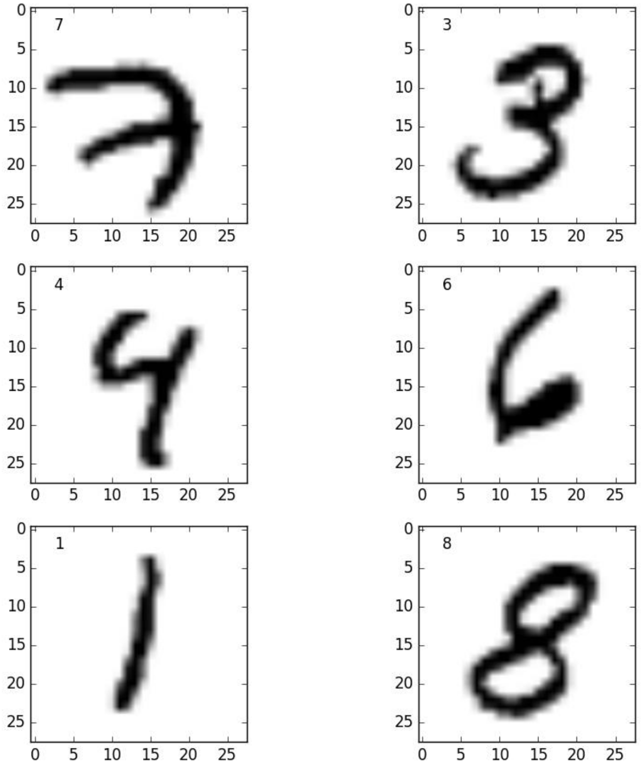
Image Analysis

DIGIT RECOGNITION

MNIST-Dataset

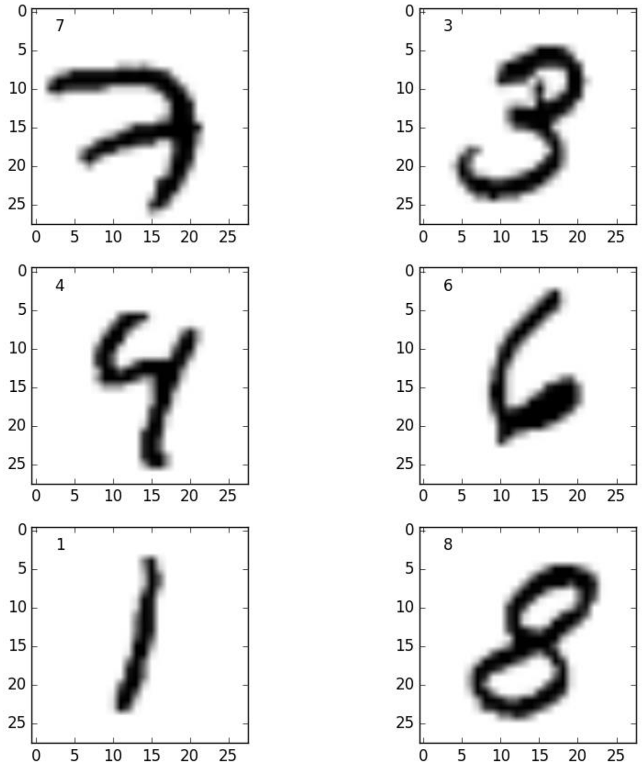


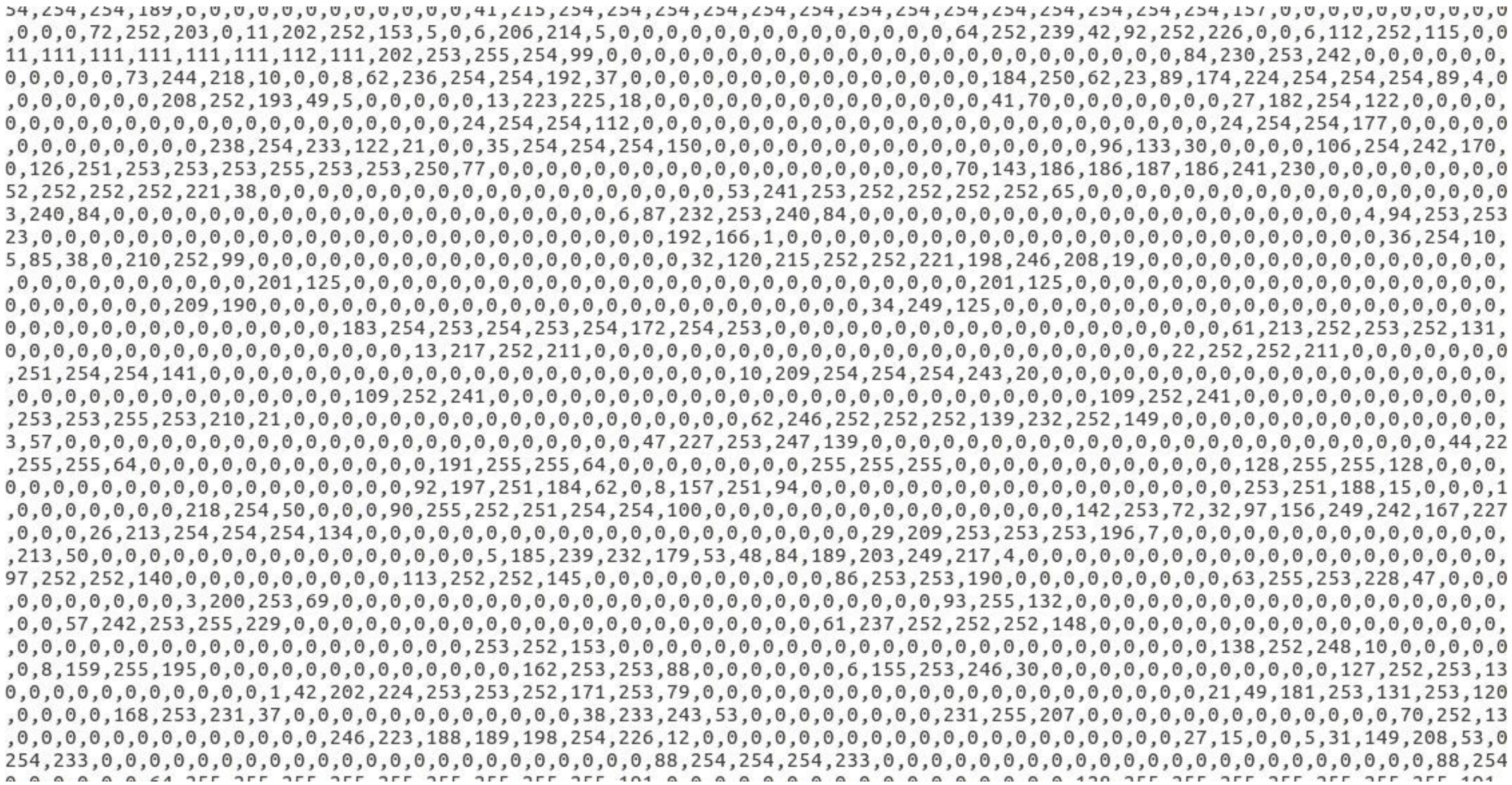
<u>Goal:</u>
The computer recognizes which digit is portrayed on the test-image by comparing it to the training-images.

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

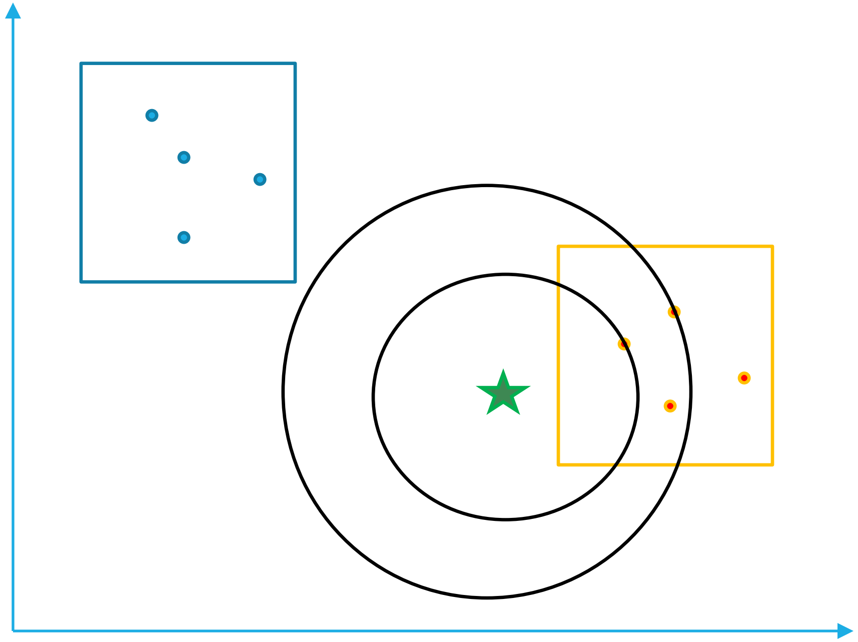




```
54,254,254,189,6,0,0,0,0,0,0,0,0,0,0,0,41,215,254,254,254,254,254,254,254,254,254,254,254,254,254,254,157,0,0,0,0,0,0,0,0,0
,0,0,0,72,252,203,0,11,202,252,153,5,0,6,206,214,5,0,0,0,0,0,0,0,0,0,0,0,0,64,252,239,42,92,252,226,0,0,6,112,252,115,0,0
11,111,111,111,111,111,112,111,202,253,255,254,99,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,84,230,253,242,0,0,0,0,0,0,
0,0,0,0,0,73,244,218,10,0,0,8,62,236,254,254,192,37,0,0,0,0,0,0,0,0,0,0,0,0,0,184,250,62,23,89,174,224,254,254,254,89,4,0
,0,0,0,0,0,0,208,252,193,49,5,0,0,0,0,0,13,223,225,18,0,0,0,0,0,0,0,0,0,0,0,0,0,41,70,0,0,0,0,0,0,0,27,182,254,122,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,254,254,112,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,254,254,177,0,0,0,0,0
,0,0,0,0,0,0,0,0,238,254,233,122,21,0,0,35,254,254,254,150,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,96,133,30,0,0,0,0,106,254,242,170,
0,126,251,253,253,253,255,253,253,250,77,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,70,143,186,186,187,186,241,230,0,0,0,0,0,0,0,0
52,252,252,252,221,38,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,53,241,253,252,252,252,252,65,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3,240,84,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,87,232,253,240,84,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,94,253,253
23,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,192,166,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,36,254,10,
5,85,38,0,210,252,99,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,120,215,252,252,221,198,246,208,19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
,0,0,0,0,0,0,0,0,0,0,201,125,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,201,125,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,209,190,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,34,249,125,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,183,254,253,254,253,254,172,254,253,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,61,213,252,253,252,131,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,217,252,211,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22,252,252,211,0,0,0,0,0,0,0
,251,254,254,141,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10,209,254,254,254,243,20,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,109,252,241,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,109,252,241,0,0,0,0,0,0,0,0,0,0,
,253,253,255,253,210,21,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,62,246,252,252,252,139,232,252,149,0,0,0,0,0,0,0,0,0,0,0,0,0,
3,57,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,47,227,253,247,139,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,44,22
,255,255,64,0,0,0,0,0,0,0,0,0,0,0,0,191,255,255,64,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,128,255,255,128,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,92,197,251,184,62,0,8,157,251,94,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,253,251,188,15,0,0,0,1
,0,0,0,0,0,0,0,218,254,50,0,0,0,90,255,252,251,254,254,100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,142,253,72,32,97,156,249,242,167,227
,0,0,0,26,213,254,254,254,134,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,29,209,253,253,253,196,7,0,0,0,0,0,0,0,0,0,0,0,0,0,
,213,50,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,185,239,232,179,53,48,84,189,203,249,217,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
97,252,252,140,0,0,0,0,0,0,0,0,0,0,113,252,252,145,0,0,0,0,0,0,0,0,0,0,86,253,253,190,0,0,0,0,0,0,0,0,0,63,255,253,228,47,0,0,0
,0,0,0,0,0,0,0,0,3,200,253,69,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,93,255,132,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
,0,0,57,242,253,255,229,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,61,237,252,252,252,148,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,253,252,153,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,138,252,248,10,0,0,0,0,0
,0,8,159,255,195,0,0,0,0,0,0,0,0,0,0,0,0,0,162,253,253,88,0,0,0,0,0,0,6,155,253,246,30,0,0,0,0,0,0,0,0,0,0,0,127,252,253,13
0,0,0,0,0,0,0,0,0,0,0,1,42,202,224,253,253,252,171,253,79,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,21,49,181,253,131,253,120
,0,0,0,0,168,253,231,37,0,0,0,0,0,0,0,0,0,0,0,0,0,38,233,243,53,0,0,0,0,0,0,0,0,0,0,231,255,207,0,0,0,0,0,0,0,0,0,0,0,0,0,70,252,13
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,246,223,188,189,198,254,226,12,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,27,15,0,0,5,31,149,208,53,0
254,233,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,88,254,254,254,233,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,88,254
```

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.

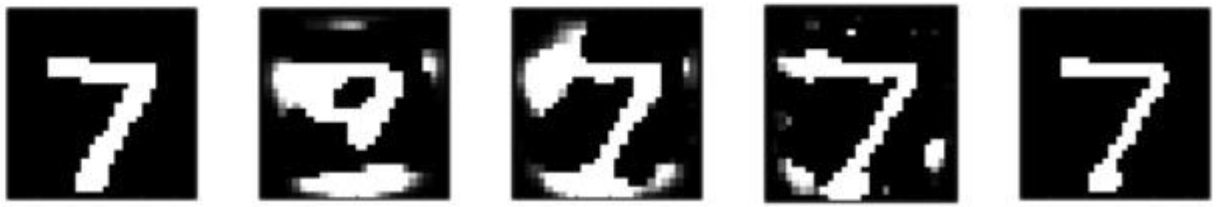
In this example with 2 dimensions
(our case has 784 dimensions)

Principal Component Analysis (PCA)

The image is a 784 dimensional vector (too many dimensions)

Solution: **PCA** Reduces dimensionality by eliminating redundancy and emphazising variances.
May increase the accuracy of the KNN algorithm.

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



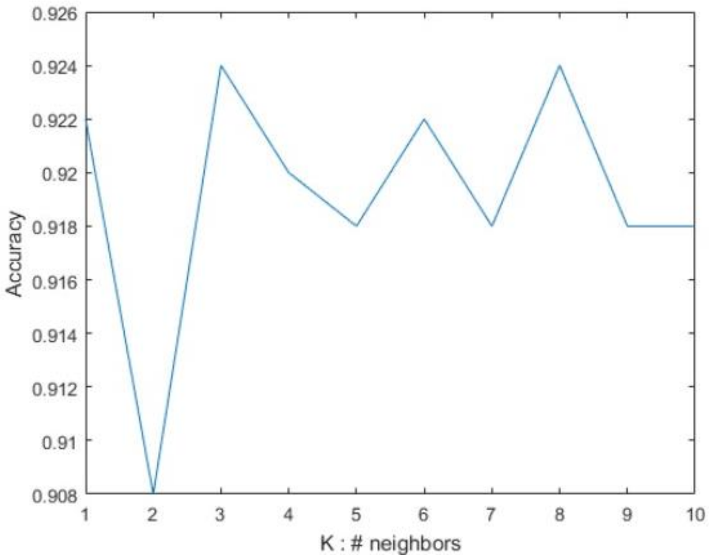
Source: PCA of handwritten digits, Tyler McDonnell

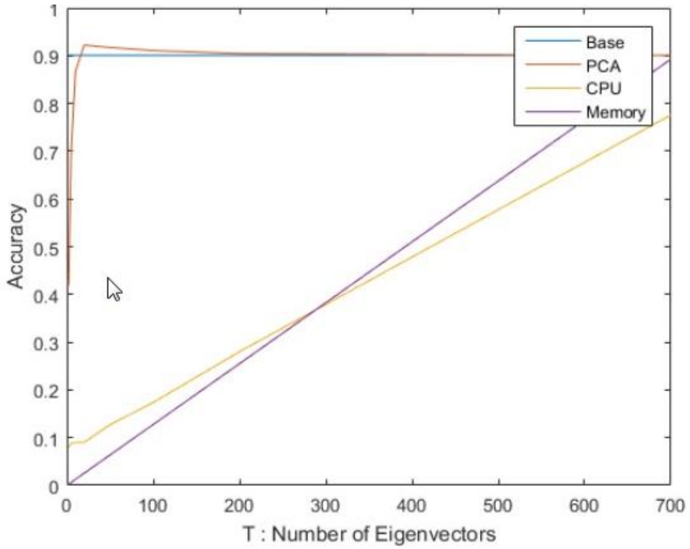
Optimizing parameters for the KNN/PCA

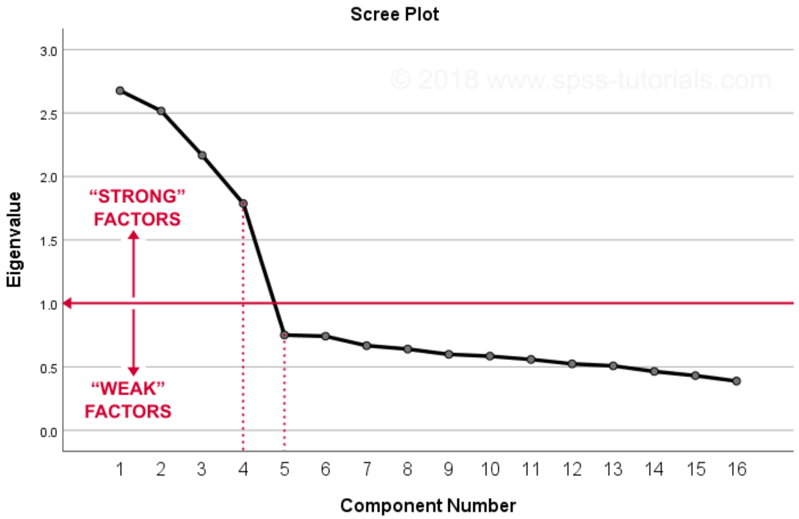
Needed a function to record accuracy rate of correctly recognized digits

The maxima of the accuracy are interpreted as the optimal numbers for **k** neighbours and **t** dimensions in PCA space.

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





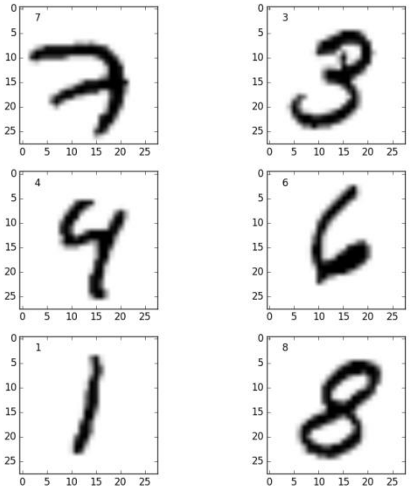


Source: PCA of handwritten digits, Tyler McDonnell

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



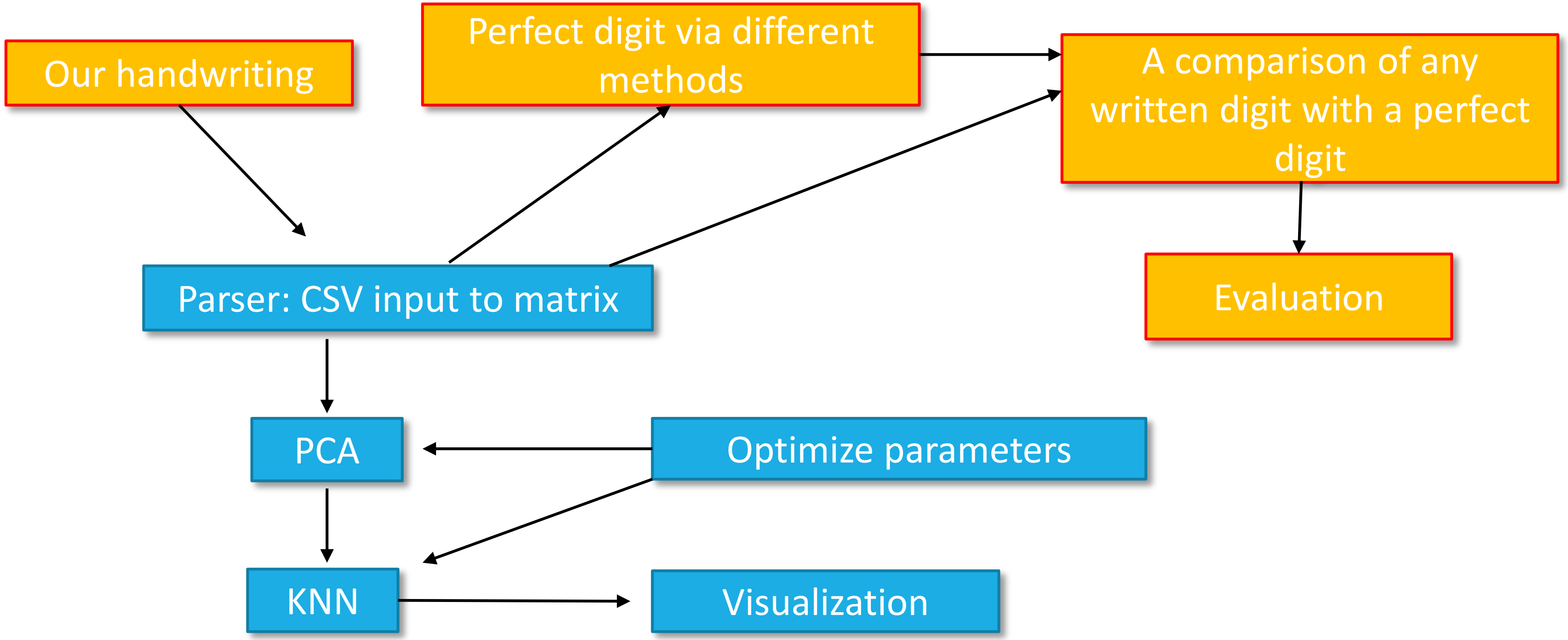
For Example:

View images of falsely recognized digits

Source: Digit Recognition Using Keras, Mukul Agrawal

Overview

Things to try out:



Our handwriting

Perfect digit via different methods

A comparison of any written digit with a perfect digit

Evaluation

Parser: CSV input to matrix

PCA

Optimize parameters

KNN

Visualization

Timeplan

Milestones	**Weeks**
Project proposal + programming skill improving	20
KNN in 784 dimensions	21
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