Introduction:

Our vision to revolutionise animal classification with AI is now on the brink of reality. As requested, we have developed a prototype that can accurately classify all current and extinct animals to an accuracy of **85.76%** and is optimised to run efficiently on any device through **0.03 GFLOPS**. This report will outline the dataset of animal images we meticulously crafted to ensure a diverse collection, as well as highlight the initial baseline model. An outline of the series of advanced deep learning techniques applied to the baseline model will be explored through an ablation study to demonstrate the impact of each modification. Finally, rigorous evaluation was used to determine the optimal balance between accuracy and efficiency, due to different models excelling in their respective areas.

Animal Dataset:

The database assembled for training contained 151 animal classes for classification comprising a total of 6270 images of animals. Each of these images were of an RGB format with a resolution of 224 x 224 pixels. Of the total, 85% or 5330 images were used for training, 5% or 313 images for validation and backpropagation adjustments and 10% or 627 images were used for testing the final validation score. This results in the model training on an average of 35 images for each animal classification. To avoid the animal dataset becoming too large for training and to improve generalisation of the model, each animal category contained a mix of noise from the environment, hidden body parts, multiple of the same animal or a blank background of the animal. This is seen for the Ailurus Fulgens (Red Panda) category in figure 1.

Figure 1: Example images of Red Panda in dataset.

Baseline Model:

The architecture of the baseline Convolutional Neural Network (CNN) was composed of a feature extractor and a classifier. The feature extractor was designed using four convolutional layers, with the first layer containing a 5x5 kernel and the rest using a 3x3 kernel. Each of these layers also utilised a bias term by default with Pytorch. With four layers, the initial layers are able to extract lower and intermediate level features such as edges, corners and textures of the animals, whereas the final layers are able to extract more abstract patterns such as shapes and even capturing a collection of body parts. Each layer of the CNN also employs a ReLU activation function paired with max pooling kernel size of 2. This increases the model's ability to fit complex mapping data from input to output as the RELU activation function introduces non-linear feature mapping. Additionally, RELU eliminates gradient vanishing, which occurs during back propagation when the gradient becomes too small due to the activation function and slows the learning rate of the model. RELU negates this through setting the gradient to 1 for positive inputs. Each layer also employs max pooling to reduce the total dimensions of the feature map, decreasing the computational load on subsequent layers, as well as extracting the most important feature within the 2x2 kernel, potentially reducing the noise within the input. While this does discard information, increasing computational efficiency and reducing noise is more valuable to the CNN. The classifier was a combination of fully connected layers, flattening out the feature map from the final convolutional layer and a log softmax activation function, converting the output from logits to log probabilities.

Performance Enhancements:

A series of optimising machine learning parameters and techniques to the baseline model were conducted using an ablation study. The effect on accuracy and efficiency were tracked and outlined below, with the summarised ablation table shown in table 1. Prior to any improvements, the baseline model was able to achieve an accuracy of 37.45% and an efficiency of 0.69 GFLOPS.

As table 1 highlights, data augmentation slightly reduced accuracy, however was able to improve the generalisation ability of the model through drastically reducing the test loss. This lead to the idea of increasing the number of epochs to allow more backpropagation to occur and allow the model to generalise better. This change led to a 3.5% improvement, however, did not improve the model as drastically as desired. This suggested a greater change to the model was needed, and as such, the optimisation function was changed from Adam to RAdam. This would potentially address any issues with unstable training with Adam due to high learning rates for initial training tests through introducing the rectified learning rate. This change seemed to work, with the model increasing in accuracy by 5.5%. The learning rate was then altered, and saw a minor increase of less than 1%. This led to the conclusion that hyperparameters provided insufficient benefit. Hence, the activation function was changed from a Rectified Linear Unit (ReLU) to an Exponential Linear Unit (ELU), whose functions are seen below:

$$ReLU(x) = max(0, x) \quad , \quad ELU(x) = \{x, \; if \; x > 0 \quad or \quad \alpha(e^{x} - 1), \; if \; x \leq 0\}$$

Switching to ELU eliminates the "dying RELU" problem, which occurs when neurons operating with ReLU become inactive because any negative value is set to 0. Since ELU sets these negative values to positive, the learning ability of neutrons

were maintained and could lead to better learning. As table 1 shows, a 3.6% increase to accuracy was observed which was expected however, was still not enough of an improvement. This finally led to transfer learning, where pre-trained models on large data sets could be used and fine tuned for our animal classification model through training on our dataset. An established image classification model was VGG16, containing 16 weight layers, 13 convolutional layers and 3 fully connected layers. This resulted in a 32.2% increase, the greatest increase to accuracy from all modifications. The sheer scale of the model however heavily decreased the efficiency of the model to 7.89 GFLOPS, proving it unusable for our application.

Table 1: Ablation study of modifications made to baseline model and their result in each Test (T'x'). DA: Data Augmentation. OF: Optimisation Function. LR: Learning Rate. AF: Actuation Function. TFL: Transfer Learning. TL: Test Loss. Acc.: Accuracy.

Models	DA	Epochs	OF	LR	AF	TFL	TL	Acc. (%)	GFLOPs
Base	✖	10	Adam	0.001	RELU	✖	5.016	37.45	0.69
T1	✔	10	Adam	0.001	RELU	✖	3.689	36.61	0.69
T2	✔	20	Adam	0.001	RELU	✖	3.893	39.11	0.69
T3	✔	20	RAdam	0.001	RELU	✖	4.241	44.58	0.69
T4	✔	20	RAdam	0.0001	RELU	✖	3.705	45.21	0.69
T5	✔	20	RAdam	0.0001	ELU	✖	3.429	48.80	0.69
T6	✔	20	RAdam	0.0001	ELU	✔	1.177	85.00	7.89

Optimising Efficiency:

While transfer learning cemented itself as the best method for optimising animal recognition accuracy, it drastically decreased the efficiency of the model. Since the model is to be run efficiently on all devices as requested, we believed trialling more efficient transfer learning models could strike the balance between the high accuracy observed with VGG16, and also efficiency. In total, 6 pre-trained models were evaluated with the same adjustments made to the hyper-parameters in Performance Enhancements to maintain uniformity in evaluation. Additionally, to evaluate which model compromises between accuracy and cost the best, a score scaling from 1 to 10 was made for how much of a percent gain was made for accuracy and efficiency from the baseline model. If a negative gain was observed, this was set to 0. These two scores would then be given equal weighting as both metrics are equally important, and a final total score out of 20 was calculated. Using this scoring criteria, the below table 2 was made evaluating each transfer learning model.

Table 2: Transfer learning scoring comparison. ENet v2: Efficiency Net Version 2. ENet b0: Efficiency Net b0. MN v3 L: Mobile Net Version 3 Large. MN v3 S: Mobile Net Version 3 Small.

Scoring	VGG16	ResNet	ENet v2	ENet b0	MN v3 L	MN v3 S
Accuracy (%)	85.00	81.99	93.44	91.30	89.72	85.76
Accuracy Gain (%)	76	71	90	86	84	77
GFLOPs	7.89	2.15	1.51	0.21	0.12	0.03
Efficiency Gain (%)	-1043	-212	-119	70	83	96
Total Score ([])/20)	7.60	7.12	8.95	15.57	16.62	17.29

The table clearly demonstrates MobileNet version 3 Small as the most effective pre-trained model when accuracy and efficiency improvement were given equal weighting. It is for this reason why this pre-trained model was chosen for our animal detection model.

Limitation and Conclusion:

After these different tests we found that using transfer learning produced the best results. Based on the architecture of the model we can pick what we want to optimise for. If the user case is on an embedded system we would aim for the model with the least amount of processing required so we would use mobile net, and if we were wanting more accuracy we would gear more towards efficiency net. There are limitations to all the models we are never going to be able to get 100% accuracy as there are edge cases that we aren't able to handle. Though we tried to negate the effects of overfitting there is also a non zero chance that the models with the highest accuracy are over fitting but that is unlikely. Overall, using transfer learning and training with the dataset that we had and tweaking different parts of the model produced a fairly accurate model that would be able to be used for animal classification.