In class, we have already learned that CNNs are able to perform object detection. However, the problem is that they are comparatively very slow and therefore do not allow real time detection. Therefore, we decided to use YOLO.. YOLO offers lower predictive accuracy (e.g. more localization errors), although operates at 45 frames per second and up to 155 frames per second for a speed-optimized version of the model.

How does YOLO work?

Stands for you only look once and it works like the following:
The approach involves a single convolutional neural network trained end to end that takes a picture as input and predicts bounding boxes and class labels for each bounding box directly/simultaneously. (A bounding box describes the rectangle that encloses an object.) Here the input image is divided into S x S grids. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts (The predicted bounding boxes look like the shown picture, the higher the confident score, the fatter the boundary box is drawn.).

For each bounding box, the cell also predicts a class, which works like a normal classification problem. YOLO was trained on the VOC dataset, which contains 20 classes.

Most of these bounding boxes will have a very low confident score, so only the bounding boxes who have a score higher than a specific threshold are kept. This will lead to the final prediction shown in the picture.
So for every picture or frame, the neural network will just run once, which makes the YOLO algorithm very powerful and fast.

Folie 2:
The neural network used in Yolo consists of alternating convolutional layers and pooling layers. The network structure changes partially in the different versions of Yolo. The output is then a tensor, which contains the complete information about the image. For example, if we have a cluster of 7x7 and 2 bounding boxes in each cell, we get an output of 7 x 7 x 30, which means that we have 30 data points in each cell. This is because 5 data points are stored per bounding box: x and y position, width w and height h and the confident score. In addition, the class probabilities are given, in the example 20.