PA Mailin, Finian

**Contents**

[Introduction 3](#_Toc183419949)

[Theory and Results 3](#_Toc183419950)

[Principal component analysis (PCA) 3](#_Toc183419951)

[Theory 3](#_Toc183419952)

[Results 4](#_Toc183419953)

[K-Means 4](#_Toc183419954)

[Theory 4](#_Toc183419955)

[Results 4](#_Toc183419956)

[Reflection 4](#_Toc183419957)

[References 4](#_Toc183419958)

[List of references 4](#_Toc183419959)

[List of illustrations 4](#_Toc183419960)

# **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 rebuilt a historical physic experiment or if we wanted to learn about machine learning in Applied Math. For Finian the decision on machine learning was clear from the beginning, because he is interested in computer programs and could imagine studying a similar topic. For Mailin the decision was harder. She was interested in both themes, but in the end, she decided to improve her programming skills. Then we started in class to program a Principal component analysis (PCA) algorithm using worksheets. This was an intensive work which included multiple inputs and worksheets where we could step by step program the algorithm. After we finished the basic algorithm, we had to pair up. For us it was directly clear that we would work together. We have worked on different projects together before and build a productive team which is also connected through a friendship. This leads to an efficient but also nice work together.

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

The idea for distinguishing different hand sing was set very fast. We did a short brainstorming 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 got 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 use the k-mean method. We got a short input n this algorithm and then sed those ideas to program it ourselves.

# **Theory and Results**

## **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.[[1]](#footnote-2)

### **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. So 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.[[2]](#footnote-3) Shown below are the first 5 eigenfaces.



To then recognize the hand sign shown in the images of the test dataset we searched for the closest image of the train set to our test image and we compared the sign of both images.

### **Results**

## **K-Means**

K-Means is 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 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.[[3]](#footnote-4) Shown below are 5 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**

# **Reflection**

# **References**

## **List of references**

## **List of illustrations**

1. Chat GPT [↑](#footnote-ref-2)
2. Wikipedia, Art. “Principal component analysis”: <https://en.wikipedia.org/wiki/Principal_component_analysis> (25.11.2024) [↑](#footnote-ref-3)
3. Wikipedia, Art. “K-means clustering”: <https://en.wikipedia.org/wiki/K-means_clustering> (25.11.2024) [↑](#footnote-ref-4)