

Translate Images on Android Device

Qiaojing Yan, qiaojing@stanford.edu

Wei Wang, wwang23@stanford.edu

Motivation

Currently, when a user needs to know the meaning of a segment of text in his native language, he usually needs to manually type the text into a dictionary or search engine to get the translation. This can be a huge burden if there is a lot of text. Some apps can extract text using OCR and show the aggregated translation to user. While this reduce the trouble in typing the text, its result doesn't show the positional information of the text. For example, if a user is studying the map of the United states, it would give an output of all the name of the states, concatenated. However, the location of the words are missing and it is unintuitive for user to know where each state is.

We want to build an Android app that can not only recognize text, but also overlap the translated text on top of the original text, thus preserve the location information. We believe this app would be useful in everyday life for foreign visitors. It can be used to recognize maps, menus, traffic signs, etc. It can also be added to an AR system to give intuitive realtime translation of text in the environment.

Our approach

We'll focus on detecting English words and overlay the Chinese or Spanish translation on top of the original image. Our approach includes following steps

Step 1: Image Capture and Preprocessing (Wei):

During preprocessing, we plan to perform the following steps:

- i. Noise reduction using median filtering
- ii. Increase contrast using adaptive histogram equalization
- iii. Optional: Binarization using Otsu's method
- iv. Use Hough transformation to get the orientation of the text.
(We need the orientation to overlay the translated the text.)
- v. Rotate text to upright position

Step 2: Optical Character Recognition (Qiaojing)

We perform OCR on the preprocessed image. We plan to use Tesseract, an open source tool for OCR. These is also an Mobile Vision API by Google. However, it is not

open-sourced so we don't plan to use it for this project. Tesseract has an Android wrapper, so we can use it in app. We will use Tesseract to extract different blocks of text.

Step 3: Translation of text (Qiaojing)

Google has some API support for translation. Since translation is not the focus of the class, we'll just use existing APIs.

Step 4: Erase original text and overlay translated words (Qiaojing and Wei)

Using the result from Tesseract, we get the bounding box of the words. We extract the text color and the background color. We can use clustering to extract these inside each bounding box.

We then erase the text (erase pixels that has the color close to text, we can define a threshold of distance). Finally, we overlay the translated text onto the image using the orientation (found in step 1) and the color information.

Challenges

One of challenge is to develop a good preprocessing procedure to increase the OCR accuracy. The quality of the results from Tesseract can be greatly influenced by preprocessing.

Another challenge is to explore what kind of functions does Tesseract provide and how to integrate that into our app.

Equipment

We are using an Android Device, which we borrowed from the lab.

Reference

[1] Bieniecki, Wojciech, Szymon Grabowski, and Wojciech Rozenberg. "Image preprocessing for improving ocr accuracy." *2007 International Conference on Perspective Technologies and Methods in MEMS Design*. IEEE, 2007.

[2] Smith, Ray. "An overview of the Tesseract OCR engine." (2007).