Evaluation of CNN architectures for text detection in historical maps

J. Nathanael Philipp
Maximilian Bryan
jonas_nathanael.philipp@uni-leipzig.de
bryan@informatik.uni-leipzig.de
Universität Leipzig
Leipzig, Germany

In maps and especially historical maps, text can be found in all sizes, fonts, spacings, orientations and curvatures. In addition, maps often have different texts intersecting each other, e.g. a name with a lot of spacing, which is written over an elevation, which again is crossing a name of a river. We evaluate different convolutional neural network (CNN) architectures to find and extract these texts. This is a necessary preprocessing step before OCR can be performed. In order to locate the text we train a neural network to classify whether a given input is text or not. Our focus is on the comparison of different outputs of the CNNs. We compare a simple classification network to a network outputting a pixel mask.

Acquiring enough training data especially for the later method is quite a time consuming task, so we further investigate a method to generate artificial training data. We compare three training scenarios. First training with images from historical maps, which is quite a small dataset. Second adding to the images the artificially generated images and third training just with the artificially generated data.

The maps all have different sizes, which makes it difficult to build a neural network to work with all of them. A more promising approach is the use of sliding windows: The map images are cut into many smaller parts, put through the network and so that then this can be used to generate a pixel wise probability distribution over the whole map. In order to make this more useful and accessible, we visualize the probability distributions and can generate bounding boxes and convert them into PAGE-XML which can be used with OCR systems e.g. Transkribus[1].

Dataset. For the training, we use excerpts from ten historical maps, of which 180 contain no text and 470 excerpts contain text. All excerpts are larger than the input to our neural networks, so during training we randomly choose a section that is then shown to the network. Additionally, we use different augmentation methods, like rotation, shearing or zooming to further add variation during training and expand what the network sees.

For the artificially created training images, we use the 180 excerpts with no text and randomly draw text on them. Font family, font size, font weights, text length, colour, orientation and position are randomly chosen. We then extract the drawn text and create the corresponding masks when needed. The artificially created images are also augmented before given to the network.

CNN. We use densely connected fully convolutional neural networks with attention. This means that all layers in the network are convolutional layers including the layers performing subsampling. Each convolutional layer with the same input size has all previous convolutional layers as input. The network has five such blocks each containing 15 convolutional layers.

The CNN has an input of 64×64. Dropout is done after each subsampling layer. The attention is calculated before each subsampling layer.

The classification is done at the end with a convolutional layer with only two output values and a softmax activation.

This concludes the first architecture predicting text/no text for a given input. The other network architecture is predicting text/no text for every pixel in the input outputting a pixel mask.

The difference between the two architectures is that an additional output for the pixel wise masks is added. For that the attention matrix of each block is taken, concatenated through transpose convolutions to reach the input size. The mask is result of a last convolutional layer outputting a single value per pixel with a sigmoid activation.

Training. For the poster we included six experiments, i.e. three classification and three pixel masks. For each experiment, we trained one network only with excerpts from the maps, one with a fifty-fifty split between excerpts and artificial created images and one with only artificial created images.

The networks each where trained with 25000 examples per epoch and validated on 2500. Early stopping was used on validation loss with ten epochs patience. So the total training time varies between 11 and 26 epochs.

Visualization. For the visualization of the predictions we used a sliding window approached with an offset 75%. For each pixel the predictions are averaged and then overlaid over the map. Meaning the darker a region is the less likely there is text and correspondingly the brighter the region the higher the probability that there is text.

[1] https://transkribus.eu/Transkribus/