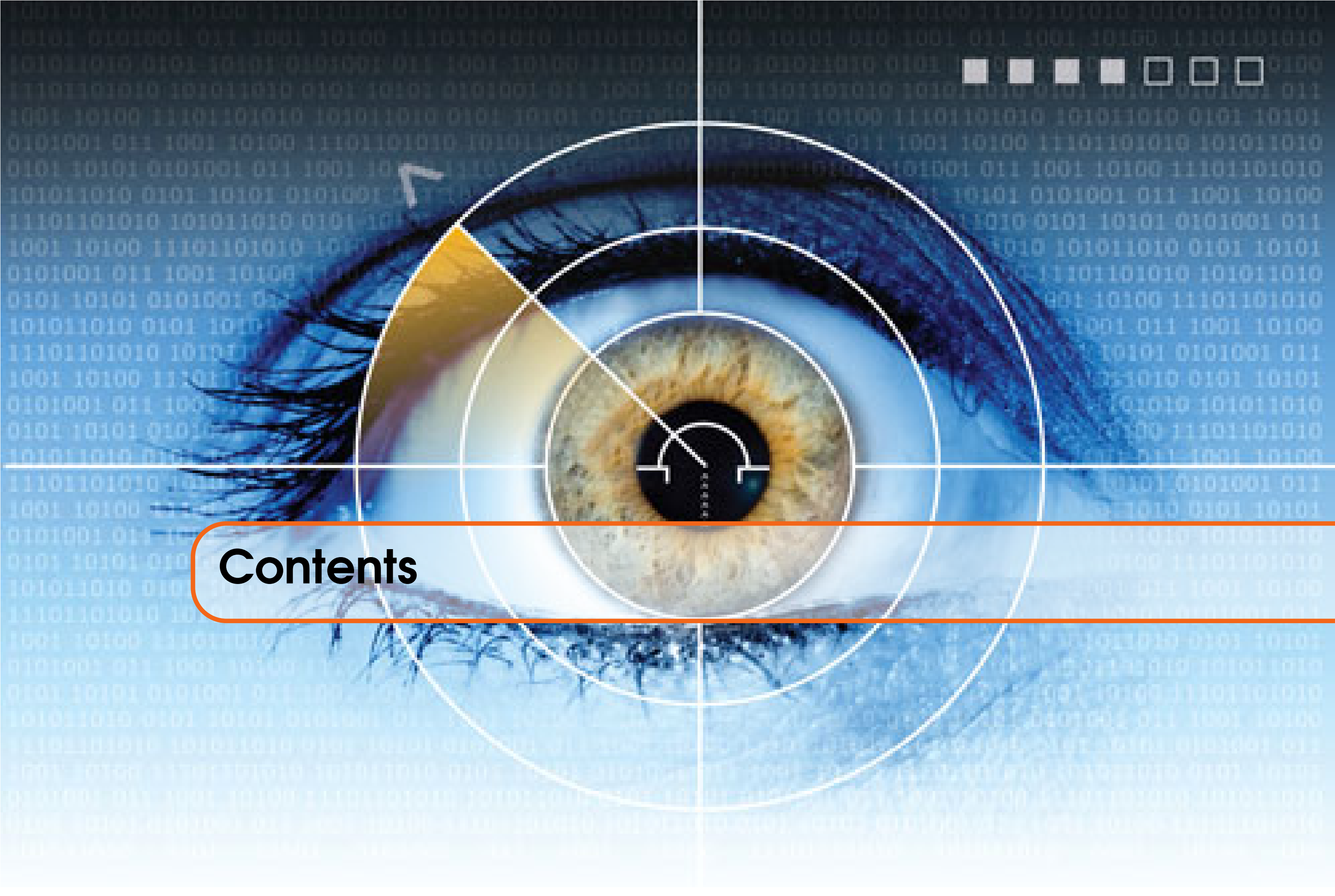
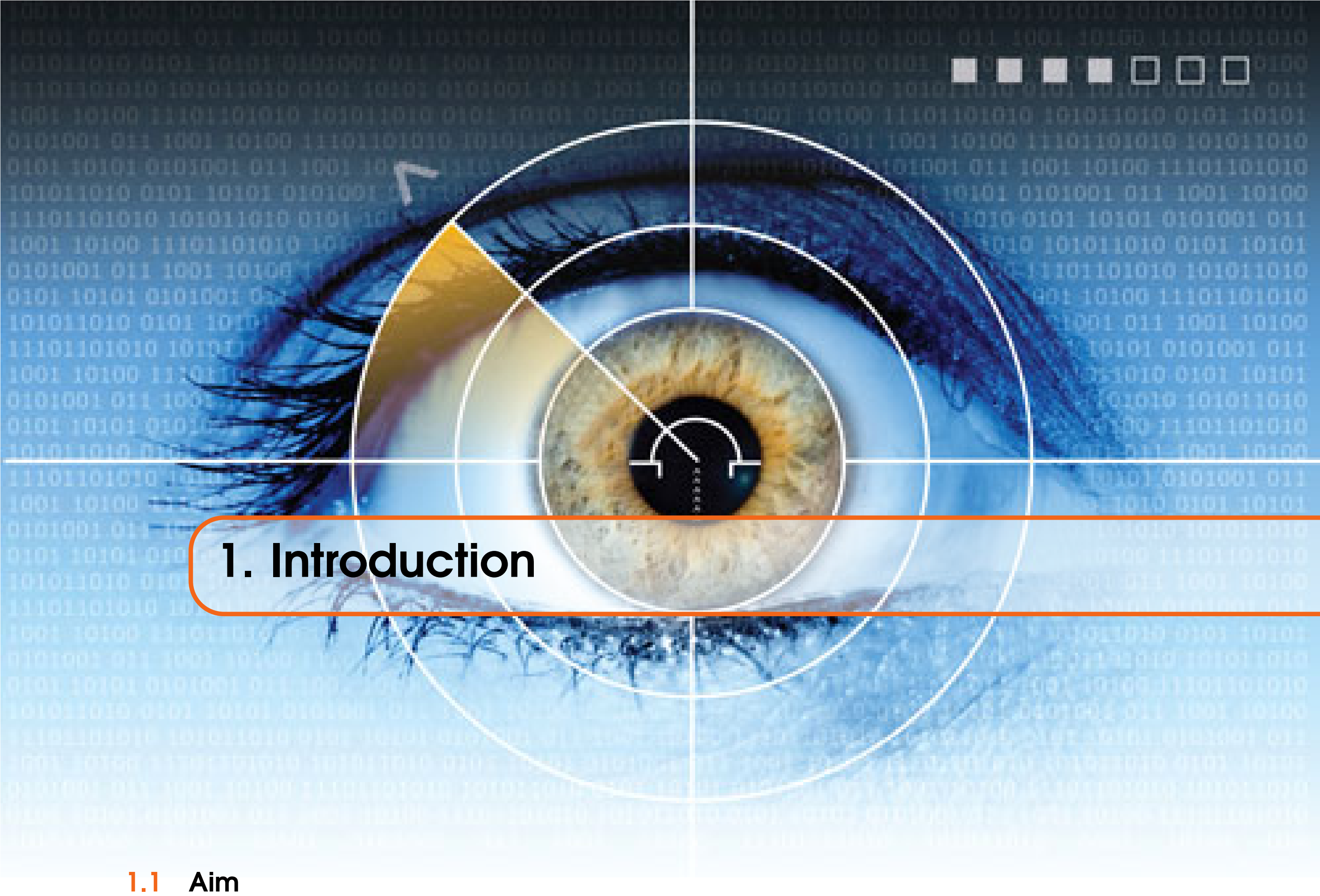
Image Similarity Matching and Classification

INFO3406: Assignment 1

Ian Cleasby SID: 430595639, Costas Korai SID: 440241607



Contents



1. Introduction

1.1 Aim

The aim of the assignment is to implement a large-scale image similarity matching system and determine classification accuracy on a common data analytics benchmark known as CIFAR 10 & CIFAR 100. Using a classifier of our choice, we will conduct image similarity matching on CIFAR 10 & 100 images. With these images we will need to produce our own classifier to classify the images and assign the appropriate labels. The report will discuss the approach our team took on classification of the images provided by the CIFAR 10 & 100 website.

1.2 The Problem

In this section we will introduce the Image Classification problem, which is the task of assigning an input image one label from a fixed set of categories. This is one of the core problems in Data Analytics that, despite its simplicity, has a large variety of practical applications. Throughout the assignment, we will take a look into the challenges that make image classification more complicated.

These challenges that make image classification complicated are:

Viewpoint variation. A single instance of an object can be oriented in many ways with respect to the camera.

Scale variation. Visual classes often exhibit variation in their size (size in the real world, not only in terms of their extent in the image).

Deformation. Many objects of interest are not rigid bodies and can be deformed in extreme ways.

Occlusion. The objects of interest can be occluded. Sometimes only a small portion of an object (as little as few pixels) could be visible.

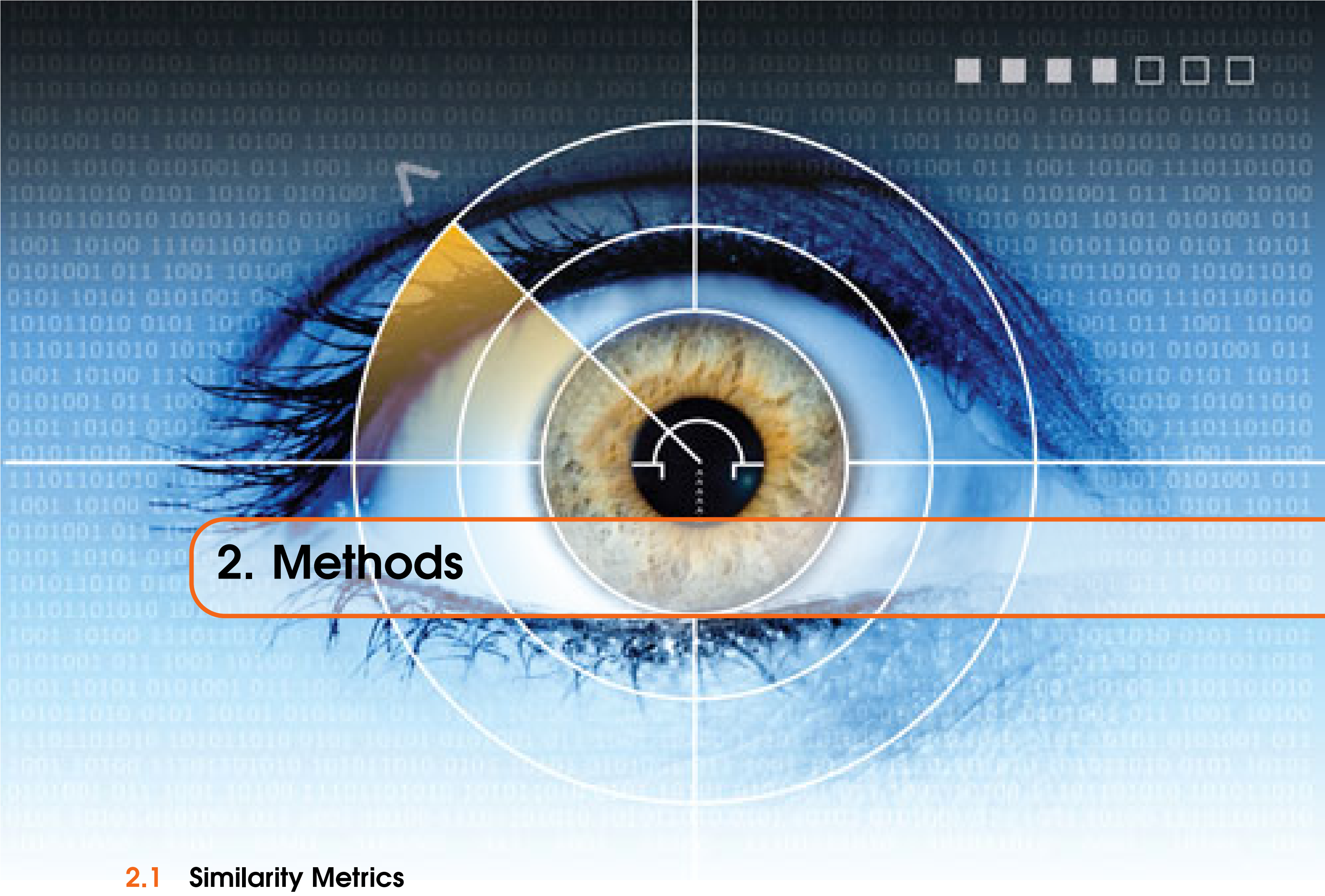
Illumination conditions. The effects of illumination are drastic on the pixel level.

Background clutter. The objects of interest may blend into their environment, making them hard to identify.

Intra-class variation. The classes of interest can often be relatively broad, such as chair. There are many different types of these objects, each with their own appearance.

1.3 Why is this study important

As one of the core problems in Data Analytics the significance of this study is easily apprehended when one looks at its numerous possible applications in real life, from security video analysis to entertainment its range of possible uses is continuously growing. As an area of study in university and in particular when studying data analytics this is extremely relevant, as the implementation of the classifier we are aiming to complete will require us to apply all we have learned so far in this unit while giving us an idea of the effort and resources needed in order for such systems to be made and run.



2. Methods

2.1 Similarity Metrics

During this assignment we looked at multiple similarity metrics and decided to concentrate on implementing the K-Nearest-Neighbour due to its simplicity and non-parametric properties which allow it to classify without making any assumptions about the distribution of the data.

2.1.1 Distance Calculating Equations

The two distance calculations we considered using are the Taxicab and Euclidean these methods will allow us to calculate the distances between the pixels within our images for our data sets. The taxicab distance is calculated as the sum of the absolute differences of the two points being compared. The equation is:

$$d_1(p,q) = \|p-q\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$

The euclidean distance is calculated as the square root of the sum of the distance between point p and q squared. The equation is:

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

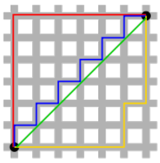


Figure 2.1: Taxicab geometry versus Euclidean distance

2.1.2 K-Nearest-Neighbour

In this unit we were introduced to the Nearest-Neighbour Classifier a method of inferring the class of an object based its distance from the remaining objects within its data set. The basic idea of this method is that similar objects have similar properties in space, in our case similar images would have similar pixel values within their given range. Looking closer at this method there are three requirements needed before it can be implemented the set of stored records, a distance metric to compute the distance between records and the number of nearest neighbours to retrieve which will have a different optimal value between varying implementations and data sets.
The set of stored records we will use are CIFAR-10 & CIFAR-100, in order to calculate distances between points in our data we will use either the Euclidean or Taxicab distance calculation methods and our value for K will be decided based on results we get from tests.

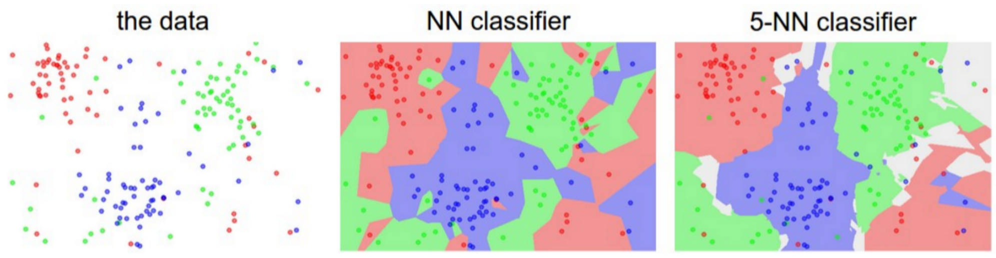


Figure 2.2: [1]KNN classification Visual

2.1.3 K-Nearest-Neighbour Robustness

For this metric we decided to test varying values of K and differing methods of distance calculation within the method itself. In order to find the most appropriate K we first implemented and test the method with K set to 1 and while using various distance calculation methods. The different methods used were the Taxicab and Euclidean, of the two we found that the Taxicab produced more accurate results and so decided to use it when testing further candidates of K. After this selection we proceeded to test KNN with the values 1,3,5,7,10 and 15, of these 1 and 10 appeared to return the highest accuracy we decided to use K equals 10 as even though it returned slightly less accurate results than K set to 1 we came to the conclusion that using k set to 10 would give us more reliable results. The reasoning behind this is simply that when using KNN the larger the value of K the smaller the effect of noise on each classification, this does come at the cost of making it harder to distinguish classes leading to the lower accuracy.
To keep our code robust the ratio of speed and accuracy needed to be considered, under other circumstances speed would have been of greater importance but due to the marking criteria of this assignment, where 5 of our 20 marks are determined by the accuracy of our classifier speed was left aside in order to prioritize increasing accuracy.

2.1.4 Tested Hyperparrameter Results

Below we have included tables showing our varied hyperparrameters and their differing affects on our results.

All Image Batches CIFAR-10 Euclidean :

K	1	3	5	7	10	11	15
Accuracy	28.7%	27.8%	28.9%	28.5%	29.4%	29.4%	29.1%

Single Image Batches CIFAR-10 Taxicab :

K	1	3	5	7	10	11	15
Accuracy	31%	30.5%	31.3%	31.9%	32.4%	32.45%	32.87%

All Image Batches CIFAR-10 Taxicab :

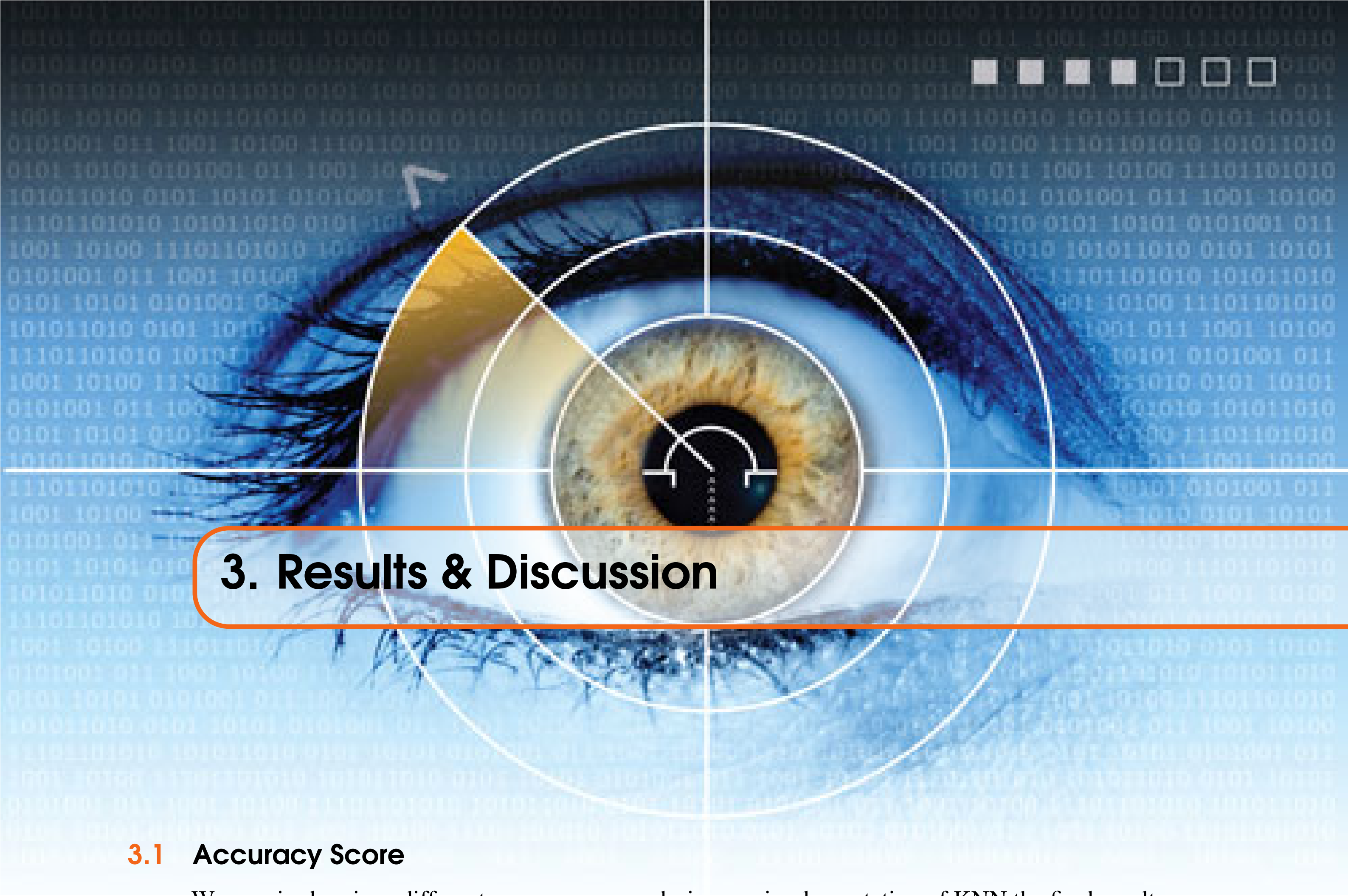
K	1	3	5	7	10	15
First Attempt	31%	30%	31.3%		32.4%	
Improved Attempt	38.5%	36.2%	37.7%	37.6%	38.1%	37.3%

Single Image Batches CIFAR-100 Taxicab Superclasses :

K	1	3	10
Fine Accuracy	19.8%	15.3%	17.2%
Coarse Accuracy	29.0%	24.5%	27.3%

2.1.5 Accuracy Variations

From our above hyperparameter testing we noticed that there was a two stage drop in the accuracy of our KNN classifier one superclass classification had a drop of approximately 10% while subclasses were now only being predicted correctly around 17% of the time. After discussion we came to realize this was most likely due to a decrease in the number of training elements per class with the increase in classes. In both Cifar-10 and Cifar-100 we had 50000 training images, of these images in Cifar-10 each class could effectively use 5000(50000 divided into 10 classes) images to train in order to predict others of the same class. While in Cifar-100 our superclasses would have 2500 and our subclasses would have only 500 images to use.
At this point it has become evident that the size of our training data has a direct affect on how well we are able to classify images.



3. Results & Discussion

3.1 Accuracy Score

We acquired various different accuracy scores during our implementation of KNN the final result we decided on was insert Score when K is set to 10. We considered 10 to be a more reliable and stable result, as it would take the classes of 10 neighbors nearest to the test case.

3.2 Confusion Matrix

Based on the predictions we obtained from our KNN classifier when run on CIFAR-10 we have constructed the following confusion matrices where the number 0-9 represent the class labels plane, car, bird, cat, deer, dog, frog, horse, ship, truck respectively.The numbers 0-19 represent the 20 super classes of CIFAR-100.

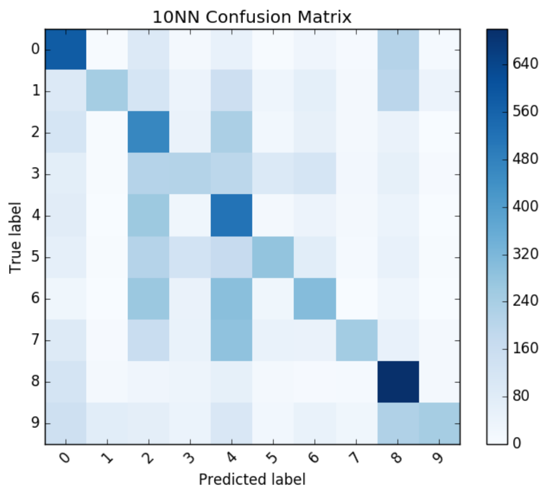


Figure 3.1: Confusion Matrix CIFAR10 K=10

Images represent the distribution of our classifications where our predicted labels and true labels correlate. Notably the darker diagonal pattern indicates where we correctly classified. For CIFAR 100, KNN = 1, classified our data more efficiently then KNN = 10, this could be due to the overfitting.

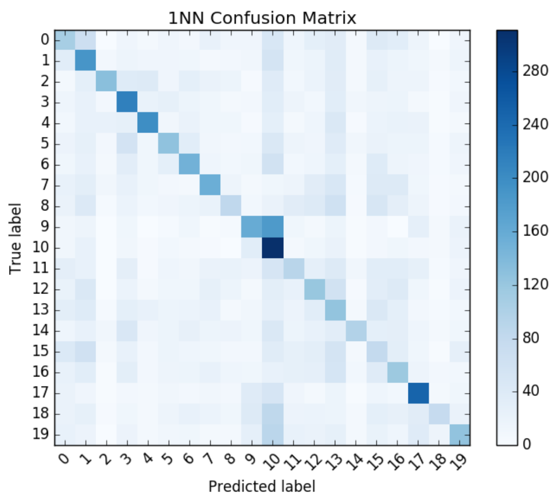


Figure 3.2: Confusion Matrix Superclass Cifar-100 K=1

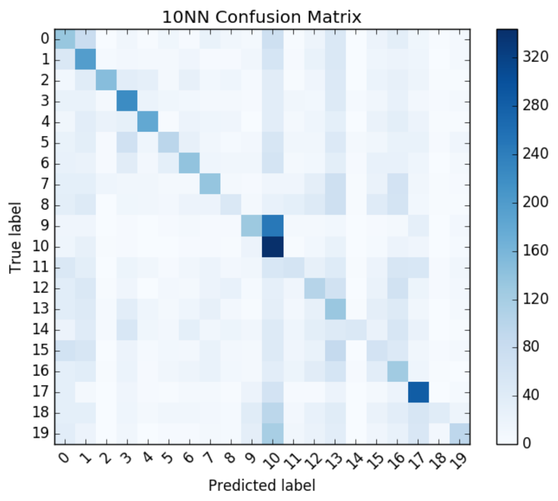


Figure 3.3: Confusion Matrix Superclass Cifar-100 K=10

As is evident by our results table and by the visuals in 3.2 and 3.3 the accuracy of KNN on CIFAR-100 when K is set to 1 and 10 is very similar. In both confusion matrices we see that the superclass with the most correctly predicted labels is class 10, large man-made outdoor things, this is represented by the darkest block within our confusion matrix plot. A possible explanation for the class' results is the presence of deformation within other classes and its absence within large man made structures, which in general will have similar shapes. Other classes such as large carnivores may have images with largely varying shapes, from lions to bears.

3.3 Reasons For Misclassification

3.3.1 Predictions

Below is a table showing the number of times the class on the left has been predicted correctly(in bold),incorrectly and by what class (on the top) it was incorrectly predicted by. The rows are represented by our predicted label and true labels represent the column and these numbers were generated from CIFAR10 where K = 10.

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Total
Plane	**581**	4	94	11	52	3	28	6	212	9	1000
Car	93	**248**	120	40	150	32	65	13	198	41	1000
Bird	119	3	**465**	44	234	21	55	11	46	2	1000
Cat	69	5	211	**216**	195	98	119	19	60	8	1000
Deer	79	1	264	25	**518**	16	41	13	43	0	1000
Dog	62	4	211	130	172	**280**	74	10	52	5	1000
Frog	26	2	267	46	296	25	**305**	1	31	1	1000
Horse	88	7	164	47	287	47	45	**251**	51	13	1000
Ship	122	12	30	33	58	16	8	9	**700**	12	1000
Truck	145	75	68	42	104	21	48	27	224	**246**	1000

In the graph below, we show a sample of some of the results being misclassified from the table above. Notably misclassifications occur notably on:

- Cats and planes are often confused with ships.
- birds and deer are often classified as one or the other. This might be occurring because of view and scale variation as well as including background clutter.
- Dogs, frogs and Horses get mostly classified as birds or deer for the same reasons above.
- Shops are the most accurate however they get classified as planes. This might be occurring because of background clutter and Illumination conditions. The sea might be confused with planes flying in the air. As shown in the sample.
- Trucks are mostly misclassified as planes or ships. Light backgrounds confuse the classifier which interprets it as a white truck.

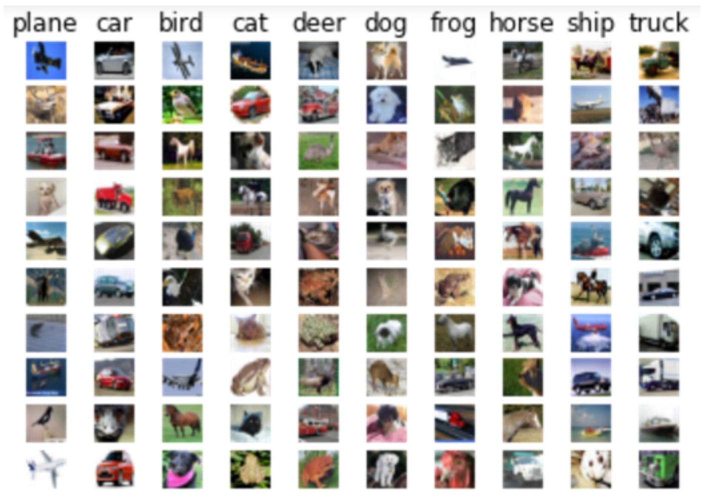


Figure 3.4: Sample Output CIFAR10 K=10

3.3.2 Misclassifications Explained

During our testing of KNN with the Cifar-10 & 100 data set we found that numerous misclassifications were occuring and while our hyperparameters can clearly be seen to affect the results that is not all there is to this. Firstly as we are doing image classification on similar but non-identical images, we can at no point expect to get a 100% accurate classification, the best one could hope for is 99% and achieving that would be very difficult and require much more time and resources than available for this study.

Another reason behind the misclassifications has to do with the method of classification itself, as we are essentially comparing the pixels of images in order to classify them similar images colour wise can be mismatched. If we think about any one image we can seperate it in to two sections the focus and the problem background clutter, in our data set the focus of each image is what we would want that image to be classified as. The problem with this is that images may be classified by their backgrounds rather than the focus of the image when the values of the pixels in the background overshadow the focus percentage wise.

Background clutter and illumination conditions made water and the sky look as light similar in comparisons, this was notably seen when comparing trucks with planes and ships. Where for animals, Background clutter, Deformation, viewpoint and scale variation make animals more difficult to classify correctly.

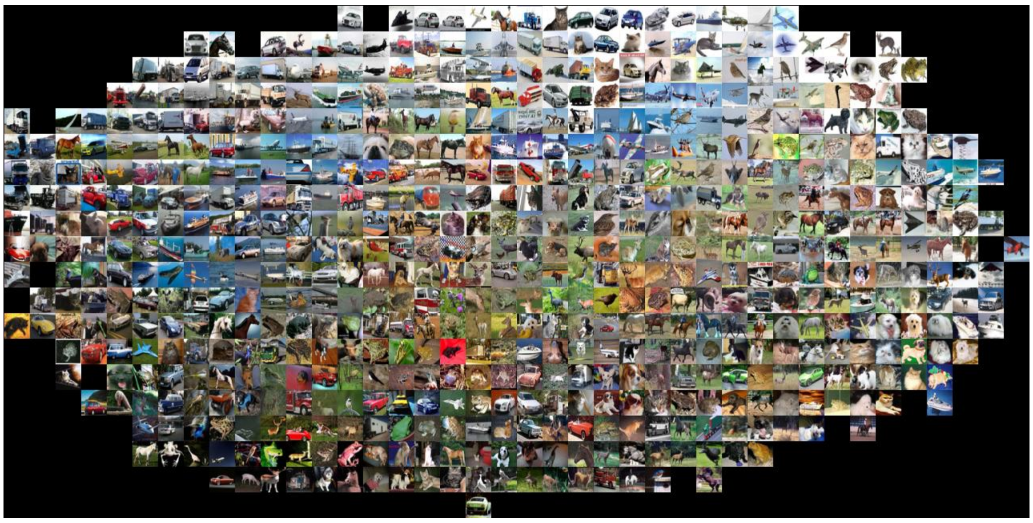
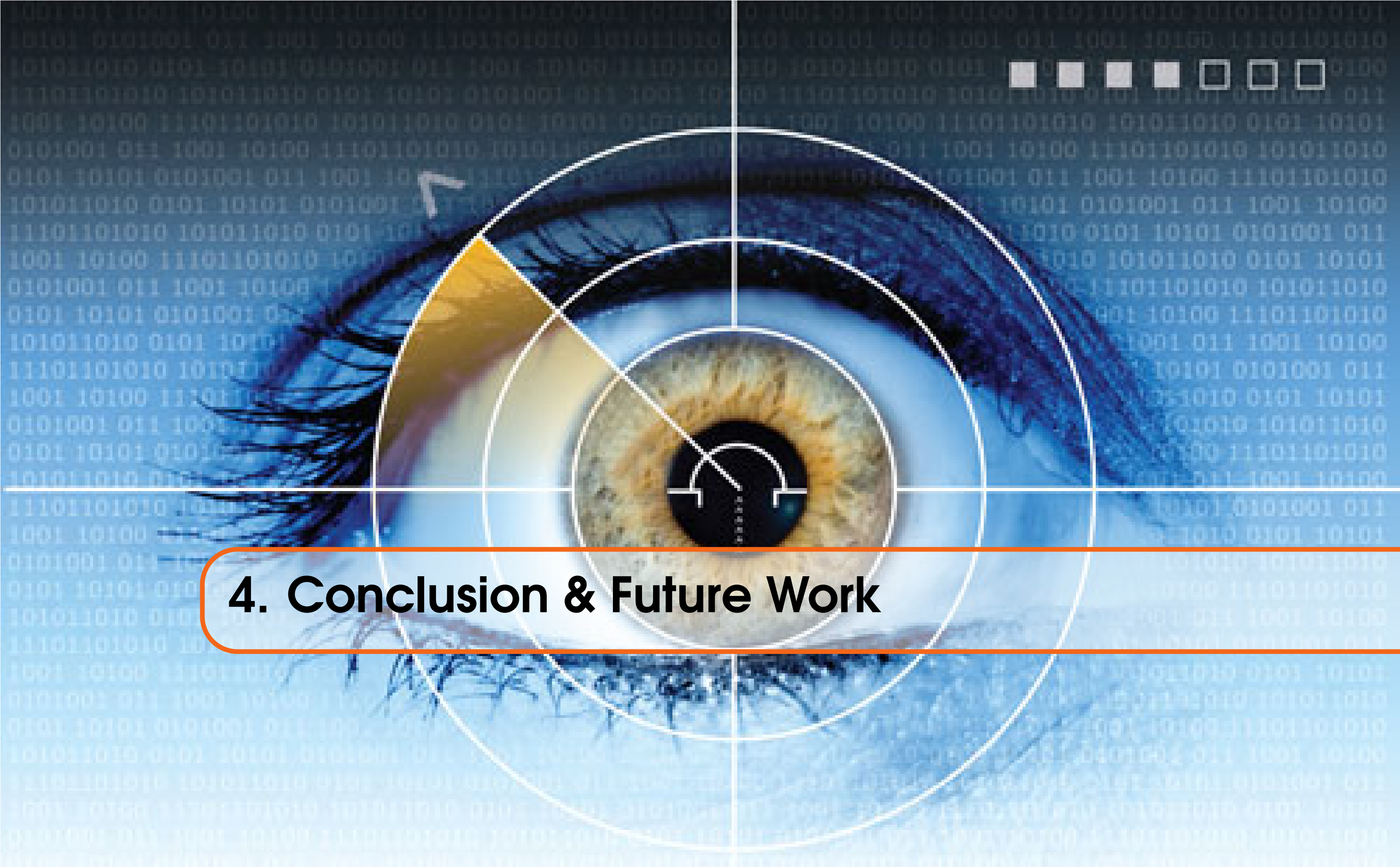


Figure 3.5: [1]The image above shows the images from the CIFAR-10 data set displayed based on their L2 distance from each other

As we see in the image above when classified on their pixel distance from each other the images within the Cifar-10 have a tendency of grouping together more so by their background rather than the classification of the images itself.



4. Conclusion & Future Work

4.1 Conclusions

In this assignment our goal was to implement a large-scale image classifier in order to determine the classes of the test images within the data sets CIFAR-10 and CIFAR-100. After completing this task we have arrived at two main conclusions, one it is difficult to classify images with naive methods such as KNN and two brute forcing such a task with a larger data set can help increase the performance of your method.

4.1.1 Conclusions Based On Results

When looking at our accuracy scores for CIFAR-10 we can see why we believe the amount of data has a large affect on the accuracy, when comparing the single image batch scores for each value of K used to the complete six image batch scores. From the results we can see that there is an obvious increase in accuracy for all value of K when the complete data set is used. This is due to the amount of training the predictor receives directly correlating to the amount of images available to gain values from. Furthermore when looking at our variation in accuracy that comes about due to our alteration of the hyperparameter K we can clearly see that it is not having as large an effect on the overall accuracy as the amount of training data is. classifier.

When testing with CIFAR 100, while our accuracy dropped in comparison to Cifar 10, we did provide results that indicated that our Classifier could still classify correctly about 30% for super class and about 20% for the sub class with K=1. Since Cifar10 and 100 are two different benchmarks, comparisons would best determined by use of other classifiers such as neural networks, SVM and multilayer perceptrons.

Overall from this study it has become obvious to us that KNN while easier to implement than others is not a very efficient or accurate classifier for this problem.

4.2 Future Work

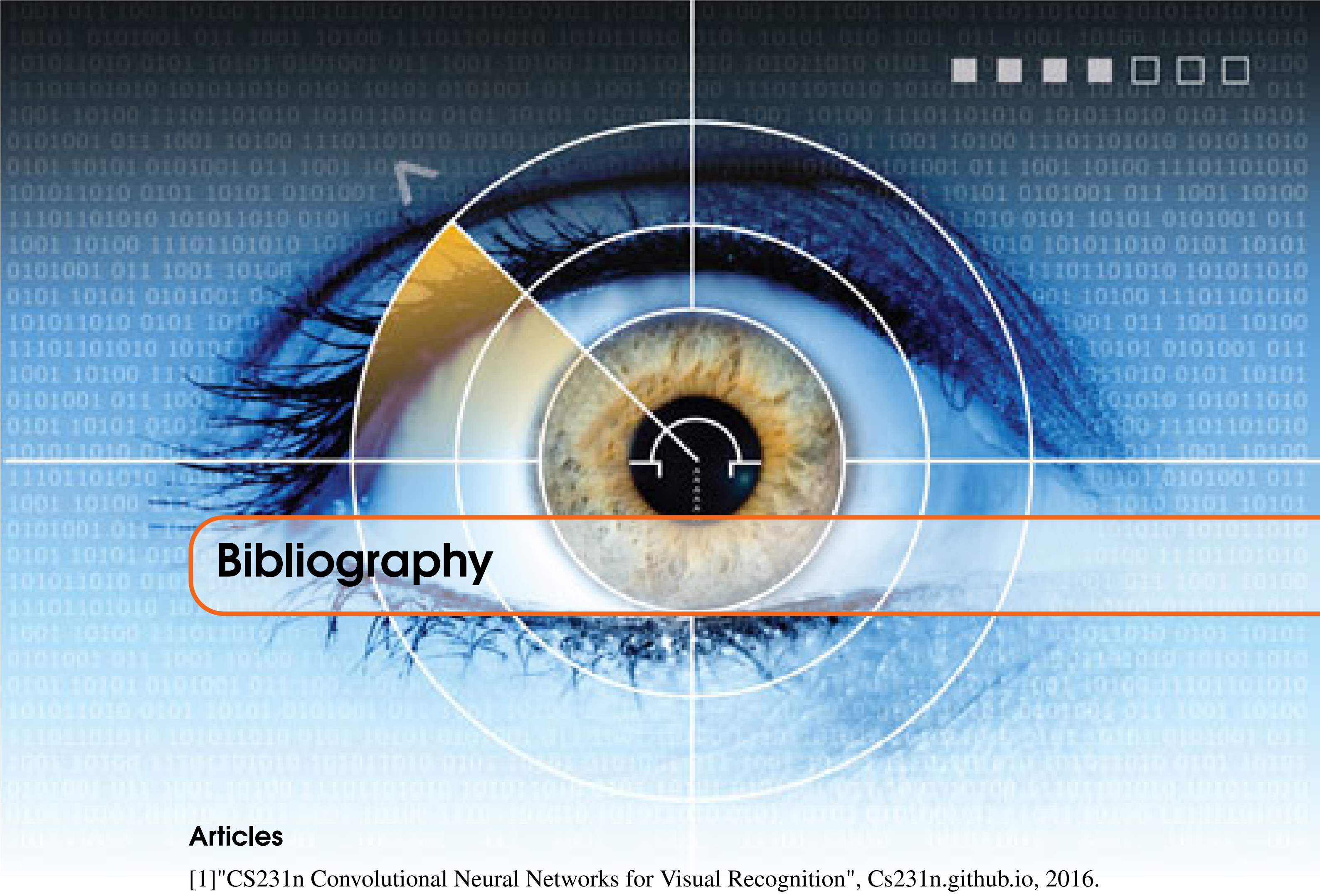
To further improve our KNN, we could utilise PCA, Softmax or use some form of normalization to improve the classification.

After completing this assignment it became obvious that testing alternate classifier methods would be the best way to find the one which would give us the greatest accuracy. Using KNN, however provided a solid benchmark that can be used as a initial start for comparison against other classifiers. With this classifier we can try other benchmarking tests such as MNIST, STL-10 and SVHN. After this our main goal would be to implement and compare the different classifiers to our current in order to find the most accurate or efficient processing method. A good area of study to look into would be neural network methods of image classification which would provide classifiers with more elegant methods than the brute force KNN we implemented.

As such in future works an initial aim will be to set up our implementation so that alternate methods can be easily added into our code with minimal alterations. In future, when approaching this problem classification with convolutional neural network would most likely be used as an initial benchmark rather than the final.

4.3 Personal Reflection

Overall, the assignment gave us a new perspective into image similarity matching and classification. We developed a better understanding on how data analytics works and the steps that a data analyst needs to go through. When working on this assignment, we revised classifiers, produced valid interesting data to investigate the reason for misclassifications, we developed new programming skills in the process of building the classifier and looked into performance improvements via hyperparameters. Upon completion of this assignment it was evident that evaluating images is not always simplistic given that the classifier has to deal with many challenges in order to provide a very accurate score (80+%).



Bibliography

Articles

[1]"CS231n Convolutional Neural Networks for Visual Recognition", Cs231n.github.io, 2016. [Online]. Available: http://cs231n.github.io/classification/#intro. [Accessed: 20- Sep- 2016].
[2]"CIFAR-10 and CIFAR-100 datasets", Cs.toronto.edu, 2016. [Online].
Available: http://www.cs.toronto.edu/ kriz/cifar.html. [Accessed: 20- Sep- 2016].