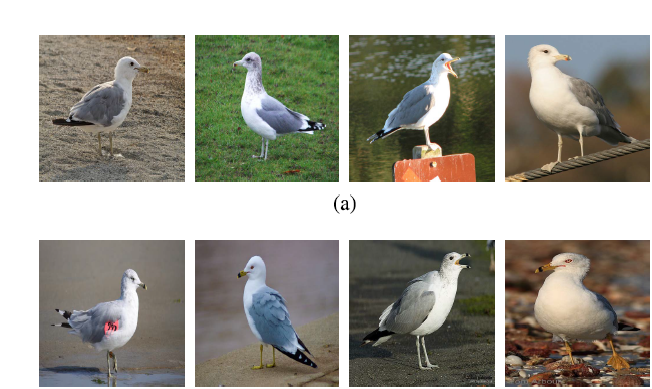
Fine-Grained Image Classification Using Modified DCNNs Trained by Cascaded Softmax and Generalized Large-Margin Losses

The Fine-Grained Image Classification task focuses on differentiating between hard-to-distinguish two different objects due to more similarity, such as species of birds, flowers, or animals; and identifying the makes or models of vehicles. In real world some bird’s species are there which are more similar and belongs to different categories and it’s hard for normal classifier to predict them into different categories or tell to computer which parts of them are similar. To overcome from such issue Deep Convolution Neural Networks (DCCN) introduce, this DCCN technique require manual hand written parts data to classify them into different categories and this manual work is not possible when dataset increases.

To overcome from above issue author has introduce modified DCNN train with softmax and Generalized Large-Margin concept to automatically predict hard to classify images. In paper they have given some birds images which are too much similar and the difference is in their beak and based on that difference only the propose classifier will predict them.



In above images we can see different birds are looking similar to one and other and there is difference only in their beak. When classifying such images with normal classifier then it will predict them all in same class and this problem can be avoided using propose modified DCCN concept.

Such above example images we can find in almost all fields take car example where different versions of same model will have high similarity and classifier has to predict which part of that image is similar not all parts.

To implement above concept this paper implements H level hierarchy where all possible versions of same car model or birds species will put in same class and all H levels will consider as root and all same models will be images and consider as leaf of that root. Similarly all images will be arrange inside H level hierarchy. All such difficult to distinguish images are called as fined grained images.

DCNN model train them with the cascaded softmax loss which will put more similar object into same class by calculating similarity using Euclidean Distance function and generalized large-margin (GLM) loss, to make the given DCNN model explicitly explore the hierarchical label structure and the similarity regularities of the fine-grained image classes. The GLM loss explicitly not only reduces between-class similarity and within-class variance of the learned features by DCNN models but also makes the subclasses belonging to the same coarse class be more similar to each other than those belonging to different coarse classes in the feature space.

Algorithm Explanation

//in below steps algorithm taking all images as training data

//max iteration to complete training part

Input: Training set T , hyperparameters λ, α1 and α2, maximum number of iterations Imax , and counter iter = 0.

//return training model as output

Output: W.

1: Select a training mini-batch from T . //while training it will choose batch size of 10 or anything based on input

//compute convolution neural network at each layer for similarity

2: Perform the forward propagation, for each sample, computing the activations of all layers.

//here using fc8 and fc9 algorithm tries to reduce error by choosing more similar images into one class by going backward till error rate reduces.

//if more similar images in one group then error rate will reduce

3: Compute the error flows of fc9 from the softmax loss (for

coarse classes). Then compute the error flows of layer

fc7 and fc8’ from layer fc9 by backward propagation,

respectively.

//tries to reduce error by finding fine grained similar images

4: Compute the error flows of fc8’ from the softmax loss (for

fine-grained classes).

//repeat below steps till the error rate goes down to max level

5: Compute the total error flows of fc8’, which is the summation of those from fc9 and the softmax loss (for fine-grained

classes). Then compute the error flows of layer fc7 from

layer fc8’ by the BP algorithm.

6: Calculate the error flows of layer fc7 from the GLM loss

according to Eq. (19) and the scaling factor λ.

7: Calculate the total error flows of layer fc7, which is the

summation of those from fc8’, fc9 and GLM loss.

8: Perform the back-propagation from layer fc7 to layer

conv1, sequentially computing the error flows of these

layers by BP algorithm.

9: According to the activations and error flows of all layers,

compute ∂L

∂W by BP algorithm.

10: Update W by gradient descent algorithm.

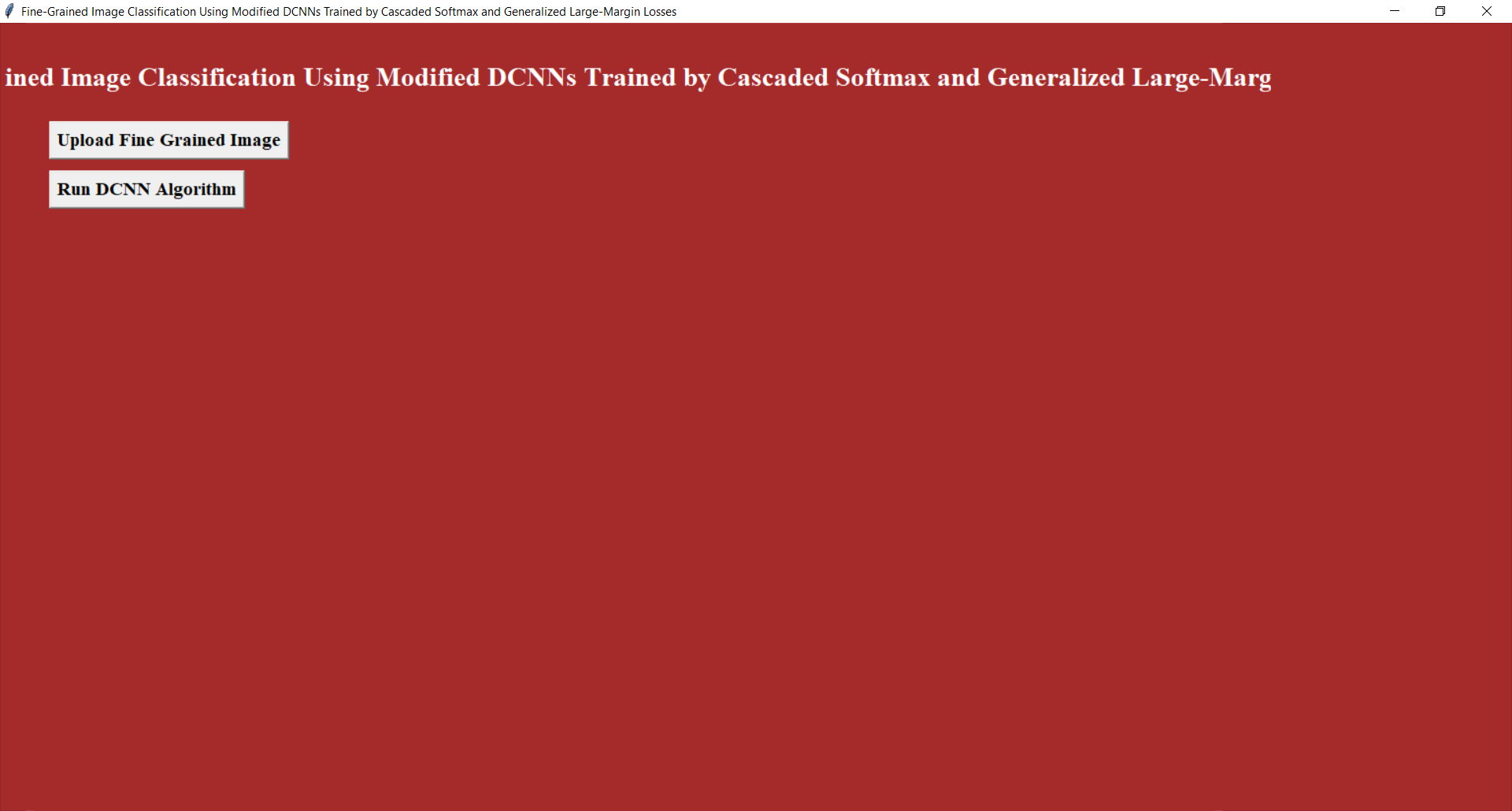
11: iter ← iter + 1. If iter < Imax , perform step 1.

Screen shots to run this project install below packages

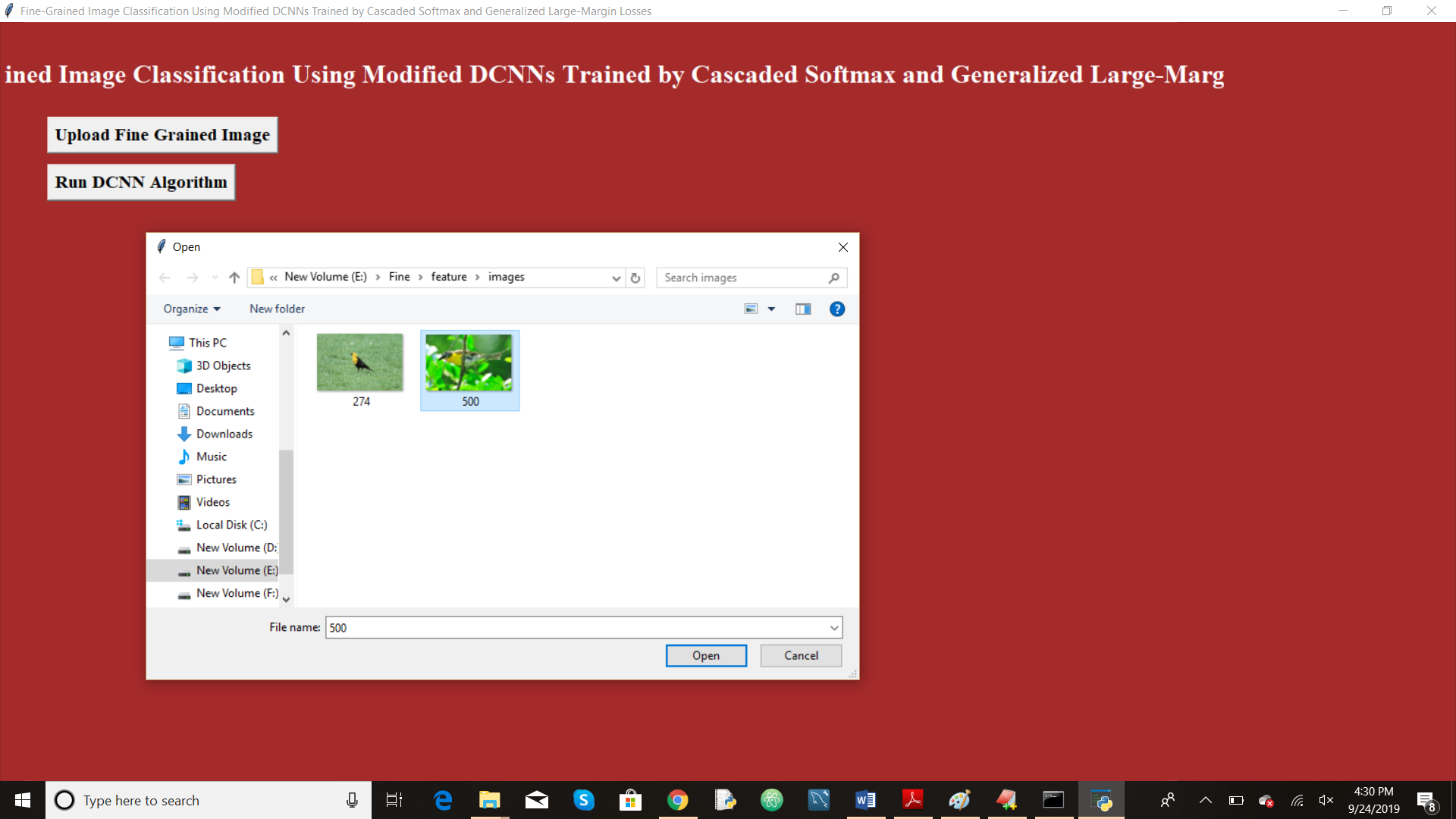
pip install sklearn

pip install scikit-image

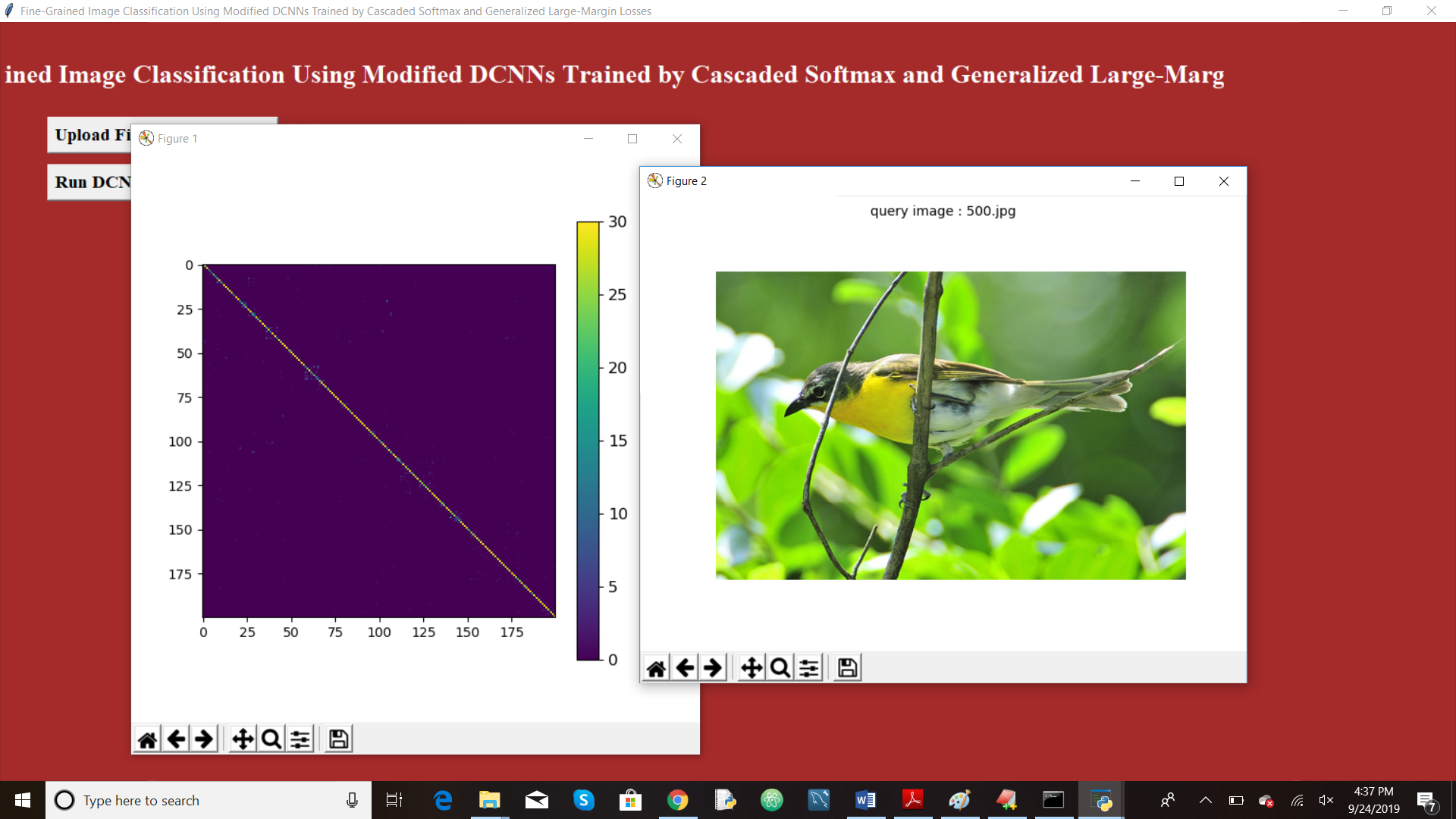
double click on run.bat file to get below screen



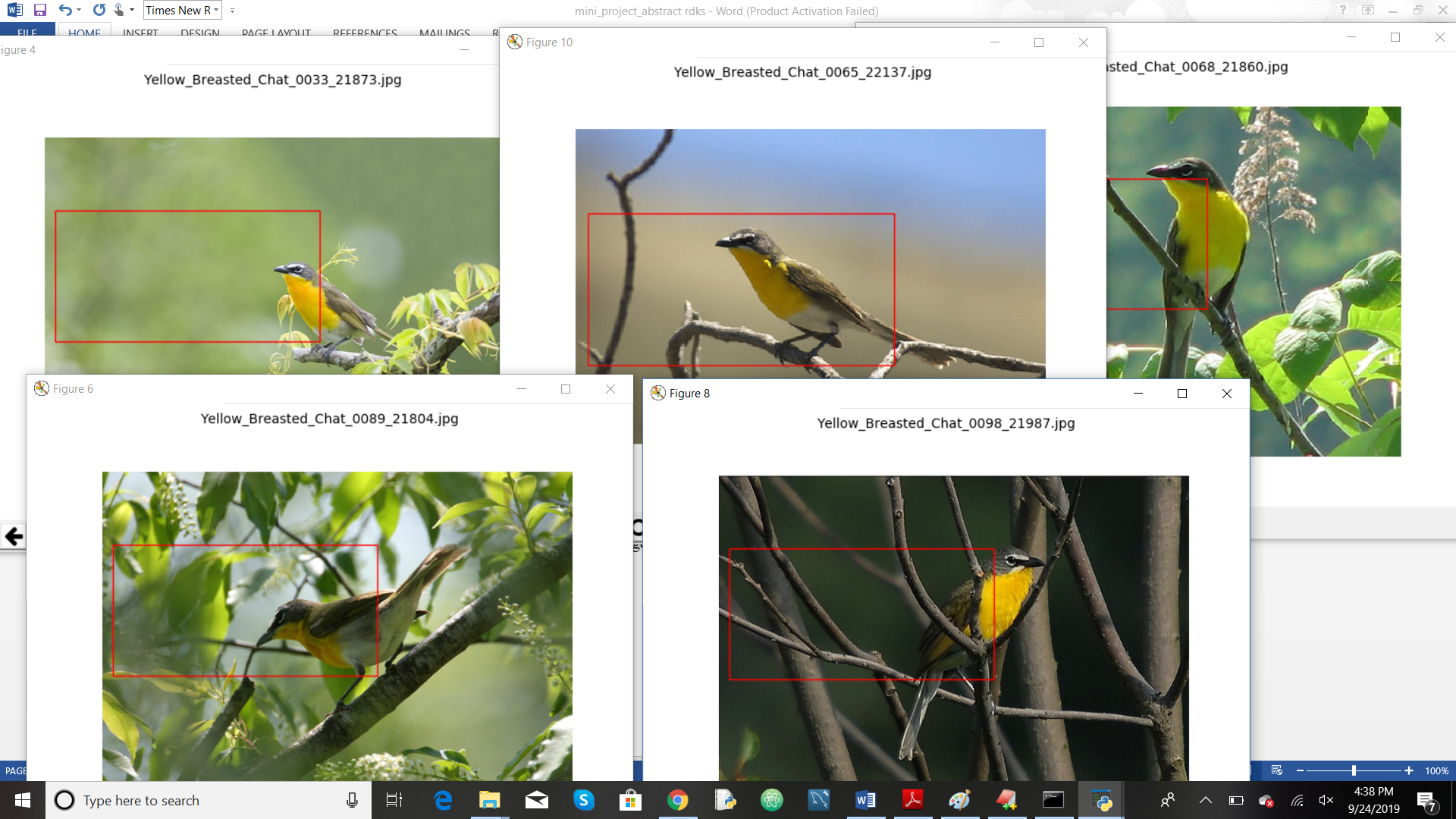
In above screen clicked on ‘Upload Fine Grained Images’ button to upload input image



In above screen I am selected one image and now click on open button to get below screen



In above screen we are seeing the uploaded query image. Now close all windows red colour window and query image window to get all similar images related to query



In above screen we can see we got five images as search result as all images are different but they got searched based on body part colour similarity. Their matched part are highlighted with red colour rectangle. With normal eyes they look different but by using this technique we can distinguish or identify them.