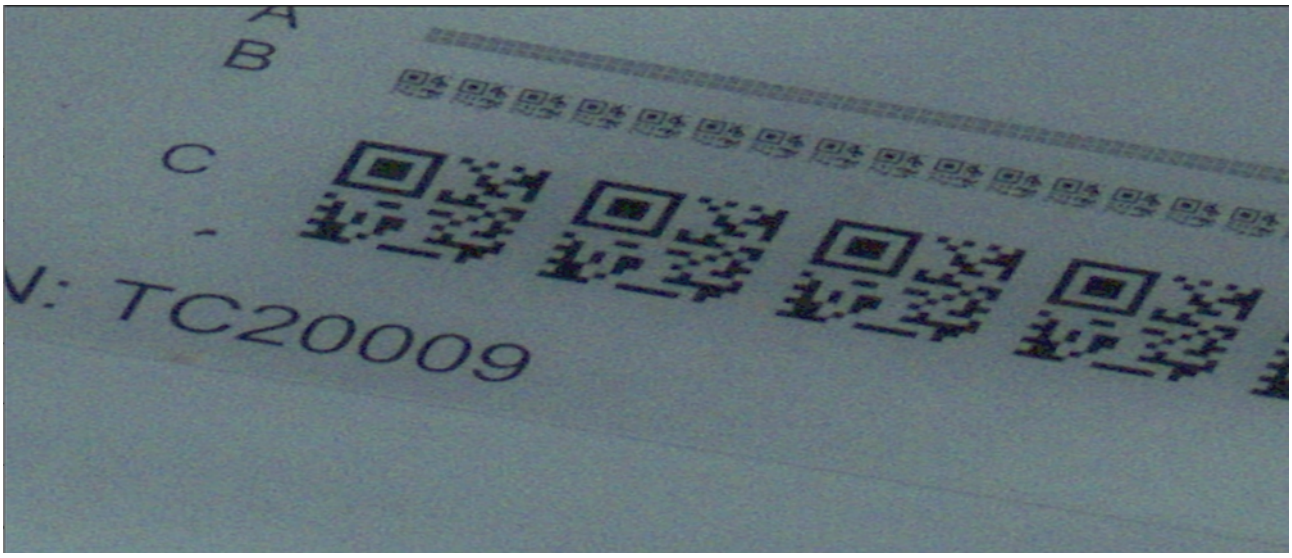
Classic Computer Vision

The image below is taken from a Microscope observing a special calibration specimen. The exercise consists in the identification of the given patterns (the big ones)

Implement the detection of the big patterns and the pixels in Python or C++. Do not decode the content stored in these patterns.

Question:

1) How this can be used?
2) What type of code is this one?
3) How you would implement the recognition and decoding of this code



Deep Learning Exercises

1) Given a CNN classifier we can consider the following building blocks, where we use H x W x C structure:

- CONV2D-K-N is a convolutional of N filters with KxK, padding 0 and stride 1
- POOL-K is a KxK pooling with stride K and padding 0
- FC-N is a fully connected

Tabulate

Layer	Activation Map	Weights	Biases
INPUT	64 x 64 x 3	0	0
CONV-9-32			
POOL-2			
CONV-5-64			
POOL-2			
CONV-5-64			
POOL-2			
FC			

2) U-Net is a common Network model for segmentation of biomedical images based on a encoder-decode architecture with skip connections. One key element of the encoder network is the Transposed Convolution that upscale the image.

Implement in Python (using any Deep Learning framework) an example of upscaling based on the Transposed Convolution:

- Take an image
- Downsample using a Convolution with a fixed kernel
- Upsample using a Transposed with a fixed kernel

Bonus: make an example of trainable upsampling using pairs of images as input: input downsampled, output upsampled.

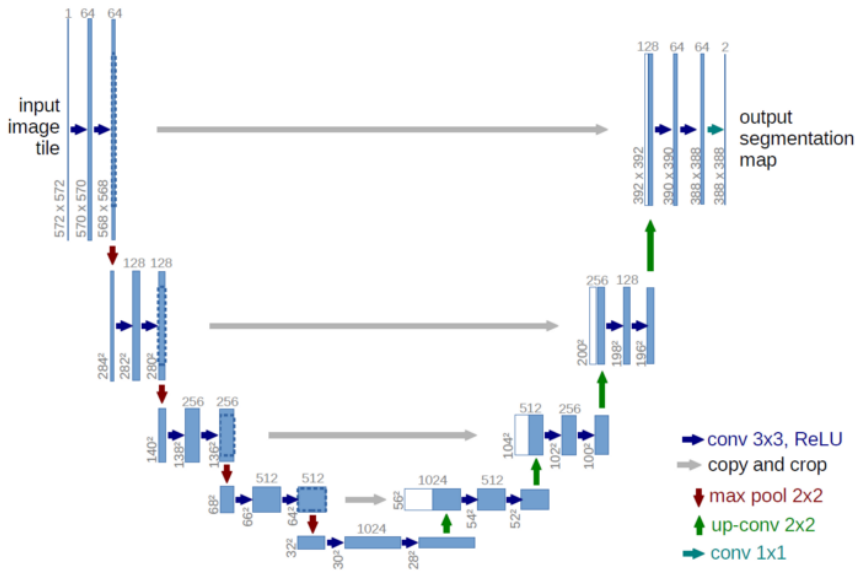


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.