Software Design 04/03/2014
SmarterBoard Meeting
Ryan Louie, Sarah Walters, Doyung Lee, and Zoher Ghadyali

How we started this project?

We decided to abandon the general document note-taker where the user would just take a picture of a whiteboard and receive a document containing the text on the whiteboard. We made this decision because this was our minimum deliverable and it would end up being much more difficult than the variety of extensions we came up with to pursue.

Thus, we broke up in teams of two and pursued two of our extensions: the hand-written code compiler and the circuit diagram generator. We set minimum deliverables we aimed to accomplish in the span of a trial week where we developed both ideas concurrently.

What did the Hand-Written Code Compiler project work out (Sarah and Zoher)?

Our minimum deliverable was to take a picture of handwritten code and recognize what language it was in. We ended up with satisfying language detection just using the keywords specific to each programming language but we ran into OCR/Tesseract problems.

Tesseract was trained off of examples in English and examples in type. Code does not follow the patterns of English and Tesseract had huge problems recognizing images containing hand-writing. Thus, we reached a roadblock that we could only overcome by retraining Tesseract which was not aligned with what we wanted to get out of this project.

After the trial period of a week, we have decided to abandon the code compiler and focus entirely on the Drawn Circuit Schematic generator.

What about the Drawn Circuit Schematic project (Doyung and Ryan)?

Doyung and Ryan have successfully produced a resistor and capacitor identifier that is correct 75% of the time. We generated 263 images by taking pictures of hand-written resistors and capacitors that were produced by different people. With those images they resized them to be all the same size, verified whether each images was of a resistor or a capacitor, and then we used a train test split to train off of 70% of the images and reserve 30% as a test group.

Then using what the classifier learned, we tried to predict on the 30% of the remaining images whether it was a resistor and a capacitor and we reached 75% accuracy.

Currently, we are taking in an image, processing it to generate a clean and understandable rendering of the image by the computer, segmenting the diagram by figuring out where the circuit wires are and following the wires to a component, and then using the identifier.

Working with the RESISTOR Team and Where We Are Going

Our project flows very well with the RESISTOR team's project. We do not have any overlap and we are developing a project that could potentially lead into their project. We are developing a converter that generates a circuit diagram from a drawing of one. Using a generated diagram, the RESISTOR team is producing how that diagram would look on a breadboard.

We want to meet with them at some point and discuss maybe a unified object system both of us will use to identify components and we would definitely like to meet sooner rather than later.

One idea that we are considering is simplifying the project as opposed to a complex object recognition system. We might have components be identified by putting red dots at the beginning and end of each component. Paul also mentioned the idea of putting histograms on top of the angles of the lines in order to interpret the wires in the diagram.

To make this proposed project pipeline work, we are going to place limits on what we are doing. We are not going to be recognizing 28 components. Our goal is to get RC circuits working well before adding other components.

Paul mentioned a failure mode of any machine learning project is that they gather data at the beginning of the project, work with that data, and only go back to get more data at the end. He suggested some form of using OpenCV to make a tighter, closed feedback loop to make debugging easier.