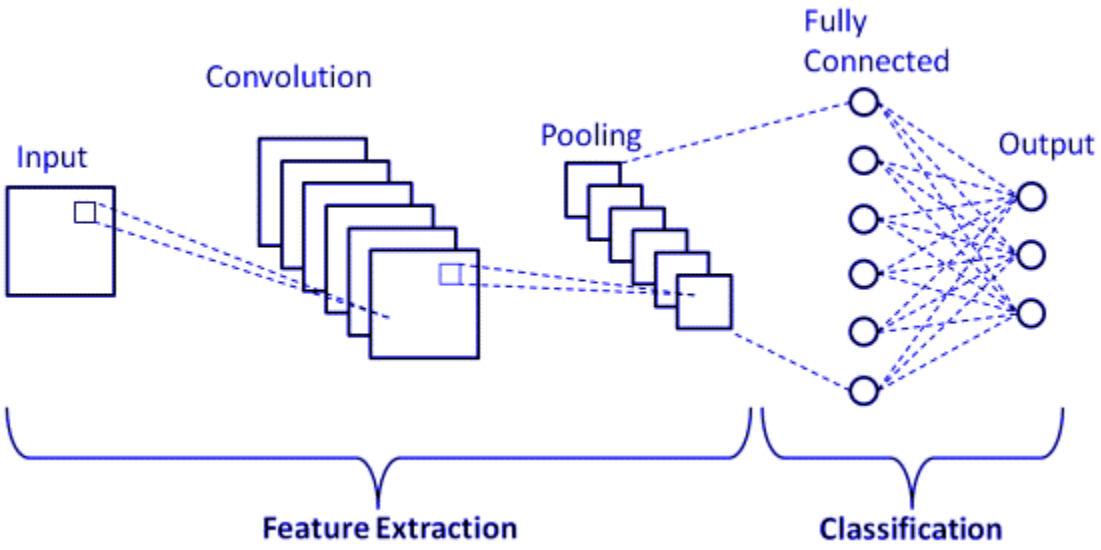
Convolutional Neural Network
for
Waste Classification



Name:	Student Number:
Bhavish Mohee	MHXBHA001
Rahul Mistri	MSTRAH001

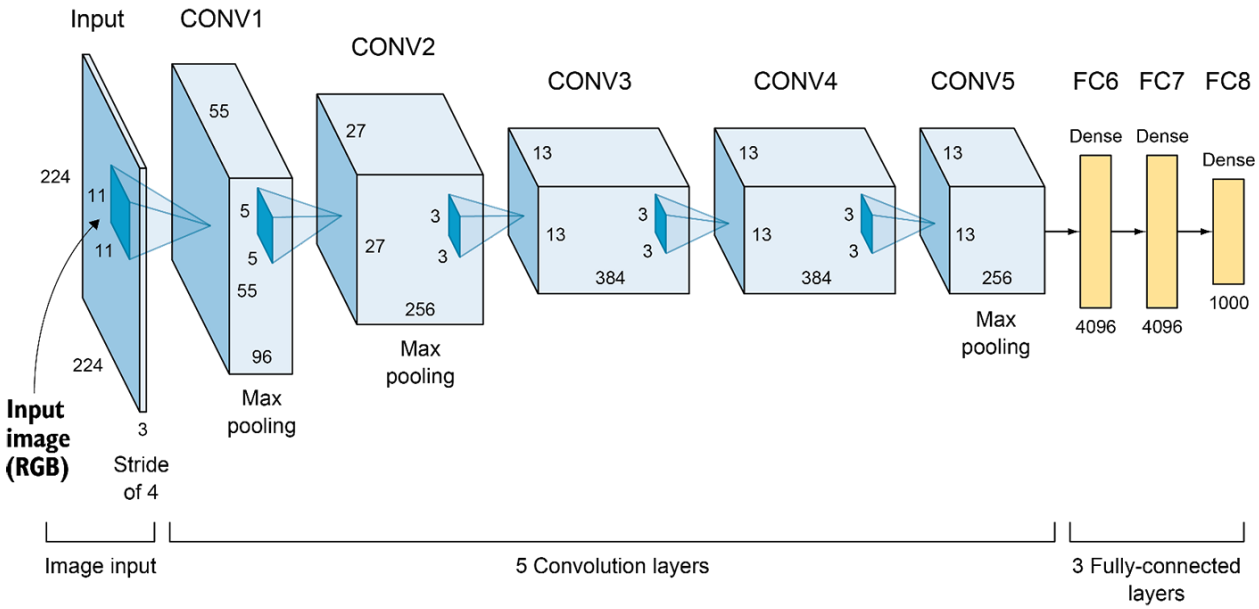
Problem Description

In the era of industrialisation and technology, the generation of waste is an inevitable outcome of human activities. The world generates around 2 billion tonnes of solid waste annually, and South Africa, alone, is responsible for roughly 2% of this number, while only accounting for about 0.7% of the world's population. In 2017, South Africa produced approximately 55.6 million tonnes of general waste and barely 34% of that is recycled. The remaining amount of waste is disposed of in landfill sites. As the amount of garbage increases exponentially, the need for more landfill space becomes more crucial. With a growing population, the use of more landfills makes no sense. Consequently, this issue leads to the improper disposal of waste. During 2017, it was estimated that 7.8 million tonnes of solid waste were improperly disposed of, instead of using the allocated sanitary landfill sites. Based on evidence from research papers and surveys, bad waste management majorly contributes to air, water and soil contamination. Eventually, this adversely affects the environment through the floral, faunal and marine ecosystems.

While the most effective way of reducing waste is to encourage reuse and encourage a culture of anti-waste, existing waste still poses a major problem. Many forms of waste disposal, such as incineration, lead to equally harmful environmental impacts such as dumps. The recycling of materials that cannot decompose organically is thus the most favoured option. This procedure brings forwards a plethora of benefits such as conserving natural resources and minimising global warming. However, the most hindering part of the whole recycling chain process is the waste selection between recyclable and organic waste types.
We present a convolutional neural network which will be trained to classify images of waste as either Organic or Recyclable waste.

The accuracy of the model will impact the usefulness of its intended purpose. If good accuracy is achieved, this model can be extended into a sorting application through more in-depth research on the matter. This concept is mainly significant to governments as it will provide all the attached benefits of recycling on a national scale as well as help mitigate costly investments into ineffective waste management systems. We do not believe that there are any ethical issues in building this model. In fact, should the model prove effective and useful to manufacturers, the model can help mitigate the negative ethical impacts in a variety of industries, as well as promote a sense of social responsibility, pride in the community and create environmental awareness.

Baseline and Incremental Model Designs



The baseline models' architecture used in this experiment was based on AlexNet, a convolutional neural network that was one of the first to use Rectified Linear Units, and boasts significantly low error rates without overfitting to the data. In 2012, AlexNet won the Imagenet challenge, which had a dataset containing about 14 million images and 1000 classes.
AlexNet consists of 8 layers, 5 convolutional layers with max-pooling, followed by 3 fully connected layers with dropout layers to prevent overfitting.
We used the architecture of AlexNet as a guiding framework when designing the models incrementally.

Model0 consisted of one convolutional layer and max-pooling, followed by 3 fully connected layers.
Model1 improved on the design of Model0, with 3 convolutional layers with the same filter and kernel size and max-pooling, followed by 3 fully connected layers as well.
Model2 followed on from the design of Model1, with an identical structure except that the filter sizes increased in each convolutional layer, as is done in AlexNet.
The design of Model3 was adapted from Model2's design and followed AlexNet more closely by using kernel sizes of (11,11), (5,5) and (3,3) in each of the respective convolutional layers. Model3 bears the most resemblance to AlexNet.
Model4 was based on design principles (Ramesh, 2018) for convolutional neural networks. Conversely to AlexNet, kernels of increasing size were used in the convolutional layers to capture small details first. Filters of increasing size were used to increase the depth of the feature space with each successive convolution. A combination of fully connected layers and dropout layers were used to help prevent the model from overfitting, just as is done in AlexNet.

Dataset and Training

In total, the dataset had approximately 25000 images, of which 80% was used for training, 10% for validation and 10% for testing.
We trained each of the CNNs for 5 epochs split into batches of 256 images. The number of epochs indicates how many times the training dataset is passed through the model for training.

Results, Model Validation and Evaluation

For each of the models below, the accuracy of the CNN on the training data and validation dataset as well as the associated losses are displayed as each model is trained. The model training was carried out locally using NVIDIA GeForce MX150 GPU instead of Google Colab system as low-performance GPUs were being assigned which massively impacted the training time for each epoch. Using Google Colab was also time-consuming; after each disconnection from an allocated GPU due to network issues or idle time restrictions, the datasets took a long time to be reloaded from Google Drive.

AlexNet

Training time: 46m11s

```
Epoch 1/5
79/79 [==============================] - 506s 6s/step - loss: 0.4790 - accuracy: 0.8043 - val_loss: 1.5385 - val_accuracy: 0.5506
Epoch 2/5
79/79 [==============================] - 483s 6s/step - loss: 0.4339 - accuracy: 0.8402 - val_loss: 0.5013 - val_accuracy: 0.7548
Epoch 3/5
79/79 [==============================] - 555s 7s/step - loss: 0.4151 - accuracy: 0.8466 - val_loss: 0.8332 - val_accuracy: 0.5068
Epoch 4/5
79/79 [==============================] - 619s 8s/step - loss: 0.3931 - accuracy: 0.8589 - val_loss: 0.6928 - val_accuracy: 0.6505
Epoch 5/5
79/79 [==============================] - 603s 8s/step - loss: 0.3747 - accuracy: 0.8696 - val_loss: 0.5401 - val_accuracy: 0.7273
```

Model0

Training time: 17m53s

```
Epoch 1/5
79/79 [==============================] - 221s 3s/step - loss: 4.7618 - accuracy: 0.7380 - val_loss: 0.4245 - val_accuracy: 0.8328
Epoch 2/5
79/79 [==============================] - 195s 2s/step - loss: 0.4549 - accuracy: 0.8194 - val_loss: 0.3746 - val_accuracy: 0.8388
Epoch 3/5
79/79 [==============================] - 195s 2s/step - loss: 0.4110 - accuracy: 0.8323 - val_loss: 0.3921 - val_accuracy: 0.8304
Epoch 4/5
79/79 [==============================] - 227s 3s/step - loss: 0.3775 - accuracy: 0.8483 - val_loss: 0.3834 - val_accuracy: 0.8368
Epoch 5/5
79/79 [==============================] - 233s 3s/step - loss: 0.3535 - accuracy: 0.8615 - val_loss: 0.3791 - val_accuracy: 0.8443
```

Model1

Training time: 16m21s

```
Epoch 1/5
79/79 [==============================] - 189s 2s/step - loss: 0.4765 - accuracy: 0.7861 - val_loss: 0.3560 - val_accuracy: 0.8340
Epoch 2/5
79/79 [==============================] - 187s 2s/step - loss: 0.4037 - accuracy: 0.8274 - val_loss: 0.3533 - val_accuracy: 0.8551
Epoch 3/5
79/79 [==============================] - 186s 2s/step - loss: 0.3732 - accuracy: 0.8439 - val_loss: 0.3741 - val_accuracy: 0.8658
Epoch 4/5
79/79 [==============================] - 192s 2s/step - loss: 0.3571 - accuracy: 0.8537 - val_loss: 0.3097 - val_accuracy: 0.8754
Epoch 5/5
79/79 [==============================] - 226s 3s/step - loss: 0.3332 - accuracy: 0.8637 - val_loss: 0.2993 - val_accuracy: 0.8762
```

Model2

Training time: 25m59s

```
Epoch 1/5
79/79 [==============================] - 327s 4s/step - loss: 0.4970 - accuracy: 0.7750 - val_loss: 0.3924 - val_accuracy: 0.8169
Epoch 2/5
79/79 [==============================] - 322s 4s/step - loss: 0.4017 - accuracy: 0.8334 - val_loss: 0.3048 - val_accuracy: 0.8818
Epoch 3/5
79/79 [==============================] - 303s 4s/step - loss: 0.3656 - accuracy: 0.8497 - val_loss: 0.3021 - val_accuracy: 0.8754
Epoch 4/5
79/79 [==============================] - 302s 4s/step - loss: 0.3487 - accuracy: 0.8597 - val_loss: 0.3210 - val_accuracy: 0.8714
Epoch 5/5
79/79 [==============================] - 304s 4s/step - loss: 0.3247 - accuracy: 0.8704 - val_loss: 0.2789 - val_accuracy: 0.8925
```

Model3

Training time: 47m27s

```
Epoch 1/5
79/79 [==============================] - 581s 7s/step - loss: 0.5690 - accuracy: 0.7175 - val_loss: 0.4017 - val_accuracy: 0.8037
Epoch 2/5
79/79 [==============================] - 561s 7s/step - loss: 0.4744 - accuracy: 0.7902 - val_loss: 0.3751 - val_accuracy: 0.8344
Epoch 3/5
79/79 [==============================] - 566s 7s/step - loss: 0.4580 - accuracy: 0.7987 - val_loss: 0.3623 - val_accuracy: 0.8388
Epoch 4/5
79/79 [==============================] - 553s 7s/step - loss: 0.4442 - accuracy: 0.8072 - val_loss: 0.3368 - val_accuracy: 0.8443
Epoch 5/5
79/79 [==============================] - 585s 7s/step - loss: 0.4238 - accuracy: 0.8178 - val_loss: 0.3565 - val_accuracy: 0.8531
```

Model4

Training time: 53m 42s

```
Epoch 1/5
79/79 [==============================] - 674s 9s/step - loss: 0.5485 - accuracy: 0.7352 - val_loss: 0.3432 - val_accuracy: 0.8388
Epoch 2/5
79/79 [==============================] - 633s 8s/step - loss: 0.4419 - accuracy: 0.8123 - val_loss: 0.3340 - val_accuracy: 0.8627
Epoch 3/5
79/79 [==============================] - 627s 8s/step - loss: 0.4039 - accuracy: 0.8312 - val_loss: 0.3355 - val_accuracy: 0.8515
Epoch 4/5
79/79 [==============================] - 634s 8s/step - loss: 0.3802 - accuracy: 0.8410 - val_loss: 0.3294 - val_accuracy: 0.8746
Epoch 5/5
79/79 [==============================] - 653s 8s/step - loss: 0.3642 - accuracy: 0.8511 - val_loss: 0.3133 - val_accuracy: 0.8718
```

Testing, Analysis Of Model Performance and Future Work

	AlexNet	**Model0**	**Model1**	**Model2**	**Model3**	**Model4**
Accuracy	0.6872	0.8778	0.8890	0.8985	0.8671	0.8627

Based on the accuracy metric scores from each model during training and testing, the data suggests the following:

- While state of the art models such as AlexNet are useful, better performance can be extracted by using the model as inspiration and performing incremental changes.
- Increasing the depth of the model can lead to an increase in performance.
- Using small kernel sizes can lead to a better model. In Model3 and Model4, the kernel size was decreased/increased respectively, however Model1 and Model2 used small, constant kernel size and outperformed them.

Each model we designed performed well in testing, with incremental gains realised between models Model0, Model1 and Model2. Taking into account the training time of the models and their results, Model2 is the most successful model.

We identify the following shortcomings of our models of which future work can be done:

- More training data should be used as CNNs typically require larger amounts of data than other models
- Given more time and computational power, all models should be trained from scratch and tested numerous times to provide a more accurate analysis
- The number of epochs for each model should be varied as well to find the optimal amount of times a particular model should see the training data. It is possible that AlexNet was overfitted after seeing the dataset for the third time, as it was peculiar that the model did not perform comparably to the other models.
- Variations to images in the image dataset, such as noise and various transformations, should be introduced for training to improve the model in training.

Reproducibility

A jupyter notebook named “RecreateTestResults.ipynb” is included in the submission directory. The notebook file generates the test data required to reproduce the test results mentioned above. Additional instructions are listed as comments in the notebook.

References

Africa, A., 2021. *South African waste management sector can still improve*. [online] Averda. Available at: <https://www.averda.com/rsa/news/south-african-waste-management-sector-can-still-improve> [Accessed 7 November 2021].

Analytics India Magazine. 2021. *Hands-on Guide To Implementing AlexNet With Keras For Multi-Class Image Classification*. [online] Available at: <https://analyticsindiamag.com/hands-on-guide-to-implementing-alexnet-with-keras-for-multi-class-image-classification/> [Accessed 7 November 2021].

Makgae, M., 2011. Key areas in waste management: A South African perspective. *Integrated Waste Management - Volume II*.

Medium. 2021. *A guide to an efficient way to build neural network architectures- Part II: Hyper-parameter…*. [online] Available at: <https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-ii-hyper-parameter-42efca01e5d7> [Accessed 7 November 2021].

Medium. 2021. *What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network?*. [online] Available at: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a> [Accessed 7 November 2021].