Number Identification

Team members:

- Urmas Akerman
- Walter Võikar

Task 1. Setting up

Github repository: https://github.com/waltervoikar/number-identification

Task 2. Business understanding

Background: Recognizing digits and numbers in natural scene images is a hard problem for computers to solve as the images can contain distracting features, images can have different contrasts and digits can overlap. Most object detection and classification use cases are on real world images. We want to improve our understanding of neural networks by solving this problem.

Business goals:
The goal of this project is to predict house numbers from pictures with at least 80% accuracy.

Business success criteria:
The model accuracy will be measured on a testing dataset separate from training and validation data.

Inventory of resources:
People: Two students.
Data: Street View House Number dataset from Stanford University that includes pictures for training and testing the model and bounding boxes and number labels for the corresponding pictures.
Hardware: Two laptops.
Software: Pycharm, Matlab

Requirements, assumptions, and constraints:

Schedule: 01.12.2025 12:00 Project requirements report
08.12.2025 12:00 Project information submission deadline
08.12.2025 12:00 Project poster PDF submission deadline
Requirements for acceptable finished work: Model with test accuracy of >80%.

Risks and contingencies: Training the model might be too computationally/memory intensive for laptops. If training is too computationally expensive then training should be done on personal PCs with a dedicated graphics card. If training the model is too memory intensive then training could be done on personal PCs or the training the model could be done with smaller chunks of data.

Terminology:
Digit - any numeral from 0-9
Number - Digit or a group of digits used to count, measure or label something.

Task 3. Data understanding

Outlining data requirements
First of all we needed a dataset with a big enough resolution (bigger than for instance 28x28 MNIST handwritten numbers dataset) and a much more general number styles, so that our trained model could be more general in number detection and classification. This also means we were not looking for datasets where the numbers were cropped (i.e. only for classification). This led us to pick the Street View House Numbers (SVHN) dataset from Kaggle, provided by Stanford University in collaboration with Jessica Li.

Describing data
The data is grouped into a test, train and an additional extra dataset. Each set contains .png image files and is accompanied by a .mat file that contains the name, bounding boxes and labels for each image in the set. Image naming conventions was for example for the first images 1.png, 2.png etc.
Each house number digit in a given picture has a set of values describing it in the following order:
“Left” - bounding box top left starting coordinate on the x-axis(integer),
“Top” - bounding box top left starting coordinate on the y-axis(integer),
“Height” - bounding box height from the starting position in pixels(integer),
“Width” - bounding box width from the starting position in pixels(integer),
“Label” - class corresponding to the digit. There are 10 classes for digits 0-9. Digit ‘1’ corresponds to label 1, digit ‘9’ corresponds to label 9. Digit ‘0’ corresponds to label 10. The label is an integer.
Each picture name in the .mat file has a corresponding set that contains a set containing label and bounding box information for each digit in the picture.
The training dataset consists of 33 402 images which have a total of 73 257 digits.
The testing dataset consists of 13 068 images and 26 032 digits.
The extra dataset has a lot more - with images containing a total of 531 131 digits, of less difficult samples, but we didn’t use this extra training data for training our model, because it seemed unnecessary and outside of our computing capabilities.

Exploring data
All images contained at least one digit, with the maximum being 6 digits in the training dataset (we set this as the maximum digits for our model). The images are of varying dimensions, but most fit in 250x100 pixels and below, with around 10-15% being of higher resolution. In the training dataset the largest image had a resolution of 876x501 pixels and the smallest image had a resolution of 25x12 pixels.

Verifying data quality
We saw that all images were of digits, with some being quite low resolution. Using a simple python script we displayed some images and the bounding boxes for the digits. Most of the images were well labeled. For instance, bounding boxes might not be on center for each and every digit, off by a few pixels in some cases, but this was not a meaningful error. There were some cases where a 3 digit street number had an extra digit to the side of the image, with that not being labeled, but it wasn’t part of the street number in my eyes so this was not worrying - probably even good for model generalisation. For around 1% of images there were some digits not labeled at all (ie 1 digit out of 3 or 1 out of 2 in an image), so some images had to be removed from the dataset for training. But overall the dataset is of good quality.

Task 4. Planning your project

Task	Walter	Urmas
Learn about the use of CNN for object detection and classification	4h	4h
Extract bounding box and label data from .mat file and convert it into numpy array	2h	5h
Preprocessing - converting the pictures to grayscale, resizing images with padding to preserve aspect ratio and scaling bounding boxes accordingly. Scaling image and bounding box data to 0-1 scale for the CNN model.	4h	4h
Training the CNN model	10h	7h
Model optimization	6h	6h
Creating the poster PDF	4h	4h

We will be using Keras for image preprocessing and to build our convolutional neural network.