ELECOMP CAPSTONE SUMMIT
FRIDAY, MAY 11, 2018

THE UNIVERSITY OF RHODE ISLAND

THINK BIG WE DO℠

OCReader

Optical Character Recognition Reader for Manufacturing Drawings

Team Members: James Luther (CPE), Kevin Ma (CPE)

Technical Director: Jonathan O'Hare

HEXAGON
MANUFACTURING INTELLIGENCE

Project Motivation

In the manufacturing industry, companies will often use computer aided design (CAD) software to help with the creation, modification, analysis, and optimization of a design. Product manufacturing information (PMI) is usually included as metadata in more comprehensive CAD models, with information in the form of geometric dimensioning and tolerancing (GD&T) symbols and glyphs. Currently, most companies have not incorporated PMI in their CAD models and resort to conveying the necessary GD&T information through 2D layout drawings, which requires human interpretation. All of the information from manufacturing drawings is extracted manually and logged into other programs. This process is time-consuming and prone to errors, which could result in massive losses for companies. Hexagon hopes that with this project, we will be able to create an application that can recognize and extract all of the relevant information from manufacturing drawings using optical character recognition (OCR) and enter the information into other programs with minimal human interaction. This will reduce error rates and greatly improve efficiency.

Key Accomplishments

- **Testing of different OCR engines:** Tested 7 different OCR engines and applications, Tesseract, JavaOCR, ImageGear for Java, ImageGear for C/C++, FineReader, LeadTools Winforms OCR Modules and Winform OCR Advantage to determine which engine to use for the foundation of our OCR engine.

- **Evaluation of OCR engines:** Evaluated each of the OCR engines by comparing and contrasting the supported languages, the requirements for training new language, its accuracy, as well as the licensing options for the respective engines. Determined that Tesseract is a good engine to use for this project.

- **Preparing training files:** Manually generated the training text and image file for Tesseract, which includes GD&T symbols (Figure 2) of different font sizes mixed with each other. We used jTessBoxEditor to create and edit the box file, which creates the correlation from an image to a character. With each individual box relating to one instance of a character, these files were frequently above 500 characters in total.

- **Training Tesseract:** Completed the training process using jTessBoxEditor. This program allowed us to handle errors in Tesseract's training process, as well as address common mistakes by logging them as ambiguous to one another. This also helped with iterating through the training process, as a completed library could be used to help generate new training files for a future iteration.

- **Testing using test harness:** We then tested our new GD&T language library with real world examples, in the form of feature control frames (Figure 3), to find its accuracy. During testing, we recorded the accuracy of all alphanumeric symbols, as well as GD&T characters, then logged the results into a spreadsheet for documentation. We then iterated through the preparation and training processes with the goal of pushing accuracy of all characters to the desired level of 90%.

- **Integrating Tesseract into Hexagon's PC-DMIS:** Integrated Tesseract with GD&T recognition into a test harness Hexagon provided (Figure 1), with the help of Hexagon developer Robert Jurca. Through this we were able to integrate Tesseract into PC-DMIS without having to work through the program's learning curve. Hexagon is now able to release this project as a function within their next PC-DMIS software release.

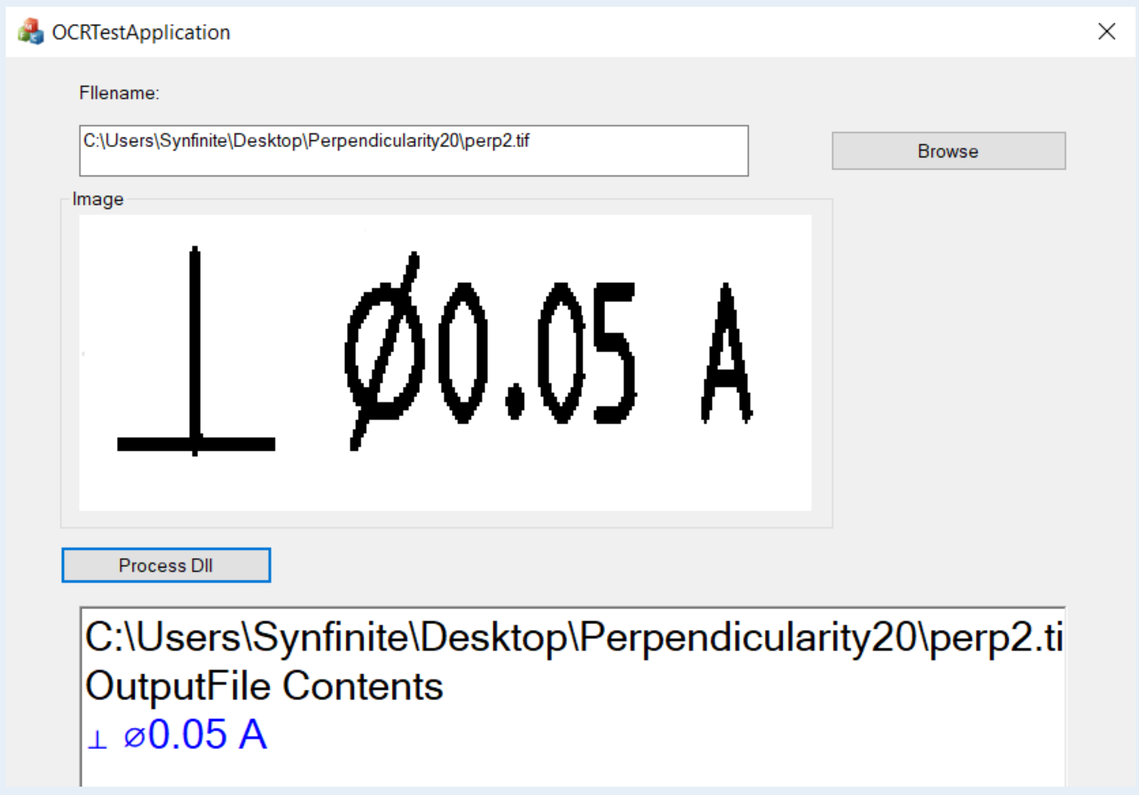

Figure 1: Screenshot example using the test application developed by Hexagon. This is used to integrate Tesseract with GD&T recognition into the company's software, PC-DMIS.

Anticipated Best Outcome

The best outcome for this project is to first train an OCR engine to recognize and extract all of the relevant GD&T symbols in the product manufacturing information with greater than 90% accuracy. The extracted information should be put into a logical format so that other programs can use it to directly create measurement routines for an automated measuring system. We would also need to integrate the OCR engine that we have trained into a standalone application as well as a mobile application, with the code being portable so that it can be used on any platform and distributed for use across multiple devices.

Project Outcome

The Best Anticipated Outcome was not achieved: the mobile application was not developed.

Figures

TYPE OF TOLERANCE	CHARACTERISTIC	SYMBOL
FORM	STRAIGHTNESS	—
	FLATNESS	▱
	CIRCULARITY	○
	CYLINDRICITY	⌭
PROFILE	PROFILE OF A LINE	⌒
	PROFILE OF A SURFACE	⌓
ORIENTATION	ANGULARITY	∠
	PERPENDICULARITY	⊥
	PARALLELISM	//
LOCATION	POSITION	⌖
	CONCENTRICITY	◎
	SYMMETRY	⌯
RUNOUT	CIRCULAR RUNOUT	↗
	TOTAL RUNOUT	⌰

Figure 2: Some of the common GD&T (Geometric Dimensioning and Tolerancing) symbols seen in feature control frames. These are the symbols we need to train Tesseract to recognize.

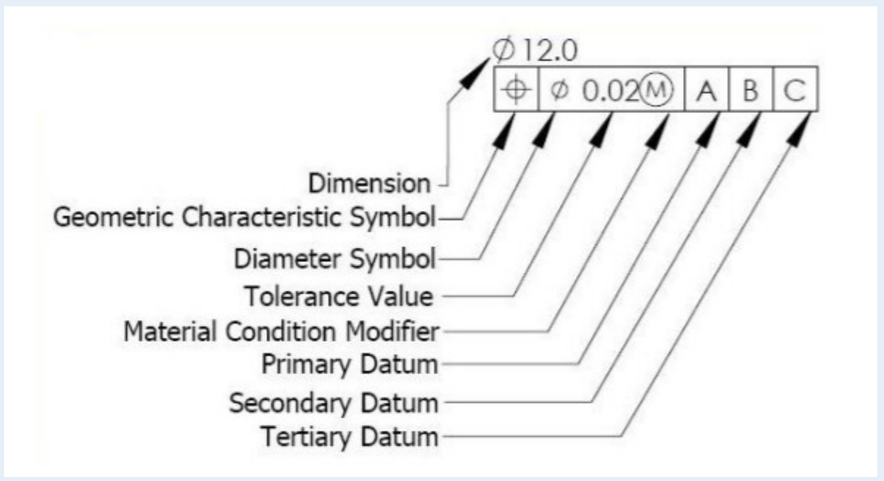

Figure 3: Example of a feature control frame used in CAD (Computer-aided Design) blueprints to describe the conditions and tolerances of a geometric control.

ELECOMP Capstone Program Director: Professor Harish R D Sunak

email: sunak@ele.uri.edu telephone: 401-874-5859 ELECOMP Website: http://web.uri.edu/elecomp-capstone