**Project Plan – Tutor World MCQ Marker**

**Aim**: to create a piece of software that takes in a series of filled-in question boxes and accurately labels the images with the answer option which has been selected – A, B, C, D, E or None.

**Details**: the MCQ question boxes used in this project will be those used by Tutor World (TW), which have 5 options – A, B, C, D, or E. The labelling will be done in two ways: firstly, using a traditional approach whereby the average pixel value in a region will determine whether a box has been selected; and secondly, a machine learning (ML) approach whereby a convolutional neural network (CNN) will be trained to classify the images. The two approaches can then be compared to identify which is more accurate and which better generalises to different examples. In the first instance, testing and training images will be generated by software created as part of the project to simulate a student’s answers. Further testing can then be done on question boxes filled in by humans to ensure the real-life efficacy of the software.

**Extensions**: in the case that accuracy of either method is shown to be extremely high, the software could then be extended in the following ways:

* Add functionality to allow multiple students’ question boxes to be fed in at one time, storing their name and answers in a .csv file to be exported to Excel for analysis.
* Add functionality to allow a full scanned page of many question boxes to be inputted and question boxes cropped out. This could be done by manually selecting coordinates of the question boxes or using a ML approach. Boxes could then be cropped based on these coordinates and fed into the algorithm.
* Add functionality to allow a variety of other answer box styles (not just the TW style) with the idea of the user telling the software where each answer option is located within a question box or using a ML approach.
* Incorporate software into a mobile app for more general use by a team of markers to mark a full page of questions with the click of a button. Marker would need to input name of student, choose which answer sheet is being marked and take a picture of it. The phone would then automatically mark and save the answers locally as a .csv file to be sent to a laptop at the end for compiling and analysis.

**Stage 1: Generate Data**

This stage is to generate the data used for testing and training purposes, rather than gather real-life data, as the quantity and processing needed would take too long to achieve. The aim is to take an image of a blank question box and use a Python library (Pillow or Scikit-Image) to draw lines in the answer options, simulating an actual response. Generate equal amounts of A, B, C, D, E, None and combinations of 2 letters. Use small random variations when drawing the lines, including line thickness, direction and colour of the line, and rotate the final image box by a small amount to simulate the variations that inevitably happen when scanning the real-life question sheets. Store the image filenames and labels in a Pandas DataFrame and export as a .csv file.

**Stage 2a: Traditional Approach**

This stage is to construct an algorithm that takes as input a series of cropped question boxes which have had answer options already selected – i.e. the data generated at stage 1. The algorithm computes the mean pixel value in the regions of the image corresponding to each multiple-choice answer option and labels the image with which options it believes have been selected. Data is stored in a Pandas DataFrame to be compared later to the true labels.

**Stage 2b: ML Approach**

This stage requires a training set different to the previous set of images, so extra data from stage 1 needs to be generated. The training data is then pre-processed to be fed into a network and trains a CNN formed using Keras whose output layer is a binary classifier in each of the 5 possible classes. Once trained, the network is tested on the same test set as stage 2a and data is stored in a Pandas DataFrame to be compared later to the true labels.

**Stage 3: Testing and Comparison of Approaches**

Compare accuracy of the two methods above and retest both methods on real-life scanned student data with “harder” examples. If accuracy is above a set threshold (e.g. 99.5%), continue onto extensions listed at the start of the project.