ACTL3143 Assignment: Image Classification of Australian Animals

William Li z5257749

Problem specification

The goal of the project is to create a model that can classify a selection of Australian native animals. There are 8 animals in the dataset:

- Cockatoo
- Kookaburra
- Dingo
- Platypus
- Kangaroo
- Seadragon
- Koala
- Wombat

Models will be assessed with Top 1 categorical accuracy.

Data collection and pre-processing

There are 200 images for each animal. The images were collected from Google Images with the help of the Image Downloader extension[1]. They were selected manually to ensure accuracy and avoid duplicates (though the latter is not guaranteed as it is not possible to perfectly recall 200 images).

All images were rescaled to 128x128, as most machine learning techniques require all inputs to be the same size. The data was split into 4 sets. Their purpose and number of examples per class are:

- Training set (96),
- Validation set 1 (32),
- Validation set 2 (32), and
- Test set (40)

Prior to training the final model, validation set 1 was used for early stopping while validation set 2 was used to tune hyperparameters. For the final model, validation set 1 was added to the training set while set 2 was used for early stopping. Reasoning for 2 validation sets is explained in Appendix 1.

The training set was augmented to get 6 times as many examples (see Figure 1).

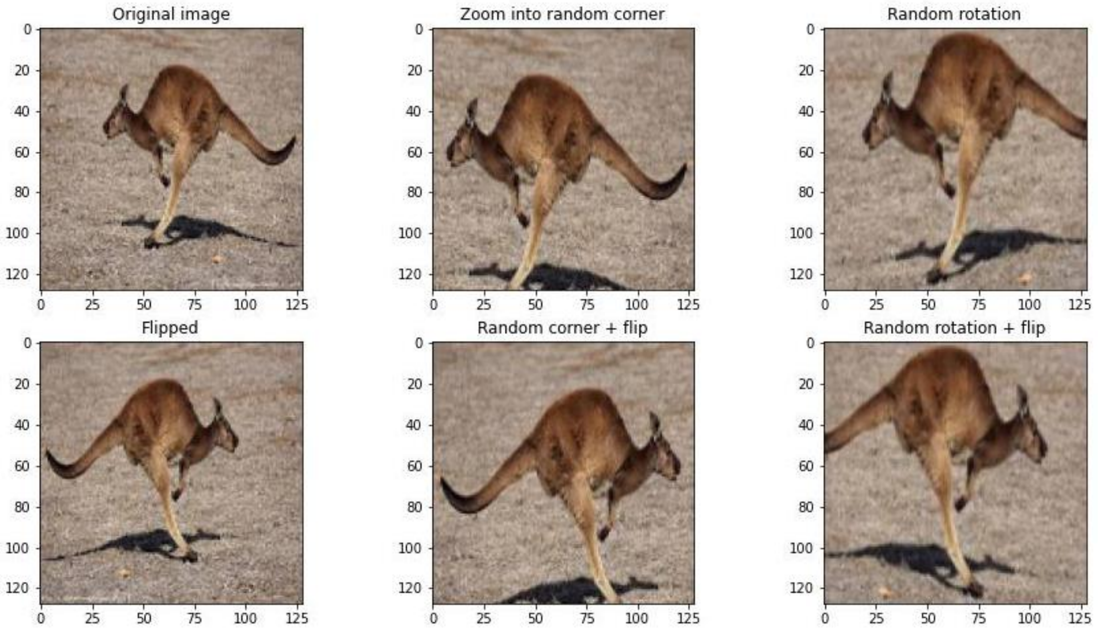


Figure 1: Example image and 5 transformed images. Transformations involve a combination of zooming into a corner at random, randomly rotating within a certain range, and flipping left-to-right.

Exploratory data analysis

Exploratory data analysis is not as crucial in image classification as in other problems. However, it is helpful to view a sampling of images (see Figure 2) as well as check that the colour channels are working as intended (see Figure 3).

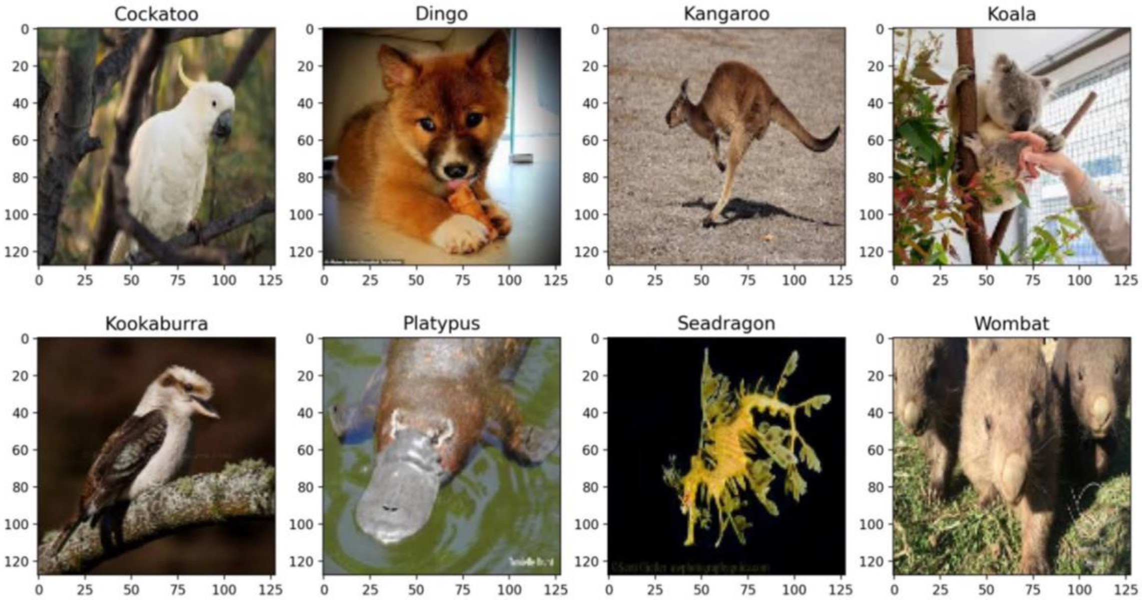


Figure 2: One random image for each of the 8 classes. Note that the aspect ratios are distorted both in absolute terms and relative to other images due to resizing from different aspect ratios to a square size (128x128).

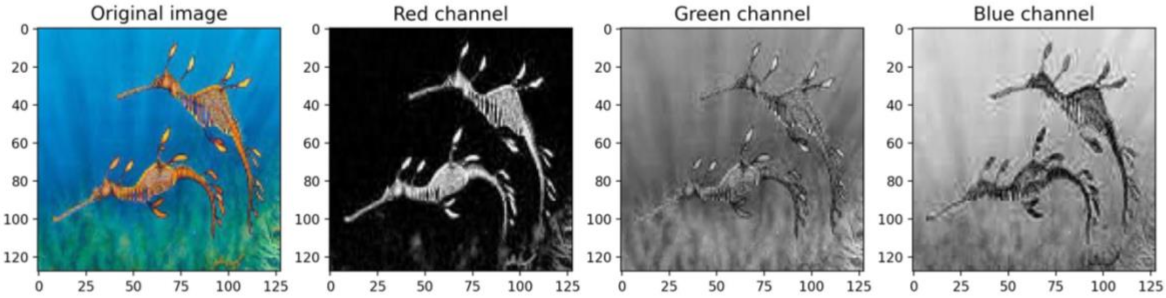


Figure 3: An image with its RGB values displayed separately in greyscale; the channels appear to be working correctly. For example, the red channel has low values (represented by dark colour on the "Red Channel") for the background which is blueish-green, and high values (represented by light colour) for the seadragons coloured orange.

Simple benchmark model: Logistic regression

Multi-class logistic regression was fit using the multinomial method. All images had to be scaled to 8x8 before fitting as this results in 8x8x3=192 inputs, which combined with an intercept results in 193 parameters. Having any more parameters than this would result in overfitting as logistic regression requires the number of observations to be much larger than the number of features.

The logistic regression model gave an accuracy of 46.88% in the validation set. It is not great but considerably higher than random guessing (which would give an accuracy of 12.50%). The confusion matrix and histogram of the probability estimates for the correct class are included in Appendix 2.

Simple neural network

The simple neural network is the most basic type of neural network. It uses dense layers where every neuron receives an input from all the neurons in the previous layer. The model in this report first resizes the image at the very start, in order to limit the input size to a reasonable scale.

The best simple neural net tried (see Figure 4 for architecture) got an accuracy of 48.83% on the validation set. This mediocre performance is expected, as immediately flattening the image into a single dimension causes it to lose information about the location of pixels relative to each other. This is also the reason why dense neural networks are no longer considered a top option for image classification nowadays.

The confusion matrix is included in Appendix 3a and the histogram of the probabilities predicted for the correct label is in Appendix 3b.

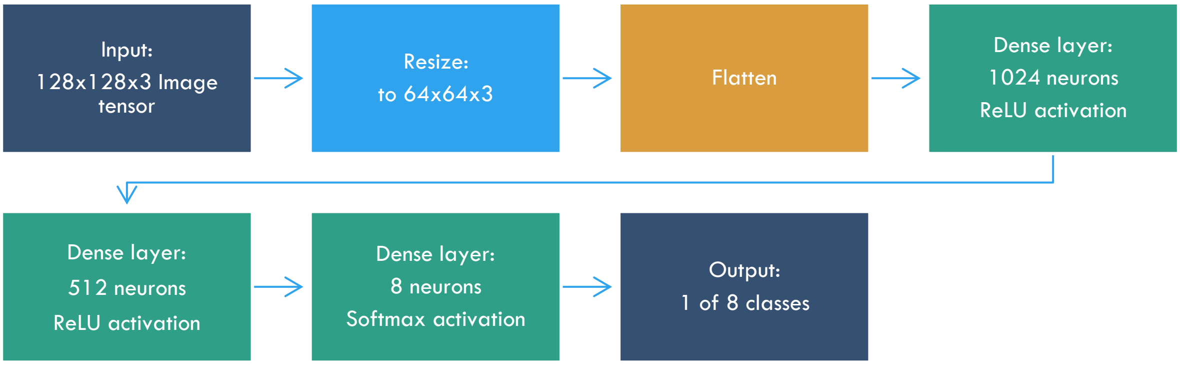


Figure 4: Simple NN architecture with the best set of parameters found through grid search

Hyperparameter tuning was completed using grid search. The process involves first choosing hyperparameters that are likely to affect model performance and defining a set of options for each. With the grid search method, the next step is to fit a separate model for all possible combinations within the hyperparameter space, and evaluate each one on the validation set to find its accuracy. The optimal hyperparameter combination is the one that results in the highest validation accuracy. It must be noted, however, that this combination may not be the best in general, as the superior performance may be due to variance when predicting on the validation set.

Table I outlines the hyperparameter tuning process for the simple neural network, showing the options for each hyperparameter and the combination that produced the highest accuracy on the validation set. Accuracy values of the top 10 combinations are included in Appendix 5a.

TABLE I: Hyperparameters tuned using grid search for Simple NN

Hyperparameter	Choices	Optimal combination
Resizing size	8, 16, 32, 64	64
Number of neurons in Layer 1	512, 1024	1024
Number of neurons in Layer 2	256, 512	512
Number of neurons in Layer 3	128, 256, No third layer	No third layer
Dropout rate	0, 0.2, 0.4	0
Activation	ReLU, tanh	ReLU

Convolutional neural network

Convolutional neural networks are one of the most common models for image classification. It makes use of convolution layers, which uses kernels that move across the previous layer to identify features within neighbouring pixels. The model also makes use of pooling layers, which groups information and reduces the dimensionality. After convolution and pooling, it is flattened into a single dimension and fed into a dense network.

The CNN architecture implemented in this report takes inspiration from classic models such as LeNet[2] and AlexNet[3]. The architecture with optimal hyperparameters is displayed in Figure 5. Table II shows the choices for each hyperparameter and the combination that produces the highest validation accuracy. Accuracy values of the top 10 combinations are included in Appendix 5b.

Using the best hyperparameters, an accuracy of 71.09% was achieved on the validation set. The confusion matrix and analysis of the probability assigned to the correct label are in Appendix 4.

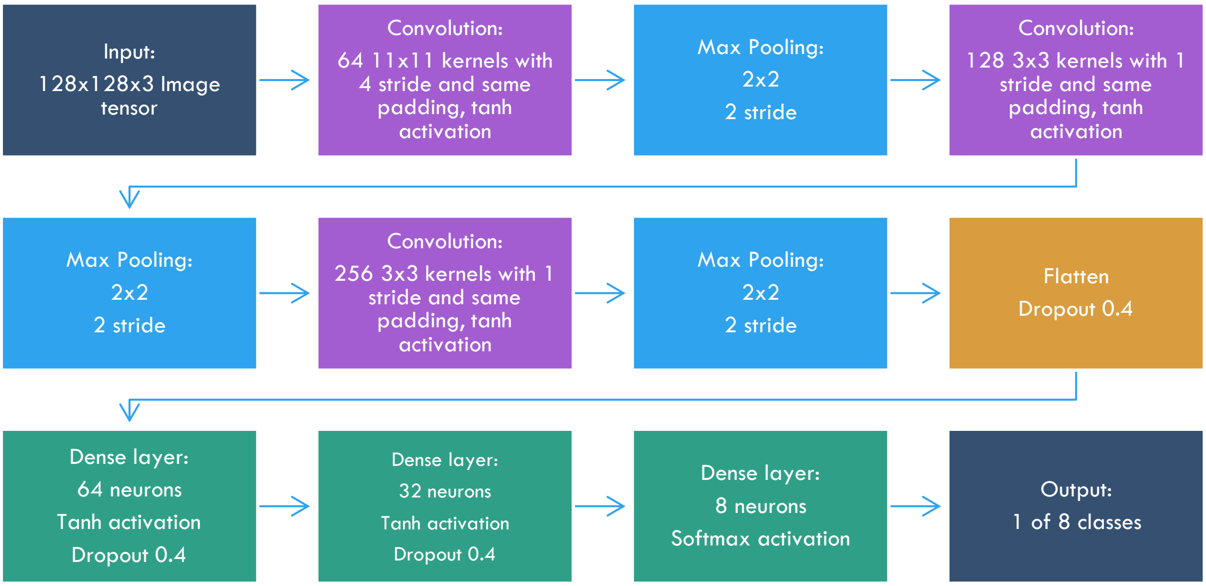


Figure 5: CNN architecture with best set of parameters found through grid search

TABLE II: Hyperparameters tuned using grid search for CNN

Hyperparameter	Choices	Optimal combination
Kernel size of 1st Convolution	5, 7, 9, 11	11
Kernel size of 2nd Convolution	3, 5	3
Pooling type	Max, Average	Max
Dropout rate	0, 0.2, 0.4	0.4
Activation	ReLU, tanh	tanh

Comparison of the 3 models on the validation set

Table III compares the performance of the three models, with CNN clearly being the best.

TABLE III: Validation accuracy of the three models implemented

Model	Accuracy on validation set
Logistic Regression	46.88%
Simple NN	48.83%
CNN	71.09%

Evaluation on the test set

The final CNN model was built with the optimal hyperparameters and trained using the training set and validation set 1 combined, with early stopping performed by validation set 2. The model was evaluated on the test set, giving an accuracy of 67.81%. This is slightly lower than the validation accuracy despite the increased training set size, as the hyperparameters were fitted specifically on the validation set. The confusion matrix is displayed in Figure 6 and the histogram of the probability estimates for the correct class is shown in Figure 7. Some examples of images in the test set and their predictions are included in Appendix 6.

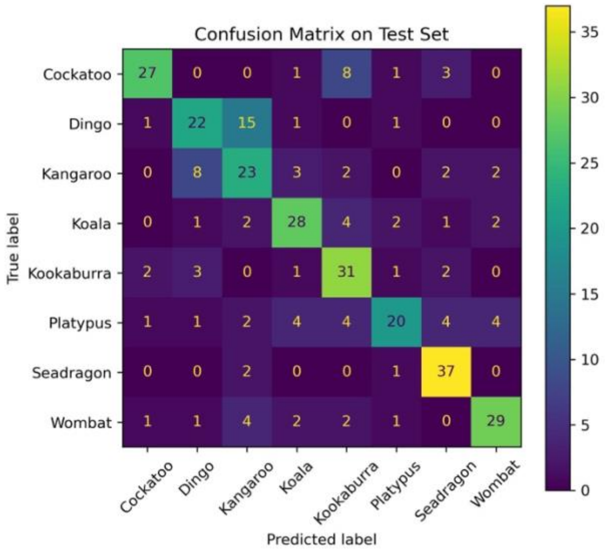


Figure 6: Confusion matrix of CNN on the test set. The model performs well overall but struggles to identify the platypus and distinguish between dingoes and kangaroos.

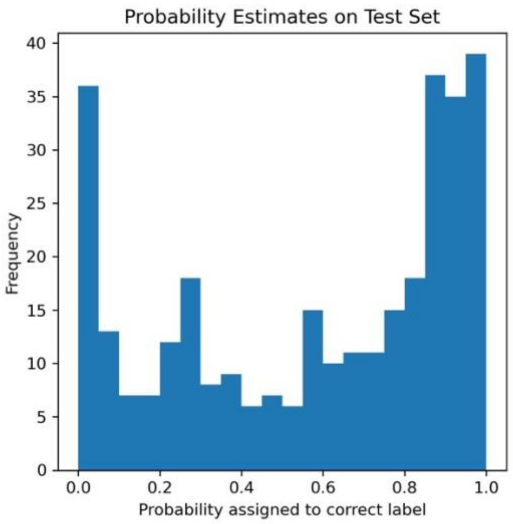


Figure 7: Histogram of probability assigned to the correct label for CNN. Note the two extremes: it has a lot of confidence both when correct and wrong.

Ethical considerations

While this specific task of classifying animals has minimal concerns regarding ethics, computer vision in general faces ethical challenges such as security and social biases.

Computer vision can potentially be misused to compromise security measures. For example, many websites rely on CAPTCHA to verify that the user is a human by providing several images and asking them to select ones with a certain object in it. However, many studies have shown that AI can easily crack CAPTCHA if given a good dataset to train on.[4,5] This may result in an influx of fake accounts, which could be used to commit fraud or manipulate crowd sentiment.

Moreover, computer vision can propagate social biases. A study on facial recognition has found that false positive matches are more likely to occur amongst females and black people than white males.[6] This has particularly serious impacts on policing, where individuals have been falsely arrested when their face was matched to an image or video at the crime scene.[7]

Conclusion

Deep learning is widely used in image classification tasks today. This project explored the use of a benchmark logistic regression and two different neural network architectures to perform image classification of Australian animals. Hyperparameter optimisation, which was conducted through grid search, was vital to find a relatively effective version of each architecture. The optimal CNN easily outperformed the other two models and achieved reasonable accuracy on the final test set.

References

1. Image Downloader Chrome extension: https://chrome.google.com/webstore/detail/image-downloader/cnpniohnfphhjihaiiggeabnkjhpaldj

2. LeCun Y, Boser B, Denker JS, et al. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*. 1989;1(4):541-551. doi:10.1162/neco.1989.1.4.541

3. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2012;60(6):84-90. doi:10.1145/3065386

4. Lorenzi D, Vaidya J, Uzun E, Sural S, Atluri V. Attacking Image Based CAPTCHAs Using Image Recognition Techniques. *Information Systems Security*. Published online 2012:327-342. doi:10.1007/978-3-642-35130-3_23

5. George D, Lehrach W, Kansky K, et al. A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs. *Science*. 2017;358(6368). doi:10.1126/science.aag2612

6. Chambers L. Five Fast Facts from the Federal Study of Demographic Bias in Facial Recognition. Published February 3, 2020. Accessed July 27, 2022. https://privacysos.org/blog/five-fast-facts-from-the-federal-study-of-demographic-bias-in-facial-recognition/

7. Fussell S. A Flawed Facial-Recognition System Sent This Man to Jail. Wired. Published June 24, 2020. Accessed July 27, 2022. https://www.wired.com/story/flawed-facial-recognition-system-sent-man-jail/

Appendix

Appendix 1: Reasoning for 2 validation sets

Some points from here: https://stats.stackexchange.com/questions/422671/early-stopping-together-with-hyperparameter-tuning-in-neural-networks

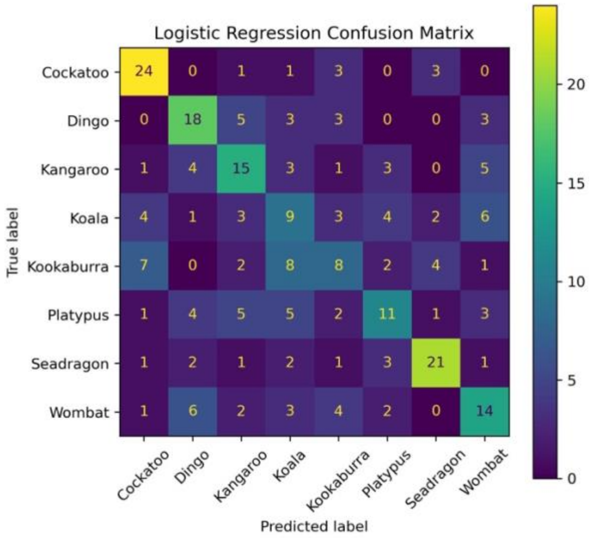
The benefit of having 2 different validation sets is that using the same dataset for both early-stopping and hyperparameter tuning would mean choosing the epoch with the best performance on the dataset, while also evaluating model performance using that same dataset. This may result in an optimistic validation error.

The downside of having two validation sets is less training data for hyperparameter tuning. This disadvantage is not present for the final model however, as one of the validation sets can be combined into the training set at that point.

After considering the pros and cons, the choice was made to use 2 validation sets.

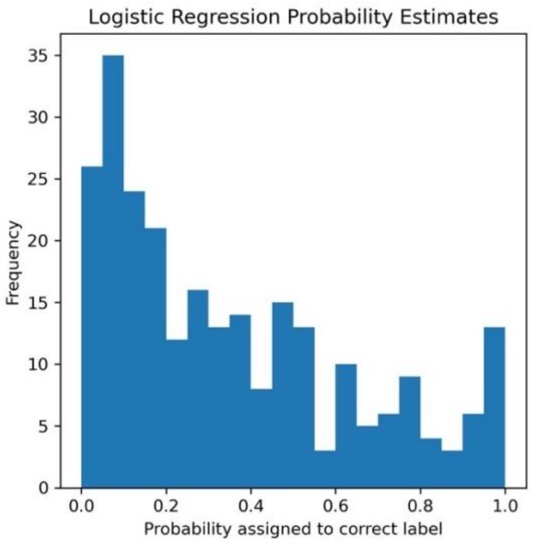
Appendix 2: Additional plots for logistic regression on validation set

2a. Confusion matrix



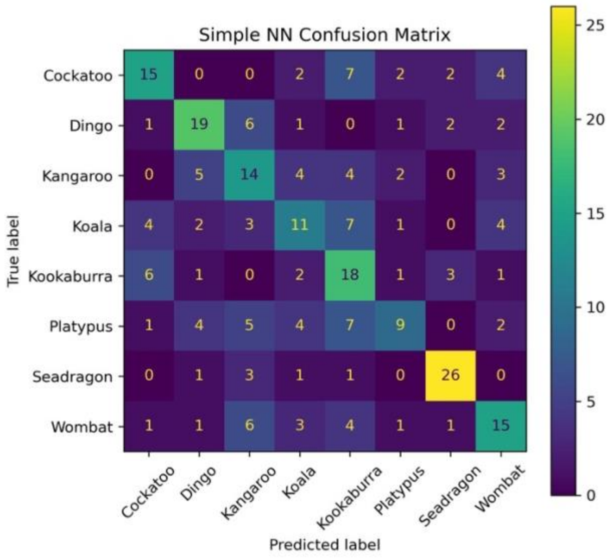
True label \ Predicted label	Cockatoo	Dingo	Kangaroo	Koala	Kookaburra	Platypus	Seadragon	Wombat
Cockatoo	24	0	1	1	3	0	3	0
Dingo	0	18	5	3	3	0	0	3
Kangaroo	1	4	15	3	1	3	0	5
Koala	4	1	3	9	3	4	2	6
Kookaburra	7	0	2	8	8	2	4	1
Platypus	1	4	5	5	2	11	1	3
Seadragon	1	2	1	2	1	3	21	1
Wombat	1	6	2	3	4	2	0	14

2b. Histogram of probability estimates

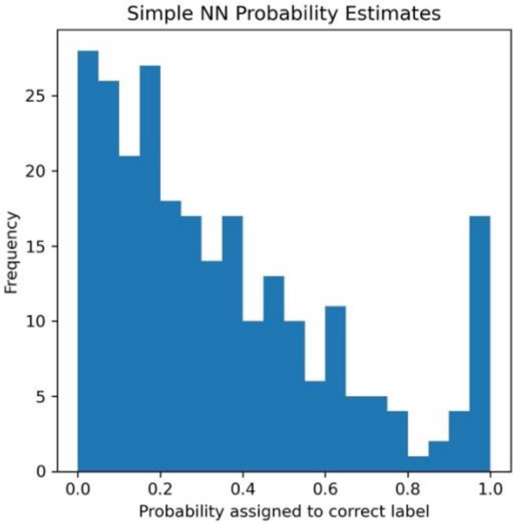


Appendix 3: Additional plots for simple neural network on validation set

3a. Confusion matrix

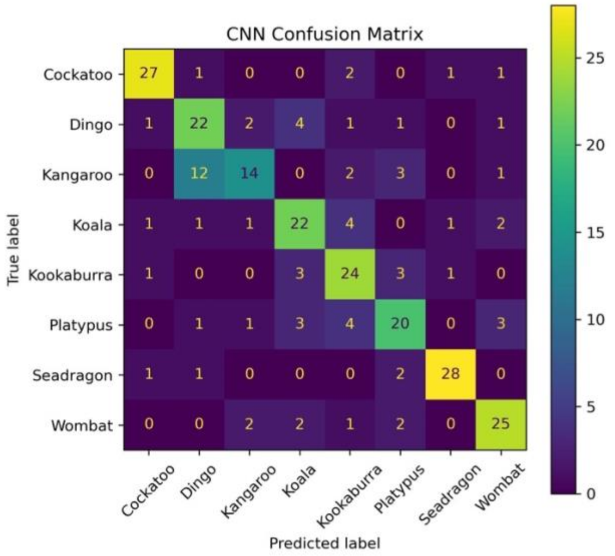


3b. Histogram of probability estimates

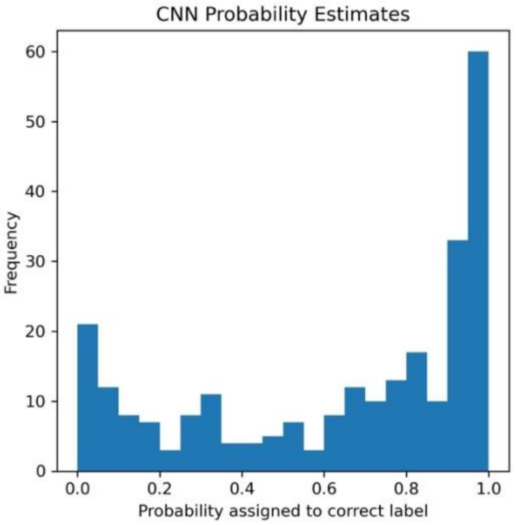


Appendix 4: Additional plots for convolutional neural network on validation set

4a. Confusion matrix



4b. Histogram of probability estimates



Appendix 5: Detailed grid search results

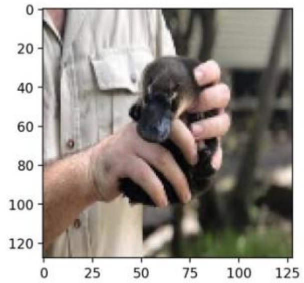
5a. Top 10 hyperparameter combinations for simple NN on validation set

Resizing size	Neurons in layer 1	Neurons in layer 2	Neurons in layer 3	Dropout rate	Activation	Validation loss	Validation accuracy
64	1024	512	no 3rd layer	0	relu	1.537749	0.496094
16	1024	512	128	0	tanh	1.542491	0.488281
16	1024	256	no 3rd layer	0	tanh	1.478022	0.488281
16	1024	256	no 3rd layer	0.2	relu	1.487772	0.488281
16	512	256	no 3rd layer	0	tanh	1.425003	0.488281
16	512	512	128	0.4	tanh	1.445847	0.484375
16	1024	512	no 3rd layer	0.2	tanh	1.458580	0.484375
16	1024	256	128	0	relu	1.557373	0.480469
16	512	256	128	0	tanh	1.495917	0.480469
16	512	256	no 3rd layer	0.2	tanh	1.411378	0.480469

5b. Top 10 hyperparameter combinations for CNN on validation set

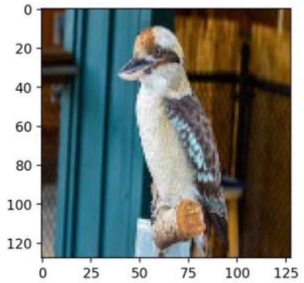
Kernel size of 1st layer	Kernel size of 2nd layer	Pooling	Dropout rate	Activation	Validation loss	Validation accuracy
11	3	max	0.4	tanh	0.848450	0.710938
11	3	max	0.2	tanh	0.902371	0.703125
7	5	max	0.2	tanh	0.944306	0.695313
5	5	max	0	tanh	0.938630	0.695313
5	5	max	0.4	tanh	1.002481	0.695313
7	3	max	0	tanh	0.954626	0.687500
9	3	max	0.2	tanh	0.898830	0.687500
5	3	max	0.2	tanh	0.924825	0.679688
7	3	max	0.4	tanh	0.960252	0.679688
5	3	max	0	tanh	0.959523	0.675781

Appendix 6: Ten examples of test images and their predictions



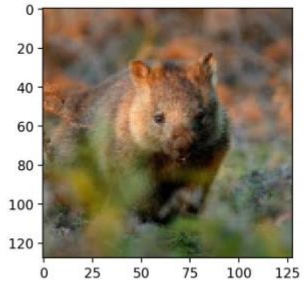
```
54.4% Kookaburra
25.9% Platypus
6.6% Koala
5.3% Dingo
4.8% Kangaroo
2.6% Wombat
0.3% Cockatoo
0.2% Seadragon

Correct label: Platypus
```



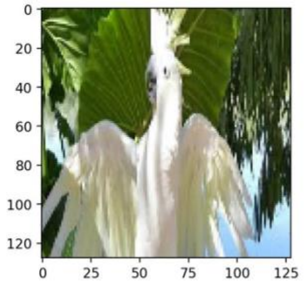
```
90.7% Seadragon
4.0% Cockatoo
3.4% Kookaburra
0.9% Dingo
0.6% Kangaroo
0.2% Koala
0.2% Platypus
0.1% Wombat

Correct label: Kookaburra
```



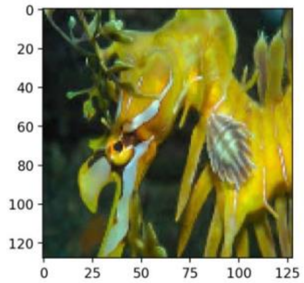
```
77.0% Wombat
11.1% Platypus
6.9% Kangaroo
3.1% Dingo
0.9% Kookaburra
0.8% Koala
0.2% Seadragon
0.1% Cockatoo

Correct label: Wombat
```



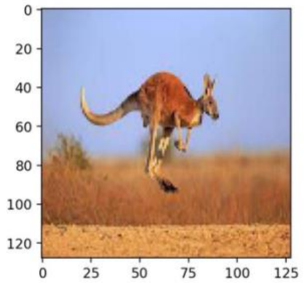
```
86.1% Cockatoo
7.9% Kookaburra
2.8% Dingo
1.6% Seadragon
0.8% Kangaroo
0.5% Platypus
0.3% Koala
0.0% Wombat

Correct label: Cockatoo
```



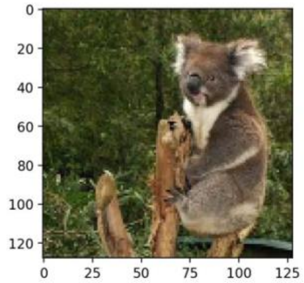
```
66.7% Seadragon
17.4% Dingo
9.6% Kangaroo
3.7% Cockatoo
1.2% Platypus
1.0% Kookaburra
0.2% Koala
0.1% Wombat

Correct label: Seadragon
```



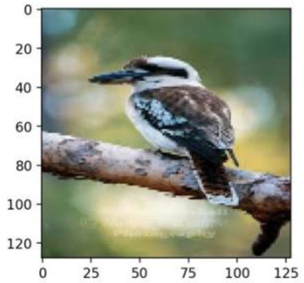
```
85.6% Kangaroo
10.2% Dingo
1.9% Kookaburra
0.9% Platypus
0.5% Cockatoo
0.4% Seadragon
0.3% Wombat
0.1% Koala

Correct label: Kangaroo
```



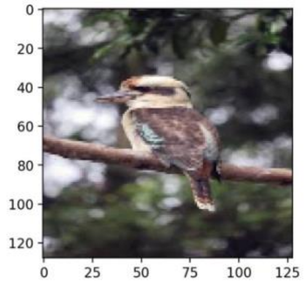
```
29.0% Koala
22.7% Kookaburra
22.5% Dingo
16.2% Kangaroo
4.4% Platypus
3.2% Wombat
1.7% Cockatoo
0.3% Seadragon

Correct label: Koala
```



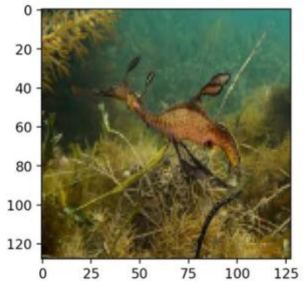
```
93.7% Kookaburra
4.3% Platypus
0.9% Cockatoo
0.4% Kangaroo
0.2% Dingo
0.2% Wombat
0.2% Koala
0.2% Seadragon

Correct label: Kookaburra
```



```
70.9% Platypus
9.0% Koala
7.8% Seadragon
4.9% Kangaroo
3.9% Dingo
1.5% Kookaburra
1.5% Wombat
0.5% Cockatoo

Correct label: Kookaburra
```



```
63.3% Seadragon
20.8% Kangaroo
8.6% Platypus
3.9% Wombat
2.0% Dingo
1.0% Kookaburra
0.4% Koala
0.2% Cockatoo

Correct label: Seadragon
```