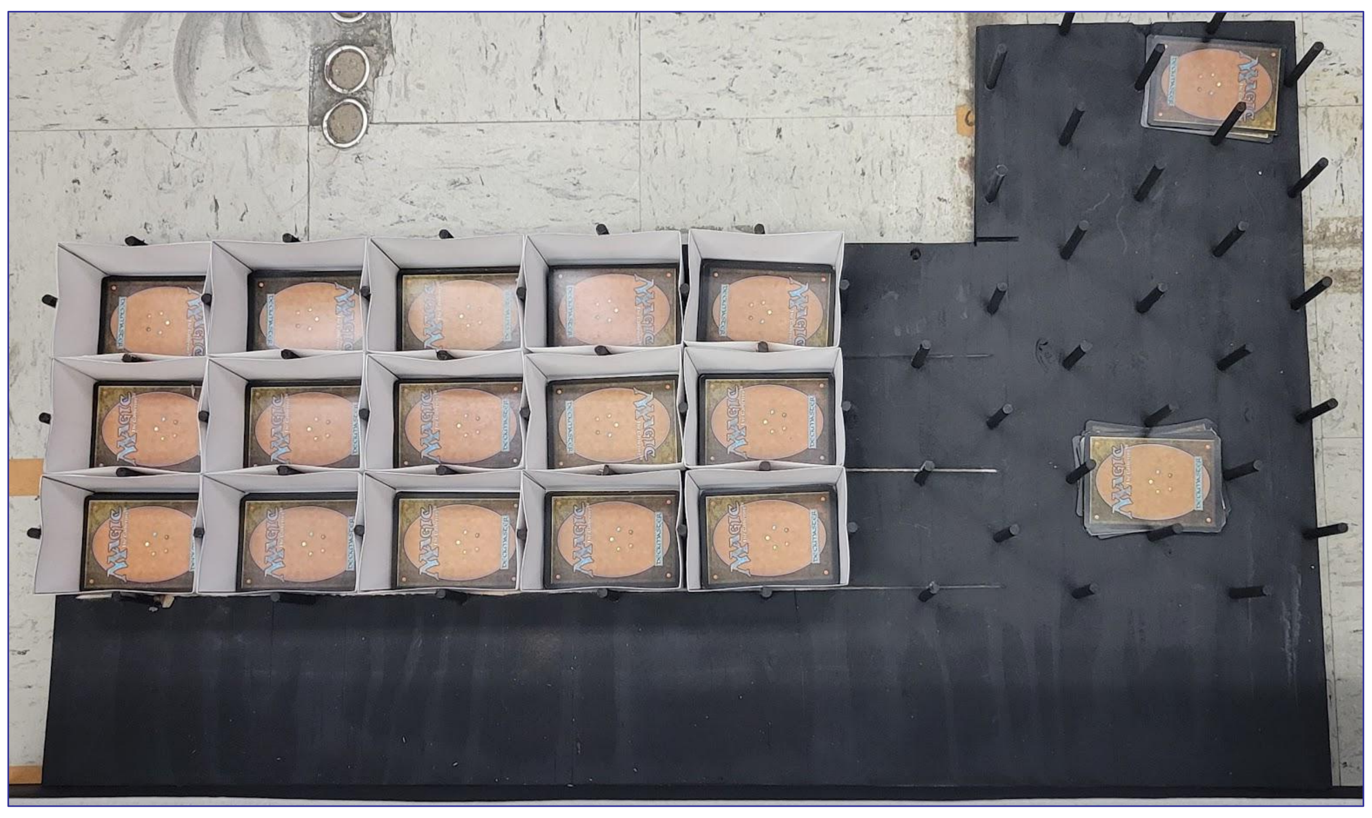
Magic: The Sortening

Henry Powell, Malcolm Okaya, Zalman Lipschitz, Liam Rees, Joseph Maloyan

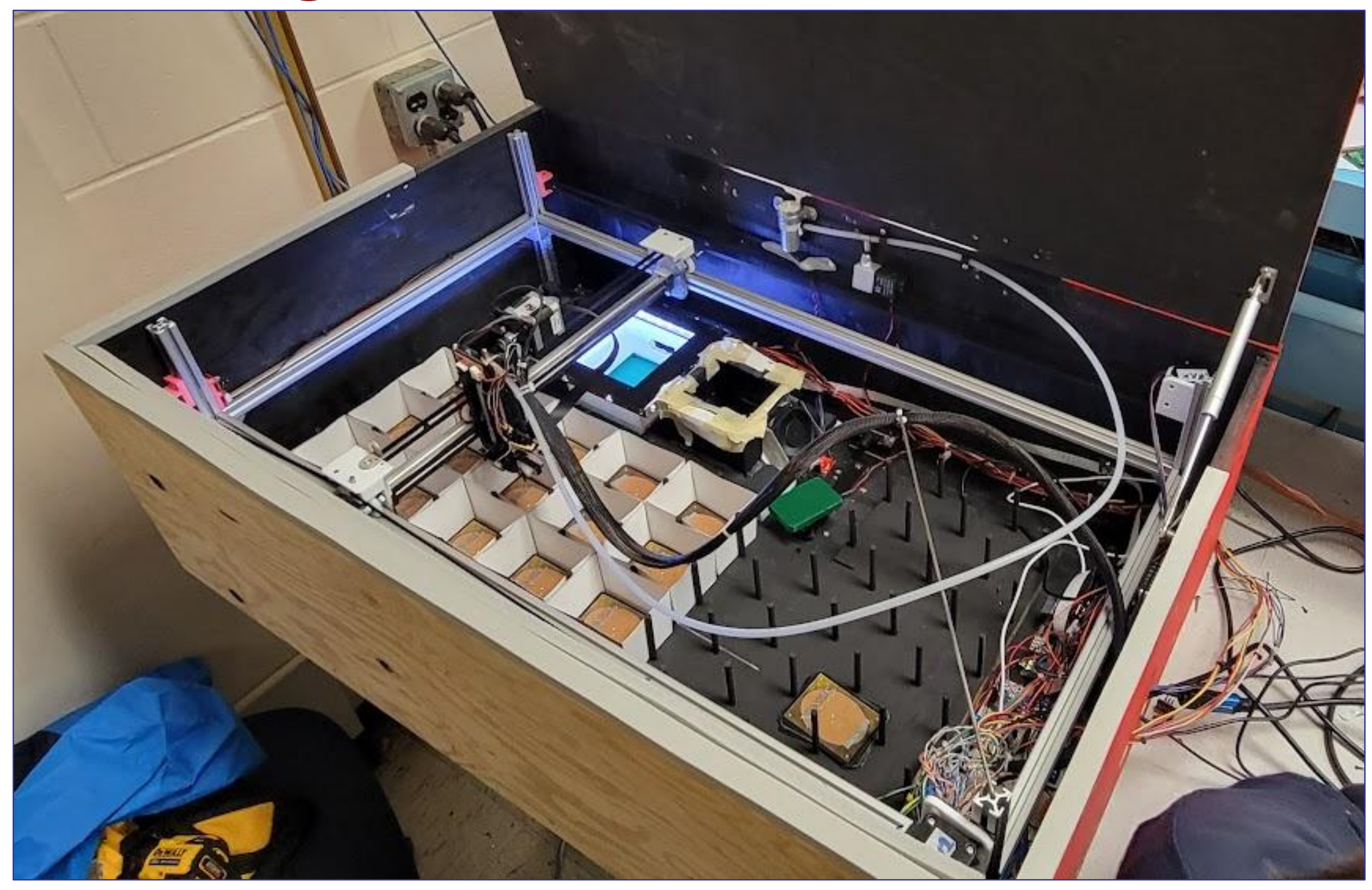
Abstract

The secondhand market for Magic cards is very large, but dealing with cards in bulk requires significant time, manual effort, and money to sort and check for damage. Current automatic sorting solutions, which are themselves quite limited, can't check for damage properly. Our solution can take a large stack of cards as an input, detect and grade each card's condition, identify the cards, check for forgeries, and sort them into several configurable bins. This allows users to properly value and sort large quantities of cards.

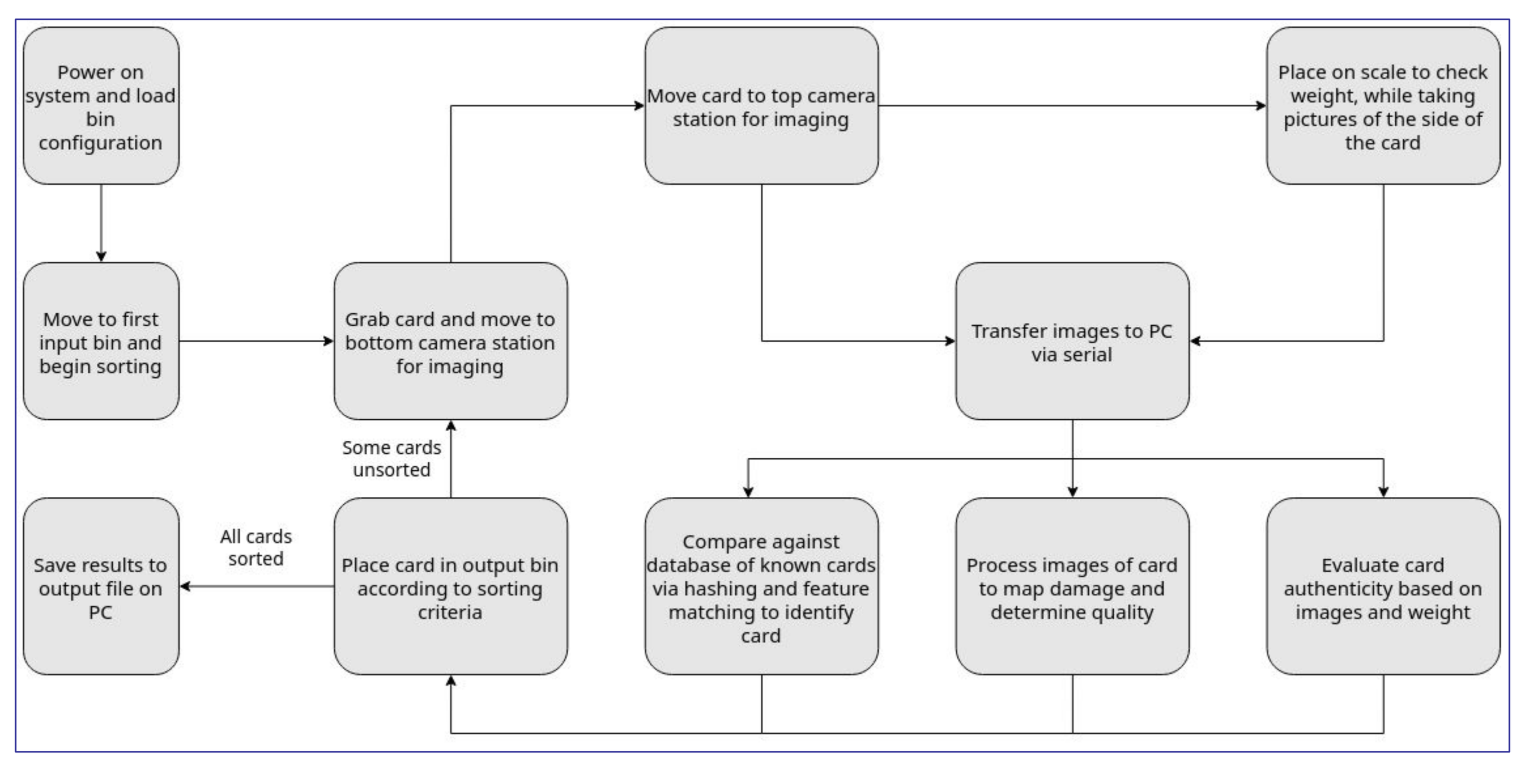
The system features 28 bins that can be configured for input or output



System Overview

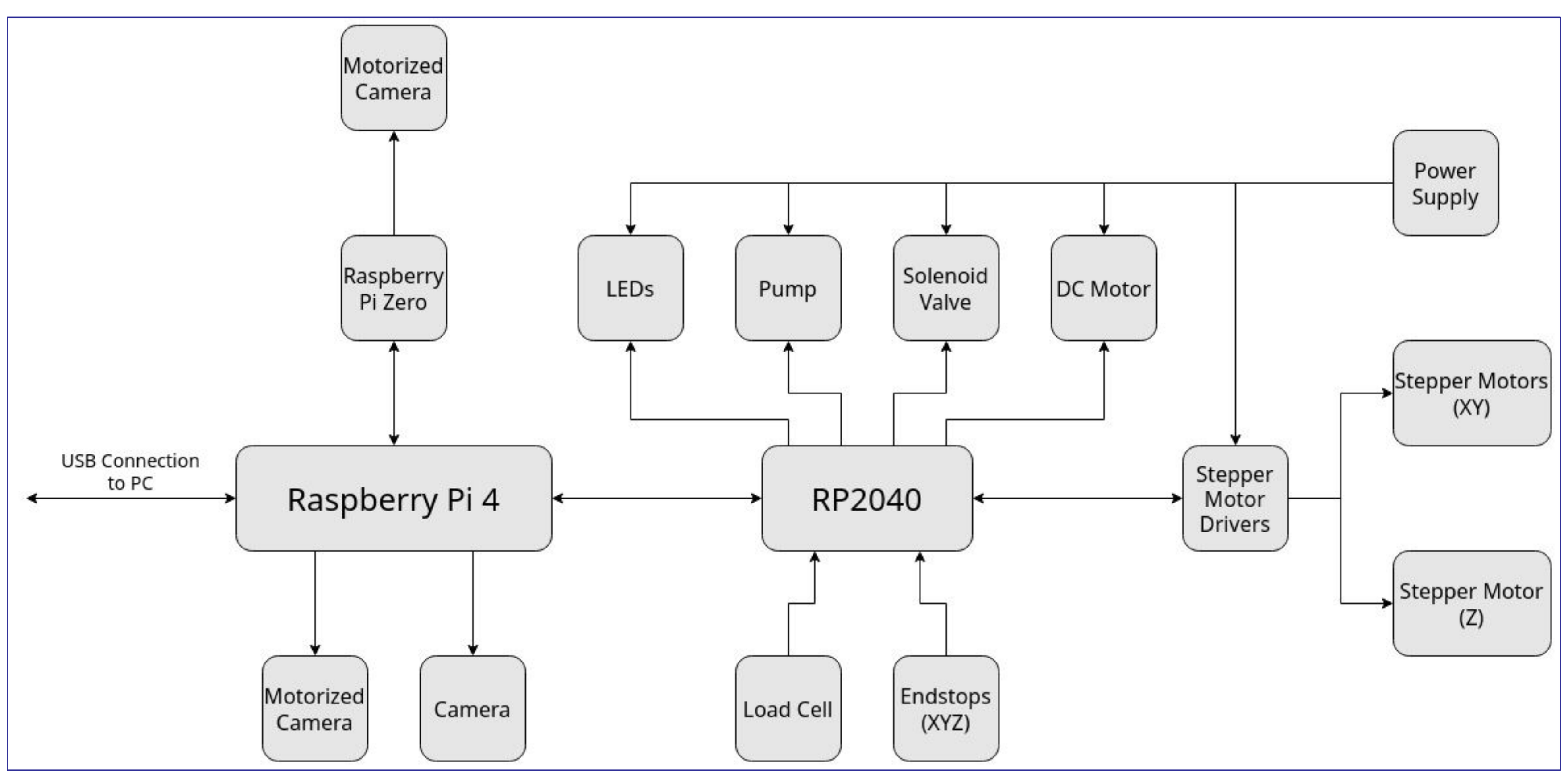


Above is an image of the system in its final state. Visible is the enclosure, each of the input/output bins, the evaluator stages, the 3-axis gantry, and the wiring/electronics.



The above diagram shows the steps of the card sorting process in software. The system block diagram is shown to the left.

Block Diagram



Specifications & Results

Item	Specification	Result
Card throughput	>2 cards/min >1000 card capacity	~70 seconds per card. 3500 card capacity.
Card condition Grading	Catch signs of damage. Provide a valid grading based on that damage.	Edge damage detection is complete and gives good results. Front/back face damage detect can generate good damage heatmaps but cannot generate a very accurate grade due to dust.
Card ID	Solution must find the name of card as well as the set it comes from. >95% accuracy	85% of cards correctly identified in most recent system test.
Forgery detection	Perform: weight check, glossiness check, printing pattern checks. 90% of fake cards should fail at least one. 90% of real cards fail none.	Weight check and printing pattern check subsystems are physically capable but are not integrated with the system software. Glossiness check was abandoned.
Card damaging	No noticeable new damage to cards from running through system.	Confirmed with stress-test by running one card through 5 times and verified with damage detection.
Output bins	Need at least 15 software-configurable output bins	28 configurable bins on the final machine.

Acknowledgements

Professor Parente demonstrated great interest in the development of the project, and encouraged us to stay on track.

Our Evaluators, Professor Gummeson and Professor Polizzi, offered good feedback and seemed to be genuinely interested by our project, which we greatly appreciated.

M5 and the All-Campus Makerspace were vital for this project. The resources, expertise, and hospitality we've received were very helpful.

Department of Electrical and Computer Engineering

ECE 415/ECE 416 – SENIOR DESIGN PROJECT 2022

College of Engineering - University of Massachusetts Amherst

SDP22

Card ID

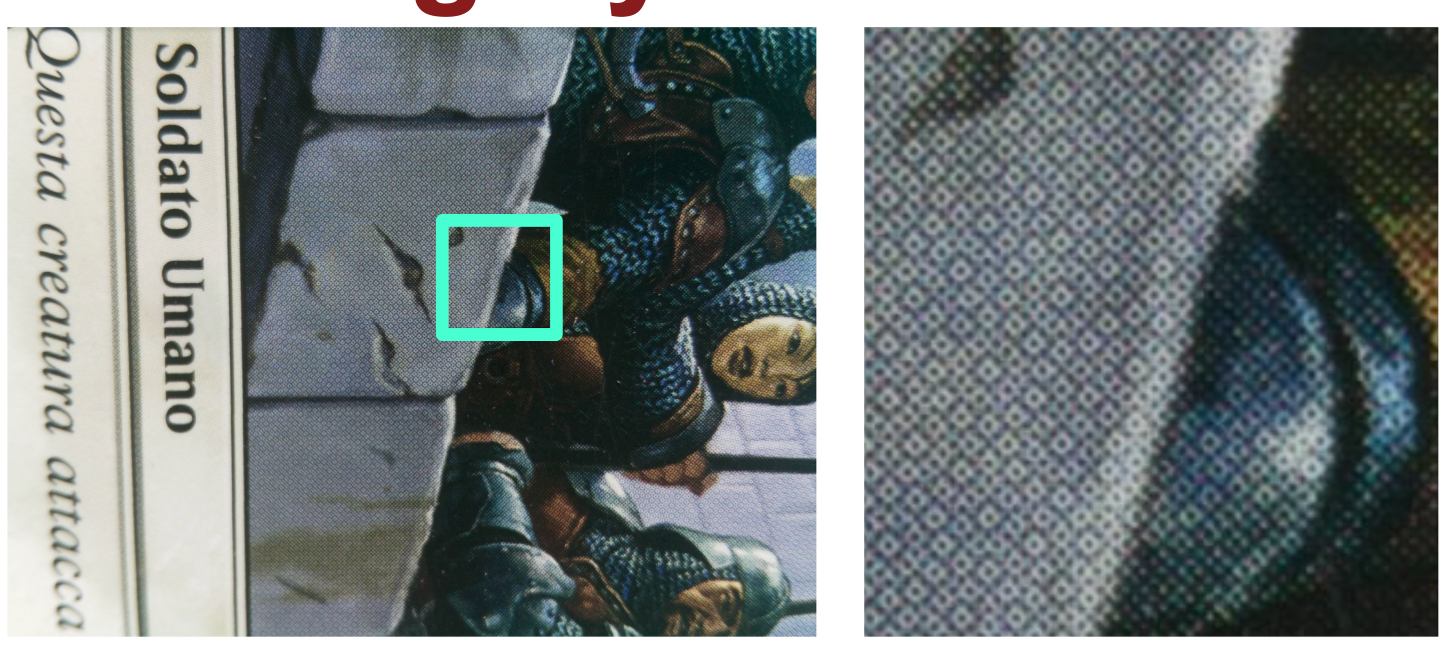
Original image shown above, matched image shown to the right

Identifying cards is an essential part of this system. This was ultimately implemented in two steps.

First, perceptual hashing is used to compare the image against a database of card images and narrow down the possibilities. The most similar cards are then compared more thoroughly using feature matching to find the best match.

While feature matching is much more accurate, it also takes significantly longer, so it is useful to use perceptual hashing first as it is much faster.

Forgery Detection



The print pattern of a legitimate *Magic* card

There are multiple ways of determining the legitimacy of a *Magic* card: measuring the weight, inspecting the surface finish (glossiness), and checking the print pattern.

Hardware for the weight check was built into the system, but the check was not implemented in software due to time constraints.

As with the weight check, the hardware for inspecting the print pattern is part of the system, but the software was not completed due to time constraints.

The surface finish check was abandoned early on due to difficulty implementing a solution that worked for all cards.

Cost

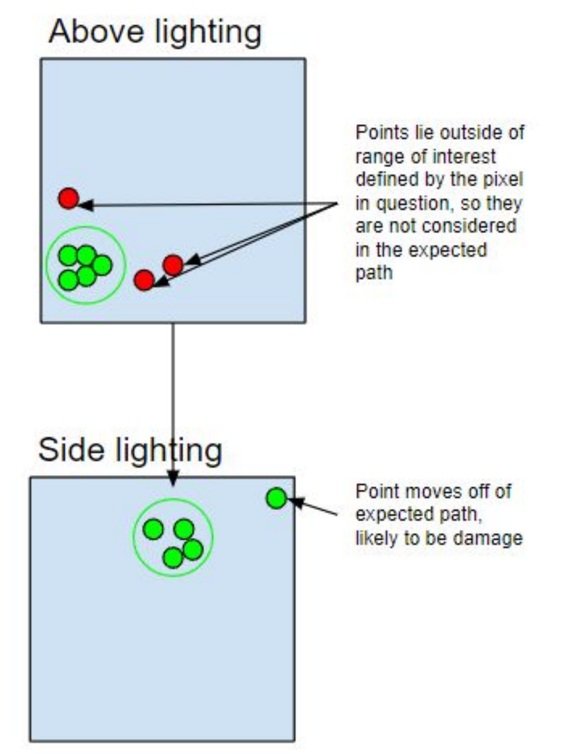
Part	Cost ($)	
	Development	Production
Stepper Motors	0	50
Endstops	0	5
Main PCB	80	4
Driver Carrier PCB	40	2
TMC2209	0	6
RP2040	10	1
12 MHz Crystal	2	0.4
128MB Flash	4	1.3
Solenoid Valve	0	13
Pump	0	9
Arducam 8MP cameras	60	60
Raspberry Pi 4	70	70
Raspberry Pi Zero	5	5
Capacitors	70	10
Resistors	3	0.5
Amplifier	24	24
Voltage Regulators	28	8
Diodes	20	7
Connectors	5	5
Power Supply	20	20
Belt	16	11
Pulleys	19	1.5
Aluminum Extrusion	83	83
Enclosure	60	60
Total	619	456.7

Damage Detection

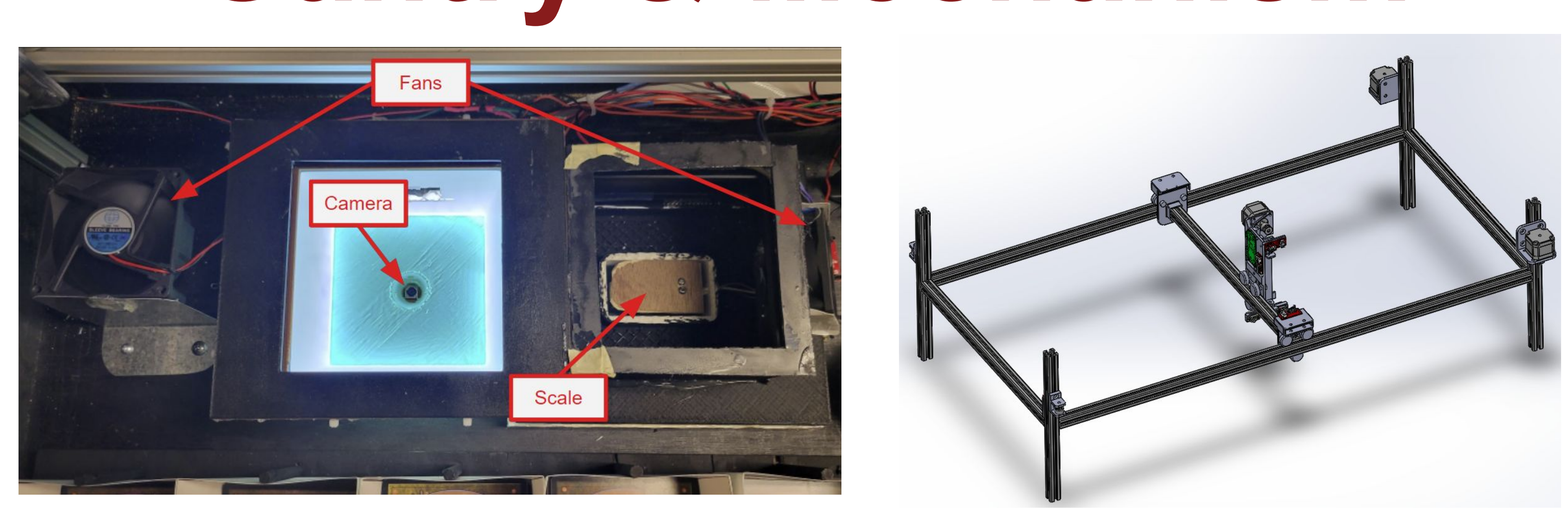
The creation of a map of surface damage (white)

The identification and mapping of edge damage (red circles)

Locating and grading the overall damage of cards is a key part of this project. This is accomplished by taking images with two different lighting conditions to map imperfections on the surface of the card. A grade can then be determined from the amount of damage and its severity.



Gantry & Mechanism

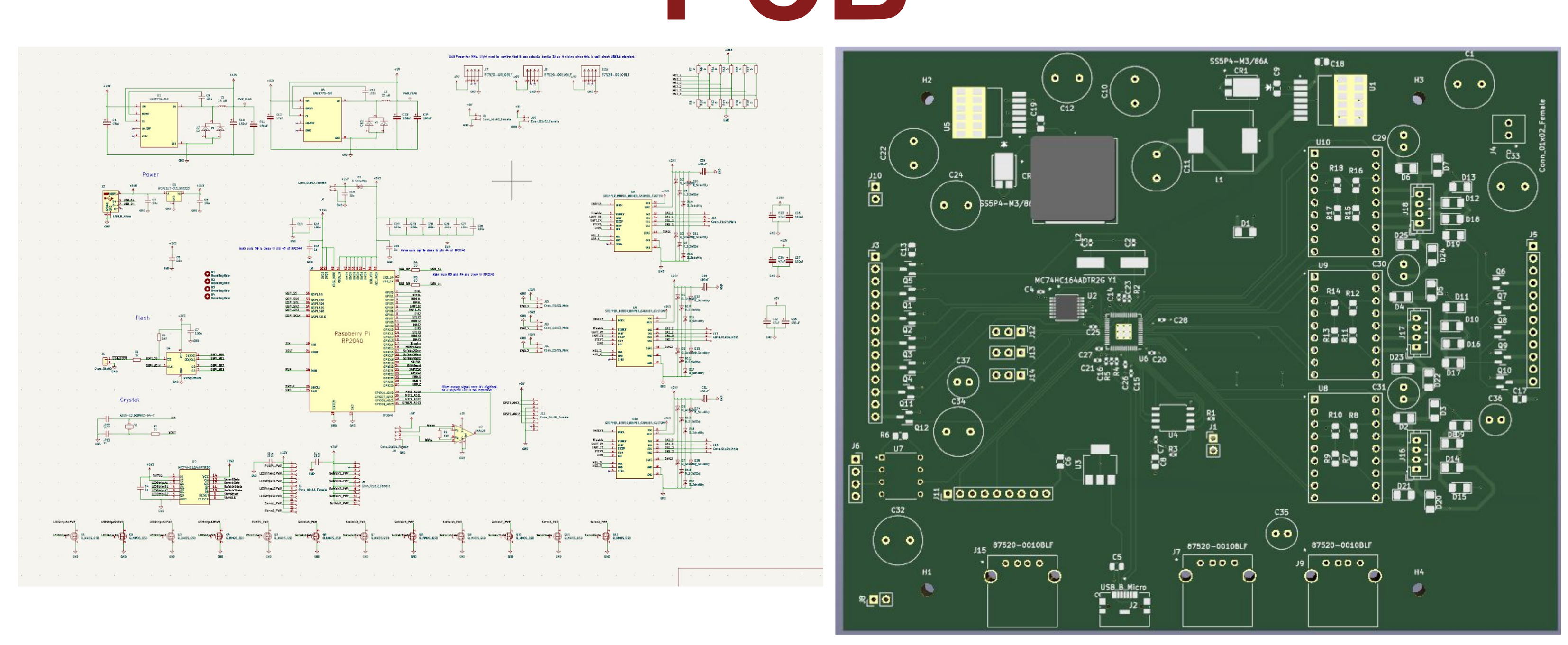


The evaluator (left) and gantry design (right)

This prototype is a pick-and-place style system based around a CoreXY gantry with a z-axis on the head. The gantry is constructed from aluminum extrusion, with 3 stepper motors that allow for the 3 degrees of freedom needed to transport cards. A vacuum pump and solenoid valve work with a suction cup to move cards around.

The evaluator consists of two stages, one for each side of the card. Each stage features a fan for removing dust to improve the accuracy of damage detection. Two sets of LEDs in each stage provide the different lighting conditions required for damage detection and card ID. The first stage features a camera facing upwards, with the card being held on the gantry head. The second stage has the card placed on a scale for forgery detection, while a second camera on the gantry head takes pictures of the other side of the card.

PCB



The full PCB layout (left) and a render of the finished product (right)

The PCB has two main functions: control the motors, lights, and sensors with the RP2040, and provide power to everything by taking the 24V input and stepping that down to 12V and 5V. Due to a possible design flaw and lack of time, the microcontroller portion of the PCB is unused in the final prototype. The rest is currently in use though, meaning the entire system can be run with one 24V power supply and a USB connection.