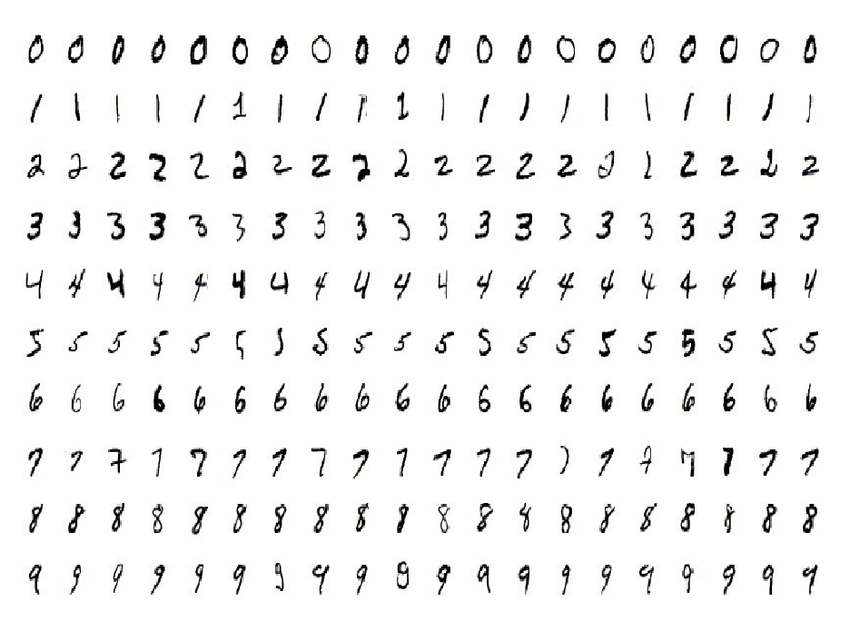
Utilising Machine Learning to Recognize Tables and the Data Within Them

Table Recognition (CompSci NEA)

Kourosh Simpkins

# Problem Analysis

There is a specific problem I aim to solve with my computer science NEA. It is a problem that several students in my Physics class have to deal with. It’s the fact that in every lesson, we spend maybe 30 to 45 minutes copying hand-written tables into excel spreadsheets, that are generally much better at data-analysis than just doing calculations by hand.

As such, lots of time is wasted copying data from one location to the other. Initially I wondered why we couldn’t just use laptops to record our data, until I discovered that it was much easier to simply scribble down numbers on a piece of paper than it was to correctly type numbers on a keyboard, especially in an experiment where large amounts of data is recorded rapidly.

The goal of this program is to save time, in particular for students who have to write on paper, but at the same time use computational statistics to successfully analyse the datasets they have created. If my program is successful, then I will have made the lives of my peers easier.

The problem I am referring to is, of course, getting a computer to recognise the handwritten digits from the famous MNIST dataset. The problem has numerous applications in different industries, including within law enforcement, which can utilise the data to automatically find a number-plate of a speeding vehicle. This is but one example, however.

Ideally, we should be able to pass a handwritten spreadsheet through a computer, making it completely editable via an application such as excel. In reality, we still spend several minutes each day copying tables that have already been written, usually with excruciating accuracy. Is it any wonder that we spend so long at our computers, when we are the ones digitising all of the data that we wish to use? As such, I spend about 30 minutes a week, translating tables from my notebooks to my digital spreadsheets, so that I can take advantage of graph drawing software and more powerful data analysis tools, such as PANDAS.

What I am proposing, is a way for users to take a picture of their handwritten table, and for it to be immediately translated into a CSV table, that can allow for ease of use in data analysis and save time for the end user.

I will achieve these goals in phases. First, I will endeavour to create a working Convolutional Neural Network (CNN) that can identify characters from the MNIST dataset with (relatively) high accuracy. My next target will be to construct another CNN that can determine whether or not there is a table in any given image, and then learn to create the table as an individual image, separate from the rest of the page. (The paper I will be referring to for the design of the DeCNT CNN can be found [here](https://www.researchgate.net/profile/Shoaib_Siddiqui7/publication/329100060_DeCNT_Deep_Deformable_CNN_for_Table_Detection/links/5c01d666a6fdcc1b8d4d0599/DeCNT-Deep-Deformable-CNN-for-Table-Detection.pdf?origin=publication_detail).)

After I have two working CNN’s, I will focus on ensuring that they can both be integrated into the same program. I will do this by treating both CNN’s as modules to be used as part of the wider program. Finally, I will endeavour to ensure that the application can actually present the data in a readable format, as a CSV document. Once it can do this, I will have a solution to the problem at hand, and I will judge it based on the requirements put forward below.

## Requirements for a Successful Design

To validate whether or not my design has worked, I will be marking it against the following criteria:

* The percentage error rate of the Convolutional Neural Network. For example, the lowest percentage error for the MNIST data is currently 0.21%. I will grade my CNN depending on how close or far away its percentage error is.
* There are given test and training sets available for different CNN applications, including for testing a Neural Networks ability to distinguish a table from the rest of a page. As such, I will judge my table Neural Network based on its percentage error when testing against the test sets.