Development of On-Device Machine Learning Tools for Medication Warning Label Recognition

Virendra Singh Rathore
Department of Computer Science
Lakehead University
Thunder Bay, Canada
vrathore@lakeheadu.ca

Dr. Sabah Mohammed
COMP5800YB Supervisor
Lakehead University
Thunder Bay, Canada
mohammed@lakeheadu.ca

Abstract— The world that we live in today has so many languages that we speak throughout. Out of more than the six thousand languages that are spoken around the world, the most common is the English language. But there are people who do not understand it. When we speak of medications, the language barrier becomes riskier. If consumers are not able to understand the warning labels on the medicines, it might prove fatal to life. So it seems important that there be a tool or an application that would help them recognize and decode the warnings and/ or the instructions that are printed the medication pill containers. This tool will prove very useful for patients across the world and also people who consume any kind of medication without understanding the procedures/ risks that are specific to those medications.

Keywords— Machine learning, On-device learning, OpenCV, Text Extraction, Translation, Image Processing, medication, language barrier

I. INTRODUCTION

In the world that we live today, travelling across the globe is no big deal. I, for one, have been lucky to have travelled across my own country, i.e. India, and have had the opportunity to travel abroad. Although we have grown past the times when we could not communicate or understand people from other parts of the world, it is sometimes difficult to decode certain forms of communications when you do not have much knowledge of the local language, or the resources to decode or provide you with the understanding of the language. For example, if you are travelling on your own to some other country, like Japan, for example, you may not be able to understand or decode the signboards, shop hoardings or the menu at a restaurant as they are all in Japanese. What if you have an application that you could take pictures with and then you get the extracted text into the language that you understand! That would be a very useful application in situations where you are not able to get any other assistance or help to translate the text.

This idea is to apply this technique to avoid mishaps and reactions that happen when consuming pharmaceutical drugs. As we all know medicinal drugs that come packaged in bottles, boxes and/ or tablet cases usually have precautionary instructions or warnings on them. It might prove very injurious

to health if the medicine is consumed without the proper understanding of the warning labels on them. So if the person is not able to read/ understand the language on these, this application finds the perfect practical use in this area. This idea is planned to be practically implemented and provided to the patients and people who consume medicines at the Thunder Bay Hospital (Thunder Bay) and on the nearby pharmaceutical drug consumers, under the suggestion and guidance of my Project Supervisor, Dr. Sabah Mohammed. This could prove quite beneficial if we achieve this text extraction and translation in the most commonly used languages in the area, which can be a few as used by International students, Hindi for example, and to the most common and popular of the aboriginal languages, Ojibwe for instance.

II. EXISTING SYSTEM AND ITS MECHANISM

A. Google Translate

Google translate has come a long way from its initial launch in April of 2006. What the application intended to do was provide the user with a tool to translate words, phrases, and sentences from one language to another. In the beginning, it was all with the translation of words, phrases, and sentences to and from a chosen set of languages. But gradually, the translate application developed to such an extent that it now has several languages spoken across the world can use voice as the input and output. The more latest and interesting addition to the feature list is the conversion and recognition of visual inputs and convert them into the desired language.

Fig. 1

It uses onboard machine learning to overlay the translated text. This couldn't be done if it was transferring the video of what is being held in front of the screen over the network. Sometimes you might be in another country, and you don't have a data connection. Furthermore, even if you had a data connection, you would want that immersive and responsive experience. Maybe we're in a region or a place where there's no network connectivity. Certainly, we don't want to be streaming images over our satellite phone connection. But machine learning, it's kind of hard, right? And making mobile apps, that's not an easy feat either. But combining the two, well, that's a real challenge. So how can we make this easier?

III. PROPOSAL

Let's talk about how we can build such a system. And do it with a way that doesn't involve tedious data collection and labelling and then spending weeks of training on a big server with GPUs and writing distributed code. And then we haven't even talked about how we're going to package that up for mobile, right? As intended, we will be using a pre-trained model that can identify the text from any given image and then extract the specific text from it. Then we can use one of the various tools available to translate the extracted text into the intended language. The bigger challenge lies in the skewing of images. The images taken can not be a perfect blend like a wallpaper, where it is easy to extract the text. So the major focus in the extraction process would be to perform image processing on the images. For example, look at the two images shown below.

Fig. 2.

Fig. 3.

Figure 2 shows a simple warning message which can be easily identified and extracted as text using methods used in the application program, but as a practical approach, we would not be getting those images. We would be dealing with images as shown in Figure 3. Here is where the background texts, images, noises and distortions in the images tend to alter the extraction of text, which is not accurate and hence the translation is

effected as well. We will be using the tools provided by OpenCV to get rid of the unnecessary elements and focus on the intended picture area, and hence extract and translate the text with greater accuracy. So to sum up, there are 2 phases in this project application;

1. To get rid of the background elements and unnecessary noises and distortions in the image and bring forth the required area in the image.
2. To use the focused area and extract the text on it and later translate it to an intended language.

We will be experimenting to achieve this target using both Java and Python programming languages and the tools provided by them to perform the intended operations on a given sample of the image dataset.

A. Performance of the application on a Computer/ Laptop

The primary goal is to design the application so as it works on the Laptop on which the code is being designed. The approach has been designed in Python as well as Java. The use of Machine learning tools can be utilized to train the system to understand and learn how to recognize alphabets and words from images, to extract them in later stages as required by the application. It can also be a better idea to use the pre-trained model, which is already capable of understanding and is trained to do that task. So, we will be using Pytesseract for the Python version of the application, and Google Vision API for the Java version of the application.

B. Extending the application to an android app

After the successful execution of the application on the machine level, the application can be easily transformed into a mobile application, as planned into an android application, using Android Studio or any other android development tools.

IV. TOOLS AND TECHNOLOGY

OPTICAL CHARACTER RECOGNITION

Optical Character Recognition or Optical Character Reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a site-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast). Tools:

- **Pytesseract** (Python-tesseract): It is an optical character recognition (OCR) tool for python.
- **Python Imaging Library** (PIL): It adds image processing capabilities to the Python interpreter
- **Tesseract**: Tesseract, a highly popular OCR engine, was originally developed by Hewlett Packard in the 1980s and was then open-sourced in 2005. Google adopted the project in 2006 and has been sponsoring it ever since. Tesseract is an **optical character recognition engine** for various operating systems. It is free software, released under the Apache License, Version 2.0 and development has been sponsored by Google since 2006. In 2006, Tesseract was considered

one of the most accurate open-source OCR engines then available.

- **Googletrans API:** It is a free python library that implements the Google Translate API.

Fig. 4. Architecture of Tesseract

V. LIMITATIONS IN THE CURRENT SYSTEM

The current system works well with images which are straight forward with text with no other unnecessary elements, as told in the proposal part, so we apply transformations using OpenCV tools which can crop out the unnecessary background objects and texts which are not in the main label of the specific target warning on the image. The approach that we will use here is to extract that area of the image where the background is yellow. This is because most of the time the warnings are engraved onto a yellow background, as seen in Fig. 2 and Fig. 3. But there are exceptions here too. There are certain images where the warning labels might not be yellow, but some other colours like as shown below:

Fig. 5. Warning Labels

The solution to such cases can be solved by defining other colour selectors too but that might cause the application to select unnecessary data that is on the image. These are just minor exceptions that we can encounter in the processing of the intentions of the application. This alters the extracted text and might lead to inaccurate, or not to the point extraction and translation of the text from these images. Another minor issue is that the extraction of the text is in multiple lines because the image might have it as such. So the solution is to convert the multiple-line string to a single line. In some cases, this might cause the meaning of the text difficult to understand or inaccurate, but that is to a bearable level.

PYTHON OUTPUTS: GLIMPSES

Below shown are the pictures which show how the code works on a few sample images given into the program application.

➤ Cropping an image

```
In [18]: runfile('C:/Users/Virendra/Desktop/tesr/test1.py',
wdir='C:/Users/Virendra/Desktop/tesr')
(5, 5, 3)
May take with food to lessen chance of stomach upse
-----
Translated(src=en, dest=hi, text=पेट खराब की कम मौका करने के लिए भोजन के साथ समय लग सकता है, pronunciation=None, extra_data={'translat...})
```

Translating the text

```
In [18]: runfile('C:/Users/Virendra/Desktop/tesr/test1.py',
wdir='C:/Users/Virendra/Desktop/tesr')
(5, 5, 3)
May take with food to lessen chance of stomach upse
-----
Translated(src=en, dest=hi, text=पेट खराब की कम मौका करने के लिए भोजन के
साथ समय लग सकता है, pronunciation=None, extra_data="{ 'translat...")
```

Fig.5. Text Extraction and translation

CONTRIBUTION AND FUTURE WORK

The proposed application system that uses images to extract the text and to translate is not a new idea, it has been around and is being worked upon by developers all around the world. The idea of applying this technology to the best areas is what the real challenge is. Me and my Project Supervisor, Dr. Sabah Mohammed discussed and plan to use this idea in the field of medication and health care by implementing the extraction and translation of this application on the medication warning and instruction labels. This will be very beneficial and life-saving for the patients and people who are consuming medicines and are not able to understand the warnings and instructions due to the language barrier, which is English for now. The future plan is to include detection, extraction and translation of these warning labels from and to various other popular languages that are used across the world to make the application more generalized.

CONCLUSION

The system is now able to process the image by focusing on the area where the actual text that is needed is present, and then extract the text from it, and then translate it to another language. A simple code block uses 'os' methods from the 'gTTs' package of python to save and play the audio file of the English text in mp3 format. The sample images have been tested for a few languages such as Hindi, French and Spanish, and it works well. As far as the aboriginal languages are concerned, the package to translate into that is still being worked upon. The code can now be moulded into an android application to be used in a mobile device. An app is a handy tool for consumers who can take images using the camera on the mobile device and then get the translated text from the images/ pictures they take from the device. This can prove to be beneficial to a lot of consumers in a variety of settings and avoid fatalities or damage when there is a language barrier.

REFERENCES

- [1] Muhammad, Ajmal, Farooq Ahmad, A. M. Martinez-Enriquez, Mudasser Naseer, Aslam Muhammad, and Mohsin Ashraf. "Image to Multilingual Text Conversion for Literacy Education." In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1328-1332. IEEE, 2018.
- [2] Patil, Mrunmayee, and Ramesh Kagalkar. "A Review on Conversion of Image to Text as well as Speech using Edge detection and Image Segmentation." *International Journal of Advance Research in Computer Science Management Studies* 2 (2014).
- [3] Brown, Peter Fitzhugh, John Cocke, Stephen Andrew Della Pietra, Vincent Joseph Della Pietra, Frederick Jelinek, Jennifer Ceil Lai, and Robert Leroy Mercer. "System for parametric text to text language translation." U.S. Patent 5,805,832, issued September 8, 1998.
- [4] Trudeau, John R. "Language translation for real-time text-based conversations." U.S. Patent 5,987,401, issued November 16, 1999.
- [5] Ribeiro, Mário RM, Duarte Jùlio, Vasco Abelha, António Abelha, and José Machado. "A Comparative Study of Optical Character Recognition in Health Information System." In *2019 International Conference in Engineering Applications (ICEA)*, pp. 1-5. IEEE, 2019.
- [6] Ribeiro, Mário RM, Duarte Jùlio, Vasco Abelha, António Abelha, and José Machado. "A Comparative Study of Optical Character Recognition in Health Information System." In *2019 International Conference in Engineering Applications (ICEA)*, pp. 1-5. IEEE, 2019.
- [7] Malmaud, Jonathan, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. "What's cookin'? interpreting cooking videos using text, speech and vision." *arXiv preprint arXiv:1503.01558* (2015).
- [8] Konečný, Jakub, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. "Federated optimization: Distributed machine learning for on-device intelligence." *arXiv preprint arXiv:1610.02527* (2016). J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [9] Liang, Ming, and Xiaolin Hu. "Recurrent convolutional neural network for object recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3367-3375. 2015.