

# **IS53055A: Data and Machine Learning for Creative Practice**

---

Final Project Report - ML Flag Identifier

**Manpreet Bance**

January 2020

## Introduction

1. The concept for the idea was conceived when thinking about different potential applications for Machine Learning and the way that data can be classified and an output can be produced based on the input and so I decided to think of an application which would be an enjoyable and creative way of achieving this. I was inspired by the work that Nick Bourdakos had done after coming across a link to his work on Twitter, this was Real-Time Object Detection using Image Classification which I thought I would be able to take parts of and play with the idea. As a result of this, I came up with the idea of Real-Time detection of flags whereby using the web camera, I would be able to identify which country a flag belongs to. I envision that this could be used in an educational environment to perhaps teach people about flags or as a tool that educators can use as a quiz tool.

## Questions

2. As previously mentioned, I was inspired by the work of Nick Bourdakos and I came across his work via Twitter originally viewing Andreas Refsgaard which I was made aware of through a post on learn.gold (<https://twitter.com/AndreasRef/status/757328087289884672>). I found this very interesting as well as a fun concept which I wanted to pursue.
3. After researching the methods that were used to implement the application that Nick had produced, I came across IBM Cloud Annotations to which he is a key developer in as well as him providing a tutorial to the way he went about creating this, I followed this tutorial only to find that it didn't train the data that I was inputting very well which were images downloaded of flags of the world. I then did some further research and looking at Nick's Twitter feed I saw a post from Daniel Shiffman who integrated p5.js to produce a similar application. Following on from this, I went on YouTube to find a tutorial from Daniel Shiffman who used ml5.js to use Machine Learning, more specifically, Image Classification to produce an output from the input of his web camera. After carrying out this further research, I started again on developing my application using ml5.js as well as p5.js which took an input from the web camera and classified every current frame against a model. I trained a model myself using a dataset I compiled containing images of flags which I had labelled and classified, this was then exported and the model was imported and used in the code I had gotten inspiration from, from the tutorial I had followed.
4. Issues I faced at the end of the development process were that as a result of a poorly trained model, some of the flags that I was testing with were unable to be recognised as they did not have enough training data or were very similar to other flags - this is evident through the demonstration video as flags such as the Scandinavian flags where the designs are very similar, the output stating the country was incorrect on several occasions but this could be easily rectified if I had more time by training more data to improve the model so they can be differentiated properly by class.
5. Code I used was inspired from the ml5.js, p5.js which was found in the documentation of both these websites. The dataset that I used to train the model was created solely by myself using the Teachable Machine created by the Google AI team and this was exported, as mentioned before, and hardcoded where it was imported into the code to carry out its function of identifying flags.
6. The application has already been compiled and does not require re-compilation, however, to run the application, the index.html file should be opened (preferably in Google Chrome or Firefox) where access to the web camera will be requested and should be allowed for the execution of the application to work correctly and the input of the web camera will detect what is being shown and the most similar frame will be compared and a string output displayed at the bottom of the canvas will read what country the flag belongs to.
7. The design of the application is very simple as I decided that I would like the functionality to take precedence over the appearance.

## References

- Andreas Refsgaard. Twitter - Using mfc's and wekinator to teach a program when to laugh while watching sitcoms. @SchoolOfMaaa @RebeccaFiebrink. 24 Jul 2016. Accessed on 29 Dec 2019 on <https://twitter.com/AndreasRef/status/757328087289884672>.
- ml5.js. Getting Started. Accessed on 29 Dec 2019 on <https://learn.ml5js.org/docs/>
- p5.js. Getting Started. Accessed 29 Dec 2019 on <https://p5js.org/get-started/>
- Nick Bourdakos. TensorFlow.js Real-Time Object Detection in 10 Lines of Code. 7 December 2018. Accessed on 29 Dec 2019 on <https://hackernoon.com/tensorflow-js-real-time-object-detection-in-10-lines-of-code-baf15dfb95b2>