## Domain Background:

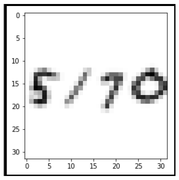
Deep learning, as defined by wikipedia, is a class of machine algorithms that use a cascade of many layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as an input. The algorithms may be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised). Deep learning architectures are often constructed with a greedy layer-by-layer method which helps to disentangle these abstractions and pick out which features are useful for improving performance, and remove redundancy in representation. This technique can be helpful when solving character and digit recognition problems. Specifically, Google uses deep learning image recognition to correctly identify house numbers that are captured by their agents when collecting information for Google Maps. The deep learning network used in this problem is a convolutional neural network (CNN). CNNs are hierarchical neural networks whose convolutional layers alternate with subsampling layers, reminiscent of simple and complex cells in the primary visual cortex [Wiesel and Hubel, 1959], and are trained with standard backpropagation. This is why CNNs apply themselves so easily to visual recognition problems, and how they are so effective at dimensionality reduction while still keep a high level of accuracy and time efficiency.

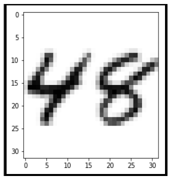
## Problem Statement:

Design and implement a deep learning model that learns to recognize sequences of digits. This will be demonstrated by training the model on a combination of the MNIST and SVHN datasets. The validity of the model will be measured by the accuracy of the digit recognition. As an extra caveat to this project, predicting the bounding box around the digits will be included as an addition goal once digit recognition has been achieved.

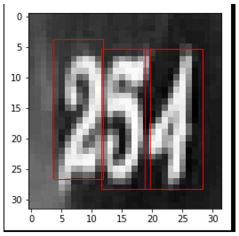
## Datasets and Inputs:

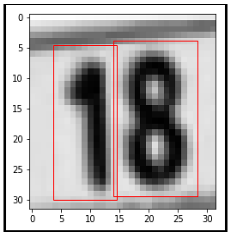
The datasets used to in this project will be: a dataset of random house numbers ranging in digit size from 1 to 5 in length created from the MNIST dataset of handwritten digits and the Street View House Numbers (SVHN) dataset. The MNIST dataset is a collection of 28x28 images of single handwritten digits, which, for the purposes of this project, have been assembled into datasets of random groupings ranging from 1-5 digits in 32x32 images.





The (SVHN) real-world images dataset, originally obtained from house numbers in Google Street View images, consists of 32x32 images of house numbers ranging between 1 and 5 digits along with their corresponding bounding boxes.





## Solution Statement:

Create a CNN capable of predicting digits using convolutional, hidden, pooling, fully connected and output layers. Train the CNN using the combined MNIST and SVHN dataset. Once the digit recognition model is sufficiently accurate, bounding box prediction will be conducted using a CNN with logistic regression using only the SVHN dataset.

## Benchmark Model:

The initial benchmark goals is set at <5% prediction error based on similar CNN models that have classified the MNIST and SVHN datasets. I am predicting that prediction error <10% will be easily achieved. Similar project have achieved as high as 99% digit recognition, but I am doubtful I will be able to reach numbers that high with my computer's performance capabilities. The bounding box benchmark is set at <15% prediction error. This will hopefully be high enough to accurately encompass the digits in most predictions, but if not, the number will be adjusted until a decent working model is designed.

## Evaluation Metric:

Accuracy will be determined using predicted digits corresponding to their ground truth labels. This is accomplished by taking the max probability of each digit prediction from sparse softmax cross entropy, comparing these digits to the true value, and taking the mean of the correct predictions over the whole batch or dataset. The calculation itself is:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$$

The bounding box accuracy will be determined by comparing if the predicted bounding box area is localized within 50% of the area of the ground truth bounding box. This equation is called intersection-over-union (IOU):

$$IOU = \frac{Area\ of\ Overlap\ between\ ground\ truth\ and\ predicted\ bounding\ box\ areas}{Area\ of\ Union\ of\ combined\ ground\ truth\ and\ predictied\ bounding\ box\ areas}$$

**Project Design:**

The MNIST dataset will be preprocessed by reducing the 28x28 images by 4 pixels from the left and right edges of the images in order to make a more compact and visually coherent image when combining them with other random MNIST digits. Once the digit is either left as a single sample or combined with other random MNIST digits (groupings between 2-5) the image will be resized to 32x32 in order to be congruent with the SVHN dataset.

The SVHN dataset only required a small amount of preprocessing by removing any housing numbers larger than 5, and any images with corrupt image data.

After the preprocessing of both datasets a combined training, validation, and testing sets will be created.

The CNN will follow a classic design of multiple convolutional, hidden, and pooling layers followed by fully connected layers and an output layer. As of the writing of this paper I plan on using 3 convolutional layers, 1 fully connected layer, and 1 output layer, but as I mentioned before this setup may have to be tweaked depending on the how well my computer can handle the processing and memory load. This same concern will drive the actual design of the CNN, so it be premature to determine the depths of the actual layers and the batch size that will be fed into the network,

AdaGradOptimizer will more than likely be used as gradient descent optimization algorithm because it works well for image recognition as it updates the learning rate in conjunction with the frequency of the data parameters: frequent parameters get small updates and infrequent parameters get larger updates. This technique works well spare datasets like the SVHN and MNIST, and its the algorithm I have had the most success with in previous image recognition problems.