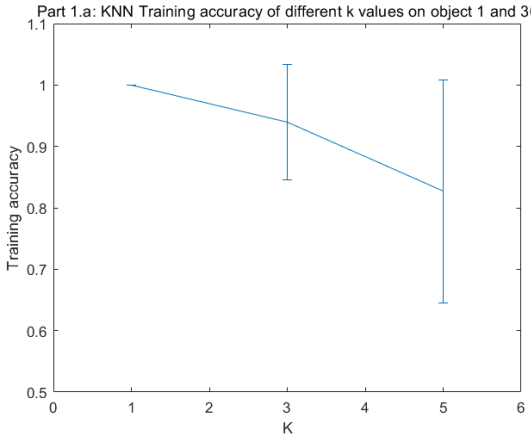
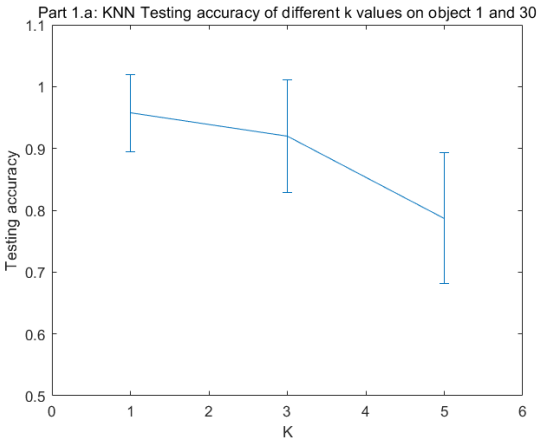
Report: Exercise 2: Face Recognition-COMP 24111

Name: Kai Chen **Student ID:** 10411044 **Date:** Oct. 20th, 2018

1. Part 1.a



(**Figure1:** KNN training accuracy of binary classifier)



(**Figure2**: KNN testing accuracy of binary classifier)

A1: The training accuracy when k=1 is **always 100%**. When k=1, we only decide an input's category according to the nearest point. When calculating training accuracy, every input comes from training set. **The nearest data point is always itself**, as they are exactly the same. But testing accuracy **differs** because **input isn't in training set now**, and the nearest point depends on the image data itself. **The nearest one may come from another object**.

A2: No, behavior differs between datasets. I calculate average accuracy of datasets. Most accuracy are higher than 85%, but the training and testing accuracy of dataset **No. 10, 22, 28, 29, 46** are both lower than 85%. Behavior differs because we **randomly separate** the dataset, so training sets of each datasets are different. **Different training sets** may result in different training and testing results and different accuracy.

A3: No, it's not good to set k as even numbers. In this case, we **only have two categories**. If we set k as an even number, it's possible finally **half data points** in the training set vote for object 1 and the **other half** vote for 30, which will **bring confuse about which to choose**. Accuracies do differ over different values of k. In this case, we only choose **3 candidate for k (1, 3, 5)**, and when k gets bigger, the **average accuracies decrease and the standard deviation increase** both in training and testing processes **(See in Figure 1 and 2)**.

A4: For my KNN, **object 1** is more difficult to classify. (1) I write a function **'object_acc'** to calculate the **prediction precise for each object**. Finally testing precise of **object 1 is 77.62%, while 30 is 100%.** (2) During testing process, I run **'ShowResult'** to show the prediction result. Many images of object 1 are **surrounded by a white box (i.e. wrong prediction)**, while those of 30 remain the same. (3) I think it's because images of object 1 are more complex.

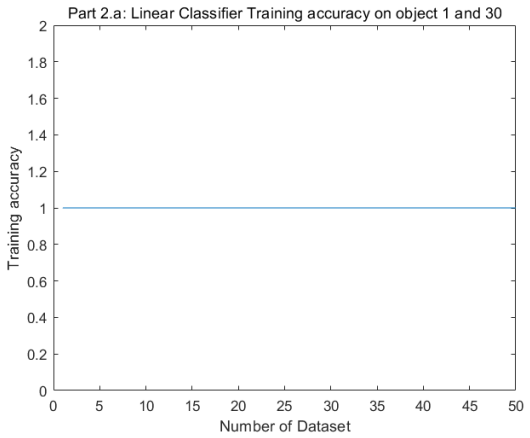
2. Part 1.b

A1: No, it's not good to select randomly. If so, images of some objects will be very likely to be more than others. For example, assume a training set where object A gets more images than B. For A, images of A have relevance with each other. They will form a cluster, and a new data point will **more likely to recognized as A**. The model may **over-fit for A**. For B, less images exist in the training set, so to some extent we get less data and information about B, and we may get **under-fit for B**.

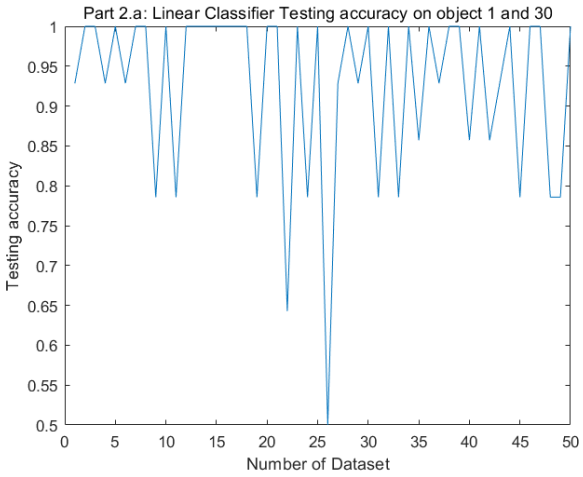
As a result, the model will be likely to **over-fit the training set** and **not generalize well**.

A2: I use **'object_acc'** and find the 10 most difficult object for my KNN is **1, 4, 3, 13, 16, 10, 40, 23, 26, 8**. They all get a precise under 80%.

3. Part 2.a



(**Figure3:** LR training accuracy of binary classifier)

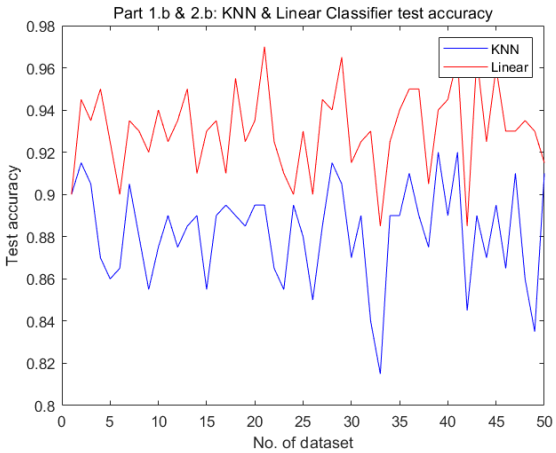


(**Figure4**: LR testing accuracy of binary classifier)

A1: My linear classifier training accuracies are all 100% in 50 datasets. (1) I think because linear classifier **has a training process**, which can find the best parameters to fit the data as well as possible. (2) All the data points will affect the **position of the decision boundary**, **so training data is made full use of**. Linear classifier **isn't so sensitive** to outliers as KNN.

A2: For my linear classifier, object 1 is harder for the **same reason as Part 1.a Q4**. My **'object_acc'** function shows **object 1 gets 85.14% test accuracy while object 30 is 100%**. Many images of object 1 are surrounded by white edges while object 30 don't.

4. Part 2.b



(**Figure5:** KNN & LR test accuracy of multi-class classifier)

Result: Figure 5 shows the test accuracy of KNN and Linear Classifier. We can see both algorithms differ a lot, but Linear Classifier outperforms KNN nearly in all the 50 datasets.

A1: From **Figure 5**, we see **linear classifier gets a higher accuracy** than KNN in nearly all of the 50 datasets, **with average 93.05% compared to 88.21%.** Now we only care **testing process**. Suppose we have **N training examples with dimension d and n testing points**. KNN gets a **time complexity O(n * N * d)** and a **space complexity** of **O(N * d)**, while linear classifier gets a **time complexity O(n * d + d)** and a **space complexity** of **O(1).**

A2: I use **'object_acc'** to find the 10 most difficult objects for my KNN is **13, 3, 1, 8, 7, 16, 36, 35, 20, 22**. They all get a precise under 90%. We can see they both classifiers struggle in **1, 13, 8, 3, 16**, but they still have their own weaknesses in the meantime.