Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science

Dept. of Computer Engineering and Microelectronics

Remote Sensing Image Analysis Group

Deep Captioning for Analysis of Remote Sensing Images in Large Archives

Master of Science in Computer Science

May, 2019

Sonali Nayak

Matriculation Number: 386995

Supervisor: Prof. Dr. Begüm Demir

Advisor: Gencer Sümbül

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, Date 16.05.2019

. . . .

Sonali Nayak

Acknowledgement

There are many people whom I would like to thank for being with me though out this journey.

First to my advisor Professor Begüm Demir, for giving me this opportunity and guiding me from beginning till the end, for which I'm so grateful. Over the last 8 months, we have exchanged hundreds of emails, several meetings and discussions. It was a great pleasure to work with such knowledgeable and experienced person, who have an eye to see things beyond I could have thought.

Next I would like to thank my supervisor Gencer Sümbül, who has helped me understand the problem and teaching me how to work and think like a researcher. It wouldn't have been possible to accomplish anything without it.

Finally I would also like to thank my parents and my sister, whose constant support and encouragement even from far away helped me achieve my goal.

Abstract

In this thesis we propose 3 frameworks for remote sensing image captioning. These frameworks are based on encoder-decoder architecture, where a convolutional neural network (CNN) is used as an image encoder, and a long short-term memory (LSTM) is used as a decoder. We also introduce an additional pretrained summarization network to our captioning framework in order to alleviate some of the limitations of the datasets that we used. The summarization network helps capture some additional information for language generation and also prevents overfitting and generalizes the model.

In the first proposed method, we combine the outputs of the captioning and the summarization model. In the second approach, we try to minimize the distance between summarization and captioning outputs using KL divergence. In the third method along with KL divergence we also use a soft attention mechanism to give more importance to parts of the image which are more relevant to the language encoding.

We use automatic and manual evaluations in order to demonstrate the quality of our models. This allows generating captions for the remote sensing images which will be helpful for search and retrieval in big data archives.

Keywords: Remote Sensing Image Captioning, Deep learning, Neural Networks, convolutional neural networks, long short-term memory, text summarization

Zusammenfassung

In dieser Arbeit werden 3 Frameworks für die Beschriftung von Fernerkennungsbildern untersucht. Diese Frameworks basieren auf einer Encoder-Decoder-Architektur, bei der ein Convolutional Neural Network (CNN) als Bildcodierer und ein Long Short Term Memory (LSTM) als Decodierer verwendet werden. Wir führen außerdem ein zusätzliches vor-trainiertes Zusammenfassungs-Netzwerk in unser Beschriftungs-Framework ein, um einige der Einschränkungen der von uns verwendeten Datensätze zu beseitigen. Das Zusammenfassungsnetzwerk hilft dabei, einige zusätzliche Informationen für die Sprachgenerierung zu erfassen, eine Überanpassung zu verhindern und das Modell zu verallgemeinern.

In der ersten vorgeschlagenen Methode kombinieren wir die Ausgaben des Beschriftungs- und des Zusammenfassungsmodells. Im zweiten Ansatz versuchen wir, den Abstand zwischen Verdichtungs- und Beschriftungsausgaben mithilfe der KL-Divergenz zu minimieren. Bei der dritten Methode verwenden wir neben der KL-Divergenz auch einen Soft-Attention-Mechanismus, um Teilen des Bildes mehr Bedeutung zu verleihen, die für die Sprachcodierung relevanter sind.

Wir verwenden automatische und manuelle Auswertungen, um die Qualität unserer Modelle zu demonstrieren. Auf diese Weise können Beschriftungen für die Fernerkennungsbilder generiert werden, die für die Suche und den Abruf in Big-Data-Archiven hilfreich sind.

Schlüsselwörter: Bildunterschrift, Tiefenlernen, Neuronale Netze, Faltungsneuronale Netze, langes Kurzzeitgedächtnis, Textzusammenfassung

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
BOW	Bag Of Words
BLEU	Bilingual evaluation understudy
CGAN	Conditional Generative Adversarial Network
CIDEr	Consensus-based Image Description Evaluation
CNN	Convolutional Neural Network
CSIC	Combining Summarized Captions to Image Captions
CSMLF	Collective Semantic Metric Learning Framework
CV	Computer Vision
FV	Fisher Vector
HOG	Histogram of Oriented Gradients
GRU	Gated Recurrent Unit
IC	Image Captioning
LSTM	Long-Short Term Memory networks
m-RNN	Multimodal Recurrent Neural Networks
NIC	Neural Image Captioning
NLP	Natural Language Processing
RNN	Recurrent Neural Network
RS	Remote Sensing
RSICD	Remote Sensing Image Captioning Dataset
SCAttKL	Summarized Captioning with Attention and KL Divergence
SCKL	Summarized Captioning with KL Divergence
SIFT	Scale-Invariant Feature Transform
TF-IDF	Term Frequency Inverse Document Frequency
USGS	United States Geological Survey
VLAD	Vector of Locally Aggregated Descriptors
i.i.d	Independent and Identically Distributed

1 Introduction

One of the fascinating things in the field of Artificial Intelligence is the ability of a machine to understand its natural surrounding and be able to communicate or present its details in an understandable human language. This particular task is well known as the task of image captioning. The scientific community has been contributing to it for decades. The advances of satellite missions help acquiring high-resolution satellite images which provides rich structural or spatial information of the ground objects. Due to the availability of these high-resolution satellite images, the task of image captioning is now applicable for remote sensing images. With time, technology, improved spatial and spectral resolution of RS images and the latest hardware resources, it can be improved a lot more. Remote sensing image captioning could be a useful tool for not only military applications, but also for RS image search and retrieval and many others such as geological survey, monitoring of natural calamities and so on [1].

When we consider the scope of natural image captioning, there has been quite some work to refer to. Although, when it comes to remote sensing image captioning, it is significantly less. This was also due to the fact that, there wasn't a proper dataset available which we could use as a ground truth in learning captioning rules. But now we have at least three captioning datasets for remote sensing images from [31] and [23], which can help implement different captioning algorithms.

Remote sensing image analysis has a lot of significance when it comes to geographical and environmental analysis. It has been considered as the fundamental problem in AI [23]. It is no longer limited just to military applications but also towards business and climatic applications. Today there are huge archives of high resolution satellite images, that are acquired from Earth Observation satellites. To describe the contents of these images, it is impossible to do it manually. A lot of information can be extracted from those images which could be valuable for recognizing various geographical distributions such as vegetation and ecosystem, urbanization and also recognizing climate and weather conditions that leads to desertification, deforestation, forest fires, flooding and so on. Having a system that can automatically describe these information can help ease the process of search and retrieval from these big data archives. For all these tasks, remote sensing image captioning can be a very useful tool.

In recent years, automatic image captioning problems were extensively addressed using deep learning techniques [23] which we also want to use in order to model our remote sensing captioning framework. There is already a considerable amount of work available for natural image captioning, and very limited research for remote sensing images which is why we want to contribute towards it. That being said, in this thesis we use deep-learning methods to accomplish RS image captioning. It is performed in 2 sub-tasks. First, the feature extraction phase which, with the help of *Convolutional Neural Network* is used to extract the features and second to convert these feature representations into natural language which is handled by *Recurrent Neural Network* (RNN) [35] or *Long-Short Term Memory Networks* (LSTM) [13]. The overall framework is considered to be an encoder-decoder framework where the encoder(CNN) accepts an image and generates encoded features, and the decoder takes the features generated from the encoder and uses them to generate a caption for a given image. A model trained over these sets of image caption pairs, can then produce a caption/sentence automatically given an image.

In the scope of this thesis, we try to develop novel methodologies to contribute to the task of image captioning of remote sensing images. Especially, our focus is to develop a neural network framework using deep learning techniques by providing captioning rules. In the end the framework will automatically produce a caption given just an image.

In the next chapter, we discuss the existing state of art work for image captioning and the related work done till now on Remote sensing image captioning. We also briefly discuss supervised learning, neural networks including Convolutional neural networks and Recurrent Neural Networks.

In the 3rd chapter, we try to implement the existing techniques available for RS image captioning, so that we can compare the results of these methods with respect to our proposed methods.

In the 4th chapter, we discuss the details of our proposed novel framework. There we discuss that how, given a set image caption pair, we can train our network to give us the caption of an image provided it is trained on a finite data set containing image caption pairs. The proposed model will process the image using one neural network(CNN) and will also process the corresponding sentences of the image using another neural network(LSTM) which then can be combined in a common embedding space to detect the similarities. The work done in this regard is based on the works of [37], [23].

In the 5th chapter we discuss the data sets used for training and modeling and the Experiments performed during the entire process of this project. The challenges faced while dealing with problems in the data sets and how we solved it.

In chapter 6, we provide the result of all the experiments that we performed and discuss the outcome that we achieved. Here we also try to evaluate our result and discuss the produced outcome and compare them with the current state of art results. And in the last chapter we conclude by giving an overview of our thesis and discussing further future prospects.

2 Foundation and Related Work

In this chapter we discuss the background involving Deep Learning and Machine Learning and how they have been used for Natural image captioning and remote sensing image captioning. Deep learning techniques have been widely used for solving complex tasks by employing layers of algorithms on huge sets of data [10], [9].

2.1 Supervised Learning

Supervised learning is considered as a task in machine learning that requires a machine to learn a function f that can map an input X to an output Y as $f : X \to Y$ with help of some example pairs $(x, y) \in (X, Y)$. For example in case of computer vision X can be an input space consisting of images of dogs, and Y can be an interval between $[0, 1]$ which can tell us the probability of a dog being in the image. So, if there is dataset of images labeled by humans whether the image contains a dog or not, it is possible to specify a function that can map the image to a label of whether it is a dog or not.

To determine this function first we need a training dataset containing n number of examples $\{(x_1, y_1), ...(x_n, y_n)\}$, which is are independent and identically distributed (i.i.d) from a distribution D. Then we search for the most consistent mapping function f for the given dataset by searching over a class of functions F and then choose a loss function that measures the difference between the predicted label and the true label. Essentially, we try to find a function $f \in F$ that satisfies:

$$f = argminE_{(x,y)\ D}L_{(f(x),y)}. \tag{2.1}$$

After finding our objective function f we can make use of the learned function that can map the elements of X to Y. Although finding the optimal function is not possible without making some assumptions since we don't know all the elements of D. But, if the data is assumed to be i.i.d, the loss can be averaged over the available training data as follows,

$$f = argmin\frac{1}{n}\sum_{i=0}^{n} L_{(f(x_i),y_i)}. \tag{2.2}$$

2.2 Neural Networks

The idea of neural networks comes from the biological inspiration of a brain's neurons. Its structure is constructed by matrix multiplication and element-wise non-linearities. An example of a 3-layered neural can be seen in Figure 2.1

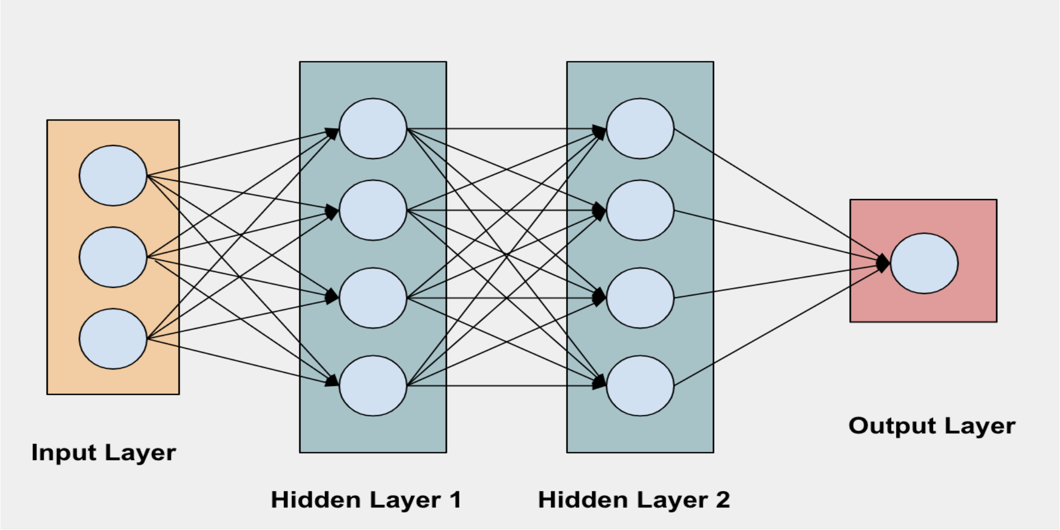


Figure 2.1: Example of a simple 3-layered neural network

All the neurons within a layer are not connected to each other but the neurons of one layer are connected to the neurons of the previous layer which enables us to evaluate the activations of a single layer with matrix multiplication. For example, a 2-layer network can be constructed as $f(x) = W_2\sigma(W_1x)$, where W_1 and W_2 are the matrices and σ is the element-wise non-linearity (e.g tanh). The last layer of the network usually doesn't contain any non-linearity and a network consisting a single layer is just a linear transformation.

2.3 Convolutional Neural Network

A Convolutional Neural Network [29] is designed to deal with data like images, videos, text sequences and so on, usually the data which has some spatial topology. So the input data in this case is treated as n-dimensional array (or a tensor). For example a 256 X 256 colored image is represented as a 265 X 256 X 3 tensor where 3 is for the number of color channels (red, blue and green).

The core building block of a CNN is a convolutional layer, which accepts an input in form of a tensor and generates output in form of a tensor as well, by convolving the input with some filters. Then we slide the filter across the input image for it convolve and compute a dot product of each of the image pixel and the filter resulting in an activation map. These activation maps are then stacked to produce a final output tensor.

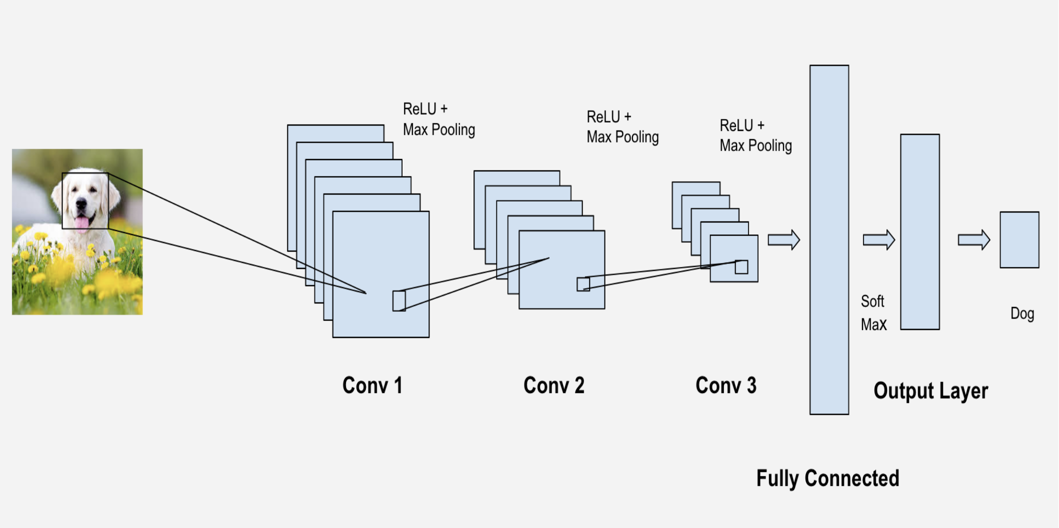


Figure 2.2: Example of a simple CNN architecture

2.4 Recurrent Neural Networks

Recurrent Neural network is practically applied for sequence generation tasks such as, language generation, language translation, speech recognition, predictions on time series data, stock predictions and so on. For example, in case of language generation, sentences are usually modeled as a sequence of words, where each word of the sequence is encoded as a one-hot vector where the vector consists of zeros at all indexes except for a single 1 at the index of the word in a fixed vocabulary. Essentially, an RNN processes a sequence of word vectors $x_1, ..., x_N$ recursively as $h_t = f_\theta(h_{t-1}, x_t)$ where θ is the parameter of function f and h_t is a hidden vector. The hidden vector h_t runs over all the input vectors x until that time step and then the function f_θ updates the vector based on the next vector.

2.4.1 Long Short-Term Memory

LSTM [13] is a variation of RNN which solves the problem of vanishing gradient problem. LSTM allows a more computationally complex interaction of the inputs x_t and hidden state h_{t-1} at each time step, which helps backpropagate errors more effectively. LSTM also has a memory vector c_t which acts as a gated cell. This allows LSTM to choose information that it can read, write or forget at each time step. In equation 2.3 , i_t, o_t and f_t are the input, output and forget gates of the LSTM and c_t and h_t are the memory and hidden state of the LSTM.

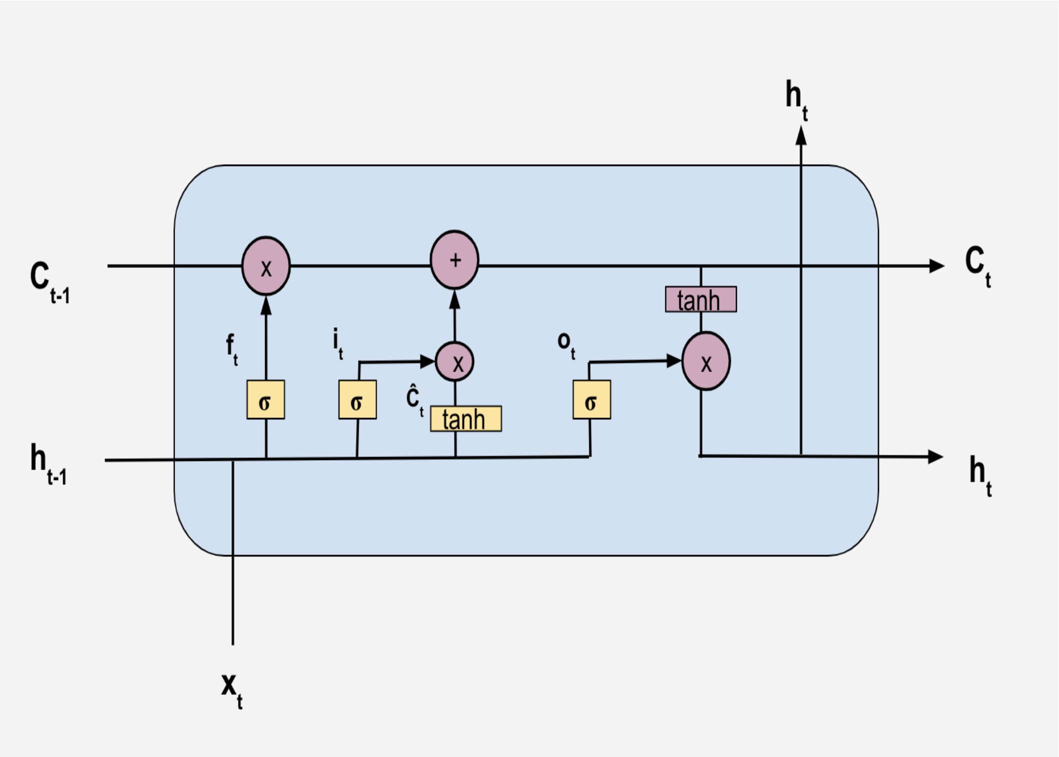


Figure 2.3: Internal architecture of LSTM

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \tag{2.3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.4}$$

$$h_t = o_t \odot tanh(c_t) \tag{2.5}$$

2.4.2 Gated Recurrent Unit

GRU [5] was also designed to solve the vanishing gradient problem in RNNs. It uses update and reset gate to decide which information to pass to the output. The update gate vector z_t at time t is given by $\sigma(W^z x_t + U^z h_{t-1})$ and the reset gate r_t at time t is also calculated similarly as $\sigma(W^r x_t + U^r h_{t-1})$ but with different weights. They have a memory content to store the relevant information which is calculated as $h'_t = tanh(W x_t + U(r_t \odot h_{t-1}))$. Finally the current memory at each step is calculated as $h_t = z_t \odot ht - 1 + (1 - z_t) \odot h'_t$. Its internal structure can be seen in Figure 2.3.

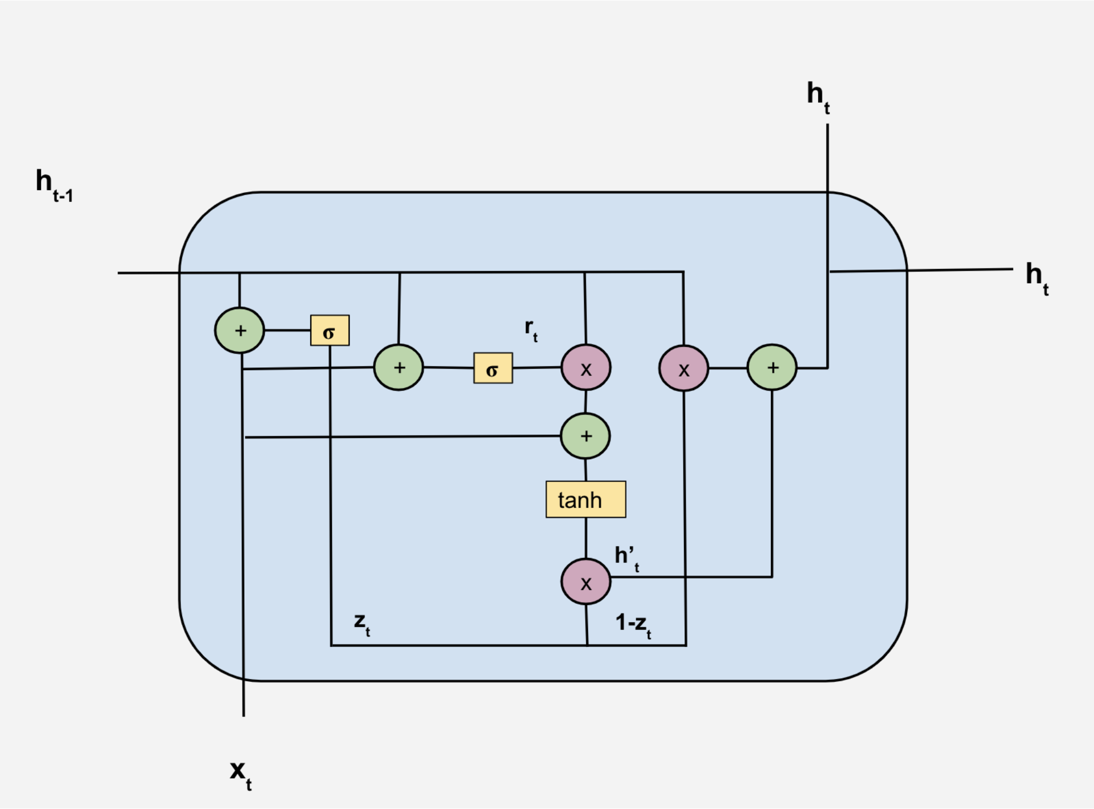


Figure 2.4: Internal architecture of GRU

2.5 Natural Image Captioning

Image Captioning task is one of the most well known tasks in the field of AI. It takes into account the 2 important application areas of AI, that is, computer vision and natural language processing. It requires understanding of the image in order to generate features and objects as well as understanding the syntax and semantics of a language to be able to generate informative description of an image.

Understanding the image for the purpose of feature extraction can be achieved using several techniques. These techniques can be categorized as Traditional Machine Learning based or Deep Machine Learning based techniques [14]. Traditional techniques are the ones in which features were manually designed or handcrafted [27], such as Scale-Invariant Feature Transform (SHIFT) [22], Histogram of Oriented Gradients (HOG) [7] and other similar techniques. Whereas the Deep Learning based techniques for image feature extraction are based on complex networks such as CNNs, which can learn the features when the image is passed through its deep layers. Usually, these deep layers, for image feature extraction are generic and can handle huge and diverse data sets to train on.

Moving forward to language understanding, this part is required in the process of image captioning, in order to decode the image features in terms of natural language sentences. These language generation parts use models like RNN (Recurrent Neural Network) or LSTM (Long-Short Term Memory networks) which accepts the input features, for example from the CNN, along with the corresponding caption of the image. The model as whole, with the combination of a CNN and LSTM can then be trained in an end-to-end manner. This trained model afterwards, can be used to generate sentences or captions, given an image.

There has been significant amount of work on natural image captioning. One of the most attractive methods for image captioning now are these encoder-decoder architecture based networks. On of the most popular works based on this framework was *Neural Image Captioning* [37] by O. Vinyals et al., in which they directly maximize the likelihood of a target sentence given an image. Junhua Mao et al. proposed *Explaining Images with Multimodal Recurrent Neural Networks* [25] which models the probability distribution of predicting the next word provided the previous words and the image. Then there was *Multimodal neural language models* by Kiros et al. [19], which uses a feed forward neural network to predict the next word provided with the image and previous words. Xu et al. proposed *Show Attend and Tell* [40] which uses attention mechanism to say “where” or “what” to look in an image, by extending the works of Machine Translation [3] and Visual Attention [2] [26]. Yang et al. proposed a review network [42] which does a number of review steps with some attention mechanism. This framework consists of three components, encoder, reviewer, and decoder. Further more, to improve the naturalness and diversity of the caption generation, a Conditional Generative Adversarial Networks (CGAN) based method [6] was proposed by Bo Dai et al., which jointly learns a generator to produce descriptions conditioned on images and an evaluator to assess how well a description fits the visual content.

2.6 Remote Sensing Image Captioning

In the case of remote sensing image, there are fewer works in the area of image captioning. One of the first research for remote sensing image captioning using deep learning techniques was published in *Deep Semantic Understanding of High Resolution Remote Sensing Image* [31] by Qu et al. in 2016. Their proposed model was based on an encoder-decode model using CNN and RNN or LSTMs, to extract features from high resolution images and then combining the associated textual information to generate sentences. The experiments were performed on two image captioning datasets i.e Sydney [44] and UCM datasets [41] both of which are very small data sets with 613 and 2100 examples respectively. They used pre-trained VGG19, VGG16 and GoogleNet networks to extract image features with RNNs and LSTMs to combine the textual information in a deep multimodel neural network model. The work published in “Can a Machine Generate Human-like Language Descriptions for a Remote Sensing Image?” [33], followed CNN based model for image feature extraction and a traditional template-based method for language generation part. They represent the remote sensing images with a combination of three things, that are the ground elements, its attributes and the relation between them. Based on these three representations, they form the language template. According to the authors Shi et al. [33], this approach was chosen because of the unavailability of captioning data set for remote sensing images. They use subjective evaluation techniques instead of objective evaluation methods as they argue that in objective evaluation, the score can depend on the style and expression of the annotator. Shortly after, with the paper: *Exploring Models and Data for Remote Sensing Image Caption Generation* [23] by Xiaoqiang Lu et al., they published the largest remote sensing captioning data set yet called RSICD, along with results of automatic captioning implementation for remote sensing images using deep learning techniques. They also used for their experimentation, 2 other data sets available for remote sensing captioning, that are UC Merced and Sydney Captioning data sets [31]. They use both traditional hand crafted feature extraction and deep learning(CNN) based methods for image understanding part and RNN and LSTM based models for language understanding parts. Experiments from multi-modal based and attention based methods were performed, making it the most insightful work considering deep learning based techniques.

During the last stages of this thesis we also came across yet 2 more papers, published closed to each other in this area of RS image captioning. First one is *Semantic Descriptions of High-Resolution Remote Sensing Images* [38], in which proposes a *collective semantic metric framework* (CSMLF), which gives 5 sentences to a text image instead of 1. In this method, they represent the images using pre-trainined convolutional neural network. They represent the sentences as a vectorized form of all the words of the sentence by GloVe, and then combine all the 5 sentences per image to have a collective sentence representation. In the end they use a metric learning method to embed images and its collective sentence to a common space. For testing, they calculate the distance between the test image and all collective sentences in training set to generate five sentences for a given test image. They argue that the complexity of remote sensing images should be described in more than just 1 sentence.

The second one is *Description Generation for Remote Sensing Images Using Attribute Attention Mechanism* [45] by Xiangrong Zhang et al., they have performed various state of art captioning methods on RSICD data set including the ones presented in [23] and [38]. In this method they use VGG16 pre-trained CNN model to extract image features and attributes for each RS image. They extract the image features from a lower level convolutional layer and high-level feature attributes from the deep fully connected layers of the CNN network. They represent the senteneces using LSTMs, where they maximize the log likelihood of generating the sentence word by word given the previous words, low-level features and high-level feature attributes of the image.

3 Neural Probabilistic and Attention Based Methods

In this section we discuss the existing frameworks used for image captioning. These methods were originally implemented for natural image captioning [37][40], but they were also experimented for remote sensing image captioning [23][33]. We also perform these experiments so that we could use them as baseline for comparing our proposed RS captioning methods. The first method is a neural probabilistic framework and the second one is an attention based framework. The overall architecture of these methods is shown in Figure 3.1. For training the model, the RS image is first passed though a pre-trained CNN encoder which extracts the features of the images. The Image features along with the corresponding captions are passed to the decoder to predict a caption one word at a time.

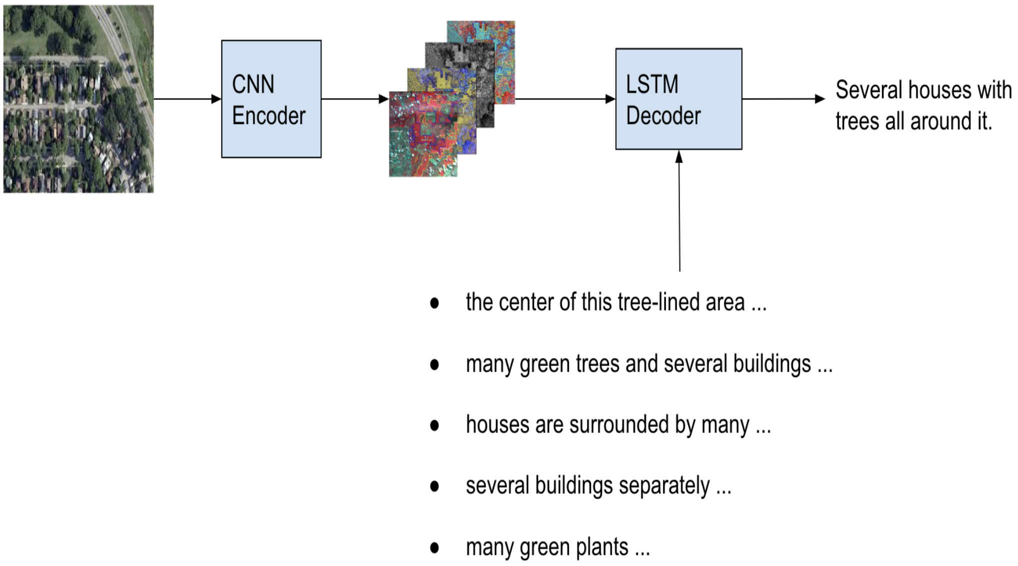

CNN Encoder
LSTM Decoder
Several houses with trees all around it.
the center of this tree-lined area ...
many green trees and several buildings ...
houses are surrounded by many ...
several buildings separately ...
many green plants ...

Figure 3.1: General outline of encoder-decoder architecture for remote sensing image captioning

3.1 Neural Probabilistic Framework

The first framework, follows the works of [37] which was done for natural image captioning and also followed by [23][33] for remote sensing image captioning. This method uses a neural probabilistic framework for captioning the images. As mentioned in [37], it is possible to create an end-to-end framework by directly maximizing the probability of the correct translation, for both training and inferencing. An encoder-decoder based architecture is used in this case, which uses a CNN for image encoding and an LSTM that encodes the variable length input into a fixed dimensional vector, and uses this representation to decode it to the required output sentences.

Let I be an input image, and C be its corresponding caption. In this approach, we try to maximize the probability of generating the correct description of an image, given an image caption pair (I, C) by using the following formulation:

$$\theta^* = argmax \sum logP(C|I; \theta) \tag{3.1}$$

where θ is our parameter. Since, C can be any sentence whose length is unbound, a chain rule can be applied to model the joint probability, over C_0,..., C_N. So, for length N the joint probability distribution will be:

$$\log p(C|I) = \sum_{t=0}^{N} \log p(C_t|I, C_0, ..., C_{t-1}). \tag{3.2}$$

(I, C) is our training example pair. Our training set consists of these image-caption pairs and we optimize it using stochastic gradient descent, over the sum of the log probabilities as described in equation(3.2).

For extracting the images features, we use pre-trained CNNs, which are trained on ImageNet [16] data. We used the fully connected layers of a ResNet [17] CNN model to perform our experiments, although we also used VGG19 [34] model in order to compare our results with other related papers. It is described in details in chapter 6. We represent each image feature as given below (equation 3.3)

$$x = CNN(I). \tag{3.3}$$

In order to represent the captions or sentences, we represent each word in the sentence by one-hot V dimensional word vector s_i, where V is the size of the vocabulary. Then the image and the words are mapped to a common embedding space, by a word embedding W_e (equation 3.4):

$$C' = W_e.s_i. \tag{3.4}$$

Caption C is then encoded as a sequence projected word vectors and has a variable length of $t-1$ (equation 4.5). This word vector is expressed by a fixed length hidden state or memory h_t. And the memory is updated after seeing a new input x_t by using a non-linear function f (equation 3.6) [37]:

$$C = (C_0, C_1, ..., C_t) \tag{3.5}$$

$$h_{(t+1)} = f(h_t, x_t). \tag{3.6}$$

As described in [37], we model $p(C_t|I, C_0, ..., C_{t-1})$ (equation 3.2) using LSTM [13], as it is clear by the experiments performed by [37], [23] and other papers on image captioning problems that, LSTMs perform better RNNs when it comes to language translation [15] [4] and sequence generation [12]. This is due the abilities of the LSTMs to deal with vanishing and exploding gradients problems.

The internal architecture of the LSTM is quite complex yet sophisticated. Figure 3.2 shows the internal architecture of the LSTM [37]. LSTM is capable of handling long dependencies by using the gates, which controls the flow of information to the networks. In the core of a LSTM [13], there is a memory cell c, which saves in its memory all the observed inputs upto a time step t. The cell state is controlled by the input, output and forget gates, to which layers are applied multiplicatively, which lets it keep the value from the respective gate if the gate is 1 or zero this value if the gate is 0. So, first the forget gate decides if the current cell value needs to be discarded, then the input gate decides if new input needs to be read and finally the output gate decides whether to output new value for the cell.

The gate and cell outputs of each step can be written as [37]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \tag{3.7}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \tag{3.8}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \tag{3.9}$$

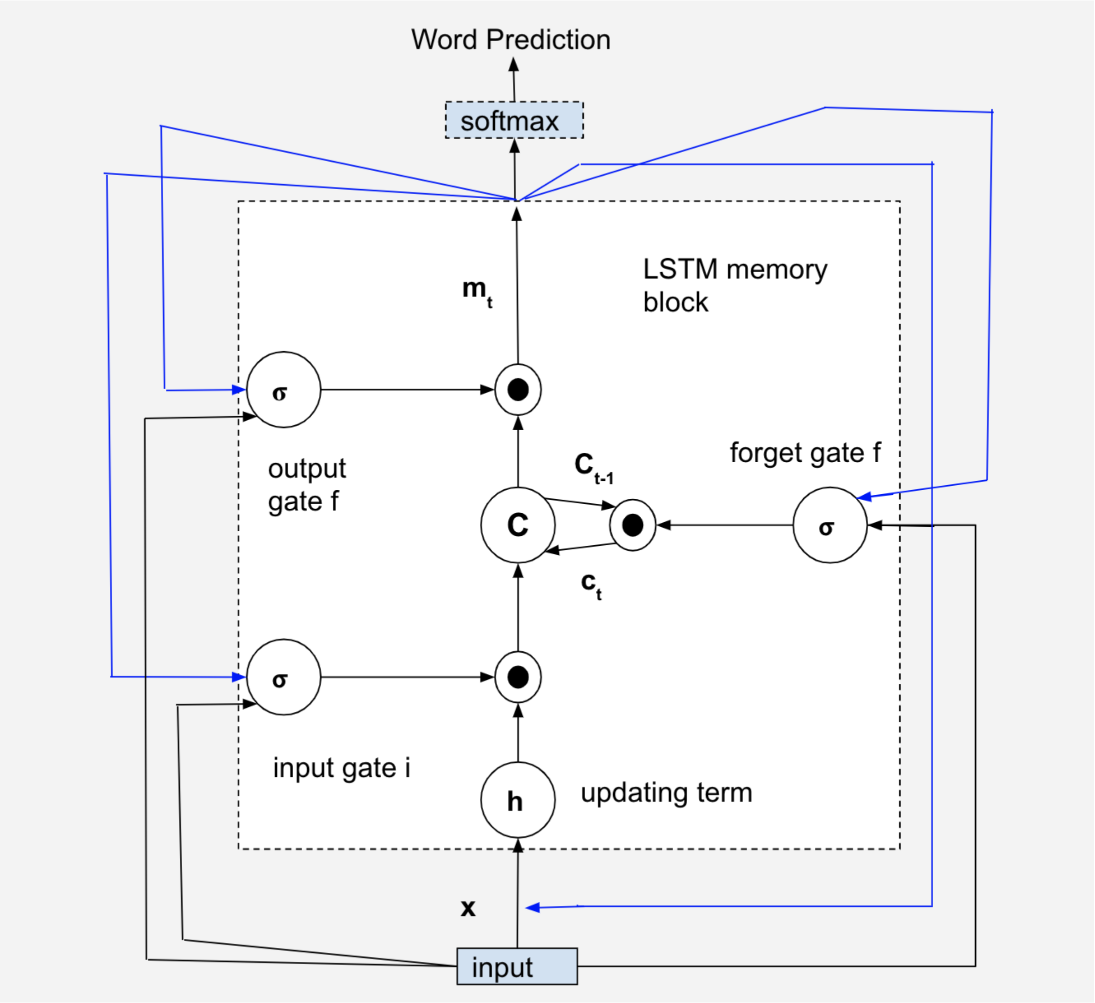


Figure 3.2: The LSTM memory block has a cell unit c which is controlled by input, output and forget gates. The blue lines shows the recurrent connections where the output m at time $t-1$ is fed back to the memory at time t from the three gates; the cell value is fed back via the forget gate; the word prediction at time $t-1$ is fed back along with the memory output m at time t to the Softmax for predicting the words. [37].

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \tag{3.10}$$

$$m_t = o_t \odot c_t \tag{3.11}$$

$$p_{t+1} = Softmax(m_t). \tag{3.12}$$

In these equations, $\odot$ represents product with a gate value and W matrices are training parameters. The multiplicative nature of the gates ensures a robust training of the LSTM and allows it deal with exploding and vanishing gradients [37]. Sigmoid $\sigma(.)$ and hyperbolic tangent $h(\cdot)$ are the nonlinearities in the equations. In the end, the output m_t is Softmaxed to generate a probability distribution p_t over all the words.

During the training phase, the LSTM predicts the caption word by word, after looking at the image and the previous words. If we think of the LSTM as an unrolled form, it would be as if a copy of LSTM memory is created for each time step and they share the same parameters along with the image and word of the sentence. The LSTM at the first time step gets the image encoding from the encoder, and first word from the caption. Then the output of the LSTM at each step is fed to the next step, for example m_{t-1} of the LSTM at time step $t-1$ is fed to the LSTM of time step t. These recurrent connections are transformed to feedforward procedure, where the procedure will look like [37]:

$$x_{-1} = CNN(I) \tag{3.13}$$

$$x_t = W_e.C_t \tag{3.14}$$

$$x_{-1} = CNN(I). \tag{3.15}$$

Here, I is the input image, $C = (C_0, ..., C_N)$ is the caption describing the image where each word is represented as one-hot vector of dimension of the size of the vocabulary. The image encodings from the CNN and the word embeddings W_e are mapped to the same space. C_0 and C_N are special start and end tokens to indicate the start and end of a caption. The end token is also used as a signal for the LSTM that all the words of the sentence have been generated and that its time to terminate the process.

The loss of our model is the sum of negative log likelihood of the prediction of correct word at each time step. The loss is minimised with respect to the parameters of LSTM, image and word embedding W_e as follows:

$$Loss(I|C) = -\sum_{t=0}^{N} \log p_t(C_t). \tag{3.16}$$

3.2 Attention Based Framework

The attention based framework mentioned in this section is based on the works of [40] who proposed two methods one is called as *soft attention* and the other is called as *hard attention*. Where the "soft" attention uses a deterministic mechanism, which is trainable by standard back-propagation methods and the “hard” attention uses a stochastic attention mechanism, which is trainable by maximizing an approximate variational lower bound or equivalently by *reinforce* [39]. These methods are incorporated based on the works of [2], [3] and [26]. Although, adding attention for vision related tasks goes back to the works by [21], [8] and [36].

In order to use the attention mechanism, we need to first represent our images and captions as almost the similar way as mentioned in previous section of *Neural Probabilistic Framework*. This framework is also an encoder-decoder based model. The encoder expects an image and extract the features from the lower convolutional layer which will allow us to focus on selected parts of the image, unlike previous method in which we extracted the features from the fully connected layer.

Let N be the number of image features extracted by the image encoder and a represent the feature vector, corresponding to the part of the image. D is the dimension of each of the feature vectors. The vector a can be formulated as:

$$a = \{\mathbf{a_1}, ..., \mathbf{a_N}\}, \mathbf{a_i} \in \mathbf{R^D}. \tag{3.17}$$

For representation of the captions, it is encoded as sequence of 1-of-K words [40], K being the size of the vocabulary and M being the length of caption (equation 3.18):

$$c = \{\mathbf{c_1}, ..., \mathbf{c_M}\}, \mathbf{c_i} \in \mathbf{R^K}. \tag{3.18}$$

The generating of sentences from the decoder part is done with help of LSTMs [13]. Attention is implemented by keeping in consideration of the previous time steps and deciding “where to look” in the current state. Based on this information the next word of the sentence is predicted [40].

The transformations within the LSTM can be represented in the following equations(3.19 to 3.21)[40]:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} Ey_{t-1} \\ h_{t-1} \\ z_t \end{pmatrix} \tag{3.19}$$

$T_{s,t} : \mathbf{R^s} \rightarrow \mathbf{R^t}$ is used to represent the transformation along with parameters

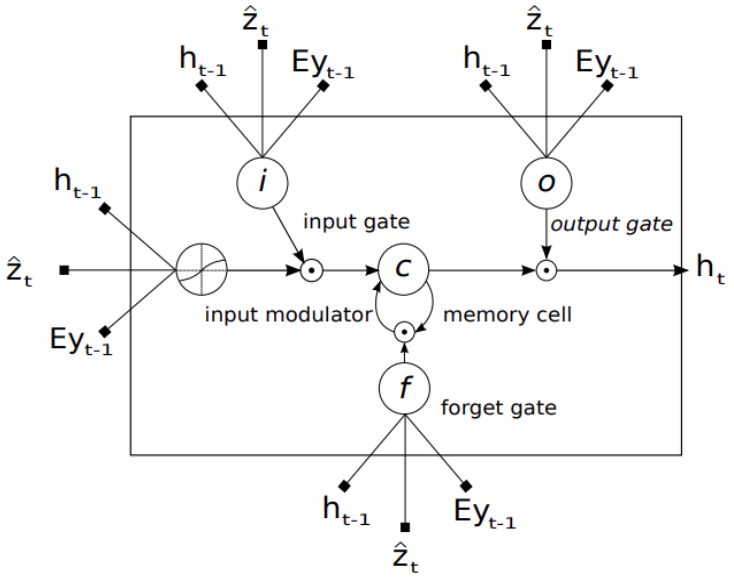


Figure 3.3: A LSTM cell, lines with bolded squares imply projections with a learnt weight vector. Each cell learns how to weigh the input components from the input gate, how to modulate it to the input modulator. It also learns to erase the memory cell via forget gate, and how to control the memory flow from the output gate. [43][40]

that are learned. $\hat{z}_t$ is the context vector that associates visual information with corresponding image region. E is the embedding matrix and m and n represents embedding and LSTM dimensions, respectively. σ denotes the sigmoid activation and $\odot$ denotes element-wise multiplication.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{3.20}$$

$$h_t = o_t \odot tanh(c_t). \tag{3.21}$$

Here, i_t, o_t and f_t are the input, output and forget gates of the LSTM and c_t and h_t are the memory and hidden state of the LSTM.

The context vector $\hat{z} \in \mathbf{R}^D$ is calculated from the annotation vectors a_i which is used to capture the location information for the corresponding image region. For each of the i^th regions, a positive weight α_i is calculated, which sums to one. This can then be interpreted by the model whether the region needs to focused for generating the next word. An attention model f_{att} is used to calculate the weight α_i for each annotation vector a_i, depending the previous hidden state h_{t-1} is as follows:

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{3.22}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{kt})} \tag{3.23}$$

where,

$$\sum_i \alpha_{ti} = 1. \tag{3.24}$$

Then, the context vector can be computed by,

$$\hat{z} = f(\{a_i\}, \{\alpha_i\}) \tag{3.25}$$

The function f will return a single vector for the given set of annotation vectors along with their corresponding weights.

The hidden and memory state of the LSTM are initialized by an average of the annotation vectors which are separately imported by two Multi-layer perceptrons(MLPs):

$$c_0 = f_{init,c}(\frac{1}{L}\sum_{i=0}^{L} a_i), \tag{3.26}$$

$$h_0 = f_{init,h}(\frac{1}{L}\sum_{i=0}^{L} a_i). \tag{3.27}$$

Then a deep output output layer is used to compute the output word probability given the state of the LSTM, the context vector and the previous word:

$$p(u_t|a, u_{t-1}) \propto exp(L_0(Eu_{t-1} + L_h h_t + L_z \hat{z}_t)) \tag{3.28}$$

Where $L_0 \in \mathbf{R}^{\mathbf{Kxm}}$, $L_h \in \mathbf{R}^{\mathbf{mxn}}$, and E are the learnable parameters which are randomly initialized.

During training, a doubly stochastic regularization is introduced as mentioned in [40]. The purpose of this is for the model to pay equal attention to all the parts of the image over the course of sentence prediction. We minimize the following penalized negative log-likelihood for the first part of loss calculation:

$$L_d = -Log(P(C|I)) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{M} \alpha_{ti})^2. \tag{3.29}$$

4 Proposed Methods for Remote Sensing Image Captioning

In this section we present our proposed methods for RS image captioning. We present three new approaches for captioning of remote sensing images. First one is *Summarized Captioning* and the second one is *Summarized Captioning with Attention*. All the proposed methods uses an additional summarization network for the captioning framework to work. Figure 4.1 shows the general architecture of the proposed summarized captioning approaches.

In our proposed methods, we first use a pre-trained ResNet152 CNN to extract the image features. The Image features along with the corresponding captions are passed to the decoder to predict a caption one word at a time. We combine the 5 captions per image and pass them to the summarization model to generate a summarized caption. Then, we either combine the outputs of the summarized model with the output from the decoder or we minimise the distance between them.

Adding summarization means to summarize the captions using a *Text Summarization* model [32] in order to generate one caption or rather one summarized caption and take that into consideration while we train our captioning model. The reason of adding this additional complexity is to deal with the shortcomings of our available dataset. As we know, that the captioning of remote sensing images are different than natural image captioning [23]. This is because, the objects are seen from the “View of God” [23]. The datasets that we use for image captioning are the high resolution satellite images and they have been annotated with 5 captions each. Although, as mentioned in the previous chapter of *Datasets*, they are quite often described with repeated captions in order to maintain overall uniqueness of the dataset. While it is reasonable to assume that all images cannot be described with exactly 5 sentences, but we need to have variations, if possible, in order for our model to learn and describe the images with proper variations yet in a generalized manner. Since we have the datasets designed in a way to be as normalized as possible, it doesn’t deal with the fact that specially when it comes

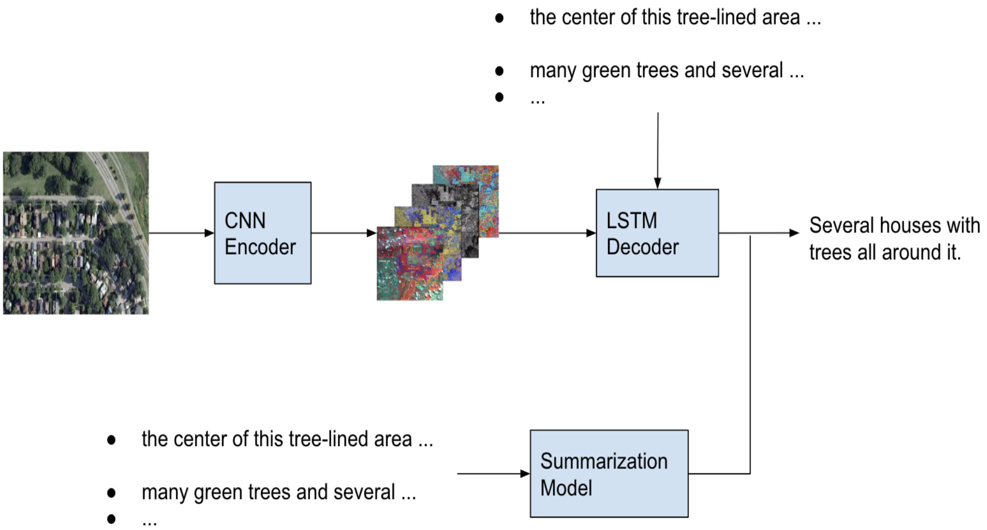

the center of this tree-lined area ...
many green trees and several ...
...
CNN Encoder
LSTM Decoder
Several houses with trees all around it.
the center of this tree-lined area ...
many green trees and several ...
...
Summarization Model

Figure 4.1: Encoder-decoder architecture for summarized captioning of remote sensing images

to remote sensing images, it is not possible for all images to be described in upto 5 captions. This normalization to have 5 captions for all images even though there are not actually 5 captions but are repeated to make it leveled could lead to over fitting of the captioning model. During our experimentation of the remote sensing data sets, we realised this problem while testing the generated captions, where certain types of images always had similar captions even though the images were different. This lead us to believe that there is more scope from the language processing perspective, which could help deal with this problem of data.

In our proposed method, we use a text summarizer along with our captioning model and use a bigger vocabulary than just the words from the captions of the dataset. The theory here is, to use the captions along with the summarized captions in order to overcome the over fitting problem, as well as to introduce newer vocabulary for our caption generator.

We use the summarization model to generate the probabilities of words in each position. In the first framework, we combine the outputs of the summarization probabilities with the outputs of the decoder. In the other 2 approaches, we calculate the KL divergence loss between the summarization probabilities and the decoder probabilities and we add it to the captioning loss.

4.1 Combining Summarized Captions to Image Captions (CSIC)

Like the the models described in the previous chapter, our model takes an image and generates a caption. We represent our image as I which generates a caption C of length L encoded words. We also need a pre-trained summarization model. In our case, we trained the summarization model ourselves, following the work provided in [32].

4.1.1 Encoder Architecture

We represent the image features using a convolutional neural Network(CNN). We use a pre-trained ResNet152 model for this feature extraction process. This pre-trained model acts as our encoder from which we remove the last fully connected layer and extract the feature vector of length 2048 for each image. In order to map these extracted features to a common embedding so that our decoder can receive this information, we pass the extracted feature vector though a linear layer which has 2048 input dimension and 512 output dimension or embed dimension W which is also the embed dimension for the decoder.

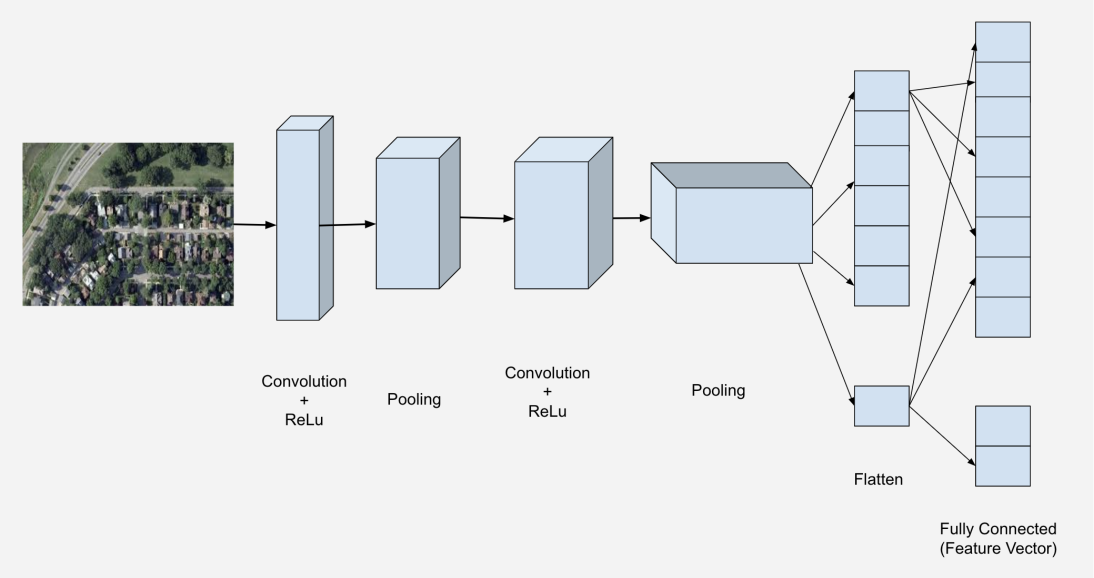


Figure 4.2: Feature extraction process of the encoder

The extracted features from CNN can be represented as follows:

$$x_{-1} = CNN_{ResNet152}(I). \tag{4.1}$$

4.1.2 Summarization Network

This network is used to generate a summarized caption from the initial set 5 caption per image. We only use a pre-trained model to generate this summarized caption. The summarization model is also an encoder-decoder model. Every word in the summarized sentence is in form of a one-hot vector representation w_t of dimension V which is the size of the vocabulary and of variable length t. This vector contains the probability scores for each word in the vocabulary, which upon decoding gives the summarized sentence. But, for this proposed method, we only need these one-hot score vector, which we will use in the decoder of our captioning model. This vector can be represented as below:

$$S = \{w_0, w_1, ..., w_t\}. \tag{4.2}$$

4.1.3 Decoder Architecture

The task of the decoder is to see encoded images coming from the encoder and generate a caption one word after other.

For this method, we first represent each word in the caption in form of a one-hot u_t of dimension V, where V is the size of the vocabulary and t is the length of the caption which can vary from caption to caption. The mathematical representation is given below:

$$C = \{u_0, u_1, ..., u_t\}. \tag{4.3}$$

Then we combine the vectors from equation 4.2 and 4.3, as they have same length of vocabulary size V, which can be represented by Y as below:

$$Y = \{w_0 + u_0, w_1 + u_1, ..., w_t + u_t\}, \tag{4.4}$$

$$Y = \{y_0, y_1, ..., y_t\}. \tag{4.5}$$

As mentioned previously, we use a bigger vocabulary of size $50,000$, to include more words during the training process. Then we project this combined word vector to the embedding space by a matrix E where it can be mapped to the image features. E is a e x V matrix, where e is the size of the embedding space:

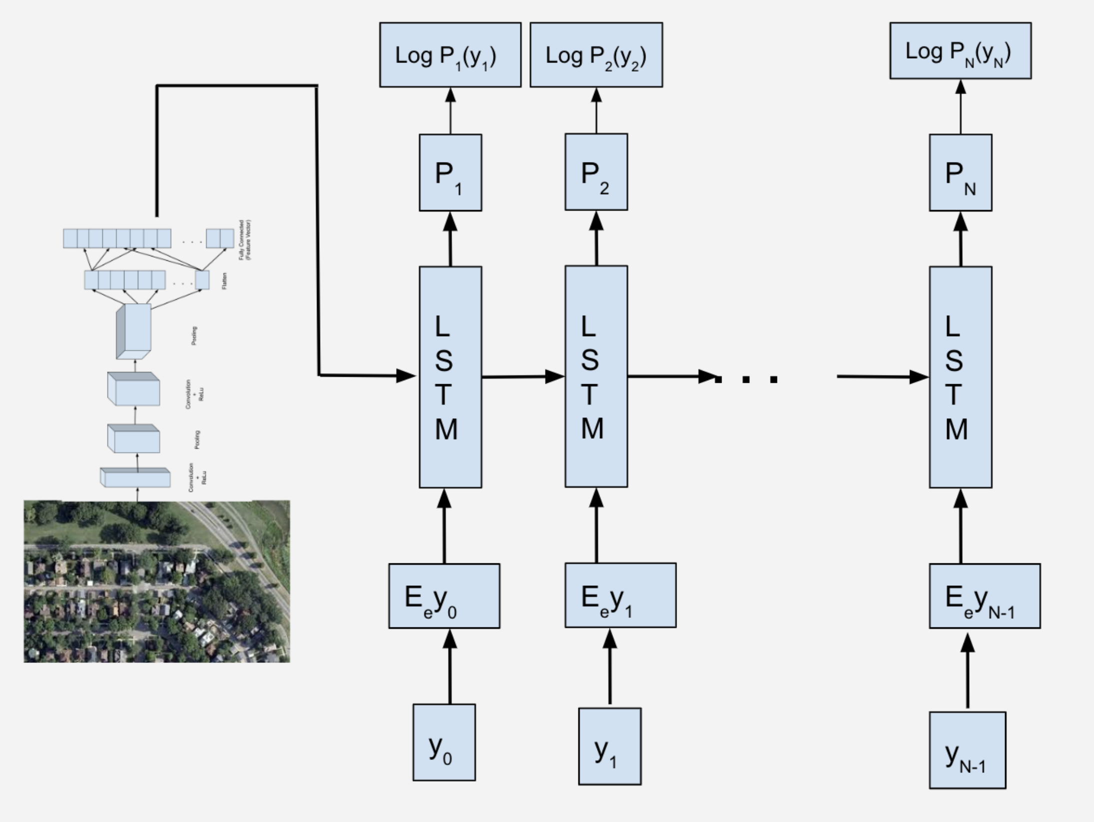


Figure 4.3: Sentence generation process of the decoder

$$y'_i = E.y_i, \tag{4.6}$$

$$Y = \{y'_0, y'_1, ..., y'_t\}. \tag{4.7}$$

In order to generate the captions, we maximize the probability of generating the correct caption, given an image:

$$logp(Y|I) = \sum_{t=0}^{N} \log p(y_t|I, y_0, ..., y_{t-1}). \tag{4.8}$$

The $t-1$ in equation(4.8) is the represented by a fixed length hidden state or memory h_t of the LSTM. Then this hidden state is updated after seeing a new input x_t (eq. 4.1) by using a non-linear function f:

$$h_{t+1} = f(h_t, x_t). \tag{4.9}$$

In the training, the model tries to predict each word of the caption one by one, after it has seen the image and its previous words as $p(y_t|I, y_0, ..., y_{t-1})$. This procedure can be formulated as:

$$h_t = \begin{cases} f(y'_t + E_d h_{t-1} + E_e x_{t-1}), t = 1, \\ f(y'_t + E_c h_{t-1}), t = 2, ..., N \end{cases} \tag{4.10}$$

In the above equation, E_d and E_e are the encoder and decoder learnable parameters respectively. The image features are only given once at step $t = 1$, the function f represents the LSTM process. h_t represents the output of state t and y'_t is the corresponding word in the sequence Y. The w'_1 is used to represent a special START token and w'_t is used to represent a special END token. The final output is softmaxed to generate the probability vector of the next word and in the end a complete sentence is generated after the model generates the END token signaling the end of sentence generation. The best model parameters are obtained in the training phase by minimizing the loss function:

$$loss(I, Y) = -\sum_{t=0}^{N} \log p(y_t). \tag{4.11}$$

4.2 Summarized Captioning with KL Divergence (SCKL)

4.2.1 Encoder Architecture

For this approach we also use a pre-trained ResNet152 model for extracting the features, where we remove the last fully connected layer of the network and extract a feature vector of length 2048 for each image. The image features can be represented as follows:

$$x_{-1} = CNN_{ResNet152}(I). \tag{4.12}$$

4.2.2 Summarization Network

Similar to the first approach, we use a pre-trained summarization model to generate a summarized caption from the set of 5 caption per image. The words from the summarized sentence is represented as a one-hot vector w_t of dimension V which is the size of vocabulary and of a variable length t. All the words in the summarized sentence S containing the probability scores for each word in the vocabulary, can be represented as $\{w_0$, w_1,..., $w_t\}$ so,

$$S = \{w_0, w_1, ..., w_t\}. \tag{4.13}$$

4.2.3 Decoder Architecture

In this approach, we represent each word in the caption in form of a one-hot u_t of dimension V, where V is the size of the vocabulary and t is the length of the caption which can vary from caption to caption. All the words for a caption C contains the probability scores for each word in the vocabulary and can be represented as $\{u_0,\ u_1,...,\ u_t\}$. To project this word vector to the embedding space we use an embedding matrix E where it can be mapped to the image features x. E is a e x V matrix, where e is the size of the embedding space,

$$u'_i = E.u_i, \tag{4.14}$$

$$C = \{u'_0, u'_1, ..., u'_t\}. \tag{4.15}$$

During training, first we maximize the probability of generating the correct caption, given an image, which can be formulated as:

$$logp(C|I) = \sum_{t=0}^{N} \log p(u_t|I, u_0, ..., u_{t-1}). \tag{4.16}$$

So, the model tries to predict each word of the caption one by one, after it has seen the image and its previous words: $p(u_t|I, u_0, ..., u_{t-1})$. This procedure can be represented mathematically as:

$$h_t = \begin{cases} f(u'_t + E_d h_{t-1} + E_e x_{t-1}), t = 1, \\ f(u'_t + E_c h_{t-1}), t = 2, ..., N \end{cases} \tag{4.17}$$

In the above equation, E_d and E_e are the encoder and decoder learnable parameters respectively. The image features are only given once at step $t = 1$, the function f represents the LSTM process. h_t represents the output of state t and y'_t is the corresponding word in the sequence Y. The w'_1 is used to represent a special START token and w'_t is used to represent a special END token. The final output is softmaxed to generate the probability vector of the next word and in the end a complete sentence is generated after the model generates the END token signaling the end of sentence generation. The best model parameters are obtained in the training phase by minimizing the loss function:

$$L_{CE}(C|I) = -\sum_{t=0}^{N} \log p(u_t). \tag{4.18}$$

Since we have 2 distributions of the same size, that is the captioning and summarized caption (equation 4.12 and 4.15) of the size of the vocabulary, we try to minimize the distance between these using Kullback–Leibler divergence as follows:

$$L_{KL}(C|S) = -\sum_{t=0}^{N} \ln\left(\frac{u_t}{w_t}\right) u_t. \tag{4.19}$$

Finally, we combine these 2 losses, in order for our model to consider the summarized captions as well,

$$L = L_{CE}(C|I) + L_{KL}(C|S). \tag{4.20}$$

4.3 Summarized Captioning with Attention and KL Divergence (SCAttKL)

In this method we use an attention based model as proposed by [40]. They mentioned two kinds of attentions, one is a "soft" deterministic method trainable by standard back-propagation and other is "hard" stochastic attention method trainable by maximizing an approximate variational lower bound. In our approach, we only use the "soft" attention method. The reason for using attention could be seen as a way for the model to give additional importance to the parts of image encoding that could be relevant.

4.3.1 Encoder Architecture

Unlike previous approaches, where we used a fully connected layer to represent the image features, in this case we extract the features from the lower convolutional layer, which will help representing the correspondence between the feature vectors and the regions of the 2-D image. This feature extractor will produce L vectors also called as annotation vectors. Each of these vectors are a D-dimensional representation of the corresponding part of the image,

$$a = \{\mathbf{a_1}, ..., \mathbf{a_L}\}, \mathbf{a_i} \in \mathbf{R^D}. \tag{4.21}$$

4.3.2 Summarization Network

Summarization network is same as the first 2 approaches. To generate a summarized caption from the set of 5 captions a pre-trained summarization model is used. The words from the summarized sentence is represented as a one-hot vector w_t of dimension V which is the size of vocabulary and of a variable length

t. All the words in the summarized sentence S containing the probability scores for each word in the vocabulary, can be represented as $\{w_0, w_1,..., w_t\}$.

$$S = \{w_0, w_1, ..., w_t\} \tag{4.22}$$

4.3.3 Decoder Architecture

The representation of words in the caption is in form of a one-hot vector u_t of dimension V, where V is the size of the vocabulary and t is the length of the caption which can vary from caption to caption. All the words for a caption C contains the probability scores for each word in the vocabulary and can be represented as $\{u_0, u_1,..., u_t\}$,

$$C = \{u'_0, u'_1, ..., u'_t\}. \tag{4.23}$$

In the attention based approach, the information extracted from the previous state is used by the model to decide "where" to look in the current state. In the soft attention method, the weighted encoding of different parts of the image is used to recognize the parts "where" to look in the image.

To include the attention mechanism, a context vector $\hat{z} \in \mathbf{R}^D$ is calculated from the annotation vectors a_i which is used to capture the location information for the corresponding image region. For each of the $i^t h$ regions, a positive weight α_i is calculated, which sums to one. This can then be interpreted by the model whether the region needs to focused for generating the next word. An attention model f_{att} is used to calculate the weight α_i for each annotation vector a_i, depending the previous hidden state h_{t-1} :

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{4.24}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{kt})} \tag{4.25}$$

where,

$$\sum_i \alpha_{ti} = 1 \tag{4.26}$$

Then, the context vector can be computed by,

$$\hat{z} = f(\{a_i\}, \{\alpha_i\}). \tag{4.27}$$

The function f will return a single vector for the given set of annotation vectors along with their corresponding weights.

The hidden and memory state of the LSTM are initialized by an average of the annotation vectors which are separately imported by two Multi-layer perceptrons(MLPs):

$$c_0 = f_{init,c}(\frac{1}{L}\sum_{i=0}^{L} a_i), \tag{4.28}$$

$$h_0 = f_{init,h}(\frac{1}{L}\sum_{i=0}^{L} a_i) \tag{4.29}$$

Then a deep output output layer is used to compute the output word probability given the state of the LSTM, the context vector and the previous word:

$$p(u_t|a, u_{t-1}) \propto exp(L_0(Eu_{t-1} + L_h h_t + L_z \hat{z}_t)). \tag{4.30}$$

Where $L_0 \in \mathbf{R}^{\mathbf{Kxm}}$, $L_h \in \mathbf{R}^{\mathbf{mxn}}$, and E are the learnable parameters which are randomly initialized.

4.3.4 Doubly Stochastic Attention

During training, a doubly stochastic regularization is introduced as mentioned in [40]. The purpose of this is for the model to pay equal attention to all the parts of the image over the course of sentence prediction. We minimize the following penalized negative log-likelihood for the first part of loss calculation:

$$L_d = -Log(P(C|I)) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{M} \alpha_{ti})^2. \tag{4.31}$$

4.3.5 Kullback–Leibler Divergence

To include a summarization attention, as we have 2 distributions of the same size, that is the Captioning and summarized caption (equation 4.22 and 4.23) of the size of the vocabulary, we try to minimize the distance between these using Kullback–Leibler divergence as follows:

$$L_{KL}(C|S) = -\sum_{t=0}^{N} \ln\left(\frac{u_t}{w_t}\right) u_t. \tag{4.32}$$

Finally, we combine these 2 losses from equation 4.31 and 4.33, in order for our model to consider the summarized captions as well,

$$L = L_d + L_{KL}(C|S). \tag{4.33}$$

5 Dataset for Remote Sensing Image Captioning

In this section we discuss about the data sets we have used to perform our experiments. We have designed our frameworks to use these datasets and compare other related works of remote sensing image captioning. To the best of our knowledge there are 3 data sets which can be used for remote sensing image captioning which are: RSICD [23], UCM-captions and Sydney-Captions proposed by [31] based on original data sets by [41] and [44]. We also use a text summarization model within our proposed framework which we also train from scratch, for which we have used Gigaword dataset [20].

5.1 Remote Sensing Image Captioning Dataset(RSICD)

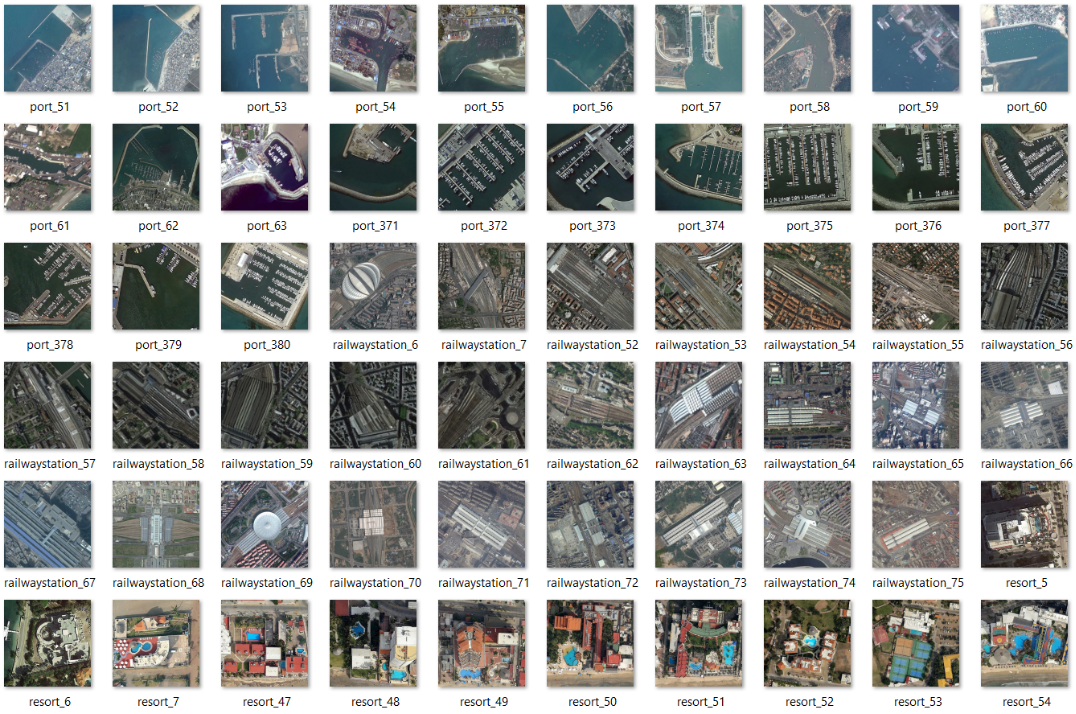


Figure 5.1: Example of images in RSICD dataset [23].

RSICD [23] is one of the biggest data set available till now, for remote sensing image captioning. This data set consists of over ten thousand images from various sources such as, Google Earth, Baidu Map, MapABC, Tianditu. There are 10921 images in total with 224x224 pixels sizes of various resolutions. The examples of images in RSICD dataset can be seen in Figure 5.1

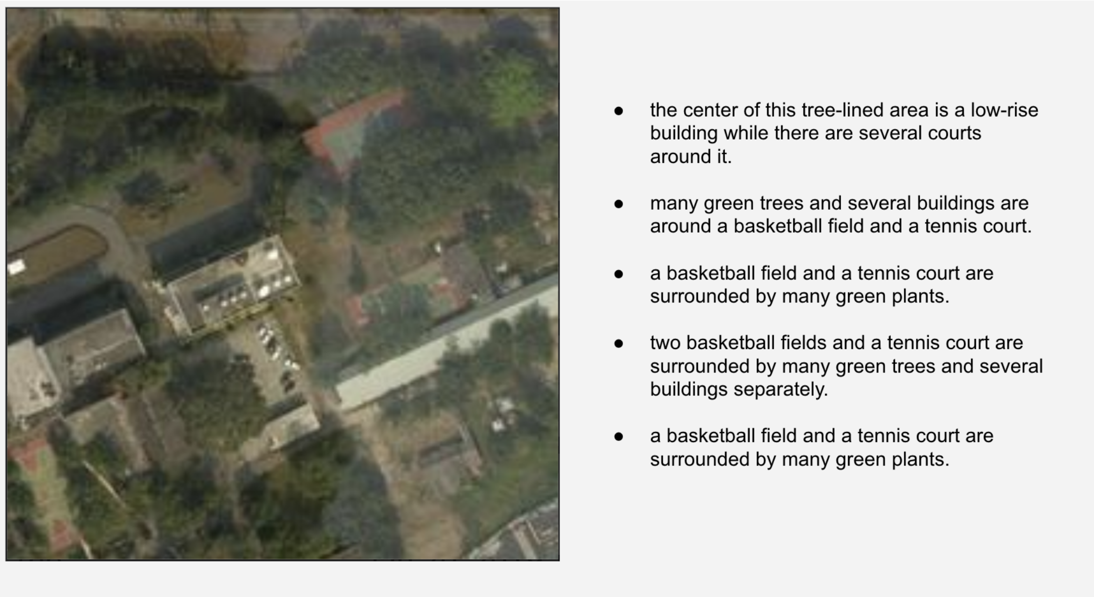
Table 5.1: Number of images of each class in RSICD datasets (the total number of images is 10921)

class	Number	class	Number
Airport	420	Farmland	370
Bare Land	310	Forest	250
Baseball Field	276	Industrial	390
Beach	400	Meadow	280
Bridge	459	Medium Residential	290
Center	260	Mountain	340
Church	240	Park	350
Commercial	350	School	300
Dense Residential	410	Square	330
Desert	300	Parking	390
Playground	1031	Pond	420
Viaduct	420	Railway Station	260
Sparse Residential	300	Storage Tanks	396
Resort	290	River	410
Port	389	Stadium	290

Each image is described with 5 captions, although not all of the images have 5 unique captions. In total there are 24333 sentences forming a vocabulary of 3323 words. The distribution of captions per image are as follows: 724 images have 5 different captions, 1495 image have 4 different captions, 2182 images have 3 different captions, 1667 images have 2 different captions and 5853 images have only 1 caption. However, [23] have mentioned that they have extended the number of caption in cases where images are described with less than 5 captions, by randomly duplicating the captions for those cases. Which makes a total of 54605 number of captions in the data set. Figure 5.2 shows some examples of images and their captions of the RSICD dataset.

5.2 UCM Captions Dataset

The original UCM dataset [41] has images from 21 land-use classes selected from aerial orthoimagery with a pixel resolution of one foot. The classes includes



- the center of this tree-lined area is a low-rise building while there are several courts around it.
- many green trees and several buildings are around a basketball field and a tennis court.
- a basketball field and a tennis court are surrounded by many green plants.
- two basketball fields and a tennis court are surrounded by many green trees and several buildings separately.
- a basketball field and a tennis court are surrounded by many green plants.

Figure 5.2: Example of images and corresponding five sentences per image selected from RSICD dataset. [23].

agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks and tennis court. Each class contains 100 images of 256x256 pixels with a resolution of 0.3048m, making a total of 2100 images.

These images were manually extracted from many large images of the United States Geological Survey (USGS) National Map Urban Area Imagery from the regions: Birmingham, Boston, Buffalo, Columbus, Dallas, Harrisburg, Houston, Jacksonville, Las Vegas, Los Angeles, Miami, Napa, New York, Reno, San Diego, Santa Barbara, Seattle, Tampa, Tucson, and Ventura.

In order to perform the task of image captioning, [31] captioned the images from UCM dataset [41]. They annotated each image in UCM dataset with 5 sentences, which sums up to 10500 total number of sentences. All the 5 sentences per image are different from each other, however, sentences of images of same class are similar in nature.

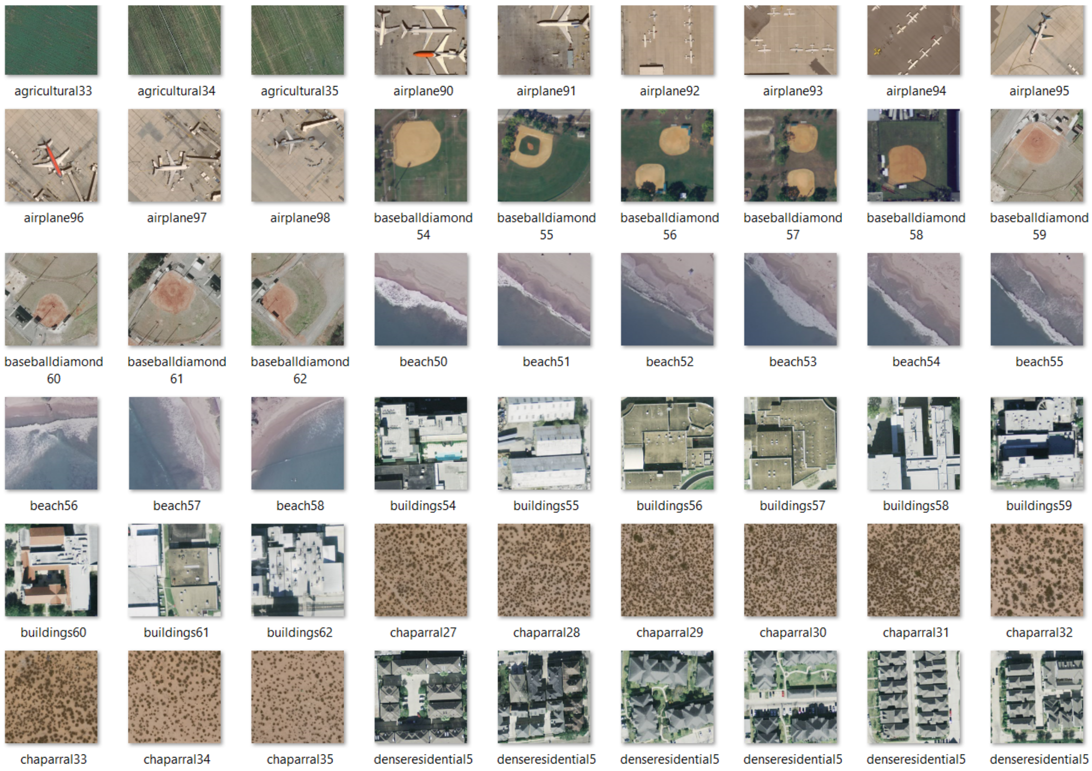
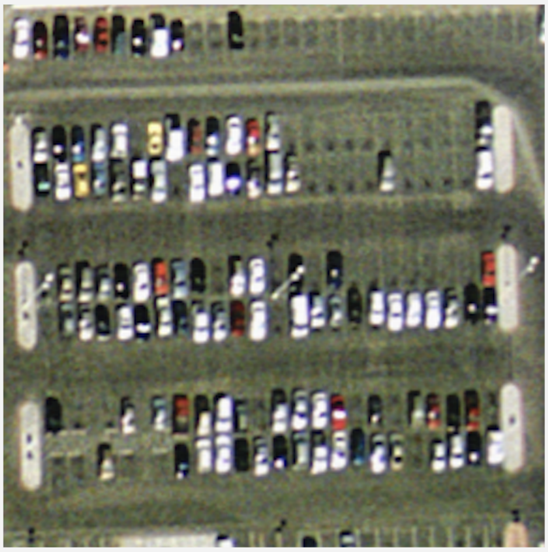


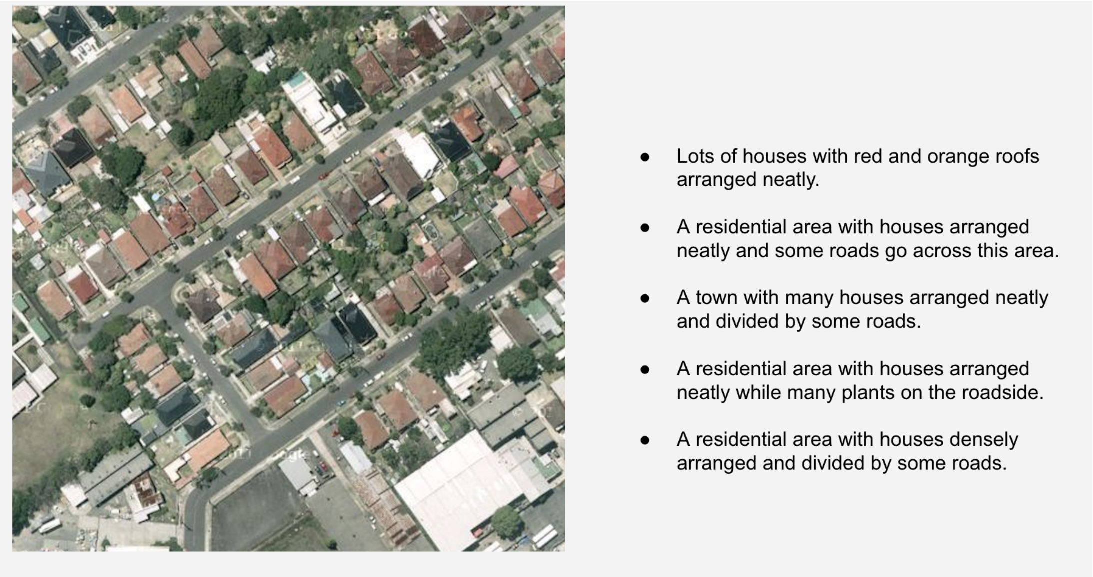
Figure 5.3: Example of images selected from UCM image dataset [41].



- Lots of cars parked neatly in the parking lot.
- Lots of cars parked in lines in the parking lot.
- Many cars parked neatly with many empty parking spaces in the parking lot.
- It is a parking lot with many cars parked neatly and many parking spots are free.
- Many cars parked in lines with many free parking spots in the parking lot.

Figure 5.4: Example of images and corresponding five sentences per image selected from UCM captioning dataset [31].

5.3 Sydney Captions Dataset



- Lots of houses with red and orange roofs arranged neatly.
- A residential area with houses arranged neatly and some roads go across this area.
- A town with many houses arranged neatly and divided by some roads.
- A residential area with houses arranged neatly while many plants on the roadside.
- A residential area with houses densely arranged and divided by some roads.

Figure 5.5: Example of images and corresponding five sentences per image selected from Sydney captions dataset [31].

The Sydney dataset [44] was build from a large satellite image of Sydney, Australia, which was acquired from Google Earth. The image was of 18000x14000 pixels and a spatial resolution of 0.5 m. There are seven classes in the data set, which are: residential, airport, meadow, rivers, ocean, industrial, and runway.

Table 5.2: Number of images of each class in Sydney datasets (the total number of images is 613)

class	Number
residential	242
airport	22
meadow	50
rivers	45
ocean	92
industrial	96
runway	66

It has a total of 613 images. However, similar to UCM data set ([41]), this dataset was also annotated by [31] to have 5 captions per image. In total it has 3065 captions.

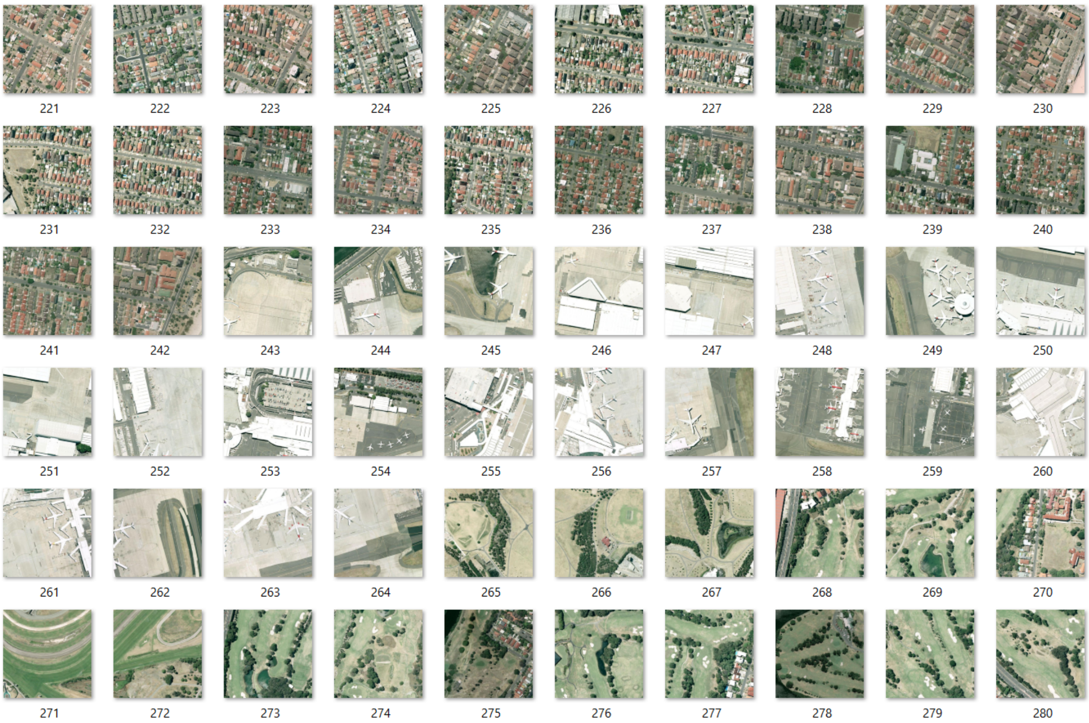
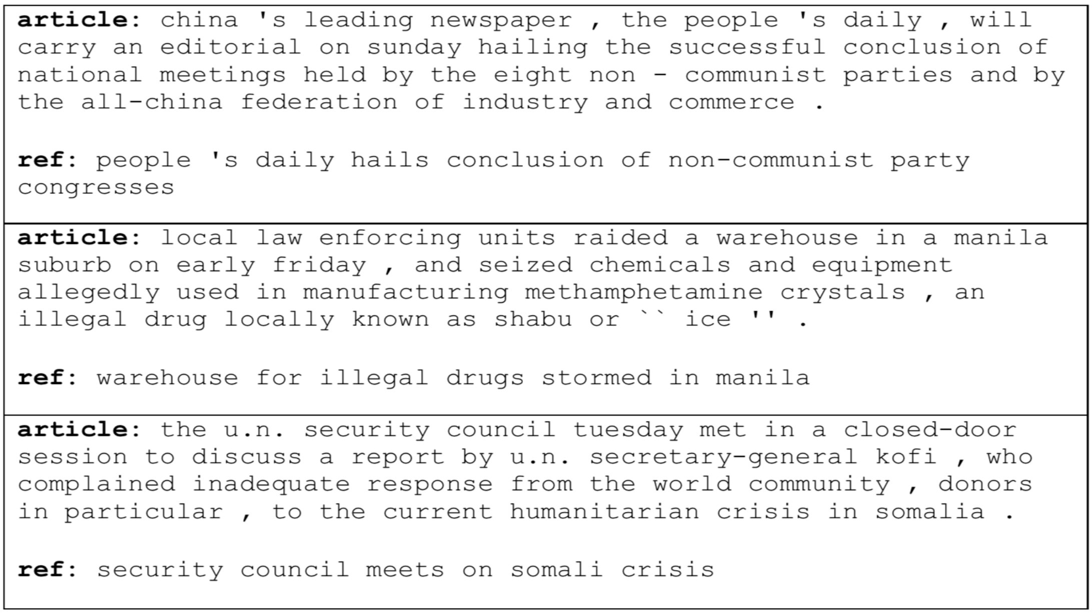


Figure 5.6: Example of images selected from Sydney image dataset


article: china 's leading newspaper , the people 's daily , will carry an editorial on sunday hailing the successful conclusion of national meetings held by the eight non - communist parties and by the all-china federation of industry and commerce .

ref: people 's daily hails conclusion of non-communist party congresses

article: local law enforcing units raided a warehouse in a manila suburb on early friday , and seized chemicals and equipment allegedly used in manufacturing methamphetamine crystals , an illegal drug locally known as shabu or `` ice '' .

ref: warehouse for illegal drugs stormed in manila

article: the u.n. security council tuesday met in a closed-door session to discuss a report by u.n. secretary-general kofi , who complained inadequate response from the world community , donors in particular , to the current humanitarian crisis in somalia .

ref: security council meets on somali crisis

Figure 5.7: Examples from the Gigaword dataset

5.4 Gigaword Dataset

Gigaword dataset used in this experiment [11] [28], consists of standard Gigaword, preprocessed with Stanford CoreNLP tools [24]. We need this dataset for the task of summarizing the captions, which is a part of our proposed RS image captioning approach. Gigaword is a corpus of article-headline pair which consists of around 9.5 million articles sourced from various news services.

6 Experimental Results

In this section, we present all the experiments that we have performed and their results. We have experimented five different approaches, two of which are the current state of art approaches and three proposed ones, with varying parameters of training. We used the 3 different RS image captioning datasets mentioned in chapter 5. For all the datasets, we used the training, validation and test split as given in the respective datasets, which are 80% for training, 10% for validation and 10% for test sets. All the experiments were performed using this split. First, we present the results from the neural probabilistic and attention frameworks and compare the obtained results with the ones shown in [23] and [45]. Finally, we present the results obtained by our proposed methods and compare the results with other approaches.

The evaluation metric used for evaluating our experiments is BLEU. BLEU stands for bilingual evaluation understudy which is an algorithm to evaluate the quality of machine translated texts automatically[30]. It measures the closeness of machine translation with one or more reference human translation according to a numerical metrics, which is proposed in the paper [30]. As a result it generates the score which determines the performance of the machine translated texts. Essentially, it compares n-grams of the generated sentence with the n-grams of the reference translation and then counts the number of matches. So, the score will be better if the machine translation is closer to a human translation.

6.1 Results of Neural Image Captioning

To perform this experiment, we followed the approaches of [37] [23] using the RS captioning datasets and compared the results with [23] as they also performed similar experiment with same datasets. This method requires a CNN to extract the features. We used VGG19 [34] and ResNet152 [17] pre-trained models for the feature extraction part of this experiment and LSTM [13] for the language generation part. We set the embedding and hidden state dimension of the LSTM as 512 and learning rate as 0.0001. The training was done on NVIDIA T4 GPU which took about one day to train. As we can see in table 6.1, the model trained with

ResNet152 pre-trained model performs better than VGG19 pre-trained model, which is why we decided to perform all the other experiments using ResNet152 pre-trained CNN model for image feature extractor.

We compared our results with [23] paper. They have used VGG16, VGG19, AlexNet and GoogleNet pre-trained CNN models for feature extraction. According to their findings, the model trained with VGG19 pre-trained CNN has the highest bleu scores among the models with other pre-trained CNNs. We compare our scores of VGG19 pre-trained model with the mentioned paper [23] and see that our experiment has better results. And since we see that the model trained with ResNet152 pre-trained CNN performs better than the one trained with VGG19, it was obvious to choose ResNet152 for the rest of the experiments.

Table 6.1: Results of NIC approach with 2 pre-trained CNN models for RSICD dataset.

	bleu1	bleu2	bleu3	bleu4
VGG19	0.6127	0.4341	0.3420	0.2717
ResNet152	0.6380	0.4623	0.3609	0.2890

Table 6.2: Results of NIC approach with pre-trained CNN model for UCM dataset.

	bleu1	bleu2	bleu3	bleu4
ResNet152	0.8035	0.7167	0.6572	0.6030

Table 6.3: Results of NIC approach with pre-trained CNN model for Sydney dataset.

	bleu1	bleu2	bleu3	bleu4
ResNet152	0.7755	0.6666	0.5803	0.5023

6.2 Results of Attention Based Captioning

For attention based captioning, we have tested the framework proposed in [40] for soft attention approach. We trained our model with stochastic gradient descent using Adam algorithm [18]. We compared our results for this approach with [23] and found that we have significantly better results for bleu1 through bleu4 for

RSICD and UCM datasets, and a slightly less bleu4 score for Sydney dataset. Since bleu1, bleu2 and bleu3 scores are much better for Sydney dataset with our experiment, we can safely say that our implementation of soft attention works very well overall.

Table 6.4: Results of attention based captioning method for all three datasets.

Dataset	bleu1	bleu2	bleu3	bleu4
RSICD	0.6541	0.4794	0.3763	0.3028
UCM	0.8286	0.7512	0.6939	0.6389
Sydney	0.8035	0.6985	0.6210	0.5552

As far as model parameters are concerned, we used an embed size of 512 and hidden state of LSTM also as 512. Learning rate is initialized as 0.00008 in the beginning, which decays by 20% if there are 8 consecutive epochs without any improvements in blue4. The experiment was performed on 3 different RS captioning datasets, the result of which can be seen in table 6.4. The training was conducted on NVIDIA T4 GPU which took around one day to train.

6.3 Results of Proposed Method of Combining Summarized Captions to Image Captions (CSIC)

In this proposed method for RS image captioning, we use two pre-trained models, one is ResNet152 CNN model for extracting image features and another for summarization network to provide summarized captions. We add the outputs from the summarization model with the output of captioning model at each step. We train both the summarization model and the captioning model on a common vocabulary set. We decided on the size of the vocabulary to be 50000. First, we train the summarization model with the following model parameters: Learning rate of 0.001, Embed size of 256 and the hidden dimension of 512. It took about 3 to 4 days for training. Then we train our captioning model. For this training, we initially use a learning rate of 0.00008. We use early stopping with bleu score for terminating the training. We keep track of the bleu score per epoch, then terminate the training where the bleu score stops rising after a certain number of epochs for example in this case we stopped training if there was no improvement in bleu score for 20 consecutive epochs.

Table 6.5: Results of proposed method of combining summarized captions to image captions (CSIC) for all three datasets.

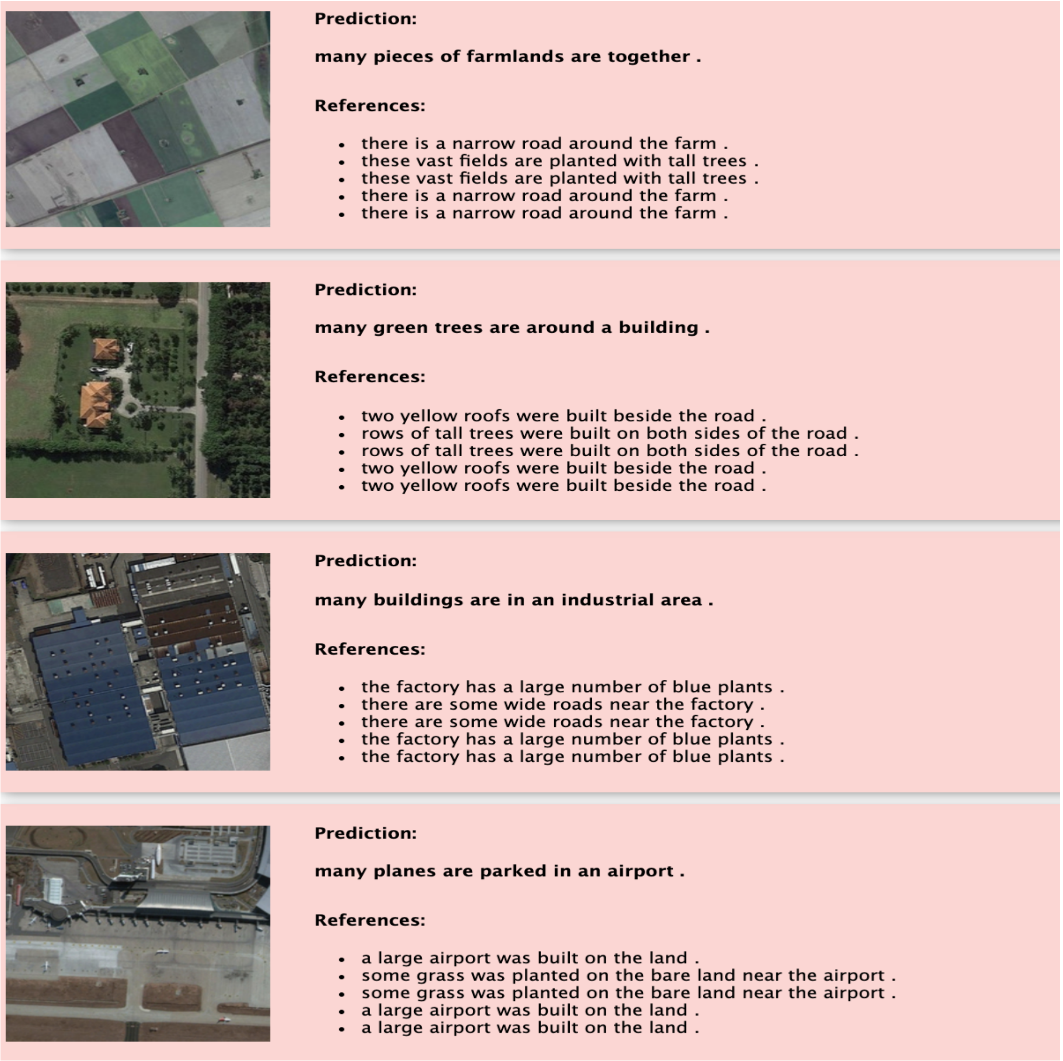
Dataset	bleu1	bleu2	bleu3	bleu4
RSICD	0.6327	0.4533	0.3501	0.2774
UCM	0.7949	0.7092	0.6498	0.5923
Sydney	0.7832	0.6865	0.6152	0.5458

We can see some of the outputs from the test set of RSICD dataset in Figures 6.1 to 6.3. The red colour background of the figure represents that the bleu score is between 0.0 to 0.2, the yellow colour indicates that the bleu score is between 0.2 to 0.6 and the green colour background indicates that the bleu score is between 0.6 to 1.0. We also do the same, for displaying some of the actual outputs generated by each of the proposed model.

Figure 6.1 shows the output captions generated for some of the test images of the RSICD dataset. These examples have bleu scores less than 0.2, but they are not incorrect captions. In each of these example cases, the predicted captions are correct although not very similar to the reference captions. Similarly, the second Figure 6.2 which has example outputs with bleu scores less than 0.6 are also very accurate. In Figure 6.3, which has examples of output generated by the model with bleu score more than 0.6 are precise and informative, and highly correlated with the reference captions. An image can be described in many ways, and measuring the correctness of these generated captions with BLEU or any other metric is not sufficient. For this reason, we also perform a manual evaluation for our proposed approaches. To do this, we manually checked all the generated captions from the test set and divided them into three categories following the approaches from [38]. We create three different buckets and each generated caption per image is assigned to precisely one of them. If the generated caption is grammatically correct and has most of the objects or contents of the image described, then we put it to the “Correct” list. If the generated caption is partially correct, which means, it missed few details of the image, or if the description has slight grammatical incorrectness, then we put it to “Partially Correct” list. If the generated sentences are grammatically incorrect or only captured just a few contents of the image, then we assign it to the “Incorrect” list. In table 6.6, we present the manual evaluation result for the method of combining summarized captioning with image captioning. We can see that the majority of the generated captions are “correct” prediction of the image.

Table 6.6: Results of manual evaluation for the method of combining summarized captioning with image captioning for RSICD dataset.

Bucket	Percentage
Correct	63.13
Partially Correct	27.17
Incorrect	9.70



Prediction:

many pieces of farmlands are together .

References:

- there is a narrow road around the farm .
- these vast fields are planted with tall trees .
- these vast fields are planted with tall trees .
- there is a narrow road around the farm .
- there is a narrow road around the farm .

Prediction:

many green trees are around a building .

References:

- two yellow roofs were built beside the road .
- rows of tall trees were built on both sides of the road .
- rows of tall trees were built on both sides of the road .
- two yellow roofs were built beside the road .
- two yellow roofs were built beside the road .

Prediction:

many buildings are in an industrial area .

References:

- the factory has a large number of blue plants .
- there are some wide roads near the factory .
- there are some wide roads near the factory .
- the factory has a large number of blue plants .
- the factory has a large number of blue plants .

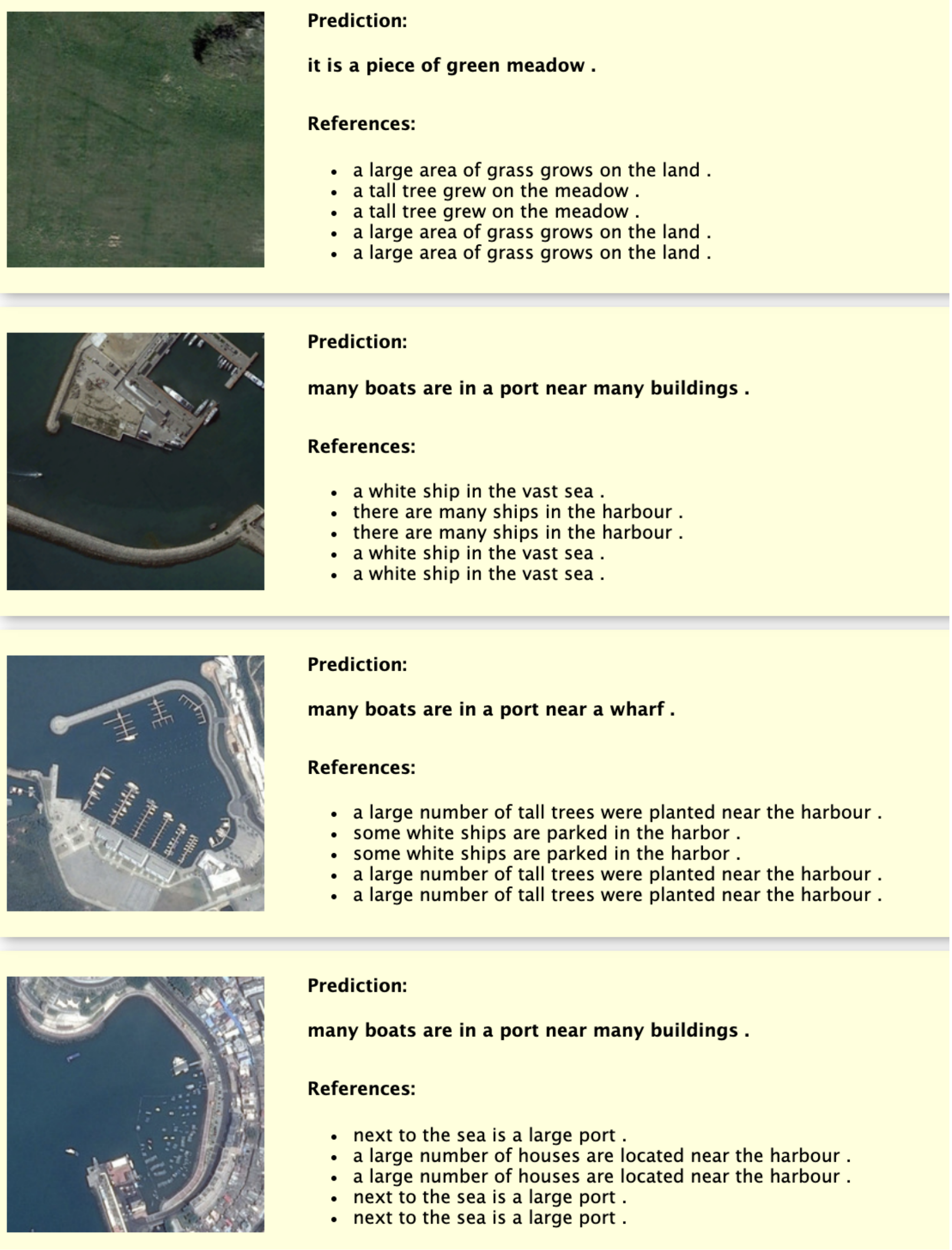
Prediction:

many planes are parked in an airport .

References:

- a large airport was built on the land .
- some grass was planted on the bare land near the airport .
- some grass was planted on the bare land near the airport .
- a large airport was built on the land .
- a large airport was built on the land .

Figure 6.1: The example set 1 of the output generated by the approach of combining summarized captions to image captions for some test images from RSICD dataset.



Prediction:

it is a piece of green meadow .

References:

- a large area of grass grows on the land .
- a tall tree grew on the meadow .
- a tall tree grew on the meadow .
- a large area of grass grows on the land .
- a large area of grass grows on the land .

Prediction:

many boats are in a port near many buildings .

References:

- a white ship in the vast sea .
- there are many ships in the harbour .
- there are many ships in the harbour .
- a white ship in the vast sea .
- a white ship in the vast sea .

Prediction:

many boats are in a port near a wharf .

References:

- a large number of tall trees were planted near the harbour .
- some white ships are parked in the harbor .
- some white ships are parked in the harbor .
- a large number of tall trees were planted near the harbour .
- a large number of tall trees were planted near the harbour .

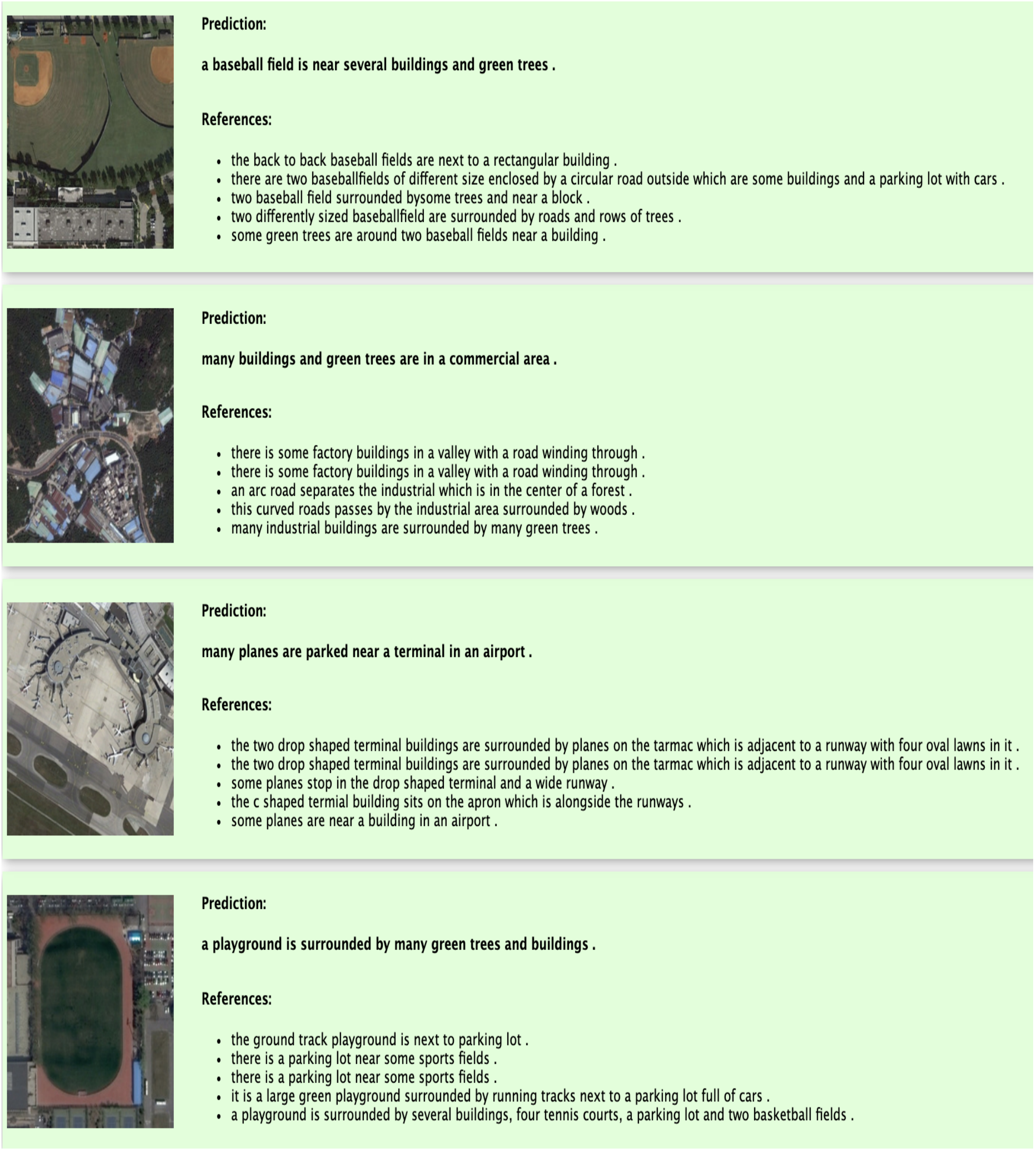
Prediction:

many boats are in a port near many buildings .

References:

- next to the sea is a large port .
- a large number of houses are located near the harbour .
- a large number of houses are located near the harbour .
- next to the sea is a large port .
- next to the sea is a large port .

Figure 6.2: The example set 2 of the output generated by the approach of combining summarized captions to image captions for some test images from RSICD dataset.



Prediction:

a baseball field is near several buildings and green trees .

References:

- the back to back baseball fields are next to a rectangular building .
- there are two baseballfields of different size enclosed by a circular road outside which are some buildings and a parking lot with cars .
- two baseball field surrounded bysome trees and near a block .
- two differently sized baseballfield are surrounded by roads and rows of trees .
- some green trees are around two baseball fields near a building .

Prediction:

many buildings and green trees are in a commercial area .

References:

- there is some factory buildings in a valley with a road winding through .
- there is some factory buildings in a valley with a road winding through .
- an arc road separates the industrial which is in the center of a forest .
- this curved roads passes by the industrial area surrounded by woods .
- many industrial buildings are surrounded by many green trees .

Prediction:

many planes are parked near a terminal in an airport .

References:

- the two drop shaped terminal buildings are surrounded by planes on the tarmac which is adjacent to a runway with four oval lawns in it .
- the two drop shaped terminal buildings are surrounded by planes on the tarmac which is adjacent to a runway with four oval lawns in it .
- some planes stop in the drop shaped terminal and a wide runway .
- the c shaped termial building sits on the apron which is alongside the runways .
- some planes are near a building in an airport .

Prediction:

a playground is surrounded by many green trees and buildings .

References:

- the ground track playground is next to parking lot .
- there is a parking lot near some sports fields .
- there is a parking lot near some sports fields .
- it is a large green playground surrounded by running tracks next to a parking lot full of cars .
- a playground is surrounded by several buildings, four tennis courts, a parking lot and two basketball fields .

Figure 6.3: The example set 3 of the output generated by the approach of combining summarized captions to image captions for some test images from RSICD dataset.

6.4 Results of Proposed Summarized Captioning with KL Divergence (SCKL)

The experimental set up for this model is similar to the neural captioning model, but with the addition of summarization network. Then the loss is calculated by using cross-entropy and KL Divergence loss functions. Same parameters were used for learning rate, embed and hidden dimensions as in the previously proposed method. Like in the previous method, we keep track of bleu score per epoch and terminate the training process when the bleu score stops improving after 25 epochs. The training performed on NVIDIA T4 GPU which took around two days to train. The results from this method is described below can be seen in table 6.7.

Table 6.7: Results of proposed method of summarized captioning with KL divergence (SCKL) for all three dataset.

Dataset	bleu1	bleu2	bleu3	bleu4
RSICD	0.6328	0.4600	0.3576	0.2844
UCM	0.8037	0.7257	0.6717	0.6206
Sydney	0.7798	0.6722	0.5930	0.5216

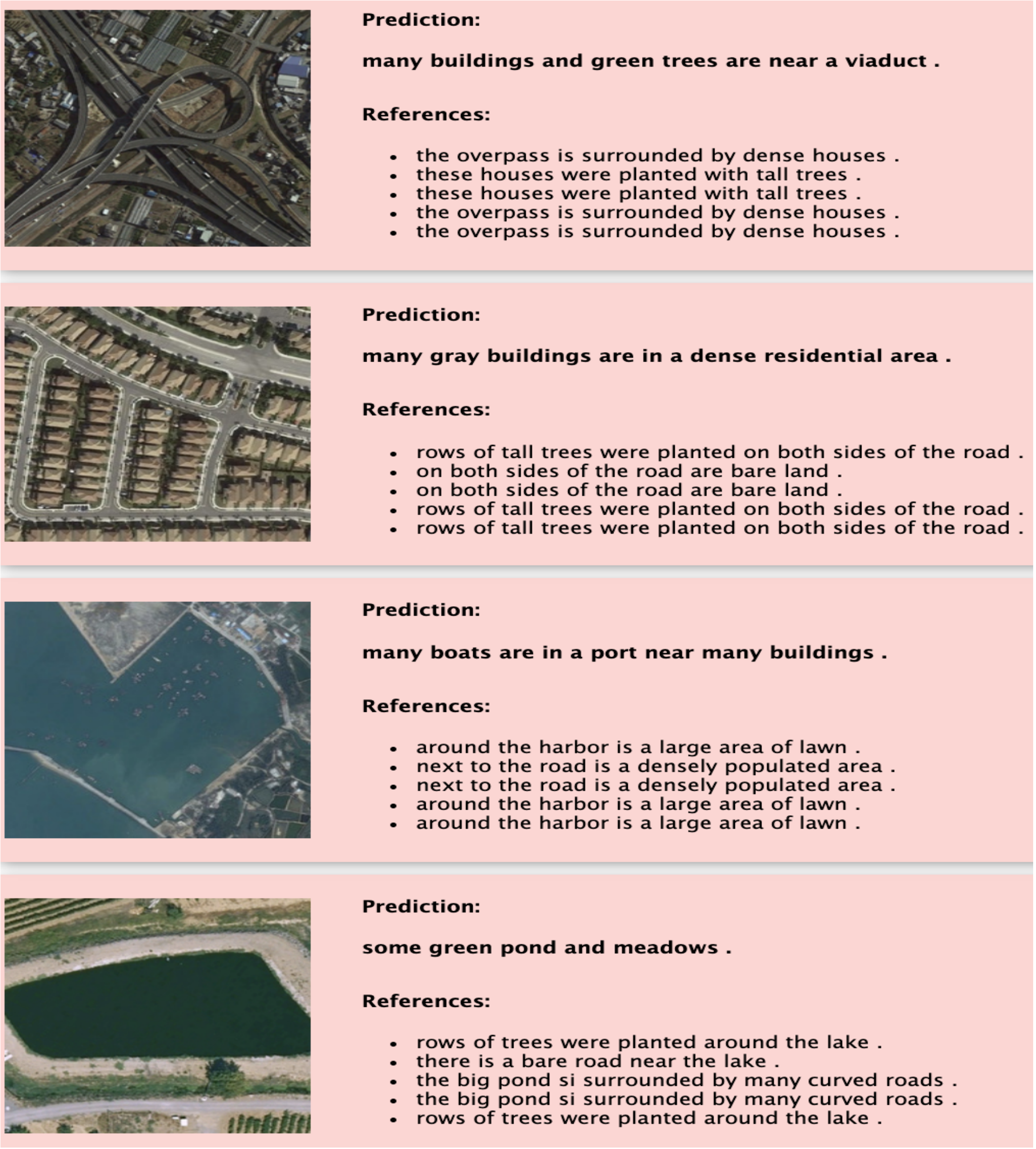
Some of the generated captions from the test set of RSICD dataset for this method is given in Figures 6.4 to 6.6. We have the same background colour coding as mentioned in the previous section.

As we can see in Figure 6.4, the generated sentences with bleu1 score less than 0.2, which are likely to be not so good machine translations, are in fact correct and meaningful translations. In this case, the model is smart enough to detect key elements in each of the example images, such as viaducts, dense residential area, port and pond. Even though they are not very similar to the ground truth references, still they are meaningful and capture correct information from the images.

The manual evaluation for this method is shown in table 6.8. We can see that, majority of the captions are correctly predicted and the number of completely incorrect captions are less. So, we can say that the model performs well for most of the cases.

Table 6.8: Results of manual evaluation for the proposed method of summarized captioning with KL divergence for RSICD dataset

Bucket	Percentage
Correct	63.04
Partially Correct	28.64
Incorrect	8.32



Prediction:

many buildings and green trees are near a viaduct .

References:

- the overpass is surrounded by dense houses .
- these houses were planted with tall trees .
- these houses were planted with tall trees .
- the overpass is surrounded by dense houses .
- the overpass is surrounded by dense houses .

Prediction:

many gray buildings are in a dense residential area .

References:

- rows of tall trees were planted on both sides of the road .
- on both sides of the road are bare land .
- on both sides of the road are bare land .
- rows of tall trees were planted on both sides of the road .
- rows of tall trees were planted on both sides of the road .

Prediction:

many boats are in a port near many buildings .

References:

- around the harbor is a large area of lawn .
- next to the road is a densely populated area .
- next to the road is a densely populated area .
- around the harbor is a large area of lawn .
- around the harbor is a large area of lawn .

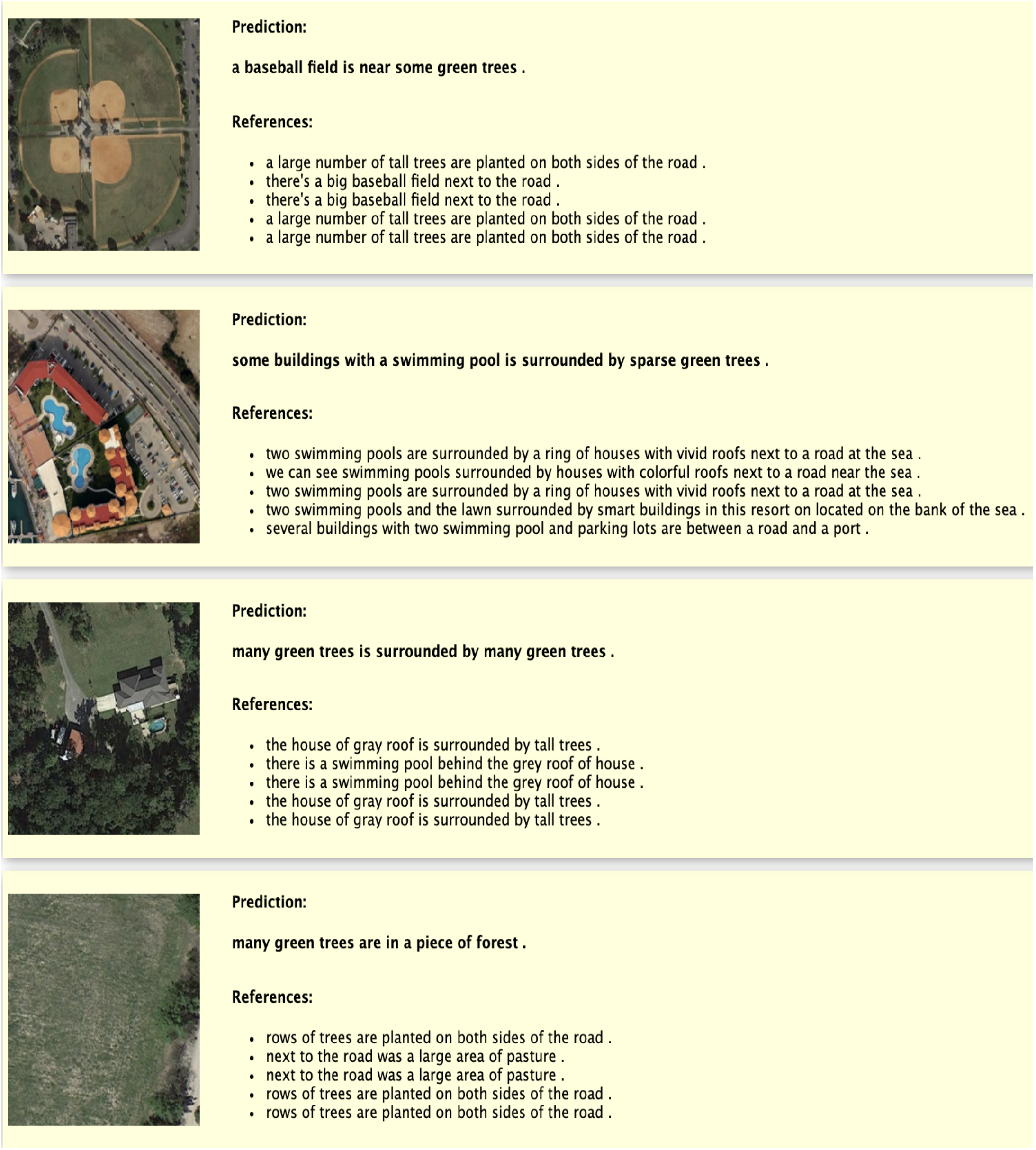
Prediction:

some green pond and meadows .

References:

- rows of trees were planted around the lake .
- there is a bare road near the lake .
- the big pond si surrounded by many curved roads .
- the big pond si surrounded by many curved roads .
- rows of trees were planted around the lake .

Figure 6.4: The example set 1 of output generated by the approach of summarized captioning with KL divergence for some of the test images from RSICD dataset.



Prediction:

a baseball field is near some green trees .

References:

- a large number of tall trees are planted on both sides of the road .
- there's a big baseball field next to the road .
- there's a big baseball field next to the road .
- a large number of tall trees are planted on both sides of the road .
- a large number of tall trees are planted on both sides of the road .

Prediction:

some buildings with a swimming pool is surrounded by sparse green trees .

References:

- two swimming pools are surrounded by a ring of houses with vivid roofs next to a road at the sea .
- we can see swimming pools surrounded by houses with colorful roofs next to a road near the sea .
- two swimming pools are surrounded by a ring of houses with vivid roofs next to a road at the sea .
- two swimming pools and the lawn surrounded by smart buildings in this resort on located on the bank of the sea .
- several buildings with two swimming pool and parking lots are between a road and a port .

Prediction:

many green trees is surrounded by many green trees .

References:

- the house of gray roof is surrounded by tall trees .
- there is a swimming pool behind the grey roof of house .
- there is a swimming pool behind the grey roof of house .
- the house of gray roof is surrounded by tall trees .
- the house of gray roof is surrounded by tall trees .

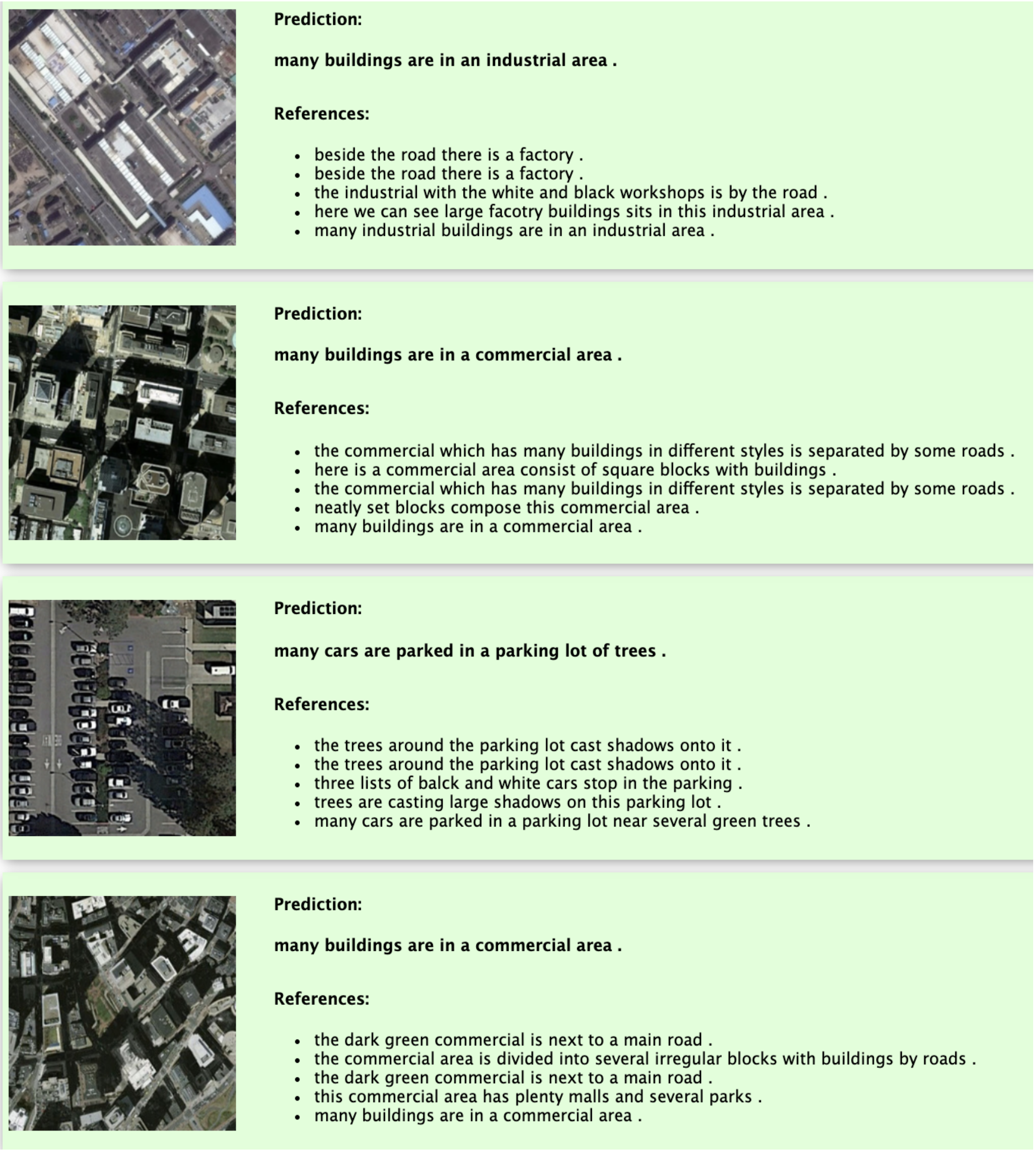
Prediction:

many green trees are in a piece of forest .

References:

- rows of trees are planted on both sides of the road .
- next to the road was a large area of pasture .
- next to the road was a large area of pasture .
- rows of trees are planted on both sides of the road .
- rows of trees are planted on both sides of the road .

Figure 6.5: The example set 2 of output generated by the approach of summarized captioning with KL divergence for some of the test images from RSICD dataset.



Prediction:

many buildings are in an industrial area .

References:

- beside the road there is a factory .
- beside the road there is a factory .
- the industrial with the white and black workshops is by the road .
- here we can see large facotry buildings sits in this industrial area .
- many industrial buildings are in an industrial area .

Prediction:

many buildings are in a commercial area .

References:

- the commercial which has many buildings in different styles is separated by some roads .
- here is a commercial area consist of square blocks with buildings .
- the commercial which has many buildings in different styles is separated by some roads .
- neatly set blocks compose this commercial area .
- many buildings are in a commercial area .

Prediction:

many cars are parked in a parking lot of trees .

References:

- the trees around the parking lot cast shadows onto it .
- the trees around the parking lot cast shadows onto it .
- three lists of balck and white cars stop in the parking .
- trees are casting large shadows on this parking lot .
- many cars are parked in a parking lot near several green trees .

Prediction:

many buildings are in a commercial area .

References:

- the dark green commercial is next to a main road .
- the commercial area is divided into several irregular blocks with buildings by roads .
- the dark green commercial is next to a main road .
- this commercial area has plenty malls and several parks .
- many buildings are in a commercial area .

Figure 6.6: The example set 3 of output generated by the approach of summarized captioning with KL divergence for some of the test images from RSICD dataset.

6.5 Results of Proposed Summarized Captioning with Attention and KL Divergence (SCAttKL)

In this method, we use a soft attention based captioning approach along with summarized captioning to train our model. Model parameters used are 0.00008 learning rate which is decreased by 20% if there are no improvements after 10 epochs. The hidden and embed dimensions are initialized to 512 each, and the size of the vocabulary used for both pre-training of the summarization model and the captioning decoder is set to 50000. Similar experimental setup with an increased vocabulary was also performed, but no significant changes were observed, and the training took much longer which is why we decided to use a vocabulary size of 50000. The training was performed on NVIDIA T4 GPU which took around 2.5 days to train. The results from this method is described below:

Table 6.9: Results of proposed method of summarized captioning with attention and KL divergence for all three datasets

Dataset	bleu1	bleu2	bleu3	bleu4
RSICD	0.6512	0.4732	0.3669	0.2916
UCM	0.8160	0.7408	0.6882	0.6374
Sydney	0.7780	0.6821	0.6031	0.5324

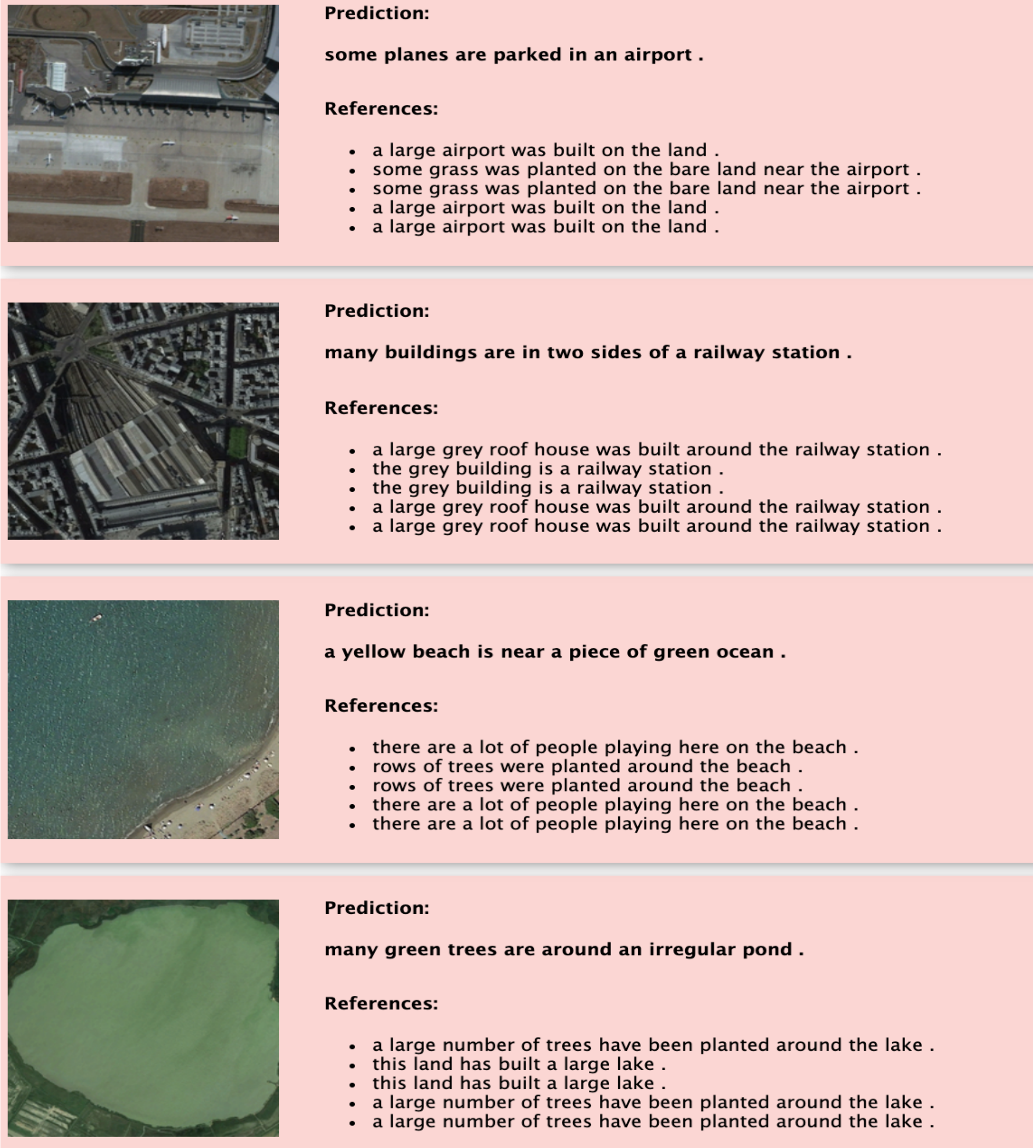
In table 6.10, we can see the results from the manual evaluation of *summarized captioning with attention and KL divergence* approach for RSICD dataset. We can see that 67.26% of the images have “correct” captions. This shows that this proposed model works very well and the captions generated for the images, describes them correctly. Secondly, the number of “Partially Correct” captions are 24.06% which is still more than “Incorrect” captions. This means, over all the model has very good performance when judged from a human point of view.

Table 6.10: Results of summarized captioning with attention and KL divergence for RSICD dataset

Bucket	Percentage
Correct	67.26
Partially Correct	24.06
Incorrect	8.68

Some of the generated captions from this method can be seen in Figures 6.7 to

6.9. As mentioned in the previous sections, the red background indicates image captions which has bleu1 score less than 0.2, the yellow background indicates captions with bleu1 score between 0.2 and 0.6. The green background indicates examples with bleu1 score more than 0.6.



Prediction:

some planes are parked in an airport .

References:

- a large airport was built on the land .
- some grass was planted on the bare land near the airport .
- some grass was planted on the bare land near the airport .
- a large airport was built on the land .
- a large airport was built on the land .

Prediction:

many buildings are in two sides of a railway station .

References:

- a large grey roof house was built around the railway station .
- the grey building is a railway station .
- the grey building is a railway station .
- a large grey roof house was built around the railway station .
- a large grey roof house was built around the railway station .

Prediction:

a yellow beach is near a piece of green ocean .

References:

- there are a lot of people playing here on the beach .
- rows of trees were planted around the beach .
- rows of trees were planted around the beach .
- there are a lot of people playing here on the beach .
- there are a lot of people playing here on the beach .

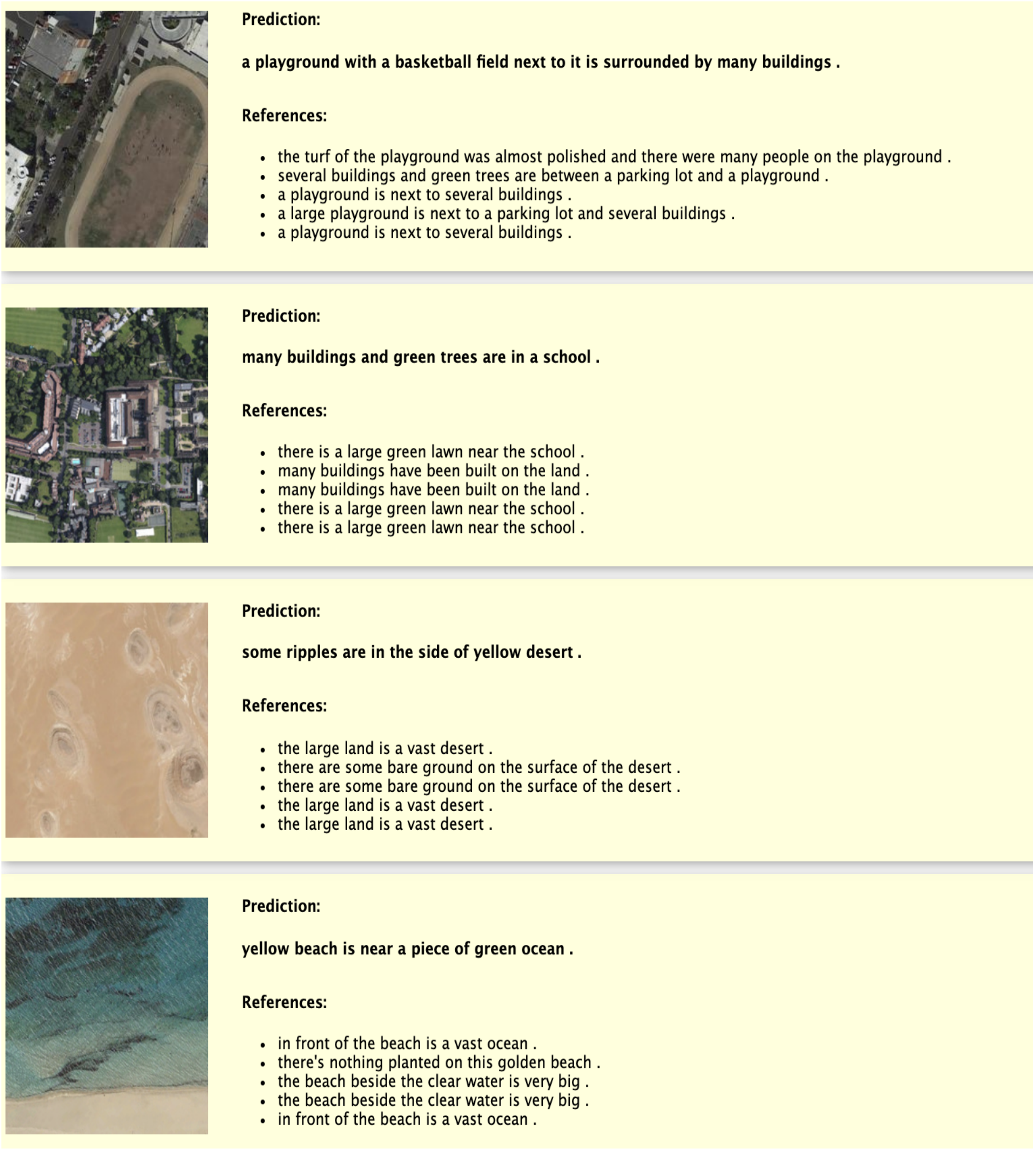
Prediction:

many green trees are around an irregular pond .

References:

- a large number of trees have been planted around the lake .
- this land has built a large lake .
- this land has built a large lake .
- a large number of trees have been planted around the lake .
- a large number of trees have been planted around the lake .

Figure 6.7: The example set 1 of output generated by the approach of summarized captioning with attention and KL divergence for some of the test images from RSICD dataset.



Prediction:

a playground with a basketball field next to it is surrounded by many buildings .

References:

- the turf of the playground was almost polished and there were many people on the playground .
- several buildings and green trees are between a parking lot and a playground .
- a playground is next to several buildings .
- a large playground is next to a parking lot and several buildings .
- a playground is next to several buildings .

Prediction:

many buildings and green trees are in a school .

References:

- there is a large green lawn near the school .
- many buildings have been built on the land .
- many buildings have been built on the land .
- there is a large green lawn near the school .
- there is a large green lawn near the school .

Prediction:

some ripples are in the side of yellow desert .

References:

- the large land is a vast desert .
- there are some bare ground on the surface of the desert .
- there are some bare ground on the surface of the desert .
- the large land is a vast desert .
- the large land is a vast desert .

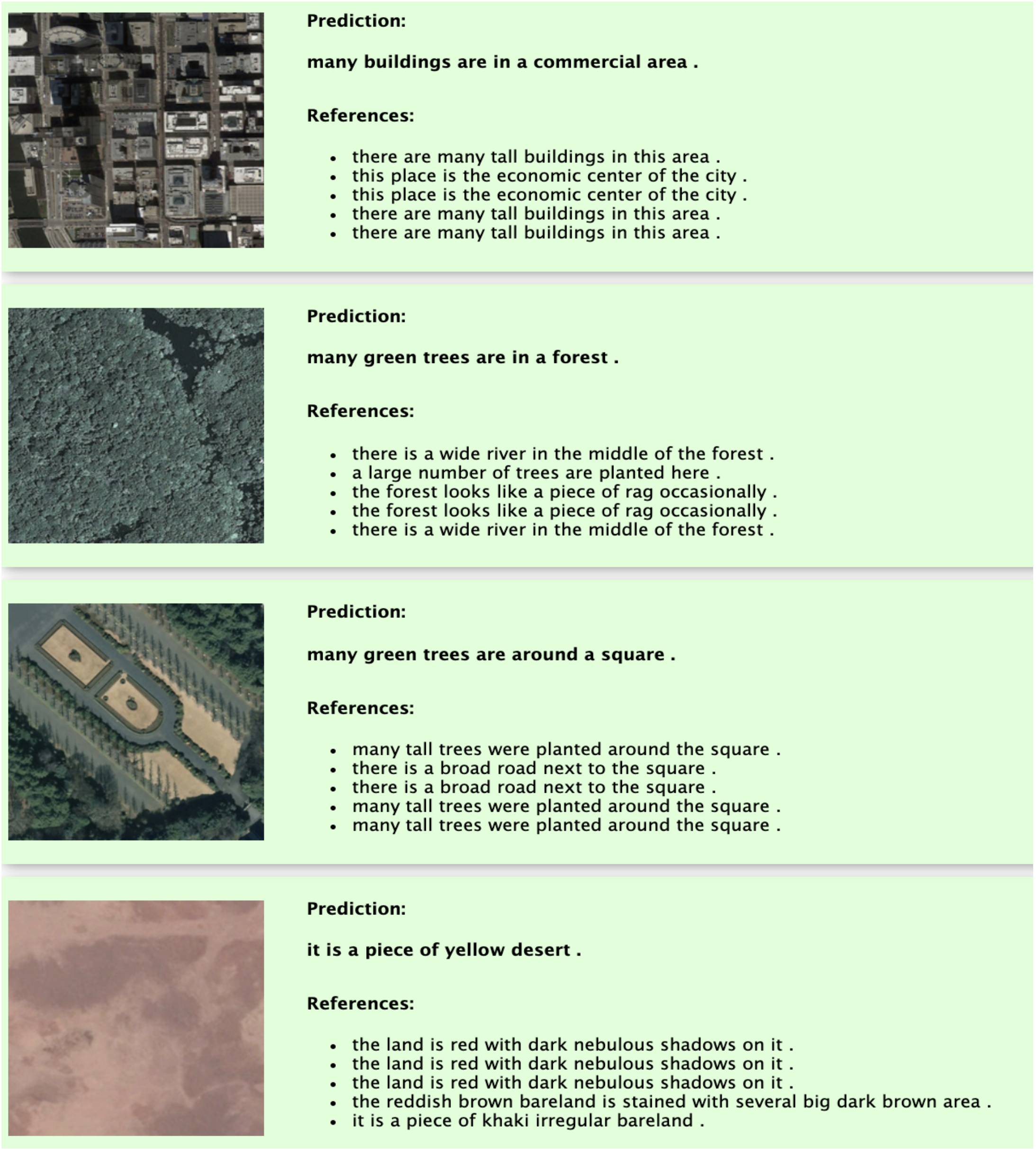
Prediction:

yellow beach is near a piece of green ocean .

References:

- in front of the beach is a vast ocean .
- there's nothing planted on this golden beach .
- the beach beside the clear water is very big .
- the beach beside the clear water is very big .
- in front of the beach is a vast ocean .

Figure 6.8: The example set 2 of output generated by the approach of summarized captioning with attention and KL divergence for some of the test images from RSICD dataset.



Prediction:

many buildings are in a commercial area .

References:

- there are many tall buildings in this area .
- this place is the economic center of the city .
- this place is the economic center of the city .
- there are many tall buildings in this area .
- there are many tall buildings in this area .

Prediction:

many green trees are in a forest .

References:

- there is a wide river in the middle of the forest .
- a large number of trees are planted here .
- the forest looks like a piece of rag occasionally .
- the forest looks like a piece of rag occasionally .
- there is a wide river in the middle of the forest .

Prediction:

many green trees are around a square .

References:

- many tall trees were planted around the square .
- there is a broad road next to the square .
- there is a broad road next to the square .
- many tall trees were planted around the square .
- many tall trees were planted around the square .

Prediction:

it is a piece of yellow desert .

References:

- the land is red with dark nebulous shadows on it .
- the land is red with dark nebulous shadows on it .
- the land is red with dark nebulous shadows on it .
- the reddish brown bareland is stained with several big dark brown area .
- it is a piece of khaki irregular bareland .

Figure 6.9: The example set 3 of output generated by the approach of summarized captioning with attention and KL divergence for some of the test images from RSICD dataset.

6.6 Comparison among the Proposed Methods

Table 6.11: Combined result of all the proposed methods for all three dataset

	Dataset	bleu1	bleu2	bleu3	bleu4
CSIC	RSICD	0.6327	0.4533	0.3501	0.2774
SCKL	RSICD	0.6328	0.4600	0.3576	**0.2844**
SCAttKL	RSICD	**0.6512**	**0.4732**	**0.3669**	0.2816
CSIC	UCM	0.7949	0.7092	0.6498	0.5923
SCKL	UCM	0.8037	0.7257	0.6717	0.6206
SCAttKL	UCM	**0.8160**	**0.7408**	**0.6882**	**0.6374**
CSIC	Sydney	**0.7832**	**0.6865**	**0.6152**	**0.5458**
SCKL	Sydney	0.7798	0.6722	0.5930	0.5216
SCAttKL	Sydney	0.7780	0.6821	0.6031	0.5324

In table 6.9 we have combined results of all the proposed experiments for all the three datasets. We can see that for RSICD and UCM dataset, the proposed method of Summarized captioning with KL divergence with attention has the best result. For RSICD the bleu4 score is better for the method of Summarized captioning with KL divergence without attention, but it is not significant and almost same as the method of Summarized captioning with KL divergence with attention.

Although, for Sydney dataset, we have better bleu scores from the method of combining summarized outputs to image captions. This is because the Sydney dataset is comparatively smaller than the other two datasets. Smaller datasets are not very efficient for training deep neural networks as they don't generalize easily. So the simpler model provides better results than the more complex ones.

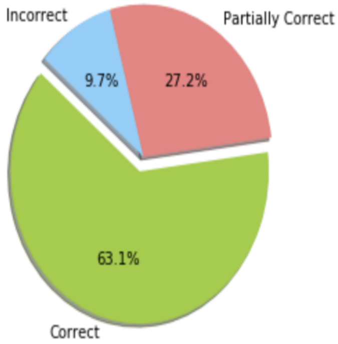


Figure 6.10: Result of manual evaluation for the method of combining summarized captioning with image captioning for RSICD dataset.

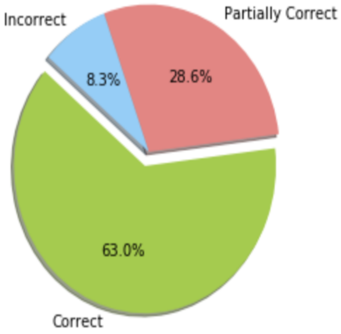


Figure 6.11: Result of manual evaluation for the method of summarized captioning with KL divergence for RSICD dataset.

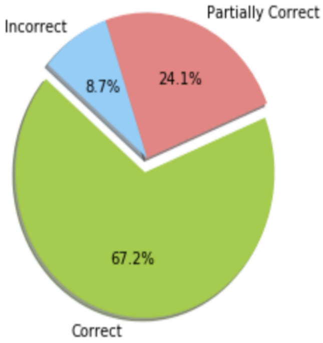


Figure 6.12: Result of manual evaluation for the method of summarized captioning with attention and KL divergence for RSICD dataset.

In Figure 6.10, 6.11 and 6.12 we can see the results of the manual evaluation in form of pie charts. We see that majority of captions in all of the proposed approaches are correctly described by the respective models, which is more than 60% of correct predictions for all three proposed method. The method of "summarized captioning with attention and KL divergence" has most correct predictions, which is more than 67% It is also interesting to note that the number of *incorrect* captions are less than 10% for all the proposed models. This ensures that all the proposed models are very reliable and can be used for captioning images in large data archives.

7 Conclusion and Discussion

In this thesis we introduces 3 novel remote sensing image captioning techniques for high resolution remote sensing images. All our proposed methods uses a summarization model in addition to the captioning framework which helps generate the captions for RS images in a concised and generalized way. All the proposed methods uses deep learning techniques.

In the proposed method of combining summarized captions with images captions we use a ResNet152 pre-trained model for image encoding and an LSTM based network for modeling caption generation. We then use a pre-trained summarization model trainined on Gigaword dataset to generate a summarized caption from the 5 ground truth captions per image. Finally we combine the output from the decoder of the captioning LSTM and the output from the decoder of summarization LSTM which both produces a probability score vector of all the resulting words in the generated caption. We use this combined score vector to calculate the training loss in order for our model to give consideration towards the summarized captions along with the ground truth captions.

In the second proposed method, also use a ResNet152 pre-trained model as the encoder of captioning framework and LSTM as decoder. In this model also we use a pre-trained summarization network, but instead of combining the output score vectors of the captioning and the summarization models, we try to minimize the distance between the 2 generated distributions with KL divergence loss. We the add this to the CrossEntropyLoss which minimizes the log likelihood of the prediction of correct word with respect to the gound truth.

In the third proposed method uses a soft attention mechanism to give more focus on parts of the image and language encoding which the model thinks is important. This is done by generating a weighted matrix which is computed by an attention network. First the encoder generates image encodings, which is then transformed to the initial hidden and cell state in order to map them to the LSTM(decoder). In the decoder, at each step, the image encodings and the previous hidden states are used to generate weights for each pixel from the

Attention network. The decoder tries to generate the next word, after seeing the previously generated word and the weighted average of the encodings. We also have a pre-trained summarization model in which the decoder of the model tries to generate summarized caption from 5 combined captions one after the other. The loss calculation in this case, includes a CrossEntropyLoss, a doubly stochastic regularization and a KL divergence loss to minimize the distance between the generated captions and the summarized captions.

Experiments were performed on 3 different RS image captioning dataset shows that the models are have better performance in comparison to the state of art RS image captioning methods. Our manual evaluation shows that the generated captions, in majority of the cases are very precise and accurate without any errors. The purpose of adding a summarization network in our proposed approaches was deal with some of the drawbacks of the dataset. The datasets consists of 5 captions per image but in majority of the cases all the 5 caption were same, which could lead to overfitting of the model. We didn't want to create another dataset by removing such repetitions, which is why we added the summarization network to overcome this problem. The implementation of attention mechanism along with a much bigger vocabulary also ensures more reliable outcome in case a our model receives images which are not similar to our training examples. By doing so, we enrich our model to be more generic. This means that, even being a supervised model our model doesn't suffer from bad result in case of new data.

The results of our proposed models were evaluated using BLEU evaluation metric and also using a manual evaluation method. The results of the previous existing methods and our proposed methods are almost similar but we are using much bigger vocabulary than that of the original datasets which the existing automatic evaluation metrics is unable to capture. Also, the BLEU scores of our proposed models are significantly better than the results from the papers "Exploring Models and Data for Remote Sensing Image Caption Generation" [23] and "Semantic Descriptions of High-Resolution Remote Sensing Images" [38]. Although, the results of the paper "Description Generation for Remote Sensing Images Using Attribute Attention Mechanism" [45] are better than our results. For this reason we performed a manual evaluation by counting the number of *correct*, *partially correct* and *incorrect* captions. We see that in all of the proposed methods, majority of the images are predicted with correct captions. For all the proposed methods, the number of correct predictions are more than 60%. Even the *partially incorrect* captions capture some information about the image if not all. This makes our proposed model reliable for captioning remote sensing images.

In future we would like to create a test dataset with a manually annotated summarized caption, which could be used to compare the results of the existing and the proposed methods using automatic evaluation metrics. This would also help us see the affect of summarized caption with respect to normal captioning approaches and how well each method generalizes.

This work could help annotating big data archives with captions which would be very useful in analyzing it in a much faster way. This is also applicable for natural image captioning, in order to generate a summarized caption.

Bibliography

[1] J. Aroma R & K. Raimond. “A review on availability of remote sensing data”. In: *2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)*. 2015, pp. 150–155.

[2] Jimmy Ba, Volodymyr Mnih & Koray Kavukcuoglu. “Multiple Object Recognition with Visual Attention”. In: (Dec. 2014).

[3] Dzmitry Bahdanau, Kyunghyun Cho & Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *CoRR* abs/1409.0473 (2015).

[4] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Oct. 2014, pp. 1724–1734.

[5] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *CoRR* abs/1406.1078 (2014).

[6] Bo Dai et al. “Towards Diverse and Natural Image Descriptions via a Conditional GAN”. In: *CoRR* abs/1703.06029 (2017).

[7] N. Dalal & B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1.

[8] M. Denil et al. “Learning Where to Attend with Deep Architectures for Image Tracking”. In: *Neural Computation* 24.8 (2012), pp. 2151–2184.

[9] A. Géron. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2017.

[10] I. Goodfellow, Y. Bengio & A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. 2016.

[11] David Graff et al. *English gigaword*. Linguistic Data Consortium, Philadelphia., 2003.

[12] Alex Graves. “Generating Sequences With Recurrent Neural Networks”. In: *CoRR* abs/1308.0850 (2013).

[13] Sepp Hochreiter & Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780.

[14] Md. Zakir Hossain et al. “A Comprehensive Survey of Deep Learning for Image Captioning.” In: *ACM Comput. Surv.* 51.6 (Feb. 2019), 118:1–118:36. ISSN: 0360-0300.

[15] Sutskever Ilya, Vinyals Oriol & V. Le Quoc. “Sequence to Sequence Learning with Neural Networks”. In: *CoRR* abs/1409.3215 (2014).

[16] ImageNet. Last Accessed on 20/04/2019. URL: `https://http://www.image-net.org/`.

[17] He Kaiming et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015).

[18] Diederik Kingma & Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).

[19] Ryan Kiros, Ruslan Salakhutdinov & Rich Zemel. “Multimodal Neural Language Models”. In: *Proceedings of the 31st International Conference on Machine Learning.* Ed. by Eric P. Xing & Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. 2014, pp. 595–603.

[20] G. Klein et al. “OpenNMT: Open-Source Toolkit for Neural Machine Translation”. In: *ArXiv e-prints* (2017).

[21] Hugo Larochelle & Geoffrey E Hinton. “Learning to combine foveal glimpses with a third-order Boltzmann machine”. In: *Advances in Neural Information Processing Systems 23.* Ed. by J. D. Lafferty et al. 2010, pp. 1243–1251.

[22] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[23] Xiaoqiang Lu et al. “Exploring Models and Data for Remote Sensing Image Caption Generation”. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.4 (2017), pp. 2183–2195.

[24] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze. *Introduction to Information Retrieval.* 2008.

[25] Junhua Mao et al. “Explain Images with Multimodal Recurrent Neural Networks”. In: *CoRR* abs/1410.1090 (2014).

[26] Volodymyr Mnih et al. “Recurrent Models of Visual Attention”. In: *CoRR* abs/1406.6247 (2014).

[27] Loris Nanni, Stefano Ghidoni & Sheryl Brahnam. “Handcrafted vs Non-Handcrafted Features for computer vision classification”. In: *Pattern Recognition* 71 (June 2017).

[28] Courtney Napoles, Matthew Gormley & Benjamin Van Durme. “Annotated Gigaword”. In: 2012, pp. 95–100.

[29] Keiron O’Shea & Ryan Nash. “An Introduction to Convolutional Neural Networks”. In: *ArXiv e-prints* (Nov. 2015).

[30] Kishore Papineni et al. “BLEU: A Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.* ACL ’02. 2002, pp. 311–318.

[31] B. Qu et al. “Deep semantic understanding of high resolution remote sensing image”. In: *2016 International Conference on Computer, Information and Telecommunication Systems (CITS).* 2016, pp. 1–5.

[32] Abigail See, Peter J. Liu & Christopher D. Manning. “Get To The Point: Summarization with Pointer-Generator Networks”. In: *CoRR* abs/1704.04368 (2017).

[33] Z. Shi & Z. Zou. “Can a Machine Generate Humanlike Language Descriptions for a Remote Sensing Image?” In: *IEEE Transactions on Geoscience and Remote Sensing* 55.6 (2017), pp. 3623–3634.

[34] K. Simonyan & A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations.* 2015.

[35] Ilya Sutskever, James Martens & Geoffrey E. Hinton. “Generating Text with Recurrent Neural Networks”. In: Jan. 2011, pp. 1017–1024.

[36] Yichuan Tang, Nitish Srivastava & Ruslan Salakhutdinov. “Learning generative models with visual attention”. In: *CoRR* abs/1312.6110 (2013).

[37] O. Vinyals et al. “Show and tell: A neural image caption generator”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2015, pp. 3156–3164.

[38] B. Wang et al. “Semantic Descriptions of High-Resolution Remote Sensing Images”. In: *IEEE Geoscience and Remote Sensing Letters* (2019), pp. 1–5.

[39] Ronald J. Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine Learning* 8.3 (1992), pp. 229–256.

[40] Kelvin Xu et al. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *CoRR* abs/1502.03044 (2015).

[41] Yi Yang & Shawn Newsam. “Bag-of-visual-words and Spatial Extensions for Land-use Classification”. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS ’10. 2010, pp. 270–279.

[42] Zhilin Yang et al. “Encode, Review, and Decode: Reviewer Module for Caption Generation”. In: *CoRR* abs/1605.07912 (2016).

[43] Wojciech Zaremba, Ilya Sutskever & Oriol Vinyals. “Recurrent Neural Network Regularization”. In: *CoRR* abs/1409.2329 (2014).

[44] F. Zhang, B. Du & L. Zhang. “Saliency-Guided Unsupervised Feature Learning for Scene Classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.4 (2015), pp. 2175–2184.

[45] Xiangrong Zhang et al. “Description Generation for Remote Sensing Images Using Attribute Attention Mechanism”. In: *Remote Sensing* 11.6 (2019).