**Research – Week 5**

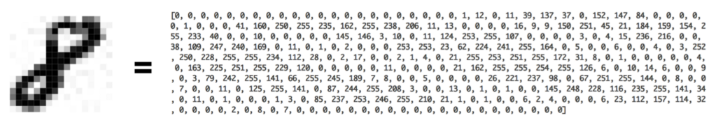
* **You should begin looking at how to train an ML model to classify images that could be deployed with Tensorflow Lite or some other alternate mobile ML library – first task is finding out what your options are for implementation, then you can look to evaluate them. Most likely we would be using a pre-trained model that is then adapted for our particular task.**

* **Useful Links**
  + <https://developers.google.com/machine-learning/practica/image-classification>
  + <https://www.tensorflow.org/tutorials/images/classification>
  + <https://iq.opengenus.org/basics-of-machine-learning-image-classification-techniques/>
  + <https://www.tensorflow.org/lite>
  + <https://medium.datadriveninvestor.com/how-to-build-a-machine-learning-app-choosing-the-best-image-recognition-api-297d4d97c84c>
  + <https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/>
  + <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
  + <https://www.kdnuggets.com/2016/06/top-machine-learning-libraries-javascript.html> - top js machine learning libraries
* **Useful Academic Articles**
  + <https://www.sciencedirect.com/science/article/pii/S2352711017300869> - FAIMS
  + <https://arxiv.org/abs/1511.06348> - accuracy of CNN
  + <https://dl.acm.org/doi/abs/10.1145/3334480.3382839?casa_token=KhJ9HMZK7OEAAAAA:AkpeoltzyiqZVQ-H-b3ZJ-IBtUk1jkxx9AW9McyEgRvoePrSHc-SVen61hJwC3igvvTZFGv-kA_0jQ>
  + <https://ieeexplore.ieee.org/document/9238174>
    - Neural Networks for image recognition
  + <https://towardsdatascience.com/mobile-phones-an-image-classification-problem-8dd207205c0d>
  + <https://dl.acm.org/doi/pdf/10.1145/212094.212114> - overfitting
  + <https://proceedings.neurips.cc/paper/2019/file/ee39e503b6bedf0c98c388b7e8589aca-Paper.pdf> - overfitting
  + [https://iopscience.iop.org/article/10.1088/1742-6596/1168/2https://doi.org/10.1007/s10664-020-09818-7 /022022/pdf](https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022/pdf) - overfitting

**How to Train a ML Model for Image Classification**

* What is Image Classification?
  + Complex issue for machines
  + Image classification is an important task within the field of computer vision
  + Refers to the labelling of images into one of a number of pre-defined classes
  + The issue is:
    - There can be potentially *n* number of classes in which a given image can be
    - To check each image and classify them manually would require an extensive effort i.e. (*n = 100)* classes with an input of *(i = 10,000)* images
    - This effort could be mitigated by using computer vision to automate the task and classify the image
  + Real-world examples:
    - Labelling an x-ray as cancer or not
      * Binary classification
    - Classifying a handwritten digit
      * Multiclass classification
    - Assigning a name to a photograph of a face
      * Multiclass classification
    - Autonomous driving
      * Image classification model that recognises objects like:
        + Vehicles
        + Pedestrians
        + Traffic Lights
        + Signs
* Structure of an Image Classification Task
  + 1. Image Pre-processing
    - The aim of this process is to improve the image data by suppressing unwanted distortions and enhancement of some important image features which will help for future processing
      * The computer vision models can benefit from this improved data to work on
    - Pre- processing
      * Common name for operations with images at the lowest level of abstraction
        + Lowest level of abstraction describes how a system stores data
        + Physical level

Describes low-level data structures in detail

* + - * Both input and output are intensity images
      * Computers can’t comprehend images
        + There has to be a way to convert images to numbers for the computer to understand
        + Pre-processing
    - Steps of Image Pre-processing
      * Read image
        + Store the path to the image dataset into a variable and then create a function to load folders containing images into arrays so computers can deal with it



* + - * Resize image
        + Images can vary in size
        + It’s better for the AI algorithm if a standard size is established for each image
        + This can be achieved by resizing each image



* + - * Data augmentation
        + A way of creating new data with different orientations
        + Beneficial as you can generate more data from limited data and prevents overfitting

Overfitting – is a fundamental issue in machine learning which prevents from generalising the models to well fit observed data on training data, as well as unseen data on the testing set

The presence of noise, limited size of the training set, complexity of classifiers all contributes to the occurrence of overfitting

The model may perform perfectly on the training set whilst alternatively fitting poorly on the testing set. This is a result of the model having problems with the testing set due to the variance between the training set and the testing set

* + - * + Data Augmentation Techniques

Grey scaling of image

The image is converted to grey scale and the machine will assign each pixel a value based on how dark it is

The numbers are put into an array where the computer will do computations

Grey scale conversion code example

Reflection

Flipping an image vertically or horizontally (some frameworks don’t provide vertical flipping)

Background pattern

Description automatically generated

Gaussian blurring

Blurring an image using a Gaussian function

Reduces image noise



Histogram equalisation

Increases global contrast of an image using the image intensity histogram

Text

Description automatically generated

Rotation

Rotates the image

Image may not keep its original dimensions

Text

Description automatically generated

Translation

Moving the image along the X and/or Y axis

Useful as objects can be located anywhere in the image

Forces feature extractor to look everywhere

* + 2. Detection of an object
    - Detection refers to the localisation of an object which means the segmentation of the image and identifying the position of the object of interest
  + 3. Feature extraction and training
    - Crucial step
    - Statistical or deep learning methods are used to identify the most interesting patterns of the image, features that might be unique to a particular class and that will, later on, help the model to differentiate between different classes
    - This process where the model learns the features from the dataset is called model training
  + 4. Classification of an object
    - Categorises detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns
* Image Classification Techniques
  + Support vector machine
    - A supervised machine learning algorithm used for both regression and classification problems
    - For classification problems, it separates the classes using a linear boundary
    - Builds a hyper-plane or set of hyper-planes in high dimensional space
    - Good separation between the two classes is achieved by the hyperplane that has the largest distance to the nearest training data point of any class
    - Algorithm depends on kernel function being used
      * Linear kernel
      * Gaussian kernel
      * Polynomial kernel

Text

Description automatically generated

* + Decision trees
    - A supervised machine learning algorithm
    - At its core is the tree data structure only, using a couple if/else statements on the features selected
    - Decision trees are based on a hierarchical rule-based method and permits the acceptance and rejection of class labels at each intermediary stage/ level
    - This method consists of 3 parts
      * Partitioning the nodes
      * Finding the terminal nodes
      * Allocation of the class label to terminal node



* + K nearest neighbour
    - The simplest machine learning algorithm
    - The algorithm simply relies on the distance between feature vectors and classifies unknown data points by finding the most common class among the k-closest examples

Diagram

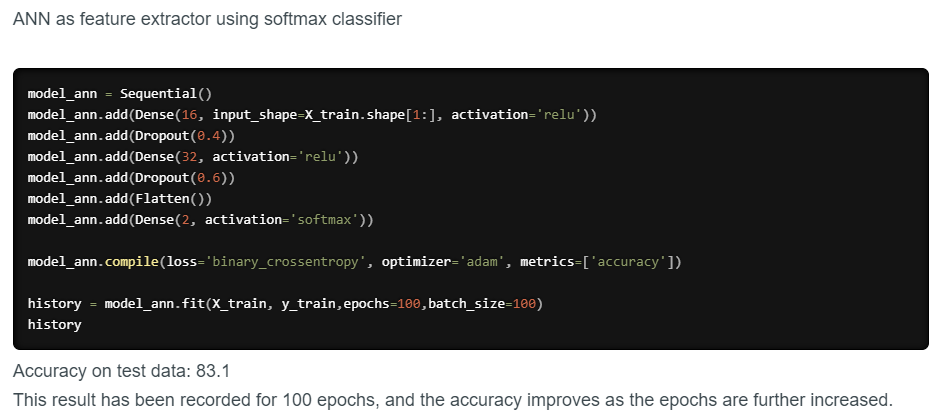
Description automatically generated

* + - Here we can see there are two categories of images and that each of the data points within each respective category are grouped relatively close together in an n-dimensional space
    - In order to apply the k-nearest Neighbour classification, we need to define a distance metric or similarity function. Common choices include the Euclidean distance and Manhattan distance

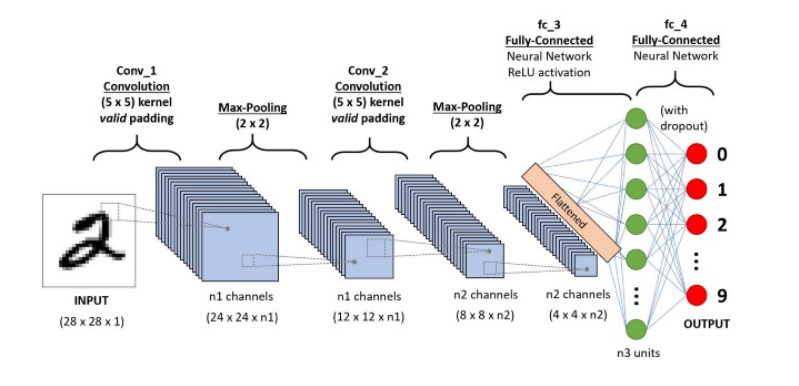
Text

Description automatically generated

* + Artificial neural networks
    - Inspired by properties of biological neural networks
    - Artificial Neural Networks are statistical learning algorithms
    - Are used for a variety of tasks
      * Simple classification tasks
      * Computer vision
      * Speech recognition
    - ANNs are implemented as a system of interconnected processing elements, called nodes, which are functionally analogous to biological neurons
    - The connections between different nodes have numerical values, called weights, and by altering these values in a systematic way, the network is eventually able to approximate the desired function
    - The hidden layers can be thought of as individual feature detectors, recognizing more and more complex patterns in the data as it is propagated throughout the network. For example, if the network is given a task to recognize a face, the first hidden layer might act as a line detector, the second hidden takes these lines as input and puts them together to form a nose, the third hidden layer takes the nose and matches it with an eye and so on, until finally the whole face is constructed. This hierarchy enables the network to eventually recognize very complex objects.



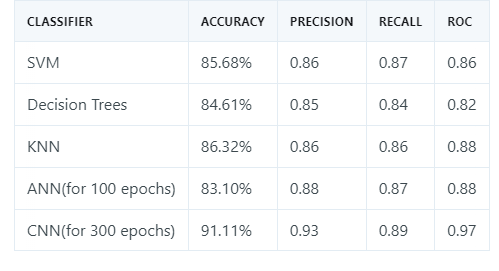
* + Convolutional neural networks
    - Special architecture of artificial neural networks
    - Uses some of its features of visual cortex – have achieved great results
    - Convolutional neural networks are comprised of two very simple elements
      * Convolutional layers
      * Pooling layers
      * Elements of CNN such as above are easy to understand
    - Although simple, there are near infinite ways to arrange these layers for given computer vision problems



Text

Description automatically generated

* + Performance of Image Classification Techniques



* How Accurate will this be?
  + This was a question asked early on
  + Seems that convolutional neural networks are the most accurate technique for image classification
  + For medical image classification it seems that the larger the training data size the more accurate the classification was
    - It specifies that training data size of 200 is most accurate with nearly 100% accuracy on some images of the body
    - Is this useful for our problem?
      * Probably the technique we would want to implement

**Pre-trained Machine Learning Model**

* Most of the work will be focused on extraction, formatting and pre-processing
* Trained on large general data set
* What pre-trained model are we thinking about using?

**Options for Implementation**

* Tensorflow lite
  + Tensorflow JS
    - Can execute ML models in JS
    - ML in the
      * Browser client side
      * Server side > Node.js
      * Mobile native via react native
      * Desktop native > Electron
      * IOT devices via Node.js on Raspberry Pi
      * <https://www.tensorflow.org/resources/learn-ml/basics-of-tensorflow-for-js-development>
* Scikit-Learn
  + Tensorflow is better for deep learning
  + Scikit-Learn is better for traditional machine learning
  + <https://scikit-learn.org/stable/>
  + <https://www.npmjs.com/package/scikit-learn>
* Amazon Rekognition
  + [https://medium.com/@glen.bray/text-detection-with-mobile-camera-using-react-native-and-aws-rekognition-7826b3e2aeef \](https://medium.com/@glen.bray/text-detection-with-mobile-camera-using-react-native-and-aws-rekognition-7826b3e2aeef%20\)
  + <https://morioh.com/p/4b289b022040>
  + Best for in depth facial recognition
    - Facial comparisons for same person
    - Facial feature detection
* Einstein Vision
  + Einstein Image Classification
    - Good for scale
    - Create and train models to detect and classify images
    - Use of pre-trained model?
* Clarifai
  + Easy to implement API for tagging images
  + The API has strong concept modelling
  + Allows for potentially creating and training our own models to test against
* Google vision api
  + Good for detecting extensive details about the image
* Mobile vision api
  + Deprecated now ML Kit
* ML Kit
  + Vision Api’s
* Flask + Keras
  + Python framework Flask
  + Machine learning library Keras
  + REST API capable of classifying images
  + Can use with React
  + <https://medium.com/sopra-steria-norge/build-a-simple-image-classification-app-using-react-keras-and-flask-7b9075e3b6f5>
  + Example project structure:

