

## Assignment 3: Multi-digit Number Recognition

### Learning objective:

In class we use the MNIST dataset. The MNIST dataset is preprocessed to ensure that each image has the same dimensions and contains only a single number. In this assignment we use the SVHN (Street View House Numbers) dataset and will learn how to deal with images of different dimensions and containing multiple digits.

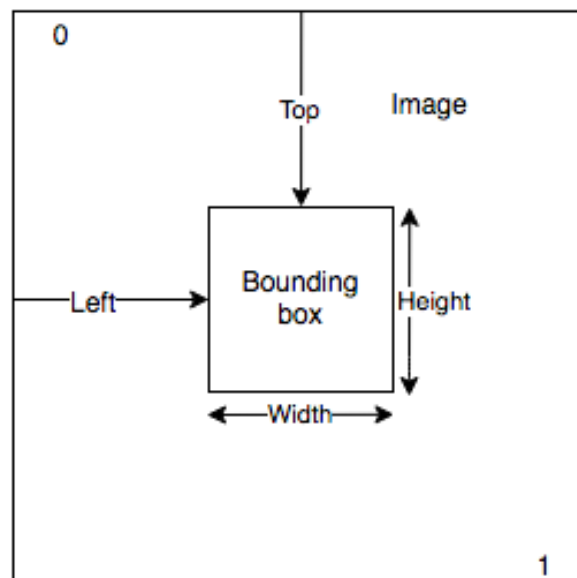
### Data:


Download the data at <http://ballings.co/SVHN.zip>. There should be 13,068 test and 33,402 training images. There is a `digitStruct.json` in both folders. Some Windows users have reported that using the built-in tool in Windows Explorer does not show all the files. WinRAR (<https://www.win-rar.com/start.html?&L=0>) solves this issue.

The images (both training and testing) are stored in PNG format and have variable dimensions. There is also a `digitStruct.json` file. For each image we have the bounding box information for each digit and its label. The labels are specified as follows: Digit '1' has label 1, ... , '9' has label 9 and **'0' has label 10**. To understand the bounding box information try to draw the boxes on some of the images using, for example, `tf.image.draw_bounding_boxes`.

The bounding boxes in this file use the following format:

- Top: number of pixels between the top of the image and the top of the bounding box
- Left: number of pixels between the left side of the image and the left side of the bounding box
- Height: number of pixels between the bottom and top of the bounding box
- Width: number of pixels between the left and right side of the bounding box



33402.png	digitStruct.json
	<pre>{   "boxes": [     {       "width": 7.0,       "top": 10.0, #measured from top of image       "label": 1.0,       "left": 35.0, #measured from left of image       "height": 25.0     },     {       "width": 15.0,       "top": 8.0,       "label": 6.0,       "left": 44.0,       "height": 25.0     },     {       "width": 17.0,       "top": 9.0,       "label": 9.0,       "left": 62.0,       "height": 25.0     }   ],   "filename": "33402.png" }</pre>

### Goal:

Learn a model to identify the multi-digit number in an image with the highest possible accuracy. If the longest number in the dataset is k digits, then this will require k+1 softmax layers. For details see <https://arxiv.org/abs/1312.6082>

### Hints:

- Tutorial on loading and preprocessing images:  
[https://www.tensorflow.org/tutorials/load\\_data/images](https://www.tensorflow.org/tutorials/load_data/images)

- Reading images from disk and resizing (image\_size argument)  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/utils/image\\_dataset\\_from\\_directory](https://www.tensorflow.org/api_docs/python/tf/keras/utils/image_dataset_from_directory)

**Deliverables:**

- Presentation (PDF or PowerPoint) about model performance, things you tried that worked and did not work, details on training, challenges, ...
- TensorFlow Code (.py)
- Presentation should be no longer than 5 min long. Please avoid delays and try out your setup in class before the presentation.