2024



Project work/report?

Recognizing hand signs using principal component analysis

By Mailin Brandt and Finian Landes

Supervisor: Robyn Steiner-Curtis

Gymnasium Liestal

Primary Subject: Applied Mathematics/Physics

Contents

[1. Introduction 3](#_Toc185178174)

[2. Guiding Question 3](#_Toc185178175)

[3. Theory and Results 3](#_Toc185178176)

[3.1 Dataset 3](#_Toc185178177)

[3.2 Principal component analysis (PCA) 4](#_Toc185178178)

[3.2.1 Theory 4](#_Toc185178179)

[3.2.2 Results 5](#_Toc185178180)

[3.3 K-Means clustering 6](#_Toc185178181)

[3.3.1 Theory 6](#_Toc185178182)

[3.3.2 Results 6](#_Toc185178183)

[4. References 12](#_Toc185178184)

[4.1 List of references 12](#_Toc185178185)

[4.1 List of illustrations 12](#_Toc185178186)

[5. Acknowledgements 13](#_Toc185178187)

[6. Appendix 13](#_Toc185178188)

# Introduction

The aim of our project was to distinguish five different hand signs using machine learning. At the beginning of the semester, we had to decide if we wanted to do our project in physics and rebuild a historical physics experiment or if we wanted to learn about machine learning in applied mathematics. For Finian the decision on machine learning was clear from the beginning on, because he is interested in computer science and could imagine studying a similar topic. For Mailin the decision was harder. She was interested in both topics, but in the end, she decided to improve her programming skills. We started in class to program a principal component analysis (PCA) algorithm from scratch. This included multiple theory inputs and worksheets where we could step by step learn the function of the algorithm. After we finished the basic algorithm, we had to pair up. For us it was clear that we would work together. We have worked together on different projects before and are a productive team which is also connected through friendship. This leads to an efficient but also nice work together.

We could then use this basic algorithm for our own project and optimize it for our own needs.

The idea for distinguishing different hand sing was set very fast. We did a short brainstorming session with all the ideas which we had in mind. We had a lot of different ideas but soon concluded that we find the topic of hand recognition interesting. Different ideas for working with hands came into our minds. To distinguish different hands was for us a bit boring so we thought of other ways and emojis came into the discussion. We thought about comparing our hands to emojis, but we turned down this idea, but the emojis were used as an inspiration for our decision on the different hand signs. We decided on five different hand signs.

We created our own data set. Our algorithm was tested using our data set and optimized to the needs of our dataset.

Because we had an efficient way of creating our own data set, we had enough time and motivation to write an additional algorithm. After some evaluation we decided to implement the k-means clustering algorithm. We got a short input for this algorithm and then used those ideas to program it ourselves.

# Guiding Question

# Theory and Results

## Dataset

For our program we needed to create a dataset containing our five hand signs, shown by different hands. To create this in an efficient manner we built a camara rig from wooden boards, shown below:



Figure 1: Camera Rig used to take pictures of the hand signs

We then ensured that the light was the same for each person. Using a Leica (*Reference)* Camera we then took square images. This would simplify the process later when processing the images. Furthermore, our camera rig was set up exactly to fit one hand into the frame so we wouldn’t need to crop our images further. Worth mentioning here is that the camera was really overkill, as we rescaled the images drastically, so a phone would have sufficed as well but some kind of tripod would still be recommended to make identical pictures.

To automate the process, we wrote a program which automatically named the images, so all the image names contained the name of the hand sign. Our program also rescaled the images to 256x256pixels and converted them to grayscale. For our other programs to work the images were read row by row to form a vector. All of this was done using os[[1]](#footnote-2) and Pillow[[2]](#footnote-3).

We chose our five hand signs arbitrarily but ensured that they were quite different from each other so recognition would be easier.



Figure 2: Processed images of the five hand signs. From left to right: Easy, Metal, Peace, Thumbs up, Pistol

## Principal component analysis (PCA)

Principal component analysis (PCA) is an algorithm used to simplify complex datasets. It identifies the most important patterns and by doing so it helps to reduce complexity but maintain the essential information.[[3]](#footnote-4)

### Theory

PCA works by transforming the original dataset into a new coordinate where the axes (principal components) capture the maximum variance, the difference between a point and the mean. To do this the data is first normalized by subtracting the mean of all points from each point. Using this data the covariance matrix is calculated. This matrix tells us whether the change in one feature is associated with the change in another feature. In our case this matrix would tell us how the brightness of one pixel is related to the brightness of another pixel in another image. For example, the center of the hand which is in the same place for each hand sign might have a high covariance and the fingers which differ in all pictures have a low covariance. Using that matrix, we calculate the eigenvectors, which are called eigenfaces or eigenhands in our specific case. These eigenvectors are the directions of the axis which show the most important features of the data, ordered from the most relevant to the least relevant. This ordering allows us to disregard the last eigenvectors without losing important data.1



Figure 3: The first five eigenfaces

To then recognize the hand sign shown in the images of the test dataset we searched for the closest image, vector, of the train set to our test image and so the test image would likely show the same sign as the train image it was closest too.

### Results

To prove our concept, we first created a small dataset only consisting of three pictures of three different hand signs of Mailin’s hand. We tried our algorithm on these pictures with a random train and test set. The test set contained three random chosen pictures which were then excluded from the train set. We then saw that our algorithm worked. We had a success rate between 66 and 100 percent, which was higher than just guessing. Therefore, we could conclude that our algorithm worked with distinguishing different hand signs.

We then created a full dataset containing 150 pictures of Mailin’s hand and decided on ten different signs. With this data set and a randomly created train and test set we came to a success rate around 90 percent. All those 150 pictures were then used as a train set. As a test set, we did three pictures per sign of Finian’s hand. This led to the success rate dramatically dropping to about 50 percent.

We concluded that our dataset was not diverse enough with only Mailin’s hand and therefore added more people's hands. To be more efficient we reduced the number of signs to five and only did ten pictures per sign. At the end we had four times ten pictures per sign and once 15 pictures per sign and therefore overall 275 pictures in our train set. When we used Mailin’s and another person’s hand as train set the success rate to recognize an unknown hand ‘s sign rose to 93 percent and with a third hand to 100 percent. Adding more people ‘s pictures did not change the success rate.

Overall, we could therefore conclude that our principal component analysis algorithm worked perfectly with recognizing five different hand signs on an unknown hand.

When trying to further optimize our PCA algorithm, we investigated using only the most significant eigenfaces, which means the *n* first ones. Using only 50 instead of 275 we did not have any decrease in our success rate. So, with only around 20 % of the eigenfaces we still had a perfect score, this would make the algorithm way more efficient, if we were to drastically increase our dataset size to a point where performance played a role.

## K-Means clustering

K-Means clustering is, as the name implies, a clustering algorithm. It separates a dataset into the most compact *k* different clusters. Important to note here is that K-means is nondeterministic, which means that for different starting conditions or so-called different initial centroids the algorithm produces different clusters.

### Theory

The Algorithm works by first choosing *k* centers, called centroids, for the clusters, these can be chosen arbitrarily or with some kind of criteria. For simplicity’s sake we chose k points from our dataset to be our initial centroids. The algorithm then goes over each point and assigns it to the closest centroid. Then the centroids are updated by taking the mean of the points assigned to the old centroid. This gets repeated until some kind of converging criteria is fulfilled. In our case this criterion was that the sum of all distances between the old and new centroids was nearly zero.[[4]](#footnote-5)



Figure 4: Five arbitrarily chosen centroids when dividing our dataset into 15 clusters

To score our k-means we built a function which takes in the number of images per sign per cluster. Then we calculated a score over all clusters where a score of 100% would mean all clusters only contain one sign and the lower the score the more mixed the clusters are.

### Results

Even though the results of the PCA were already perfect we went on to create a clustering algorithm. This algorithm was never intended to be used to classify the images but more to analyze patterns in similar pictures, classified as similar by the computer. To make it run faster we used the already by PCA optimized data as points. At first our results were bad, so the signs were distributed evenly in each cluster, as we found out later, this was linked to our initial centroids which had been random points in space. After modifying the initialization to the way mentioned above, choosing random points from the dataset, we got better results. Using five clusters the results were still quite mixed up, so the clusters did not make any sense for us. Increasing the number of clusters made the clusters better and we found that 15 to 20 clusters worked best for our dataset. Looking at these clusters many things became clearer for us. The algorithm made quite pure clusters for thumbs up and metal which makes sense as the signs are fairly different from the others. Peace and Pistol on the other hand got clustered together often, from this we can conclude that the thumb being out was a more relevant feature for the algorithm. Whether the index finger or pinky was shown seemed to be less relevant. Shown below are three randomly chosen clusters from a run with a high score, dividing our data into 15 clusters.

In cluster one the main clustering criteria seems to be two fingers pointed out next to each other, the position of the hidden fingers does not seem to be well recognizable for the computer.



Figure 5: Cluster one

In cluster two you can see the phenomenon mentioned above, as it seems that the thumb is the important factor to order as well as the position of the second finger, seen in the last image, thumbs up, and in the two easy in the first row.



Figure 6: Cluster two

In cluster three the K-means algorithm seems to ignore the pinky which might be because the pinky is small, or the index finger and thumb are so close together that this information outweighs the pinky being there as well.



Figure 7: Cluster three

* 1. **Reflection**

Our project overall worked well. We had a nice and productive working environment. We sometimes struggled a bit with our code but managed to solve it quite fast. We are very pleased with the results of our PCA Algorithm. We did not expect it to wort that well after such a short amount of time. The program to name and optimize our pictures saved us a lot of time and we would definitely recommend doing it this way.

We are aware of the fact that our dataset is not very diverse. It only contains with-colored hands. We do not know if our programs also work on darker-colored hands, but we can assume that it won’t work based on other studies presented to us by Mrs. Steiner. If we had had more time on our project we would increase the diversity in our dataset with different colored hands. This also applies to our test set, we only tested with one person the same age as the others in the train set. It would also be interesting to know whether an older and or younger person would influence our results and if so by how much.

At the beginning of working with the K-means clustering algorithm we were unsure if the algorithm worked because we had very bad results. This resulted in us rewriting the algorithm in an empty file with two-dimensional data so it would be plottable. From this we found that our initializing was very bad, but that our algorithm worked. In the end the issue was with the data that came from the PCA algorithm as we used it not converted to the right format.  
The results of K-Means clustering were good, but not very satisfactory. Other clustering methods might have worked better here. The issue with our dataset was that overlapping features in hand signs, such as the presence or position of a finger, sometimes seem to be more important than differences, e.g. whether the pinky is shown or not. Another way we could cluster our data would be clustering which does not assign items to a single cluster. Rather we would assign points to all the clusters, with a weight which tells us how much the point belongs to the cluster. This would be so called fuzzy clustering. [[5]](#footnote-6) In our case we could extend our K-Means clustering to fuzzy C-Means[[6]](#footnote-7) clustering. That could be better suited for our use case. We could then give the user some kind of certainty with each picture that it is the corresponding hand sign.

# References

## List of references

Python Documentation, Art. “os”:

<https://docs.python.org/3/library/os.html> (02.12.24)

Jeffrey A. Clark and contributors, Pillow Documentation:  
<https://pillow.readthedocs.io/en/stable/> (02.12.24)

Wikipedia, Art. “Principal component analysis”:   
<https://en.wikipedia.org/wiki/Principal_component_analysis> (25.11.2024)

Wikipedia, Art. “K-means clustering”:

<https://en.wikipedia.org/wiki/K-means_clustering> (25.11.2024)

Wikipedia, Art. “Fuzzy Clustering”:

<https://en.wikipedia.org/wiki/Fuzzy_clustering> (13.12.2024)

Medium, Art. “ Fuzzy C-Means Clustering (FCM) Algorithm”: <https://medium.com/geekculture/fuzzy-c-means-clustering-fcm-algorithm-in-machine-learning-c2e51e586fff> (13.12.2024)

## List of illustrations

[Figure 1: Camera Rig used to take pictures of the hand signs 4](#_Toc185170414)

[Figure 2: Processed images of the five hand signs. From left to right: Easy, Metal, Peace, Thumbs up, Pistol 4](#_Toc185170415)

[Figure 3: The first five eigenfaces 5](#_Toc185170416)

[Figure 4: Five arbitrarily chosen centroids when dividing our dataset into 15 clusters 7](#_Toc185170417)

[Figure 5: Cluster one 8](#_Toc185170418)

[Figure 6: Cluster two 9](#_Toc185170419)

[Figure 7: Cluster three 10](#_Toc185170420)

# Acknowledgements

First and foremost, we want to thank Mrs. Steiner-Curtis for her support and guidance in this project. Furthermore, we want to thank everyone from our class willing to let us photograph their hand for our dataset.

Finally, we want to thank our families for reviewing our text and providing us with a camera for creating our dataset.

# Appendix

1. Python Documentation, Art. “os”: <https://docs.python.org/3/library/os.html> (02.12.24) [↑](#footnote-ref-2)
2. Jeffrey A. Clark and contributors, Pillow Documentation: <https://pillow.readthedocs.io/en/stable/> (02.12.24) [↑](#footnote-ref-3)
3. Wikipedia, Art. “Principal component analysis”: <https://en.wikipedia.org/wiki/Principal_component_analysis> (25.11.2024) [↑](#footnote-ref-4)
4. Wikipedia, Art. “K-means clustering”: <https://en.wikipedia.org/wiki/K-means_clustering> (25.11.2024) [↑](#footnote-ref-5)
5. Wikipedia, Art. “Fuzzy Clustering”: <https://en.wikipedia.org/wiki/Fuzzy_clustering> (13.12.2024) [↑](#footnote-ref-6)
6. Medium, Art. “ Fuzzy C-Means Clustering (FCM) Algorithm”: <https://medium.com/geekculture/fuzzy-c-means-clustering-fcm-algorithm-in-machine-learning-c2e51e586fff> (13.12.2024) [↑](#footnote-ref-7)