Real Time Translation of Malayalam Notice Boards to English Directions

Akshay K.
Department of Computer Science Engineering
TocH Institute of Science and Technology, Ernakulam, Kerala 682313

Aravind Das A. M.
Department of Computer Science Engineering
TocH Institute of Science and Technology, Ernakulam Kerala 682313

Carral Vincent
Department of Computer Science Engineering
TocH Institute of Science and Technology, Ernakulam Kerala 682313

Betty Babu
Department of Computer Science Engineering
TocH Institute of Science and Technology, Ernakulam Kerala 682313

Rasmi P. S, Phd
Department of Computer Science Engineering
TocH Institute of Science and Technology, Ernakulam Kerala 682313

ABSTRACT
Neural Machine Translation (NMT) is an emerging technique depicting impressive performance, better than traditional machine translation methods. It is observed that NMT models have a strong efficacy to learn language constructs, improving performance. Considered as one of the toughest Indian languages to learn and comprehend, Malayalam is extensively used in Road Signs and Notice Boards in Kerala as it increasingly becomes India's tourism hub. In this paper, the barrier faced by the tourists is resolved by providing real-time translation to English. The results obtained show that accuracy can be improved by incorporating Deep Learning and Natural Language Processing (NLP) in translation. This paper is envisioned to not only convert notice boards but also translate Malayalam that is written and printed on all mediums.

General Terms
Deep Learning, Natural Language Processing (NLP), Neural Machine Translation (NMT)

Keywords
Binarization, Grayscale, NLP unit, Translation unit, OpenCV

1. INTRODUCTION
Language translation is a much useful prowess in today's globalized world. It allows people from all corners of the world to be linked and to share information. With the presence of many languages with diverse characteristics in various countries, communication across different linguistic groups can be facilitated to a great extent by providing real-time translators. Due to the complexities of grammar and the ease with which context meanings can be lost, translation has become a major branch of learning. Machine Translation (MT) is the method of converting one natural language into another, preserving the meaning of the input text. But it is a challenging task in the case of Indian languages. Hence Neural Machine Translation (NMT) that uses deep neural networks is used. Unlike traditional translators, NMT builds and trains a single and large neural network that reads a sentence and outputs correct translated text. With this in mind, the aim is to create a real-time translator that translates Malayalam to English which utilizes a smartphone camera which in turn will help tourists in Kerala to understand different signboards and notices found throughout the state. The focus of the paper will be on improving the accuracy of the translation by incorporating NLP and Deep Learning in translation and is envisioned to not only convert notice boards but also translate Malayalam that is written and printed on all mediums. Unlike other translators, the system will capture text from live video than from images through sophisticated models that are trained to detect and translate the texts in a natural scene. The aim is to create software with a simple interface translator that can be used without pre-qualified knowledge and continue improving the project until the perfect translation is possible.

2. LITERATURE REVIEW
There are many existing language-translation applications but the one that can translate Malayalam text to English with better accuracy is very less. "Google Translate" is a free multilingual machine translation service developed by Google, to translate text. It supports 103 languages, and the software uses optical character recognition (OCR) to identify text in photos and translate the words including Malayalam to English translation. Translations need cellular data or Wi-Fi on iOS, but Android users are able to download offline language packs to use as needed. "Microsoft Translator" is another multilingual machine translation cloud service provided by Microsoft, to translate text. It uses machine translation to create instantaneous translations from one natural language to another. The service supports 65 language systems but doesn't support Malayalam to English translation. Microsoft wins a major point over Google with the superior design of its real-time conversation mode, & this feature makes it easier to have natural conversations with the people you meet on travels. Another translation application is "Word Lens" which uses built-in-cameras on smartphones to scan and identify the text in one natural language and translate and display the words in another language on the device's display. The outputted words were displayed in the original context on the original background, and the translation was performed in real-time without connection to the internet. Word Lens also doesn't provide Malayalam to English translation. Hence there is no such existing system that can translate Malayalam to English in real time

.

3. OBJECTIVES

The main objectives of the project are to develop a mobile application that uses a built-in camera to translate text from images of Malayalam notice boards to English. Also, perform translation that meets the correctness and conciseness of content. The system should use image processing techniques for image extraction and recognize the sentences. To apply Natural Language Processing (NLP) and Deep Learning to increase the efficiency and accuracy of the translation. Also to make a comparative study of currently working translators that utilizes image processing & NLP.

4. METHODOLOGY

The system is mainly divided into two modules: Text extraction module and Text Translation module.

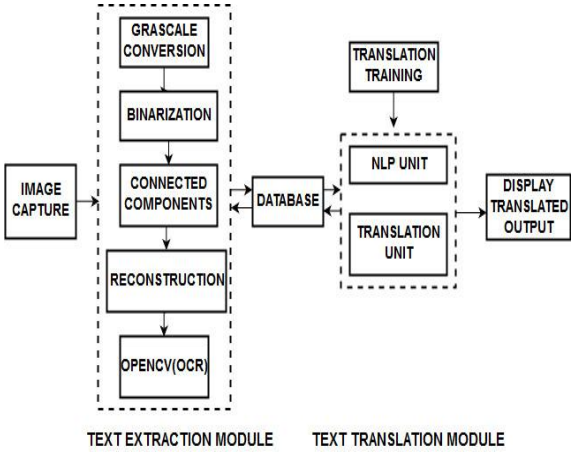

Fig 1: Block diagram of the system.

4.1 TEXT EXTRACTION

A Text Information Extraction (TIE) system receives an input in the form of a still image or a sequence of images. The images can be in grayscale or color, compressed or uncompressed, and the text in the images may or may not move. The TIE problem includes text detection, localization, tracking, extraction and enhancement, and recognition.

Firstly, a colored image is converted to grayscale. A color image that includes color information for each pixel is converted into grayscale images that have a range of shades of gray without apparent color.

A binary image is a digital image that can have two possible values for each pixel which is a single bit 0 or 1. The name black and white is used to represent the bits. To form a binary image, select a threshold intensity value. Pixels with greater intensity value than the threshold are considered as 0 (black) and pixels with intensity less than the threshold value are changed to 1 (white). Thus the image is changed to a binary image.

For text detection and localization, the first step is to find the connected components. Two pixels are said to be connected if they are neighbors and their gray levels specify a certain criterion of similarity between pixels. If S represents a subset of pixels in an image, two pixels' p and q are said to be connected if there exists a path between them consisting entirely of pixels in S. For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S.

After finding the connected components, check the transitions in the values of pixels horizontally. Transitions can be either from bit 0 to bit 1or bit 1 to bit 0. A larger number of transitions from black to white or vice versa are present in case of text regions and the background region will have a lesser number of transitions. If the allocated amount of changes for each row is between two thresholds (low and high thresholds), the row potentially would be considered as text area and the up and down of this row would be specified. Next, search vertically for finding the exact location of the text and ignoring these rows as a text. After the extraction of text regions from images, the text regions become a bit distorted and difficult to read. Recover these components using the original image. The distorted and original images are compared with each other and the pixels which are erased or disfigured are recovered.

For text recognition, OpenCV OCR is used. In order to perform OpenCV OCR text recognition, Tesseract is required which includes a highly accurate deep learning-based model for text recognition. It performs text detection using OpenCV's EAST text detector, a highly accurate deep learning text detector used to detect text in natural scene images. Once the text regions are detected with OpenCV, extract each of the text ROIs and pass them into Tesseract, enabling them to build an entire OpenCV OCR pipeline. A tesseract can work very well under controlled conditions.

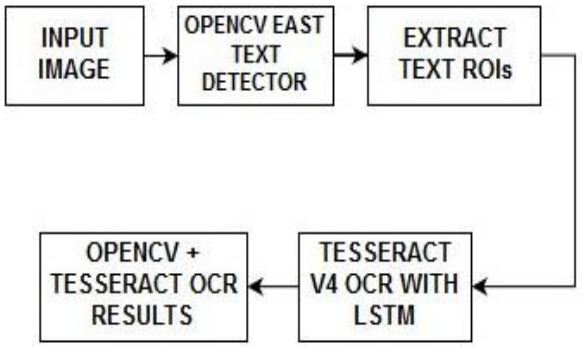

Fig 2: OpenCV OCR pipeline.

Deep learning-based models have managed to obtain unprecedented text recognition accuracy, far beyond traditional feature extraction and machine learning approaches. It was only a matter of time until Tesseract incorporated a deep learning model to further boost OCR accuracy — and in fact, that time has come. Tesseract (v4) supports deep learning-based OCR which is more accurate. The underlying OCR engine itself utilizes a Long Short-Term Memory (LSTM) network, a kind of Recurrent Neural Network (RNN).

4.2 TEXT TRANSLATION

4.2.1 NLP UNIT

NLP or Natural Language Processing is the concept used to interpret the free text and make it analyzable. It is done with the help of deep learning networks that act as a lookup table for understanding semantics and translation. In this module, the plain text is obtained from the previous module in order to subject it to the application of NLP and obtain translated results. The conventional methods include the use of RNN or LSTM which relied on maintaining knowledge about the relationship with all the previously encountered states of recurrent nodes. This necessitated heavy dependence on long term dependencies and the simple vector representation of the input prevented the system from properly interpreting

sentences that had words with equal importance and every word is key.

4.2.2 RNN (Bidirectional LSTM & Self Attention)

For the current scenario, choose something much more advanced than the ordinary LSTM. The Bidirectional LSTM is an LSTM where the hidden weights are fed not only to the next iteration in sequence but also the previous one.

The proposed sentence embedding model consists of two parts. The first part is the bidirectional LSTM, and the second part is the self-attention mechanism, which provides a set of summation weight vectors for the LSTM hidden states. These set of summation weight vectors are dotted with the LSTM hidden states, and the resulting weighted LSTM hidden states are considered as an embedding for the sentence. It can be combined with, for example, a multilayer perceptron to be applied on a downstream application. The figure shows an example when the proposed sentence embedding model is applied to sentiment analysis, combined with a fully connected layer and a softmax layer. Besides using a fully connected layer, propose an approach that prunes weight connections by utilizing the 2-D structure of matrix sentence embedding.

Suppose there is a sentence, which has n tokens, represented in a sequence of word embeddings,

S = (w1,w2,⋯wn)

Here w_i is a vector standing for a d-dimensional word embedding for the i-th word in the sentence. S is thus a sequence represented as a 2-D matrix, which concatenates all the word embeddings together. S should have the shape n-by-d. Now each entry in the sequence S is independent with each other. To gain some dependency between adjacent words within a single sentence, use a bidirectional LSTM to process the sentence and concatenate each with to obtain a hidden state h_t.

Let ‘u’ denote the hidden unit number for each unidirectional LSTM. For simplicity, note all the n h_ts as H, who have the size n-by-2u.

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(w_t, \overleftarrow{h_{t+1}})$$

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t, \overrightarrow{h_{t-1}})$$

H = (h1,h2,⋯hn)

The aim is to encode a variable length sentence into a fixed size embedding. That is achieved by choosing a linear combination of ‘n’ LSTM hidden vectors in H. Self-attention mechanism is used to compute the linear combination. The attention mechanism takes the whole LSTM hidden states H as input, and outputs a vector of weights ‘a’,

a=softmaxws2 $tanh\left(W_{s1}H^T\right)\big)$

Here W_{s1} is a weight matrix with a shape of d_a-by-2u and w_{s2} is a vector of parameters with size d_a, where d_a is a hyper parameter that can be set arbitrarily. Since H is sized n-by-2u, the annotation vector ‘a’ will have a size n. The softmax()˙ ensures all the computed weights sum up to 1. Then sum up the LSTM hidden states H according to the weight provided by ‘a’ to get a vector representation m of the input sentence. This vector representation focuses on a particular component of the sentence, for example, a set of related words or phrases. Therefore, it is expected to reflect a component of the semantics in a sentence. Also, there can be multiple components in a sentence that together forms the overall semantics of the whole sentence.

Thus, to represent the overall semantics of the sentence, multiple m’s are needed that focus on different parts of the sentence. Thus perform multiple hops of attention. If ‘r‘ different parts to be extracted from the sentence, extend the w_{s2} into a r-by-d_a matrix, note it as W_{s2}, and the resulting annotation vector ‘a’ becomes annotation matrix A.

Formally,

$$A = softmax\left(W_{s2}tanh\left(W_{s1}H^T\right)\right)$$

Here the softmax()˙ is performed along the second dimension of its input. Deem Equation above as a 2-layer MLP without bias, whose hidden unit numbers is d_a, and parameters are $\{W_{s2}, Ws_1\}$.

The embedding vector m then becomes an r-by-2u embedding matrix M. Compute the r weighted sums by multiplying the annotation matrix A and LSTM hidden states H, the resulting matrix is the sentence embedding,

M = AH

This above matrix M acts as a source for encoding the relevant data for translation. It acts as a lookup table that represents weights and hidden state values of each neuron which corresponds to one word in the source language. Similarly, another one is generated for the output destination language.

4.2.3 TRANSLATION UNIT

The translation unit can be visualized as a phase module, the phases being, encoding and decoding. The LSTM mentioned in the previous unit converts the input sequence to a fixed size feature vector that encodes primarily the information which is crucial for translation from the input sentence and ignores the irrelevant information. After the encoding process, the context vector is obtained - which is like a snapshot of the entire source sequence which is used further to predict the output.

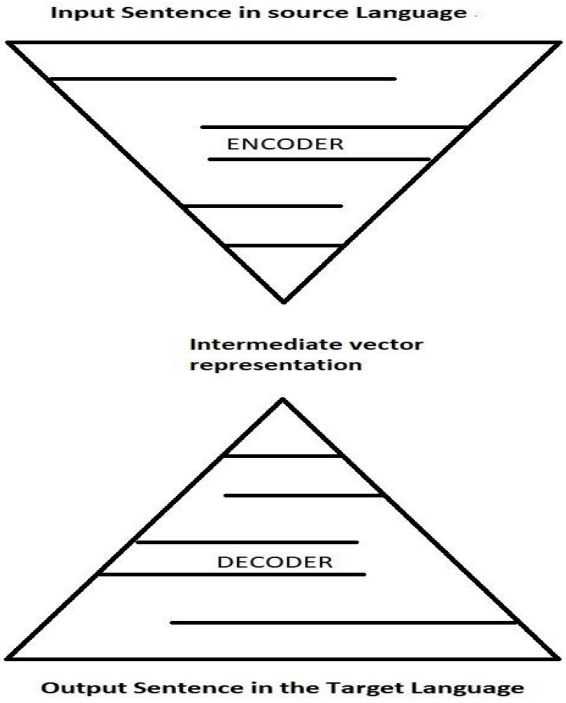

Fig 2: Translation Unit.

A dense layer with softmax similar to a normal NN, but the difference is that it is time distributed i.e. one of these for each time step. The top layer thus will have one neuron for every single word in the vocab, and hence the top layer will be huge in size. This finally acts as one giant lookup table for translating the source language, given as input to it. Every input sentence passed through the model travels through the neurons corresponding to the words and embedding and the decoding part of the model similarly does the reverse in the other language to provide the translated output.

Here use Sequence-to-Sequence (seq2seq) models that are used for a variety of NLP tasks, such as text summarization, speech recognition, DNA sequence modeling, among others. The accuracy of translation can be increased by using a bulk amount of data with some text preprocessing (text cleaning) done on it.

5. RESULTS AND DISCUSSIONS

The testing of the system was done with different kinds of sentences in the Malayalam language. The simple sentence contains only one independent clause and no dependent clauses which are adequate for notice board translation. This Malayalam to English translation system generates correct meaningful English sentences as output in most of the cases. The system works well for all simple sentences in their 9 tense forms, their negatives, and question form.

A group of sample input sentences with the tabulated outputs is shown below in the table to give a picture of the results obtained. Results include both correctly obtained output and incorrectly obtained output.

Table 1. Input and the corresponding Output results

INPUT SENTENCES	DECODED SENTENCES
അത് വിനോദമാണ്	That's fun.
അവന് നീന്താൻ കഴിയും.	He can swim.
പതുക്കെ നടക്കുക.	Walk slowly.
അത് വിവേകപരം ആണ് .	That's his.
എനിക്ക് കാൾ വേണം.	I am sure.

The evaluation of this translation system was done manually. The human experts in translation evaluate the translation quality of this example based on neural machine translation. The quality of the translation is measured by the accuracy of the translated sentence in English. About 75% of accuracy has been obtained. The translation system completely relies on the dataset that contains examples of already translated words, phrases, and sentences. System performance can be improved by training with cleaned data.

The model that used the character level embedding model slightly outperformed the word-based model with more accurate translated results even when morphological rich languages like Malayalam are used since the dataset is of small size. The word-based model's performance can be expected to increase drastically with respect to the increase in data.

Due to insufficient data, the model is made to overfit onto the training data and hence the prediction of trained data has a loss of 0.0064 and the model will not be able to predict new sentences at the current level. This problem can be solved by increasing the amount of data the model is trained with and by utilizing NLP for processing the dataset.

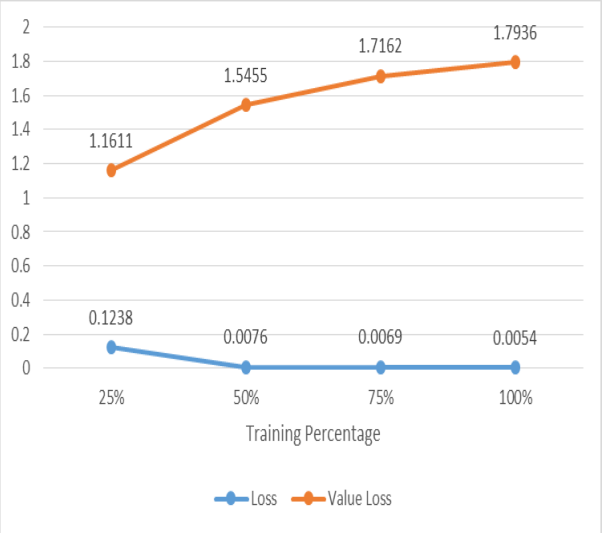

Fig 2: Graph showing the change in loss during each stage of training.

6. FUTURE WORK

To incorporate different modes like real-time voice-voice translation, text-voice, voice-text, etc. and making the application to translate language in all forms of communication and provide translations for other native languages like Tamil, Telugu, etc. to increase the use and scope of growth. In the future, the application can be modified to have all the functionality of online translation to offline translation and thereby making the application independent and decrease the requirements and make the translation available for all types like earpiece device, augmentation devices, smart watches, etc. The application can be made better by continuously training the model with a cleaner dataset and increase the accuracy of the model.

7. CONCLUSION

Deep Learning algorithms combined with Image Processing and NLP will provide an understandable translation of languages and further accuracy can be increased by the utilization of clean data for training the models that work inside the application that requires limited computational resources and hence be available in devices such as mobiles. The work proposed and built a new approach for Malayalam to English translation there are varieties of applications for this translation system. In Kerala, all the population is not so familiar with English. So such kinds of systems will offer a great contribution to society if it's available for the public. And hoping that this system can be efficiently used by everyone if it is released as an open source.

8. REFERENCES

[1] Bartz, Christian, Haojin Yang and Christoph Meinel. "STN-OCR: A single Neural Network for Text Detection and Text Recognition." CoRR abs/1707.08831 (2017).

[2] Sooraj, S & K, Manjusha & Kumar, M & Kp, Soman. (2018). "Deep learning based spell checker for Malayalam language. Journal of Intelligent & Fuzzy Systems", 34. 1427-1434. 10.3233/JIFS-169438.

[3] Remya Rajan, Remya Sivan, Remya Ravindran, K.P. Soman, " Rule Based Machine Translation from English to Malayalam" 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, DOI: 10.1109/ACT.2009.

[4] Rongxiang Weng, Shujian Huang, Zaixiang Zheng, Xinyu Dai and Jiajun Chen, " Neural Machine Translation with Word Predictions" State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210023, China.

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

[6] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang and J. Xie, "A Hierarchy-to-Sequence Attentional Neural Machine Translation Model," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 3, pp. 623-632.doi: 10.1109/TASLP.2018.2789721.

[7] W. Bieniecki, S. Grabowski and W. Rozenberg, "Image Preprocessing for Improving OCR Accuracy," 2007 International Conference on Perspective Technologies and Methods in MEMS Design, Lviv-Polyana, 2007, pp. 75-80.doi: 10.1109/MEMSTECH.2007.4283429.

[8] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, "Google's Neural Machine Translation System: Bridging the Gapbetween Human and Machine Translation", arXiv:1609.08144v2 [cs.CL], October 2016.

[9] Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, "Achieving Human Parity on AutomaticChinese to English News Translation", arXiv:1803.05567v2 [cs.CL], June 2018

[10] Bradski G, "The OpenCV Library", Dr Dobb's Journal of Software Tools, 2000.

[11] Patel, Chirag & Patel, Atul & Patel, Dharmendra. (2012), "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study", International Journal of Computer Applications. 55. 50-56. 10.5120/8794-2784.

[12] Daniel Watson, Nasser Zalmout and Nizar Habash, "Utilizing Character and Word Embeddings for Text Normalizationwith Sequence-to-Sequence Models", Computational Approaches to Modeling Language Lab, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

[13] Sina Ahmadi.2017, "Attention-based encoder-decodernetworks for spelling and grammatical error correc-tion", Master's thesis, Paris Descartes University.

[14] B. Zhang, D. Xiong and J. Su, "Neural Machine Translation with Deep Attention," in IEEE Transactions on Pattern Analysis and Machine Intelligence.doi: 10.1109/TPAMI.2018.2876404.

[15] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi and S. Jain, "Machine translation using deep learning: An overview," 2017 International Conference on Computer, Communications and Electronics (Comptelix), Jaipur, 2017pp.162167.doi:10.1109/COMPTELIX.2017.8003957.