



# Machine Learning & Magic (Magic The Scanning)

Sahan Jayalath



Entertainment Technology, NYCCT, Brooklyn New York 11201

## What is Machine Learning and Why use it for Magic the gathering cards?

Machine Learning is when a program is designed to take in information and then execute commands without a human giving it explicit instructions. It is a ever growing space within the software world. It has been used in applications spanning almost every industry and sector of the world. But how does it work and how can it be used by the everyday person other than asking questions over chat GPT or using filters on your face for social media? I decided to answer that question with magic! Specifically using machine learning to detect magic cards with the intention of using it to feedback information to the user with the help of a webcam.



## Creating Models with Machine Learning & the problems that came with it

In order to implement Machine Learning within a program, it needs to have some sort of basis to learn from, you can think of this as how someone would use the alphabet or dictionary to learn a language. There are several model training tools and libraries that software engineers can pick from. Some of the most popular ones are TensorFlow, Teachable Machine, Pytorch, Numpy and so many more. Initially I used Tensorflow which is a much more sophisticated model creation tool. For a project involving 'Magic The Gathering' cards, Tensorflow would ask a Software Engineer (me) to feed it dozens if not hundreds of images related to the various objects that need to be detected. However it takes incredibly long to do this with tensorflow. Halfway through development I had to switch to a different Model creation tool, 'Teachable Machine'. Teachable Machine let me build a somewhat accurate model in two weeks with a data set of seven cards



- Great for accuracy
- Highly customizable
- Can take months to create a single model (before even testing the model)

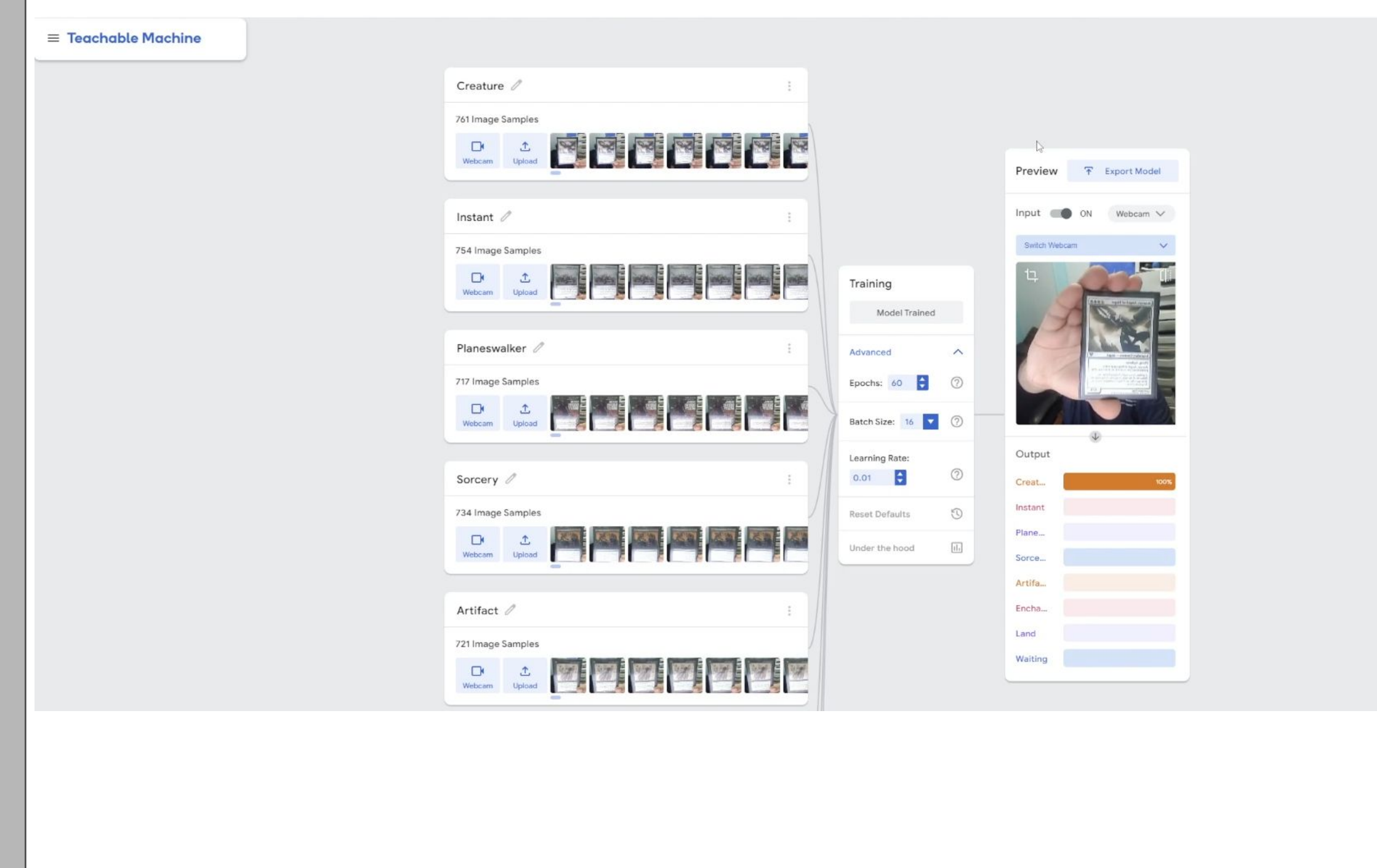
- Fast Model Creation
- Easy to input and tweak model
- Not the most accurate learning rate

## Results

After getting the trained model from Teachable Machine, I had to figure out the logic of how the program would work between recognizing the cards, determining if the card was correct and then finally displaying card information back to me. I built the program in Python through Visual Studio and was eventually able to confirm the model was connected to the seven cards.



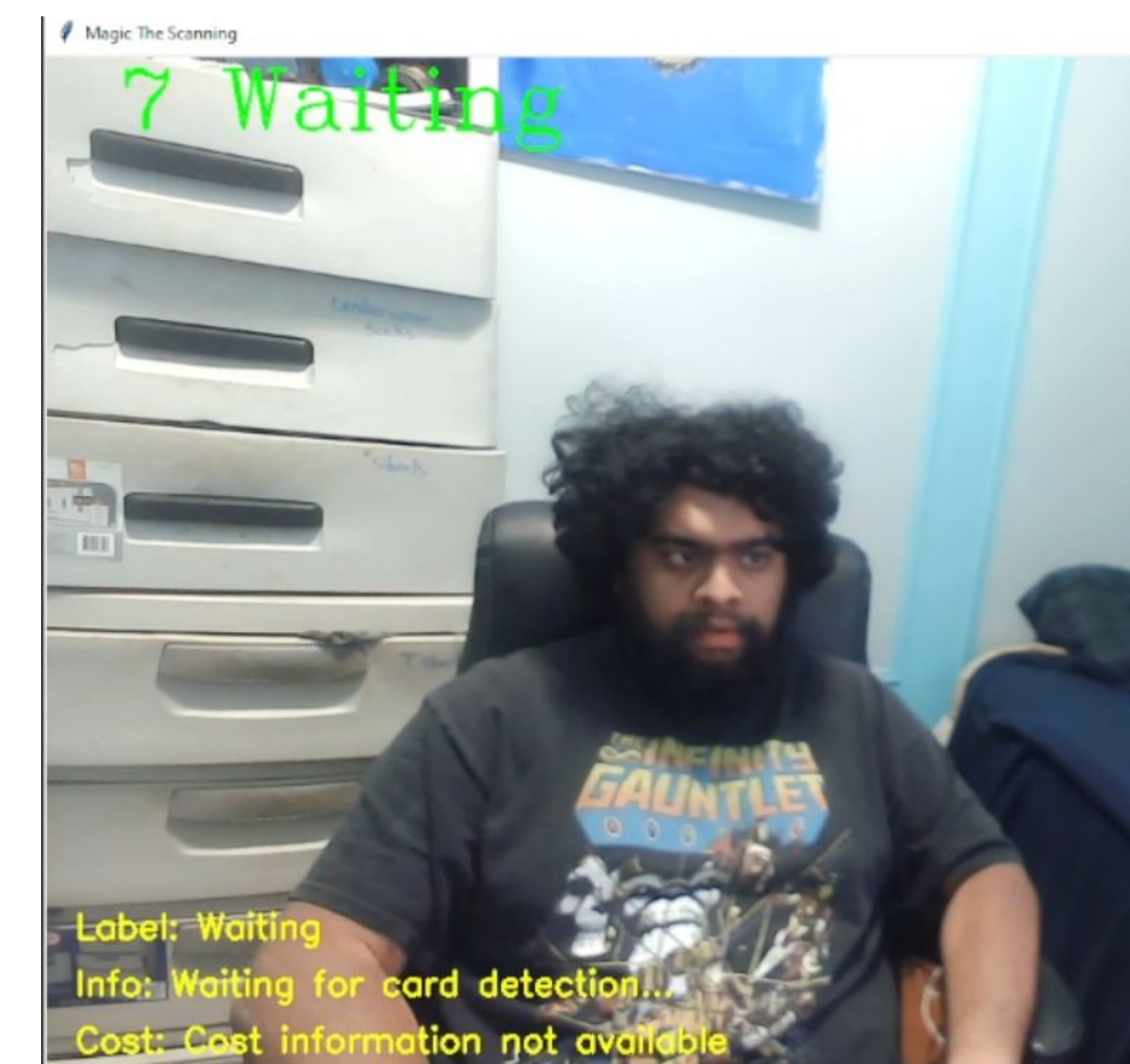
More time was spent between editing the script to be used in the python program and adjusting the model's learning rate. Eventually I was able to reach a medium where program would return the card type of one of the seven cards that was being detected (as you can see above). At some point I decided I would adjust the model to have them detect the card type instead of the the card name. This eventually became the organization system for the model and the program itself would try to detect the card type based on the model imported from teachable machine



## Actual Performance of the Program

By the time the model was actually detecting each magic card differently enough there was only two weeks before presentation. This meant I needed to find a way to take the model and now actually have it communicate back to the user when the correct card was determined from the list of available options in the model. At first the model simply gave back the user the type of card being detected. Once the program begins and the webcam window opens up, there are two major states that are being interpreted by the program

- 1) Waiting for a card to be read



- 2) A card is being read and is displaying relevant information about the card back to the user



## So how would I improve this program given more time?

If I had more time to invest into how to deploy an app using a reliable model to detect magic cards there's a series of steps that I would follow:

- 1) Find out how to reliably train a large set of objects that need to be detected and determine what is the best model training method.
- 2) Find out how I can take that same set of objects to annotate and create variations of the same image to feed into a that same model
- 3) Decide on how the interface should flow from detecting the correct card to storing its name in the memory so it could then print out useful information back to the user. Perhaps in the form of a .txt file or even sending an email with the recorded information.

## Conclusion

All in all, I think this kind of technology could be useful for things other than being made to be used for detecting magic cards. For example, half way through the semester I had to help categorize my father's medication so he could understand what time of the day he needed to take his medicine and I thought to myself, it would be incredibly easy to organize medicine if a machine could detect the name of the pill based on a model trying to figure out the correct one based on size, shape and color. Potentially this kind of tech could be used by a pharmacy to just dump in all of the pills a patient would need for that week or however long duration into a machine which would then organize small compartments and have those preorganized containers ready to go for the patient.

## Acknowledgments

Special Thanks to Professor Adam Wilson :  
For helping me understand the logic of models and troubleshoot issues during development

Special Thanks to Professor Allison Berkoy:  
For introducing me to teachable machine and giving me the opportunity to understand how it works within the context of program.