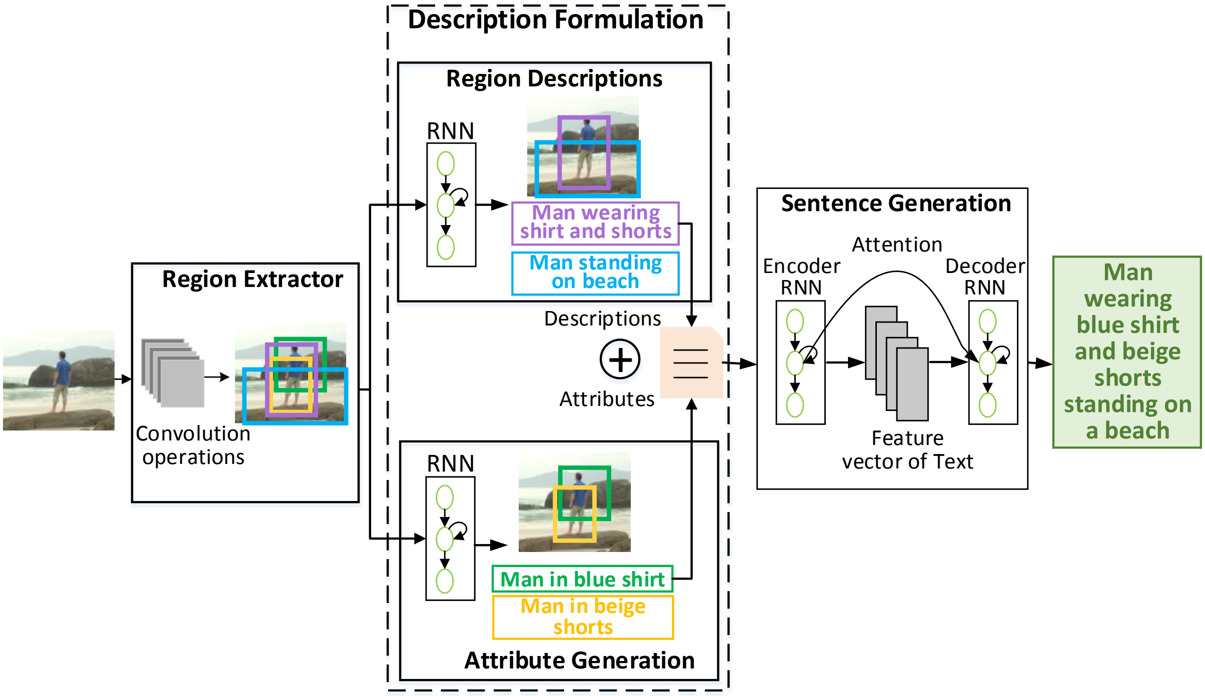


Figure 8. Model overview of CaptionNet

3.1 Region Extractor

Region extractor (see Figure 9) is used to extract region proposals from the original image along with their probability of having object. VGG-16 [27] is used as Convolutional Neural Network (CNN) to extract features from the whole image. These features are then passed to the famous Region Proposal Network (RPN) to extract regions. Extracted regions are compressed in the form of matrix by Recognition network to create final output.

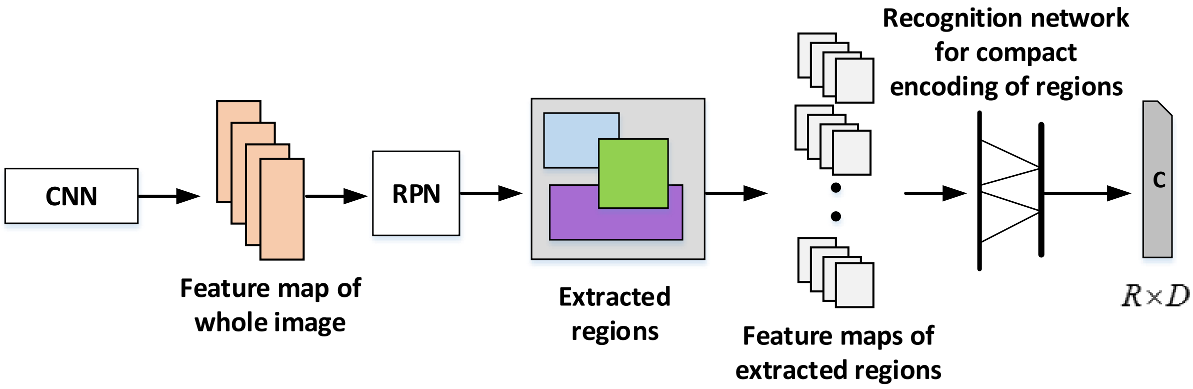

CNN
Feature map of whole image
RPN
Extracted regions
Feature maps of extracted regions
Recognition network for compact encoding of regions
C
$R \times D$

Figure 9. Region extraction block

3.1.1 VGG-16 as Convolutional Neural Network

Visual Geometry Group created very deep convolutional neural network to extract visual features for recognition purposes. We used VGG-16 which performed best among all the variants of VGG. It is the largest sequential structured neural network which gave best results for visual recognition. It is the simplest CNN with the maximum possible depth which stacked the convolutional and max-pooling layers. All layers contain padding and stride of 1 with 3×3 filters. All max pooling layers are also of fixed size i.e. 2×2 having stride of 2. When a colored image of 3 channels represented by dimension of $3\times W\times H$ is passed to this VGG-16 network, it will be converted into feature map of $256\times W'\times H'$ dimension where $H'=H/16$ and the width $W'=W/16$. This is the extracted feature map of overall image without any region detection.

3.1.2 Region Proposal Network (RPN)

Object detection was done by Regions with CNN (R-CNN) [28]. R-CNN uses selective search to extract region proposals. Selective search is an external methodology employed by R-CNN which extracts 2000 region proposals having highest objectness scores (high probability of having object). These region proposals must be converted to a size that CNN can use as input. So these are warped into fixed size for CNN. Region proposals are then fed to CNN to convert into feature map of 4096 dimension. This CNN contains 5 convolutional layers and 2 fully connected layers. Extracted feature map is then passed to Support Vector Machine (SVM) [29] to classify it. This feature map is also fed to bounding box regressor to output the bounding boxes (locations) of the regions. Three different modules i.e. Convolutional Network, SVM, bounding box regressor make the training slow and computationally expensive.

To overcome the computational speed problem, Fast R-CNN [30] was introduced. Fast R-CNN moved the convolutional part to the start of the procedure. It first used convolutional network to find the feature map of the overall image at once. It then uses this feature map and the regions (extracted by the same method) as input to the RoI pooling layer to compute feature vector of fixed length. Each feature vector is then passed through fully connected layers to fed into two separate modules. One is the same bounding box regressor to find 4 values i.e. bounding boxes and the other module is softmax which computes the probabilities of the classes which will be used for classification.

Pipeline of both R-CNN and Fast R-CNN is still too much complex consisting of multiple modules and one external selective search too. To make the pipeline and the training process a little less cumbersome, Faster R-CNN [25] was introduced. Faster R-CNN reduced much of the complexity and increased speed by using a special network for region extraction called Region Proposal Network (RPN) instead of selective search. Because of RPN, this architecture removed the need to use any external method for region extraction and it just extracts regions by having a look on the image. After region extraction the same pipeline works here too. Regions are passed to RoI pooling layer then FC layers and then passed to two different branches i.e. softmax and bounding box regressor to output classification results (probabilities of classes) and coordinates of bounding boxes of the regions.

Now let's explore the most important module of Faster R-CNN which is closest to our need for region extraction i.e. Region Proposal Network (RPN).

Extracted feature map of the first module i.e. CNN is fed into RPN. This feature map is converted into low dimensional feature map by sliding a small network over it. The resulting feature map is passed to two fully connected layers. First is regression layer which is used to extract the bounding boxes and the second is classification layer used to classify it. One low dimensional feature map can have maximum k possible region proposals which the inventor called anchors. As the creator by default used 3 aspect ratios and 2 scales so the value of k will be 9. Regression layer will give $4k$ output where 4 represents 4 coordinates of the bounding box. Similarly, the classification layer will give $2k$ output representing 2 probabilities of object and not an object. The final output of the regression layer will be 36 and of the classification layer will be 18 until the default values are changed. The most important property of this anchor based approach is that it is translation invariant. If any object is translated in the original image the corresponding anchors will also get translated and the same architecture will be able to detect it without any changes.

Anchors are then regressed to form region proposal coordinates and objectness scores by passing through 3×3 convolution having 256 filters, ReLU and 1×1 convolution having $5k$ filters so if $k=9$ then there are 45 filters.

Regions are filtered to get only R=256 best region proposals in which P=256/2 are positive region proposals and 256-P are negative region proposals. This filtration will remove the unnecessary region proposals to reduce the amount of computations that the next modules will have to do. Positive region proposals are extracted by putting a condition over intersection over union (IoU) that if IoU is greater than or equal to 0.7 and similarly negative region proposals are those whose IoU is less than 0.3. This process is called Box Sampling.

The best selected region proposals are converted to same sized feature map as they were of different sizes or different aspect ratios. This conversion process is done using ROI pooling layer by the inventors of Faster R-CNN. ROI pooling layer takes feature map of the whole image and the best selected region proposals as input to find and extract the feature map of those proposals from the whole image feature map. On the training stage, ROI pooling layer back propagates to only feature map of the original image but it does not back propagate to region proposals. To overcome this limitation and to make the system train end-to-end, ROI pooling mechanism is replaced by bilinear interpolation just like done by DenseCap [24]. Hence, the ROI pooling layer can now back propagate the weights to the coordinates of proposals.

Bilinear interpolation also involves a sampling grid. Sampling grid is a linear function of the best selected region proposals which will allow the weights to be back propagated to the region coordinates of the proposals which in turn makes the end-to-end training easy.

Samples are sent to sampling grid, after that these are given as input to the bilinear sampler. The second input to the bilinear sampler is convolution feature map of the original image to give region features as output for further processing.

Input feature map F having $V \times W' \times H'$ dimension, and a region proposal is passed to bilinear interpolation [24, 31] which will create output feature map M having dimension $V \times X \times Y$ for which sampling grid of shape $X \times Y \times 2$ is created that is used to link values of M to coordinates of F. Sampling kernel used for this purpose is s given as follows:

$$s(d) = \max(0, 1 - |d|) \tag{3.1}$$

For all the region proposals the overall output will be of dimension $R \times V \times X \times Y$ after computing from bilinear interpolation.

3.1.3 Recognition Network

Recognition network is an extra layer of refinement for the objectness scores and positions of the regions. It is a simple two layer fully connected network containing rectified linear unit and dropout technique for regularization. Flattened regions are passed into the recognition network to output a code of dimension $D = 4096$ which is simply an encoded version of the visual representation of region in a compact form. For all regions the matrix will be of the dimension $R \times D$.

3.2 RNN for language generation

RNN based language generation was done on the same principles as done by many previous researches [32]. As RNN has the capability to remember the generated token on previous and next time steps. RNN suffers from vanishing and exploding gradient problems [33] where the gradient values become too small or too large for increasing time steps. As a solution for this problem, LSTMs [12] are introduced which uses memory cell as a unit. Memory cells have different types of gates which controls the flow of the gradients. After the advent of LSTM, many kinds of researches used it as an RNN. It also performed really well for sequence generation tasks. We used LSTM as RNN because of its such capabilities.

If the training sequence containing tokens of words is $t_1, ..., t_L$, we give input to the LSTM $L+2$ tokens i.e. $a_{-1}, a_0, a_1, ..., a_L$ in which a_{-1} represents the encoded region features which is encoded by the recognition network, a_0 represents the START token used to indicate the starting of the sequence generated by the LSTM, the rest of the sequence tokens are the word tokens got by encoding t_x where $x = 1, 2, 3, ..., L$. The LSTM works as a function given below:

$$h_t; b_t = f(h_{t-1}; a_t) \tag{3.2}$$

This is the LSTM recurrence formula in which h represents the hidden states, a is input and b is output at time step t. Generated output sequence will contain tokens for time steps $0, 1, ..., L-1$ and also an END token at time step L. There are 512 tokens and hidden states. On testing stage, the LSTM is fed with a_{-1} and the next most likely token is sampled and fed to the LSTM for next time step. Similar process is repeated again and again until the said END token is generated.

We trained two LSTMs separately with same structure and working. First is Region description generator module which is trained on region descriptions for the objects and second is Attribute generator module trained on object attributes. We will now have a look on these modules one by one.

3.2.1 Region description generator

The description of each region taken out by region extractor is generated by region description generator. Region description generator contains an LSTM trained on region descriptions. Region extractor extracts regions in the form of bounding boxes along with their probabilities of having objects in them. This output of the region extractor serves as input to the region description generator. This module works on the concept that any part of the image is also a complete image in itself and can be described in a complete human understandable sentence. Specially the regions having high objectness scores can be described well because it provides the surety that there is something describable and it's not a background or anything useless. These sub-images (part of the original image) not only contains objects but also object relationships because there can be multiple objects in the bounding box and there can be multiple bounding boxes which are overlapping (see Figure 10). This object level information can be used to describe each object separately. These all separate object descriptions will be useful for the description of the whole image. Dataset used for region description generator LSTM is Visual Genome [26] which contains many regions with their region descriptions for each image. Dataset for region description generator along with other datasets used is discussed in Section 4. Output of this module will be textual representation in the form of complete and grammatically correct sentences of the sub-images. All these region descriptions will be used by joining them while creating full image caption.

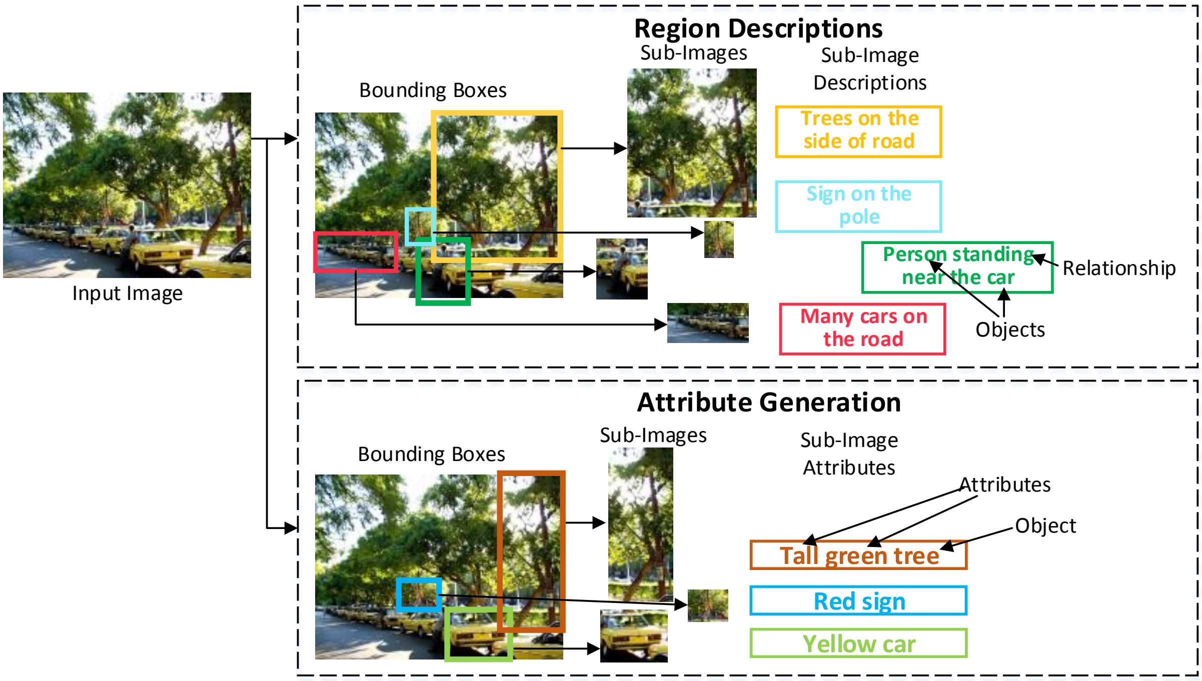


Figure 10. Region description and attribute generation

3.2.2 Attribute generator

Attribute generator is used to recognize object attributes present in the scene. Attribute generator also works on the same principles. It also contains an LSTM. Difference is that it is trained on object attributes and not on region descriptions. Region descriptions show the relationships among the objects and the actions in the sub-image whereas attributes describe the object itself e.g. red sign, yellow car. These attributes play significant role in detailed image captioning. As the outer details are covered by region descriptions and the object details are covered by object attributes. Output regions of the region extractor acts as sub-images which are fed into attribute generator which detects and generates the attributes explaining the objects in the sub-images. Dataset for attribute generator along with other datasets used is discussed in Section 4. Output of this module will be textual representation in the form of complete and grammatically correct sentences of the sub-images containing both attributes and object names in each sentence (see Figure 10). All the attributes of the sub-images will be joined along with region descriptions to form a complete and detailed image caption.

3.3 Sentence generator

Generated region descriptions and attributes should now be joined to form a single detailed caption but there is no tool available till now that can even join the words to form a sentence. So for this purpose, we formed sentence generator which will be used to join attributes and region descriptions to form a detailed image caption. Sentence generator is developed using the basics of sequence-to-sequence (seq2seq) models [34-36]. Sequence-to-sequence models are used in various tasks e.g. automatic language translation, text summarization, text generation, speech recognition etc. Our sentence generator works similar to the language translator so we will first see how the language translation is done by the machines.

Traditional method for machine translation was phrase based machine translation [37-39] in which complete sentence is broken down in to multiple parts and then each part is translated phrase by phrase. When these translated parts are joined the whole structure of the sentence is not like human generated sentence. So the best method would be to work just like human beings, they do not break the sentence, they read the full sentence and translates it completely after understanding the meaning. Human do not translate word by word, they convert the sentence to its meaning in their mind and then convert that meaning to the translated sentence. The same principal works in our case. If we just join the words, it will not be a well-structured sentence and will look meaningless. So we have to generate an understanding of all the region descriptions and attributes and generate the whole sentence from scratch using that understanding. This concept is adopted here in encoder-decoder based framework Figure 11.

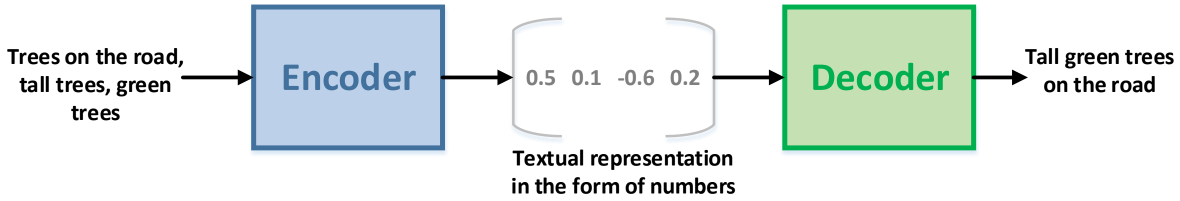

Trees on the road, tall trees, green trees
Encoder
0.5 0.1 -0.6 0.2
Textual representation in the form of numbers
Decoder
Tall green trees on the road

Figure 11. Sentence Generator Basic Structure

Basic sentence generator is shown in the Figure 11, Encoder consumes the input sequence of text and generates textual understanding of it in the form of "thought" vector which is basically numbers. This "thought" vector is then consumed by decoder to form a meaningful and complete sentence. This encoder-decoder framework is similar to how human join the words by making a kind of representation.

The best choice for sequence generation and handling is Recurrent Neural Network (RNN) which is used by most of the seq2seq models. RNN is used both as encoder and decoder. There are different types of RNNs available e.g. vanilla RNN [40], Long Short Term Memory Network (LSTM) [12], Gated Recurrent Unit (GRU) [41]. RNNs also have a choice of different combinations of depth and direction.

For sentence generation we used unidirectional LSTM with multi layers. More configuration will be discussed afterwards. The system will use two LSTMs, the encoder one just makes the number encoding and decoder one will generate target sentence while making prediction for the next word. More about the structure can be studied from Luong [42].

3.3.1 Word Embedding

To find and generate the text representation, the model first finds the source and target embeddings which will correspond to the actual word representations. A vocabulary is first needed for both source and target sentences. Vocabulary of maximum V words is selected which are most frequent and the rest of the words are ignored making them <UNK> token i.e. unknown. We have generated the vocabulary by collecting all the unique words in the source sentences and similar vocabulary for target is also generated from target sentences. Weights are learned during training phase which will be used later on testing phase.

3.3.2 Encoder-Decoder

The retrieved word embeddings are passed to the encoder-decoder framework as input. The framework contains RNNs for both encoding the source sentence and decoding for the target sentence. The encoder is initialized with zero vectors while the decoder should know the information about the source sentence which is why its starting state is initialized with the ending hidden state of encoder as shown in Figure 12. This process is similar to vanilla seq2seq architecture which is good for small sized sentence generation but for complex scenarios having larger sentences, passing only a single hidden state to the decoder is not enough because a single state represents a very little information. To avoid this information bottleneck, we used attention mechanism.

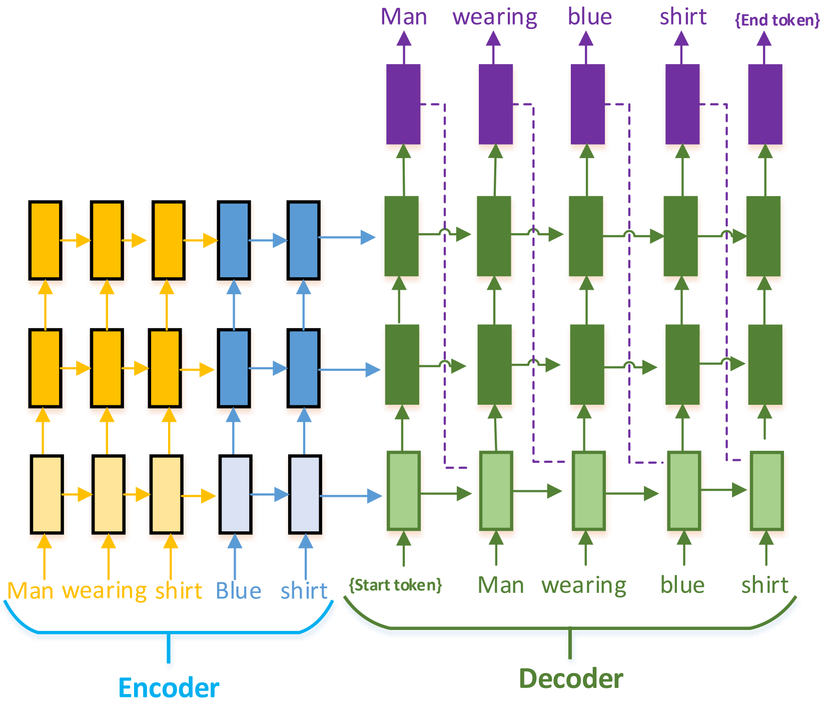


Figure 12. RNNs based encoder decoder framework

3.3.3 Attention Mechanism

Attention mechanism helps to improve the quality of large sentences. The attention mechanism (Figure 13) needs two types of inputs on each hidden state. First, the values of all the hidden states are passed to the current hidden state. Second, the resulting value of last attention is also passed to current hidden states so that decision made should be affected by last decision. Further details of how the attention mechanism works are:

1. Comparison is made between current hidden state of the decoder with all the encoder hidden states to calculate what is called the attention weights.
2. These attention weights are used to calculate weighted average which is called context vector in this scenario.
3. This context vector is then combined with the value of current hidden state of the decoder part to find the attention vector which is the final output of the attention mechanism.

4. The attention vector is then provided as input to the decoder in the next time step. This process is called input feeding.

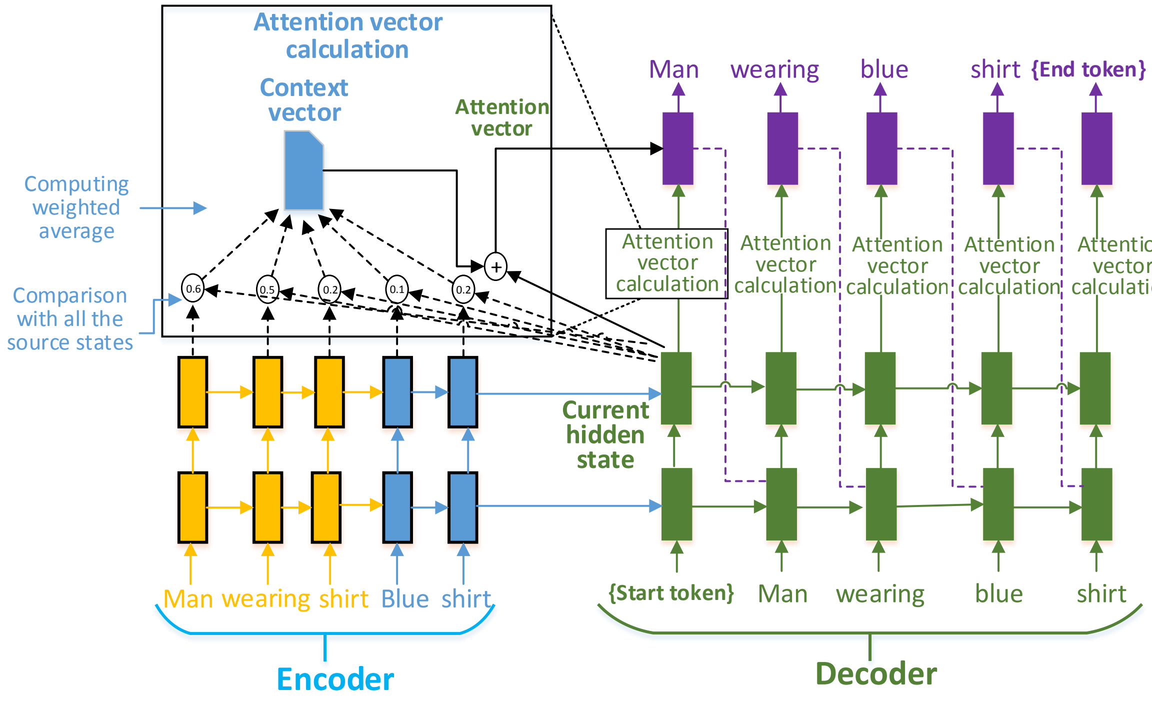

Attention vector calculation
Context vector
Attention vector
Computing weighted average
Comparison with all the source states
0.6
0.5
0.2
0.1
0.2
+
Man
wearing
blue
shirt
{End token}
Attention vector calculation
Attention vector calculation
Attention vector calculation
Attention vector calculation
Attention vector calculation
Current hidden state
Man wearing shirt
Blue
shirt
{Start token}
Man
wearing
blue
shirt
Encoder
Decoder

Figure 13. RNNs based encoder decoder framework with attention mechanism

As shown in Figure 13, all the hidden states of the encoder are fed to the current hidden state of decoder and the feeding of the attention vector to the next time step is basically informing the model about the previous attention decision. Both of these things are important for the attention mechanism.

The attention based encoder-decoder mechanism is able to produce detailed and decent sentence descriptions by joining region descriptions and object attributes as shown by the empirical results in Chapter 5.

3.4 Model training and optimization

Both the region description and attribute generation modules are combined individually with region extraction module to form an encoder-decoder formulation for training purposes. Both of them are trained for 50,000 iterations with 512 units at each hidden layer of LSTMs in region description and attribute generation modules. Regularization is done using dropout of 0.5 to reduce any kind of overfitting.

Optimization technique used for CNN training is stochastic gradient descent (SGD) with a learning rate of 1×10^{-6} whereas optimization for the training of region and attribute descriptions modules has been done using adaptive moment estimation [43].

Weights pre-trained on ImageNet [11] has been used to initialize the CNN to enhance its classification power. It was further fine-tuned after 1 epoch with visual genome [26] dataset by freezing first four layers of the network because initial layers represent features of basic shapes e.g. edge, line etc. which can be better extracted using the same weights learned from ImageNet [11] dataset.

Sentence generation module contains two LSTMs (encoder and decoder) consisting of 200 units in each of their 2 hidden layers which are trained for 20,000 iterations. Regularization is done using a dropout of 0.2. Stochastic gradient descent has been used for optimization while training with a learning rate of 1.0. We have used high learning rate to short the time for training as in text training the model usually converges fast, while in feature training the learning rate needs to be low otherwise the loss will not decrease much and the model does not converge to local minima.

Batch size for the training of region and attribute description blocks is 1 as the input image is further converted into region proposal bounding boxes. The mini-batch for the training of sentence generation module consists of 128 natural language sentences.

Chapter 4

Materials

There are several datasets available for the task of image captioning e.g. MSCOCO [44], Flickr8k [45], Flickr30k [46], Pascal 1k [7]. For our modular approach, we needed dataset for each module separately. For object region descriptions, we used Visual Genome [26] dataset. Similarly, for attribute generation we used Visual Genome after some preprocessing. For sentence generation module, we created our own dataset which is built using IAPR TC-12 [17] and MSCOCO descriptions.

Some most commonly used datasets for image captioning tasks are mentioned in the table. Details of the available and generated datasets for this research are discussed in the sub-sequent sub sections.

4.1 Visual Genome

Computer vision has progressed much in perceptual tasks like scene classification, object detection etc. Datasets for these tasks are also simple consisting of image and ground truth values. Apart from perceptual tasks, there are some cognitive tasks as well which require not only recognition but also require to provide reasoning like visual question answering and image captioning. For such cognitive tasks, machines are still far away from human level accuracy. One of the major reason behind this deficiency is that the models solving such problems need to identify the objects, understand the relationships among the objects, detect object attributes and similar semantics of the scene. Visual genome is a dataset aim to provide such semantics for the models to train. The dataset contains 108,077 images and some other things including region descriptions, visual question answers, object instances, attributes, relationships. All these things jointly form the complete understanding of the visual scene and are mapped to Wordnet synsets to solve the synonyms issue i.e. multiple names of a single object can be handled. The proposed methodology uses region descriptions and object attributes to train two modules based on RNN i.e. Region description generation and attribute generator.

4.1.1 Region Descriptions

Visual genome contains multiple regions and region descriptions for each image. Total number of region descriptions are 4,297,502 for all images. A generic example of what the region descriptions are and how they are related to regions with bounding boxes is shown in the Figure 14.

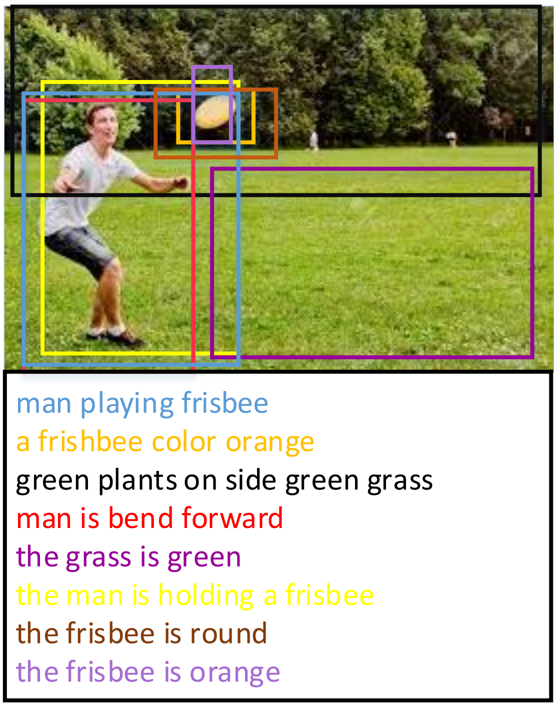


Figure 14. Region bounding boxes and region descriptions

Region descriptions are used not only to recognize the objects but also to describe the relationships between objects e.g. the sentence “man playing frisbee” is shown in the Figure 14. “playing” is a word that is describing the relationship between man and the Frisbee. These region descriptions are given by Visual genome dataset in the form of JSON file making correspondence between image id, region bounding boxes and region descriptions.

4.1.2 Object Attributes

One image can have multiple objects, among which many objects have different properties and features which are called attributes e.g. the feature “gray” in the sentence “shirt is gray” shown in the Figure 15. These properties are useful most of the times to distinguish them from rest of the objects. Apart from differentiation, these properties and features are useful to describe the details of objects. For instance, “red” in the example above is useful to provide inner details of the object i.e. hat. Major difference between region descriptions and attributes is that the region descriptions are helpful in describing the inter object details whereas the attributes are handy in describing the intra object details. The dataset "Visual Genome" provides multiple attributes against the objects in each image. Total unique objects in Visual Genome are 75,729 and total number of attribute-object instances are 1,670,182 whereas total unique attributes are 40,513. On average, one image has 16.08 attributes which will become very beneficial in describing the overall image in details. A made-up example of how attributes are related to objects in the regions is shown in the Figure 15.

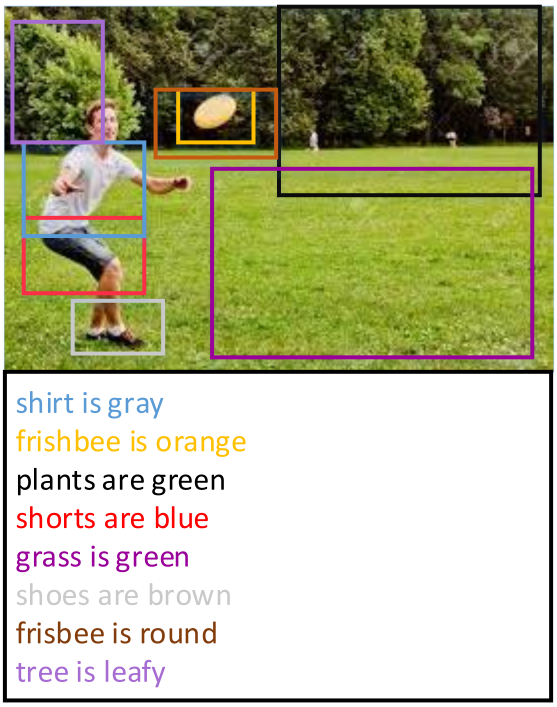


Figure 15. Region bounding boxes and attributes related to them

These attributes are given by Visual genome dataset in the form of JSON file making correspondence between image id, object name, object id, region bounding boxes and attributes list. To train the attribute generation RNN module, we joined these object names with attributes to form sentences so there will be n number of sentences if there are n attributes of any object. To elaborate, consider an object "tree" having attributes "tall" and "green". The generated sentences will be "tall tree" and "green tree". Using this type of data for training, the trained RNN will now output both the attributes and the object names.

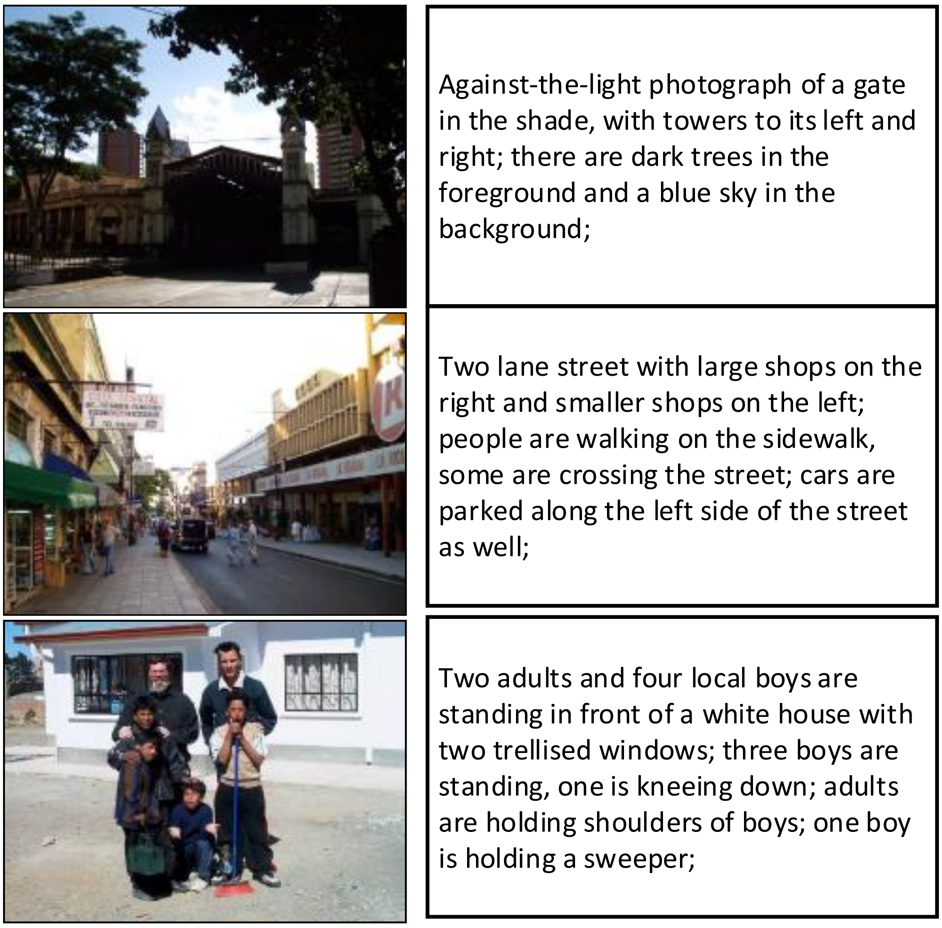
Both the generated region descriptions and object attributes are then joined by sentence generation module to form a single sentence.

4.1.3 Preprocessing

Visual genome dataset is preprocessed to form $h5$ output file which will contains image and region boxes along with ground truth i.e. region descriptions or attributes. This is done by first excluding all the ground truth sentences which have a character length greater than 40. Moreover, all the tokens (words, numbers or characters) which appear less than 5 times in whole dataset are replaced by <UNK> token which represents a special unknown token. This kind of preprocessing will help in excluding outliers and learning better weights for the RNN.

4.2 Text Data for Sentence Generation Module

One of the major challenge for the proposed research is to create a dataset which can be used for sentence generation module because currently no dataset is available which will help in training the LSTM networks to join attributes and region descriptions into a complete detailed image caption.



Against-the-light photograph of a gate in the shade, with towers to its left and right; there are dark trees in the foreground and a blue sky in the background;

Two lane street with large shops on the right and smaller shops on the left; people are walking on the sidewalk, some are crossing the street; cars are parked along the left side of the street as well;

Two adults and four local boys are standing in front of a white house with two trellised windows; three boys are standing, one is kneeing down; adults are holding shoulders of boys; one boy is holding a sweeper;

Figure 16. Examples of IAPR TC-12 dataset descriptions

The proposed research used International Association of Pattern Recognition Technical Committee 12 (IAPR TC-12) [17] benchmark which provides images with detailed descriptive lines of text. Their detailed nature gives us a benefit while training the LSTMs so that the learned model will generate detailed description. As shown in Figure 16, the descriptive lines are not so advantageous for us because if these lines are provided as ground truth, the model will learn to generate multiple lines and not a single caption which we needed. To overcome this, we not only used IAPR TC-12 dataset but also used MSCOCO dataset along with it. We will further explore pre-processing performed on these datasets in sub-sequent subsections.

4.2.1 IAPR TC-12

This benchmark is basically used for image retrieval tasks but considering its detailed descriptive nature, we used it for training sentence generation module so that our generated sentences will become detailed. IAPR TC-12 dataset contains descriptive lines against 20,000 images out of which we used 13,000 for training of sentence generator and 7000 are used for evaluation of the overall system. These descriptive lines are semi-column (;) separated and object attributes are also merged in the sentences. So we separated them from semi-column and made region descriptions. For attributes we used Parts-Of-Speech (POS) Tagging to separate all the words that are adjectives because adjectives are the properties of the objects which represent object attributes.

POS Tagging:

POS tagging is done by a POS tagger software/tool that just reads the input text and assigns a token representing parts of speech to each word in the input text. As an example, consider the original IAPR TC-12 description sentences and their POS tagged sentences in Table 1.

The tagged tokens represent parts of speech as depicted by Penn Treebank Project [47]. For example, in Table 1 (1st row, 2nd column), yellow is an adjective so its tagged as JJ where JJ represents adjectives. Some other token representations that are useful for our research are:

- JJ - Adjective
- JJR - Adjective, comparative
- JJS - Adjective, superlative
- NN - Noun, singular or mass
- NNS - Noun, plural
- NNP - Proper noun, singular
- NNPS - Proper noun, plural
- VB - Verb, base form
- VBD - Verb, past tense
- VBG - Verb, gerund or present participle
- VBN - Verb, past participle

- VBP - Verb, non-3rd person singular present
- VBZ – Verb, 3rd person singular present

Rest of the tags can be ignored as they won't be used in our research. All the nouns represent the objects in the sentences whereas adjectives represent the attributes of those objects. The POS tagging of adjectives will help to extract the ground truth attributes for sentence generator.

The extracted region descriptions and attributes are then concatenated by dot separations to form a single line of text against each image.

Table 1. IAPR TC-12 Sentences and their POS tagged version

Original Sentence	**POS Tagged Sentence**
a yellow building with white columns in the background; two palm trees in front of the house; cars are parking in front of the house; a woman and a child are walking over the square;	a_DT yellow_JJ building_NN with_IN white_JJ columns_NNS in_IN the_DT background_NN ;_: two_CD palm_NN trees_NNS in_IN front_NN of_IN the_DT house_NN ;_: cars_NNS are_VBP parking_NN in_IN front_NN of_IN the_DT house_NN ;_: a_DT woman_NN and_CC a_DT child_NN are_VBP walking_VBG over_IN the_DT square_NN ;_:
two brown rocks in the sea at a brown sandy beach with a brown cliff behind it; green bushes in the foreground, a blue sky in the background;	two_CD brown_JJ rocks_NNS in_IN the_DT sea_NN at_IN a_DT brown_JJ sandy_JJ beach_NN with_IN a_DT brown_JJ cliff_NN behind_IN it_PRP ;_: green_JJ bushes_NNS in_IN the_DT foreground_NN ,_, a_DT blue_JJ sky_NN in_IN the_DT background_NN ;_:

The above mentioned POS tagging is also used after the creation of final dataset to reduce the amount of vocabulary as discussed in sub-section 4.2.3. Different combinations of the mentioned tags i.e. adjectives, nouns and verbs can be used to reduce the amount of vocabulary and to increase the accuracy of the system. In our experiments, only tagging adjectives was helpful, the reason of which is also mentioned in sub-section 4.2.3.

4.2.2 MSCOCO

Microsoft Common Objects in Context (MSCOCO) was invented by Microsoft which contains 1.5 million object instances. The dataset provides ground truth for segmentation, object detection and for image captioning. For image captioning it provides 5 captions per image which can be used for census based evaluation tools like Consensus-based Image Description Evaluation (CIDER) [48], although 5 captions are still not enough for this tool but it's still better than other datasets having only one caption.

We passed images of MSCOCO into DenseCap [24] (a state-of-the-art region caption generation tool) to generate its region descriptions. These generated region descriptions are then compared with all 5 image captions given with the dataset to find the closest caption which will be used as target sentence. The comparison is done by using a similarity calculating tool called "DISCO" [49] explained below. The final target sentence is then POS tagged in a way similar as explained above to find the object attributes.

DISCO Similarity Measure:

DISCO (extracting DIStributionally related words using CO-occurrences) [49] is a similarity measuring java library which is used to compute semantic similarity between phrases as well as between words. It is based on statistical analysis of large corpora (text or document collections). It does the similarity matching by using a pre-processed database of vectors called word space. This database contains vector of each word mapped to the vectors of similar or highly related words. This database helps to find the closeness/similarity of the word vectors of both the input phrases. The similarity between the vectors is calculated by using COSINE similarity. COSINE similarity is a measure that finds the cosine of angle between the vectors. It is computed by calculating the dot product between the unit vectors and giving a value between 0 to 1 as output. A value 0 means that the vectors (phrases) are 100% similar and a value 1 means the vectors (phrases) are totally different from each other. Few examples of how DISCO similarity worked in our case is shown in Table 2.

Table 2. Region descriptions, MSCOCO captions and best selected caption.

Region descriptions	**Five image captions by MSCOCO**	**Best selected caption**
a brick building with a clock tower, a clock tower, a cloudy blue sky, a statue on top of a	• a clock tower has a very unique roof with a bell on top	a clock tower has a very unique roof with a bell on top

building, the tower is green, clock on the building, the roof is green, a window with a clock, a clock on the building, white clouds in blue sky	• A large tall building with a clock on top. • An old steeple with a bell tower and clock. • The bell tower has a patterned roof and a clock. • a clock at the top of a pink tower	
dog wearing a blue hat, a blue helmet, a blue helmet, dog wearing black helmet, a brown and white jacket, the head of a person, the person is sitting in the car, helmet on the head, blue hat on the head, blue helmet on the head	• A helmeted boy is riding behind a helmeted dog who is steering. • A girl wearing a motorcycle helmet and a dog wearing a camo hat • A girl and a pug dog riding in a motorcycle side car. • A person wearing a helmet sitting in a vehicle with a dog who is also wearing a helmet. • A boy and a pug wearing helmets in a car.	A girl wearing a motorcycle helmet and a dog wearing a camo hat

These region descriptions and object attributes extracted from MSCOCO dataset descriptions are then concatenated similarly by dot separations to form a single line of text against each image.

4.2.3 Dataset Splits

Both the datasets computed from IAPR TC-12 and MSCOCO are then shuffled with each other for better training and fair evaluation. The resulting dataset is then POS tagged to reduce the amount of words in the vocabulary. Only the adjectives are POS tagged and are replaced with the same "JJ" word. This is done only with adjectives so that the accuracy of the system does not decline. The POS tagged dataset is subsequently divided into three splits having 58,702 captions in the training set, 14,675 captions in the validation split and 18,344 captions in the test split. After reducing adjectives, total vocabulary for source training set is 4,829 whereas for target training set is 7,817. The vocabulary contains words, numbers and single characters appearing in the captions. Examples of source and target text for both MSCOCO and IAPR TC-12 datasets are shown in the table.

Table 3. Final versions of source and target sentences for Sentence Generation.

	Source text	**Target text**
MSCOCO	a boy in a JJ shirt . a JJ and JJ baseball field . the boy is wearing a JJ shirt . a silver metal pole . boy has JJ hair . a man holding a baseball bat . the hand of a person . the shirt is JJ . the shorts are JJ . JJ boy . JJ shirt	a JJ boy in a JJ shirt has a baseball bat .
IAPR TC-12	a woman with a helmet , trousers , a jacket and an safety jacket is sitting on a mountain bike on a gravel road to a bus in the foreground . a mountain landscape with a house and mountains with snow behind it . clouds in the background . . JJ helmet . JJ trousers . JJ jacket . JJ safety . JJ gravel . JJ bus . JJ bus . JJ mountain . JJ mountains . JJ clouds . JJ clouds	a woman with a JJ helmet , JJ trousers , a JJ jacket and an JJ safety jacket is sitting on a mountain bike on a JJ gravel road JJ to a JJ bus in the foreground ; a JJ mountain landscape with a house and JJ mountains with snow behind it ; JJ JJ clouds in the background ;

Chapter 5

Experimental Results and Analysis

In this chapter, we perform validation of your proposed methodology on the available and generated datasets explained in Chapter 4. Our approach is modular consisting of three major modules which are all trained separately on different datasets which are attributes dataset of visual genome, region descriptions of visual genome and generated textual dataset for sentence generation. After the training, the system is evaluated using a subset of images of IAPR TC-12 dataset. After some cleaning (excluding descriptions with random characters) 6802 images are used. IAPR TC-12 is used rather than any other image captioning dataset because of the nature of detailed descriptions provided in this dataset.

The evaluation is done using the same evaluation metrics that are used for image captioning tasks. These evaluation metrics compare the semantic similarity between the generated sentence of our proposed methodology and the ground truth caption provided in the dataset. The details of the metrics are provided below.

5.1 Evaluation Metrics

The validation of the proposed methodology is done using variations of BLEU, ROUGE and METEOR. We will further see how these tools work.

5.1.1 BLEU

BLEU [50] stands for Bilingual Evaluation Understudy which is basically a modified form of precision. BLEU was widely used for calculating the accuracy of automatic machine generated translations from one language into another language because of its high correlation with human generated judgements. The evaluation of the image captioning works similarly i.e. semantically comparing the machine generated caption with the ground truth captions so BLEU can be used here easily.

BLEU can be used in different variations which are BLEU-1, BLEU-2, BLEU-3 and BLEU-4. BLEU-1 is computed by using 1-grams, BLEU-2 using bigram and so on. These variations are used to make sure that the word order and occurrences are also considered otherwise the text having bunch of related words will be rewarded more.

BLEU also serves as a benchmark to check the correctness of any new evaluation metric. It is mostly used for checking the correctness of the whole corpus. Generated sentences are compared with ground truth sentences, the scores for all sentences are then averaged over the whole corpus.

BLEU is computed using geometric mean of n-gram precision. It also adds a small penalty to short sentences to give less importance. If c is the length of the candidate sentence and r is the length ground truth/reference sentence, then the brevity penalty BP will be:

$$BP = \begin{cases} 1 & if \;\; c > r \\ e^{(1-r/c)} & if \;\; c \leq r \end{cases} \tag{5.1}$$

The above mentioned brevity penalty is used in the final BLEU formula as given below:

$$BLEU = BP.\exp\left(\sum_{n=1}^{N} w_n \log p_n \right) \tag{5.2}$$

where p denotes n-gram precision which can be any between 1 to 4 to compute BLEU-2, BLEU-2, BLEU-3 and BLEU-4. In the above equation w_n denotes positive weights which will sum up to 1. So be default the weights are uniform which means $w_n = 1/N$.

BLEU formulation will output a value between 0 and 1 where 0 indicates that the generated sentence is totally different from ground truth sentence while 1 indicates that both are totally similar.

5.1.2 ROUGE

ROUGE [51] stands for Recall-Oriented Understudy for Gisting Evaluation which is an evaluation metric just like BLEU but instead of precision it uses recall to check how much the words in the generated sentences are present in the reference or ground truth sentence. Let's first consider what is the formula for precision and recall. Recall is simply division of "number of overlapping words" with "total number of words in the ground truth" i.e.

$$\mathrm{Re}\,call = \frac{no.\ of\ \ overlapping\ \ words}{total\ \ words\ \ in\ \ ground\ \ truth} \tag{5.3}$$

And precision is:

$$\mathrm{Pr}\,ecision = \frac{no.\ of\ \ overlapping\ \ words}{total\ \ words\ \ in\ \ generated\ \ sentence} \tag{5.4}$$

The precision will tell how many words are relevant in the generated sentence but if the generated sentences are long as in our case then the precision will become very small which is why precision alone is not a feasible option. Therefore, the better way is to use F-measure which is a combination of both precision and recall. In cases, when there are always expected to be lengthy sentences then the best option will be to use recall only because precision will become less important.

To explore how the ROUGE measure works, let us consider its different formulations.

ROUGE-N: ROUGE-1 will check the overlap of unigrams between generated sentences and ground truth sentences. Similarly, ROUGE-2 will check for bigrams between both sentences and so on. e.g. the ground truth is "man riding motorbike on the road" and the generated sentence is "man on motorbike on the road" now the bigrams will be:

Ground truth sentence bigrams	**Generated sentence bigrams**
man riding riding motorbike motorbike on on the the road	man on on motorbike motorbike on on the the road

Using these bigrams, the ROUGE-2 recall will be:

$$\mathrm{R}_{ROUGE2} = \frac{3}{5}$$

And ROUGE-2 precision will be:

$$P_{ROUGE2} = \frac{3}{5}$$

both are same because there are equal number of bigrams in ground truth and generated sentence but the precision becomes low and low when the sentences become lengthy.

ROUGE-1 can be used with high granularity ROUGE measures to show how much the ordering of the words are closer to the ground truth sentence.

The final formula for ROUGE-N will be:

$$ROUGE-N = \frac{\sum_{X \in (Ground\ Truth)} \sum_{gram_n \in X} Count_{matching}(gram_n)}{\sum_{X \in (Ground\ Truth)} \sum_{gram_n \in X} Count\ (gram_n)} \tag{5.5}$$

where n is the length of n-grams used and Count$_{\text{matching}}$ is the total number of n-grams that match in both the ground truth and generated sentences.

ROUGE-S: It can also be called skip-gram coocurrence because it measures if any pair of words are in the order. It matches every possible pair of words by allowing arbitrary gaps between words. Skip-grams or pair of words can be seen in the example, if the sentence is "man wearing red hat" then the skip-grams will be “man wearing”, “man red”, “man hat”, “wearing red”, “wearing hat”, “red hat”.

Recall for skip-grams will be:

$$R_{skip2} = \frac{SKIP2(A,B)}{C(x,2)} \tag{5.6}$$

where $SKIP2(A,B)$ denotes the number of 2 sized skip-grams matching between sentence A and B, x is the size of sentence A and $C(x,2)$ is the combination showing how many 2 sized skip-grams are possible of sentence A.

Similarly, the precision for skip-grams will be:

$$P_{skip2} = \frac{SKIP2(A,B)}{C(y,2)} \tag{5.7}$$

The final formula for ROUGE-S i.e. skip-bigram based F-measure will be:

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \tag{5.8}$$

where β is used to control the relative importance of precision P_{skip2} and recall R_{skip2}.

ROUGE-L:

ROUGE-L works by finding the longest common subsequence (LCS) between ground truth sentence and the generated sentence. Subsequently the length of LCS is used to decide how much the sentences are similar. Recall will be calculated as:

$$R_{LCS} = \frac{LCS(A,B)}{x} \tag{5.9}$$

where x is the size of sentence A. Similarly, the precision will be calculated as:

$$P_{LCS} = \frac{LCS(A,B)}{y} \tag{5.10}$$

where y is the length of sentence B. The final value of ROUGE-L can thus be calculated as below:

$$F_{LCS} = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS}+\beta^2 P_{LCS}} \tag{5.11}$$

ROUGE-W is also a variant of ROUGE-L which calculates weighted LCS between the sentences. ROUGE-L is the most commonly used variant of ROUGE evaluation metric. Therefore, for our system evaluation, ROUGE-L has been employed among various variants of ROUGE.

5.1.3 METEOR

Metric for Evaluation of Translation with Explicit ORdering [52] also known as METEOR is the most widely used evaluation measure for evaluation the machine generated translations. Its evaluating power is closest to the human because of the fact that METEOR has many in-built features. It handles word level correspondences in a better way. Other features include stemming, exact word matching and synonymy matching. METEOR solves the problems present in IBM's BLEU i.e. BLEU's brevity penalty is not enough to cover up the lack of including recall. Recall is an important measure to be included. METEOR first forms certain alignments between generated sentence and one or more reference sentences. The option is present for more than one reference sentences if more are available, then the best score is reported among all the scores against reference sentences. Moreover, the alignments are produced by matching exact, stem, paraphrase and synonym of words and phrases. Given a machine generated sentence and reference or ground truth sentence, METEOR is calculated by forming alignment between unigrams so that a unigram in one sentence is mapped to 0 or 1 unigram of other sentence. No mapping is generated in the same sentence, similarly no unigram is mapped to more than one unigram. Thus, Fmean is calculated by computing harmonic mean between precision and recall by giving more weight i.e. 9 to recall so the equation becomes:

$$Fmean = \frac{10PR}{R+9P} \tag{5.12}$$

To cater the length issue for longer sentences, METEOR also calculates a penalty. To calculate the penalty, first the number of chunks are counted by grouping the unigrams in the generated sentence that are mapped to the unigrams in the ground truth sentence in a fewest possible way such that chunk containing unigrams in the adjacent position in the generated sentences is mapped to the chunk containing unigrams in the adjacent position in ground truth sentence. In this way longer the n-grams that match, fewer the number of chunks. Similarly, less n-grams match, larger the number of chunks and thus increasing the penalty which is calculated as:

$$Penalty = 0.5 \times \left(\frac{no.\ of\ \ chunks}{no.\ of\ \ unigrams\ matched} \right)^3 \tag{5.13}$$

Using the above mentioned penalty and the Fmean, METEOR score can be calculated as:

$$M\ = Fmean \times (1 - Penalty) \tag{5.14}$$

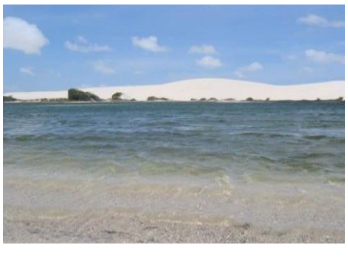
Hence, M is the final METEOR score that highly resembles with the human judgements.

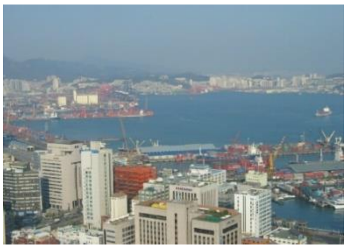
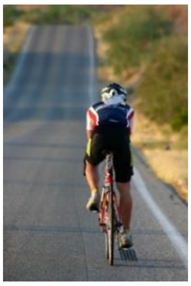
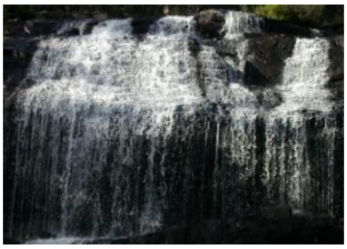
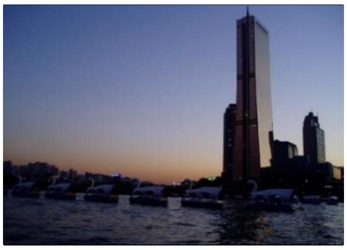
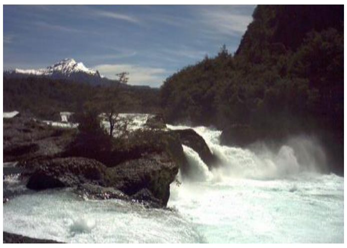
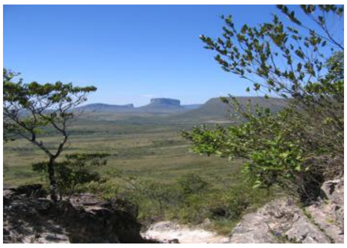
5.2 Qualitative Results

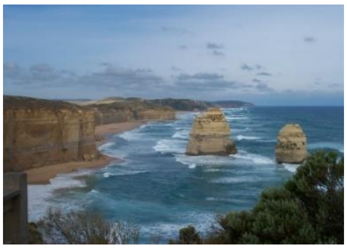
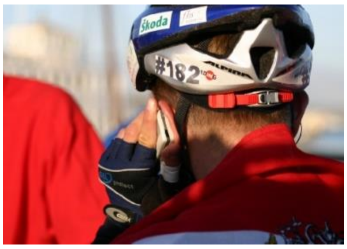
A dense evaluation has been performed on the proposed architecture to test the quality of descriptions generated by it for complex scenes as well as for simple scenes. Many of the scenarios exists, when there are few objects in the scene but those objects are hard to recognize and describe. In such cases, CaptionNet has performed really well as compared to other approaches. Qualitative comparison on some of the diverse scenes has been shown in Table 4.

3rd column of Table 4 shows the visual results of the descriptions produced by the proposed network. Specifically, we compared the results with NeuralTalk [53] (1st column) and Show, Attend and tell [54] (2nd column) so that both types of mechanisms can be compared i.e. with attention and without attention. As visually, attention mechanism only ensure the correctness of the objects present in the description and while ensuring it, it often loses many of the objects and words while generating the descriptions which makes the attention mechanism to generate less detailed descriptions as compared to the mechanisms without attention. Results of both these mechanisms and the proposed mechanism can be seen. Clearly the proposed network has successfully described the region details of the image and the overall description is dense and more descriptive in comparison to existing state-of-the-art methods. For instance, in Table 4 (1st row), one can easily note that the overall network has successfully recognized white clouds in the blue sky and the colour of water. In 3rd row, the proposed CaptionNet excels in describing the details of the biker as compared to other methodologies. In the 4th row, the proposed system has identified that it's a rock in the water while other networks failed to recognize it. Similarly, in all other examples, the network has included most of the object details in the image description clearly highlighting that the proposed CaptionNet describes the scenes in a more detailed and dense manner.

Table 4. Qualitative Results - Comparison with state-of-the-art techniques.

Images	*NeuralTalk [15]*	*Show, Attend and Tell [18]*	*CaptionNet*
	a large body of water with a boat in the background.	a view of a large body of water.	a body of water with blue water, white clouds in a blue sky in the background.

	a large body of water with a bridge in the background.	a view of a city with a city.	the buildings are in the background, a large ship in the water, a large white building and a white boat in the water.
	a woman riding a bike down a street.	a person riding a bike down a street.	a man wearing a shirt and a black helmet is riding a bicycle.
	a black and white photo of a bird in the woods.	a black and white photo of a person on a surfboard.	a large rock in a large body of water.
	a large body of water with a bridge in the background.	a view of a large body of water.	a scene to a white building, a blue sky in the city.
	a man riding a wave on top of a surfboard.	a person riding a snowboard down a snow covered slope.	trees on the shore, a landscape, a large mountain range in the distance.
	a view of a mountain in the distance.	a lone giraffe standing in the middle of a field.	a small green tree with a tall green tree in the middle of an area.

	a view of a lake with a bridge and a river.	a large body of water with a boat in the background.	a boat with blue white water, a large body of water and a large rock in the water.
	a man wearing a hat and a hat on a motorcycle.	a man wearing a helmet and a helmet on a motorcycle.	a man on a motorcycle with a red shirt and a pair of shoes on the street.

5.3 Quantitative Results

The quantitative results using the above-mentioned evaluation metrics using 6802 images are shown in the Table 5. As depicted, the proposed network has outperformed the other two state-of-the-art techniques in all the performance measures showing that the network is capable of describing complex scenes in a more detailed way. The difference in depth and quality of the descriptions is due to the fact that complex scenes contain multiple objects with attributes where the performance of normal encoder-decoder framework is limited while the proposed architecture detects and describes those multiple objects individually along with their attributes to form better descriptions for complex scenes.

Table 5. Comparison of quantitative with state-of-the-art techniques.

Evaluation Metric	**Network Models**		
	NeuralTalk [53]	**Show, Attend and Tell [54]**	**CaptionNet**
BLEU-1	0.091	0.080	**0.128**
BLEU-2	0.047	0.041	**0.064**
BLEU-3	0.025	0.022	**0.031**
BLEU-4	0.013	0.011	**0.016**
METEOR	0.6241	0.60	**0.6865**
ROUGE-L	0.215	0.207	**0.216**

5.4 Evaluation Dataset

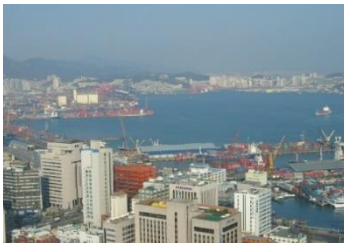
Evaluation of the proposed methodology can only be done on a dataset that contains lengthy descriptions against images such as IAPR TC-12 and not on MSCOCO. To elaborate, consider the following example descriptions shown in Table 6.

Table 6. Sample descriptions of IAPR TC-12 and MSCOCO dataset along with CaptionNet generated description

IAPR TC-12	MSCOCO	CaptionNet generated output
a yellow building with white columns in the background; two palm trees in front of the house; cars are parking in front of the house; a woman and a child are walking over the square;	A group of elderly travelers around a bench near the ocean	trees on the shore, a landscape, a large mountain range in the distance.

Consider IAPR TC-12 dataset, it contains semi column separated text lines which together describe all the contents of the image while MSCOCO contains simple one-line caption that do not describe full image contents but it only tells overall what is happening in a short sentence.

Table 7. Detailed nature description generated by CaptionNet

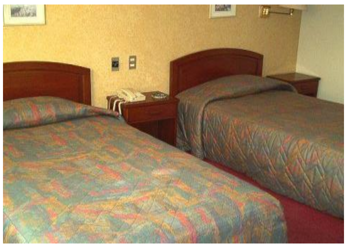
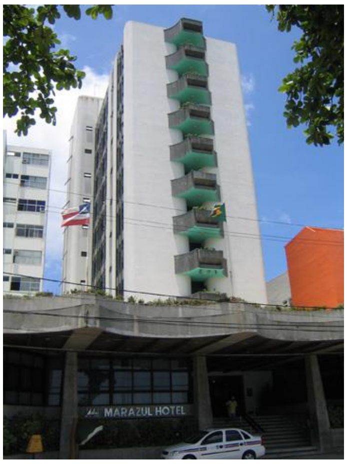
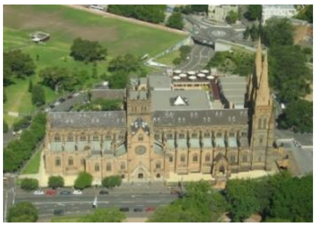
Image	CaptionNet generated output
	the buildings are in the background, a large ship in the water, a large white building and a white boat in the water.

The proposed CaptionNet aims to generate a caption which describes multiple objects in one line. Because in the example (Table 7) short sentence cannot describe what is happening in the scene completely. So to evaluate it MSCOCO is not suitable as its captions do not contain much of the information related to the scene. Nature of CaptionNet's captions is a bit different from MSCOCO Therefore, if evaluating dataset caption does not contain much words then how a long sentence with many words can be compared with it? Moreover, all the evaluating metrics e.g. BLEU works by comparing words (unigrams, bigrams etc.,) which also depicts the importance of having large number of words in evaluating dataset. Considering the training phase, if we only train our system using IAPR TC-12 dataset, the system will generate multiple semi column separated lines and not one line caption. That semi column separated paragraph is not useful in many cases. Another reason is that the IAPR TC-12 dataset is small. So to include the single line generation nature in the model and to make dataset large, MSCOCO is also included along with IAPR TC-12 dataset for training.

5.5 Discussion of Results

As shown in the above qualitative and quantitative results, the proposed architecture performs better to provide more details in the final description specially in the cases when there are many objects or the scene is complex. It also seems to be performing well in case of partial occlusions as shown in Table 4 (4th row) where the "large rock" is not clearly visible failing other architectures to describe it, but the proposed CaptionNet has not only described it but also mentioned that it is in the body of water. The reason behind this is that instead of describing image as a whole, it breaks the image into objects and generates their individual descriptions. These descriptions are passed to two LSTMs networks in the sentence generation module that incorporates the scene context and fuse these region descriptions semantically into a complete image caption. To remove false positives in the region extraction, only the objects with high objectness scores are kept while others are discarded, these retained objects are then passed to sentence generation module for single line caption creation. Not only this, the sentence generation module also uses attention mechanism to join the region descriptions and features of the objects. The attention mechanism makes it feasible to generate large sentences without inserting any unneeded word that is not actually present in the image. Without attention mechanism, large sentences (as in our case) cannot be generated with much accuracy. For complex scenarios (having multiple objects e.g. 2nd row in Table 4) requiring large sentence caption passing only one single hidden state to the decoder is not enough because single state represents a very little information. Detecting maximum possible objects and using attention mechanism while fusing them into a sentence results into a fine detailed image caption. The result of all this careful engineering is that the final sentence includes maximum possible objects present in the input image while reducing much of the possibility of inserting any false detection in the final output. Moreover, since the feature extracting CNN i.e. VGG-16 is initialized with weights pre-trained on ImageNet [11] dataset containing over 1000 object categories making it to extract features of diverse categories which are helpful for region extraction module and thus useful for better image description generation.

Table 8. Two special cases: 1) When the architecture described parts of the objects too, 2) When the architecture generated redundant objects

Images	*NeuralTalk [15]*	*Show, Attend and Tell [18]*	*CaptionNet*
	a bedroom with a bed and a table	a bedroom with a bed and a bed.	a bed with a comforter and a wooden headboard
	a large building with a clock on the top of it	a large building with a clock on it.	a tall building in the background, a tall building, a tall buildings, a tall buildings, a tall buildings, a tall buildings, a tall building in the background
	a large building with a clock on the top of it.	a group of people standing around a building.	a large building with large building, white and buildings, a large building and a large building

The proposed architecture is also able to detect and describe parts of the objects e.g. "headboard" of bed shown Table 8 (1st row and 3rd column). This incredible capability is introduced in the architecture because of the fact that it extracts objects to include in the description e.g. headboard. Similarly, it can detect and describe attributes highlighting the inner object details e.g. "wooden" in the same example (1st row and 3rd column), relationships between objects e.g. "with" in the sentence "a bed with a comforter" where "comforter" and "bed" are two different objects etc. Although both of these objects may or may not be in the same bounding box, the proposed system succeeded in finding the relationship between them. Objects, relationships and attributes can be seen in all of the other examples in Table 4. This type of in-depth detection helps to explain minor details of the scene.

Attribute training is done using dataset that contains generic images and natural scenes i.e. visual genome [26]. The network can be made category specific by training on different types of datasets e.g. cars dataset [55]. If the system is trained on cars dataset, then it will start generating descriptions which will include car models and years. Similarly, the network can be trained on garments dataset [56] for more detailed attribute detection of the clothes or parts of the body. Ability of the system to generate more descriptive statements in terms of a specialized area can be increased by using these kinds of category specific datasets. This can be done in a similar way as we did with region extraction module which was pre-trained on ImageNet dataset and further trained as encoder-decoder framework on visual genome. Fine-tuning can be done by freezing the initial layers, in this way we will retain the weights of initial layers because they contain the learning of basic shapes like curves and edges which will also be present in our category specific dataset. Fine-tuning can simply be done by further training the networks for more epochs.

Another thing to discuss is that short vocabulary helps in training the sentence generation module easily and with consuming less time. To benefit from this, we used POS tagging to reduce the vocabulary and to make sentence generation more easy and reliable. As discussed earlier, attributes are adjectives and objects are nouns. We only POS tagged the adjectives for the training of sentence generation module. There can be few other options like POS tagging nouns or POS tagging nouns and adjectives both. As we experimented, nouns are also a good option for POS tagging to make vocabulary size even more shorter but that will make it difficult to replace the nouns with original objects from source sentence after the sentence generation output and thus making the description a little un-reliable. So the safest and more dependable way to reduce the amount of vocabulary with POS tagging is to tag only the adjectives which will be replaced back at the end by matching which adjective is related to which object in the source sentence.

The presented approach works very well in describing scenes with much details, a situation still needed to be mentioned is when the architecture repeats certain objects in the description e.g. "tall building" in Table 8 (2nd row and 3rd column) and "large building" in Table 8 (3rd row and 3rd column). The main reason behind this issue is that the region extraction block extracts regions and retain only those regions which have high probability of object present in them, one object can be detected with different bounding boxes so there is possibility that the same object is selected with different bounding boxes if all the bounding boxes have high objectness scores. Huge number of regions can be detected by the sliding window of the RPN so all of them cannot be included in the scene descriptions. Moreover, the greedy approach to select regions with high objectness scores works very well for full image captions. Therefore, we cannot completely remove the repeated objects problem because of the mentioned reasons. Another big reason is that there can be multiple instances of the same object in the scene in real e.g. there can be many large buildings in the scene which are very difficult to differentiate, so the repeated objects problem cannot be completely eliminated.

Another situation worth discussing here when the architecture lacks to generate excellent descriptions is shown in Table 4 (last row, 3rd column) where CaptionNet generated the description of the scene as: "a man on a motorcycle with a red shirt and a pair of shoes on the street". There are attention mechanisms employed in different modules of the system locally but overall system still does not have any global mechanism to match the generated words of the sentence generation module with the objects in the image which is why some kind of noise is expected to be introduced in the final description e.g. the words in "a pair of shoes" do not represent any of the objects in the scene but these words are still inherited from other objects of the scene while generating the caption. Sentence generation module has the in-built power to inherit some other aspects and objects apart from the actual objects of the scene but the power is not controlled by a global attention mechanism.

Chapter 6

Conclusion and Future Work

Real life images can be described in more details when individual objects and aspects of the scene are described prior to generating full image caption. In this research, a deep learning based modular architecture CaptionNet is proposed which exploit individual object information present in the input image to form dense image captions. CaptionNet extracts inter object details in the form of region descriptions and intra object information in the form of object attributes in order to present a dense level image caption. This modular architecture consists of three modules: First, the **region extraction module** detects the regions based on objects along with their confidence/objectness scores using an adapted RPN network. Second, the **language generation module** takes the extracted regions to produce region descriptions and attributes by using the visual to text mapping it learnt before. Third, the **sentence generation module** merges these two types of text descriptions using two LSTMs based encode-decoder framework, thus producing syntactically and grammatically correct single line but detailed sentence as image caption. The system is evaluated using IAPR TC-12 dataset which contains complex scenes together with their lengthy detailed descriptions. The qualitative and quantitative results obtained from experiments indicate that the proposed CaptionNet has shown superior performance over existing state-of-the-art image captioning architectures.

Although the system has shown high accuracy, the architecture can be expanded to improve the results and can be extended for specialized domains. Some of the future research directions are as follows:

- **Non-Rectangular Regions:** Currently the system is extracting rectangular region proposals which can be altered so that system can detect morphed regions. This can be done by using Spatial transformer network [31] or RotNet [57]. These kind of networks can be employed at the first layer of CNN or while using RPN to achieve global or local results. Moreover, such architectures help to extract rotation invariant features which means any rotated object can also be described with more accuracy using these networks.
- **Specialized Dataset:** Specific to some specialized domain datasets e.g. the “cars” dataset [55] can be used to fine-tune the overall system so that the architecture can be used to

generate captions describing aspects of that domain such as parts of the cars. This will increase the descriptive capability of the system.

- **Relationships Module:** A separate module to handle and describe inter object relationships can be added just like our region description and attributes generation modules. Apparently, region descriptions include both object relationships and attributes but experiments have shown that the accuracy and describing power of the system increased by detecting and including attributes separately. This shows that the overall system can further be improved by incorporating the relationships module separately.
- **Improved Dataset:** The sentence generation module is trained on automatically generated dataset. There is no other dataset exists as per our knowledge, that can be used to join chunks of sentence or words to form a complete detailed sentence. The accuracy of the module can be increased tremendously by using more carefully generated dataset by some human experts.
- **Translated Descriptions:** Sentence generation module is derived from seq2seq models which are mostly used for machine translation task. This capability of sentence generation module can be exploited such that the image captions are generated after translation in some other language. This can be done by using cross language dataset in which source region and attribute descriptions are in English whereas the target captions are in another language.

Bibliography

[1] C. Ye, Y. Yang, R. Mao, C. Fermüller, and Y. Aloimonos, "What can i do around here? deep functional scene understanding for cognitive robots," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2017, pp. 4604-4611: IEEE.

[2] M. Maynord, S. Bhattacharya, and D. W. Aha, "Image surveillance assistant," in *Applications of Computer Vision Workshops (WACVW), 2016 IEEE Winter*, 2016, pp. 1-7: IEEE.

[3] S. S. Trundle, R. J. McCarthy, J.-P. Martin, A. J. Slavin, and D. J. Hutz, "Image surveillance and reporting technology," ed: Google Patents, 2015.

[4] J. T. Nganji, M. Brayshaw, and B. Tompsett, "Describing and assessing image descriptions for visually impaired web users with IDAT," in *Proceedings of the Third International Conference on Intelligent Human Computer Interaction (IHCI 2011), Prague, Czech Republic, August, 2011*, 2013, pp. 27-37: Springer.

[5] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, 2005, pp. 253-262: ACM.

[6] C. C. Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," 2017.

[7] A. Farhadi *et al.*, "Every picture tells a story: Generating sentences from images," in *European conference on computer vision*, 2010, pp. 15-29: Springer.

[8] G. Kulkarni *et al.*, "Baby talk: Understanding and generating image descriptions," in *Proceedings of the 24th CVPR*, 2011: Citeseer.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881-2890.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 12, pp. 2481-2495, 2017.

[11] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision,* vol. 115, no. 3, pp. 211-252, 2015.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[13] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait, "High-performance OCR for printed English and Fraktur using LSTM networks," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 2013, pp. 683-687: IEEE.

[14] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned lstm-based natural language generation for spoken dialogue systems," *arXiv preprint arXiv:1508.01745,* 2015.

[15] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128-3137.

[16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015, pp. 3156-3164: IEEE.

[17] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The iapr tc-12 benchmark: A new evaluation resource for visual information systems," in *International workshop ontoImage*, 2006, vol. 5, p. 10.

[18] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048-2057.

[19] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov, "Review networks for caption generation," in *Advances in Neural Information Processing Systems*, 2016, pp. 2361-2369.

[20] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, vol. 6.

[21] C. Liu, F. Sun, C. Wang, F. Wang, and A. Yuille, "MAT: A multimodal attentive translator for image captioning," *arXiv preprint arXiv:1702.05658,* 2017.

[22] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," *arXiv preprint arXiv:1704.03899,* 2017.

[23] L. Zhang *et al.*, "Actor-Critic Sequence Training for Image Captioning," *arXiv preprint arXiv:1706.09601,* 2017.

[24] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4565-4574.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[26] R. Krishna *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision,* vol. 123, no. 1, pp. 32-73, 2017.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[29] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications,* vol. 13, no. 4, pp. 18-28, 1998.

[30] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083,* 2015.

[31] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017-2025.

[32] !!! INVALID CITATION !!! [13, 14, 16, 18].

[33] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks,* vol. 5, no. 2, pp. 157-166, 1994.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104-3112.

[36] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534-4542.

[37] M. Huck and A. Birch, "The Edinburgh machine translation systems for IWSLT 2015," *Proc. of IWSLT, Da Nang, Vietnam,* 2015.

[38] L. Jehl, P. Simianer, J. Hitschler, and S. Riezler, "The Heidelberg university English-German translation system for IWSLT 2015," *Proc. of IWSLT, Da Nang, Vietnam,* 2015.

[39] M. Mediani, J. Niehues, E. Cho, T.-L. Ha, and A. Waibel, "The KIT translation systems for IWSLT 2015," in *Proc. of IWSLT, Da Nang, Vietnam.*, 2015.

[40] D. E. Rumelhart, J. L. McClelland, and P. R. Group, *Parallel distributed processing*. MIT press Cambridge, MA, 1987.

[41] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555,* 2014.

[42] M.-T. Luong, E. Brevdo, and R. Zhao, "Neural Machine Translation (seq2seq) Tutorial," *https://github.com/tensorflow/nmt*, 2017.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[44] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, 2014, pp. 740-755: Springer.

[45] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using Amazon's Mechanical Turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010, pp. 139-147: Association for Computational Linguistics.

[46] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics,* vol. 2, pp. 67-78, 2014.

[47] B. Santorini, "Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision)," *Technical Reports (CIS),* p. 570, 1990.

[48] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566-4575.

[49] P. Kolb, "Disco: A multilingual database of distributionally similar words," *Proceedings of KONVENS-2008, Berlin,* vol. 156, 2008.

[50] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311-318: Association for Computational Linguistics.

[51] C.-Y. Lin, "Recall oriented understudy of gisting evaluation," ed, 2005.

[52] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65-72.

[53] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3128-3137.

[54] X. Kelvin *et al.*, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," 2015/06/01, 2015. Available: http://proceedings.mlr.press/v37/xuc15.html

[55] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, 2013, pp. 554-561: IEEE.

[56] J. Shen *et al.*, "Unified structured learning for simultaneous human pose estimation and garment attribute classification," *IEEE Transactions on Image Processing,* vol. 23, no. 11, pp. 4786-4798, 2014.

[57] D. Saez, "Correcting Image Orientation Using Convolutional Neural Networks," ed, 2017.