**NCTU 2019 CV HW5:**

**Structure from Motion (SfM)**

**Hsin-Yu Chen, Yuan-Syun Ye**

**May 2019**

**1 Introduction**

In this homework, we need to builds a classifier to categorize images into one of 15 scene types. We must complete the task by three way, respectively:

1. Tiny images representation + nearest neighbor classiﬁer
2. Bag of SIFT representation + nearest neighbor classiﬁer
3. Bag of SIFT representation + linear SVM classiﬁer

That is, we need to Implementation two kinds of images representation with knn classiﬁer and linear SVM classiﬁer.

Also, we tried CNN and ResNet with pre-train model to classify the images.

**2 Implementation**

**2.1 images representation**

**2.1.1 Tiny images representation**

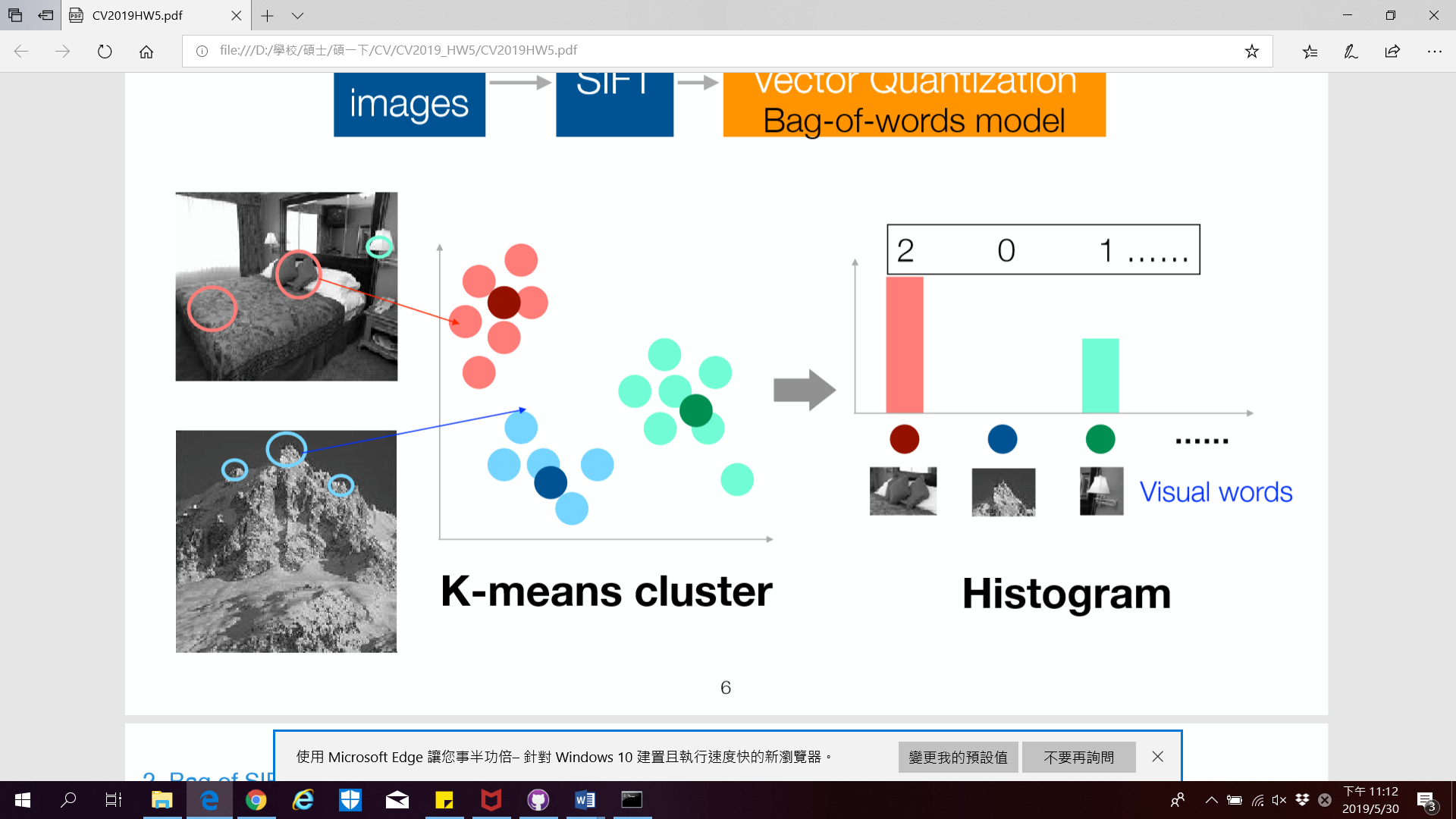
In Tiny images representation, all we need to do is resize the image to 16\*16.

Then because we need to compute the distance for nearest neighbor classiﬁer, we reshape the image into shape 1\*256

**2.1.1 Bag of SIFT representation**

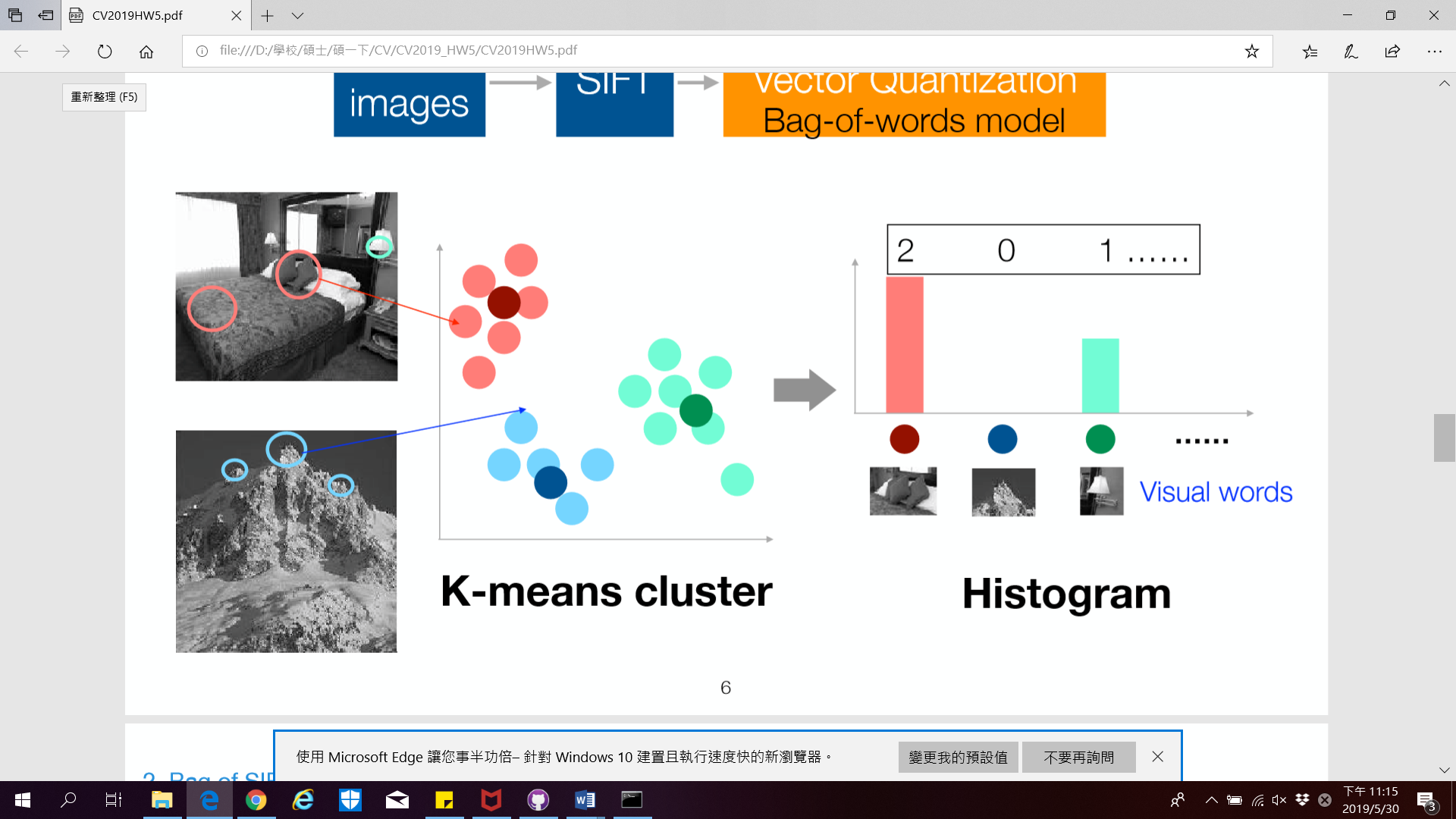
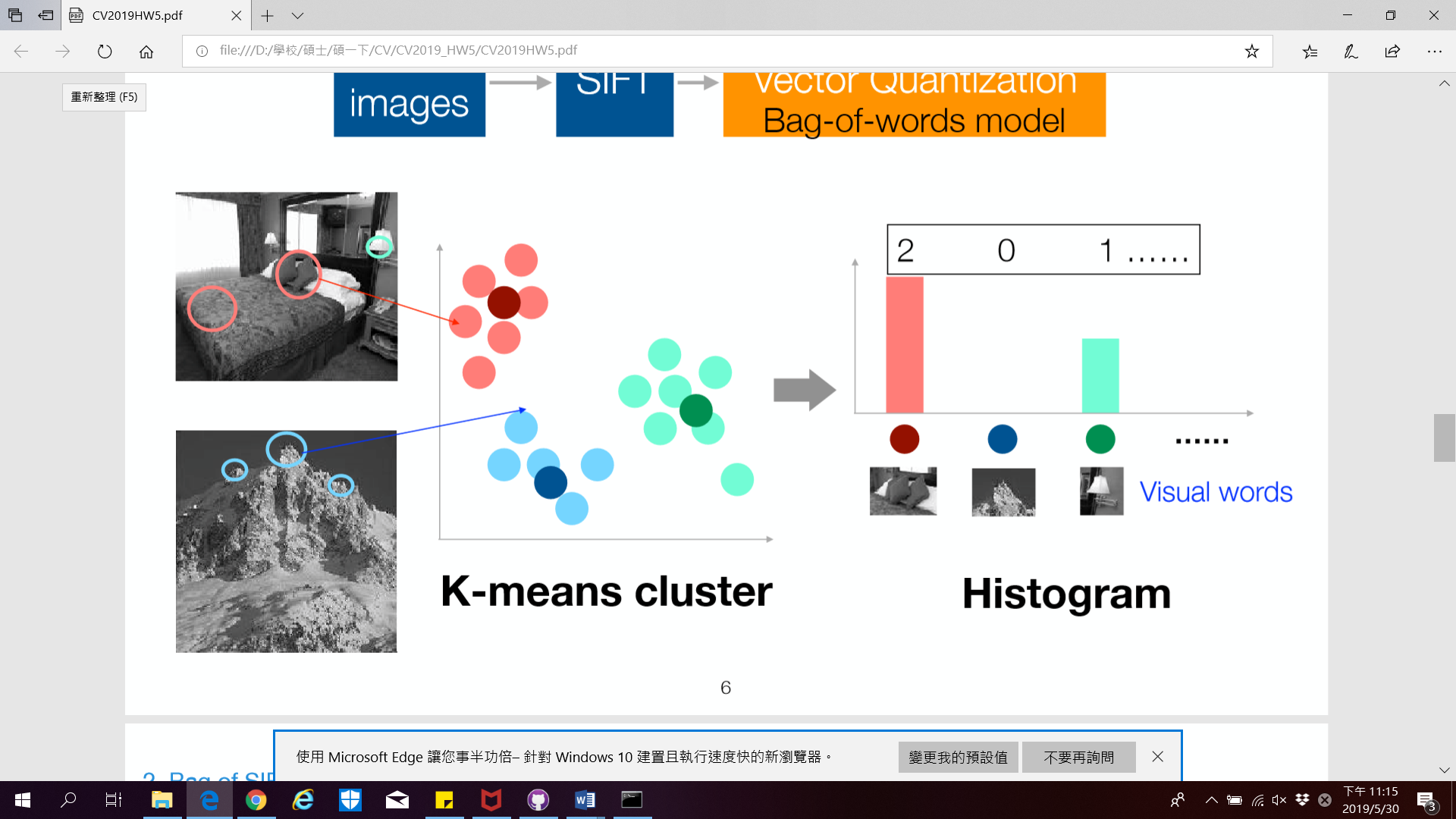
We use SIFT to find the descriptors for each image, then stack all the descriptors in a numpy array.

Then we do k-means clustering to find out k clustering center from all the descriptors, where we set k = 300.



🡺Find k clustering center

Next, we classify features of each image and calculate the histogram of them. Then the histogram is the Bag of SIFT representation of the image.

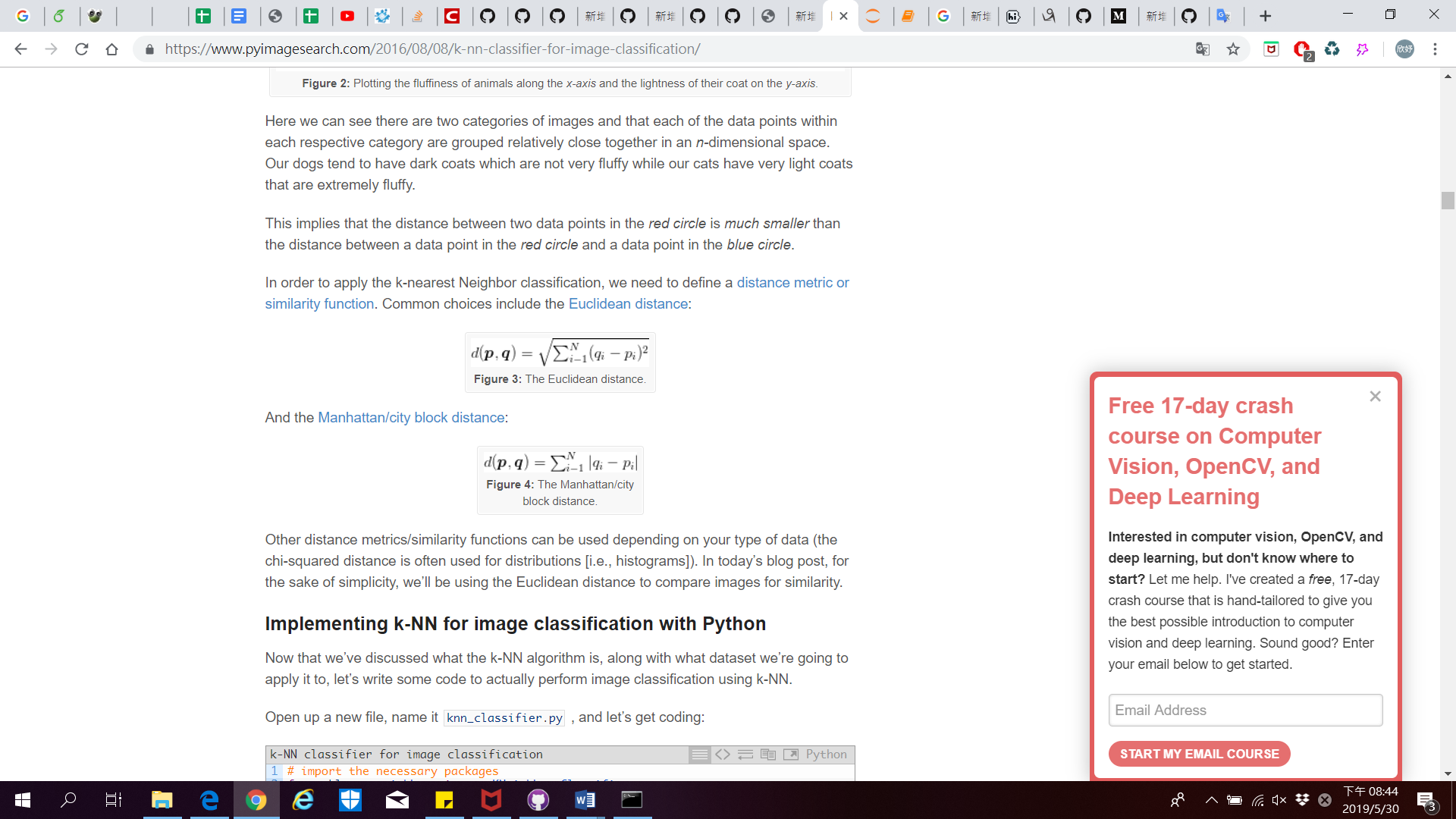
 

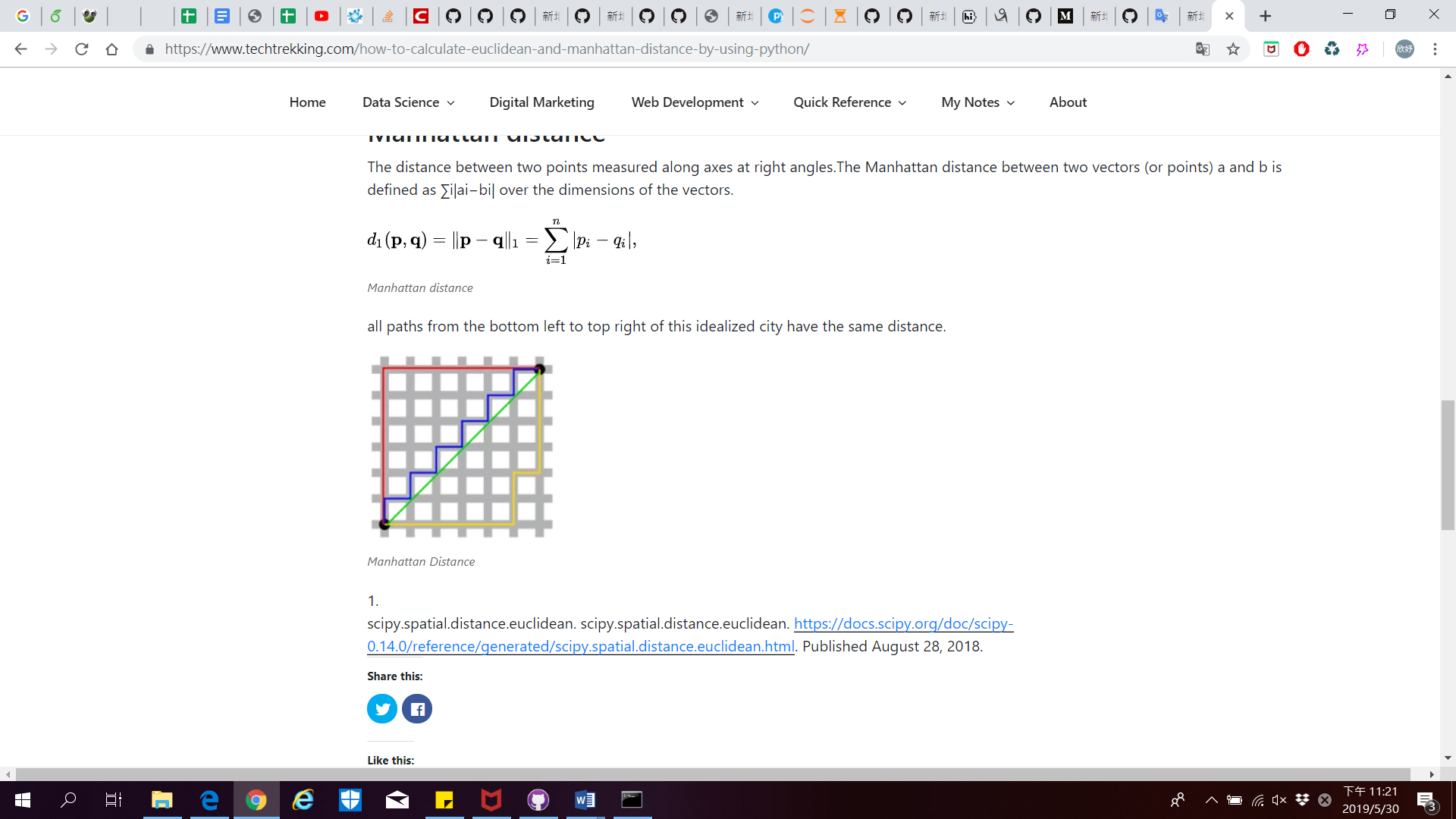
🡺 Calculate the histogram of image features

**2.2 Classiﬁer**

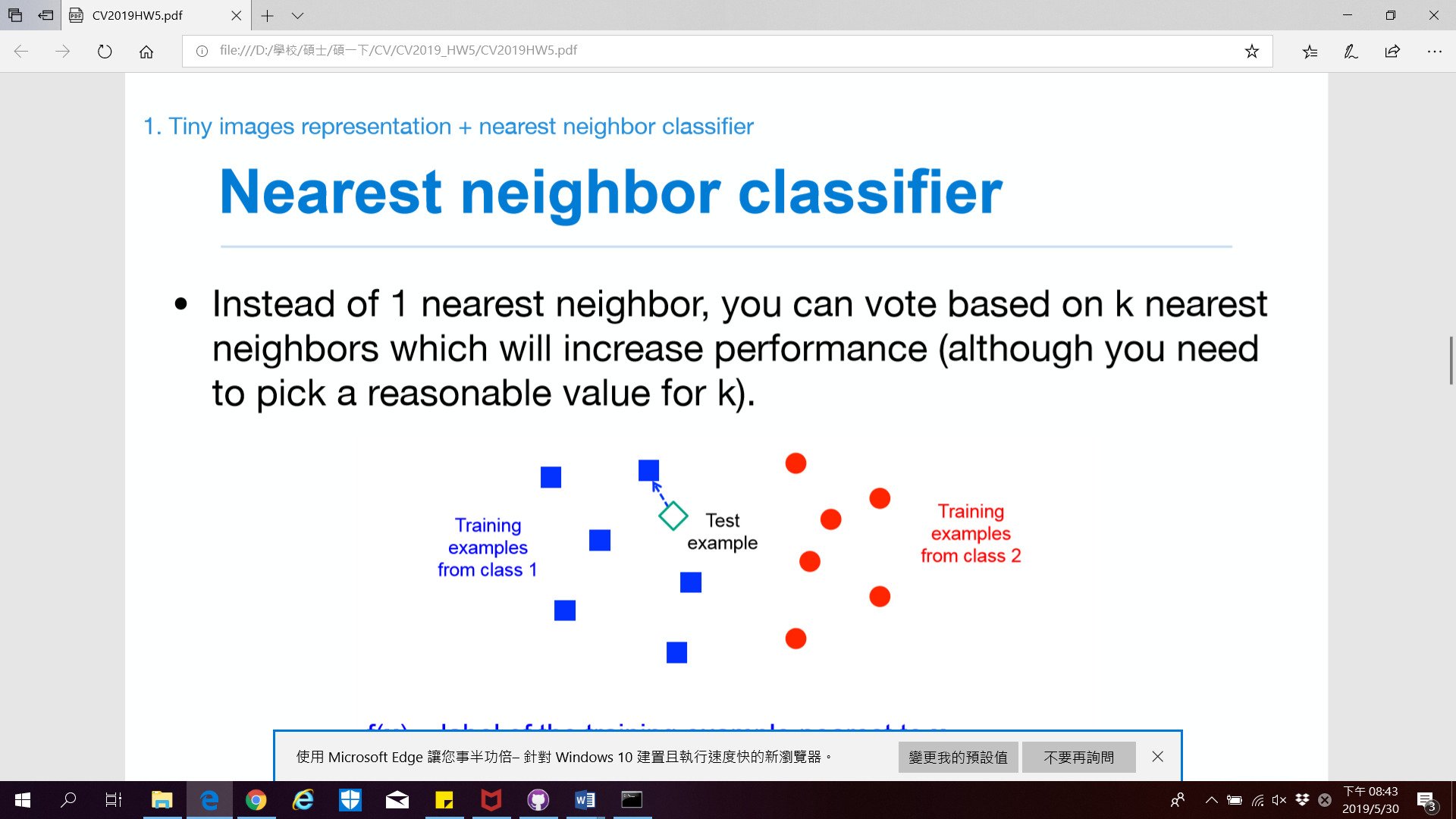
**2.2.1** **Nearest neighbor classiﬁer**

We Implementation both Euclidean distance and Manhattan distance to compute distance between the test image and each training data.

Euclidean distance 

Manhattan distance 

Then setting the label of test image as same as the training data which has shortest distance with the test image.



🡺 Set the test example label to class one

**2.2.2 Linear SVM classiﬁer**

**2.3 Deep Learning**

**2.3.1 CNN model**

We use 2 layer CNN model with Adam as optimizer, and Cross Entropy as loss function to train the model.

**2.2.2 ResNet model with pre-train**

Because the training dataset are too small and the accuracy doesn’t seem good, we constructs a pre-train ResNet-34 model. The model has four layers with Adam as optimizer, and Cross Entropy as loss function.

**3 Experimental**

**3.1 Tiny images representation + nearest neighbor classiﬁer**

Accuracy: %

**3.2 Bag of SIFT representation + nearest neighbor classiﬁer**

Accuracy: %

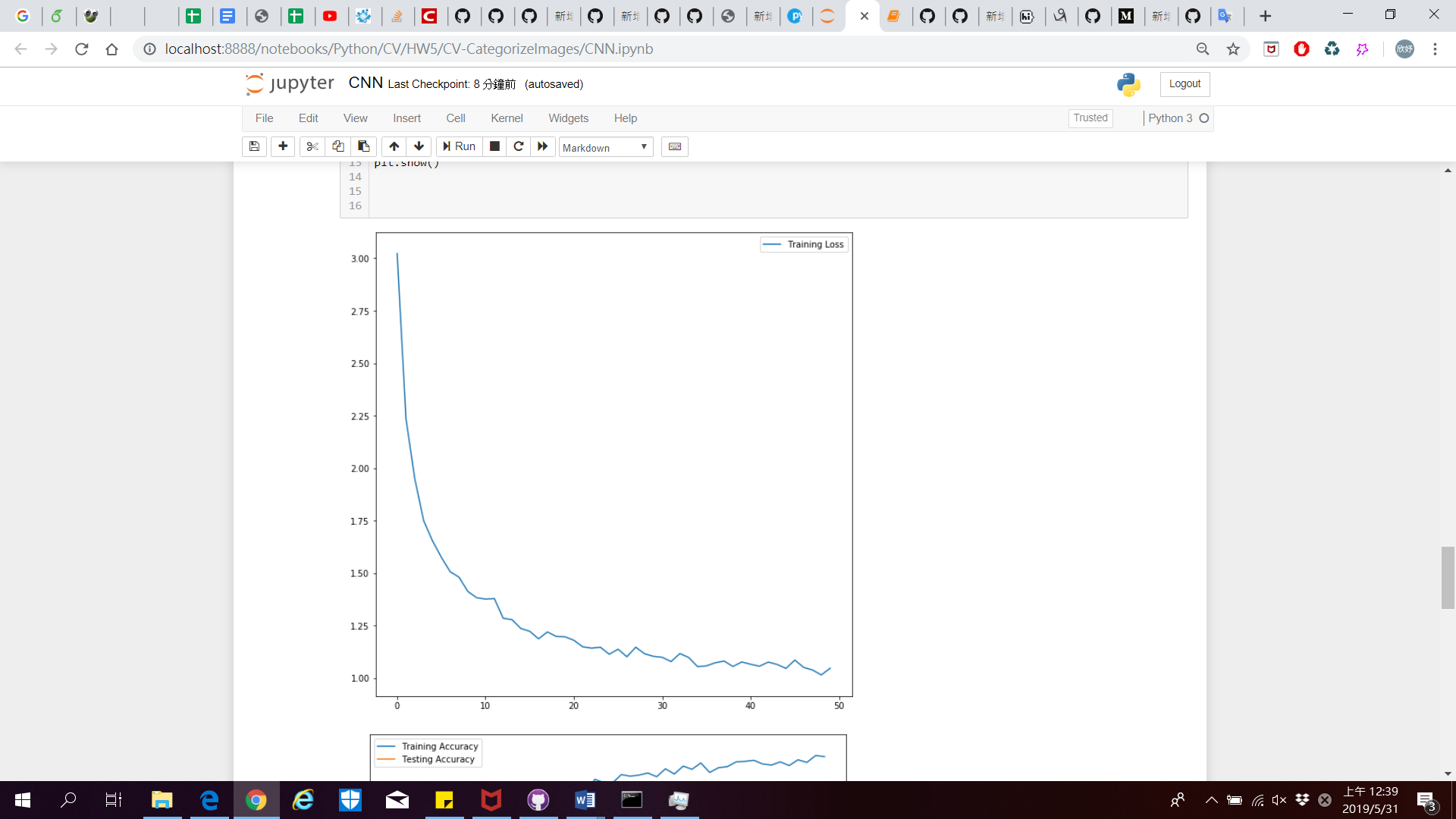
**3.3 Bag of SIFT representation + linear SVM classiﬁer**

Accuracy: %

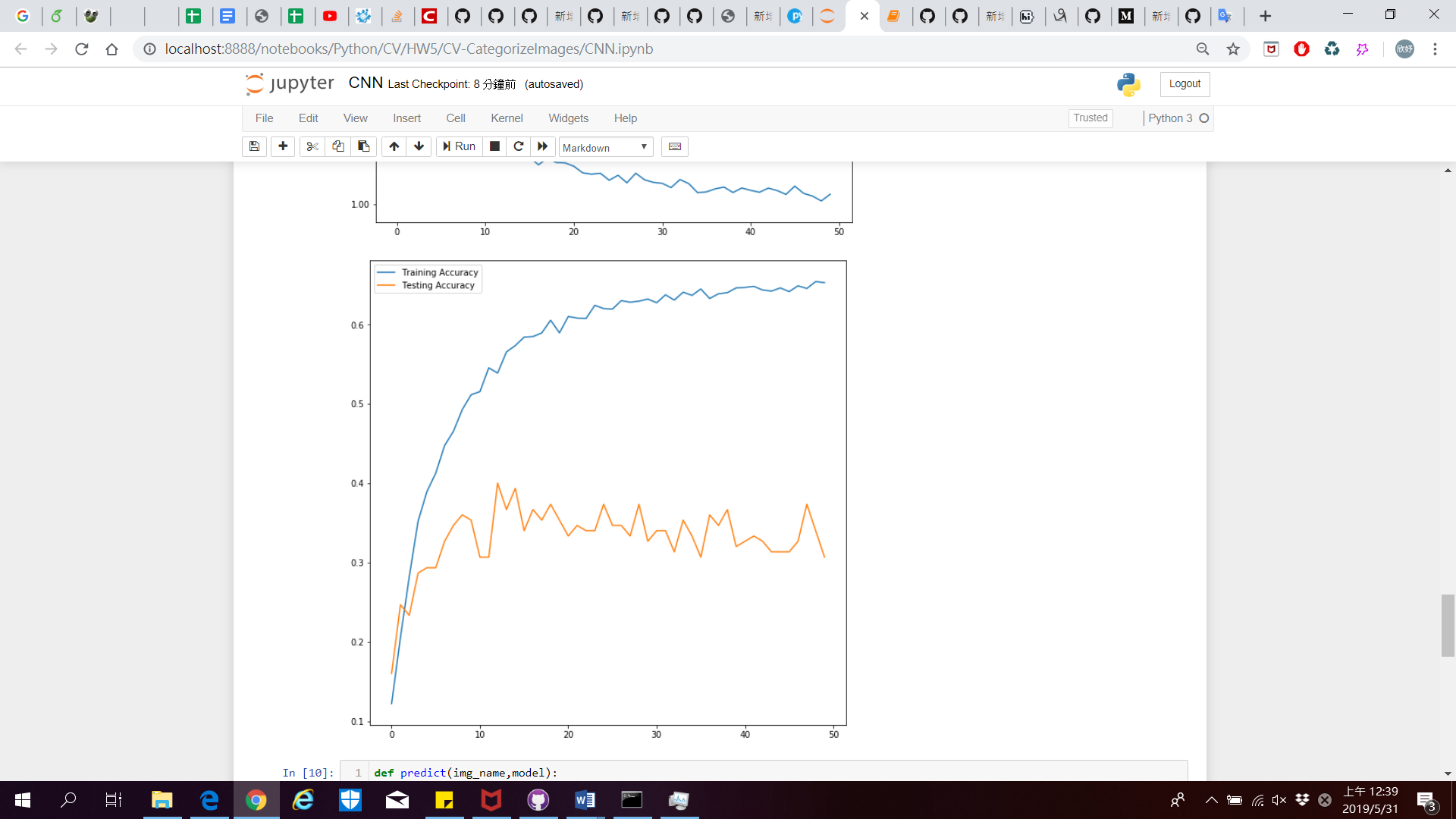
**3.4 Deep Learning by CNN model**

Accuracy: 30-35%

Because the training dataset are too small and the accuracy doesn’t seem good



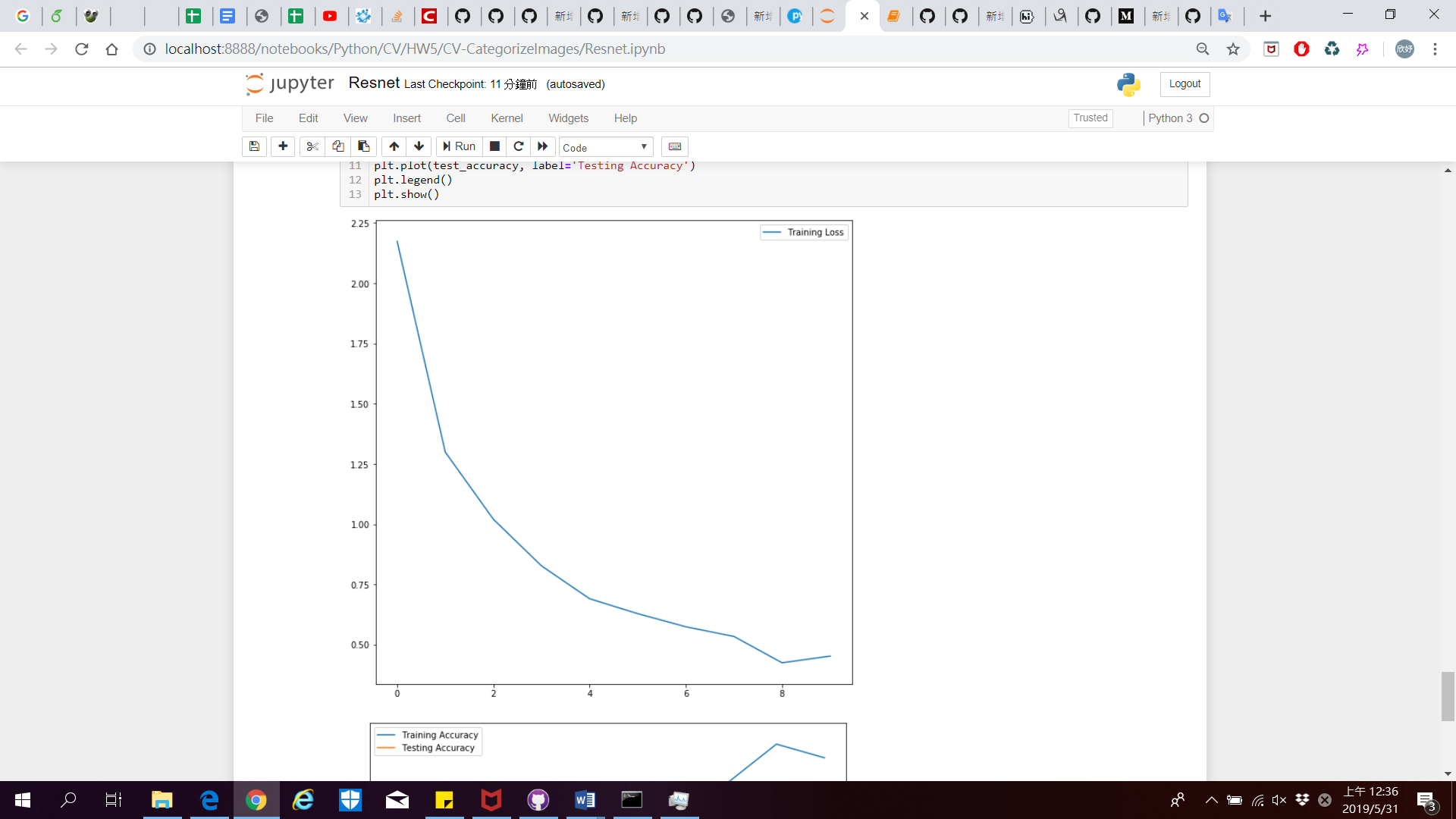
🡺 Training Loss



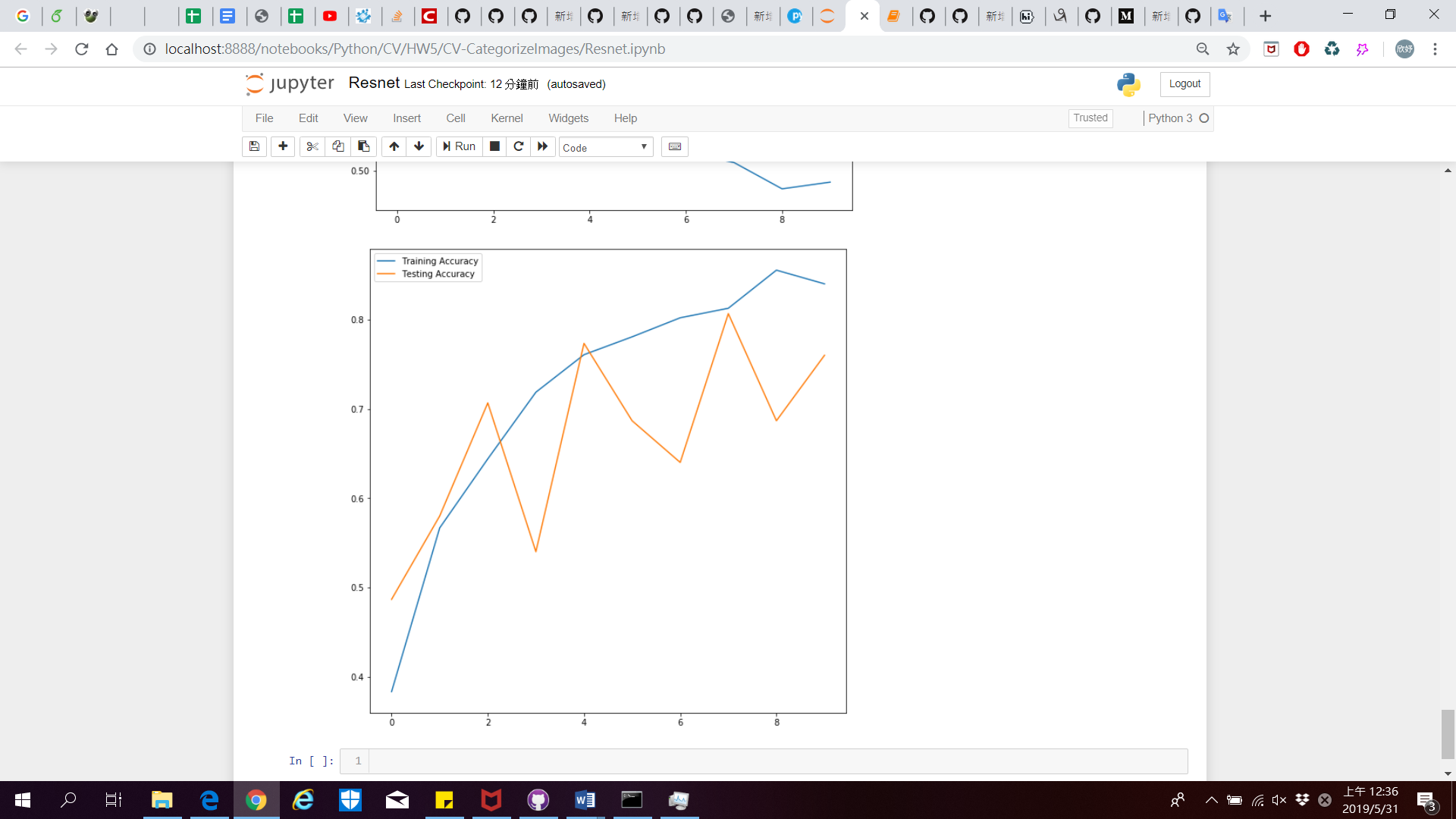
🡺 Training Accuracy & Test Accuracy

**3.5 Deep Learning by pre-train ResNet model**

Accuracy: 75-80%



🡺 Training Loss



🡺 Training Accuracy & Test Accuracy

**4 Discussion**

**5 Conclusion**

**6 Work Assignment Plan**

This homework divided into two parts. Yuan-Syun Ye is responsible for the part of cording by MATLAB and checks this report. Hsin-Yu Chen is responsible for the part of cording by python and the writing of the report.

**References**

https://www.pyimagesearch.com/2016/08/08/k-nn-classifier-for-image-classification/