# Weekly Progress Update

Weekly Presentation

Joardan Wibisana

21504745@student.curtin.edu.au

IAMAI sustAInable

**School of EECMS**
**Curtin University**

26-04-2024

Curtin University

# Project Goal:

To undertake this task you will need to learn and demonstrate understanding in the following areas (among other things):

- Be able to utilise the MNIST dataset as your main training dataset.
- Become acquainted with software libraries required for the task in C++ including but not limited to OpenCV (for image processing) and Eigen (for linear algebra).
- Develop algorithms to segment individual digits from your input photos and post-process them into a standardised format using OpenCV.
- Understand and develop from scratch (i.e., no use of specialist libraries) at least one of the three following classifiers: Decision Tree, Support Vector Machine (SVM), Neural Network.
- Understand the basic principles behind evaluating classification accuracy in machine learning tasks (e.g., recall vs precision, ROC curves, F-scores etc).
- Develop your app using modern Object Oriented class based design principles including encapsulation, inheritance, composition, polymorphism, and at least two different design patterns e.g., factory classes, delegate classes, etc.
- Use the CMake build system to manage the compilation, linking, and packaging of your app in two different OS environments (Windows and Linux).

This will be a very challenging project for you to undertake and I hope you can get started on it soon. You will likely spend a lot of time initially setting up your development environments in Windows and Linux. I suggest you become familiar (if you aren't already) with VirtualBox – virtualisation software which will allow you to more seamlessly develop across two OS's at the same time. You should definitely proceed initially with setting things up to build and deploy a simple HelloWorld style app using C++ and CMake across both OS platforms. From there, iterate your app to start including the necessary libraries and functionality. You will encounter difficulties with library version compatibilities and probably a lot of hard to solve linker issues so make as much time as possible to work on this.

As for the actual development task, learning C++ shouldn't be too difficult, but understanding the utility and importance of Object Oriented design when designing and building reasonably sized apps is key to being able to undertake a task of this size without becoming lost in the source code. You will likely also benefit from employing an IDE though that's up to you.

Please concentrate on the aspects of image processing using OpenCV, and encoding features as vectors for processing using simple linear algebra (LA). Neural nets and SVMs in particular are very LA friendly algorithms. Classification Trees are easier to understand and design using an analogous data structure that allows for efficient routing through nodes (memory locations). Exploring NNs and SVMs will allow you to practice your applied mathematics, going the CT route will allow you to explore designing and building a complex data structure in C++ so benefits either way – I don't mind what you choose to do.

Figure: Project Goal Overview

Curtin University

# What I Did This Week:

- Made a convolutional neural network with 98% accuracy based on predicted results matching labels over the size of the evaluation set
- Developed a Deep Q-Network (DQN) for Connect 4

# Other Things I Did Throughout:

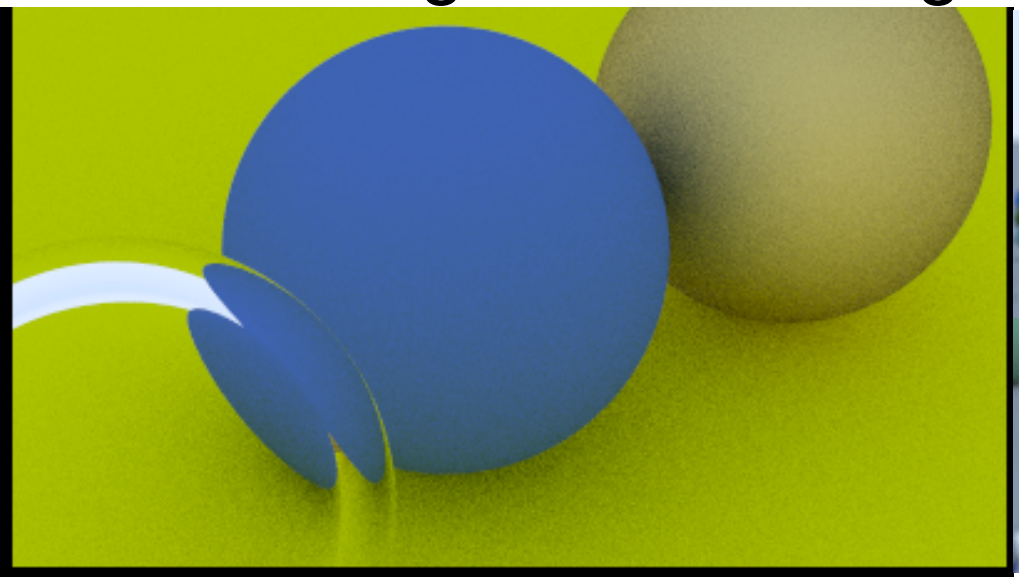

Figure: raytrace1

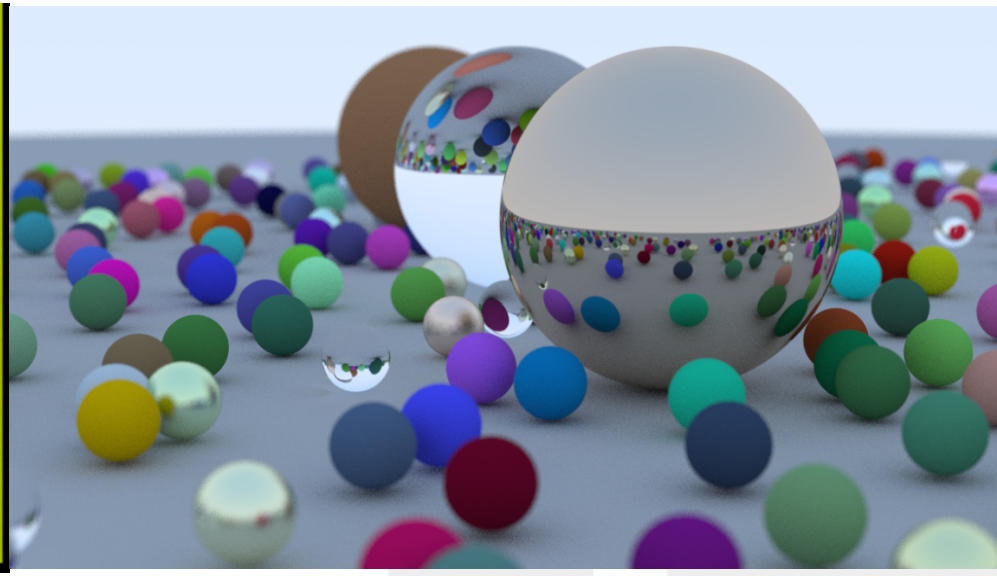

Figure: raytrace2

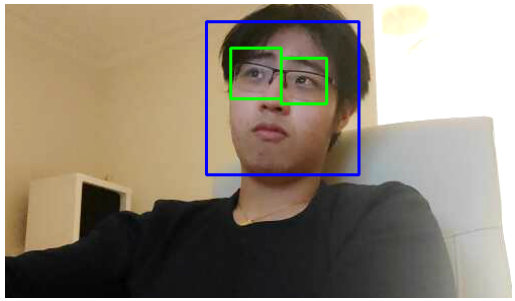

Figure: Test OpenCV face detection

Curtin University

# Results of CNN:

```
.......
.......
.......
.X.....
XOO....

Reward: [0.0, -0.0], Done: False

Player 1 moves in column 3
.......
.......
.......
.......
.X.....
XOOX...

Reward: [-0.2, 0.2], Done: False

Player 2 moves in column 2
.......
.......
.......
.......
.XO....
XOOX...

Reward: [0.2, -0.2], Done: False

Player 1 moves in column 2
.......
.......
.......
..X....
.XO....
XOOX...

Reward: [1.0000000000000002, -1.0000000000000002], Done: False
```

Curtin University

# Results of CNN:

```
Epoch 1 - Test Accuracy: 0.9752
Epoch 2 - Test Accuracy: 0.9778
Epoch 3 - Test Accuracy: 0.9797
Epoch 4 - Test Accuracy: 0.9807
Epoch 5 - Test Accuracy: 0.9803
```

Figure: Result

# Results of CNN:

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

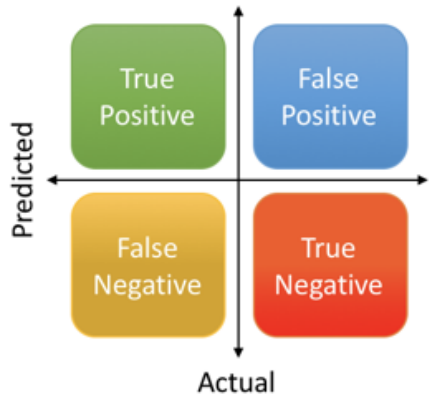


Figure: Project Goal Overview

# Thank you!

## Contact:

21504745@student.curtin.edu.au

Curtin University