Final Capstone Submission – Report

Since the beginning of the bootcamp, I did my best to explore the world of Data Science. One area that particularly caught my attention was computer vision. Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do. The more I learned about computer vision, the more I become fascinated about its applications. I knew I wanted to do an entry level computer vision project for my capstone. This led me to research whether **we can use convolutional neural networks to classify human activity through videos?**

Graphical user interface

Description automatically generated with medium confidence

I found the [dataset](https://www.crcv.ucf.edu/data/UCF101.php), UCF 101, from the University of Central Florida. The dataset was developed in the University’s Center for Research in Computer Vision. It contains 13320 videos from 101 action categories. All the videos are in avi format and ranges from 2-6 seconds. Once you extract the dataset, all the videos are stored in their respective action category folder. The videos, however, are not split by train and test splits. This was a good dataset for my capstone project as it built for learning computer vision. The first published project studying the dataset provided an accuracy of 43.90%. I soon realized trying to classify 101 action categories is extraordinarily complex. Reducing the dataset to 10 classes will help lower model complexity and computational load.

Diagram, timeline

Description automatically generated

The bulk of my time was spent preprocessing the data. I took the approach to break the videos into frames. The infographic above outlines the general steps taken in preprocessing the data. I want to batch my data using an ImageDataGenerator so I need to ensure all the frames are placed in the respective action category folder. In addition, the frames must be separated in train, validation, and test sets.

Table

Description automatically generated

For modelling, I chose to use a 2D convolutional neural network to try to classify action categories found in the videos. I experimented with many neural network architectures that I built. Most of the results had validation accuracy hovering around 50%. One hypothesis for the low validation accuracy is perhaps there is one or two classes that the model severely fails to classify correctly. This can pull the overall accuracy down.

I came to realized that my low validation accuracy was decent comparatively. If we were to randomly guess a class, there will be a 1 in 10 chance of being correct and the original dataset had an initial accuracy of [43.9%.](https://www.crcv.ucf.edu/data/UCF101.php) All in all, my best model performed better than randomly guessing and outperforms the initial accuracy of the original dataset.

Chart

Description automatically generated

My final model evaluation was chosen based on the model that had the best validation accuracy. Naïve Model3 had a validation accuracy of 54%. The true accuracy of the model can be found looking at the diagonal of the matrix. The model correctly identifies Lunges, HorseRiding, HorseRace very well. We can also see that BrushingTeeth and Nunchucks are negatively impacting the overall accuracy. The classes in which BrushingTeeth and Nunchucks are mistaken for is very odd and have no logical correlation (human-wise). My initial hypothesis on why the validation accuracy is correct. There are two classes that is critically lowering the overall accuracy.

I plan to continue refining this project even after the Bootcamp. The most immediate action is identifying why some classes are being severely misclassified. I really want to dive deeper on that exploration. Next, I want to experiment with Transfer Learning Models: VGG-16, ResNet50, Inceptionv3, and EfficientNet. I hope that using Transfer Learning Models may improve the validation accuracy. Another method I want to test out is using Recurrent Neural Networks (RNN). Since we decided to break the videos into frames, RNN should shine since the videos follows a set sequence.

I also want to upload this project to Git Pages. Its important for an aspiring data scientist to create an online presence. Creating an online portfolio is a step in the right direction. Finally, I want to deploy the model. I want to be able to upload a video to an API and return a classification result. Currently, I am thinking on using Flask or Streamlit for the framework.

The inspiration of the capstone project was to teach myself computer vision. I developed a foundational understanding on neural networks and image classification. Neural Networks are black box models so I can never truly explain how it works. However, I want to continue learning about computer vision. This capstone project was an excellent steppingstone.