SIGNBOARD TRANSLATION FROM INDIAN VERNACULAR LANGUAGES

Vishnu Teja Chikkala, Rajpreet Srivastav, Sunil Kumar

ABSTRACT

India has, along with English, 22 constitutionally recognized / scheduled languages, written in 13 different scripts. Facilitation of seamless, real-time translation services between these remains a challenge, both due to limited availability of corpora and, the limitations of speed, accuracy and flexibility of existing statistical translation models and libraries. In this research, an attempt has been made to create an end-to-end model for the detection, recognition and translation / transliteration of text in a source language to a target language, using deep learning to train neural network models on a combination of synthetically generated and freely-available data sets, and deploy the same on a simple, easy-to-use and scalable Android App, specifically targeting inter-state traveler.

Keywords-Translation, Transliteration, Indian Languages, Natural Language Processing, Deep Learning, Computer Vision

1-INTRODUCTION

The objective of this paper is to describe the development methodology of a simple and easy-to-use Android mobile app which provides a two-click, picture-to-text translation / transliteration service for Indian vernacular languages, using deep neural networks and natural language processing models trained for text detection, recognition and translation tasks on collected and freely-available data sets.

2-LITERATURE REVIEW

C.Kurian et al. concluded that the Indian community faces a "Digital Divide" due to dominance of English as mode of communication in higher education, judiciary, corporate sector and Public administration at Central level whereas the government in states work in their respective regional languages[1].

From the 2011 Census of India, it is observed that. while 99 % of the population speak one of these scheduled languages in various dialects (which number in the thousands), according to Census 2011,

the total percentage of English speakers is at 10 %, and that too is skewed towards the urban population [2].

N.P. Desai et al noted that traditionally NLP had been approached with statistical methods such as Hidden Markov Machines (HMM), Support vector machine(SVM), Conditional Random Field(CRF), Naive Bayes(NB), etc, which take a large amount of tagged/annotated data (corpus) to statistically analyze and learn the language characteristics , and suggested that deep learning methods or a 'connectionist approach' could return better results[3]. The reasons for the same suggested by them and other authors, in brief, were -

the simplicity of the solution in rapidly prototyping and establishing practically effective systems

the lower cost of annotation of the training data [4],

they attempt to more closely emulate the learning process of biological brains [3] [5] [6], among other reasons.

However, A. Kunchukuttan et al., note that the collection of a uniform corpus and standard data sets for training models remains a challenge across all regional languages. The large number of morphological variations across Indic languages also contributes to this issue[7]. This study also proposed the creation of a "large-scale, general-domain" corpus for 10 Indian Languages across 2 scripts[7]. J. Philip et al., in their study, also highlighted several efforts at collating standard data sets and corpora for Indian languages[4].

Sharma et al., 2017 concluded that almost all existing Indian language machine transliteration systems are based on statistical and hybrid approach[8].

A few models proposed in this field are as follows-

Authors	Model	Accuracy
Bhanja et al., 2019 [9]	CNN-LSTM Architecture using ResNet	93% (NITS-LD), 89% (OGI-MLTS)
J. Philip et al, 2019 [4]	IL-Multi,	34% (cold-start), 40%, (with transferred learning)
Arafat, S. Y. et al, 2020 [10]	FasterRCNN - CNN - RRNN Architecture	99%

Table 1: List of Proposed Models

3-METHODOLOGY USED

3.1 Dataset Preparation

The training dataset for the text detection and text recognition tasks consists of a large number of images

containing scene text, synthetically generated. Each of the images has a corresponding annotation, listing the bounding boxes and text script present in the images.

The test dataset for text detection and text recognition tasks consists of actual images containing natural scene text.

The train and test data sets are both in form of xml files containing

serialized pairs of source language script and corresponding target language script.

Figure 1 Detection / Recognition Training Dataset

Figure 2 Transliteration Training Dataset

3.2 Text Detection

The text detection task will be carried out by a Faster Region-based Convolutional Network (Faster R-CNN) with Feature Pyramid Network (FPN; for bounding box tightening) from the FAIR Detectron2 kit, which has been trained for object detection on the COCO dataset. We will fine-tune this model to the task at hand by training and validating further on the above scene text database.
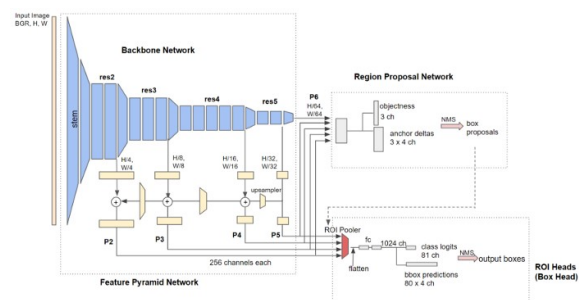

Figure 3 Detectron2 Faster RCNN with FPN

3.3 Text Recognition

The text recognition task will be carried out by an encoder-decoder model setup which takes the cropped bounding box of text as input. The encoder is a Convolutional Neural Network (CNN) while the decoder is a Long Short-Term Memory model (LSTM). Connectionist Temporal Classifification (CTC) loss will be used to eliminate duplicate recognition of the same letter by adjacent CNN features
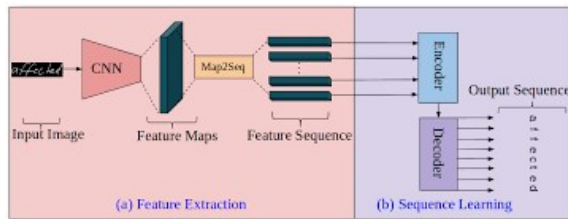
Figure 4 CNN-GRU Architecture

3.4 Text Transliteration

The transliteration task will carried by a LSTM - LSTM encoder-decoder model with attention mechanism, which takes source language script as input and generates target language script as output
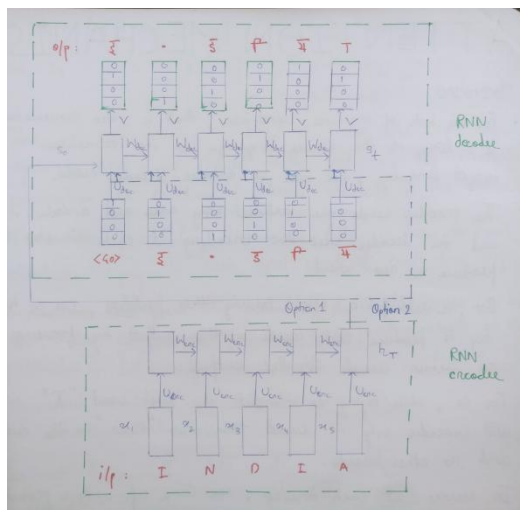

Figure 5 GRU-GRU Architecture

3.5 App Design

The model will be mounted on and integrated with the app with the help of Tenserflow Lite. This will make it capable to run inference on user captured images containing natural scene text. The app will make use of Google Firebase API to provide UI functionality and user services. .

4-DESIGN

The basic functioning of the app is as follows:

User captures a photo of the signboard. The image is resized so as to be suitable as input to the model. The model takes the image as input. The text within the message is detected, extracted and transcribed to target language. The text output (or error message, in case of failure to generate output within threshold confidence), is displayed on screen.
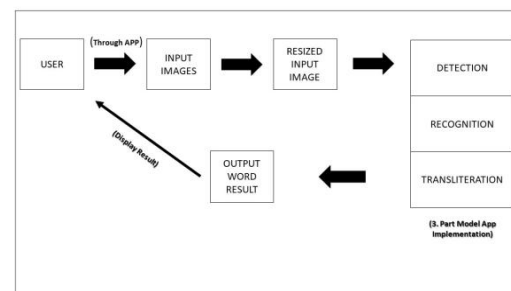

Figure 6 App Architecture

The artificial neural network (ANN) behind the core functioning of the app is made up of 3 models performing consecutive tasks. That is, the output of a preceding model will be fed as input to the succeeding model, and thus they act as one model unit.
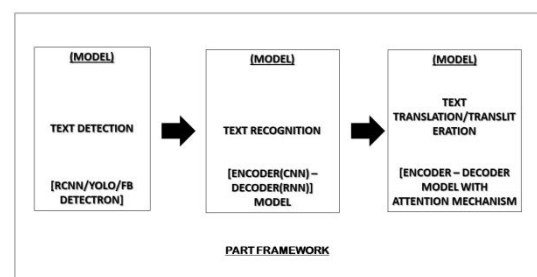

Figure 7 Model Architecture

5-RESULTS

The final outcome is an Android app which can detect and extract text of source language in a natural scene, from a user-captured image, and translate and present the text in the target language.

6-CONCLUSION AND SCOPE FOR FUTURE

The rapid dissemination of infomation in digital format, coupled with the complexities of having such linguistic diversity in our nation, necessitate the development of computer vision and natural language processing models which facilitate communication across languages.

It is also imperative that these solutions be scalable, accessible and distributable to serve the large population of our country.

The aim of the project, to develop a signboard translation system, was an example of how deep learning and artificial neural networks can be harnessed to make life more convenient. It is a simple app which detects, extracts and translates from source to target language, text captured as part of a natural scene by a user, and can be especially useful to interstate and international travellers. Various neural network models were used to train on a set of synthetically generated and curated data and the best performing model, given the time constraints, was selected for evaluation. However, these models, even in this limited scope, still have room for improvement using more diverse testing and use of techniques such as dropout,batch normalization, ensemble modelling, etc.

This project can also be extended to work with longer sequences of text, and in particular, other languages. given an appropriate curation of dataset. These can then be made available to the user of the app as downloadable packs.

A long-term goal of this project will be to make an easy-to-use API, so that, given the datasets, other developers using natural scene detection, recognition and transliteration as a part of their projects (for example, detection of labels from medicine bottles, or very accurate newsreader apps) can easily train and deploy models.

7-REFERENCES

1. C. Kurian and K. Kannan Balakrishnan.(2008). "Natural language processing in india prospects and challanges".*In Proceedings of the International Conference on "Recent Trends in Computational Science.*

2. ORGI. "Census of India: Comparative speaker's strength of

Scheduled Languages-1951, 1961, 1971, 1981, 1991 ,2001 and 2011"

3. N. P. Desai and V. K. Dabhi. (2021) "Taxonomic survey of hindi language NLP systems", *arXiv preprint arXiv:2102.00214,*

4. J. Philip, V. P. Namboodiri, and C. Jawahar (2019), "A baseline neural machine translation system for indian languages", *arXiv preprint arXiv:1907.12437.*

5. M. Rosca and T. Breuel. (2016) "Sequence-to-sequence neural network models for transliteration.". *arXiv preprint arXiv:1610.09565.*

6. T. Deselaers, S. Hasan, O. Bender, and H. Ney (2009). "A deep learning approach to machine transliteration". *In Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 233–241.*

7. A. Kunchukuttan, D. Kakwani, S. Golla, A. Bhattacharyya, M. M. Khapra, P. Kumar, et al. (2020). "Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages." *arXiv preprint arXiv:2005.00085.*

8. A. Sharma and D. Rattan.(2017) *"Machine transliteration for indian languages: A review."*

International Journal of Advanced Research in Computer Science, 8(8),

9. Bhanja, C. C., Bisharad, D., & Laskar, R. H. (2019). "Deep residual networks for pre-classification based Indian language identification*." Journal of Intelligent & Fuzzy Systems, 36(3), 2207-2218.*

10. Arafat, S. Y., & Iqbal, M. J. (2020). "Urdu-text detection and recognition in natural scene images using deep learning*." IEEE Access, 8, 96787-96803.*