Motherboard Inspection and Testing Automation Project
UTDesign I Team 1675, Spring 2023
Mentor: Dr. Marco Tacca
Corporate Mentors: Ben Kazora, Roy Lin, Ricky Huang
Members:
Gruhit Padsala (CE)
Hrishab Desai (CE)
Nishesh Shrestha (EE)
Elijah Evans (EE)
Abhishek Thakurathi (CE)
Kennan Wu (CE)
Bhabishya Shrestha (CE)
Sponsor: SMS InfoComm

Background
SMS InfoComm currently repairs motherboards manually by hand. The process of manually inspecting and testing the motherboards is time-consuming and prone to human errors. The most difficult step in the current procedure is the identification of the motherboard model before testing.

Project Goals
• Design and build a portable mount system using a camera and lights to capture high quality images for processing
• Develop software to visually inspect a motherboard for common defects and I/O ports used for testing.
• Use text recognition software and image processing techniques to identify the model of a motherboard
• Design a test automation system

Project Overview

Image Capture
Detect and Model ID Detection
ID Text Recognition
Testing

Image Capture
When the program runs, the camera is activated. The human is prompted to place the motherboard on the base plate and click "capture image". If the model number is unreadable, the human is prompted to retake the image. After numerous failed attempts, the program asks the human to manually input a model number and then tests for that motherboard are ran.

Defect and Model ID Detection:
The defect detection algorithm was trained on a custom dataset using RoboFlow and YOLOv8. The dataset contains sample motherboards with the defect and hardware components annotated. In the dataset, there are some motherboards with no defects (golden images) and others that contain some cosmetic issue. The ID block was part of our training dataset. The program was specifically trained to detect the defects as well as identify the location of the ID block on the motherboard. Once identified, the program crops out and enlarges the ID block so it can be easily utilized by the ID recognition software.

ID Text Recognition:
When the text recognition id is called, two things are done. First, the software attempts to read the 2D barcode on the ID block to extract the motherboard ID. Next, the program will begin to send the image taken of the ID block through several preprocessing algorithms and OCR the result of each algorithm. The OCR result is then sent to a post-processing algorithm that adjusts the raw OCR data to fit the motherboard ID syntax. This post-processed result can then be sent to a matching algorithm, which checks if the OCR result matches all the syntax required and will output "match" or "does not match". Because there are several algorithms being used from the 2D barcode extraction and a multitude of OCR attempts, we can create a confidence rating on if the motherboard ID has been identified correctly.

Testing:
Once the model number is identified, the program communicates with the PXE Server to request the tests that correspond to the identified motherboard. The human is then prompted to plug the motherboard into the computer and the tests are loaded and ran. The results are sent back to the PXE server.

Results
The experiments with the sample motherboards showed promise. We were able to detect all the ports on the motherboard, as well as the motherboard ID block. The detection algorithm was fast and reliable. Defects were generally harder to detect. The text recognition results were also very good. The software is able to read 2D barcodes, as well as the text from the ID block. The OCR is generally able to detect most characters correctly, but instances where characters are detected incorrectly. A confidence rating is generated for the OCR result. The result of the 2D barcode reading is also compared with the OCR result to check for consistency.

Conclusion
The automated system we are developing provides an ideal mixture of Artificial Intelligence and human guidance. The process of model recognition and defect detection is made more efficient using the algorithms we have implemented. This in combination with the intangibles of a human mind results in a motherboard detection and testing procedure that is more coherent than the one currently used at SMS InfoComm.

Hardware Development

Materials:
The only materials that can pre-fabricated were, the base plate, some m13 screws , and the rods which are connected to the 3D-printed Rod-Mounting base.

Camera:
The camera we used is called Logitech BRIO UHD 4K Webcam (HD Auto Light Correction And Wide Field of View).

3D Printed Parts:
The mounting base for the rod as well as the camera mount. We also had to print additional mounts for our Litra light and a few parts to secure the rods, so they are anchored properly.

Lights:
We made use of the Litra Glow Premium LED Streaming Light with True Soft, adjustable monitor mount, brightness & color temp settings, desktop app control for PC/Mac

Software Development

Object Detection:
• Ultralytics YOLOv8: Open-source deep learning object detection algorithm that utilizes a fully convolutional neural network. The algorithm used a custom dataset to train a model that could be used for deployment.
• Roboflow: Application used to create a custom dataset. Images were annotated and the dataset was exported for training, validating, and testing. Extremely compatible with Ultralytics and good at creating extra images for datasets.
• Google Colab: Used to develop code using Python and Linux. Provides access to an Nvidia GPU, which was required for training a model.

Text Recognition:
• OpenCV: open-source computer vision and machine learning library that allows for image manipulation and detection. The algorithm utilizes OpenCV to output the ID string of motherboards through several preprocessing steps to extract the motherboard ID.
• Tesseract: an open-source AI developed by Google that allows for OCR. In our case, we use OCR to detect and read text from the motherboard and translate it into an ID string

Ethics
By creating a system that can automate the identification and testing of damaged motherboards, we can reduce the turnaround time for repairing motherboards and the number of human errors. Repairing more motherboards will allow us to extend the product lifetime of the motherboards. We can also save materials and production time since new motherboards will not have to be produced to replace the damaged motherboards.