Magic The Scanning

By Sahan Jayalath

What is ‘Magic The Scanning’ About?

Magic The Scanning is an python app which asks the user to present to present ‘Magic The Gathering’ cards to the webcam in order to recognize them and display information about the card back to the user. ‘Magic The Gathering’ cards are part of a tradable card game owned by Hasbro toys. Players often construct ‘decks’ with ‘Magic The Gathering’ cards and they are meant to be played against in a setting of two or more players. I originally wanted to make this app to help keep track of my own decks using a machine learning model. Decks of 60 or 100 cards are often hard to keep track of as players (including me) will sometimes update, change or take cards from the deck. This app was originally meant to create a list of cards presented to the webcam, detect the card name based on the model created and would rapidly allow the user to catalog their cards in a text file or as an xml file.

So how would this app work?

To simply put it, this would be a python app that would use a model creation program which can be used in a python script called ‘tensorflow’. Tensorflow makes high level models based on images fed to it through various processes.

Once the images are processed they need to be reformatted to match things like the input size, normalize pixel values and convert the color space of each image .

Once the images have been processed the logic of the program would run like this:

1) The model is loaded in the python script

2) A feed from the webcam is being fed into the program which takes every frame and uses it to detect

Something from the given model

3) based on a accuracy or confidence rating, the program would try to predict what class or label the card is

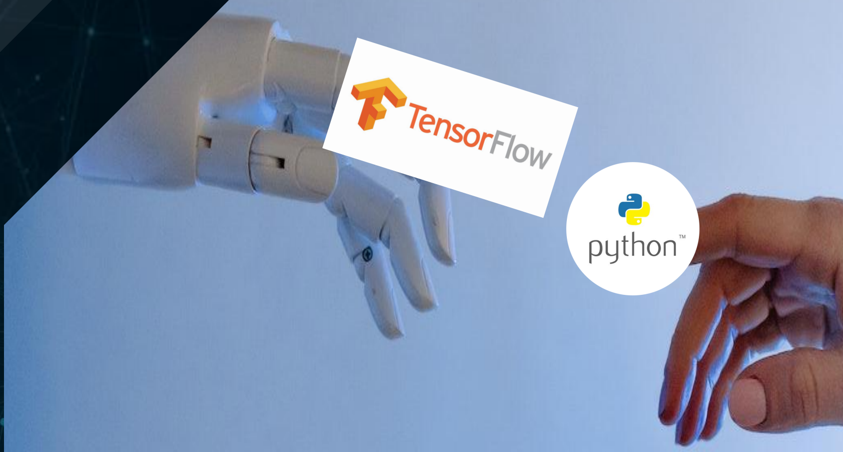
Being shown to it, if not it would return a default state where the program is detecting no cards.

4) Once the model reaches a certain confidence or accuracy rating, it displays the results of the label that is

being detected.

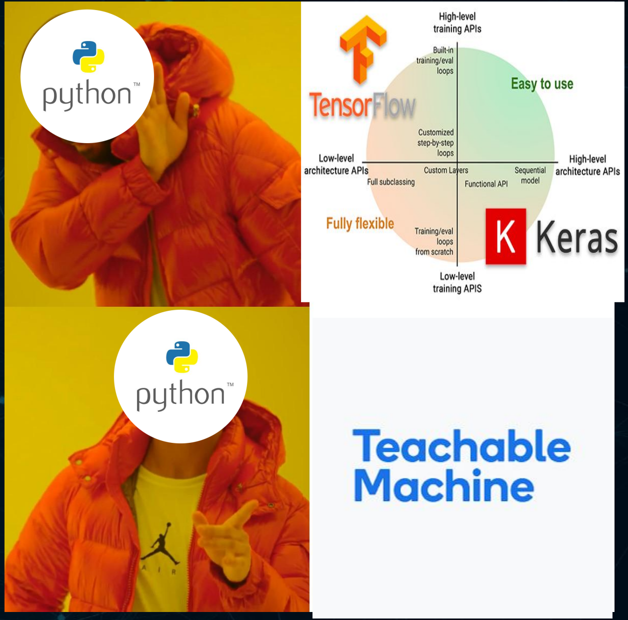
Once the program is running, those four steps happen every frame and because the model runs in at a rate

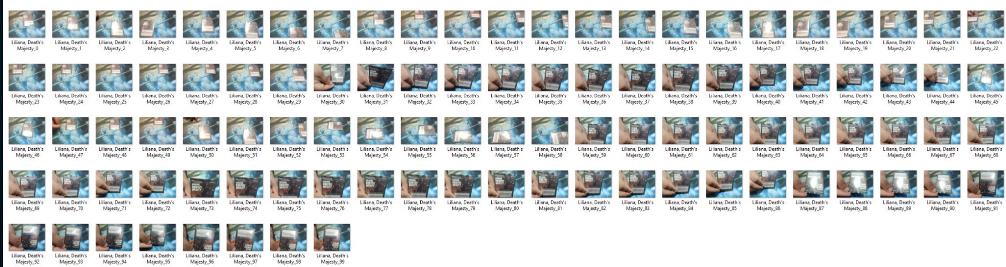
of 30 frames per second, that means these four steps happen 30 times every second.



What went wrong & The Pivot

Half way through the project it quickly dawned on me that making a custom model through tensorflow (with the help of Keras) took a considerable time and effort to feed and sanitize the image data for each class. Even within the scope of seven cards for this project, it took about two days to properly distinguish one card from another when testing in the model within a simple test run of the program. At this point i had to make a decision to either make an accurate model or one that could take in all of the cards i was using for the project. So i needed a way to quickly, train a model, sanitize the images and then annotate them so the webcam could recognize them from something else? So I turned to Teachable Machine





Why teachable machine was (kinda) perfect for this project

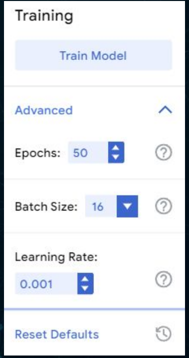
Teachable machine allowed me to quickly use my webcam as an image classifier to take more than 700 images for each card fairly quickly. Once I set up the 7 classes (one for each card that the model has been trained to detect) I was able to adjust two important factors during the training process:

1) Epochs: refers to the amount of time the model will run through your entire set when your program is set to recognize something in a frame

2) Learning Rate: Refers to how the model will learn to distinguish one class from another.

Adjusting these two factors, along with finding the right amount of cards to feed the model for each card, took roughly about a week of adjusting. Once the model was exported. I could finally create a python script to better illustrate how the app worked.

People with no idea about AI saying it will take over the world:
My Neural Network:
Dog


Training

Train Model

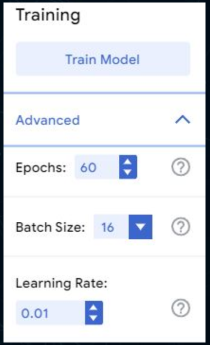
Advanced

Epochs: 50

Batch Size: 16

Learning Rate: 0.001

Reset Defaults


Training

Train Model

Advanced

Epochs: 60

Batch Size: 16

Learning Rate: 0.01

Demo Time

How would I complete this project?

If I had more time to invest into how to deploy an app using a reliable model to detect magic cards theres 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.

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.

Thank You!

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.

Additional Resources:

https://www.youtube.com/watch?v=BZGhRSajybk&ab_channel=Ethan

https://www.youtube.com/watch?v=g4qADQE3IBA&ab_channel=AzuTechnology