Final Project

**Pokemon classification, information extraction.**

In the given Assignment, the images, screenshots of the Pokemon GO app are presented with the challenge of classifying the test images, detecting and interpreting the text values for CP, HP and Stardust, as well as the center of the ‘CP Semicircle’ and the Current Level of the CP semicircle.

**Classification System:**

In order to solve this assignment, the technique used is a modified Bag-Of-Words that also incorporates Color Features. This Classification system is utilized to find the ID of the Pokemon, which corresponds to the species, and answers the question “Which Pokemon is this”

Similar Bag of Words systems are also utilized to work with other parts of the image in order to train and understand the CP, HP, and Stardust Values.

This assignment involves the use of images to train and obtain a vocabulary of features. Each image is split into sub images in order to focus the training of classifiers only on relevant image details.

In order to explain the process of how this Classification is done, it is necessary to draw an analogy to the bag of words model for data storage.  
  
In bag of words for text data, we store each document as a list of unique words in the document along with the frequency of occurrence of each word. As an analogy, in order to make a bag of words for image data we need to come up with a list of unique ‘visual words’ and count their frequency in each image.

These visual words can be a set of features extracted from the image. For example: Certain images like in the Bulbasaur category (ID=001) and the Ivysaur (ID=002) category may have some similar features which can be considered as the same visual words. In this way we can form a vocabulary with which we can classify the images using Bag-of-Words classifiers.

The Feature extraction in this case may not be sufficient to get good accuracy, since Pokemon are cartoons and when feature extraction occurs in grayscale images, the cartoons tend to have similar image features (eyes, tails, teeth, may all be considered visually similar without taking color into consideration). Thus we need to modify this Bag-Of Words classifier to use color as well.

Step 1- Extracting features.

One of the most effective methods for feature extraction is the SURF feature extraction. Using   
the detectSURFFeatures we obtain the points representing interesting points in the image. From this we use extract features function to get the local features at that point.

First we crop out the images into sub-images in order to focus on the important parts of the image. To do this, images are re-sized and the relevant regions of the image are cropped and stored separately. In doing so we develop sets of images- a set of Pokemon images, a set of CP images, a set of HP images and a set of Stardust images.

We run this feature extraction for all the images in the data set and make an array of all these 1087 images. This is the full set of the features. From this we can develop a bag of words.

Step 2- Creating Bag of Words

Now that we have the set of features we can use K means clustering on the set of features in order to get a set of K clusters. Each of these clusters represents the set of features that are similar to each other and can be considered a visual word. This process generates a set of K visual words.

We convert each image from a set of features to a 1 dimensional vector where each column refers to a single visual word. If the feature is similar to the K’th cluster, the value of the Kth column is incremented to show the frequency of visual words in the vector.

To do this it is easier to use the inbuilt hist() method to create a histogram on the cluster data with respect to the K centroids. In doing so we can generate a bag of words that describes the entire data set.

Step 3- Color Features.

Now that we have a histogram that represents the grayscale features of the image, it is important to also take into consideration the Color features of the image. This is especially important for Pokemon detection because color data is an important differentiating factor between multiple Pokemon.

This can be done by calculating and adding some color features. To do so we can start by splitting the image into 3 channels, R, G and B. Once the image is split into channels we can calculate the mean values of each of these channels and then normalize it by dividing by 255. These normalized features can be appended to the histogram as color features of the image.

Additionally, another possible color feature is the calculation of Otsu thresholding values of the R, G and B channels of the images. This is also Normalized and then used as a color image feature.

We also perform binarization of part of the image in order to better isolate the CP value. To do this we convert the image to binary with a threshold of 0.95 so that the white parts of the image are only remaining. This white part of the image is the CP Text value. Once this value is isolated we are able to get a better result as the background noise features and colors are discarded.

Step 4- Training Models.

Once we are able to obtain a histogram of features, and we also have vectors containing label values of images for each criteria of information, we are able to train the classifiers with the histograms and the corresponding labels. The model finally saved contains a classifier for CP, for HP, for ID value and for Stardust value. This classifier is used in the testing stage.

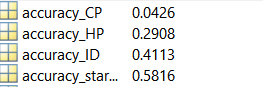
Step 5- Testing

Now that the bag of words has been created we can use the K nearest neighbours algorithm to calculate distance between any test data (which has been processed in the same way and converted into a histogram of features) and all the vectors in the training data.

We find the K vectors that have the lowest distance which implies high similarity. Out of these K vectors we find which label has the highest occurrence and we assign this label to the test data.

Aside from this we also need to find the circle position for the Level value and the circle position for the level semicircle. We do this using the imFindcircles method on the thresholded binarized image, after performing erosion in the case of the small circle, so that the unnecessary circles are discarded from the image. The range of circle sizes is specified as a ratio of image size itself.

The accuracy for this system has been found to be the following:



**Challenges**

**Lot of the images are unusable as they have been deliberately saved in an Indexed colorspace in order to make them more difficult to work with.**

A workaround is possible in order to re-obtain the images as RGB but this involves the use of the map from the original image file. As a workaround I have saved one of the maps into the model in order to allow the pokemon\_stats function to reform the Indexed images as RGB

It is challenging to work with CP values as in some cases they are blocked by Pokemon  
Aside from that, since there is a very very wide range of possible CP values it may not be the best solution to let the bag of words be used for finding CP.

Stardust and HP are similar numerical values but the accuracy is better since the number of possible categories for those values are somewhat less.

The selection of K values for the number of cluster centers, the selection of number of features extracted from each image, the selection of k as the number of nearest neighbours to be used in the K Nearest Neighbours algorithm are the main parameters that required a lot of tweaking and tuning. Depending on the values of these parameters there is significant amount of difference in the result.