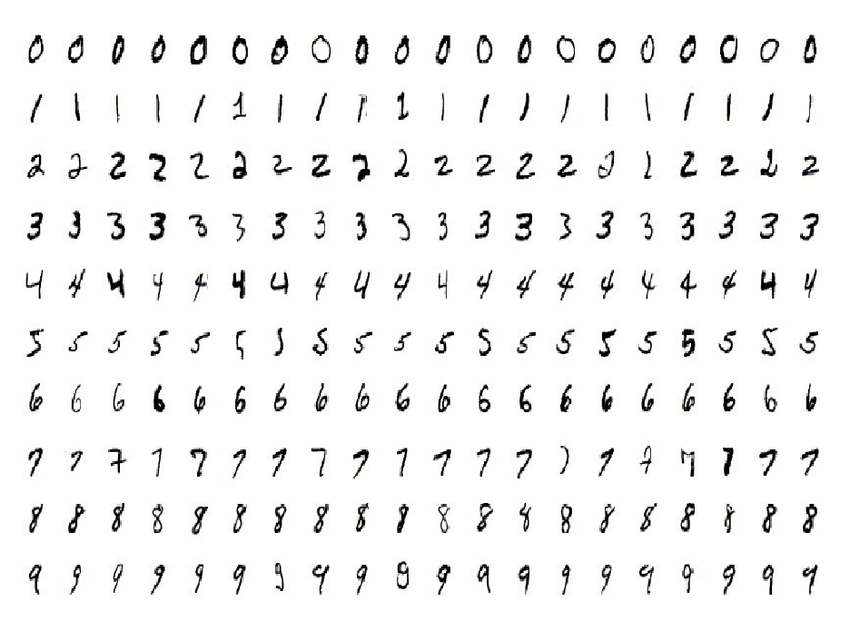
Utilising Machine Learning to Recognize Tables and the Data Within Them

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 by saving them valuable time in the laboratory.

In the case of a working design, I should be able to pass an image through the program, which will detect any tables and split them, recording any data within a cell to a csv file. I shouldn’t need to fiddle around with reading a number on a page and entering it into a spreadsheet, sometimes in the wrong column or wrong row. It won’t make someone miraculously better at data analysis, but it will increase productivity in the laboratory.

One merely needs to take a picture and pass it through a program. It adds simplicity to a workflow, removing the constant need to move between workstations (due to the layout of the laboratory).

## 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.
* For recognition of a table within an image, I will utilise the OpenCV module (v4.1.0.25), which is a library designed around opensource computer vision (where the name OpenCV comes from). I will compile a series of images of tables to test the ability of the overall program to perform the way I expect it to when recognising tables, particularly with hand drawn tables. There are also certain given datasets that can be used to test programs similar to mine so I may utilise a pre-determined table analysis data set, to ensure that I don’t add any biases to what the program can or cannot do.
* To implement the software into an easy to use design, I will endeavour to implement my workings into a JavaScript file that will allow for easy implementation into websites and apps, which will increase the use cases for the program.