



University of Hamburg  
MIN Faculty  
Department of Informatics

# Project Report Machine Learning

**Kaggle**  
**Google Landmark Recognition Challenge**

---

Marcel Rohde  
Thorben Wiese



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Data Analysis</b>	<b>3</b>
<b>3</b>	<b>Features</b>	<b>9</b>
3.1	Dimension Reduction . . . . .	9
3.2	Clustering . . . . .	11
<b>4</b>	<b>Conclusion</b>	<b>13</b>
	<b>References</b>	<b>14</b>

---



---

# List of Figures

2.1	Landmark_ID Distribution . . . . .	4
2.2	Image for Landmark_ID 9633 . . . . .	5
2.3	Image for Landmark_ID 6051 . . . . .	5
2.4	Variances . . . . .	6
2.5	Maximum Pixel Values . . . . .	6
2.6	Minimum Pixel Values . . . . .	7
2.7	Amount of Pixels Scatter Plot . . . . .	7
2.8	Amount of Pixels Histogram . . . . .	8
3.1	PCA result . . . . .	10
3.2	Most frequent landmarks from PCA . . . . .	10

---

# 1 Introduction

Kaggle is a platform owned by the Kaggle Inc. which is owned by Alphabet Inc. (Google) providing multiple so called challenges in the field of data science, predictive modeling and data analysis. The Kaggle challenge aims at solving so far unsolved tasks or finding a better solution for already solved tasks in a crowd-sourcing fashion. Some challenges can be solved for monetary prizes, others are hosted for knowledge or training purposes. In order to solve a challenge one must register and submit a solution to the platform to get a score for the submitted solution [3].

The Google Landmark Recognition Challenge aims at detecting different landmarks in images, such as the Eiffel Tower or the Leaning Tower of Pisa [1].

We chose the Google Landmark Recognition Challenge because we are interested in image processing tasks and the challenge provides a lot of training data. We also wanted to try out Convolutional (Deep) Neural Networks which we previously discussed in the lecture.

The data provided with the challenge contains mainly of two CSV files. The file for training (train.csv) providing the IDs, URLs and Landmark IDs and the file for testing (test.csv) providing IDs and URLs [2].

In order to work with the images we wrote a script to scrape all URLs and download the images to our local machine. The script will be attached to this report.

Our approach is to first analyse the data to get a good understanding of it with the help of characteristic numerical values such as value ranges, min or max values and variances, which will be plotted for better visual understanding.

---



## 2 Data Analysis

Since the data provided with this challenge is contained purely out of images, we will focus on image-related characteristical numerical values for our analysis.

We selected the distribution of landmarks, the dimenions of the images, the amount of pixels, the pixel range (from min to max) and the pixel variance (max - min) to get a better understanding of what the images' shapes are.

The training dataset holds 1225029 URLs to images and the corresponding IDs and Landmark IDs. The test dataset holds 117703 URLs to images with the corresponding IDs and will be purely used for validation of the trained model in Assignment 11.

We have selected about 78.000 images of the training set and about 20.000 images of the test set for computation time reasons. The calculation for all the following results can be found in the assignment10.py script attached to this report.

The distribution of the training set Landmarks can be seen in figure 2.1. The vast majority of landmarks occurs less than 100 times or even once. Only a few occur very frequently. The most frequent landmarks are displayed in figure 2 which ID 9633 and ID 6051 being the most frequent ones. Figure 2.2 and 2.3 show images of the training dataset with these Landmark IDs.

The variance or pixel range of the images can be seen in figure 2.4. It is clear to see that most images have a variance from 255, which is the full range of a pixel. The Distribution of the maximum pixel value and minimum pixel value can be found in figure 2.5 and 2.6. Both show a normal distribution towards 255 for the maximum values and 0 for the minimum values. The total amount of pixels is displayed as a scatter plot in figure 2.7 and as a histogram in figure 2.8.

---



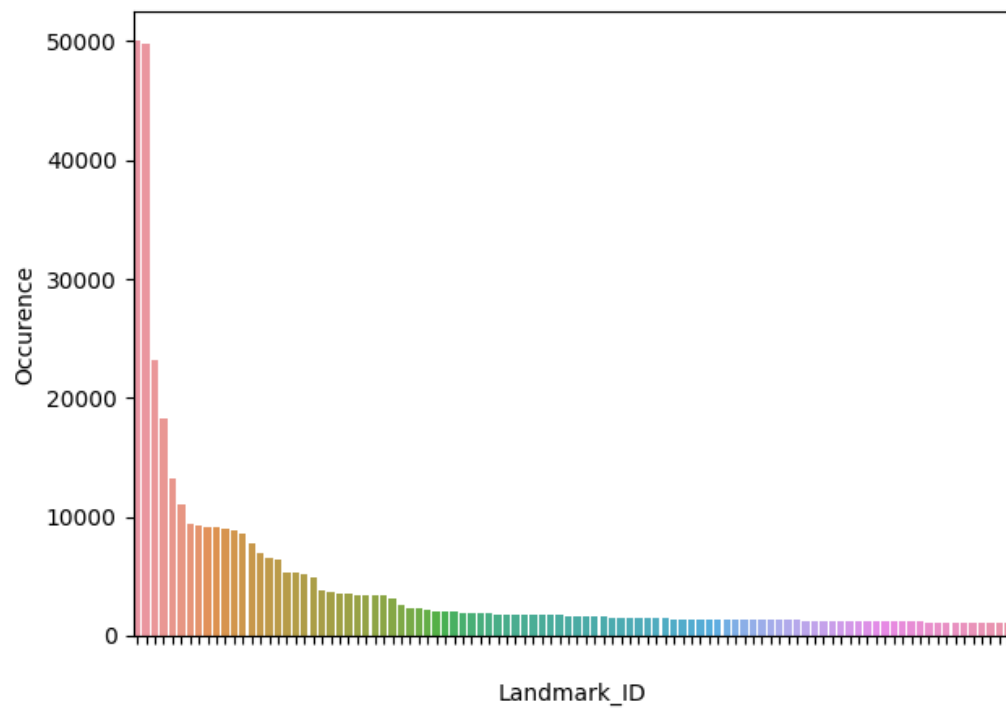


Figure 2.1: Landmark\_ID Distribution

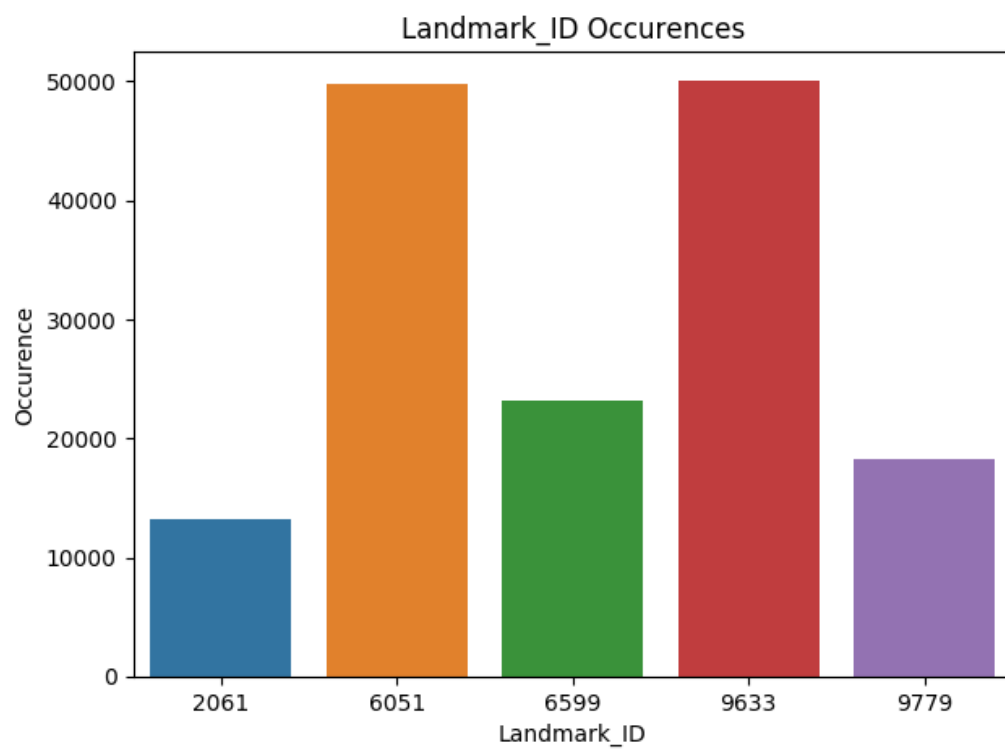




Figure 2.2: Image for Landmark\_ID 9633



Figure 2.3: Image for Landmark\_ID 6051

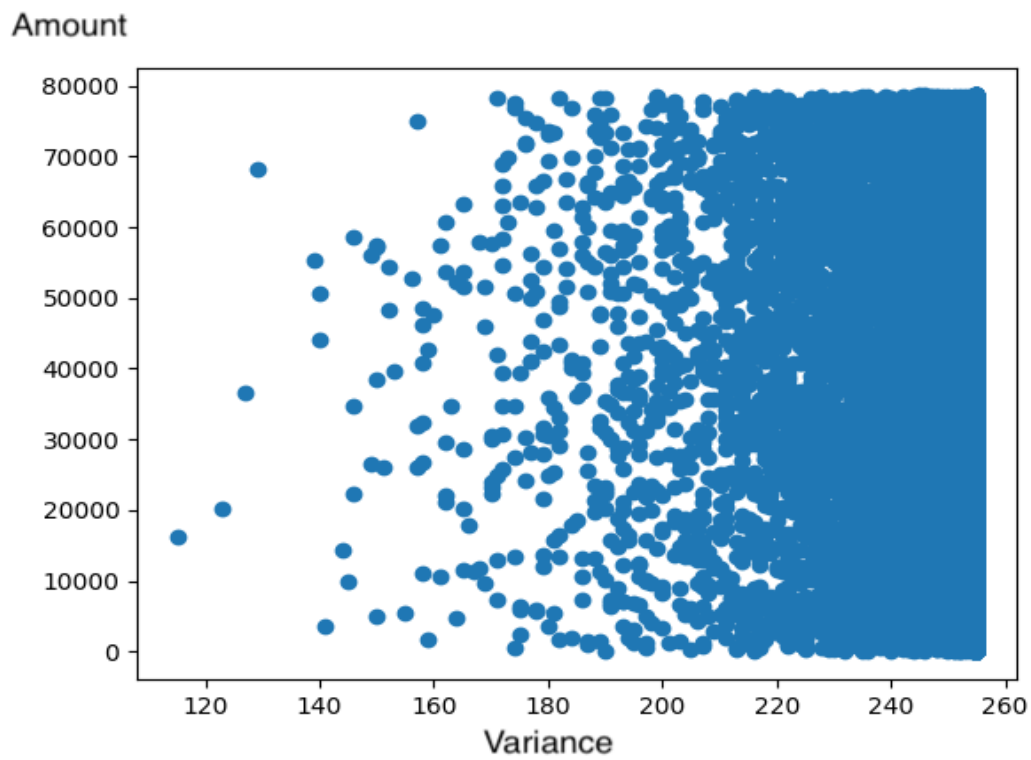
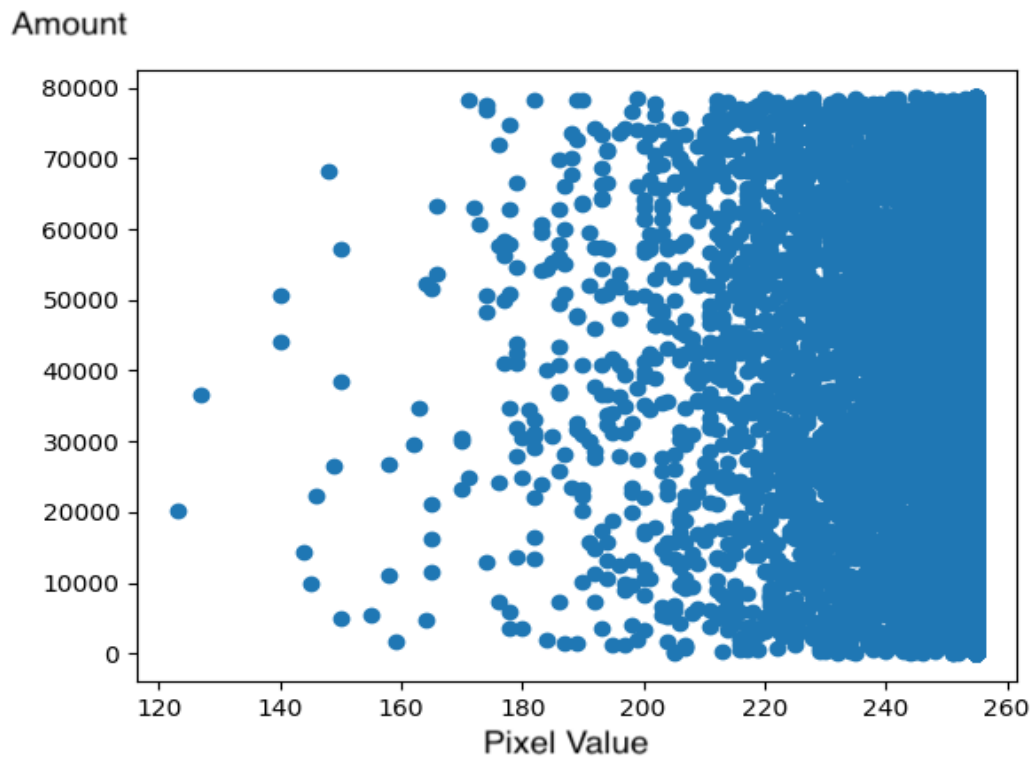


Figure 2.4: Variances

Figure 2.5: Maximum Pixel Values

---

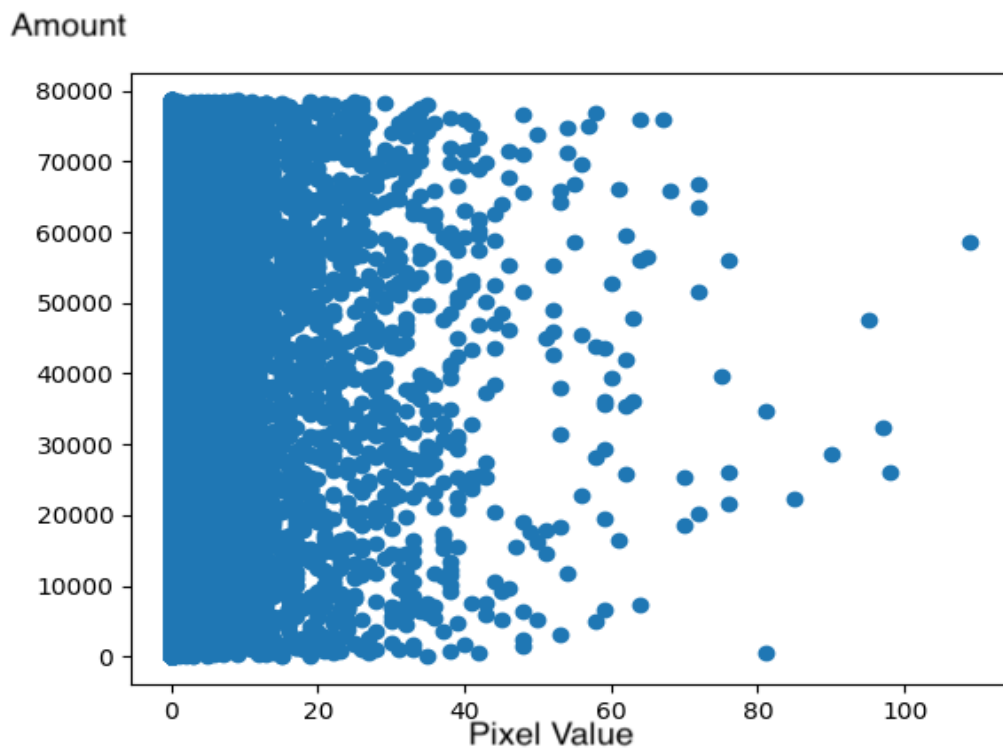


Figure 2.6: Minimum Pixel Values

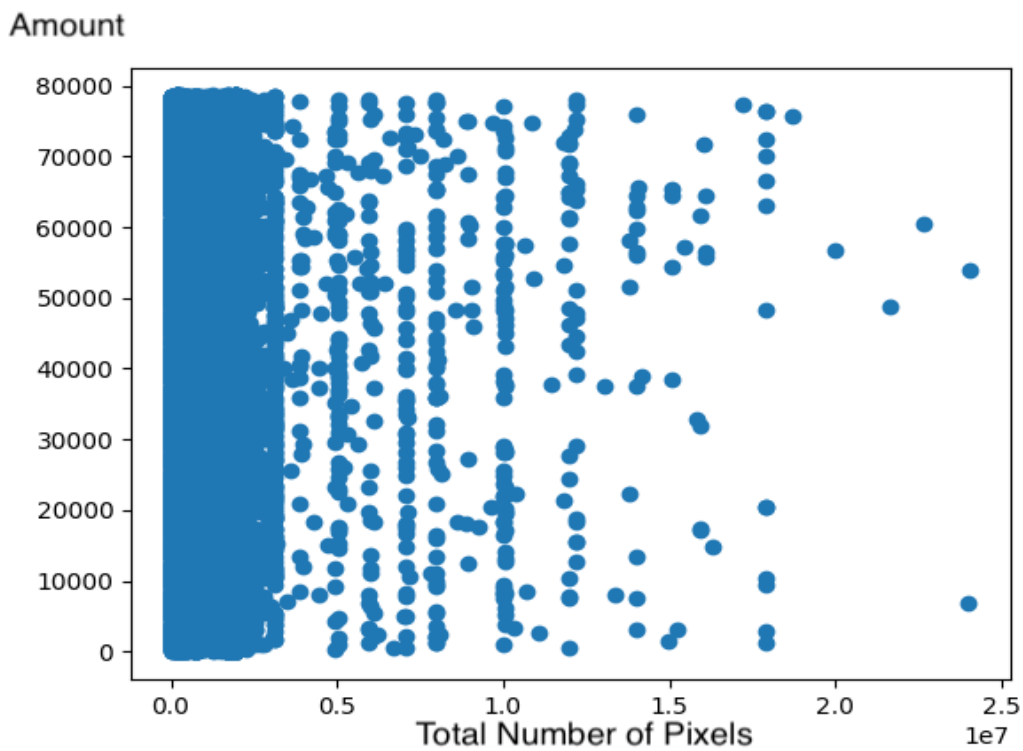


Figure 2.7: Amount of Pixels Scatter Plot

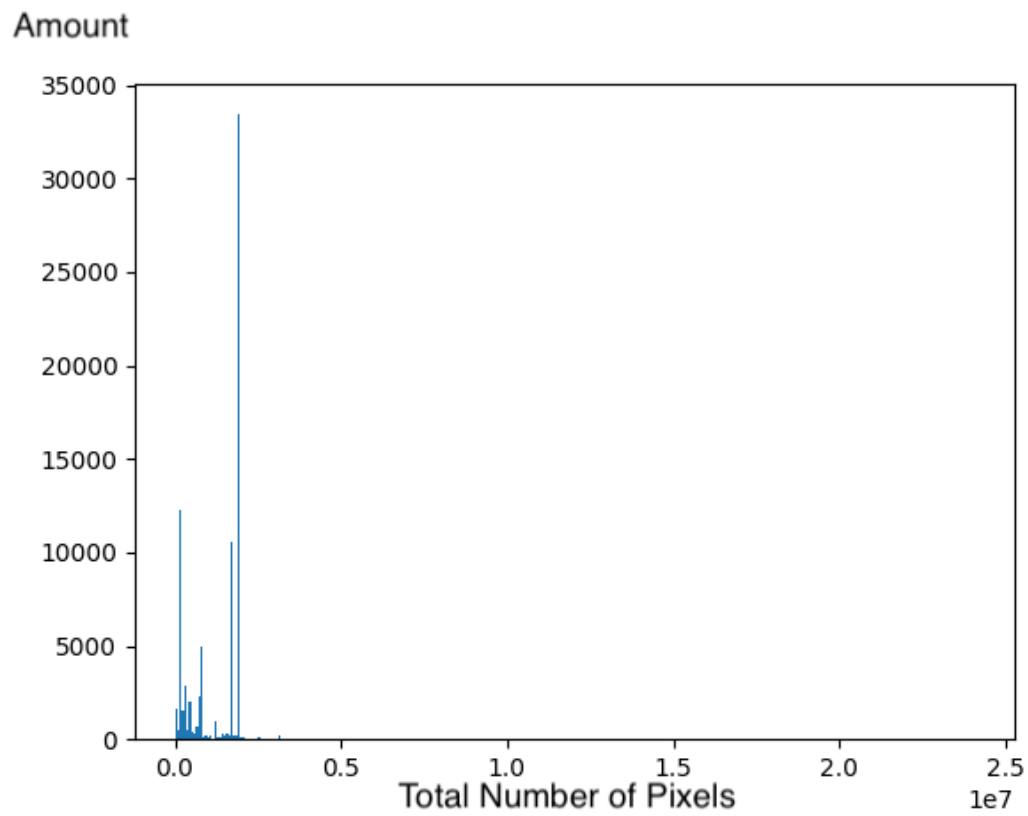


Figure 2.8: Amount of Pixels Histogram

## 3 Features

### 3.1 Dimension Reduction

For the feature extraction we did a dimension reduction with Principal Component Analysis (PCA). For this first attempt we defined

$$x = [shape0, shape1, shape2, max, min, variance, numPixels]$$

as our list  $x$  of features.

The first three features in the list define the image shape, where `shape2` is the channel. The other features are as defined in section 2. Because PCA is affected by scale we scaled the list  $x$  of features in our data with `StandardScaler` (sklearn) before applying PCA.

After standardization we defined a number of principal components and project the seven dimension into two dimensions / principal components. Together these two components contain 77% of information (result: `pca.explained_variance_ratio` [0.39748574 0.37312782]).

We visualize the result of PCA with scatter (matplotlib), which you can see in figure 3.1. The most frequent landmarks can be seen in figure 3.1.

---

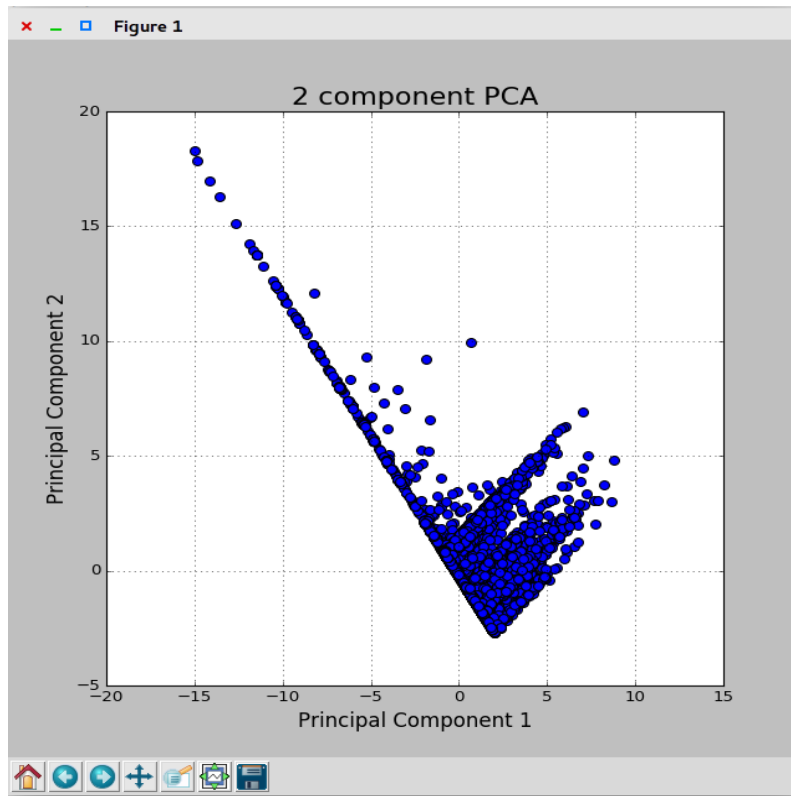


Figure 3.1: PCA result

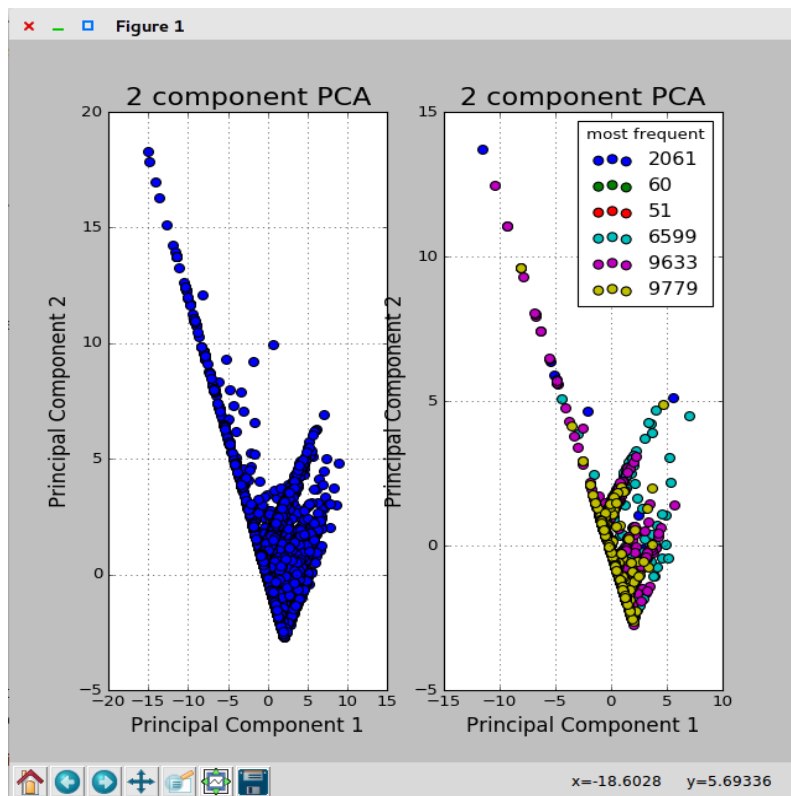


Figure 3.2: Most frequent landmarks from PCA

## 3.2 Clustering

The clustering of the image features is a key task in this challenge. We worked out a first approach at feature extraction with the Principal Component Analysis in the previous section trying to extract the most significant features in order to leave out the insignificant ones. This is a good way of reducing dimensions for both improved simplicity and performance. However, this is also the main task of Assignment 11 and would require too much of next week's work. Our approach is to create a feature map with a deep convolutional neural network and to classify the images with these extracted features.

---





## 4 Conclusion

The dataset is well distributed and well fitting for creating a module that can recognize landmarks in different kinds of images. The images have a various number of shapes and dimensions and are therefore a good basis for a versatile training.

Our analysis showed that a deep learning approach with a convolutional neural network is a fitting choice for this problem due to the amount of data and the difficulty of the task. We will primarily focus on a subset of the provided data for this project in order to keep the calculation time at an appropriate level.

---



# Bibliography

- [1] INC., Kaggle: *Google Landmark Recognition Challenge*. <https://www.kaggle.com/c/landmark-recognition-challenge>. Version: June 2018
- [2] INC., Kaggle: *Google Landmark Recognition Challenge Data*. <https://www.kaggle.com/c/landmark-recognition-challenge/data>. Version: June 2018
- [3] INC., Kaggle: *Kaggle*. <https://www.kaggle.com>. Version: June 2018
-

