Assignment 4

April 22, 2019

Sharat Vyas

ITCS 5152

# Introduction:

The goal of this project was to classify various Pokémon and detect various attributes from Pokémon Go screenshots. Some of the attributes we had to pick up included HP, CP and Stardust. We were free to go about choosing any algorithms/methods to classify the Pokémon and detect the required attributes. I decided to build a neural network to classify the Pokémon and decided to use optical character recognition to detect the attributes. I decided to model my neural network architecture after the [sample architecture](https://www.mathworks.com/help/deeplearning/examples/create-simple-deep-learning-network-for-classification.html) provided by MATLAB in their documentation. For the detection of the attributes, I decided to use MATLAB’s ocr() function. A detailed explanation follows.

# Approach:

I began by breaking the problem space into multiple parts. First, I knew I had to train a classifier to be able to look at an image of a Pokémon and identify it. I decided to try using a neural network to do this. To classify the Pokémon, first I resize the input image to 600x1100. Then I crop the image to be of approx. size 315x330. I made the assumption that since all the images are the same size, the image I crop should always contain just the Pokémon and nothing else that can impact the image being fed to the model. I then fed this image to my model for classification.

My model architecture was simple. It was composed of 6 convolutional layers, where each layer is followed by a max pooling layer. It then fed a fully connected layer with output size 139 since there were a total of 139 possible classes. When I initially started my model, the accuracy was about 30% but by the end I was almost at 80%. I did a few things that drastically improved my accuracy. First, I tuned a lot of the parameters I was using including the learning rate, the solver being used, and the max epochs. However, the biggest strides came when I decided to generate more training data. I wrote up a script that took every training image, applied various levels of gaussian filtering to the image and then saved a new version with the id not changing. What this did was increase the number of images I could train my model on as well as make my model much more robust to the distortions from the validation set images.

To pick up the textual attributes like CP and HP, I decided to use MATLAB’s optical character recognition. I initially thought that applying OCR on the whole image was a good idea, but I noticed that I was picking up a lot of noise, as a result I decided to apply OCR on subsets of the image. To pick up the HP, I decided to take the upper ~30% of the image, binarize the image to make the HP value very clear and then apply OCR. Once OCR was applied, I used regular expression to try and select the HP value. This approach helped me get to about 77% accuracy. To pick up the HP, I applied a similar approach where I would crop a portion of the middle of the image and apply OCR. With HP and stardust I had a much more difficult time picking up the values. I was encountering a lot of noise, eventually I was able to get accuracy for HP to almost 60% and stardust to roughly 48%.

The last part of the assignment was to identify the location of the Pokémon’s level. To do this, I used MATLAB’s imfindcircles() method. I fed the method a subset of the upper half of the image and specified that I was looking for very bright circles. This seemed to do the trick for me.

# Examples:



Sample Input Image



Image Fed to MATLAB OCR to find Pokémon’s Level

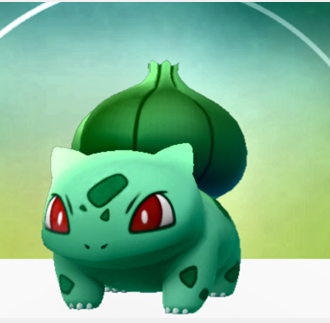


Image Fed to model to classify Pokémon and return the Pokémon’s ID



Image Fed to MATLAB OCR to get the Pokémon’s CP



Image Fed to MATLAB OCR to get the Pokémon’s HP

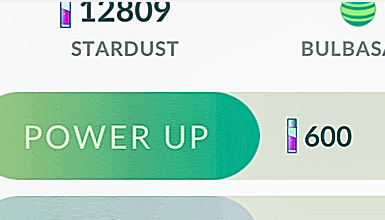


Image Fed to MATLAB OCR to get the Pokémon’s stardust

# Difficulties:

I faced quite a few difficulties on this project. First, I had a very difficult time dealing with the noise when using MATLAB’s OCR. I noticed many occasions where numbers like 0 or 2 would be read in as an O or Z. This caused many failures when trying to use regular expressions to pick up numbers like HP or stardust. Another difficulty I faced was getting the accuracy of my model to exceed 80%. I tried creating more image distortions to make the model better at handling various types of noise, but it wouldn’t cross the 80% mark. I also tried tuning the actual architecture but couldn’t cross the threshold.

# Conclusion:

Overall this project was a very interesting one because it was the first time, we were trying to extract multiple things from an image. I think that using a neural network was a very cool experience. From my readings, I see it could’ve been applied on all our previous projects as well and would likely have performed better that the chosen methods of implementation. If I had some more time to work on this assignment, I would have likely tried to use neural networks to do the optical character recognition myself as well. I think training the network on the fonts that were present on the screenshots would’ve helped my improve my accuracy, especially on picking up values like stardust.