

## Project Reflection

At the end of this project I now realize that things that we take for convenience from things such as machine learning and A.I detection require a unseen work. Everything from Face detection to voice recognition requires tremendous amounts of training and tuning from an engineer to make sure the model preforms as accurately as possible. For this project I used something that's still having trouble being trained, Magic the Gathering cards. There is a webapp that the company that owns Magic The Gathering operates (Hasbro), this app is called 'Spelltable'. This app allowed players who couldn't really play in person between 2020 to 2022 due to the coronavirus so Hasbro implemented a way for magic players to simulate playing in person over webcam with a maximum of four players. The major draw this app had to it being to play over webcam instead of something like Zoom or Skype, was it had a feature built in where the webcam could detect a card as it was being cast or played and then present that information to all other players. This feature however was incredibly buggy, from my own experience and experiences I've heard from other players. It would often present the wrong card or no card at all.

So going into this project it was mostly a means of exploration and practice to see how and why magic cards were so hard to detect, but also to see if I could in anyway make this detection system better. To be blunt with myself, I definitely shot too close to the moon and I ended up setting my scope big by choosing to use a model to detect a hundred magic cards. By the time I got the model trained on the tenth card, it had already been two days and at that rate I would waste roughly three quarters of the semester on training alone let alone fine tuning and

actually programming the program which would detect and show the information of the cards back to the user.

After pivoting my scope from an entire deck to one card to represent each type of card in Magic the gathering, the project became much easier to actualize. To this end I also pivoted away from complex model creation methods such as using Tensorflow to preprocess the images and OpenCV to help annotating the images so they could be detected. In the end I decided to go with a friendly user facing model creation program in the form of Google's teachable machine. I was able to create and actualize 7 classes, one for each card being detected and one class which detected me holding nothing. This also showed me the difference between accuracy and speed in different models. Where Teachable machine prioritized speed in the detection of the cards and creation of the model it lacked accuracy. Whereas creating my own custom model using Tensorflow would have allowed me to test for specific things which would have led to greater accuracy, but would have taken far too long in time for this project to be completed.

Now that the project is done, I have an intermediate understanding on how machine learning and detection models work. This project gave me the confidence to think about a path potentially down this as a career path, however at the very least it gave me understanding as a coder on how to implement other forms of input other than a key press or a mouse click. As we move further towards the future I think the future will be based even less on tactile input such as buttons but things such as voice and camera will become more commonplace. Even in the space of web development, search engines have already begun deploying searches for users based on voice rather than typing out a query. I hope this understanding of how models are made and work within the space of machine learning will ultimately land me a creatively and monetarily fulfilling career path in software engineering.