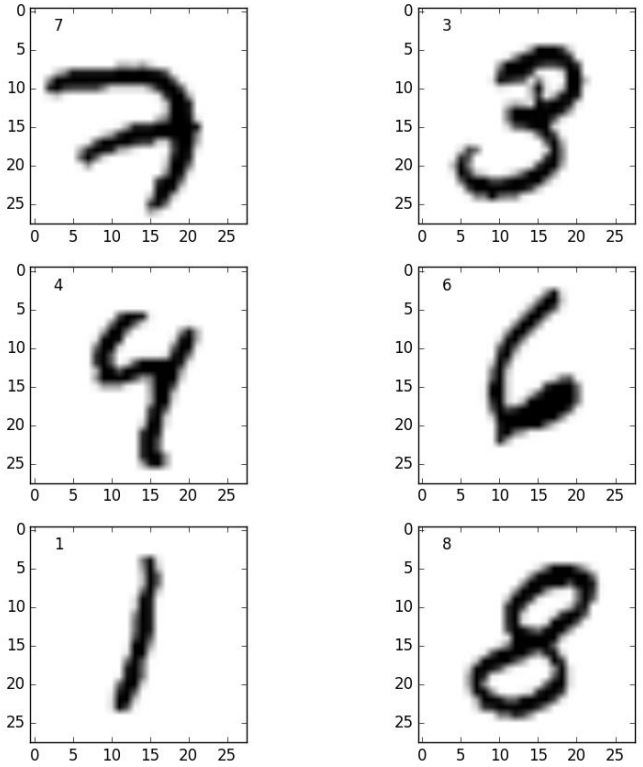
Image Analysis

DIGIT RECOGNITION

MNIST-Dataset

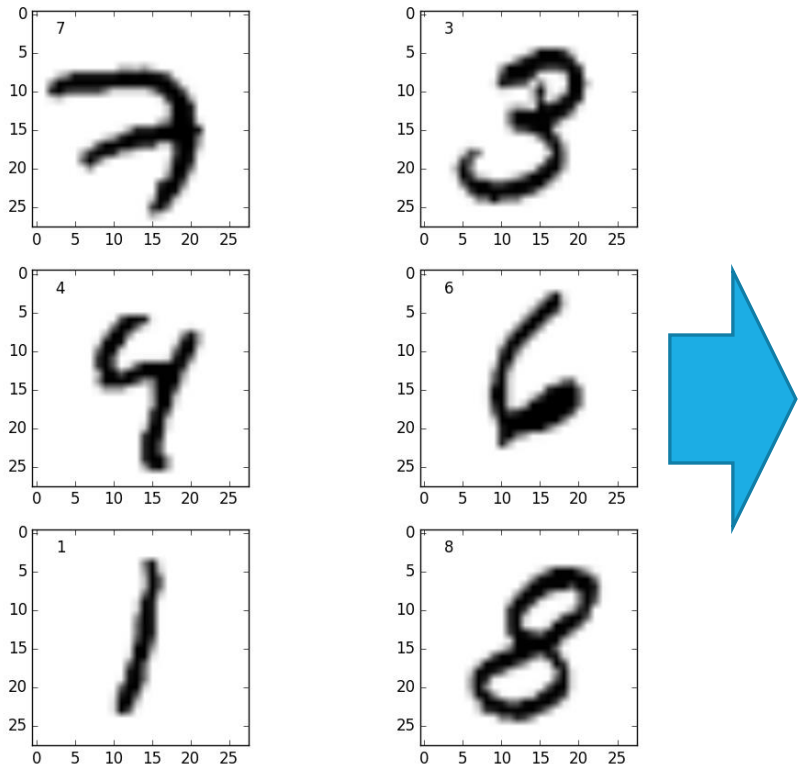


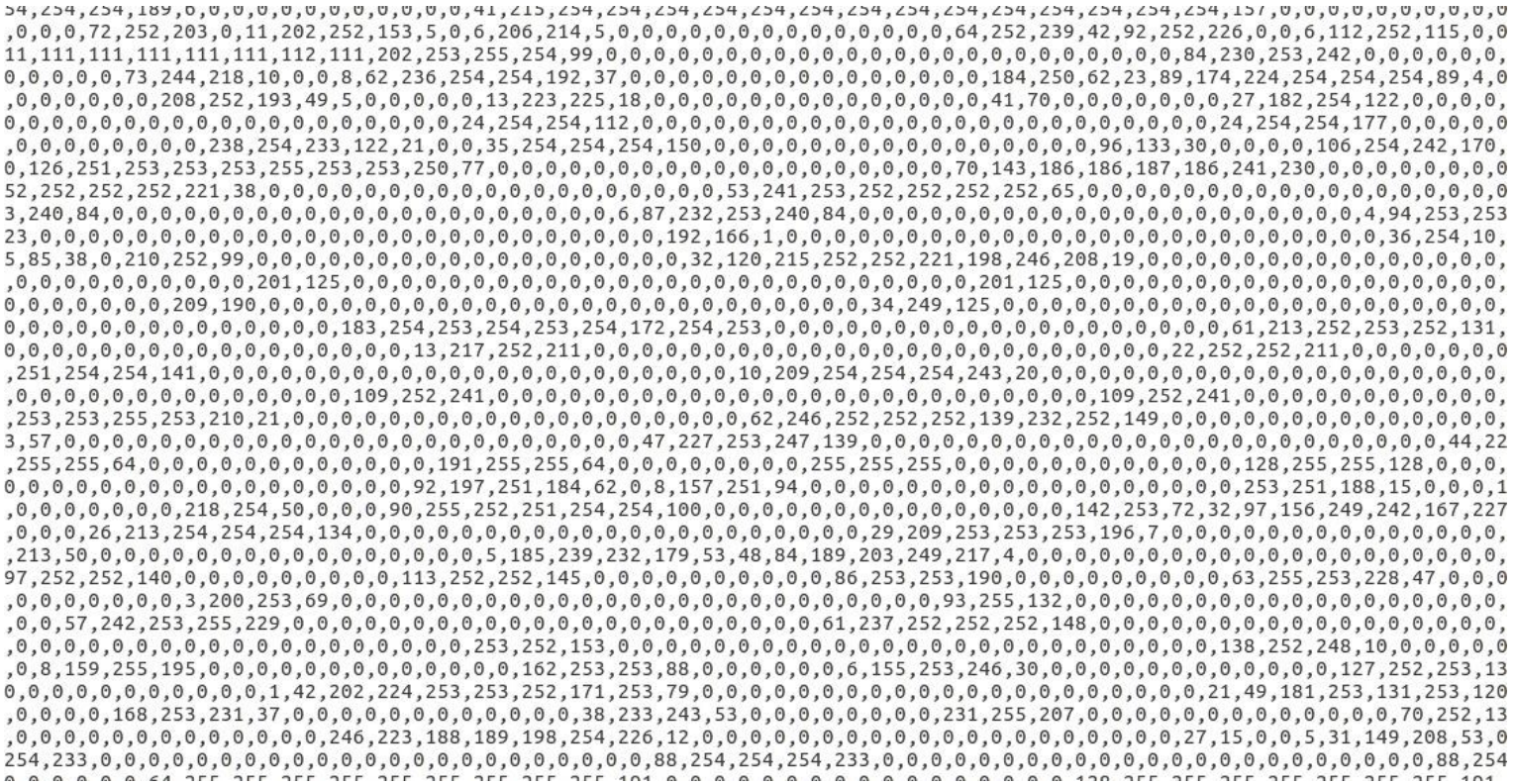
Goal:
The computer recognizes which digit is portrayed on the test-image by comparing it to the training-images.

Input:
Images (28x28=784 pixels) of handwritten digits: centralized and grayscaled
Training set: 60 000 Images
Test set: 10 000 Images

Output:
Prediction of the written number.

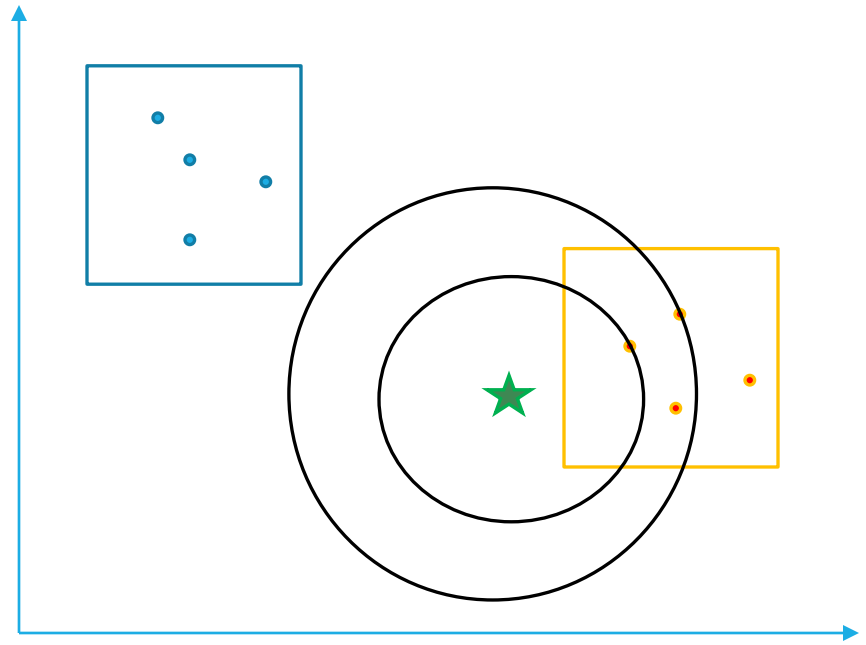
MNIST-Dataset





We see our image-data in this form

K-Nearest Neighbours (KNN)



Which K training vectors are the nearest to our test-vector (euclidian distance)

-> Which k training images are the most similar to our test image.

In this example with 2 dimensions
(our case has 784 dimensions)

Principal Component Analysis (PCA)

The image is a 784 dimensional vector

Solution: **PCA** Reduces dimensionality by eliminating redundancy and emphazising variances.
Not all 784 dimensions are needed in order to recognize a digit

Vectorspace with 784, 2, 20, 100 and 700 dimensions

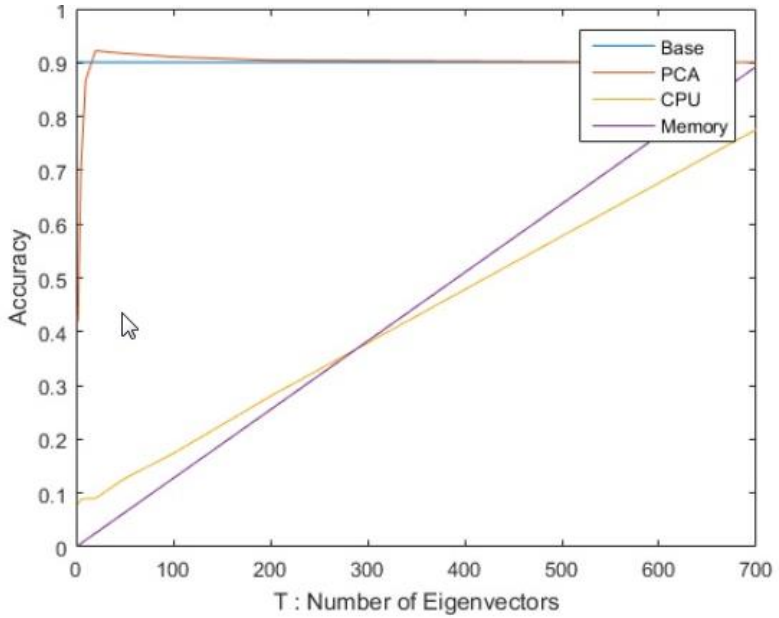
Source: PCA of handwritten digits, Tyler McDonnell

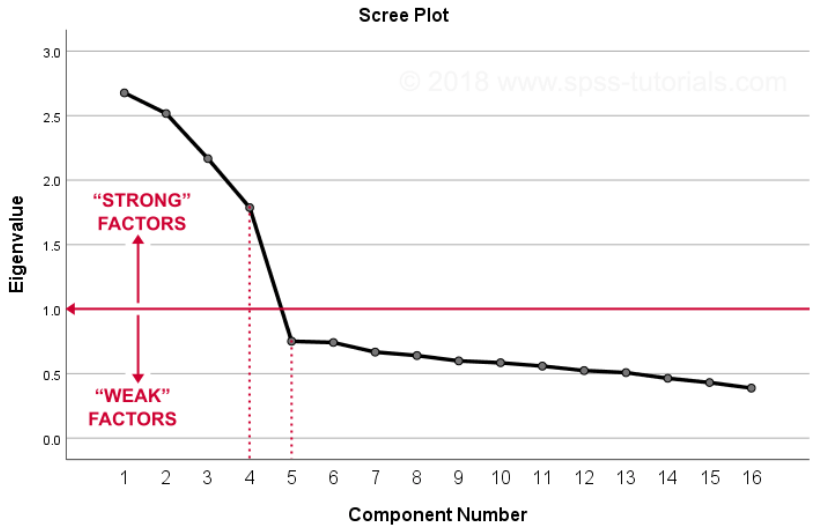
Optimizing the parameters for the PCA

Needed a function to record accuracy rate depending on different numbers of eigenvectors

The maximum of the accuracy is interpreted as the optimal number for **t** dimensions in PCA space.

For optimal **t** finding scree plot and “elbow” scree test could also be used

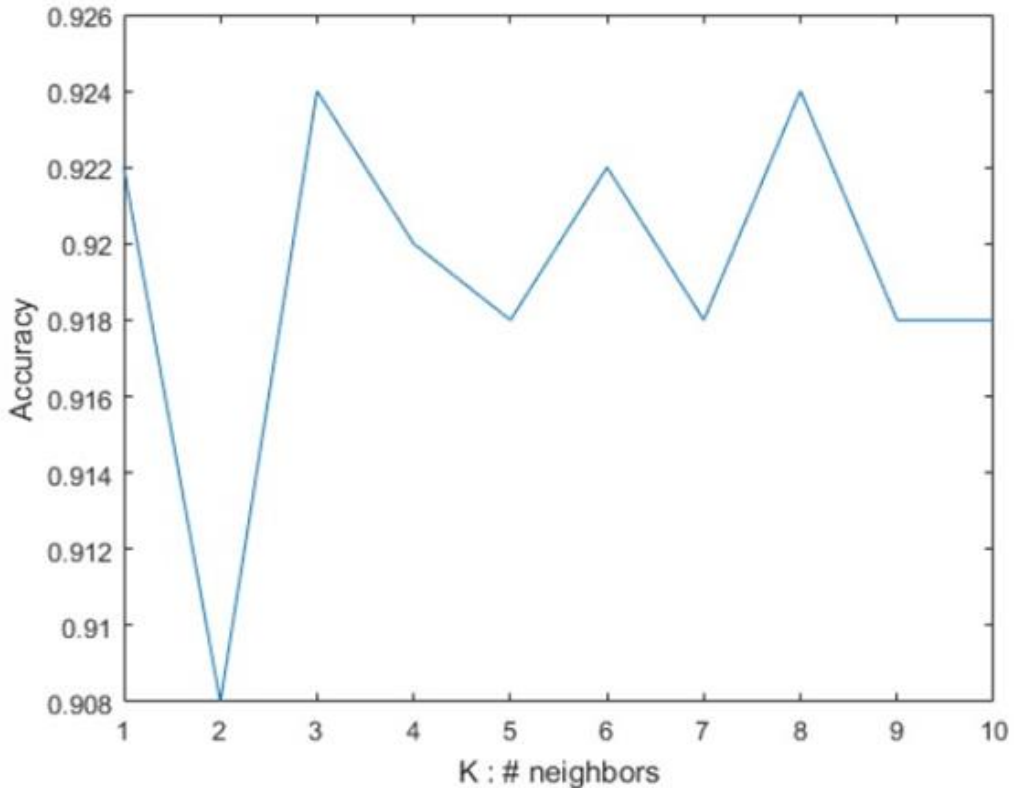




Source: PCA of handwritten digits, Tyler McDonnell

Optimizing the parameters for KNN

The number **k** of neighbours in KNN can also be optimized by plotting different **k** against the accuracy

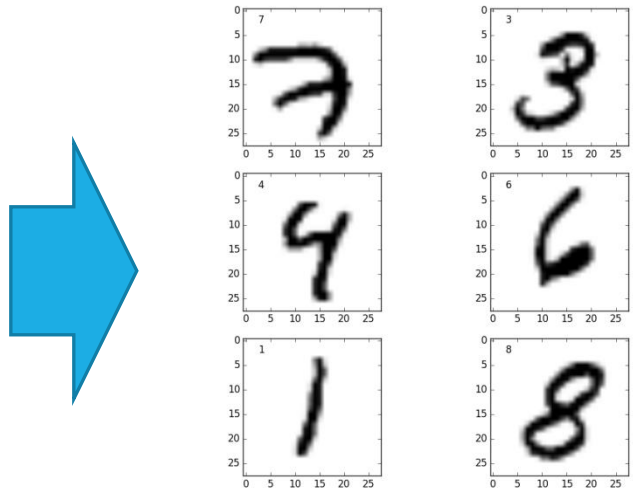


The main idea: to take the one with the best performance but not too small -> outlier sensitive

Visualization

Our imagedata consists only of numbers, so we reverse-transform it to a common image format.

	Pixel1	Pixel…	Pixel784
Image1	0	0	0
Image2	0	0	64
Image3	0	24	0
Image…	9	0	8
Image x	0	0	57

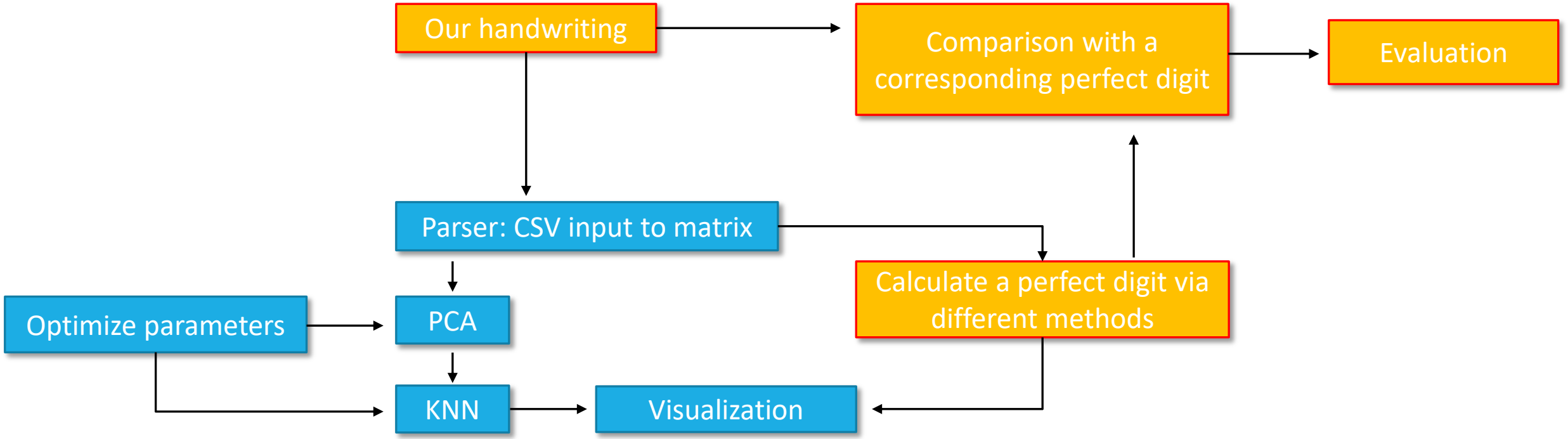


<u>For Example:</u>

View images of falsely recognized digits

Source: Digit Recognition Using Keras, Mukul Agrawal

Overview



Things to try out:

We have time, but we need more!

Milestones	Weeks
Project proposal + programming skill improving	20
KNN in 784 dimensions	21+22
optimal **k** number, success-rate from k plotting function	22+23
Visualization of recognized/falsely recognized digits	24
PCA, optimal number **t** of dimensions	25+26
KNN in PCA-space, success rate from **t** and **k** plotting, optimal **k** for PCA	27
Analysis of our handwriting, search for a perfect digit, comparison of perfect digits found via different methods	28
Writing exercise: write a digit, which you see on the screen and our script will evaluate how well it is written.	29
Preparation to final presentation	30