Mobile Sign Translator for the Thai Language

Tomas Tinoco De Rubira
Department of Electrical Engineering,
Stanford University,
ttinoco@stanford.edu
April 29, 2011

Project Proposal

People traveling abroad often face difficulties due to language barriers. Others even avoid going to places with different languages due to a fear of facing such difficulties. With the processing capacity of today’s mobile devices, however, it is possible to design portable systems that can help lower these barriers. For example, in [FGZ+11] and [LE08], the authors describe smartphone-based systems that can detect and translate text in images captured from a phone’s camera. Similarly, in [YGZ+01], the authors describe a portable system that can automatically recognize and translate signs written in Chinese. These systems can provide significant help to travelers and give people more confidence in exploring new places.

In this project, I intend to implement a system similar to the ones described in [FGZ+11], [LE08] and [YGZ+01], that runs on an Android phone, detects signs written in Thai and translates them to English. The specific goals are the following: First, develop an Android application that, using an image processing algorithm based on clustering, extracts a Thai text region present in an image captured from the camera, and sends it to a server. To simplify this task, I assume that the user can move the phone, as necessary, to place the text of interest at the center of the viewfinder, with a horizontal orientation. Second, develop a Python-based server that receives the extracted and processed text image from the Android application, converts it to a string of characters using Google’s Tesseract OCR system, translates the string to English using Google Translate, and returns the text to the Android application.

One of the main challenges of systems of this type is that of automatic text detection. In [FGZ+11], the authors allow user input via a touch on the screen at the location where a word is present, and then use image gradients to find the edges of a box that contains the selected word. In [LE08], the authors use the Adaboost algorithm to train a classifier able to detect text in cluttered scenes. In [YGZ+01], the authors use a multiresolution approach that relies on edge detection and adaptive search. The method that I intend to implement in this project is similar to the one proposed in [FGZ+11], in the sense that it requires user
input. The input is in the form of placing the text at the center of the viewfinder. Then, instead of using image gradients, I intend to use an algorithm based on K-Means clustering for finding the boundaries of the text region. I have performed some initial experiments with this approach, and the results seem promising.

References

