A project report on

AUTOMATED IMAGE CAPTIONING SYSTEM

Submitted in partial fulfillment for the award of the degree of

B.Tech.

in

Information Technology

By

Saurabh Mathur (14BIT0180)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING (SITE)

APRIL 2018

A project report on

AUTOMATED IMAGE CAPTIONING SYSTEM

Submitted in partial fulfillment for the award of the degree of

B.Tech.

in

Information Technology

By

Saurabh Mathur (14BIT0180)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING (SITE)

APRIL 2018

DECLARATION

I hereby declare that the thesis entitled "Automated Image Captioning System" submitted by me, for the award of the degree of Specify the name of the degree VIT is a record of bona fide work carried out by me under the supervision of Prof. Daphne Lopez.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: **Signature of the Candidate**

CERTIFICATE

This is to certify that the thesis entitled "Automated Image Captioning System" submitted by Saurabh Mathur (14BIT0180) School of Information Technology & Engineering VIT, for the award of the degree of BTech in Information Technology is a record of bonafide work carried out by him/her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfills the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of the HOD**

Internal Examiner **External Examiner**

ABSTRACT

Many forms of media frequently lack alternative text for images. This lack of image captions hampers the accessibility of their content. Hiring people to write captions for those pictures is often prohibitively expensive. The use of an automated system to write the captions would be a viable alternative. The objective of the project is to develop such a system which can generate descriptive captions for a given image. Advances in Computer Vision have led to the development of Artificial Neural Network(ANN) based feature extractors like the Convolutional Neural Network (CNN). These feature extractors can convert digital images into rich high-level representations using a statistical model previously learned from labeled data. Additionally, ANNs can be modified to model sequences of symbols like sentences. One such type of ANN is the Recurrent Neural Network (RNN).The proposed implementation of the image captioning system would feed the features extracted by a CNN to an RNN. The RNN would generate a caption based on the previously obtained image features.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Daphne Lopez Professor, School of Information Technology, Vellore Institute of Technology, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Information Technology.

I would like to express my gratitude to Dr. G. Vishwanathan, Mr. G.V. Selvam, Dr. Anand A Samuel, Dr. S. Narayanan, and Dr. Ashwani Kumar Cherukuri, School of Information Technology, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Dinakaran M., Associate Professor & HOD, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date: **Name of the student**

CONTENTS

CONTENTS iii

LIST OF FIGURES v

LIST OF ACRONYMS vii

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND..... 1
1.2 PROJECT MOTIVATION ..…..... 4
1.3 OBJECTIVE..... 5
1.4 ORGANIZATION OF REPORT.....6

CHAPTER 2

OVERVIEW AND PLANNING

2.1 PROPOSED SYSTEM 8
2.2 CHALLENGES .....… 9
2.3 ARCHITECTURE.....….9
2.4 HARDWARE REQUIREMENTS.....…20
2.5 SOFTWARE REQUIREMENTS.....….20

CHAPTER 3

LITERATURE SURVEY

3.1 LITERATURE SURVEY..... 21

CHAPTER 4

SYSTEM DESIGN

4.1 HIGH LEVEL DESIGN..... 25
4.2 LOW LEVEL DESIGN..... 27

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 DATASET..... 29
5.2 MODEL ARCHITECTURES…….....33

CHAPTER 6

RESULT AND DISCUSSION

6.1 OUTPUT 37

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 CONCLUSION.....47
7.2 SCOPE FOR FUTURE WORK..... .47

LIST OF FIGURES

Figure 1.1 : The world's first digital image…….....….....…1
Figure 1.2 : Image alt-text tags on top 10 most visited websites.....…..... 5
Figure 2.1 : An example of the structure of the CNNs – LeNet-5..... 10
Figure 2.2 : An example of the convolutional layer..... 11
Figure 2.4 : A recurrent neural network unrolled in time……..... 13
Figure 2.5: The Vanishing gradient problem in RNNs……..... 16
Figure 2.6: LSTM memory cell……..... 17
Figure 2.7: Gradient flow in LSTM……..... 19
Figure 4.1: High level design architecture……..... 25
Figure 4.2: Image pre-processor flowchart……..... 27
Figure 5.1: Details of images in the MS COCO dataset…..... 30
Figure 5.2: Details of captions in the MS COCO dataset…..... 30
Figure 5.3: Joined image and caption details in MS COCO dataset……..... 31
Figure 5.4: Example 1 of an image and its captions in the MS COCO dataset..... 32
Figure 5.5: Example 2 of an image and its captions in the MS COCO dataset..... 32
Figure 5.6: Inception-V3 model architecture……..... 34
Figure 5.7: Caption generator model architecture……..... 35
Figure 5.8: Beam search (red) as compared to greedy search (green)……..... 36
Figure 6.1: Output example 1……...……37
Figure 6.2: Output example 2……...……38
Figure 6.3: Output example 3……...……39

Figure 6.4: Output example 4……...……40
Figure 6.5: Output example 5……...……41
Figure 6.6: Output example 6……...……42
Figure 6.7: Output example 7……...……43
Figure 6.8: Output example 8……...……44
Figure 6.9: Output example 9……...……45
Figure 6.10: Output example 10…..……..……...……46

LIST OF ACRONYMS

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LSTM Long Short-Term Memory Network

Chapter 1

INTRODUCTION

1.1 BACKGROUND

1.1.1 DIGITAL IMAGES

An image is a representation of the visual perception of an object or a person. An image may be a sculpture, a painting, a photograph or even a mental image in a person's mind. Images have been used by humans as early as the stone age. Images do not require prior knowlege to understand. The ubiqitousness of images made it necessary to develop ways to represent images in formats that can be used with computers.

In 1957, Russell Kirsch used a drum scanner to create the world's first digital image. The scanner converted the image to a rectangular block of bits. The computer connected to the scanner stored a high bit, or a 1 for places where there was light and a low bit, or a 0 for places that did not have light. Each of the bits here were called a picture element or, a pixel.

Figure 1.1 : The world’s first digital image

Concretely, a two dimensional digital image can be represented as a two dimensional array of pixels. Modern digital images are of 5 types:

1) Black and White Images, where each pixel can have two values - high or low. This is the type of image that Russell Kirsch created. Each pixel can be stored a single bit.

2) Grayscale Images, where each pixel can have a range of values showing how light or dark that pixel is. Generally, grayscale images have pixels sized one byte each.

3) RGB Images, where the intensity values of the three primary colors Red, Green and, Blue are used to represent a pixel. Each primary color component of each pixel can be stored in one byte each. Thus, RGB images can have 16,777,216 different colors.

4) 4. RGBA Images, which are similar to RGB images but have an additional alpha component.

1.1.1.2 ADVANTAGES OF DIGITAL IMAGES

1) The processing of digital images is faster and more cost-effective.
2) Copying a digital image is easy, and the quality of the image can be preserved.
3) Images can be used in a variety of media by changing the image format and resolution.
4) It is more environmentally friendly to use digital images. The alternatives like film or print require the use of synthetic chemicals which cause environmental pollution.

1.1.2 CHALLENGES OF IMAGE BASED CONTENT

In 1989 Sir Tim Berners-Lee invented the World Wide Web. The rapid spread of the world wide web has led to a lot of data being transmitted everyday over the internet. This has brought about various challenges like the security of the transmitted data and the accessibility of the multimedia content. The enormous popularity of digital images and their difference in representations from text and images add another dimension to the problem.

Over the years, many challenges regarding the use of digital images for multimedia content have been identified. Some of the challenges are:

1) Preventing misuse of copyrighted images.
2) Preserving privacy of users exchanging data.
3) Disguising the nature and content of data sent.
4) Universal accessibility of image content.

1.1.3 OVERVIEW OF MACHINE LEARNING SYSTEMS

Machine Learning is the application of statistical methods to make predictions about the future using data about the past. It is a statistical approach at building AI Agents which works by finding patterns in data and using the same patterns to make predictions. Machine learning tasks can be categorised into three types:

1) Supervised Learning
2) Unsupervised Learning

Supervised Learning is the most common form of machine learning. It consists of two phases - training and testing.

1) In the training phase, a set of pre-labelled data is used to establish a mathematical relation between the data and labels.
2) This mathematical relation, also called a model is used in the testing phase to predict labels for unlabelled data.

For example, consider the problem of classifying an image as a "Cat" or "Not a Cat". In the training phase we would use a set of images which are labelled as either "Cat" or "Not a Cat" to build a model between the pixels of the image and whether it is a Cat or not. In the testing phase, the model would be used on new unlabelled images to predict whether they contain Cats or not. Examples of supervised machine learning methods include linear regression, decision trees and artificial neural networks.

In Unsupervised Learning the training data does not include explicit labels or targets and the system finds patterns from data without human supervision. For the cat example, an unsupervised learning system will not require pre-labelled images of cats and not-cats. Commonly used unsupervised learning methods include k-means clustering and the apriori algorithm.

However, conventional machine learning do not work directly on raw data. Instead, engineers and domain experts select and design features which are values derived from the raw data that are strongly related to what we want to predict. A feature can be thought of as an alternate representation of the data. In the cat example, the (Red, Green, Blue) color value of the center pixel could be used as a feature to determine if the picture is of a Cat.

Deep Neural Networks are a subset of machine learning methods that attempt tackle the problem of feature selection. These variants of Neural Networks, have many hidden layers that allow them to "learn" a better representation of the raw data. Thus, they can work directly with raw data eliminating the need for feature selection. Given enough hidden layers Deep Neural Networks can model any function and are hence called Universal Approximators.

1.2 PROJECT MOTIVATION

According to the World Health Organization, over a billion people, about 15% of the world's population, have some form of disability and 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. Many of these people use screen magnifiers, and some use screen readers, which is software that reads digital text aloud, to access multimedia content.

As of June 2017, 51% of the world's population has internet access. This rapid penetration of internet usage coupled with the low cost and convenience has incentivized various service providers and content creators to shift their

businesses to the web. However, this poses various challenges with respect to providing the same level of service to people with disabilities, especially those visually impaired. One of the major challenges is making image-rich content on the web universally accessible since popular content based websites fail to provide alternative text or alt-text for images.

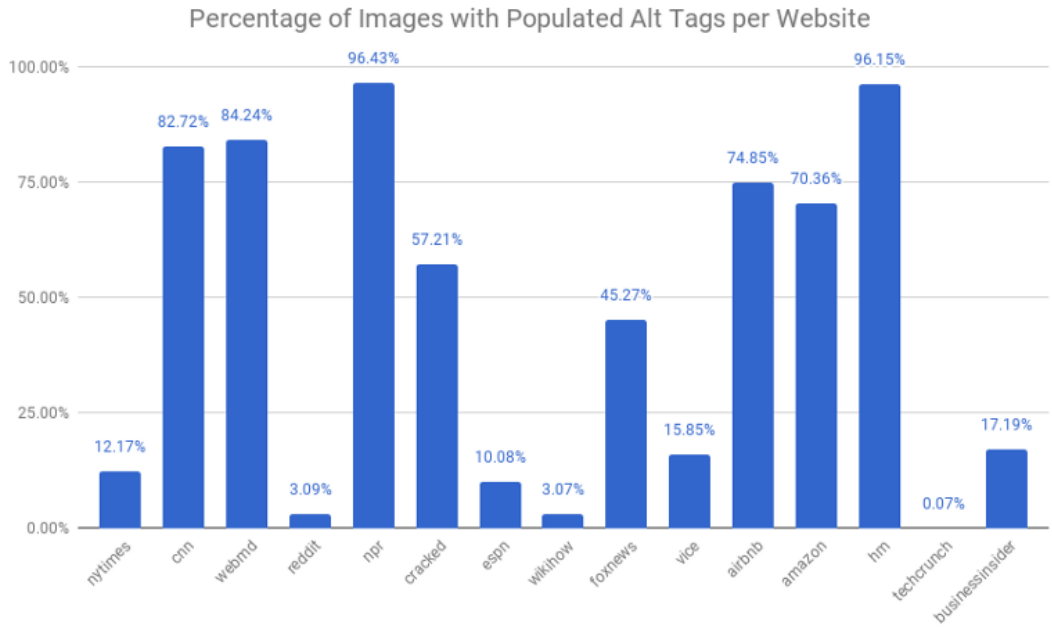


Figure 1.2 : Image alt-text tags on top 10 most visited websites.

1.3 OBJECTIVE

The prime objective of the project is to develop a solution that addresses the accessibility problem inherent in image-rich content. Images are inherently inaccessible to people who are unable to see them. This not only includes the visually impaired but also the people without a high speed internet connection who are unable to download the images. Thus, the aim of this project is to develop a system that automatically inserts a sensible alt-text description for images which lack it.

Images used in multimedia can be categorized into the following three types.

1) The image conveys simple information (e.g, a photograph, icon, or logo)

2) The image conveys complex information (e.g., a chart or graph)
3) The image is purely decorative, not informative

Since it does not add any new information to the content, type 3 can be safely ignored with regards to universal accessibility. Further, type 2 images cannot be described effectively using alt-text which is chiefly intended for concise descriptions. Type 1 images, on the other hand can be described in a short sentence or two. This makes them ideal for being captioned by an automated system. This project deals with the problem of generating captions automatically and without human supervision, for images that convey simple information.

The task of image captioning involves analyzing the content of an image and generating a description in the form of text that characterizes some salient aspects of the input image. This task is a subset of combined language-vision tasks that simulate a lot of everyday tasks requiring comprehension of both visual and textual information. For this task, not only are robust scene understanding and elaborate natural language generation algorithms necessary, they must also be compatible with each other. Further, deciding which part of the image is salient is also application-specific.

However, for the purpose of this project, the aim is to generate a grammatically correct description of a given image. Additionally, since data is available only for captions in the English language, the project deals with generating captions in English only.

1.4 ORGANIZATION OF REPORT

The report is organized in four phases, which are:

1. Planning
2. Design
3. Implementation and Testing
4. Results

1.4.1 PLANNING

The planning phase comprises of the challenges faced and the initial overview of the project. It also contains the assumptions about the end user and their workflow, so that the program will be familiar and intuitive. It also consists of the hardware and software requirements and the description of each component and the architecture specification.

1.4.2 DESIGN

The design phase deals with the high level and low level design. It is an important phase as it deals with how the project will be structured and will decide what dependency various parts will have on each other. Finally test cases are generated to ensure smooth functioning of the system and to keep in line the actual output with the expected output.

1.4.3 IMPLEMENTATION AND TESTING

The actual implementation and the testing of the implementation itself is done in this phase. It consists of writing the code, training the model and performing unit and integration testing.

1.4.4 RESULTS

In this phase the test results are checked against the expected output and it is made sure that there is minimum to no deviations. Finally the results are tabulated. Any deviation in the output is reported. This consists of the sample outputs against various test inputs.

Chapter 2

OVERVIEW AND PLANNING

2.1 PROPOSED SYSTEM

The system developed is an automated caption generation system which consists of many pre-processing components, a caption generation model and, a caption decoder. The image reader that reads the image from the camera or from disk. This image is converted from the storage format like JPEG or PNG to an array representation. An RGB image of dimensions 512x512 is converted to an array of shape 512x512x3.

The image pre-processor normalizes the array and resizes the image to the fixed dimensions of 128x128. Nearest Neighbor interpolation is used to fill-in missing values. The normalized image is passed to the feature extractor. The feature extractor analyses the image and finds salient features in it. These extracted features can be thought of as a compressed gist of the information contained in the image. This gist of the image is passed to the caption generator.

The caption generator generates probability score values for words at multiple consecutive time-steps. These score values are used by the caption decoder to sample a caption and translate it into words. The words are joined and outputted.
The system can be installed in screen reader and browsers as an extension. When activated, it reads the image in question and generates a descriptive caption for the image.

The feature extractor is a convolutional neural network pre-trained on a large-scale object classification task. This makes it ideal for extracting shapes and features from images. The caption generator forms the core of the model and it is trained using a large image-caption dataset on a machine with a high powered GPU.

2.2 CHALLENGES

The major challenge is to reduce the cost of training and operation of the system. While extensive research has been done in making image captioning systems more accurate, very little work has been done to make the systems feasible enough to be deployed on small machines like a low-end computer or a mobile phone.

Another challenge is to ensure a large enough sample space for the training data so that the system works in the real world. This means finding an image-caption dataset which is not only large but also does not contain a lot of noise and contains images from real-world situations.

2.3 ARCHITECTURE

2.3.1 BACKGROUND

To order to fully appreciate and understand the architecture and functioning of the system it is necessary to have an understanding of:

- Convolutional Neural Networks
- Recurrent Neural Networks

These will be covered in this chapter explaining each concept used in detail with appropriate figures.

2.3.2 CONVOLUTIONAL NEURAL NETWORKS

The convolutional neural network is a special type of feed forward neural network. In the traditional neural network, the neurons of every layer are one-dimensional. The convolutional neural network is designed to process multi-dimensional data. The CNN has two important concepts - local connections and parameters sharing. These concepts reduce the amount of parameters which should be trained.

There are three main types of layers to build CNN architectures:

1) the convolutional layer,

2) the pooling layer, and
3) the fully-connected layer.

The fully-connected layer is just like the regular neural networks. The convolutional layer can be considered as performing the convolution operation many times on the previous layer. The pooling layer can be though as downsampling by the maximum of each block of the previous layer. We stack these three layers to construct the full CNN architecture.

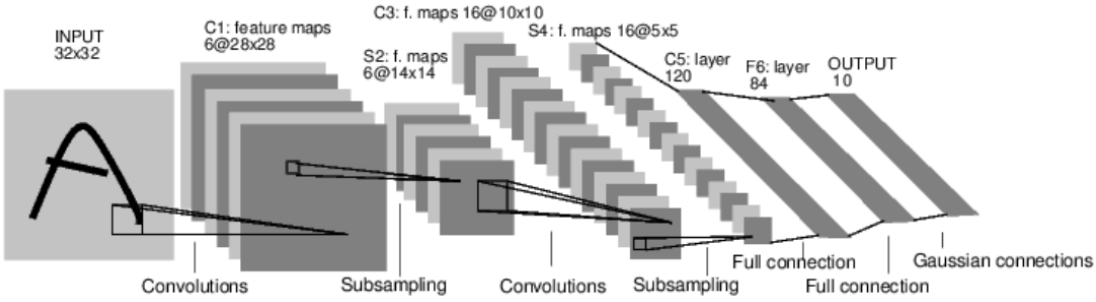


Figure 2.1 : An example of the structure of the CNNs – LeNet-5

In image processing, the information of an image is represented as pixels. But if we use a full connected network, we will get too many parameters. For example, a 512x512 RGB image will have 786,432 parameters per neuron. So if we use a three layer neural network architecture, we may need over 3 million parameters. The large number of the parameters makes the whole process very slow and would lead to overfitting.

After some investigation of images and optical systems, it has been shown that the features in an image are usually local and the low-level features are noticed first in the optical system. So the full connected network can be reduced to the locally connected network. This is one of the main ideas in the CNN.

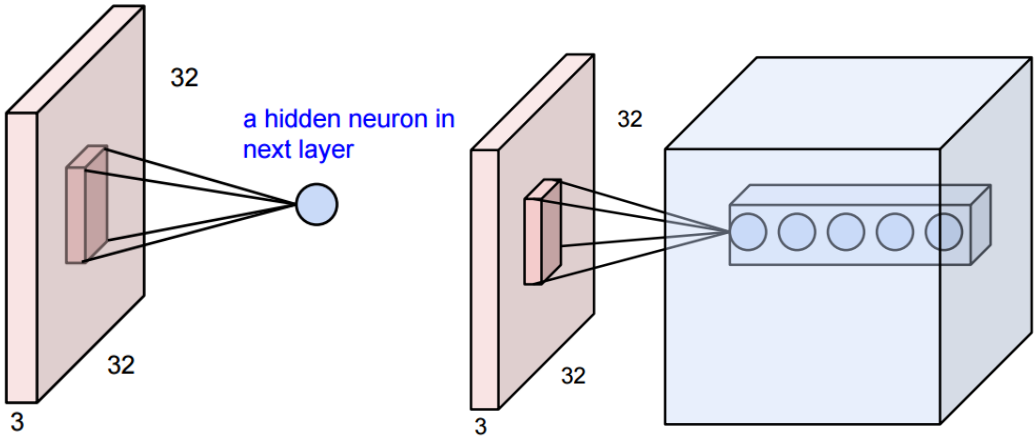


Figure 2.2 : An example of the convolutional layer

Just like the mostly image processing systems do, we can locally connect a square block to a neuron. The block size can be 3x3 or 5x5 for instance. The physical meaning of the block is like a feature window in some image processing tasks. By doing so, the number of parameters can be reduced to very small but it will not lower the performance. In order to extract more features, the same block can be connected to another neuron. The depth in the layers is how many times the same area is connected to different neuron. For example, the same area is connected to 5 different neurons. So, the depth is five in the new layer in the Figure above.

Note that the connectivity is local in space and full in depth. That is, all depth information (for example, RGB 3 channels) is not connected to next neuron but just the local information in height and width. So there might be 5x5x5 parameters in the Figure for the neuron after the blue layer if we use the 5x5 window. The first and second variables are height and width of window size and the third variable is depth of the layer.

The window is moved inside the image and the next layer can also have height and width and be a two-dimensional one. For example if we move the window 1 pixel each time, or stride 1, in a 32x32x3 image and the window size is 5x5 there are 28x28xdepth neurons in the next layer. We might find that the size is decreased (from 32 to 28). So in order to preserve the size, we add zero pad to the border in general.

Back to the example above, if we pad with 2 pixels, there are 32x32xdepth neurons in the next layer which keep the size in height and width.
The next key idea is parameter sharing. Consider the example in the Figure. There are 32x32x5 neurons in the next layer with stride 1, window size 5x5 and with zero-pad, and the depth is 5. Each neuron has 5x5x3=75 parameters (or weights). So there are 75x32x5=384000 parameters in the next layer. The idea is that we can share the parameters in each depth. That is 32x32 neurons in each depth use the same parameters. So there are only 5x5x3=75 parameters in each depth and 75x5=375 parameters in total. This greatly decreases the amount of parameters. By doing so, the neurons in each depth in the next layer is just like applying convolution to the image and the learning process is like learning the convolution kernel.

In the traditional neuron model, the sigmoid function is commonly used as the activation function. The activation function allows the network to model non-linear functions. Some simpler alternatives to the sigmoid have been proposed. One of them is Rectified Linear Units (ReLUs). The function is $f(x) = \max(0, x)$. It was found that the models with ReLUs needs less iteration time while reaching the same training error rate. So more and more CNNs models use ReLUs as the activation function.

Although the locally connected networks and parameter sharing are used, there are still many parameters in the neural networks. So, pooling layers are inserted in the networks. These layers can progressively reduce the amount of parameters and hence the computation time in the networks. The pooling layer applies downsampling to the previous layer by using the max function. It operates independently on each depth of the previous layer. That is, the depth of the next layer is the same as that of the previous layer.

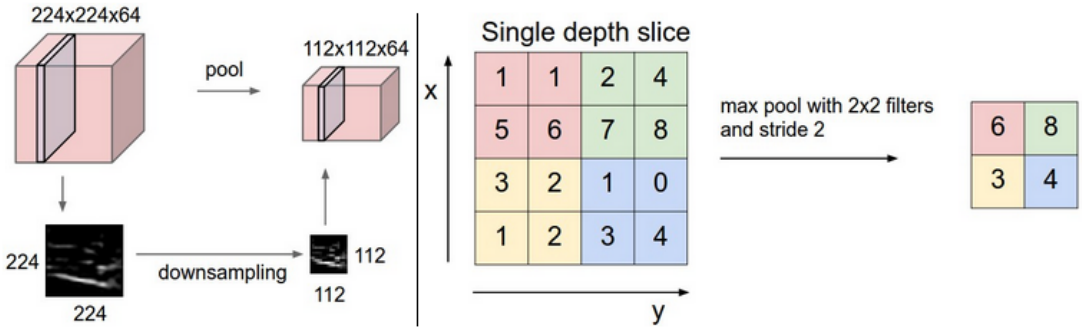


Figure 2.3 : A simple example of the pooling layer

There are two type of pooling. If the window size equal to stride, it is called traditional pooling and if the window size is larger than the stride, it is called overlapping pooling.

In additional to max pooling, other functions are also used. For example, the average of the values can be used instead of the max function.

2.3.3 RECURRENT NEURAL NETWORKS

Generally there are two kinds of neural networks which are feedforward neural networks and recurrent neural networks. A feedforward neural network is an artificial neural network where connections between the units do not form a cycle. In other word, in feedforward networks processing of information is piped through the network from input layers to output layers. In contrast, a recurrent neural network (RNN) is an artificial neural network where connections between units form cyclic paths. RNNs are called recurrent because they receive inputs, update the hidden states depended on the previous computations, and make predictions for every element of a sequence. By unrolling an RNN in time, it can be considered as a deep neural network (DNN) with indefinitely many layers.

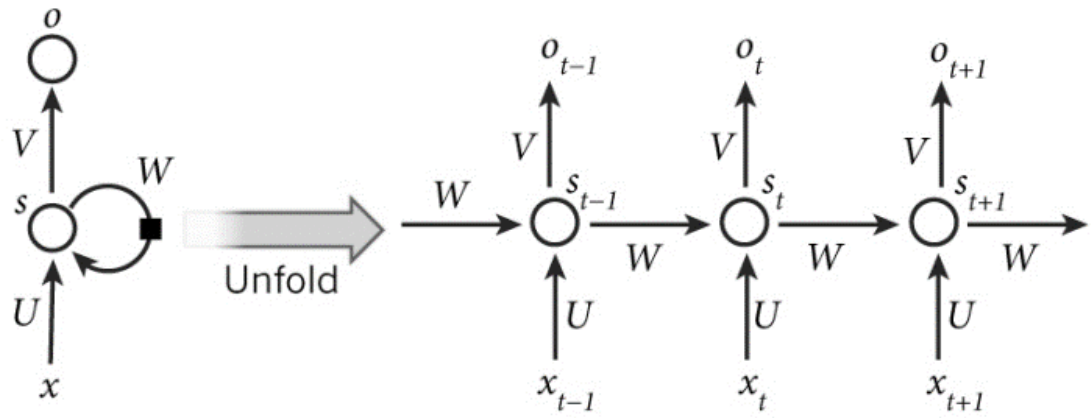


Figure 2.4 : A recurrent neural network unrolled in time.

We can consider RNNs as neural networks with memory to keep information of what has been processed so far. RNNs are very powerful dynamic systems for sequence tasks, such as speech recognition or handwritten recognition. They are powerful because they can maintain a state vector that implicitly contains information about the history of all the past elements of the sequence.

The RNN makes predictions by matrix multiplications as follows:

$$S_t = f(U\ x_t + W\ S_{t-1})$$

$$y = g(V\ S_t)$$

In these equations x_t is the input at time step t. S_t is the hidden state at time step t which is in fact the memory of the network and it is calculated based on the input at the current step and the previous hidden state. f is an activation function which transforms the inputs of the layer into its outputs and allows us to fit nonlinear hypotheses. Common choices for f are tanh and ReLUs. S_{-1} which is required to initialize the first hidden state is typically set to all zeroes. The output of the network is y which is calculated by a nonlinear function of matrix multiplication of V and S_t . In fact this nonlinear function, g, is the activation function for the output layer and usually it is the softmax function. It is simply a way to convert raw scores to probabilities. Unlike feed forward neural networks, which have different parameters at each layer, a RNN shares the same parameters (U, V, W) across all steps.

The feedforward neural networks can be trained by backpropagation algorithm. In RNNs, a slightly modified version of this algorithm called

Backpropagation through Time (BPTT) is used to train the network. The backpropagation algorithm can be extended to BPTT by unfolding RNN in time and stacking identical copies of the RNN. As the parameters that are supposed to be learnt (U, V and W) are shared by all time steps in the network, the gradient at each output depends not only on the calculations of the current time step, but also the previous time steps.

In RNNs, a common choice for the loss function is the cross-entropy loss which is given by:

$$L(y', y) = (-1/N) (\Sigma_n y'_n \log_n)$$

In this formula, |y'| is the number of training examples, y is the prediction of the network and y' is true label. The parameters U, V and W can be calculated during training by minimizing the total loss on the training data. One popular approach to do this is Stochastic Gradient Descent (SGD). The idea behind SGD is iterate over all our training examples and during each iteration, we update the parameters into a direction that reduces the error. These directions are calculated by the gradients on the loss function respect to U, V and W.

While RNNs are powerful structures, practically they are hard to train. One of the main reason is "vanishing gradient problem". While in theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps. This means in practice the range of contextual information that standard RNNs can access are limited. It has been shown that the influence of a given input on the hidden layer, and therefore on the network output, either decays or blows up exponentially as it pipes through RNN. In fact, it is hard for an RNN to bridge gaps of more than about 10 time steps between relevant input and target events.

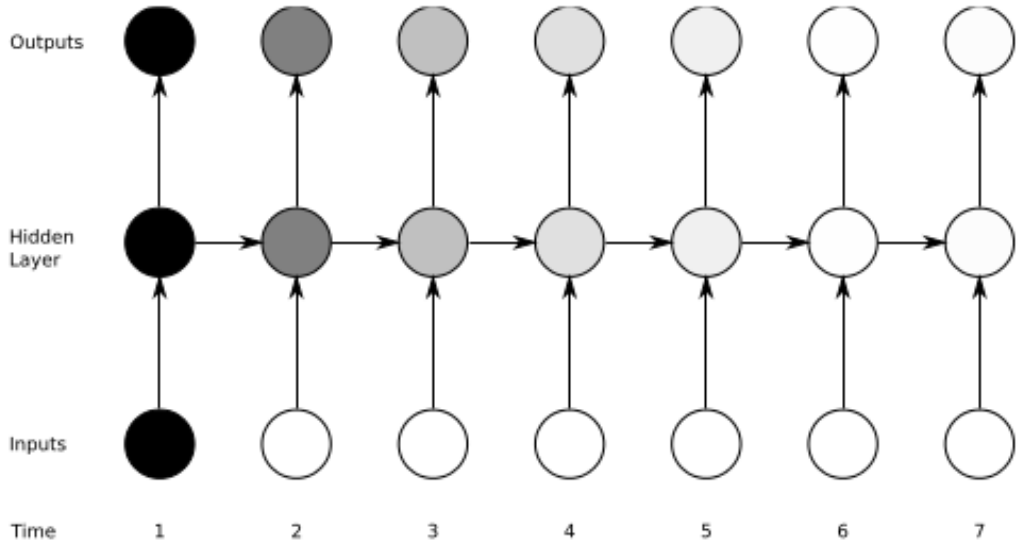


Figure 2.5: The Vanishing gradient problem in RNNs

Fortunately there are a few approaches to overcome this shortcoming of RNN. For example, W matrix can be initialized properly to combat the vanishing gradient problem. Further, using ReLU instead of tanh or sigmoid activation functions can reduce the effect of vanishing gradients. However, the most successful solution is to use Long Short-Term Memory (LSTM).

Long Short Term Memory networks are a special kind of RNN architecture that are capable of learning long-term dependencies. LSTM can learn to bridge time intervals in excess of 1000 steps even in case of noisy, incompressible input sequences, without loss of short time lag capabilities. This is achieved by multiplicative gate units learn to open and close access to the constant error flow. LSTM networks can outperform alternative RNNs and Hidden Markov Models (HMM) and other sequence learning methods in numerous applications such as speech recognition and handwriting recognition. For example, deep bidirectional LSTM achieved the best known results in automatic speech recognition which is 17.7% phoneme error rate on the classic TIMIT natural speech dataset. Also it won the ICDAR handwriting competition for the best known results in unsegmented connected handwriting recognition.

LSTM network introduces a new structure called a memory cell.

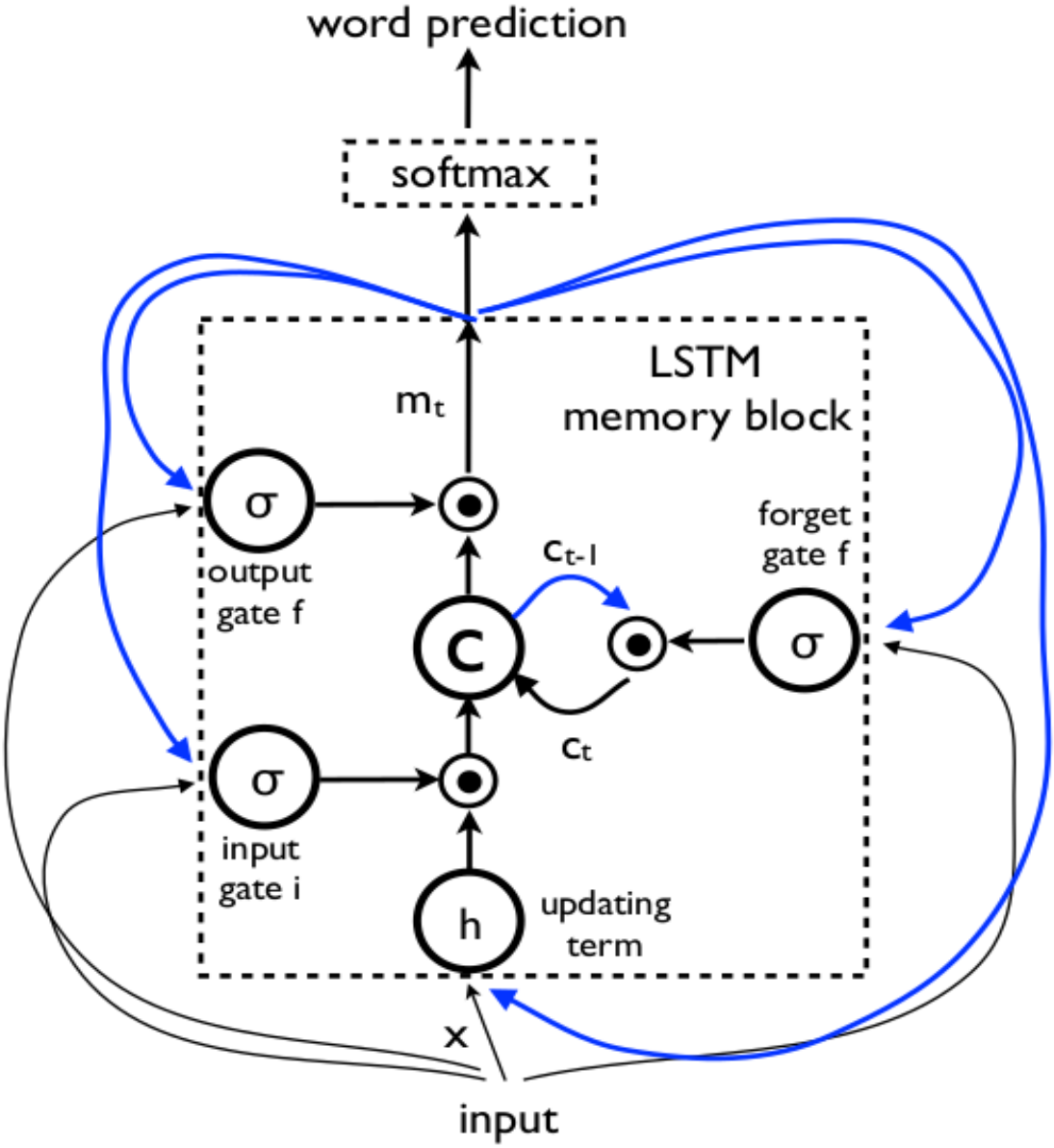


Figure 2.6: LSTM memory cell

Each memory cell contains four main elements: the input gate, forget gate, output gate and a neuron with a self-recurrent. These gates allow the cells to keep and access information over long periods of time. LSTM calculates the hidden states by a set of equation as follows:

$$i = \sigma(x_t U^i + s_{t-1} W^i)$$

$$f = \sigma(x_t U^f + s_{t-1} W^f)$$

$$o = \sigma(x_t U^o + s_{t-1} W^o)$$

$$g = \tanh(x_t U^g + s_{t-1} W^g)$$

$$c_t = c_{t-1} \circ f + g \circ i$$

$$s_t = \tanh(c_t) \circ o$$

$$y = \text{softmax}(V s_t)$$

In these equations, i, o, f and g are related to the input gate, forget gate, output gate and self-recurrent respectively. i indicates how much of the new information will be let through the memory cell. f is responsible for information should be thrown away from memory cell. Every one f in means keeping information, while every zero means get rid of this information. o decides how much of the information will be passed to expose to the next time step and also to output. g is related to a neuron with a self-recurrent loop. c_t can be called as the internal memory of the memory cell which is the sum of element-wise multiplication of previous internal memory state by the forget gate, and element-wise multiplication of self-recurrent state with input gate. Finally, s_t is related to the hidden state which can be calculated by element-wise multiplication of the internal memory with the output gate.

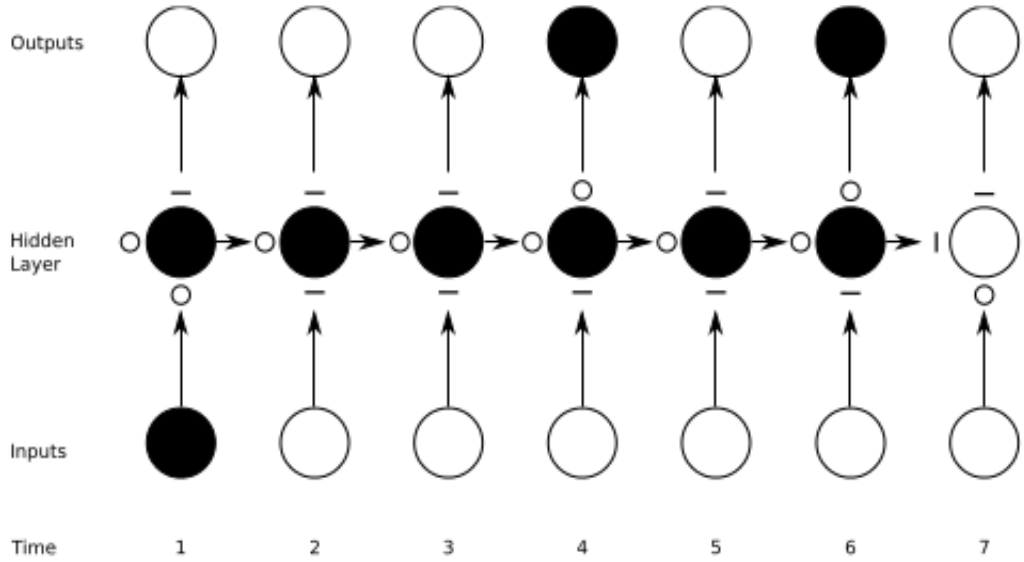


Figure 2.7: Gradient flow in LSTM.

Traditional RNNs can be considered a special case of LSTMs. If we set the input gate all ones (passing all of the new information), the forget gate all zeros (forgetting all of the previous memory) and the output gate to all ones (exposing the whole memory), we almost get standard RNN with just a small difference which is the tanh term that squeezes the output a little bit. In fact by training the parameters of the gates, an LSTM learns to handle the long-term dependencies.

2.4 HARDWARE REQUIREMENTS

- 8 GB RAM
- Nvidia GPU with a compute rating > 3.5
- Intel i5 7th Generation CPU
- 50 GB Physical Storage Space

2.5 SOFTWARE REQUIREMENTS

- MacOS X 10.11 (El Capitan) or later. Ubuntu 16.04 or later; Windows 7 or later.
- Python 3
- CUDA 9 Toolkit
- cuDNN v7.0
- Tensorflow v1.6
- Keras v2.1.5
- NumPy v1.14
- Matplotlib v2.2.2

Chapter 3

LITERATURE SURVEY

3.1 LITERATURE SURVEY

The problem of generating textual descriptions from visual data have been studied as early as the 1990s. However, the work chiefly addressed generating descriptions of videos [1,2]. Many works have tried to treat this as a retrieval problem instead of a generation problem by retrieving the description that best describes the input image from a large database of images and description fragments embedded in a representation format [3,4,5,6,7].

The modern approaches to the problem of generating image captions can be classified into two paradigms: bottom-up and top-down. The bottom-up methods start with a set of words related to various aspects of that image and then combine those words into a single coherent caption[8,9,10,11,12,13]. On the other hand, the top-down approach is to start with a representation of the image and translate it into a description word-by-word or character-by-character[14,15,16,17,18,19].

The current state of the art method is a top-down approach. It employs a joint neural network based model to generate image caption from an image instead of using several separate models to extract the gist of the image, to build sentences and so on [19]. This family of end-to-end image captioning models have their roots in the advances in machine translation and conversational models. Recurrent Neural Network based models have been used to achieve state-of-the-art performance without fragmenting the task into a series of separate tasks [20]. Such models, called sequence-to-sequence models, consist primarily of two parts – an encoder and a decoder. The encoder converts the input sentence into a fixed length intermediate representation, which in turn is used by the decoder to generate the output sentence. The general architecture of an end-to-end image caption generator consists of a Convolutional Neural Network(CNN) based encoder and a Recurrent Neural Network(RNN) based decoder.

A Convolutional Neural Network(CNN) is a specific subset of feed forward neural networks and the CNN was designed to process multi-dimensional data like a two dimensional image having three color channels [21]. CNNs consist of three types of layers – Fully Connected layers, Convolution layers, and Pooling layers. The convolution layers exploit the fact that values that in spatial data, local values such as pixels in the same neighborhood are more correlated. Pooling layers merge similar features and thus reduce the dimensionality of their input. There are many types of pooling layers including max pooling and average pooling which return the maximum value of their input and the average of all values in their input respectively (Citation Needed). Most implementations of CNN based architectures use the Rectified Linear Unit(ReLU) activation function [22].

A key feature of CNN based architectures is their depth, that is, they consist of many layers. Many techniques have been proposed to ensure that deep CNN architectures consisting of millions of parameters do not overfit their training data. Batch normalization[23] and dropout[24] are two such regularization methods. While batch normalization regularizes the CNN's training by introducing noise into each layer by normalizing their input, dropout works by randomly setting a fraction of the layer's inputs to zero.

A Recurrent Neural Network(RNN) is a modified version of the feed forward neural network[22]. The RNN has a feedback mechanism which allows it to process sequential input like a sentence consisting of words. Due to this addition of a feedback loop, the RNN is trained by a modified version of backpropagation called backpropagation-through-time.

Since the RNN suffers from issues like vanishing gradient and exploding gradient which make training difficult, some alternatives have been proposed[22]. The Long Short-Term Memory cell(LSTM)[25] and the Gated Recurrent Unit (GRU)[26] are two popular alternatives to the RNN which leverage a gating mechanism to control the gradient flow during training. Further, to improve performance on natural

language text generation tasks, various improvements have been proposed like Attention Mechanisms[27,28] and Bidirectional RNNs[29] have been proposed (Citation Needed). The attention mechanism is employed in encoder decoder models and it allows the decoder access to each intermediate state of the encoder with weights signifying the importance of each state instead of a fixed size encoded representation. An attention mechanism allows the decoder RNN to decide which parts of the input should be focused on. Bidirectional RNNs consist of two RNNs such that if one processes a sentence from start to the end, the other processes the sentence from the end to the start. The outputs of the two RNNs are concatenated as the output of the Bidirectional RNN. Bidirectional RNNs have been shown to perform better than unidirectional RNNs at tasks like Handwriting Generation and Machine Translation[30].

While the end-to-end image caption generation neural network can be trained using randomly initialized weights, steps to convergence can be reduced by using the weights of a neural network trained to perform a related task. For instance, the CNN encoder network can be replaced by some layers of a CNN trained on the task of multi-class classification. Due to the extreme popularity of the ImageNet Competition, many CNN architectures have been developed and the weights of some of the trained models have been released. Some of the prominent architectures whose weights are freely available are VGG16, VGG19 [31], ResNet50[32] and, InceptionV3[33] .

In the literature, there are chiefly two ways of sampling captions from the output probabilities of the neural network – greedy search and beam search sampling. While the greedy method takes the word with maximum probability at each time-step as the output, the beam search approach takes the top k most probable words and yields the caption with the maximum sum of scores of each constituent word. This makes the greedy approach fast but not optimal and the beam search approach slower but more likely to be optimal.

Quantitatively evaluating the correctness a generated caption is a hard problem.

Many metrics have been proposed which compare the generated caption with a reference caption and yield a value that correlate well with human judgement. Bilingual Evaluation Understudy(BLEU)[34], Recall-Oriented Understudy for Gisting Evaluation(ROUGE)[35], Metric for Evaluation of Translation with Explicit Ordering (METEOR)[36], Consensus-based Image Description Evaluation(CIDEr) [37] and, Perplexity are some of the metrics commonly used in the literature.

Due to the recent surge in interest in the problem of generating captions from visual data, many datasets have been released. Flickr8k[3] and Flickr30k[38] consist of 8,000 and 31,783 images respectively, having 5 captions per image. The images in these datasets have been obtained from the Flickr image sharing website. The Pascal VOC[39] dataset was originally intended for the task of object classification but it also includes descriptions of 50 images from each of the 20 object classes. Each image has 5 captions. The PASCAL-50S[40] dataset consists of 1,000 images each having 50 captions. The SBU[5] dataset has 1 million images fetched from the Flicker website along with user provided descriptions. While this is a fairly large dataset, the SBU dataset is generally not preferable for the task of image captioning as the descriptions by the image owner may be noisy. The Microsoft Common Objects in Context (MS COCO)[41] is a large-scale object detection, segmentation, and captioning dataset. This dataset has 82783 images for training and 40504 for validation and testing. Each image has 5 captions. MS COCO is one of the most popular datasets for the task of image captioning and has been used to benchmark many works.

Chapter 4

SYSTEM DESIGN

4.1 HIGH LEVEL DESIGN

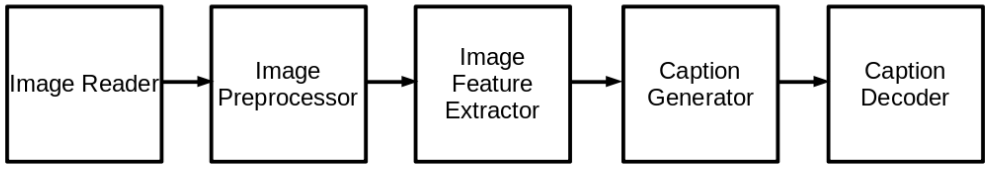


Figure 4.1: High level design architecture

The project consists of various components which include :

- Image reader
- Image pre-processor
- Image feature extractor
- Caption generator
- Caption decoder

4.1.1 IMAGE READER

This component is responsible for acquiring the input image. It allows reading from a camera and reading an image from the physical storage. The image reader decompresses and decodes the image file from file formats like JPEG or PNG. The decoded image is represented as a multi-dimensional array for further processing.

4.1.2 IMAGE PRE-PROCESSOR

The image pre-processor resizes the image array to a fixed size. This is necessary as convolutional neural network take fixed size input. The resized image's pixel values are normalized. This is accomplished by dividing each value by 255, which is the maximum possible value.

4.1.3 IMAGE FEATURE EXTRACTOR

The image feature extractor takes the pre-processed image as input. This component extracts the salient features from the image. These features can be thought of the gist of the image. This gist contain information about various shapes and low level feature. The image features are extracted by transfer learning using the layers of a pre-trained convolutional neural network.

4.1.4 CAPTION GENERATOR

The caption generator model takes an image as an input, extracts useful information and "translates" it into a suitable description. Therefore, while training, we try to maximize the probability of the correct description given the image. The parameters of the model given the training sample can be expressed as

$$\theta^{*} = \text{argmax}_{\theta} \Sigma_{(S,I)} \log p(S|I;\ \theta)$$

where,
S is the correct description,
I is the image and,
θ represents the parameters of the model.

The probability p can be represented using chain rule as

$$\log p(S|I;\theta) = \sum_{n} \log p(S_n|I,S_0,S_1,...S_{n-1};\theta)$$

where,
S is a sentence represented as $(S_0,S_1,...S_{n-1})$ and,
$S_0,S_1,...S_{n-1}$ are words in the sentence.

Therefore, we try to optimize the sum of probabilities using stochastic gradient descent while training.

4.1.4 CAPTION DECODER

This component is responsible for actually generating captions from the probabilities outputted by the caption generator. The caption decoder works in conjunction with the caption generator model. The generated sequence of word indices is converted to readable text by translating the indices to words and joining the words by a space character.

4.2 LOW LEVEL DESIGN

4.2.1 IMAGE READER

The image reader is implemented as a python function. It reads images from disk using the Image.open function from the Python Imaging Library. The read and decoded image object is converted to a NumPy array of size (height, width, channels).

4.2.2 IMAGE PRE-PROCESSOR

This component performs pre-processing operations on the NumPy array as vector operations. The image is resized and the pixel values are scaled and normalized.

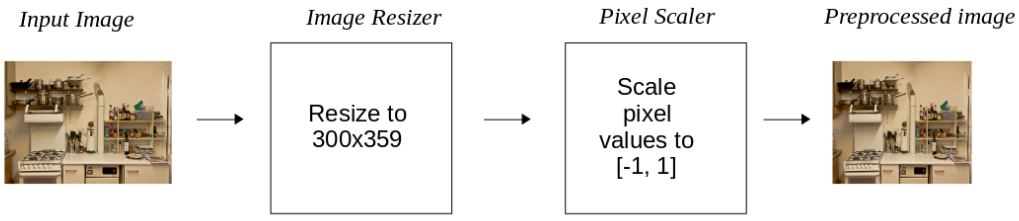


Figure 4.2: Image pre-processor flowchart.

4.2.3 IMAGE FEATURE EXTRACTOR

The Image Encoder extracts latent features from the preprocessed image. Deep convolutional neural networks have achieved state-of-the-art performances in image classification in recent years. Specifically, for this project, the Inception-V3 CNN is used.The CNN is pre-trained on the ImageNet Dataset for the task of Large Scale Object Classification. The Inception-V3 CNN achieved the best performance on large scale classification on the ILSVRC 2012 classification benchmark. Each Image Feature Vector contains 2048 extracted features.

4.2.4 CAPTION GENERATOR

The Language Model generates word probabilities at multiple time-steps using the encoded representation of the image. This is implemented using an LSTM. During the training phase, this LSTM is trained on Image-Caption Data. The caption generator language model is implemented using LSTMs. At Each time-step,the neural network model receives two inputs - the extracted image feature vector and the partially generated caption. The model estimates the probabilities of the occurance of the next word. The model is trained by optimizing the Categorical Cross Entropy Loss using the ADAM optimization algorithm.

4.2.5 CAPTION DECODER

This component generates sentences using the word probabilities from the caption generator model. While many techniques exist for decoding sentences from a sequence of word probabilities, Beam Search and Greedy Sampling are the two most popular ones.

1) Greedy Sampling: where we just sample the first word according to word probabilities, then provide the corresponding embedding as input and sample next probabilities, continuing like this until we sample the special end-of-sentence token.
2) Beam Search: iteratively consider the set of the k best sentences up to time t as candidates to generate sentences of size t + 1, and keep only the resulting best k of them.

Chapter 5

SYSTEM IMPLEMENTATION

5.1 DATASET

5.1.1 OVERVIEW

The Microsoft Common Objects in Context (MS COCO) dataset is used for training thecaption generator. MS COCO is a large-scale object detection, segmentation, andcaptioning dataset. It has several features:

1) Object segmentation
2) Recognition in context
3) Superpixel stuff segmentation
4) 330K images (>200K labeled)
5) 1.5 million object instances
6) 80 object categories
7) 91 stuff categories
8) 5 captions per image
9) 250,000 people with keypoints

The Image Caption Dataset consists of images and sentences in English describing theseimages. This dataset has 82,783 images for training and 40,504 for validation and testing. Each image has 5 captions

5.1.2 IMAGES

Information about images is stored in the “images” field of the JavaScript Object Notation (JSON) format file captions_train2014.json.

id	coco_url	date_captured	file_name	flickr_url	height	license	width
57870	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 16:28:13	COCO_train2014_000000057870.jpg	http://farm4.staticflickr.com/3153/2970773875_...	480	5	640
384029	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 16:29:45	COCO_train2014_000000384029.jpg	http://farm3.staticflickr.com/2422/3577229611_...	429	5	640
222016	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 16:37:59	COCO_train2014_000000222016.jpg	http://farm2.staticflickr.com/1431/1118526611_...	640	1	480
520950	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 16:44:40	COCO_train2014_000000520950.jpg	http://farm8.staticflickr.com/7007/6413705793_...	427	3	640
69675	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 16:46:33	COCO_train2014_000000069675.jpg	http://farm8.staticflickr.com/7156/6415223357_...	480	4	640

Figure 5.1: Details of images in the MS COCO dataset.

5.1.2 CAPTIONS

Information about the captions is stored in the “annotations” field of the JSON format file captions_train2014.json.

id	caption	image_id
48	A very clean and well decorated empty bathroom	318556
67	A panoramic view of a kitchen and all of its a...	116100
126	A blue and white bathroom with butterfly theme...	318556
148	A panoramic photo of a kitchen and dining room	116100
173	A graffiti-ed stop sign across the street from...	379340

Figure 5.2: Details of captions in the MS COCO dataset.

5.1.2 JOINED DATA

The image and caption data can be joined on the image_id key. On joining, rows showing images and their corresponding captions are obtained.

id	caption	image_id	coco_url	date_captured	file_name	flickr_url	height	license	width
48	A very clean and well decorated empty bathroom	318556	http://images.cocodataset.org/train2014/COCO_t...	2013-11-15 05:00:35	COCO_train2014_000000318556.jpg	http://farm4.staticflickr.com/3133/3378902101_...	640	1	480
67	A panoramic view of a kitchen and all of its a...	116100	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 19:43:48	COCO_train2014_000000116100.jpg	http://farm9.staticflickr.com/8084/8329525274_...	182	2	640
126	A blue and white bathroom with butterfly theme...	318556	http://images.cocodataset.org/train2014/COCO_t...	2013-11-15 05:00:35	COCO_train2014_000000318556.jpg	http://farm4.staticflickr.com/3133/3378902101_...	640	1	480
148	A panoramic photo of a kitchen and dining room	116100	http://images.cocodataset.org/train2014/COCO_t...	2013-11-14 19:43:48	COCO_train2014_000000116100.jpg	http://farm9.staticflickr.com/8084/8329525274_...	182	2	640
173	A graffiti-ed stop sign across the street from...	379340	http://images.cocodataset.org/train2014/COCO_t...	2013-11-15 06:07:46	COCO_train2014_000000379340.jpg	http://farm1.staticflickr.com/1/163009_b84730e...	640	3	480

Figure 5.3: Joined image and caption details in MS COCO dataset.

Using this joined table, examples of images and their corresponding captions can be seen.

```
the kitchen is full of spices on the rack
A kitchen with counter, oven and other accessories.
A small kitchen that utilizes all of its space.
This small kitchen has pots, pans and spices on display
A VERY SMALL KITCHEN WITH A STOVE AND A SHELF OF POTS


Image is of dimensions 640 by 427
```

Figure 5.4: Example 1 of an image and its captions in the MS COCO dataset.

```
A graffiti-ed stop sign across the street from a red car
A vandalized stop sign and a red beetle on the road
A red stop sign with a Bush bumper sticker under the word stop.
A stop sign that has been vandalized is pictured in front of a parked car.
A street sign modified to read stop bush.


Image is of dimensions 480 by 640
```

Figure 5.5: Example 2 of an image and its captions in the MS COCO dataset.

5.2 MODEL ARCHITECTURES

5.2.1 IMAGE FEATURE EXTRACTOR

The image feature extractor is implemented as an Inception-V3 convolutional neural network.

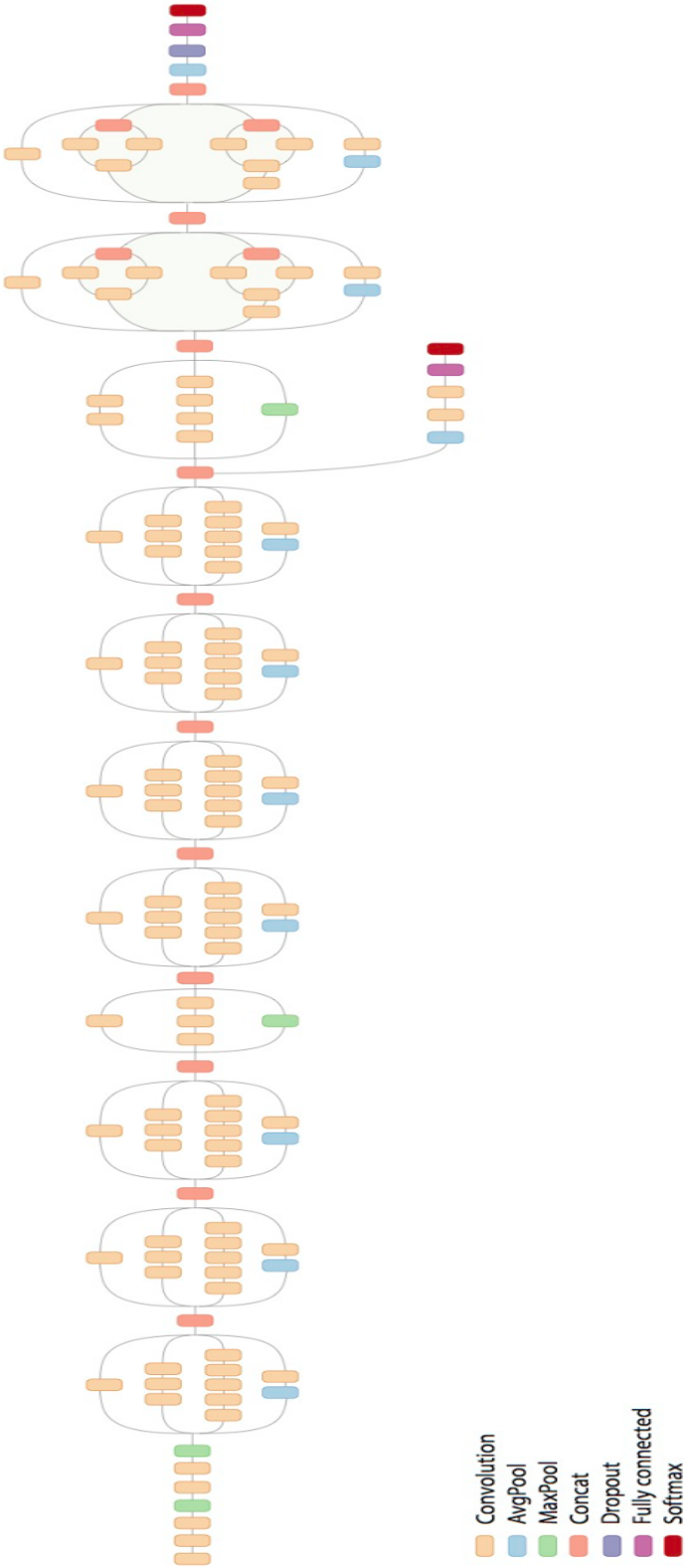


Figure 5.6: Inception-V3 model architecture

5.2.2 CAPTION GENERATOR

The Caption generator language model is implemented using LSTMs.

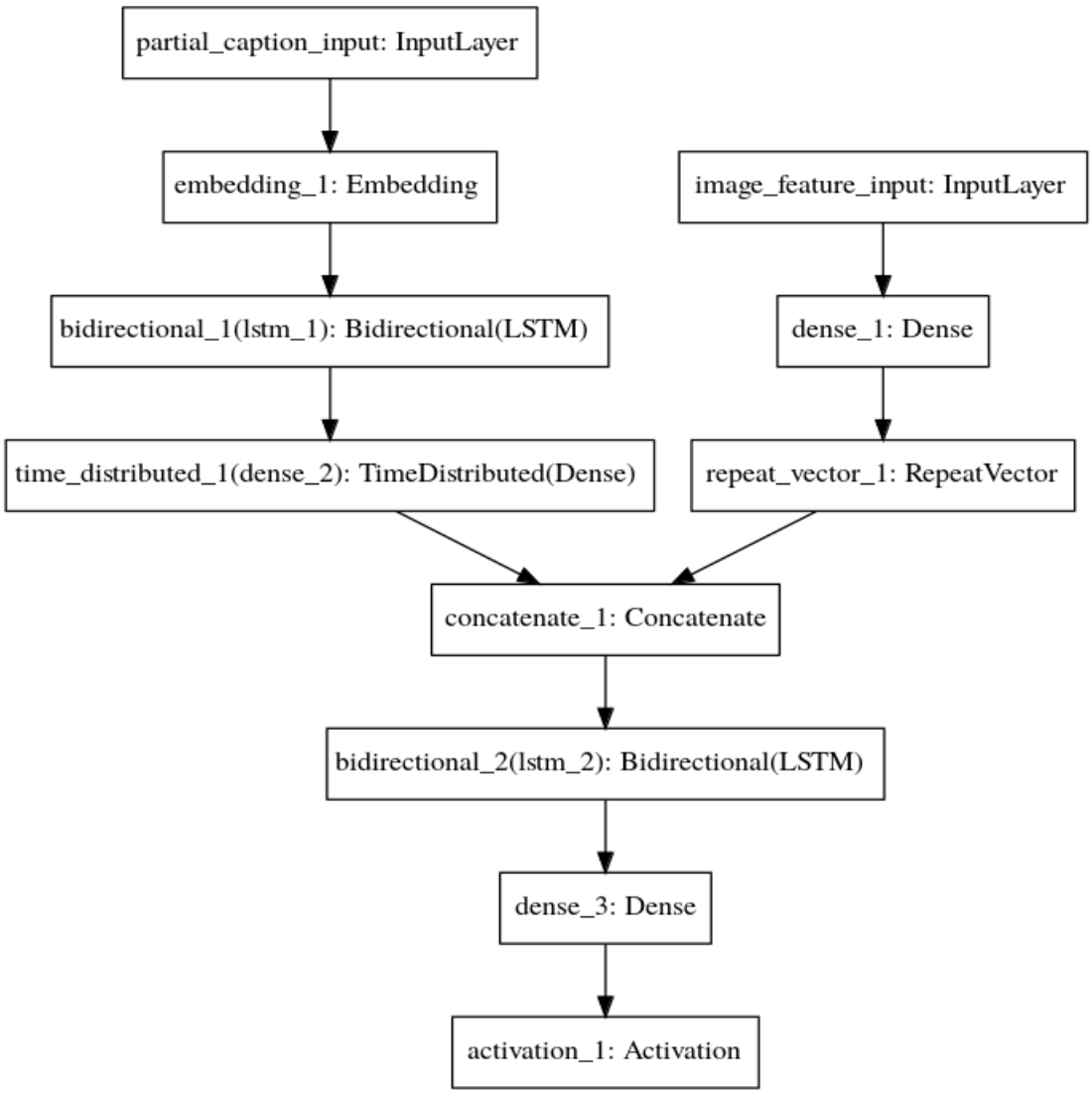


Figure 5.7: Caption generator model architecture

5.2.2 Caption Decoder

The Caption decoder implements both beam search and greedy search strategies. While greedy search is fast, the beam search strategy is more optimal and yields more grammatically correct captions.

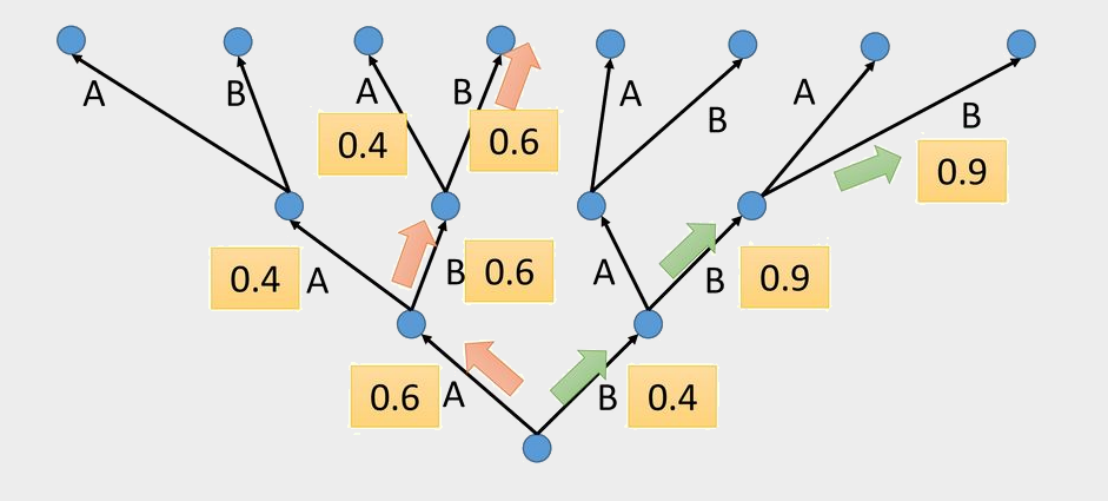


Figure 5.8: Beam search (red) as compared to greedy search (green)

Chapter 6

RESULTS AND DISCUSSION

6.1 OUTPUT

../data/train2014/COCO_train2014_000000387857.jpg
Greedy Search: a living room with a table and a table and a table
Beam Search: a view of a living room next to a fireplace END END END in

Caption Annotations
A wooden table topped with pie and cake.
A log house shows plates, a table, and dishes on display.
A wood room has a wood table, chairs, and bench in it.
Actors in an old cabin in a museum
The home is constructed of wood including the furniture within.

Figure 6.1: Output example 1

../data/train2014/COCO_train2014_000000579846.jpg
Greedy Search: a bathroom with a large toilet and a window
Beam Search: a picture of a bathroom in a bathroom END END END END END UNK

Caption Annotations
A nearly empty house with a bicycle mounted on the wall.
A hall opening onto a bathroom, bedroom, and two other rooms.
A view of a hallway in a house.
There is a hallway leading to a door where bicycle tires are showing.
A house hallway with all the bedroom doors open.

Figure 6.2: Output example 2

```
../data/train2014/COCO_train2014_000000465765.jpg
Greedy Search: a man is skiing down a snowy slope
Beam Search:   a group of people skiing down a snow covered slope END END END in


Caption Annotations
Three snowboarders travel down the side of a snowy hill.
Three men snowboarding down slope with trees in the background.
A group of snowboarders sliding down a snowy mountain.
Three young people snowboard down a snowy slope.
Three people are enjoying a day of skiing during the winter.
```

Figure 6.3: Output example 3

../data/train2014/COCO_train2014_000000573518.jpg
Greedy Search: a cat is sitting on a table with a laptop
Beam Search: a close up of a black and white photo of a keyboard END and

Caption Annotations
A black colored keyboard for a computer system.
A black computer keyboard with lcd panel and numeric pad.
A wireless keyboard compatible with role playing games.
A computer keyboard, with black keys and white lettering.
A black computer keyboard with a bunch of keys on it

Figure 6.4: Output example 4

../data/train2014/COCO_train2014_000000173185.jpg
Greedy Search: a plate of food with a banana and a spoon
Beam Search: a close up of a bunch of food on a table END END for

Caption Annotations
A ladle of soup being placed into a bowl.
a person serving them self a bowl of soup
A cook dishes a stew from a pan onto a plate.
A person holding a bowl while spooning vegetable soup into it.
A person scooping from a large pot of soup on the stove into a soup bowl.

Figure 6.5: Output example 5

../data/train2014/COCO_train2014_000000095430.jpg
Greedy Search: a bathroom with a sink and a sink
Beam Search: a picture of a bathroom next to a door END END END END and

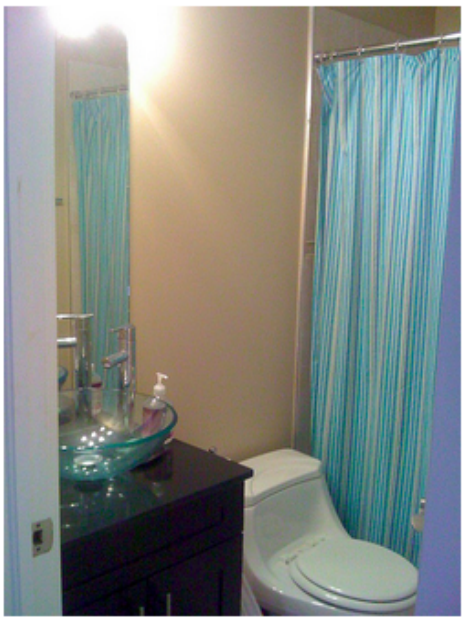
Caption Annotations
A small bathroom has been fitted in a wood motif
A vanity and sink next to a window in a home.
A sink is shown in front of a window.
A large bathroom with a sink underneath a window.
A sink built under a window in the bathroom area.

Figure 6.6: Output example 6

../data/train2014/COCO_train2014_000000279672.jpg
Greedy Search: a large kitchen with a large large window
Beam Search: a picture of a bunch of food on a table END END END on

Caption Annotations
An herb that is in front of a toaster oven.
A vegetable in front of a toaster oven.
An evergreen sprig is sitting on top of an appliance.
A microwave is next to sage or some kind of spice.
This is a toaster with a green stem from a tree in front of it.

Figure 6.7: Output example 7



../data/train2014/COCO_train2014_000000017866.jpg
Greedy Search: a bathroom with a sink and a sink and a sink
Beam Search: a picture of a bathroom in a bathroom END END END END END and

Caption Annotations
A bathroom with a blue shower curtain over the bathtub.
A bathroom with, blue shower curtains, a toilet, and sink.
Bathroom with shower, toilet and sink and vanity view.
The side view of the entrance to a bathroom.
This is a home bathroom with blue shower curtains.

Figure 6.8: Output example 8

../data/train2014/COCO_train2014_000000165499.jpg
Greedy Search: a bathroom with a sink and a sink
Beam Search: a picture of a bathroom next to a bath tub END in a bathroom

Caption Annotations
A bathroom shower with glass doors and tile walls.
a bathroom with a white sink shower and toilet
Bathroom that has a toilet, shower, and sink.
A modern bathroom has a corner shower that's clear.
A bathroom with a see-through shower door.

Figure 6.9: Output example 9

../data/train2014/COCO_train2014_000000237886.jpg
Greedy Search: a kitchen with a sink and a sink
Beam Search: a picture of a picture of a small bathroom

Caption Annotations
A his and hers bathroom sink with a lighted mirror.
His and hers bathroom sink with the lights turn on by mirror.
A vanity mirror over a two face bowl counter.
A vanity with two sinks an alighted mirror over it
a bathroom with two sinks and a large mirror

Figure 6.10: Output example 10

Chapter 7

CONCLUSION & FUTURE WORK

7.1 CONCLUSION

This project is an attempt at improving the accessibility of image rich multimedia content. While the alt-text attribute is supported by nearly all major systems, it requires the content creator to provide an alternative text. Unfortunately, a lot of the popular websites do not have alternative text for their image content. The project employs machine learning to generate short sensible descriptive captions for images that lack alternative text. This system is not only helpful to the visually impaired but also useful for people with slow Internet connections and people using text-only browsers. The system leverages transfer learning for image feature extraction and thus reducing the training time and resources spent for training. The MS COCO dataset provides sufficient data to allow the system to generalize to most real world situations.

Hence the automated image captioning system helps the visually impaired make sense of multimedia content.

7.2 SCOPE OF FUTURE WORK

The current implementation of the captioning system can be improved by incorporating capsule networks. Further, such a caption generation system can

be incorporated into an embedded system that helps the visually impaired experience the world around them by describing it to them.

Chapter 8

REFERENCES

[1] Gerber, R., & Nagel, H. (1996). Knowledge representation for the generation of quantified natural language descriptions of vehicle traffic in image sequences. ICIP.

[2] Yao, B.Z., Yang, X., Lin, L., Lee, M.W., & Zhu, S. (2010). I2T: Image Parsing to Text Description. Proceedings of the IEEE, 98, 1485-1508.

[3] Hodosh, Micah et al. “Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics (Extended Abstract).” IJCAI (2013).

[4] Gong, Y., Wang, L., Hodosh, M., Hockenmaier, J., & Lazebnik, S. (2014). Improving Image-Sentence Embeddings Using Large Weakly Annotated Photo Collections. ECCV.

[5] Ordonez, V., Kulkarni, G., & Berg, T.L. (2011). Im2Text: Describing Images Using 1 Million Captioned Photographs. NIPS.

[6] Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., & Ng, A.Y. (2014). Grounded Compositional Semantics for Finding and Describing Images with Sentences. TACL, 2, 207-218.

[7] Karpathy, A., & Fei-Fei, L. (2015). Deep Visual-Semantic Alignments for Generating Image Descriptions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3128-3137.

[8] Elliott, D., & Keller, F. (2013). Image Description using Visual Dependency Representations. EMNLP.

[9] Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D.A. (2010). Every Picture Tells a Story: Generating Sentences from Images. ECCV.

[10] Fang, H., Gupta, S., Iandola, F.N., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Zitnick, C.L., & Zweig, G. (2015). From captions to visual concepts and back. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1473-1482.

[11] Kuznetsova, P., Ordonez, V., Berg, A.C., Berg, T.L., & Choi, Y. (2012). Collective Generation of Natural Image Descriptions. ACL.

[12] Lebret, Rémi et al. “Simple Image Description Generator via a Linear Phrase-Based Approach.” CoRR abs/1412.8419 (2014): n. pag.

[13] Li, S., Kulkarni, G., Berg, T.L., Berg, A.C., & Choi, Y. (2011). Composing Simple Image Descriptions using Web-scale N-grams. CoNLL.

[14] Chen, X., & Zitnick, C.L. (2015). Mind's eye: A recurrent visual representation

for image caption generation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2422-2431.

[15] Donahue, Jeff et al. “Long-Term Recurrent Convolutional Networks for Visual Recognition and Description.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 2625-2634.

[16] Karpathy, Andrej and Li Fei-Fei. “Deep Visual-Semantic Alignments for Generating Image Descriptions.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 3128-3137.

[17] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A.L. (2015). Learning Like a Child: Fast Novel Visual Concept Learning from Sentence Descriptions of Images. 2015 IEEE International Conference on Computer Vision (ICCV), 2533-2541.

[18] Mao, J., Xu, W., Yang, Y., Wang, J., & Yuille, A.L. (2014). Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). CoRR, abs/1412.6632.

[19] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. ICML.

[20] Vinyals, O., & Le, Q.V. (2015). A Neural Conversational Model. CoRR, abs/1506.05869.

[21] LeCun, Y. (1998). Gradient-based Learning Applied to Document Recognition.

[22] Goodfellow, I.J., Bengio, Y., & Courville, A.C. (2015). Deep Learning. Nature, 521 7553, 436-44.

[23] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML.

[24] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929-1958.

[25] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural computation, 9 8, 1735-80.

[26] Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. EMNLP.

[27] Mnih, V., Heess, N., Graves, A., & Kavukcuoglu, K. (2014). Recurrent Models of Visual Attention. NIPS.

[28] Luong, T., Pham, H., & Manning, C.D. (2015). Effective Approaches to Attention-based Neural Machine Translation. EMNLP.

[29] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks : the official journal of the International Neural Network Society, 18 5-6, 602-10.

[30] Doetsch, P., Zeyer, A., & Ney, H. (2016). Bidirectional Decoder Networks for

Attention-Based End-to-End Offline Handwriting Recognition. 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 361-366.

[31] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

[32] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[33] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826.

[34] Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. ACL.

[35] Lin, C. (2004). ROUGE: A Package For Automatic Evaluation Of Summaries.

[36] Lavie, A., & Denkowski, M.J. (2009). The Meteor metric for automatic evaluation of machine translation. Machine Translation, 23, 105-115.

[37] Vedantam, R., Zitnick, C.L., & Parikh, D. (2015). CIDEr: Consensus-based image description evaluation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4566-4575.

[38] Young, P., Lai, A., Hodosh, M., & Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. TACL, 2, 67-78.

[39] Everingham, M., Gool, L.V., Williams, C.K., Winn, J.M., & Zisserman, A. (2009). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88, 303-338.

[40] Vedantam, R., Zitnick, C.L., & Parikh, D. (2014). Collecting Image Description Datasets using Crowdsourcing. CoRR, abs/1411.3041.

[41] Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft COCO: Common Objects in Context. ECCV.