# Main task:

The task is designing a neural network that learns how to label the housing numbers.

# Pre-processing steps.

The input is a 4D matrix which its last dimension indicated the number of images. We need to change the order of the input dimensions in a way that the number of images is the first and the number of channels is the last dimension. We used the transpose function from the Numpy library to rearrange the input matrix.

Moreover, to prepare the data, we changed the data from color to grayscale by using the inner product of the images with a vector. This will reduce the size of the images and increase the learning time. Also, images are normalized by dividing by 255. For the labes, firstly we change the label 10 to 0 as label ten is indicating the number 0. Secondly, we use a Keras function to make the labels categorical.

# Network architecture:

In this assignment, we harvest the power of the convolutional neural networks to achieve very high accuracy. We do this by preprocessing the data, using ... convolutional layers and one fully connected layer. Also, we use normalization methods to cope with the overfitting problem.

Here is the structure of the model:

## ...Model summary

.

.

type of layers, activation function, regularization

better make a table:

. A depiction of the network structure is provided below.

## Use flowchart

# Why this model:

One can use many different architectures for the model, yet not all of them provide a reasonable solution. In our design, we need to consider several points. First, we need to develop a model which can fit our training data with high accuracy. Second is the issue of overfitting. That is if we do not use proper methods to control overfitting, our test accuracy will be unacceptable even though the training error is minimal. The third issue is the problem of the time and resources that we need to train our model. Using many layers might lead to a higher accuracy but for this assignment, our time and computational power are limited. For example, the model that Goodfellow et al., (2013) designed took six days to get trained. Here are our justifications for selecting the mentioned model considering the issues discussed above.

Studying the Goodfellow et al.,(2013) work gave us an idea about a perfect model structure. They used "eight convolutional hidden layers, one locally connected hidden

layer, and two densely connected hidden layers" [1]. However, due to our computational restrictions, we used ... convolutional layers to detect different patterns from the images. Also, we used a fully connected layer after convolutional layers with ... neurons to achieve a high accuracy.

To control overfitting, we used l2 regularizer with a penalties size of 0.002 ... in the dense layer. We tried l1 as well; however, the results were better with l2. Moreover, we used dropout to ignore ...20 % of the weights before feeding the data to the fully connected layer. Also, we used the max pooling technique to reduce the overfitting and to increase the training speed. We tried several models, and we released that although this technique makes the training process faster, it can reduce the overall accuracy. Therefore, it should be used with caution.

Finally, to control the training time, we convert the images to grayscale to reduce the input size. Each colored image has three channels. By changing them to grayscale, we reduce the number of channels to one and increase network training speed. Max Pooling and dropout techniques also helped to reduce training time. Moreover, we tried to keep the model as simple as possible by limiting the number of dense layers.

To conclude, we tried several different model structures with different parameters. The presented model is the simplest model that can achieve an accuracy above 91%.

# How the model is trained:

After establishing the model, we need to compile the model and define a loss function, an optimizer, and a metric. For the loss function, with the help of the Keras documentation, we use the "categorical cross-entropy" function as we have ten different categories. We select Adam optimization algorithm as it is the efficient, default optimizer. For the metric we used accuracy which is the fraction of the images that are correctly classified.

When fitting the model, we need to decide the number of the epoch and the batch size. By trial and error, we found out that after... epochs this model stops improving so we need to stop after .. epoch. Also, the batch size we used is 100 as using all the data to train the model take a long time.

Results:

Go to <https://www.tensorflow.org/tutorials/keras/basic_classification>