COMP3330: Machine Intelligence
Semester 1, 2021

Project B: Deep Learning Project Group

Olivia Favos C3188124 and Mirak Bumnanpol C3320409

1 Introduction

In this assignment, we will explore various training models and apply transfer learning to obtain training and evaluation data. This report details how the first task, Skin Cancer MNIST: HAM10000 Data, uses ResNet50 and MobileNet to run and compare two different training models. The second task, Plastic Waste in the Environment, is to develop a computer vision system which can detect and classify waste that can be collected. This report will provide a brief summary of the outcome of the training models through the use of experiments on different training networks. All team members put in the same amount of work.

2 Skin Cancer MNIST: HAM10000 Data

For this task, we train a neural network to classify images of pigmented skin lesions for automated diagnosis. This task required that we run and compare at least two different model architectures, using both a larger and a smaller network to train the Skin Cancer MNIST: HAM10000 dataset [4]. Using pre-trained models of ResNet50 as the larger CNN, and MobileNet as the smaller CNN, we compared features and differences of the two model architectures when used on the dataset.

The below figures show the analysis of the dataset that we worked with.

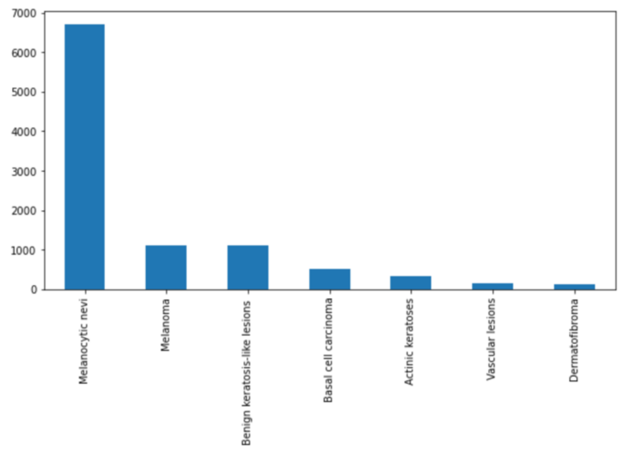

Figure 2: Skin Lesion Stats in Dataset

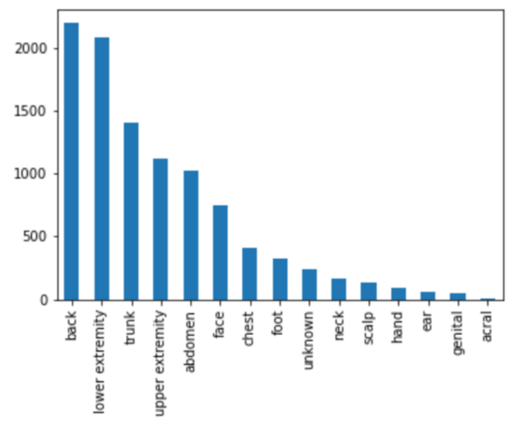

Figure 1: Localisation of Skin Lesions Stats

Both models used the same CNN as it was found to work more accurately. We also kept getting an error when trying to parse the dictionary for the key 'images', so a larger CNN with more classifications and batch normalisation was used instead of a smaller one. The CNN consists of a convolutional stack, which contains convolutional filters, batch normalisation, ReLU activations and max pooling. Batch normalisation is applied after the convolution but before activation. The model uses transfer learning to help support faster training time and is more efficient than training the models bottom up due to lack of computational hardware. The data is cross-validated with both a test set and a validation set to avoid the issue of overfitting. Overfitting is also avoided by utilising data augmentation (datagen function in the code) and the ImageDataGenerator method. The optimiser used was Adam and Keras was heavily utilised to build the CNN. Both models were trained using fixed hyper parameters for consistency purposes.

Overall, we found that the training model which used ResNet50 performed more accurately and faster than the training model used with MobileNet. The training results and summaries are detailed under their respectful headings below.

2.1 ResNet50

For the training data, we apply some transformations, which allow us to generalise more effectively and prevent overfitting. ResNet models often implement double or triple layer skips which contain ReLu and batch normalisation in between. ResNet50 uses 50 convolutional layers, and all transfer learning models have a predefined shape of 1x3x224x224. It shows batch size, number of channels, and a width and height respectively. The weights used are based on ImageNet.

This model uses a larger network to experiment and see if the neural network is trained faster and more accurately. The 30 epochs took an average of approximately 24.87 seconds to train. As a cross-validation method was used, we are able to see that the highest training accuracy of the model was 99.85% but once the model is evaluated, the average validation accuracy becomes 76.15% with loss at 1.1201. The ResNet50 model trained quite fast but could've had a better average validation accuracy.

2.2 MobileNet

MobileNet is a streamlined architecture which utilises depth-wise separable convolutions to construct light weight deep neural network.

In terms of output performance, there was a noticeable lag that was not there for ResNet50. The 30 epochs took an average of approximately 29.77 seconds to train. This is definitely longer than the ResNet50 model but can be attributed to the nature of MobileNet as is it more efficient and effective when used with mobile applications with a smaller dataset. The average validation accuracy for the MobileNet sits at 75.15% and loss at 1.0511. The highest training accuracy is at 99.35%. Hence, we are able to see that training on MobileNet results in a slower time and slightly lower accuracy than training on a larger network such as ResNet50.

3 Plastic Waste in the Environment

This task involved developing a computer vision system for a fleet of autonomous ground robots and air drones which are capable of detecting and classifying waste into four classes: plastic bags, plastic bottles, other plastic waste, and no plastic in the image. Based off the model's evaluation, the waste can then be collected.

3.1 Collecting Data

Although we were provided a selection of images on Blackboard to use for this task, to more effectively train our model we sought to find alternate ways to obtain data for the given task. We collected additional images of plastic bottles from Google Image searches, using targeted keywords to find appropriate data. Alongside this, we took some of our own photos in a variety of different outdoor environments, including parks, the beach, and the University campus. Combining the data, we had 200 images total, with approximately half containing plastic waste.

3.2 Training the Network

For this task we chose to develop our own convolutional neural network model using Keras, which allowed us to build the model layer by layer using the Sequential class. In the figure below, the architecture of the model is detailed.

Layer (type)	Output Shape	Parameters
Conv2D	(None, 198, 198, 16)	448
Activation (ReLU)	(None, 198, 198, 16)	0
MaxPooling2	(None, 99, 99, 16)	0
Conv2D	(None, 97, 97, 32)	4640
Activation (ReLU)	(None, 97, 97, 32)	0

MaxPooling2D	(None, 48, 48, 32)	0
Conv2D	(None, 46, 46, 64)	18496
Activation (ReLU)	(None, 46, 46, 64)	0
MaxPooling2D	(None, 23, 23, 64)	0
Flatten	(None, 33856)	0
Dense	(None, 512)	17334784
Activation (ReLU)	(None, 512)	0
Dense	(None, 4)	2052
Activation (Softmax)	(None, 4)	0

Figure 2: Convolutional Neural Network Architecture

Due to a lack of available computational resources and time constraints, we minimised the total dataset count to the initial 40 images provided in Blackboard. Ideally, we wanted to train a broader set of data with transformations such as rotations, colour shifting, and image flipping so that the model could detect plastic waste objects with greater accuracy.

The model was compiled with a categorical cross-entropy loss function, since we are dealing with more than 2 label classes.

3.3 Experiments and Results

After splitting the dataset into testing, training, and validation data and building our model, we ran a variety of experiments both as a group and independently. The most successful result was the RMSprop. The code we have submitted for this project is what provided these results.

We experimented with different batch sizes, however, did not see a significant difference between the values used. RMSprop was chosen as the final optimisation function for the project, with the default learning rate applied. Throughout experimentation, we also used the Adam optimisation function, however it performed worse than RMSprop.

Unfortunately, the visualisations of the prediction model appear to be largely broken. Despite being provided the 4 different classifications, it does not appear to predict any class aside from "noplastic" and "otherplastic". An example of the output given from our prediction visualisation is shown in the figure below.

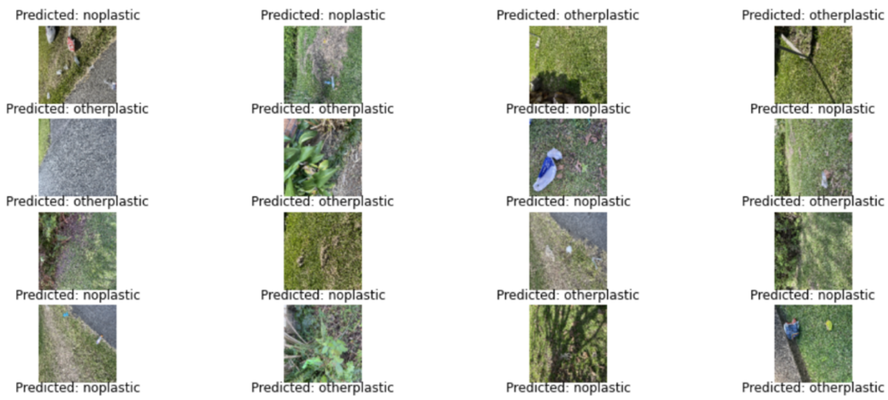

Figure 5: Prediction visualisation of the plastic waste data

4 Conclusion

Consequently, we have explored various training models and detail the results of training ResNet50 and MobileNet with the Skin Cancer dataset. It was clear that ResNet50 provided more accurate results in a shorter time period. Likewise, we were able to train our CNN in the Plastics task to achieve an 86% accuracy with a smaller but more diverse dataset.

References

1. Keras Team. 2021. Keras Applications. Available at: https://keras.io/api/applications/. [Accessed 09 June 2021].
2. Kenny Jaimes. 2021. Using ResNet for image classification. | by Kenny Jaimes | Jovian—Data Science and Machine Learning Tutorials. Available at: https://blog.jovian.ai/using-resnet-for-image-classification-4b3c42f2a27e. [Accessed 09 June 2021].
3. Keras Team. 2021. MobileNet and MobileNetV2. Available at: https://keras.io/api/applications/mobilenet/. [Accessed 09 June 2021].
4. Skin Cancer MNIST: HAM10000 | Kaggle. 2021. Skin Cancer MNIST: HAM10000 | Kaggle. Available at: https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000. [Accessed 09 June 2021].