Simple Face Recognition Signal & Systems Final Project

Aidan Sharpe & Elise Heim

Contents

Title Page		1
1	Introduction	3
2	Protocol, Results, and Discussion	3
	2.1 The Discrete Cosine Transform - An Introduction to Image Pro-	
	cessing	4
	2.2 Turing a Rectangle into a Line	5
	2.3 Using the Lowest Frequencies for Facial Recognition	6
	2.4 Recognizing Faces	7
3	Conclusion	9
R	References	

1 Introduction

Facial recognition technology includes matching faces from an image with one from a database of faces. Facial recognition occurs in a variety of different contexts. For example, the . Passports include pictures of individuals, and are often necessary when traveling between countries. When passing through customs, the identity of a person has to be verified by comparing them to their passport picture. Historically, this has been accomplished by having one person look for similarities and differences between the person and their picture. Nowadays, more and more facial recognition devices are used for this purpose. These devices automatically perform this comparison, and are trained using large training datasets for improved accuracy.

There are multiple steps when it comes to facial recognition. As described by Mahmoud Hassaballah from South Valley University, these three important tasks include "face detection from a scene, feature extraction and representation[, ...] and face matching/classification." The first of these often includes detecting the edge of a face from its background. The second major task involves extracting features from the face in order to pack information, cleanse noise, as well as extract prominent features of a face. The final task requires classifying all of these features to determine the individual. All of these step are necessary to accurately recognize a face [1].

This technology is important because technological advancements mean that it can be used in more complex circumstances. With the rapid expansion of the internet, security concerns are increasing. Facial recognition systems can fall into the wrong hands, and be used for malicious reasons. Armed with facial recognition software and a video or picture posted on social media, hackers can discover sensitive information about a person. Even on an account that does not state the name of the individual, information such as their identity, residence, or location of work can be discovered through the internet. This means that it is important that this technology be used for benevolent purposes.

The objective of this project was to create a facial identification system in MATLAB. This was accomplished using an AT&T database. The Discrete Cosine Transform (DCT) of an image and a k-nearest neighbor (kNN) learning classifier were also utilized. kNN is a classifier which uses proximity to make predictions about the grouping of a data point. Then, the system will be assessed on its success rate, which will be plotted to identify the best values of k and dimension feature vector. This provides a greater understanding of facial identification.

2 Protocol, Results, and Discussion

This lab exercise had four components, each building on the last. The first component involved loading an image, and exploring the data from a frequency standpoint. The second component took a deeper look into how the frequency data was obtained. This insight was then applied in the third component to transform a subset of an image database containing 40 subjects with 10 images per subject into a database of frequency information. The efficacy of the frequency database was then tested in part four by comparing new images of the same subjects to the ones previously encoded.

2.1 The Discrete Cosine Transform - An Introduction to Image Processing

The first part of this lab exercise took an image from the AT&T Laboratories Face Database and broke it apart into its fundamental frequencies. What does frequency have to do with a picture? As a matter of fact, a lot!

All digital images are just a series of numbers. When interpreted by a graphics processor and a monitor, these numbers are transformed into different shades and colors of light. For now, however, we will focus on the numbers.

In this lab, all of the images were greyscale, so each pixel was assigned a brightness without any color data. In fact, that's all greyscale is, a brightness. By interpreting the brightness as a signal with varying intensity over space, it becomes much more obvious how the image is made up of frequencies. In this case, the frequencies were interpreted using a two-dimensional discrete cosine transform.

The output of the two-dimensional discrete cosine transform is also two dimensional. The width and height of the output are the same as the input image. In some sense, the input and output have the same resolution.

While the discrete cosine transform turns image data into frequency data, the inverse discrete cosine transform turns frequency data back into image data. The full process of turning an image into frequency and recovering the original image from the frequency data is seen in figure 1.

Putting an image through a discrete cosine transform and recovering it works quite well. The input image and the output image look identical. In fact, by treating the original image and the recovered image as matrices, where each entry is a pixel with a certain brightness, taking the difference of the two matrices yields the zero matrix with the same dimensions as the image. Since the difference is zero everywhere, we conclude that the original and recovered images are perfectly identical.

2.2 Turing a Rectangle into a Line

To be able to perform vector operations on the frequency data, the twodimensional information must be transformed into one-dimensional information. This was done by starting at the upper left corner and zigzagging across the image to the bottom right corner. This process is seen in figure 2. Imagine pinching the top-left corner with your left hand and the bottom right with your right hand. Then pull the ends apart into a narrow strip.

Looking back at the DCT performed in figure 1, most of the information is in the top left corner, while the rest of the image is mostly empty. By unwrapping this image into a vector of frequency information, the plot shown in figure 3a



Figure 1: Two dimensional DCT and inverse DCT

is obtained. Notice how little the higher frequencies contribute. To better see what is going on with the lower frequencies, only the first 500 frequencies are shown in figure 3b. Even still, most of the information is heavily concentrated in the lowest frequencies.

For almost all photographs, most of the image information is low frequency information. Most real-world photographs will have few sharp edges. Seen below in figure 4, while the frequency data is different for two different images, it is concentrated at lower frequencies for both. In general, images can be reduced to mostly low-frequency information.

Does this mean that the higher frequencies can be discarded to save data? Yes! In fact, the JPEG compression algorithm uses the two dimensional DCT to aid in compressing images. Unfortunately, there is a tradeoff to using this approach in a compression algorithm. While the data savings are massive, sharp details are easily lost. As seen in figure 5, an image using only a quarter the data is still completely recognizable. However, some of the hard edges have become blurred. This is because only high frequency data can produce sharp edges, and it is this high frequency data that has been discarded.

Knowing that the lowest frequencies are enough to distinguish a picture will help when using the frequency information to aid in facial recognition.

2.3 Using the Lowest Frequencies for Facial Recognition

The image of the face shown above is the first in a database of 400 images. Each image is greyscale and of the same 92 by 112 resolution. The database contains forty subjects and ten distinct images of each subject. This section of the exercise involved taking the DCT of the first five images from each subject



Figure 2: 2-D to 1-D using a zigzag approach



(a) The full image in the frequency domain (b) The first 500 frequencies in the image

Figure 3: Plotting the image in the frequency domain

and truncating it to a desired number of frequencies.

In effect, a database of frequency information was established. Since low frequency information is enough to make out the most important features of images, the similarity of two images can be compared by looking at the difference in low frequency information.

Therefore, a small number of frequencies should suffice for building a passable facial recognition system. Since there are only forty subject, it is unlikely that two subjects will look very alike. This low-resolution approach should work well given the scenario. For much larger datasets, a more advanced approach should be taken since there will likely be less variation between subjects.

To accomplish the task of setting up the frequency database, the program, face_recog_knn_train.m was provided. The program was provided with a range of subjects, in this case, 1 through 40, and a number of frequencies for each entry to include. The database was saved to disk in a file called raw_data.mat. This file could then be loaded to set up the appropriate database values in memory.



Figure 4: Comparing two subjects with 9, 35, and 100 frequencies

2.4 Recognizing Faces

The recognition approach implemented was the k-nearest neighbors algorithm. At a higher-level, this algorithm compares the input to all the values in the database and then sorts the values in the database based on which entries were closest to the input. The first k entries in the sorted list are selected, and whichever element appears most often in the selection is returned as the best guess.

In the context of the frequency database, the algorithm works as follows. First, an image of a face to be recognized is input into the algorithm. The DCT of the image is then taken, and the resulting frequency data is truncated to match the database. This frequency data comes in the form of a column vector. When the database data is read into memory, each entry is a row vector of frequency data. To find the distance between to an entry in the database, the Euclidian distance of the difference vector must be taken. To be able to



(a) All 10,304 frequencies



Figure 5: The effects of removing high frequency data

take this difference in the first place, however, the frequency column vector is transposed into a row vector to match the form of the database entries.

Just as the distance between two points in two dimensions is the length of the vector connecting them, the same is true in any number of dimensions. To find the length of the vector, the square root of the sum of the squares of each component is taken. This can be hand calculated for a vector of dimension N using equation 1.

$$\|\vec{v}\| = \sqrt{\sum_{i=1}^{N} v_i^2}$$
(1)

These distances are stored in a list, which is then sorted in ascending order. In doing so, the shortest distances appear first. The first k elements of the list are paired back to their corresponding subjects to get a list of the k closest subjects. The subject that appears the most in this list is the subject recognized by the algorithm.

To evaluate the efficacy of this facial recognition algorithm, it was tested on the remaining five images for each subject. If the algorithm correctly guessed the subject, it was counted as a "hit". The success rate of this algorithm was defined as ratio of "hits" to total images tested.

The algorithm was evaluated at several different values of k, and the database was tested with a range of frequency quantities. The results of these tests are seen in figure 6. The different values of k tested were 1, 3, 5, and 7, and the different number of frequencies tested were 25, 40, 55, 70, 85, and 100. The results clearly show that the increasing the number of frequencies does not do much to improve the results, but increasing the value for k has a severe negative

impact on accuracy.

Since there are only 5 database entries for each subject, when k is close to or greater than 5, the chance that other subjects fall within the k nearest neighbors increases. The likelihood that one or two images of a subject are close to another is quite high, but for larger values of k, random chance plays a much bigger role.

The reason that the number of frequencies does not seem to play much of a role in accuracy probably has to do with the fact that most of the information is encoded in the lowest few frequencies. Increasing the number of frequencies increases the amount of resolution with diminishing returns. There may also be a sweet spot since high frequency information may introduce unwanted noise into the system.



Figure 6: Efficacy of k-nearest neighbors with different values for k and number of frequencies

3 Conclusion

In order to increase understanding of facial identification systems, this laboratory was completed. This system uses a kNN classifier as well as DCT to turn images into frequencies. An AT&T database of faces was also utilized for the system. By plotting the DCT of one image and then taking the inverse, one can see how an image can be converted into signals and then recreated. By taking the logarithm of these signals, one can observe how the greater DCT coefficients are concentrated in the upper left-hand corner.

For the next section, the program findfeatures.m was provided to find the one-dimensional DCT feature vector of a picture. This image is then converted into a DCT feature vector by zigzagging through the image in order to scan and place each of the coefficients into a one-dimensional array. Although this program was run for multiple individuals in the database, most of the information in the images remained in low frequencies. This goes to prove that lower frequencies are enough for authentication in facial recognition.

Another program was supplied for the following section. Titled

face_recog_knn_train.m, it utilizes the previously used findfeatures.m program to find the vector features of the first five pictures of each subject. It also labels the subjects, and uses the results to train the face recognition system and output the results to a file called raw_data.mat. The raw data file includes information such as the number of subjects, feature vectors and labels which are all necessary for facial recognition.

The last part of this laboratory involved using the DCT and kNN classifier to identify the last five pictures of each subject to determine the accuracy of the facial recognition system. The pictures are used to label and determine the training vectors with the smallest distance from the recognized vector. Once this is found, the feature vector is then labeled as the subject. Afterwards, the success rate of the system was determined to be above 90% accurate.

Through the analysis of different programs and implementation of facial recognition technology, much knowledge can be acquired. Through the use of an AT&T database, finding the DCT of an image, and kNN classifiers, a facial recognition system was developed. The success of the system was proven, and a greater understanding of facial recognition was established.

Facial recognition systems are utilized by governments and private companies globally. While they are not perfect, they are still incredibly useful [2]. Facial recognition technology is an incredible advancement that will only improve with time and optimization. Such systems have the potential to be used in convenient and creative ways that seem difficult to dream up.

References

- M. Hassaballah and Saleh Aly. "Face recognition: challenges, achievements and future directions". In: *IET Computer Vision* 9.4 (2015), pp. 614-626. DOI: https://doi.org/10.1049/iet-cvi.2014.0084. eprint: https: //ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/ietcvi.2014.0084. URL: https://ietresearch.onlinelibrary.wiley. com/doi/abs/10.1049/iet-cvi.2014.0084.
- Wikipedia contributors. Facial recognition system Wikipedia, The Free Encyclopedia. [Online; accessed 12-December-2023]. 2023. URL: https:// en.wikipedia.org/w/index.php?title=Facial_recognition_system& oldid=1187927222.