## Introduction

This project aims to effectively develop a model of supervised learning which is able to effectively predict hand-drawn drawing into 6 classes which are “banana”, “boomerang”, “cactus”, “crab”, “flip flops” and “kangaroo”. A training set and test set containing different drawings of each class has been provided, the prediction model has been developed and derived using these inputs as the only source.

A close up of a logo

Description automatically generated

Figure . A sample of sketches given in the data set.

### Data Exploration

It is extremely important to train with a balanced dataset as machine learning classifiers are sensitive to the proportions of the dataset. This is because in imbalanced datasets, these classifiers tend to favour the class with the largest number of observations which may lead to misleading accuracies.

For example, consider a dataset with two distinct classes, where one consists of 95% of the dataset and the other class consists of the remaining 5%. A model that predicts the majority class every time would yield an accuracy of 95%. Clearly this is deceiving and not the desired behaviour.

A picture containing drawing

Description automatically generated

Figure 2. In the Pictionary training dataset, it can be seen that the dataset is perfectly balanced between the 6 classes. There are 1000 images in each class giving a total of 6000 images. We therefore do not have to worry about working with an unbalanced data.

The entire 6000 images were looked over for any outlier images. As can be seen below, these images include scribbles, black boxes and text of the words instead of drawings. Some images are of different objects altogether. For example, there is a lollipop that is labelled as flip flops.

Training was done including and not including these images. It was found that leaving these images in resulted in a higher final total accuracy of approximately 1%. It can be speculated the network somehow managed to find patterns in these images to help it predict on the test set even though we couldn't see any ourselves.

A close up of a keyboard

Description automatically generated

Figure 3. Outlier Images found in the training dataset.

## Methodology

The task at hand was to build an image classifier that could classify images into 6 distinct classes. In recent years, the clear winner out of the machine learning algorithms are deep learning approaches, in particular using a convolutional neural network.

### Deep Learning

Deep learning is a subset of machine learning in which neural networks are trained to receive input data and make a prediction based on that data. Neural networks are based on the human brain and are built from many neurons which are arranged in layers from the input layer to the output layer.

In the simplest type of neural network, the fully connected neural network or multi-layer perceptron (MLP) all the neurons from one layer connect to all the neurons in the next layer. These neurons in the next layer receive the outputs of the neurons from the previous layer as its input. The dot product between this input vector and a weight vector added with a bias is the output of the neuron (after an activation function is added).

It is the weight vector that each neuron in an neural network possesses that are tuned during training. A neural network that possesses a single hidden layer is sufficient to approximate any function. It is this ability for a neural network to be a universal function approximator that makes a neural network so powerful.

A close up of a device

Description automatically generated

Figure . Neural Network Process

### The Chosen Model

The Convolutional Neural Network (CNN) has been chosen as the model to undertake the requirements of this project. Convolutional neural networks are a class of neural networks that are particularly suited for image recognition.

One of the problems with a simple fully connected neural network with image recognition is due to its fully connected nature, connections between pixels far away from each other in the image are given the same importance as connections between pixels close together. Although the neural network can learn to subdue these connections this spatial structure is left unexploited. MLPs also can end up having a huge number of connections between neurons, especially if the image is of high resolution.

The main difference of the CNN to other neural networks is that this network model possesses a convolutional layer.  In the convolutional layers, different filters (kernels) are applied to the input images to understand the various features of an image. Filters can be used to assist in understanding aspects of an image such as identifying edges, they also assist in performing operations such as sharpening or blurring the image. Each filter(kernel) assists the model in understanding and learning more about the input image and the relationships between certain pixels within the input image.

Parameters for these convolutional layers also need to be chosen for the effectiveness and efficiency of the network. Common parameters include strides and padding.

• Strides – Number of pixels to shift over in the input before a filter is applied again to the input

• Padding – When filter does not fit perfectly with the input. The input can be padded with zeros or parts of the image can be dropped where filters cannot be applied.

### Prevention of Overfitting

The main concern during the training of the network is overfitting. Overfitting is where the classifier starts to model the small random variations in the training set. This hinders the network's ability to generalize to unseen images and can reduce its accuracy.

Unfortunately, neural networks are highly prone to overfitting, unless the number of observations in the training set is extremely large. Therefore, much care must be placed in ensuring overfitting is minimized.

### The following strategies have been used in this model to prevent overfitting>

* Network Architecture, simpler models tend to overfit more
* Data augmentation

Data Augmentation is a strategy that helps significantly increase of data without collecting any new data. Common techniques involve cropping, padding and flipping.

Augmentation Techniques used in the chosen model are:

o Rotations – Input images are randomly rotated by angles up to 15 degrees

o Affine Transformation – random transformations, but collinearity and ratios of distances remain the same. Affine rotations have been disabled and translations of up to 10% are occurring.

o Cropping – Images are cropped randomly up to 10%.

o Horizontal Flipping – 50% of the images are flipped horizontally.

These techniques provide the model with image variations that enables the network to learn further important features from the current dataset which provides limited quantity and diversity and significantly reduces overfitting.

* Splitting into training, validation and test to check whether overfitting
* Batch Normalization

Batch Normalisation is a process in which the output of an activation layer is normalised by standardising the output based on batch mean and standard deviation. The exact reasons for why batch normalisation works is still relatively unknown. However is speculated that this enables greater stability of the neural network and would also likely speed up the training process. This is because the input layer benefits because all input features will have a small range as a result of the standardisation.

* Drop out

The dropout is another regularisation method used to reduce overfitting in neural networks. Models such as this with large neural networks and small datasets can overfit the training data very easily.

During the training of the model, some neurons are randomly temporarily removed.

As a neural network learns, neuron weights settle into their context within the network. Weights of neurons are tuned for specific features providing some specialization. Neighbouring neurons become to rely on this specialization, which if taken too far can result in a fragile model too specialized to the training data. This reliant on context for a neuron during training is referred to complex co-adaptations.

You can imagine that if neurons are randomly dropped out of the network during training, that other neurons will have to step in and handle the representation required to make predictions for the missing neurons. This is believed to result in multiple independent internal representations being learned by the network.

The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data.

Dropout rate values range from 0 to 1. 1 is for no dropouts and 0 is for no inputs. In common practice dropout rates range between 0.5 to 0.8. Using batch normalisation in the model allows the use of a lower dropout rate and as such 0.5 has been chosen.

* Learning rate choice

Learning rate is the parameter which chooses how quickly a model learns the model. It is also referred to as step size and is the rate which determines the amount that “the apportioned error that the weights of the model are updated with each time they are updated.”

Higher learning rates will enable a model to learn faster but may risk the possibility of a lower quality final set of weights. A smaller learning will produce an optimal set of weights but the model will take longer to train.

During training, the learning rate is typically lowered after a certain number of epochs. This allows the model to quickly train towards a general optimal region, and for the lower learning rate to be used to refine the final set of weights.

After trial and error, the learning rate used in the final model was 0.001, which was dropped to 0.0001 after 200 epochs. The model trained for 250 epochs.

* Early Stopping, don't train for too long

When training neural networks it is important to assess how long to train the model. When training is kept to a minimum this will likely underfit the model. Training the model for too long will result in the network overfitting the training set.

Early Stopping occurs when the training of the model is stopped when the validation dataset performance begins to diminish. This can be assessed from the Net Loss and Net Accuracy graphs provided in this report. Both these reports suggest the Validation set performance begins to diminish when x = 25(not sure what x represents). Hence, this should be used as the Early stopping to ensure that the model is not overfitted, but still contains enough information for efficient performance on the test set.

* Might try ensembling networks
* L2 Regularization

L2 Regularization is a regularisation technique in which the size of the weights is added to the loss function, thereby punishing models with large weights. Intuitively, simpler models are less likely to overfit. Therefore by encouraging the model to use small weights, overfitting is reduced. A parameter of 0.0001 was chosen for the weight decay. This is quite a small number and in practice, L2 regularization was not found to be too helpful, however it was left in as it did not seem to lower the accuracy.

### 

## Results and Discussion

## Conclusion

References

Source: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

Image source <https://www.researchgate.net/profile/Facundo_Bre/publication/321259051/figure/fig1/AS:614329250496529@1523478915726/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o.png>

<https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/>

Source: <https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/>

Source: <https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/>

Appendix