# Image-Based Localization of User-Interfaces

A Project Presented To

The Faculty of Department of Computer Science
San José State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Riti Gupta

December, 2019

The Designated Project Committee Approves the Master's Project Titled

Image-Based Localization of User-Interfaces

By

Riti Gupta

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

December 2019

| | |
|---|---|
| Dr. Christopher Pollett | Department of Computer Science |
| Dr. Robert Chun | Department of Computer Science |
| Dr. Kong Li | Department of Computer Science |

# ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to Dr. Christopher Pollett for his pertinent guidance, advice, and collaboration throughout the duration of my project. I consider myself to be extremely fortunate to have had an opportunity to work with someone as brilliant as him.

I am also grateful to my committee members Dr. Robert Chun and Dr. Kong Li for providing their valuable feedback and guidance.

Finally, I thank my wonderful parents and friends for their countless hours of support and encouragement along the way.

## ABSTRACT

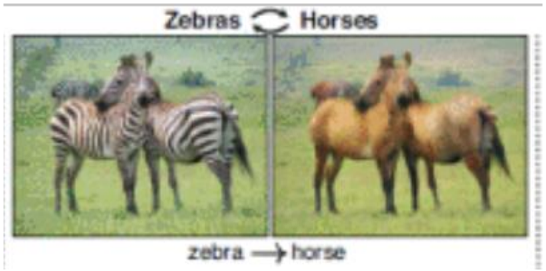
Image localization corresponds to translating the text present in the images from one language to other language. The aim of the project is to develop a methodology to translate the text in image captions from English to Hindi by taking context of the images into account. A lot of work has been done in this field [22], but our aim was to explore if the accuracy can be further improved by consideration of the additional information imparted by the images apart from the text. We have explored Deep Learning using neural networks for this project. In particular, Recurrent Neural Networks (RNN) have been used which are ideal for sequence translations and would meet the needs of this project which involves text sequences. This technique of image localization would be beneficial in a lot of fields. For example, in order to make the text data accessible to everyone, text data should be translated in multiple languages spoken by people across the world. This will help in the growth at the rural areas and countries where English is not spoken by giving them access to data in their local languages. This could also benefit tourists who would then be able to understand the sign boards and posters in a foreign country. With accurate data translation, the old manuscripts can also be translated to English upon which further research can be carried out.

***Index terms:*** **Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Neural Network, Deep Learning, Optical Character Recognition (OCR)**

# TABLE OF CONTENTS

# 1. INTRODUCTION

Data can be in various mediums including, but not limited to, text, image, video and audio-based format. The aim of this paper is to focus on the translation of textual data contained in the captions of images. A lot of work has been done with respect to image to image translation to learn the mapping G : X →Y between input and output images [1]. Image translation can be in various flavors like transforming the black and white image into a colored image, transforming a blurred image into a clear image and so on. This is usually done using Generative Adversarial Networks GANs which learn a function G(z) by conditioning the input image to target image [1]. Figure 1 below shows some examples of image translations.



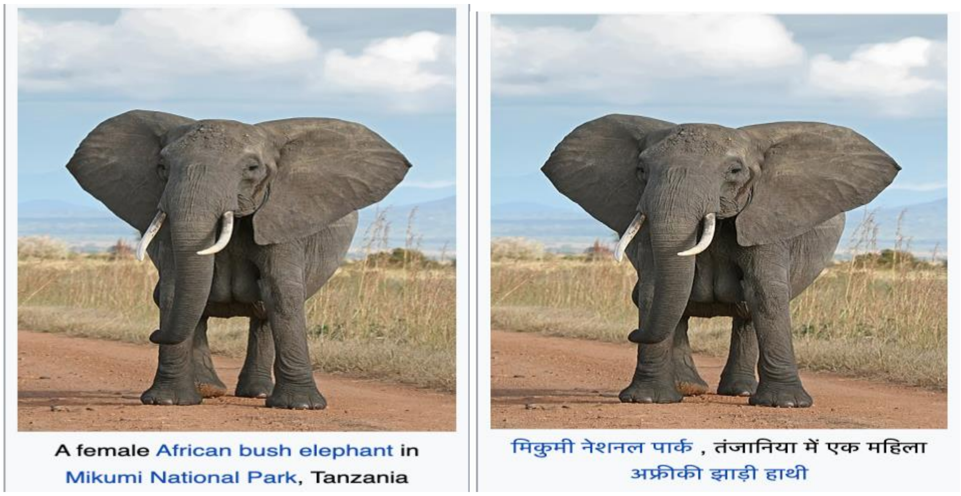

A female African bush elephant in Mikumi National Park, Tanzania
मिकुमी नेशनल पार्क , तंजानिया में एक महिला अफ्रीकी झाड़ी हाथी

Figure 1. Image translation examples (Blurred → Clear, Zebra → Horse, English → Hindi).

For translating the text contained in images from one language to other, we thought of two approaches. One was to translate the image directly from one language to other language using pixel to pixel transformation by understanding the edges and shapes of the text. Second was to first extract the textual information from the images and then perform the translation.

As a part of this project, I worked on the second alternative of extracting the text and then performing sequence to sequence translation. Lot of machine learning models have already been built for sequence to sequence translation. These models perform translation by passing the information of previous timesteps to current timestamp of the sequence improving the accuracy of the models as shown in Figure 2. The aim is to transform the input sequence $X_{t-1}X_tX_{t+1}$ to output sequence $O_{t-1}O_tO_{t+1}$. We can see that at time step t, along with the input $X_t$, the context from earlier steps is passed to predict $O_t$.

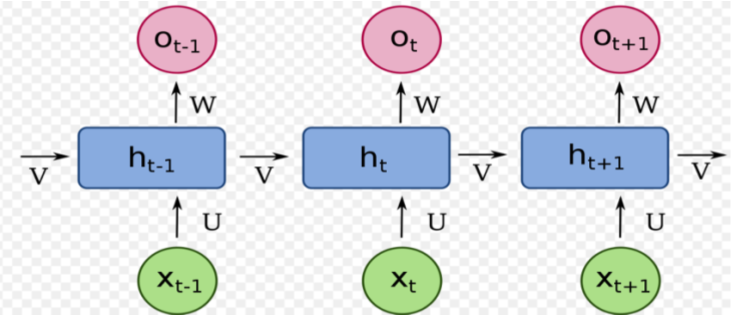
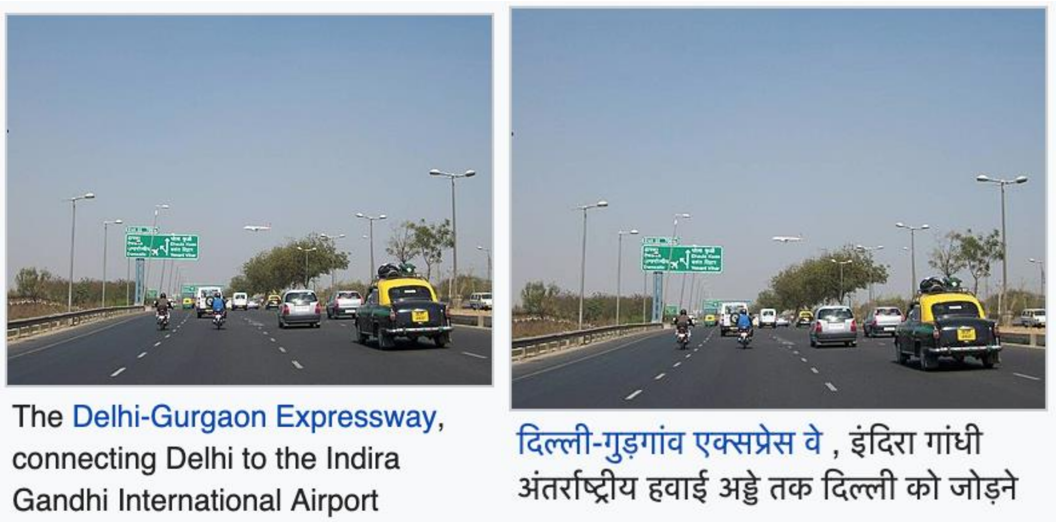


Figure 2. Sequence translation using RNN. [21]

One can take any picture from a website or capture the images with the help of camera and translate it to their required languages using various tools available currently. One of the widely used software to perform this task is Google translate [22] which performs the translation of the text contained in images by extracting the text and then performing the translation. It does not take into account the information given by the pictures. As a part of this project, we plan to explore if incorporating the context of the image in the machine learning model will improve the accuracy.



The Delhi-Gurgaon Expressway, connecting Delhi to the Indira Gandhi International Airport

दिल्ली-गुड़गांव एक्सप्रेस वे , इंदिरा गांधी अंतर्राष्ट्रीय हवाई अड्डे तक दिल्ली को जोड़ने

Figure 3. [23] **Textual information**: the delhi-gurgaon expressway, connecting delhi to the Indira gandhi international airport
**Contextual information:** road, motor vehicle, lane, asphalt, highway, transport, mode of transport, infrastructure, thoroughfare, sky

In Figure 3, the textual information is the actual text that is contained in the image. The contextual information is derived from the actions and figures present in the image. Both of these would be used together in developing a machine learning model for image localization.

Image-to-image translation by incorporating the context of the image can be useful in lot of scenarios. For example, if a tourist is travelling to a certain country and does not understand the local language, the model can be used for translation of various posters and sign boards. The model along with the text information can take the context of the as an input. The context in this scenario can be the current location of the traveler, the current time and current temperature. Apart from this, there is lot of web-based data which should be easily accessible by everyone in the world without language being a barrier for the growth of society. Even though English is the most widely spoken language in the world, there is a lot of population which does not English and would like the information to be available in their local languages. The journals and manuscripts which are not all translated to English yet also need to be translated so that they become readable by the historians and further research can be carried out on them.

Figure 4 shows the flowchart of the technique that was used for image localization for the thesis. As a part of this project, we explored all these steps and would be discussing these steps in detail in next sections.

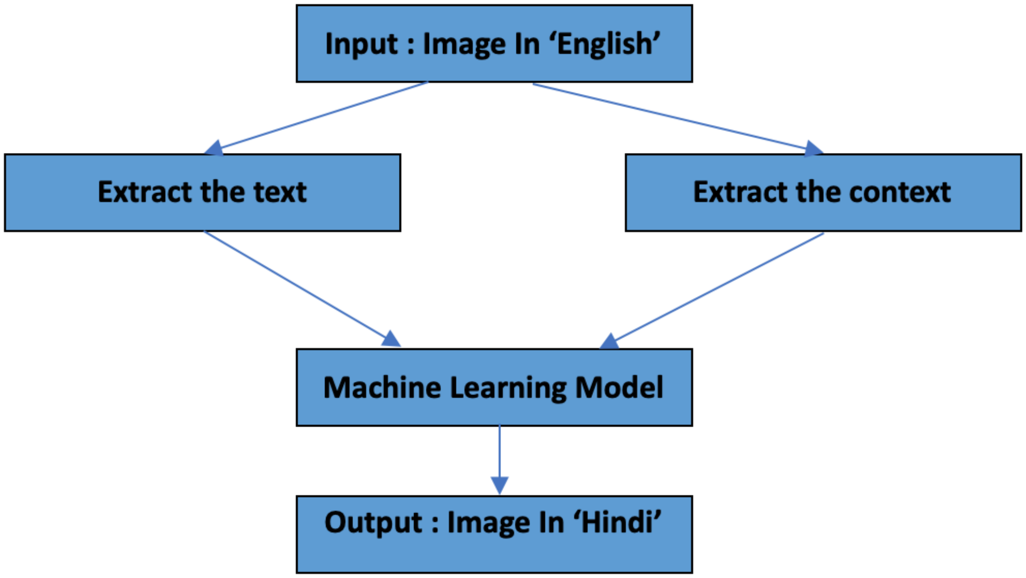


Figure 4. Flowchart of localization technique used for the project

Optical Character Recognition (OCR) has made it possible to extract the text from the images very efficiently. The earlier versions supported text extraction from PDF files and scanned paper documents but with the recent developments, the text can be extracted from complex images containing text in different fonts and sizes [24]. There are lot of image captioning tools, which can be used in generating the context of the image. The textual and contextual information gathered can together be used to train the model and generate the translated images.

# 2. BACKGROUND

In this section, we review the basics of how Neural Networks and RNNs work. Following this, details about how RNNs can be used for sequence to sequence translation are covered. This section also covers details about how text extraction and image captioning models work.

## 2.1. WHAT ARE NEURAL NETWORKS

Artificial Neural Networks (ANNs) work similar to the manner the way a neuron of a human brain works. ANN is also composed of neurons which learn with the training examples passed to them over a period of time. The information is stored in these neurons in the form of numbers and mathematical formulas which keep improving with each example passed while training the model. These neurons can later be used to make predictions on the future unseen data.

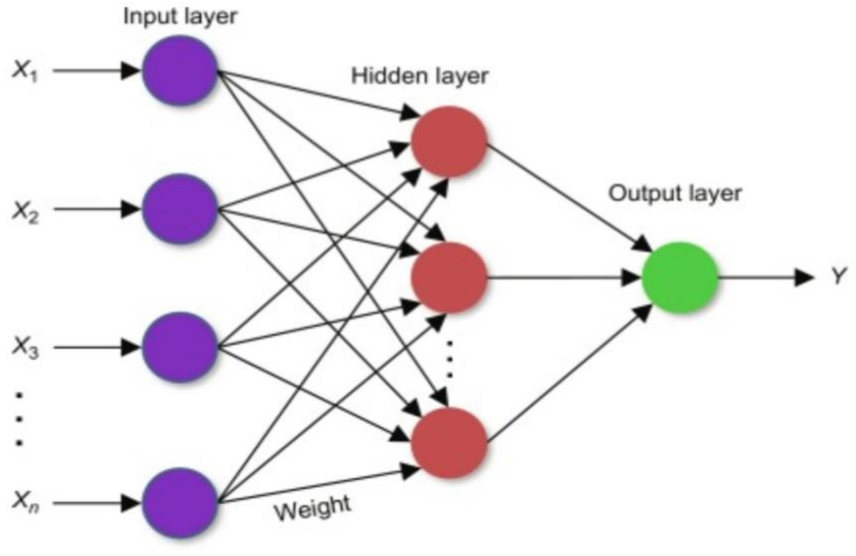


Figure 5. Artificial Neural Network

Figure 5 is an example of the structure of ANN with input layer, hidden layer and output layer stacked on the top of each other. Each layer has multiple neurons to learn different aspects of the training example. In a deep learning model, we might have hundreds of neurons with multiple layers.

Figure 6 below depicts a single neuron used to predict the output with weights, biases and inputs being passed to it. During the first run, random weights and biases are passed which keep improving with each successive example and execution of the model. As can be seen from the below figure, after multiplying the inputs by the weights, the term bias is added which is then passed to the activation function *f(S)* to make the predictions. The predictions can then be made based on the value returned by the activation function. For example, if the value returned is above 0.5, we might classify the object to be one category else other.

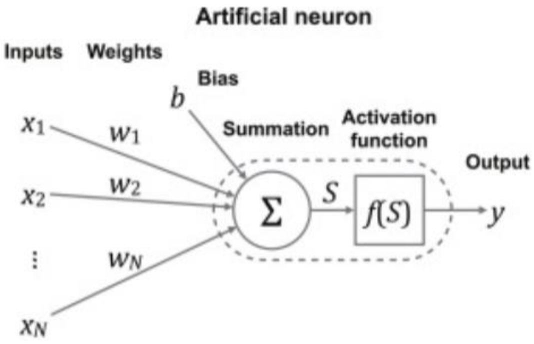


Figure 6. Structure of a single neuron

Below is the mathematical description of the Figure 6 to understand its working.

$S = x_1.w_1 + x_2.w_2 + \ldots. + x_n.w_n + b$

$x_1$, $x_2 \ldots x_n$ are inputs to the model

$w_1$, $w_2, \ldots w_N$ are weights learnt by the model b = bias (a constant number added to prevent underfitting)

f(S) = activation function.

The activation functions most widely used are sigmoid, tanh and relu as described below.

sigmoid(S) = 1 / 1 + e^-S

tanh(S) = 2/(1 + e^-2S) - 1

relu(S) = max(0,S)

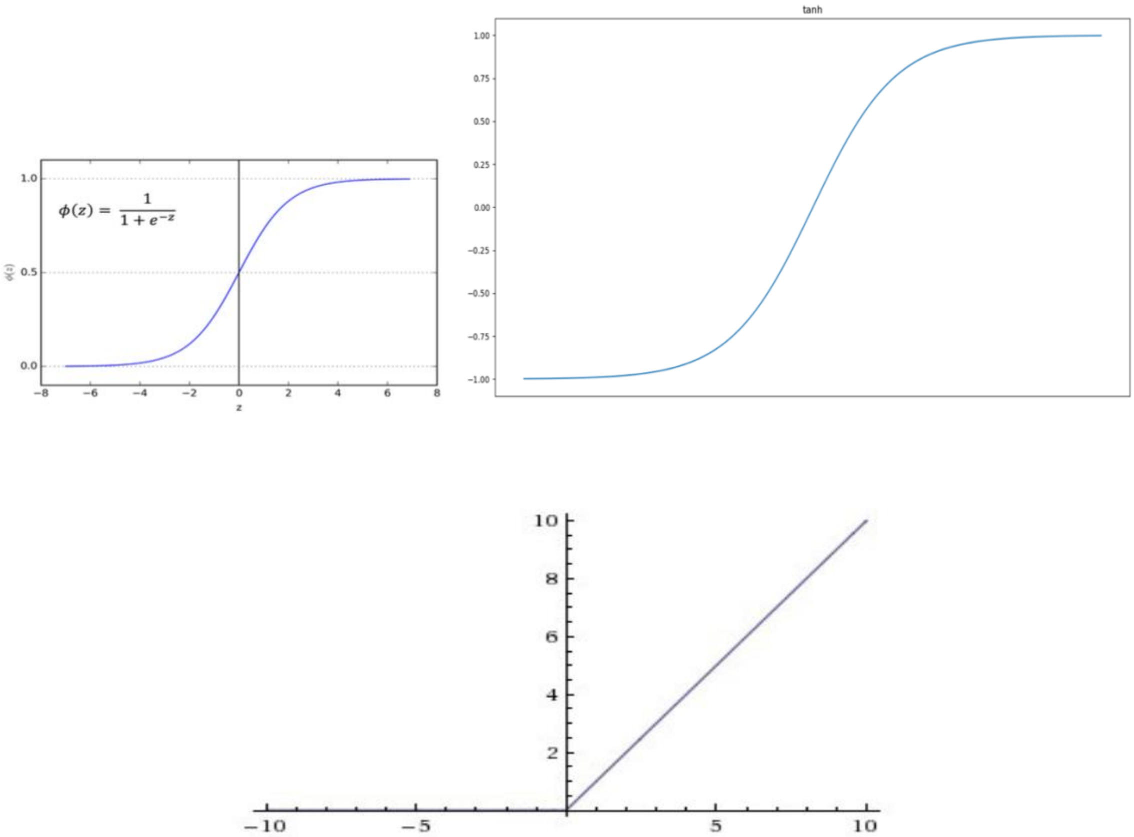


Figure 7. Graphs of activation functions (sigmoid, tanh and relu)

The neural networks learn with the magnitude of error done with each execution and adjust the weights and biases accordingly. The error is passed to the network to help the network adjust the parameters. This mechanism is also known as backpropagation.

## 2.2. RECURRENT NEURAL NETWORKS

There are various types of neural networks and each of them can be used based on the nature of the dataset and the type of predictions that need to be made. One such type is Recurrent Neural Network (RNN).

Human beings tend to learn new things on the top of the information and facts about the subject learnt so far. The previous knowledge is taken into consideration wherever it is possible to incorporate and not simply discarded. When a person watches a movie, the earlier events that have occurred in the movie are remembered by the human brain which helps in the understanding the rest of the movie. RNNs are also based on this underlying concept of storing the details of earlier events and making it easy to predict the future events. In the vanilla neural networks discussed in the previous section, all the inputs and outputs pairs are independent of each other.

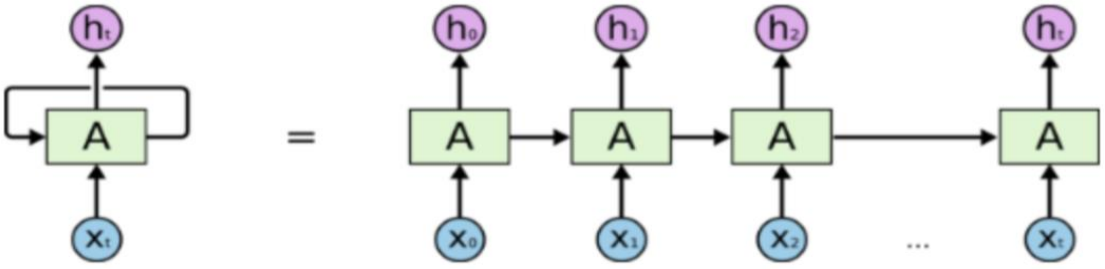


Figure 8. Expanded recurrent network [25]

Figure 8 is a portion of RNN with its expanded version on the right. $X_0X_1…X_t$ represent the inputs and $h_0h_1…h_t$ represent the predictions. The green box is RNN neuron ‘A’ having the capability to incorporate the context from previous timesteps. In the above example, we can see that at each time step, the information from previous time steps is used to make predictions in the current time step. In order to make the prediction $h_{t,}$ the context $h_{t-1}$ is passed in addition to the input $X_{t.}$ Below is the mathematical formula of the RNN neuron.

$$h_t = f(h_{t-1}, X_t)$$

*where f is an activation function like tanh, sigmoid, relu etc.*

RNNs are widely used for image captioning where the sequence of words needs to be generated from the pixels in the image [18][20]. They are also used in language translation in which sequence of words in one language are translated into sequence of words in other language. The language translation using RNN would be discussed in the next section. Some of the other applications of RNN are audio to text translation and generating subtitles for a video.

RNNs are effective in the prediction of short sequences in which lot of context need not be remembered. With large sequences, they suffer from vanishing gradient problem which occurs when small errors are multiplied large number of times eventually vanishing the error. This issue is handled with Long Term Short Memory (LSTM) neurons which selectively remember or forget the data storing only the relevant information in the cells [26].

## 2.3. SEQUENCE TO SEQUENCE PREDICTIONS USING RNNs

Sequence to sequence translations can be used in various applications like for question and answer portals and language translations. It involves generating a new array of words based on the received array of words. The length of input and output sequences can be different based on the problem being solved. We will be discussing the theory behind one such most commonly such model known has Encoder-Decoder RNN model [19]. These models have one Encoder layer and one Decoder layer. The Encoder layer learns the context of the input and passes it to the Decoder. The Decoder with the help of the context predicts the output sequence.

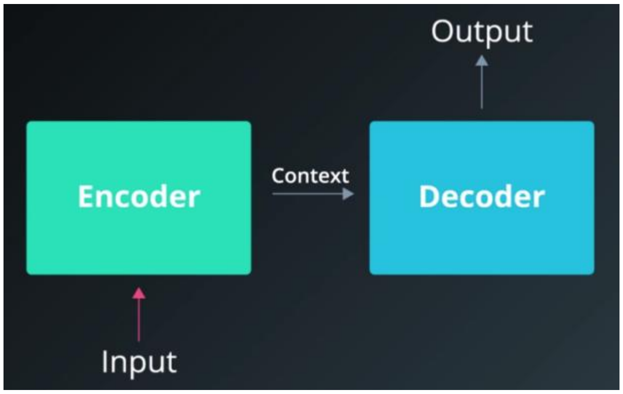


Figure 9. Encoder-Decoder RNN

Both Encoder and Decoder have recurrent cells to understand each word based on the words seen so far. Figure 10 shows the expanded version of the Encoder-Decoder RNN model. In the Encoder part of the model, at each time step the context of the previous word along with the current word are used to generate the next context. The first cell of the decoder takes the context generated by the encoder as input. Other cells of decoder take the context of previous words along with the previous output to predict the current output.

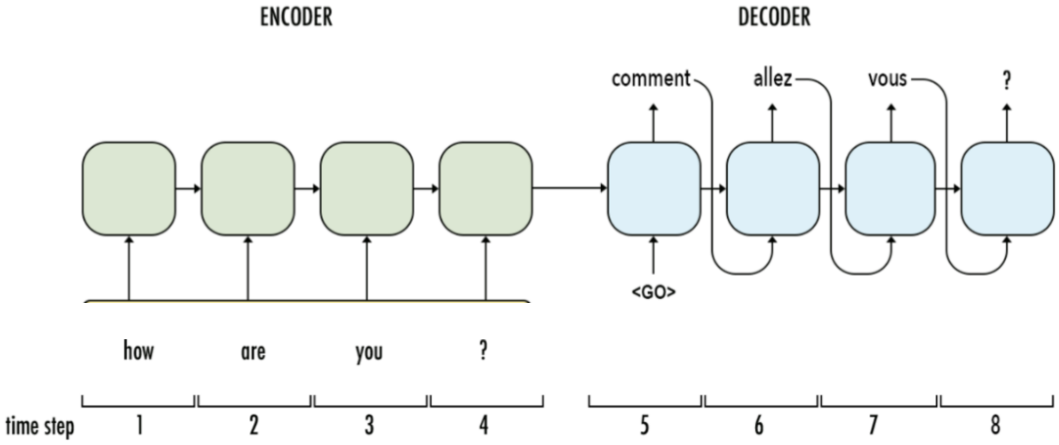


Figure 10. Expanded Encoder-Decoder RNN

We can also have Embedding layers in the Encoder-Decoder model to improve the accuracy of the model as shown in Figure 11. below. The embedding represents each word in the form of a vector of numbers.

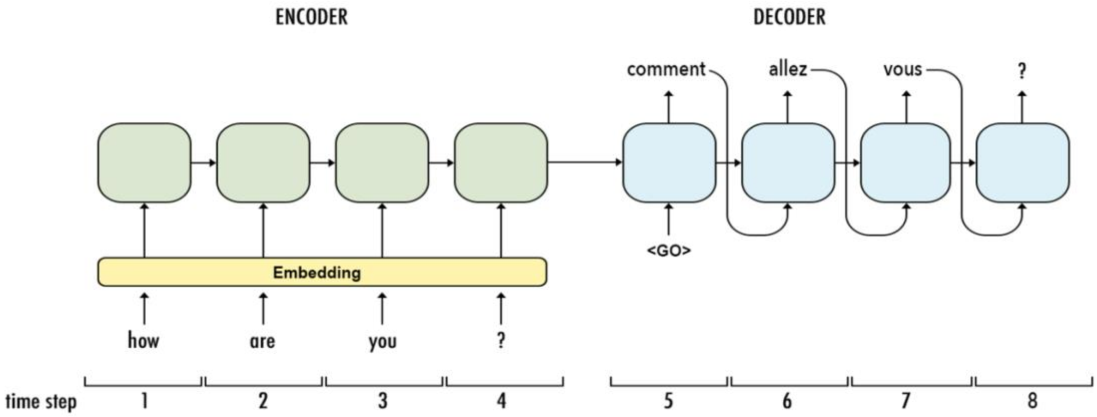


Figure 11. Expanded Encoder-Decoder RNN with Embedding Layer

For example, the word ‘car’ might be represented as ‘1.0 9.8 7.5 122.5 78 12 43 53 12.9’. The two words that have almost similar meanings would be close to each other in the vector space. For example, ‘Car’ and ‘Truck’, ‘California’ and ‘United States’ would be close to each other whereas ‘United States’ and ‘Car’ would be farther away as they are not related. Figure 12 shows the representation of the words in the vector space. Embedding layer is useful in sequence translations as it already has certain aspects of the word meanings learnt benefitting the model. The other advantage is that embedded vectors have lower dimensions as compared to other encodings like One Hot Encoding.

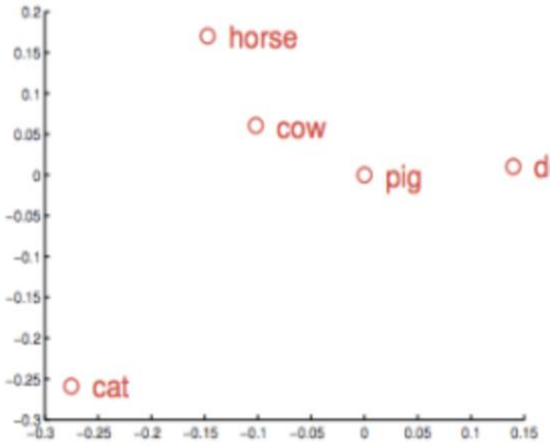
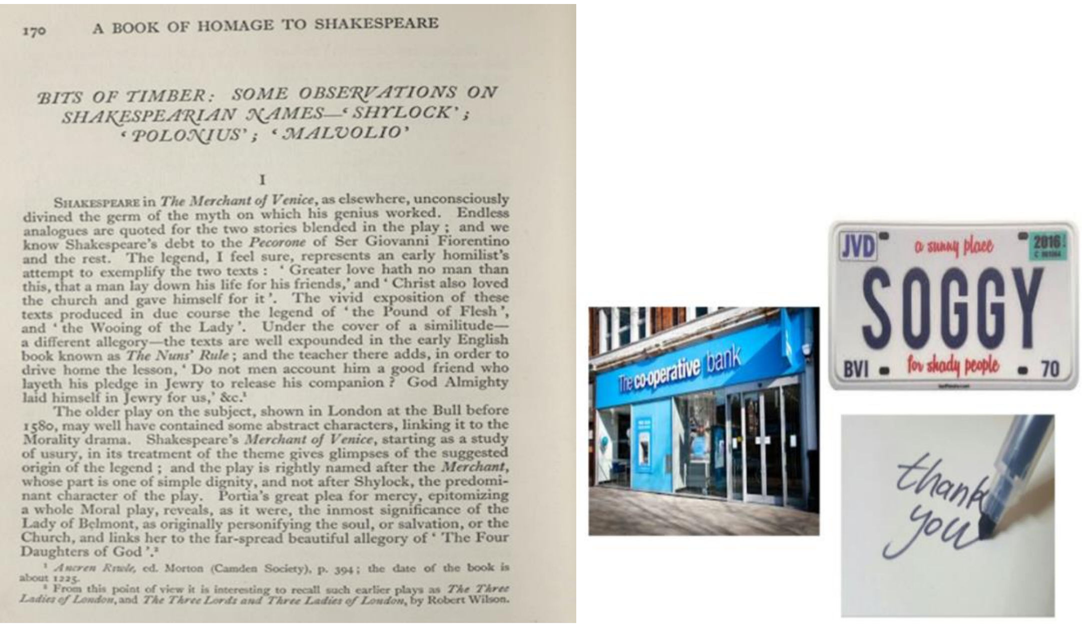


Figure 12. Words represented in Embedded vector space [27]

There are lot of pre-trained word embeddings already available which can be used as Embedding layers in the neural network. One can also self-train the embedding layer with the vocabulary of the input sequences. For this project, we used self-trained the embedding layer as the size of dataset was limited.

## 2.4. TEXT EXTRACTION

For this project, we need to extract the textual information from the images which would be needed for training the model. Text extraction from images has many useful applications in today's digital era. Everything is being digitalized ranging from manuscripts, books, zip codes, addresses, car number plates, and many more. The tasks of searching and editing in a digitalized data is very important for a user. Hence, there is an increasing need to develop models having high accuracy in detecting text from these images. There are primarily two types of images in which text might be present in the images, structured and unstructured. As shown in Figure 13, the structured images have uniformly formatted text with one font and consistent spacing between the words in one direction. The images of pages from text books is one such example. The unstructured images might have text embedded in scenes with varying sizes, fonts and colors.



170 A BOOK OF HOMAGE TO SHAKESPEARE

*BITS OF TIMBER: SOME OBSERVATIONS ON SHAKESPEARIAN NAMES—'SHYLOCK'; 'POLONIUS'; 'MALVOLIO'*

I

SHAKESPEARE in *The Merchant of Venice*, as elsewhere, unconsciously divined the germ of the myth on which his genius worked. Endless analogues are quoted for the two stories blended in the play; and we know Shakespeare's debt to the *Pecorone* of Ser Giovanni Fiorentino and the rest. The legend, I feel sure, represents an early homilist's attempt to exemplify the two texts: 'Greater love hath no man than this, that a man lay down his life for his friends,' and 'Christ also loved the church and gave himself for it'. The vivid exposition of these texts produced in due course the legend of 'the Pound of Flesh', and 'the Wooing of the Lady'. Under the cover of a similitude—a different allegory—the texts are well expounded in the early English book known as *The Nuns' Rule*; and the teacher there adds, in order to drive home the lesson, 'Do not men account him a good friend who layeth his pledge in Jewry to release his companion? God Almighty laid himself in Jewry for us,' &c.[1]

The older play on the subject, shown in London at the Bull before 1580, may well have contained some abstract characters, linking it to the Morality drama. Shakespeare's *Merchant of Venice*, starting as a study of usury, in its treatment of the theme gives glimpses of the suggested origin of the legend; and the play is rightly named after the *Merchant*, whose part is one of simple dignity, and not after Shylock, the predominant character of the play. Portia's great plea for mercy, epitomizing a whole Moral play, reveals, as it were, the inmost significance of the Lady of Belmont, as originally personifying the soul, or salvation, or the Church, and links her to the far-spread beautiful allegory of 'The Four Daughters of God'.[2]

[1] *Ancren Riwle*, ed. Morton (Camden Society), p. 394; the date of the book is about 1225.

[2] From this point of view it is interesting to recall such earlier plays as *The Three Ladies of London*, and *The Three Lords and Three Ladies of London*, by Robert Wilson.

Figure 13. Structured and Unstructured Data

Traditional techniques like Optical Character Recognition (OCR) are accurate only for structured data. Tesseract [28] was also developed for detecting the text in structured images but, later versions support complicated images as well. This was made possible by incorporating machine learning in the Tesseract application. Extracting text from unstructured data is more challenging for which machine learning models need to be incorporated in the traditional OCRs.

In order to extract text from images, the image should first be preprocessed before further processing. The preprocessing steps might include steps like removing the noise from the images and converting the image to black and white. After preprocessing, first the textual region needs to be detected and then the text needs to be extracted from the regions detected. These steps are known as text detection and text recognition respectively. In [2], the article discusses Text Detection using sliding window technique in which windows of different sizes are passed through the image to detect candidate textual regions. This might be a computationally intensive process as same pixel in the image is processed multiple times. Xinyu et al. [7] proposed a more robust technique known as Efficient Accurate Scene Text (EAST) detector. This technique can detect the text even in highly complicated images in both, horizontal and vertical directions. The EAST detector passes the image through Convolutional Neural Networks (CNN) and detects the textual features which is then passed through regression classifiers which detect the possibility of the text based on the features. After text detection, the Non-Suppression Maxima (NMS) algorithm is applied to filter out the best possible region containing maximum features amongst the suggested regions as shown in Figure 14.

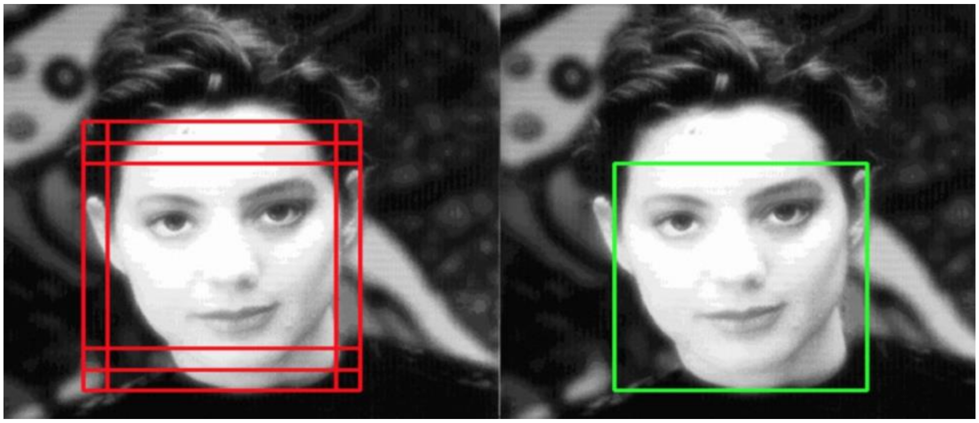


Figure 14. Non-Suppression Maxima detects best candidate region

After text detection, the next step is to detect the text. In [3], the image is converted into a 4096-sized feature vector using CNN and then passed to RNN. The RNNs take this vector as an input and predict the text.

## 2.5. IMAGE CAPTIONING

For this project, we need to extract the context of images to train the image localization models. The context can be extracted using image captioning techniques which involves the process of understanding an image and generating labels for it. In Figure 15 below, the caption of the image can be “White dog sitting in grassy area”.

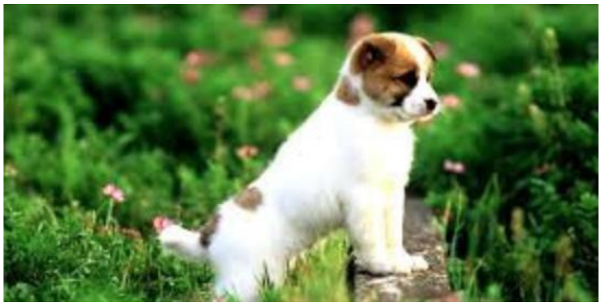


Figure 15. Image Caption: “White dog sitting in grassy area”

Image captioning is typically done using machine learning models involving computer vision, deep learning and natural language processing. These models take an image as an input and generate the caption for that image as an output. The objects, faces, colors, text, and actions in the image are used to train the model. Generating labels for the images has many useful applications. It can be used in self driving cars to understand the surroundings, for blind people to make them aware of their surroundings, and by search tools to generate similar images based on the input image having similar context.

There are many tools and software available for image captioning, viz., Google Cloud Vision [29] and Microsoft Bot [30]. The following are the basic steps used for building an image captioning model in an appropriate way based on [18] and [20].

1. *Building the vocabulary of all the words in the captions in the training data:* All the words in the training data are collected and each word in the vocabulary is assigned a unique integer index. These integer indexes are more useful in training the model rather than using words as the model is usually based on mathematical equations.

2. *Generation of Feature Vector*: CNN model is used as an encoder to generate the feature vector. As shown in Figure 16, the images need to be converted to vectors of numbers which can be used as features for training the model. CNN are useful for image processing which have different convolution matrices detecting different aspects of the image. These different aspects of the image in the form of vector would be passed on to the next layers of our model.

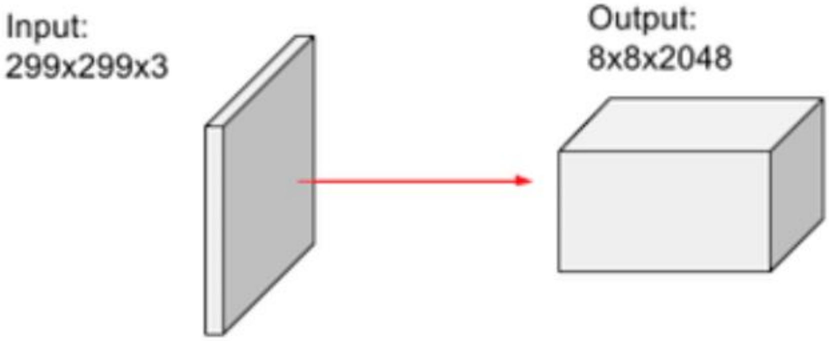
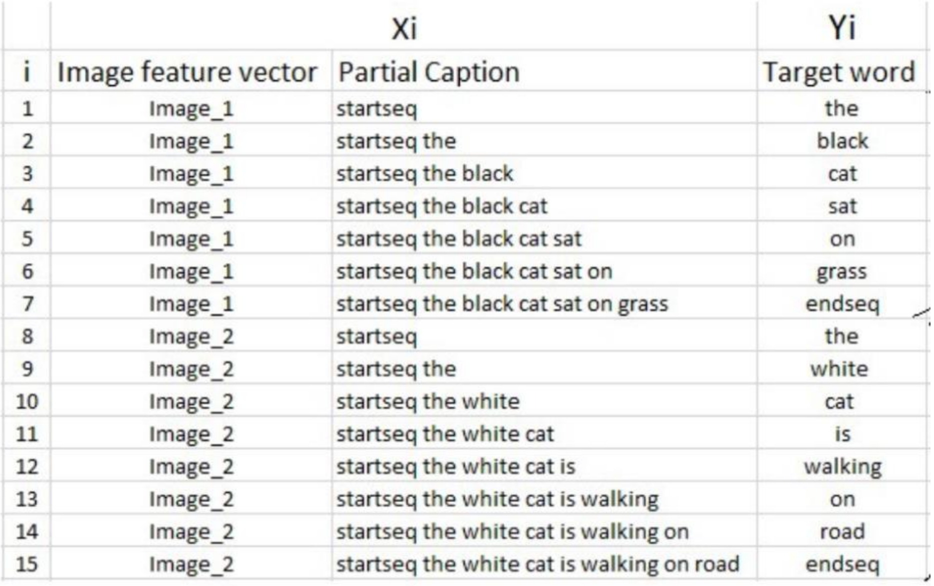


Figure 16. Generating feature vector of an image

3. *Training the model*: The feature vector along with the caption sequence is used for training the model. At each timestep, only a part of sequence is passed as an input. As shown in Figure 17, length of the partial sequence keeps increasing with each prediction of the word. The model after getting trained gives the probability distribution over all the words in the vocabulary. The word with the maximum probability is chosen to be the candidate word in the sequence. This keeps happening till we reach the end of sequence ‘endseq’.



| | Xi | | Yi |
|---|---|---|---|
| i | Image feature vector | Partial Caption | Target word |
| 1 | Image_1 | startseq | the |
| 2 | Image_1 | startseq the | black |
| 3 | Image_1 | startseq the black | cat |
| 4 | Image_1 | startseq the black cat | sat |
| 5 | Image_1 | startseq the black cat sat | on |
| 6 | Image_1 | startseq the black cat sat on | grass |
| 7 | Image_1 | startseq the black cat sat on grass | endseq |
| 8 | Image_2 | startseq | the |
| 9 | Image_2 | startseq the | white |
| 10 | Image_2 | startseq the white | cat |
| 11 | Image_2 | startseq the white cat | is |
| 12 | Image_2 | startseq the white cat is | walking |
| 13 | Image_2 | startseq the white cat is walking | on |
| 14 | Image_2 | startseq the white cat is walking on | road |
| 15 | Image_2 | startseq the white cat is walking on road | endseq |

Figure 17. Generation of next word in sequence.

Figure 18 below shows the architecture of the model discussed so far. Since, the image captioning involves sequences, we would be using RNN to understand the context of captions. The partial captions along with the feature vector are passed to the model generating next word in the sequence.

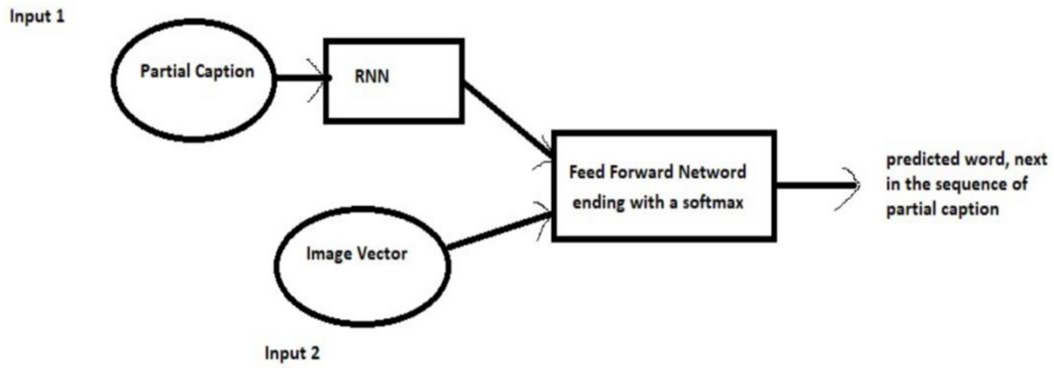

Input 1
Partial Caption
RNN
Feed Forward Netword ending with a softmax
predicted word, next in the sequence of partial caption
Image Vector
Input 2

Figure 18. Image Captioning Model Architecture.

The model can be further enhanced by adding embedding layers to understand the context of the sequences even better by translating these words in higher dimension vectors.  Since, the size of the data points may be huge exceeding the memory limits of the machine on which the model is being trained, the dataset is usually processed in batches.

# 4. DESIGN AND IMPLEMENTATION

In this section, I discuss the tools and environment that I used for this project. After that, I cover details of the dataset and the model created for solving the problem of image localization. I also discuss the various parameters that were used to train the model in the best possible way with the resources available.

## 3.1. ENVIRONMENT USED

I used Python as the programming language for this project. The reason being that Python is a flexible language with lot of deep learning libraries simplifying the job of developing machine learning applications. There is also lot of online support available to build the deep learning infrastructure using Python. One can focus on developing the machine learning model aspect rather than other technical details. The version of Python that was used for this project was 3.6.8. Python also provides lot of libraries for writing the machine learning like Keras and Scikit-learn. I used Keras which uses TensorFlow as backend. The Keras version used for this project was 2.2.5 which used TensorFlow version 1.15.0. Keras is appropriate for writing neural networks applications with rich set of APIs available. The editor used was Google Colaboratory [31] which has an online interface with most of the machine learning libraries pre-installed requiring almost no setup. The code is executed on different virtual machines with choice of Central Processing Unit (CPU), Graphics Processing Unit (GPU) and Tensor Processing Unit (TPU) giving access to more processing power and can be used when lot of computations are needed especially for neural networks with a large sized dataset containing images and video. Google

Colaboratory also gives the option to access datasets that are present on Google Drive saving the space on local machines and having everything required for the machine learning application to be online.

I also used various Google Cloud Vision Representational State Transfer (REST) APIs [29] for the experiments. The APIs give very accurate results as the pre-trained models used by them have been built with a very large sized dataset. They provide rich set of features with several options and can be imported to the application being developed. The APIs were used to extract the text from the images, translate the text from one language to other language and generating the context of the images. The text extraction API provides the option to detect the text in both structured and unstructured images with different set of commands. The language of the text is also figured out by the API itself without the need for the programmer to explicitly specify. Google Cloud Translation [22] libraries provide the functionality to translate text between various language pairs. These APIs can be used in lot of ways like integrating with the browsers or used by the programmers to build their applications. The user of the API can provide the target language to which the text needs to be translated to. Apart from text extraction and text translation, I also used the APIs for detecting the context of the images. The APIs with the help of pre-trained models can detect the various objects present in the images and generate the captions for them. All these APIs provide free usage for a fixed number of predictions above which the pricing is per prediction on using each of the APIs.

## 3.2. DATASET

The dataset required for this project was pair of images in English and corresponding translated image in Hindi. These images can be collected from various resources like different websites that have English and Hindi versions. Online resources like parliamentary websites of India have web pages both in English and Hindi for users with knowledge of either of the languages. Wikipedia also has several pages that have both English and Hindi translated versions. The images can also be collected through other mediums like taking the photos of different sign boards and posters in the streets. I experimented with different ideas to collect the dataset for this project. One of the approaches was to crawl the website using various scripts and take snapshots of Hindi and English versions of the web page. This method had various challenges and could not be used to collect the dataset. The reason being various pages do not have the web pages translated in Hindi and also it is difficult to predict the size of snapshot for English and Hindi versions as one of the languages might contain more information than the other within same size due to difference in the size of the characters and words. In Figure 19, we can see that English version takes more than half of the line whereas Hindi version takes less than half.

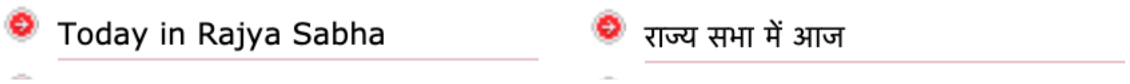
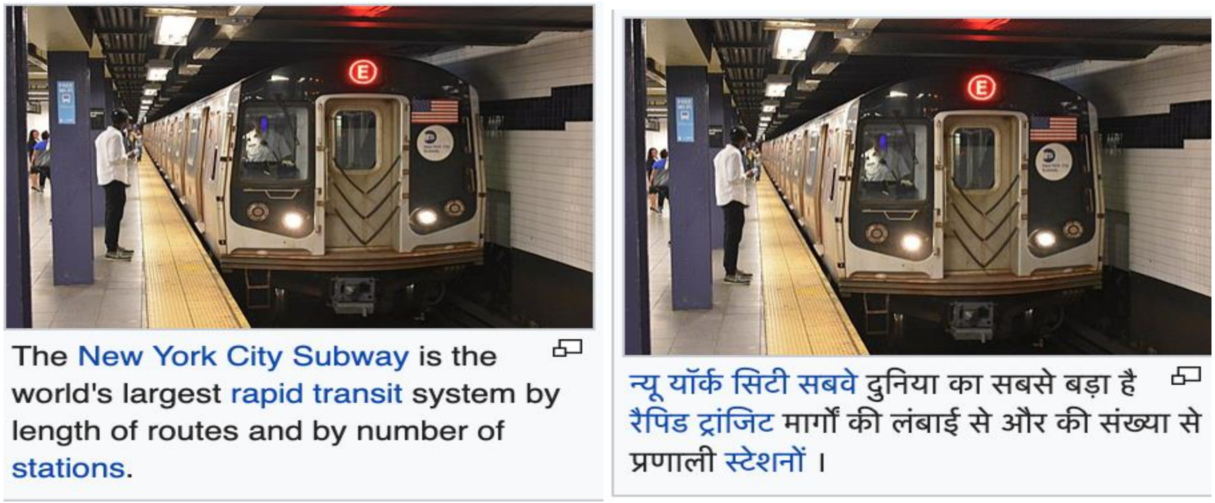


Figure 19. English and Hindi consume different fractions of line.

Due to these challenges, I took the snapshots of different pages on Wikipedia manually with help from other friends and generated the dataset of 10,000 images. Figure 20 shows a sample data point of the dataset used for training the model.



The New York City Subway is the world's largest rapid transit system by length of routes and by number of stations.

न्यू यॉर्क सिटी सबवे दुनिया का सबसे बड़ा है रैपिड ट्रांजिट मार्गों की लंबाई से और की संख्या से प्रणाली स्टेशनों ।

Figure 20. Sample dataset of English and corresponding translated Hindi image.

## 3.3. FLOWCHART OF TECHNIQUE

In order to build the machine learning model, I extracted the relevant information needed for training the model using the Google REST APIs. As can be seen in Figure 21, the context of the images is being used to train the model.

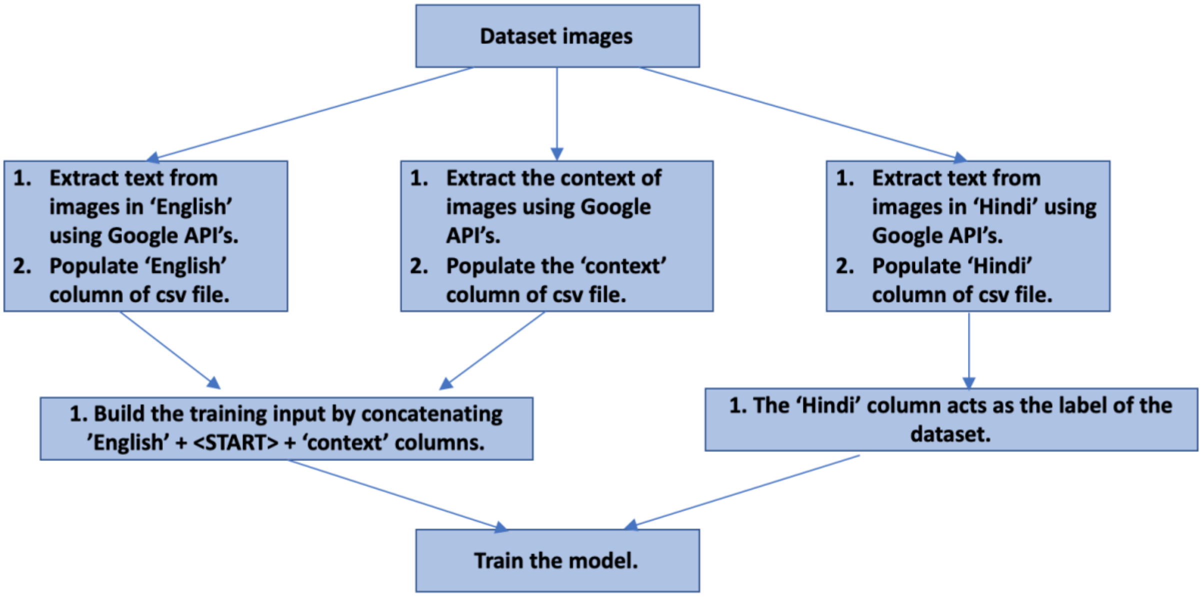

Dataset images
1. Extract text from images in 'English' using Google API's.
2. Populate 'English' column of csv file.
1. Extract the context of images using Google API's.
2. Populate the 'context' column of csv file.
1. Extract text from images in 'Hindi' using Google API's.
2. Populate 'Hindi' column of csv file.
1. Build the training input by concatenating 'English' + <START> + 'context' columns.
1. The 'Hindi' column acts as the label of the dataset.
Train the model.

Figure 21. Model training using the 'context' of the images.

The trained model with context can be used for image localization as shown in Figure 22.

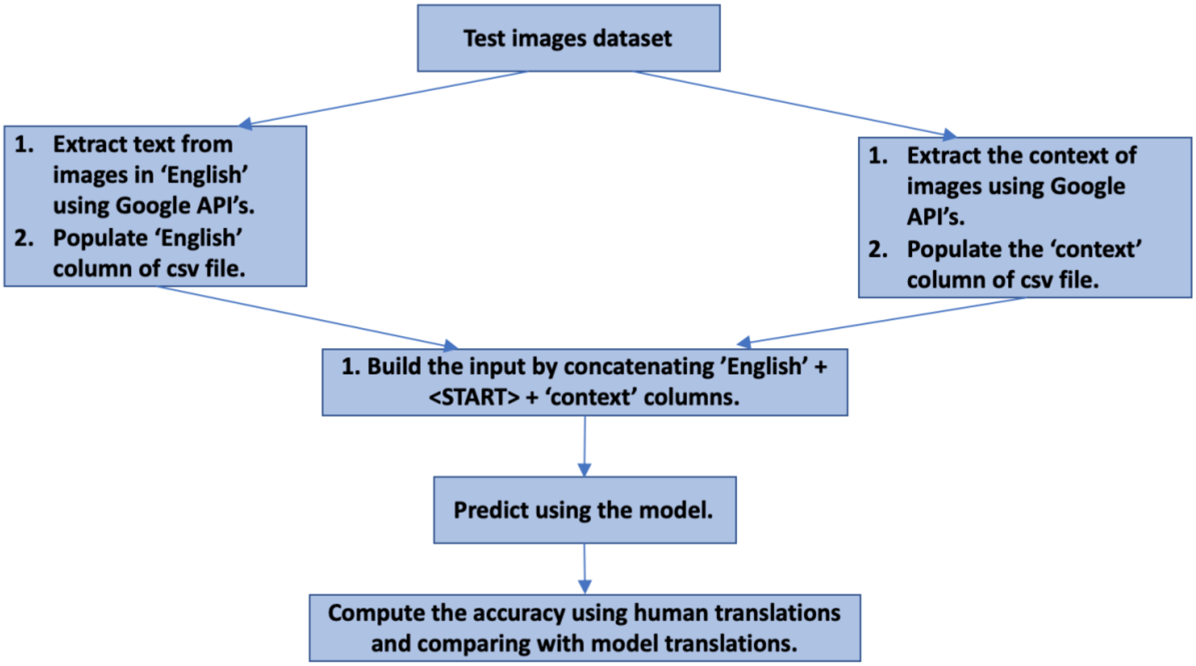

Test images dataset
1. Extract text from images in 'English' using Google API's.
2. Populate 'English' column of csv file.
1. Extract the context of images using Google API's.
2. Populate the 'context' column of csv file.
1. Build the input by concatenating 'English' + <START> + 'context' columns.
Predict using the model.
Compute the accuracy using human translations and comparing with model translations.

Figure 22. Predictions using the model trained with 'context'.

I also trained the model without using the ‘context’ of images to compare the accuracies of the models with and without context. Figure 23 below shows the flowchart depicting the training of the model without using the context and Figure 24 shows the flowchart for image localization using this trained model.

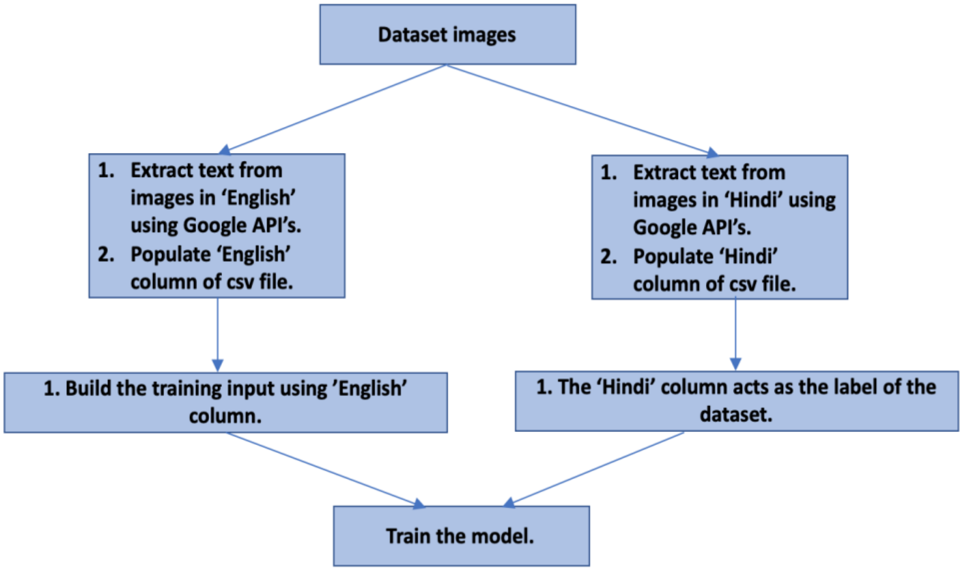

Dataset images
1. Extract text from images in ‘English’ using Google API’s.
2. Populate ‘English’ column of csv file.
1. Extract text from images in ‘Hindi’ using Google API’s.
2. Populate ‘Hindi’ column of csv file.
1. Build the training input using ’English’ column.
1. The ‘Hindi’ column acts as the label of the dataset.
Train the model.

Figure 23. Model training without using the ‘context’ of the images.

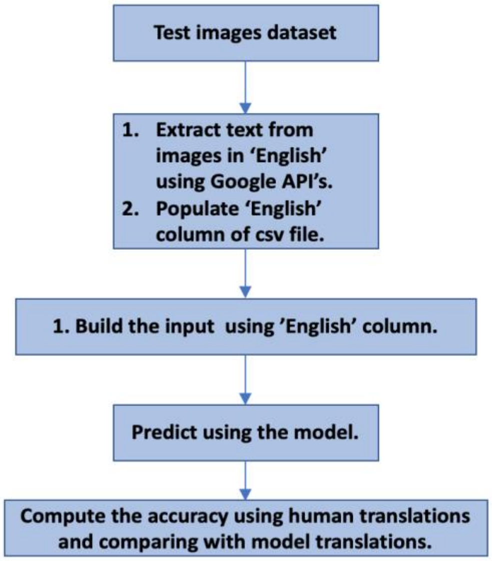

Test images dataset
1. Extract text from images in ‘English’ using Google API’s.
2. Populate ‘English’ column of csv file.
1. Build the input using ’English’ column.
Predict using the model.
Compute the accuracy using human translations and comparing with model translations.

Figure 24. Predictions using the model trained without ‘context’.

## 3.4. DATA VISUALIZATION

Below are some summary statistics for the dataset used in training the model.

```
3484 English words.
1855 unique English words.
10 Most common words in the English dataset:
"the" "of" "in" "a" "and" "india" "on" "at" "with" "by"

3762 Hindi words.
2010 unique Hindi words.
10 Most common words in the Hindi dataset:
"में" "के" "एक" "की" "का" "," "पर" "और" "भारत" "है।"
```

This information has been collected from the extracted text from the images.

## 3.5. DATA PROCESSING

The following steps were followed for processing the data.

1. Convert the sequences of the dataset containing English text and the image context to lower case.
2. Count total and unique words in the dataset for both Hindi and English versions.
3. Tokenize the words by assigning a unique number to all the words. This helps in making the training of deep learning model efficient by converting the data into a simplified format.
4. Pad the sequences with number zero to make all the inputs of equal size. Equal sized inputs make matrix multiplications simpler for the training of neural network.
5. Translate the result back to words from integers while predicting the translations with the trained model. Zero is translated to an empty character, ‘’.

## 3.6. MODEL ARCHITECTURE

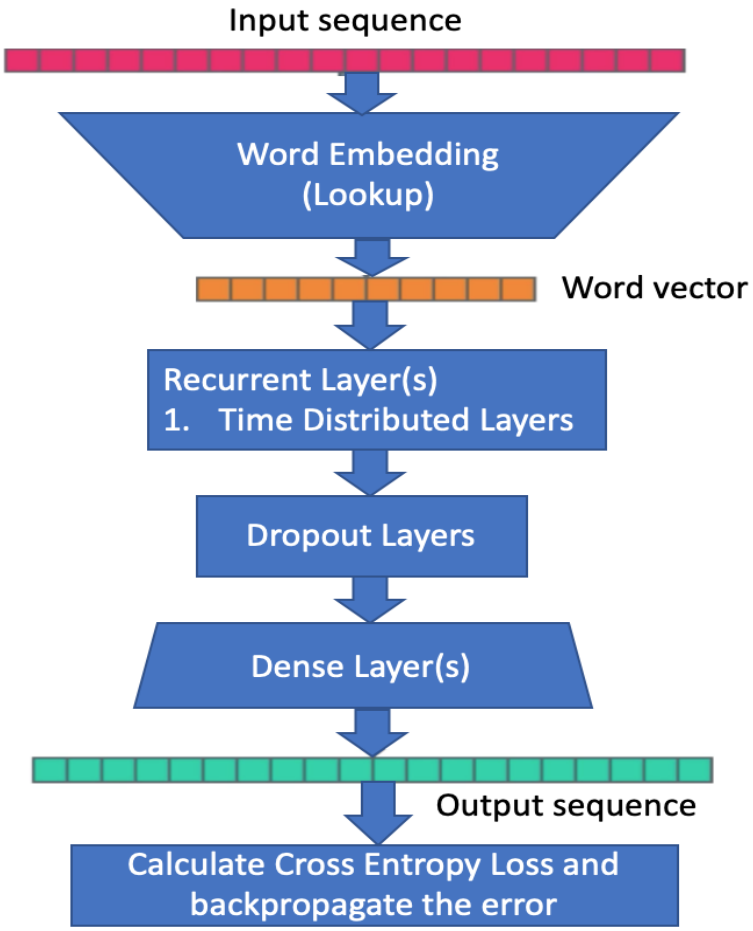


Figure 25. Model architecture for image localization.

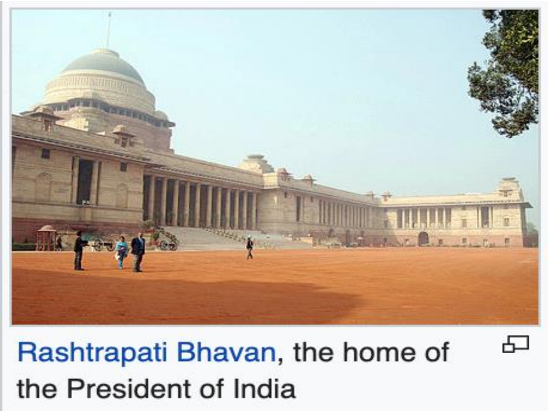
As shown in Figure 25, the model has input layers, embedding layers, recurrent layers followed by dense layers and output layers. The input layers take the input which are image text and context information after the data preprocessing step discussed above. The embedding layers stacked on the top of input layers capture the similarity between the words by converting them into an array of numbers. For this project, we are training the Embedding layers with the vocabulary of our input instead of using pre trained embedding matrices like GloVe. The recurrent layers function as an encoder capturing information at each time step which can be used in future time steps to understand the context. The context is then passed to dense layers which play the role of decoder and predict the output sequence which for this project would be the translated text in Hindi language. The time distributed layers play the role of recurrent layers by capturing the information at each timestep and passing it to future timesteps. In addition to recurrent layers, we have also used Gated Recurrent Units (GRUs) [32] in our project which selectively captures the relevant contextual information instead of keeping all the information which standard recurrent cells do. In order to avoid overfitting of the data, we have added dropout layers to the neural network. These layers randomly drop the nodes in the network at each iteration with the factor specified. We have used a softmax activation function in the output layers. The softmax function gives the probability for each possible word at each timestep from which the prediction with maximum probability is chosen. A ReLU activation function was used for each hidden layer. The loss function used in the network was ‘sparse_categorical_entropy’ loss function. Cross entropy losses are used when output of the neural network nodes is a probability distribution for each category. A Categorical Cross Entropy loss function is specifically used in scenarios when the output is one hot encoded. A Sparse Categorical Cross Entropy is often used when the output are certain fixed numbers instead of one hot encodings. These are generally used in combination with softmax activation function. The optimization function we used was the ‘Adam Optimizer’ [33] in which the learning rate of each parameter is different and updated separately as the learning progresses. In Stochastic Gradient Descent [33], a single learning rate is maintained for the entire training. With the experiments that I performed, the best results were obtained with 2000 epochs and a learning rate of 0.001. The training of the model completed in around 18 hours on the dataset of 10,000 images.

# EXPERIMENTAL RESULTS

In this section, the results obtained with the trained models are discussed. We also describe how the model trained with the image context performs as compared to the model trained without the image context. We first look over some individual predictions before giving the overall statistics of the model accuracy.

*TEST DATA 1:*

For the image in Figure 26. below, the model with context predicted, “ भारत के राष्ट्रपति का घर राष्ट्रपति भवन ” which is correct as verified by human translators.



Rashtrapati Bhavan, the home of the President of India

Figure 26. Test data image in English.

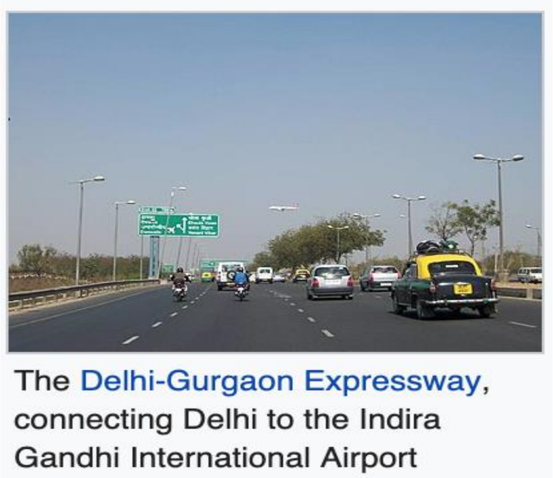
The input text and the context of the image passed to the model were as below.

*Input Text:* rashtrapati bhavan, the home of the president of india

*Image Context:* landmark, holy places, architecture, tourist attraction, historic site, building, adaptation, tourism, stock photography, photography

*TEST DATA 2:*

For the image in Figure 27. below the model with context predicted, “दिल्ली-गुड़गांव एक्सप्रेस वे , इंदिरा गांधी अंतर्राष्ट्रीय हवाई अड्डे तक दिल्ली को जोड़ने ” which is correct as verified by human translators.



The Delhi-Gurgaon Expressway, connecting Delhi to the Indira Gandhi International Airport

Figure 27. Test data image in English.

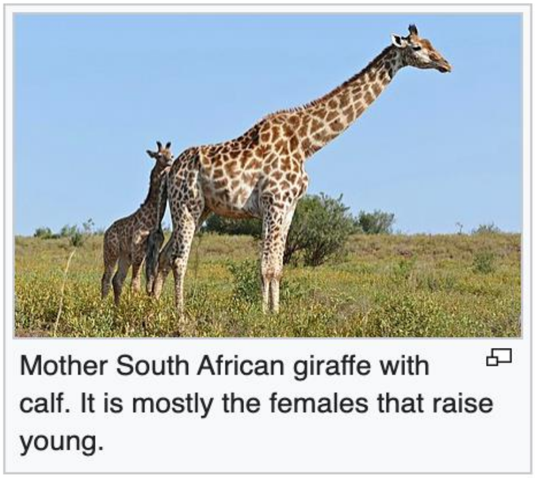
The input text and the context of the image passed to the model were as below.

*Input Text:* the delhi-gurgaon expressway, connecting delhi to the Indira gandhi international airport

*Image Context:* road, motor vehicle, lane, asphalt, highway, transport, mode of transport, infrastructure, thoroughfare, sky

For the experiments, I ran the model both with and without context for comparing the accuracies. The model trained with the image context seems to be performing better than with the model trained without the image context as can be seen through below examples.

*TEST DATA 3:*



Mother South African giraffe with calf. It is mostly the females that raise young.

Figure 28. Test data image in English.

*Input Text:* mother south african giraffe with calf. it is mostly the females that raise young.

*Image Context:* giraffe, terrestrial animal, wildlife, giraffidae, vertebrate, nature reserve, grassland, adaptation, natural environment, ecoregion

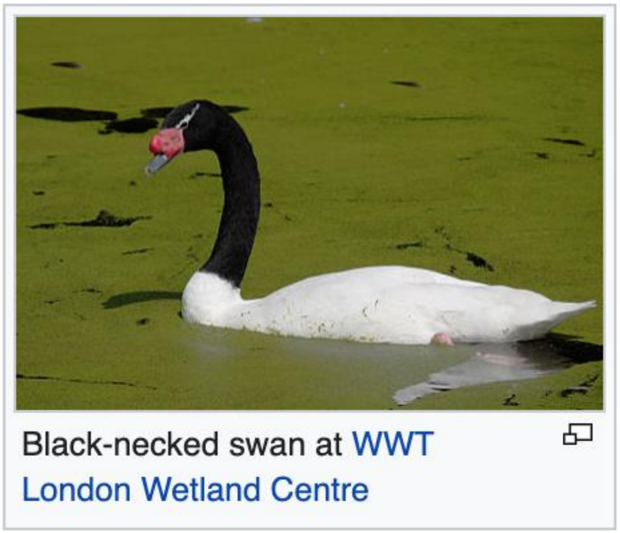
*Correct Translation:* बछड़े के साथ मदर साउथ अफ्रीकन जिराफ। यह ज्यादातर महिलाएं हैं जो युवा होती हैं।

*Prediction with The Image Context:*
माँ क्षेत्र जिराफ जिराफ़ साथ साथ 21 के लिए लॉर्ड और के बेचे से विभाजित

*Prediction without the Image Context:*
की अनुसंधान का भारत के साथ जयते सलाहकार नक्शे स्नान के के लिए में जाती है।

Here the predictions are incorrect for both, with and without using the image context. But, when predicted with the model trained using context, the model is able to partially translate the sequence (translates the Giraffes) whereas in the model trained without the context all the translated words are incorrect.

*TEST DATA 4:*



Black-necked swan at WWT London Wetland Centre

Figure 29. Test data image in English.

*Input Text:* black-necked swan at wwt london wetland centre

*Image Context:* bird, vertebrate, swan, ducks, geese and swans, waterfowl, beak, duck

*Correct Translation:* WWT लंदन वेटलैंड सेंटर में काले गले वाला हंस

*Prediction with The Image Context:* लंदन वेटलैंड सेंटर में काले गले वाला हंस

*Prediction without the Image Context:* लंदन वेटलैंड सेंटर में गले वाला हंस

In this example, the prediction with the model trained without the context does not translate the black color as specified in input text.

Table 3 below shows the comparison of model accuracy with and without taking the image context into account. The percentage of correct translations are higher when image context is being taken into account. The partial predictions are considerably better with the context where the model translates parts of text sequences.

| | CONTEXT | WITHOUT CONTEXT |
|---|---|---|
| % CORRECT | 70 | 66 |
| % PARTIALLY CORRECT | 25 | 13 |
| % INCORRECT | 5 | 21 |

Table 3. Accuracy statistics with and without using the image context

# CONCLUSION AND FUTURE WORK

In this project, I explored the image localization techniques by passing the context of the image to the machine learning model. Since we are working with sequences, the model was based on RNNs. We trained the models both with and without the image context to compare the accuracy. The model gives 4% better accuracy for translating the images when context is taken into consideration. In case of incorrect translations, the model trained with the image context is able to make partial translations 12% better as compared to the model trained without the image context with our test dataset. Accurate models for image localization can be used in lot of applications like in the localization of old manuscripts where current models do not take the image context into consideration [34]. We can further enhance the accuracy by training the model with an increased dataset. The increase in the number of epochs and experimenting with more learning rates can lead to even more accurate model. The model can be further fine tuned by experimenting with different layers like LSTM, Bidirectional layers. The models trained with different parameters can be compared and the appropriate model can be used based on the nature of the dataset and computing resources available.

# REFERENCES

[1] J.Y. Zhu, et al., “Unpaired Image-to-image Translation sing Cycle-Consistent Adversarial Networks,” in IEEE Int. Conf. on Comp. Vision, 2017. doi: 10.1109/ICCV.2017.244.

[2] R. Agarwal, Deep Learning Based OCR for Text in the Wild, https://nanonets.com/blog/deep-learning-ocr/

[3] B. Shi, X. Bai and C. Yao, “An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition”.
doi: arXiv:1507.05717v1

[4] S. Saini and V. Sahula, “A Survey of Machine Translation Techniques and Systems for Indian Languages,” in IEEE Int. Conf. on Comp. Int. & Comm. Tech., 2015.

[5] H.A. Driss, S. ELFKIHI and A. Jilbab, “Features Extraction for Text Detection and Localization,” in 5 th Int. Symp. On I/IV Comm. And Mobile Network, 2010.

[6] C.M. Thillou and B. Gosselin, Natural Scene Understanding, https://www.tcts.fpms.ac.be/publications/regpapers/2007/VS_cmtbg2007.pdf

[7] X. Zhou, et al., “EAST: An Efficient and Accurate Scene Text Detector,” 1704.03155v2 [cs.CV] 10 Jul 2017.

[8] E. Charniak, Introduction to Deep Learning, ISBN: 9780262039512192 pp. | 7 in x 9 in75 b&w illus. January 2019.

[9] O. Rippel and L. Bourdev, “Real-Time Adaptive Image Compression,” The 34th Int. Conf. on Mach. Learn., 2017. doi: arXiv:1705.05823v1.

[10] G. Toderici et al., “Full Resolution Image Compression with Recurrent Neural Networks,” arXiv e-prints.,2016. doi: arXiv:1608.05148.

[11] T. Law, H. Itoh and H. Seki, “A neural-network assisted Japanese-English machine translation system,” in Proceedings of 1993 Int. Conf. on Neural Networks.

[12] Md. M. Hossain, K.E.U Ahmed and A.R Uddin, “English to Bangla Translation in Structural Way Using Neural Networks,” in 2009 Int. Conf. on Information and Multimedia Tech.

[13] A. Rosebrock, Non-Maximum Suppression for Object Detection in Python, https://www.pyimagesearch.com/2014/11/17/non-maximum-suppression-object-detection-python/

[14] R. Agarwal, Object Detection: An End to End Theoretical Perspective, https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a

[15] C. Woodford, Optical character recognition (OCR), https://www.explainthatstuff.com/how-ocr-works.html

[16] D.G. Lowe, “Object recognition from local scale-invariant features,” in Proceedings of the 1999 IEEE Int. Conf. on Computer Vision.

[17] M. Unser, “Texture classification and segmentation using wavelet frames,” IEEE Transactions on Image Processing, Vol. 4, Issue: 11, Nov 1995.

[18] O. Vinyals, “Show and tell: A neural image caption generator,” in 2015 IEEE Int. Conf. on Comp. Vision and Pattern Recognition.

[19] T. Tracey, Language Translation with RNNs, https://towardsdatascience.com/language-translation-with-rnns-d84d43b40571

[20] H. Lamba, Image Captioning with Keras, https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8

[21] RNN neuron from https://en.wikipedia.org/wiki/Recurrent_neural_network

[22] Google Translate, https://translate.google.com/

[23] Train and Test Images, https://en.wikipedia.org

[24] G. Shpreber, A gentle introduction to OCR, https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa

[25] S. Banerjee, An Introduction To Recurrent Neural Networks, https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912

[26] M. Nguyen, Illustrated Guide to LSTM’s and GRU’s: A step by step explanation, https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[27] A Beginner's Guide to Word2Vec and Neural Word Embeddings, https://skymind.ai/wiki/word2vec

[28] What is Tesseract and how it works?, https://medium.com/@Bytepace/what-is-tesseract-and-how-it-works-dfff720f4a32

[29] Google Cloud Vision, https://cloud.google.com/vision/

[30] Microsoft Bot, https://dev.botframework.com/

[31] Google Colaboratory, https://colab.research.google.com/

[32] https://github.com/tommytracey/AIND-Capstone

[33] J. Brownlee, Gentle Introduction to the Adam Optimization Algorithm for Deep Learning, https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[34] K. Mokhtar, S.S. Bukhari and A.Dengel, “OCR Error Correction: State-of-the-Art vs an NMT-based Approach,” in 13th IAPR Int. Workshop On Document Analysis Sys., 2018.