Computer Vision – ITCS 5152

Term Project

ALGORITHMS:

ID Detection:

1. A small patch is considered from the image around the pokemon.
2. On the considered patch, I extracted HOG Features after resizing it to 64 x 64.
3. While training, I saved those features and corresponding labels.
4. While testing, the features extracted at step 2 is done similarity check with the features in the model. The highest similar feature's (in model) label is assigned to the current image.

Level Detection:

1. During training, I considered a color histogram of the patch that contains the level circle. Also, build the SVM classifier on the color histograms of the detected circle points for level, labelling accordingly.
2. First using imfindcircles on the filtered image using Gaussian filter with sigma 1, I detected several circle points and on these points. I perform similarity check with the color histogram of the first step. The one with highest similarity is considered.
3. Now, I increase the sigma value for Gaussian filter by 1 and then perform the step 2 until the sigma value reaches the threshold i.e. 10.
4. Considering all the local maximums collected at step 2, I perform the prediction using the SVM classifier built earlier, such that the one that is classified as positive sample is taken as final level value.

Circle Center Detection:

1. First, I'm considering the x value for center as half of the width of image.
2. Using two different templates, I considered the y value for the center.

CP/HP/SD Detection:

1. First, I consider the corresponding patch, where the values of CP/HP/SD respectively exist using corresponding templates.
2. Then calls the function to implement number detection on that patch.
3. In case of stardust the value returned from the above step is checked for the power if it is less than 100 i.e. number having less than 3 digits and if it is I added 2 trailing zeros to it using the assumption that the value for stardust will always be having at least 3 digits and in some cases (blur images) if zeros appear side to each other they can't be detected individual using connected component approach.
4. In case of HP the value returned is checked for number of digits and from observation it should be even, as I considered the entire patch where HP value exists and that the HP number is written two times separated by a '/'. Even in this case also I'm evaluating all the possible number of combinations and considering the maximum among them.

Number Detection:

1. For the incoming patch, I apply the binarizing function using several threshold values iteratively.
2. Since for the HP and stardust the numbers look dark than background therefore after the binarizing function, I inverted the intensity values i.e. 0 to 1 and 1 to 0.

3. After binarizing the patch, I perform connected component method to extract the individual characters in that patch.
4. For each individual character, it is isolated such that only that component exists and then resizing it into 32 x 32 image.
5. While training, I stored a feature vector of that resized image for each number (1 to 10).
6. For feature construction from the resized patch, I divided the image into 64 disjoint cells each of 4 x 4 grid.
7. From this 4 x 4, I counted the sum of the intensity pixels and considering a single value out of 16 values. The reason behind choosing this method is because it preserves the non-zero values in the positions that depicts the digit shape. Thus, I prepared a feature vector of 1 x 64 size.
8. Using that model, I compare the feature vectors of each individual component after resizing it with that model using similarity check.
9. The label for ones with the similarity measure greater than the 0.85 is considered and based on the x position of the individual component the number is formed.
10. Since I iterate among the multiple threshold values, I will be having several numbers and I preferred maximum number from them to be the final value. The reason to choose maximum is because if the threshold values might be too high or too low to correctly recognize the numbers (less digits might get recognized) and since the patch I’m considering dosen’t have any other numbers and false positives (avoided using 0.85 as filter).