Name: Luke Wilson

Module Code: CS3AM

Assignment report Title: Image Classification Using Machine Learning on the Caltech 101 Dataset

Coursework Date (when the work completed): TODO

Actual hrs spent for the assignment: TODO

Convener Names: Dr. Muhammad Shahzad, Dr. Ferran Espuny Pujol

**Abstract**

Abstract goes here

**Background and Problem to be Addressed**

For this assignment, I have chosen to train 2 machine learning models on the Caltech-101 database. This is a dataset consisting of 101 different image classes, most of which are animals, instruments or other common objects. I will be comparing the differences of two machine learning models. The two models I will be training are a Convolutional Neural Network and a Support Vector Machine for classification

**Exploratory Data Analysis (dataset description and visualisation, support with figures)**

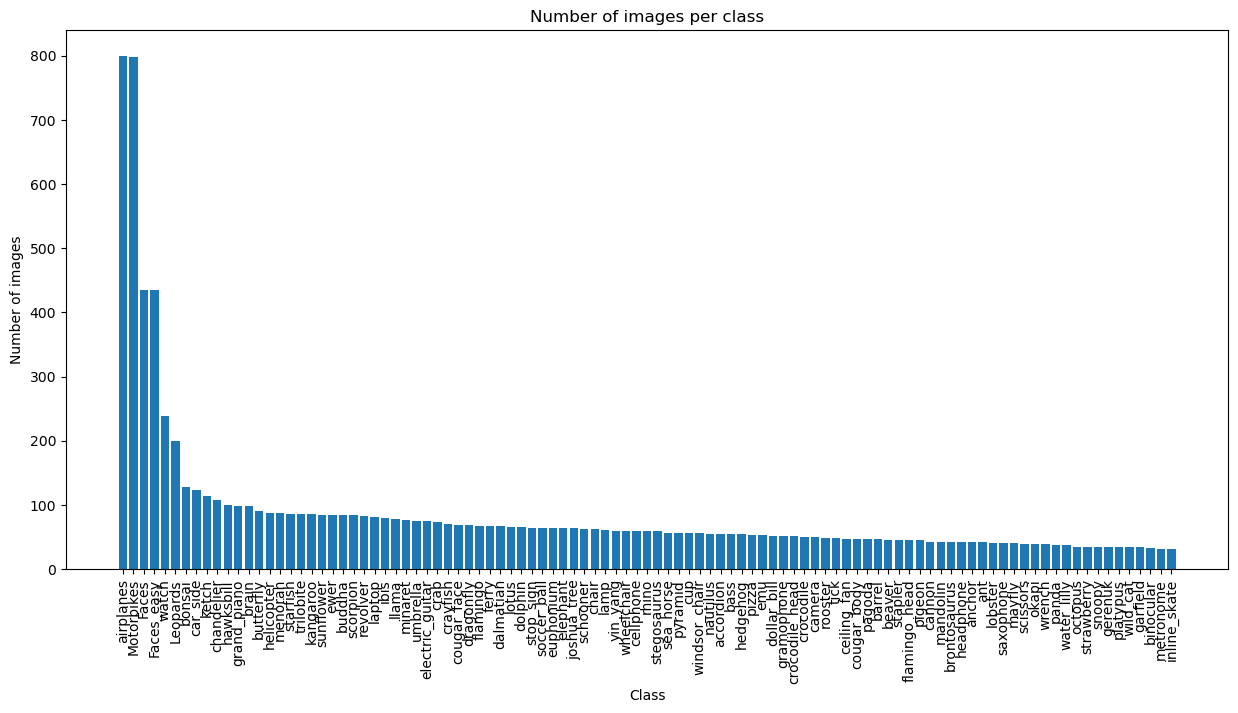
I decided to choose the Caltech-101 dataset as there is a wide range of classes and not so many total images that it would cause the training to take too long. I wanted to choose a bigger dataset than some of the ones we have been using in the labs so I can utilise some of the techniques we haven’t used from the lectures like SoftMax activation for classifying when there are more than two classes. Below is a python extract that shows the number of classes and the first 5 names.

The image below shows some numerical data about the data set

A screenshot of a computer program

Description automatically generated

The image below shows a bar graph of the number of images per image class.



**Data pre-processing and feature selection (support with Python code snippets)**

The Caltech 101 dataset contains 101 directories, each named after the class they represent and inside each of those directories is the set of images for that class. I used tf.keras.preprocessing.image.load\_img for loading the images and tf.keras.preprocessing.image.img\_to\_array for transforming the images into a NumPy array of its colour data as seen below in the code snippet.

The function above simply loops through each directory and each image inside then loads each image to a raw RGB pixel value format then adds it to an array.

I call this function using the code below, specifying that I will load all the images as 128x128 pixels. After normalising the RGB values to be in the range of 0 to 1, I call sklearn.model\_selection.train\_test\_split to get a testing and training split of my data.

**Machine learning model #1 – Convolutional Neural Network**

I decided to use a convolutional neural network for my first model as they are state of the art for image classification, and we have been studying their intricacies in class. CNNs are highly customisable in terms of their layer architecture allowing them to be tuned for each dataset. For my model, I concluded after much tweaking that the layers shown in the image below gave an acceptable 98% accuracy.

**Machine learning model #2 – Support Vector Machine**

For my second choice

**Results comparison across the models built (support with tables/figures)**

d

**Conclusion, recommenda ons, and future work**

d

**References**

d