

Mobile application for dog breed classification using deep learning and transfer learning

Siu Kwan YAU

MSc in Machine Learning and Autonomous Systems

The University of Bath

2021-2022

This dissertation may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Mobile application for dog breed classification using deep learning and transfer learning

Submitted by: Siu Kwan YAU

Copyright

Attention is drawn to the fact that copyright of this dissertation rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the author unless otherwise specified below, in accordance with the University of Bath's policy on intellectual property (see

https://www.bath.ac.uk/publications/university-ordinances/attachments/Ordinances_1_October_2020.pdf).

This copy of the dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the dissertation and no information derived from it may be published without the prior written consent of the author.

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of MSc Machine Learning and Autonomous Systems in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

This project seeks to develop a dog breed classification iOS application using techniques of deep learning and transfer learning. A total of 11 pre-trained CNN and transformer models are fine-tuned on the Stanford Dogs dataset and compared based on the criterion of accuracy, inference speed and model size. The final model uses EfficientNetV2-B3 as its backbone architecture and achieves a Top-1 accuracy of 91.94% and a Top-5 accuracy of 99.32%, which is close to the state-of-the-art results achieved by much larger models in literatures. Apart the dog breed classification model, a dog detector model is also developed as data validator for the application. The model leverages MobileNetV2 as backbone which has a lightweight structure. The model is trained with a dataset formed by a combination of Stanford Dogs and Caltech-256 datasets. The final detector model achieves a satisfactory accuracy of 99.54%. Thorough inspections on the misclassified samples of both models are performed with the aid of CNN model visualization techniques including Grad-CAM and Activation Maximisation. The final mobile application significantly surpasses other current market-leading products on AppStore in terms of accuracy, storage requirements and inference time. A demo of the final application is available at <https://vimeo.com/745855045>.

With a large variety of CNN and transformer models tried and tested in this project and summarisation of state-of-the-art results achieved in literatures, this project can not only serve as foundation for future deep learning work on the dog breed classification problem, but also as a reference to works on fine-grained classification problems in general.

Contents

CONTENTS.....	I
LIST OF FIGURES	III
LIST OF TABLES	V
ACKNOWLEDGEMENTS.....	VI
1 INTRODUCTION	1
2 LITERATURE AND TECHNOLOGY SURVEY.....	3
2.1 RELATED STUDIES	3
2.2 RELATED DATASETS	10
2.3 RELATED PRODUCTS.....	11
3 DESIGN	14
3.1 OBJECTIVES	14
3.2 COMPONENTS	15
3.3 EVALUATION CRITERION	16
4 METHODOLOGY	16
4.1 DEEP LEARNING MODELS.....	16
5 RESULTS	22
5.1 DEEP LEARNING MODELS.....	22
5.2 IOS MOBILE APPLICATION	37
6 CONCLUSION AND FUTURE WORK.....	38
7 BIBLIOGRAPHY	40
APPENDIX	49
APPENDIX I: RESULTS OF EXPERIMENT ON THE EXISTING MOBILE APPLICATIONS	49

APPENDIX II: SCREENSHOTS OF RESULT PAGES OF THE EXISTING MOBILE APPLICATIONS	50
APPENDIX III: (DOG DETECTOR MODEL) LIST OF LAND ANIMALS KEPT OR REMOVED IN THE FINAL DATASET	53
APPENDIX IV: (DOG DETECTOR MODEL) ORIGINAL IMAGES AND GRAD-CAM VISUALISATIONS OF FALSE POSITIVE PREDICTIONS.....	56
APPENDIX V: (BREED FGIC MODEL) FEATURE EXTRACTION TRAINING HISTORY OF THE 11 MODELS EXPLORED	57
APPENDIX VI: (BREED FGIC MODEL) FINE-TUNING TRAINING HISTORY OF THE 11 MODELS EXPLORED.....	60
APPENDIX VII: (BREED FGIC MODEL) CONFUSION MATRIX OF THE FINAL MODEL ON TEST SET.....	63

List of Figures

FIGURE 1-1: DOG BREEDS WITH LOW INTER-CLASS VARIANCE IN THE STANFORD DOGS DATASET	2
FIGURE 1-2: POMERANIAN WITH HIGH INTRACLAS VARIANCE IN THE STANFORD DOGS DATASET	2
FIGURE 2-1. GENERAL STRUCTURE OF A CNN.....	5
FIGURE 2-2: ILLUSTRATION OF USING AN ATTENTION-BASED TWO-STAGE ARCHITECTURE FOR THE FINE-GRAINED DOG BREED CLASSIFICATION PROBLEM.	6
FIGURE 2-3: COMPARISON BETWEEN TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING.....	8
FIGURE 2-4: ILLUSTRATION OF A GENERAL KD TEACHER-STUDENT ARCHITECTURE.....	9
FIGURE 2-5 EXAMPLES OF NOISY IMAGES IN THE STANFORD DOGS DATASET	11
FIGURE 2-6: PERFORMANCES OF THE DOG BREED IDENTIFICATION MOBILE APPLICATIONS (DOGSCANNER, DOG PAL AND DOG ID) IN THE EXPERIMENT.	12
FIGURE 3-1: UI DESIGN OF THE IOS APPLICATION.....	14
FIGURE 3-2: HIGH-LEVEL OVERVIEW ON THE END-TO-END PROCESS FLOW OF A PREDICTION REQUEST.	14
FIGURE 4-1: (DOG DETECTOR MODEL) EXAMPLE OF IMAGES WHERE APPLE’S API GIVES FALSE NEGATIVE RESULTS.....	17
FIGURE 4-2: (DOG DETECTOR MODEL) DISTRIBUTION OF SAMPLES IN TRAINING, VALIDATION AND TESTING SETS RESPECTIVELY	18
FIGURE 4-3: (DOG DETECTOR MODEL) ARCHITECTURE OF THE DOG DETECTION MODEL.....	18
FIGURE 4-4: (BREED FGIC MODEL) DISTRIBUTION OF SAMPLES IN TRAINING, VALIDATION AND TESTING SETS RESPECTIVELY	19
FIGURE 4-5: (MOBILE APP) FRONTEND-BACKEND COMMUNICATION FOR AN IMAGE CLASSIFICATION REQUEST IN THE MOBILE APPLICATION.	21
FIGURE 5-1: (DOG DETECTOR MODEL) TOP-1 ACCURACY DURING FEATURE EXTRACTION AND FINE-TUNING PHASE.....	22
FIGURE 5-2: (DOG DETECTOR MODEL) CONFUSION MATRIX.....	23
FIGURE 5-3: (DOG DETECTOR MODEL) ORIGINAL IMAGE AND GRAD-CAM OF THE FALSE NEGATIVE PREDICTIONS	23
FIGURE 5-4: (DOG DETECTOR MODEL) NEW MESSAGES IMPLEMENTED IN THE APPLICATION INSPIRED FROM FALSE NEGATIVE SAMPLES ANALYSIS	25
FIGURE 5-5: (DOG DETECTOR MODEL) FALSE POSITIVE SAMPLE LIKELY CAUSED BY THE LACK OF VARIETY OF TRAINING SAMPLES.....	26
FIGURE 5-6: (DOG DETECTOR MODEL) FALSE POSITIVE SAMPLE LIKELY CAUSED BY THE LACK OF QUANTITY OF TRAINING SAMPLES.....	26

FIGURE 5-7: (DOG DETECTOR MODEL) FALSE POSITIVE SAMPLE LIKELY CAUSED BY THE LACK OF ANIMAL TRAINING SAMPLES.....	27
FIGURE 5-8: (BREED FGIC MODEL) PERCENTAGE IMPROVEMENT IN TOP-1 ACCURACY OF THE 11 MODELS DURING FEATURE EXTRACTION.....	28
FIGURE 5-9: (BREED FGIC MODEL) HISTORY OF THE TRAINING AND VALIDATION TOP-1 ACCURACIES OF THE 11 MODELS DURING FEATURE EXTRACTION	28
FIGURE 5-10: (BREED FGIC MODEL) PERCENTAGE IMPROVEMENT IN TOP-1 ACCURACY OF THE 11 MODELS DURING FINE-TUNING.....	29
FIGURE 5-11: (BREED FGIC MODEL) TRAINING HISTORY OF THE 11 MODELS DURING FINE-TUNING	29
FIGURE 5-12: (BREED FGIC MODEL) TRAINING, VALIDATION AND TESTING TOP-1 ACCURACIES OF THE BEST MODEL FOR EACH OF THE 11 ARCHITECTURES.....	30
FIGURE 5-13: (BREED FGIC MODEL) TOP-1 ACCURACY, SIZE AND INFERENCE TIME OF THE 11 MODELS	30
FIGURE 5-14: (MOBILE APP) PERFORMANCE OF THE APPLICATION PRODUCED IN THIS PROJECT AGAINST MARKET LEADING PRODUCTS	37
FIGURE A1: IMAGES UPLOADED FOR THE EXPERIMENT AND THE CORRESPONDING CLASSIFICATION RESULTS OF THE THREE APPLICATIONS TESTED.....	49
FIGURE A2: RESULT PAGES OF DOGSCANNER.	50
FIGURE A3: RESULT PAGES OF DOG PAL.	51
FIGURE A4: RESULT PAGES OF DOG ID.	52
FIGURE A5: (DOG DETECTOR MODEL) ORIGINAL IMAGES AND GRAD-CAM VISUALISATIONS OF FALSE POSITIVE SAMPLES	56
FIGURE A6: (BREED FGIC MODEL) COMPARISON OF THE TOP-1 AND TOP5 TRAINING AND VALIDATION ACCURACIES ON THE 11 MODELS DURING FEATURE EXTRACTION	57
FIGURE A7: (BREED FGIC MODEL) HISTORY OF TRAINING AND VALIDATION ACCURACIES OF THE 11 MODELS RESPECTIVELY DURING FEATURE EXTRACTION	59
FIGURE A8: (BREED FGIC MODEL) COMPARISON OF THE TOP-1 AND TOP5 TRAINING AND VALIDATION ACCURACIES ON THE 11 MODELS DURING FINE-TUNING	60
FIGURE A9: (BREED FGIC MODEL) HISTORY OF TRAINING AND VALIDATION ACCURACIES OF THE 11 MODELS RESPECTIVELY DURING FINE-TUNING.....	62
FIGURE A10: (BREED FGIC MODEL) CONFUSION MATRIX OF THE FINAL MODEL ON TEST SET	63

List of Tables

TABLE 2-1: CATEGORISATION OF TRANSFER LEARNING METHODS BASED ON SIMILARITY OF SOURCE AND TARGET TASKS AND DOMAINS.	8
TABLE 2-2: SUMMARY OF CHARACTERISTICS OF THE TSINGHUA DOGS AND STANFORD DOGS DATASET.	10
TABLE 2-3: DOG BREED CLASSIFICATION MODEL PERFORMANCES ON THE STANFORD DOGS DATASET IN LITERATURES... ..	11
TABLE 2-4: SUMMARY OF THE FEATURES OF THE THREE FREE DOG BREED CLASSIFICATION APPLICATION WITH MOST REVIEWS ON THE UK IOS APPSTORE.....	12
TABLE 4-1: (BREED FGIC MODEL) OVERVIEW OF MODEL ARCHITECTURES EXPLORED FOR THE DOG BREED CLASSIFICATION MODEL.....	20
TABLE 5-1: (BREED FGIC MODEL) EXPERIMENT SETUPS AND RESULTS DURING HYPERPARAMETER TUNING.....	32
TABLE 5-2: (BREED FGIC MODEL) COMPARISON OF RESULTS OF THE FINAL MODEL AND RESULTS IN LITERATURES	32
TABLE 5-3: (BREED FGIC MODEL) TOP 5 MOST CONFUSED BREED PAIRS FOR THE FINAL MODEL	33
TABLE 5-4: (BREED FGIC MODEL) STATISTICS ON THE TOP 5 MOST MISCLASSIFIED DOG BREEDS	34
TABLE 5-5: (BREED FGIC MODEL) OVERVIEW OF THE WORST PREDICTIONS MADE BY THE FINAL MODEL	36
TABLE A1: (DOG DETECTOR MODEL) LIST OF LAND ANIMALS IN THE STANFORD DOGS DATASET.....	53
TABLE A2: (BREED FGIC MODEL) LIST OF BREEDS	64

Acknowledgements

I would like to express my deepest appreciation to Dr. Hongping Cai for her invaluable feedback and support throughout this dissertation project. The meetings and conversations with her were vital in helping me to understand the past and existing methodologies for computer vision problems and to shape a comprehensive analysis for the project. Special thanks also goes to my family and friends who have been a great source of emotional support through my studies.

1 Introduction

Knowing the breed of a dog is not only for satisfying the curiosity of dog owners, but also an important information that could determine the life-and-death of dogs. To dog owners and veterinarians, knowing the breed of dogs help them pay attention to breed-specific health issues and maintain the wellbeing of dogs; To shelters, misidentifying a dog's breed would not only affect the adoption rate of dogs, but might also cause unnecessary euthanasia if the dog is misclassified as legally restricted breeds such as pit-bull (Gunter, Barber and Wynne, 2018). The most accurate method of identifying a dog's breed is to conduct a canine genetic test, however, due to its costly and time-consuming nature, breed identification is usually determined by visual appearance. Human visual recognition of dog breeds has proven records of inconsistencies and errors. In a study conducted in the United States, even people with occupations closely related to dogs were only able to correctly identify the breed of a dog 27% of the time (Croy et al., 2012). The chances of error are even higher in the cases of identifying a mixed-breed dog (Gunter, Barber and Wynne, 2018).

Dog breed identification is a fine-grained image classification (FGIC) task, which mainly focuses on differentiating classes with low interclass variance and large intraclass variance. Figure 1-1 shows examples of dog breeds with similar visual appearance, but belong to different breeds. On the other hand, dogs with more visually distinctive features such as different shades of hair might be of the same breed, as shown in Figure 1-2. The similarity across breeds and the drastic difference within breeds imposes a great challenge to the visual recognition of dog breeds, both for human and computers. This project will approach the dog breed FGIC problem by making use of a large-size public dog breeds dataset, Stanford Dogs, which is a popular dataset for studying FGIC problems in general (Khosla et al., 2012).



Figure 1-1: Dog breeds with low inter-class variance in the Stanford Dogs dataset (adapted from Khosla et al., 2012)



Figure 1-2: Pomeranian with high intraclass variance in the Stanford Dogs dataset (adapted from Khosla et al., 2012)

Since the success of AlexNet in the ImageNet Large Scale Visual Recognition Challenge (Krizhevsky, Sutskever and Hinton, 2017), Convolutional Neural Network (CNN) has been the dominating technique for image classification tasks, and there are plenty of research work on the fine-grained dog breed classification problem using CNN (Jain et al., 2020; LaRow, Mittl and Singh, 2016; Tu, Lai and Yanushkevich, 2018; Uno, Han and Chen, 2018). In recent years, new state-of-the-art result on the problem is achieved using visual transformer models and hybrid transformer models (Conde & Turgutlu, 2021; Do et al., 2022; He et al., 2022). A variety of CNN and transformer architectures with proven results in literatures will be tested and compared in this project.

Most of the research work on the dog breed FGIC problem leveraged the technique of transfer learning, which makes use of deep learning models pre-trained on problems in a similar domain, then fine-tuned using the target problem dataset. With the knowledge transferability of models

in similar domain being proven in previous research, this project will adopt the technique as it is a more efficient and effective approach given the time and resource constraints for the project.

To make the classification more accessible to the general public, the best performing deep learning model found during model exploration will be deployed as an iOS mobile application. Currently, there are several dog breed identification applications in the market, however, even for the more popular applications in the United Kingdom iOS AppStore, there are numerous reviews complaining about the result accuracy. Most commonly found dissatisfactory comments on dog breed classification applications are around the classification accuracy, while some are around situations when classification results are still given even when the image contains no dog. To cope with the latter, a dog detection model will also be trained as a data validation control for the application. Apart from accuracy, another important factor for deploying deep learning models in mobile devices is the requirement on computational resources. Deep learning models are usually computationally complex and requires extensive storage requirements, which pose challenges for deploying them in smartphones as they have relatively limited computational resource (Castanyer, Martínez-Fernández and Franch, 2021; Gou et al., 2021). Therefore, model size and inference time will also be consideration factors when selecting the final model for incorporation in the application. Three current market leading products will be used as benchmarks to evaluate the performance of the final application developed.

2 Literature and Technology Survey

2.1 Related studies

2.1.1 Fine-grained image classification (FGIC)

FGIC is a difficult problem to solve due to the presence of some high intra-class variances and low-interclass variances (Zhao et al., 2017). Zheng et al. (2018) coarsely categorised the FGIC methods into **traditional machine learning** and **deep learning methodologies**.

A. Traditional machine learning (ML) methodologies

Traditional ML methodologies usually involve three steps which are *features extraction*, *feature representation* and *classification*.

I. Feature detection and description

The first step in traditional ML image classification is to detect and describe interest points in images. The regions detected are usually corners, blobs, edges and curves. These detected regions are then described into high-dimensional features vectors that are highly distinctive and invariant to scale, rotation and affine changes using feature detector-describers include SIFT (Lowe, 1999), SURF (Bay et al., 2008), HOG (Dalal and Triggs, 2005), and KAZE (Alcantarilla et al., 2012).

II. Feature representation

After features that describe local regions in an image are extracted, a popular approach would be to represent them using the Bag-of-Visual-Words (BoVW) algorithm. BoVW represents features as a histogram of occurrences of patterns in an image, clusters the features into a collection of orderless local features using unsupervised learning algorithms such as K-means, then finally assigns weight to each set of features according to weighing schemes such as TF-IDF (Gao et al., 2015; Mansoori et al., 2013). The collection of features are often referred to as “visual vocabulary” or “codebook”. However, BoVW has a major drawback of disregarding spatial relationships of local features. Lazebnik et al. (2006) proposed a solution to the problem by extending the BoVW representation with Spatial Pyramid Matching (SPM), which partitions an image into increasingly fine sub-regions of lower resolution images and compute histograms of local features within each sub-region.

III. Classification

The bag-of-features extracted are then feed into traditional ML classifiers for fine-grained classification tasks. SVM is commonly chosen as the classifier for bag-of-feature based image classification tasks (Csurka et al., 2004; Sitaula and Aryal, 2021; Yang et al., 2012; Wang et al., 2016). Other choices of classifiers include Naïve Bayes (Csurka et al., 2004; Hsu, Chen and Huang, 2015), random forest (Bosch, Zisserman and Munoz, 2007; Yao, Khosla and Li, 2011), and logistic regression (de Campos, Csurka and Perronnin, 2012).

B. Deep learning methodologies

For the past decades, computer vision has leaped forward along with the popularity of deep learning models. CNN models have reached unprecedented achievements on large scale image classification tasks, such as ImageNet. Conventional CNN is capable of capturing feature vectors by sliding small convolutional kernels through the images, selecting features in fully connected layers and outputting final prediction (Figure 2-1), which eliminates the need of a three-step process in traditional ML methodologies. However, the limited receptive field of kernels lead to CNN's weakness in capturing global information. Recent works on visual transformer attempts to address this limitation in order to attain new state-of-the-art results.

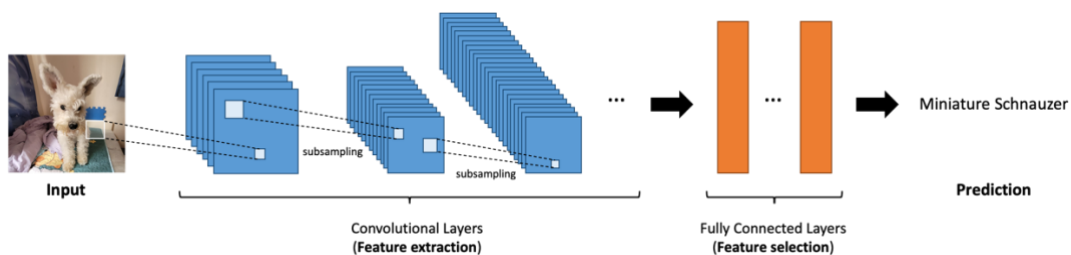


Figure 2-1. General structure of a CNN.

General CNN

CNN was first introduced by Lecun et al. (1998) for handwritten character recognition, it received limited adoption at the time due to the lack of computational resource and data required for training. Research interest in CNN only began to rocket after AlexNet achieved significantly better performance in the ImageNet Large Scale Visual Recognition Challenge than other participating classic models (Krizhevsky, Sutskever and Hinton, 2017). With the success of AlexNet and the advent of large-scale labelled data for training, innovations of various CNN architectures for computer vision tasks have been introduced (Khan et al., 2020; Zhao et al., 2017). Various data augmentation and enhancement strategies have also been introduced to address the issue of limited availability of data which hinders the performance of CNN models (Wang and Wang, 2019).

With the assumptions of locality and weight sharing, CNN uses small kernels for feature extraction, which leads to its inherent weakness in capturing long-distance relationships within an image. ResNet attempted to patch the limitation by creating shortcuts between layers and introducing global pooling, and has achieved better generalisation performance than previous works (He et al., 2015).

Most state-of-the-art CNN models can be adopted for FGIC tasks with their proven ability in yielding discriminative representations of images (Zhao et al., 2017). To further assist CNN models in focusing on more distinguishing features, recent works proposed the use of an attention-based two-stage architecture for object and object part detection and classification (Anwar, Barnes and Petersson, 2021; Xie, Li and Liu, 2018). The discriminative regions of image are first extracted by object detection frameworks such as R-CNN, bilinear CNN, CNN with Generative Adversarial Network (GAN) and YOLO, then passed to pre-trained state-of-the-art CNN models for classification (Lin, Roychowdhury and Maji, 2015; Xie, Li and Liu, 2018; Zhao et al., 2018).

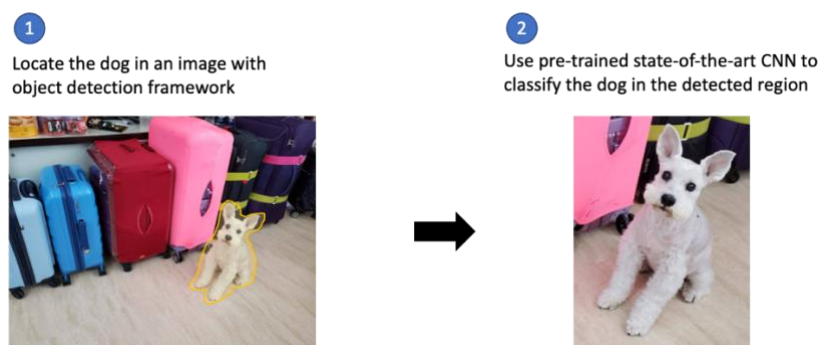


Figure 2-2: Illustration of using an attention-based two-stage architecture for the fine-grained dog breed classification problem.

Visual transformers

Inspired by transformer's successes in NLP, Dosovitskiy et al. (2020) presented the possibility of applying this self-attention-based architecture in the field of computer vision with their Vision Transformer (ViT) model, which in theory are more capable than CNN in capturing long distance relationships. Overall, ViT and other transformer-based models are computationally more expensive than CNN models, and lags behind similar-sized state-of-the-art CNN models. However, they outperform state-of-the-art CNN models when more training data becomes available (Dosovitskiy et al., 2020).

Hybrid of CNN and transformers

Researchers believe that the performance differences between ViT and CNN models can be explained by their level of inductive biases (Dai et al., 2021; d'Ascoli et al., 2021; Guo et al., 2021). The strong inductive bias of CNNs on locality allows them to reach high performance even with limited data, while visual transformers have minimal inductive bias which limits their ability in handling smaller-scale data, yet allow sufficient flexibility for them to learn better than CNNs on larger-scale data. ConViT attempts to combine the best of both structures

by introducing a self-attention layer that flexibly introduce the inductive bias of CNN into its transformer-based structure (d'Ascoli et al., 2021). CvT introduces the structures of convolutional token embedding and convolution transformer block to fuse desirable properties of CNN into the visual transformer structure (Wu et al., 2021). CoAtNet vertically stacks convolution and attention blocks to produce the hybrid model, and achieves the new state-of-the-art result on ImageNet by pretraining on 3 billion images from Google's internal JFT-3B dataset (Dai et al., 2021). All these attempts achieved similar or superior accuracies over their baseline CNN and visual transformer models with similar or smaller model sizes. However, as transformer models are heavy-weight models, most studies tend to compare the performance of transformer and hybrid transformer models with large CNN models with a large number of parameters, which might not be suitable for computationally limited mobile devices. In order to deploy the model on mobile devices with low latency, Apple researchers introduced MobileViT, which surpasses mobile-optimised CNN models in terms of accuracy, yet the model still falls significantly behind in terms of inference time (Mehta & Rastegari, 2021).

2.1.2 Transfer learning

With the popularity of CNN in computer vision tasks, there have been numerous studies on dog breed classification using CNN (Imran and Athitsos, 2020; Jain et al., 2020; LaRow, Mittl and Singh, 2016; Tu, Lai and Yanushkevich, 2018; Uno, Han and Chen, 2018), and a considerable amount of them leverage the technique of transfer learning. Unlike traditional machine learning methods where separate models are trained with different tasks, transfer learning allows knowledge learned from previous tasks to be transferred to other tasks in same or related domains (Pan and Yang, 2010) (Figure 2-3). For instance, it has been popular for researchers to adapt models pretrained on ImageNet to fine-grained image datasets (Zhao et al., 2017). The use of transfer learning alleviates problems such as insufficient training data and high training overhead, and sometimes result in better performance than traditional machine learning methods (Liu et al., 2019). Pang and Yang (2010) summarised the different approaches of transfer learning into three major branches, based on the similarity of the source and target tasks and domains (Table 2-1).

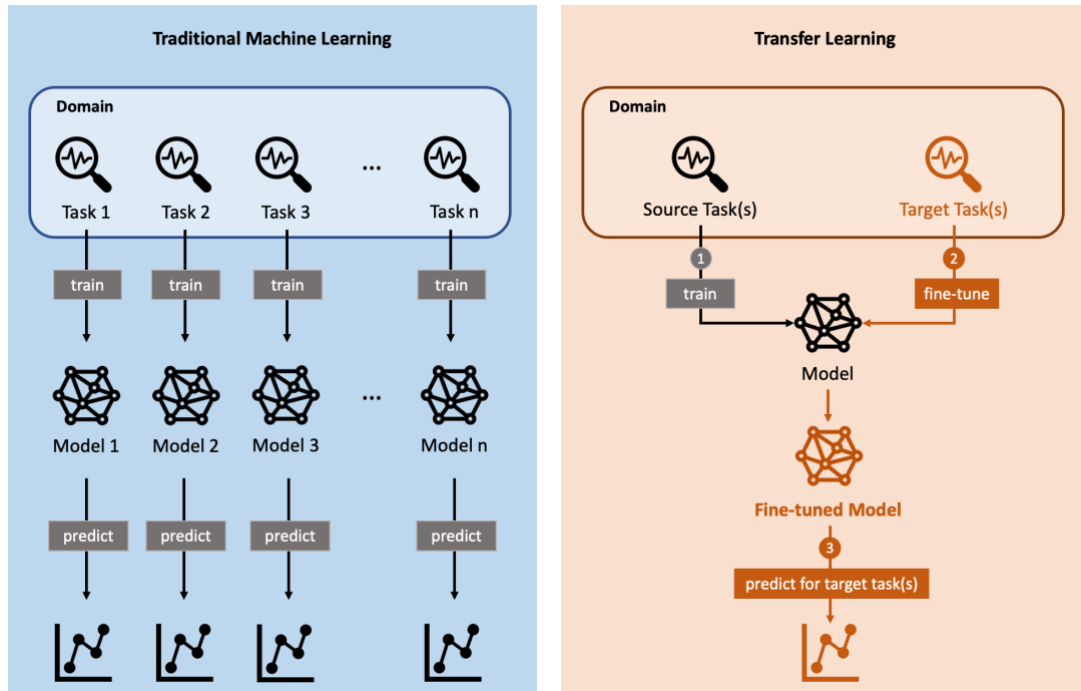


Figure 2-3: Comparison between traditional machine learning and transfer learning. Traditional machine learning trains separate model for different tasks. In transfer learning, model trained on tasks in same or similar domain is fine-tuned on the new target task and used for prediction in the new tasks.

Table 2-1: Categorisation of transfer learning methods by Pan and Yang (2010) based on similarity of source and target tasks and domains.

Transfer Learning Category	Similarity of		Availability of	
	Source and Target Domains	Source and Target Tasks	Labelled Source Data	Labelled Target Data
Inductive	Same / Different but related	Different but related	<u>Yes:</u> Multi-task learning	Yes
			<u>No:</u> Self-taught learning	
Unsupervised	Same		No	No
Transductive	Different but related	Same	Yes	No

2.1.3 Model compression

Deploying deep neural networks (DNN) on mobile devices is a challenging issue due to insufficient memory and limited availability of computational resources in the devices (Wang et al., 2021). To cope with this challenge, various model compression techniques have been developed.

A. Pruning

Pruning aims at reducing channels, filters, neurons or layers that can be eliminated or set to zero while maintaining the model accuracy to a certain threshold (Cheng et al., 2020; Menghani,

2021; Mishra et al., 2020). This introduces sparsity to the network and reduces the size and time needed to run the DNNs.

B. Quantisation and Binarisation

Quantisation compresses the model by reducing the precision for weights and activations, which are often in 32-bit floating-point values (Menghani, 2021). A smaller model size can be resulted by quantising weights, and an improvement on latency can also be achieved with quantisation on activations. Binarisation is a more extreme quantisation which restricts the values of weights to only 0 and 1. It achieves a higher level of compression yet with a cost of greater degree of performance degradation (Mishra et al., 2020).

C. Knowledge distillation (KD)

KD is a representative method for the purpose of model compression and acceleration. Ba and Caruana (2013) demonstrated the possibility of transferring knowledge learned by a deep teacher model to a much shallower student model without a huge loss in accuracy (Figure 2-4). It was also found that student models are less prone to overfitting problem. Hinton, Vinyals and Dean (2015) further developed on the idea by distilling knowledge from an ensemble of models into a single model, and summarised the technique as KD.

As opposed to aforementioned methods which aim to transfer the logits of teacher models to students, Chen et al. (2018) proposed a method named Knowledge Distillation via Feature Maps (KDFM) which targets at transferring feature maps in the last layer of teacher model to the student model. The method is assisted by the use of GAN to force student models to generate similar feature maps as the inputs from teacher models. The KDFM models achieved comparable accuracies with state-of-the-art DNNs with faster inference time. However, the best student models to be used in KDFM is yet to be studied.

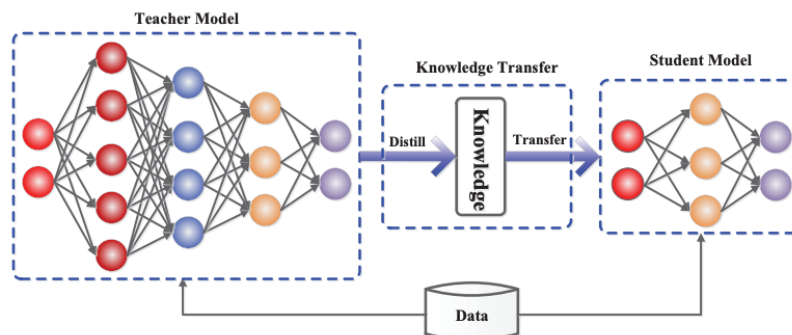


Figure 2-4: Illustration of a general KD teacher-student architecture (adapted from Gou et al., 2021)

2.2 Related datasets

Two large-scale dog datasets suitable for the dog breed fine-grained classification tasks are publicly available, namely Stanford Dogs and Tsinghua Dogs. A summary of the characteristics of each dataset is listed in Table 2-2.

Table 2-2: Summary of characteristics of the Tsinghua Dogs and Stanford Dogs dataset.

Characteristics	Tsinghua Dogs	Stanford Dogs
Number of dog breeds	130	120
Number of images	70,428	20,580
Average number of images per dog breed	200 to 7,449, in proportion to occurrence frequency in China	150 to 200
Annotations available	Class labels, Bounding boxes of head and body	Class labels, Bounding box
Source of data	Data capture system in China and Stanford Dogs	ImageNet
Year of publication	2020	2012

2.2.1 Tsinghua Dogs

The Tsinghua Dogs dataset is the most recent and largest dataset for the fine-grained dog breed classification problem (Zou et al., 2020). However, due to its recency, there have not been extensive studies performed on the dataset. Features of the dataset are listed in Table 2-2.

2.2.2 Stanford Dogs

The Stanford Dogs dataset was published by researchers from the Stanford University in 2012 (Khosla et al., 2012). It contains a total of 20,580 annotated dog images of 120 breeds, with an average of 150 to 200 images per breed. Images from ImageNet under the ‘Canis familiars’ node with resolution over 200 x 200 were included in the dataset, and blurry, noisy and duplicated images were manually filtered. However, after some scrutinisation, it is found that some noisy images exist in the dataset (Figure 2-5), which could mislead models trained without reliance on bounding boxes. The dataset has been widely used as one of the benchmark datasets in numerous FGIC researches (Wei et al., 2021), and is hence chosen over the Tsinghua Dogs dataset for this project for better comparison of performances on the models developed.

Table 2-3 lists the Top-1 accuracies on the dataset achieved in recent literatures. Transformer models outperform recent CNN methods and achieve state-of-the-art performances on the dataset. All transformer models achieve over 90% accuracy on the dataset.



(a) Inclusion of human faces in images



(b) Multiple dog breeds in a single image

Figure 2-5 Examples of noisy images in the Stanford Dogs dataset

Table 2-3: Dog breed classification model performances on the Stanford Dogs dataset in literatures




Type of model	Method	Backbone	Top-1 Accuracy
CNN	Pairwise Confusion (Dubey et al., 2018)	DenseNet-161	83.75%
CNN	SEF (Luo et al., 2020)	ResNet-50	88.8%
CNN	MPN-COV + SEB (Song et al., 2022)	EfficientNet-B5	93.0%
CNN	API-Net (Zhuang et al., 2020)	DenseNet-161	89.4%
CNN	API-Net (Zhuang et al., 2020)	ResNet-50	88.3%
CNN	API-Net (Zhuang et al., 2020)	ResNet-101	90.3%
CNN	WS-DAN (Imran & Athitsos, 2020)	Inception-V3	92.2%
Visual Transformer	ViT (Conde & Turgutlu, 2021)	ViT-B_16	93.2%
Visual Transformer	ViT-SAC (Do et al., 2022)	ViT-B_16	94.5%
Visual Transformer	TransFG (He et al., 2022)	ViT-B_16	92.3%

2.3 Related products

Three free dog breed classification applications in the UK iOS AppStore with the highest number of ratings¹ are studied and explored, which are DogScanner, Dog Pal and Dog Identifier. No information regarding the model architecture of these applications are published by the application developers. Features of the three applications are summarised in Table 2-4.

¹ It is assumed that number of ratings and popularity of an application is positively related.

Table 2-4: Summary of the features of the three free dog breed classification application with most reviews on the UK iOS AppStore2. (App Store, 2017; App Store, n.d. - a; App Store, n.d. - b; NowGaming, n.d.; Siwalu Software, n.d.)

Features	 DogScanner	 Dog Pal	 Dog ID
Image upload channel	Capture from camera or upload from album		
Size of app	112.4 MB	38.9 MB	14.9 MB
Dog breeds supported	More than 370	-	All
Accuracy claimed	More than 90%	-	-
Information shown on classification result page	<ul style="list-style-type: none"> • Primary and secondary matches • Confidence of the results 	<ul style="list-style-type: none"> • Prediction and possible look-alike • Proportion of a breed in mixed breed predictions 	<ul style="list-style-type: none"> • Primary and secondary matches • Confidence of the results

To gain insights into the performances of these applications in terms of accuracy and inference speed, 8 images were uploaded to the applications for classification. Details of the images uploaded and the results of the experiment can be found in Appendix I. Screenshots of result pages of the applications are available in Appendix II. A summary of the performances is shown in Figure 2-6. For all three applications, an error message is shown when they detect no dog in the image.

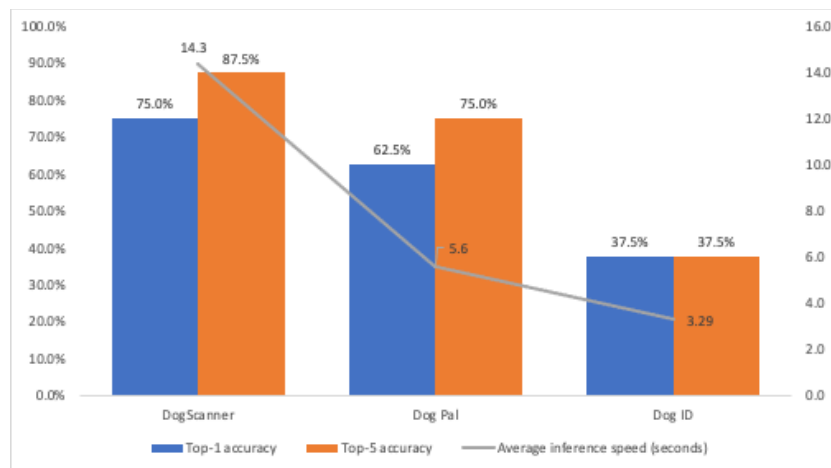


Figure 2-6: Performances of the dog breed identification mobile applications (DogScanner, Dog Pal and Dog ID) in the experiment.

² Indicated with dash are information not publicly available.

2.3.1 DogScanner

DogScanner has the highest number of reviews among all three applications. In the experiment, it achieves the highest Top-1 and Top-5 accuracies. However, its inference speed is significantly slower than the other two applications. The better accuracy, longer inference time and much larger application size might imply that the application leverages a more complex model than the other two applications.

On the result page, the predictions along with the confidence of each prediction are displayed. The application classifies a dog as mixed breed when there are more than two significant matches predicted by its model.

2.3.2 Dog Pal

Dog Pal ranks second in terms of the number of reviews. In the experiment, it achieves slightly worse performance than DogScanner, yet its average inference speed is significantly faster.

As opposed to DogScanner, the numeric value of prediction confidence is not shown. When a dog is predicted to be a mixed breed dog, the proportion of breeds is shown, which might be interpreted as the confidence of each breed prediction. When a dog is predicted to be a purebred, a look-alike breed is also suggested. However, it is unsure that whether the look-alike breed suggested is based on expert knowledge or its model.

2.3.3 Dog ID

In terms of Top-1 and Top-5 accuracies, Dog ID performs the worse in the experiment. However, it has the shortest inference time. This might indicate that its predictions are based on a much simpler model than the other two applications.

The application does not give mixed breed prediction. It makes a single prediction about the breed of a dog, along with some other possible matches and the confidence of each prediction. As opposed to DogScanner, the confidences of the predictions do not sum to one, which might indicate that the two applications adopt different activation functions or output structures.

3 Design

3.1 Objectives

The objective of this project is to develop an iOS mobile application that allows users to upload a dog image and outputs the top breed matches among the 120 breeds in the Stanford Dogs dataset. Figure 3-1 shows the UI design of the application, while Figure 3-2 illustrates the end-to-end flow of a prediction request from image upload to display of prediction results in the application.

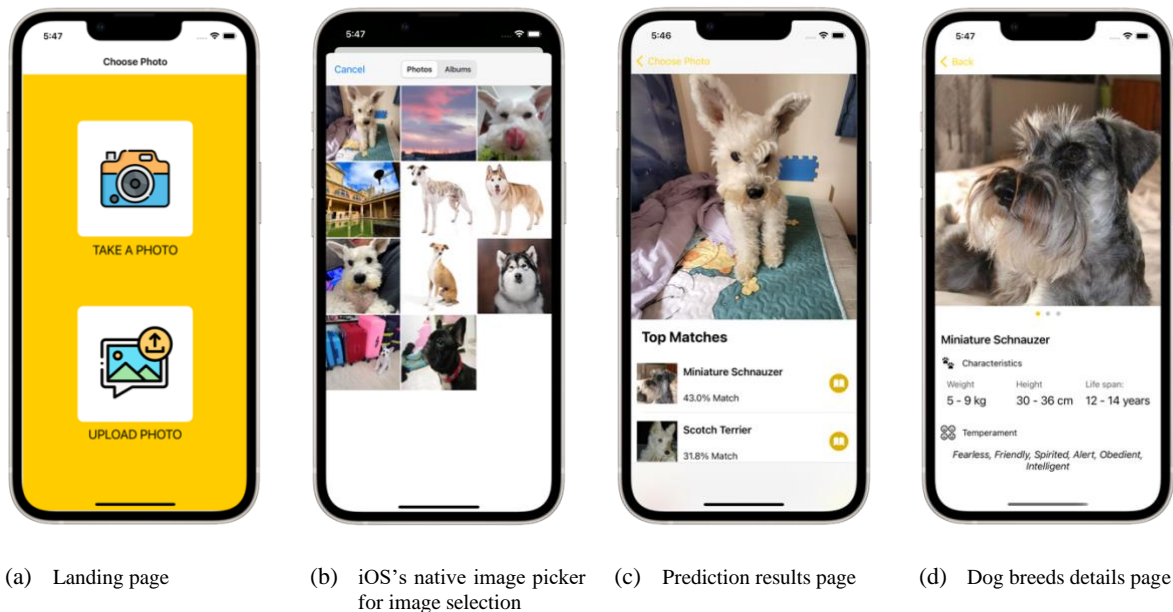


Figure 3-1: UI Design of the iOS application.

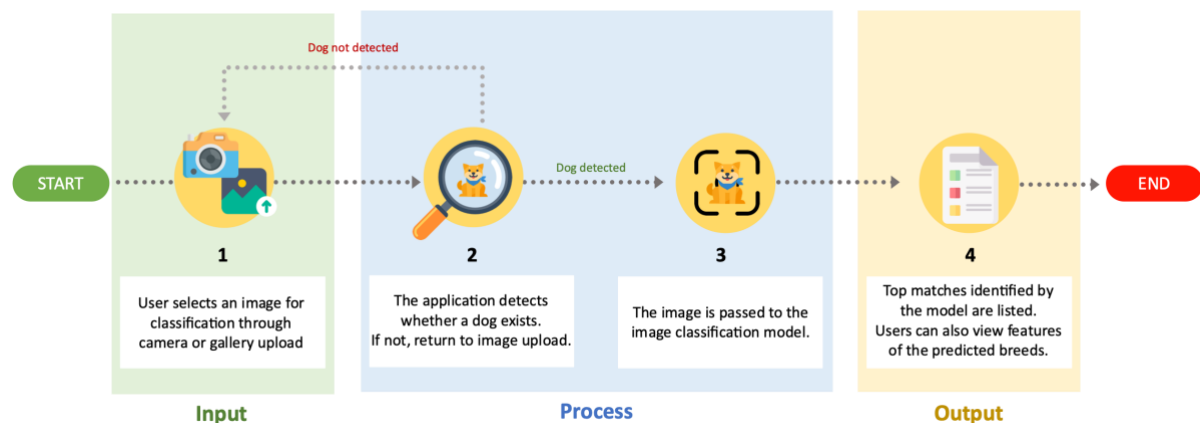


Figure 3-2: High-level overview on the end-to-end process flow of a prediction request.

3.2 Components

3.2.1 Input: Image upload and preprocessing

The mobile application will be able to collect user input through camera or gallery upload. After the image is inputted, it will be passed for preprocessing, such as center-cropped and normalisation, according to the specific input requirements of the deep learning models.

3.2.2 Process: Deep learning models prediction

A. Dog detection model

The image uploaded by user will first go through a dog detector model to confirm the existence of a dog in the image before it is passed to the dog prediction model. If no dog is detected in the image, user will be prompted to re-select an image for classification and given tips on taking a good quality photo. Without this model as a data validation step, the application will output prediction results even when user uploads a photo without any dog as the FGIC model is only trained with dog images to learn distinctive patterns of dog breeds. Besides detecting the existence of dogs, implementing the model can also ensure that only upright images are passed to the classification model. This prevents inaccurate prediction due to incorrect orientation as most DNN models are not rotation invariant. Layman users usually have little understanding on how deep learning models give predictions, they might have lower trust on the prediction results if the application outputs dog breed prediction when there is no dog in the image, or give inaccurate result due to incorrect orientation of input images. It is expected that the overall user experience can be improved with this additional layer of data validation. As no subsequent processes will be performed if the model detects no dog, the final model should aim to minimise the possibility of false negative results. Also, as there will be two deep learning models embedded in the application, and dog breed FGIC is a much harder problem than dog detection, the final model size should be as lightweight as possible with limited impact to the storage requirement and efficiency of the application.

B. Dog breed FGIC model

The FGIC model will take the preprocessed image as input and output the predicted confidence for each of the 120 breeds in the Stanford Dogs dataset. The predictions will then be ranked in descending order according to the confidence values. The model should be able to achieve accuracy of over 90% on the testing set, which has been achieved with both CNN and

transformer models in previous literatures. The final model size is expected to be larger than that of the dog detector model.

3.2.3 Output: Predictions and breed information display

There will be a “**Prediction**” page that display 5 dog breeds with the highest confidence level predicted by the FGIC model. If any of the top 5 matches has a confidence value close to or equal to 0%, they should not be shown to users as the results are non-intuitive. Considering that not all users are familiar with all the dog breeds, there will be a “**Details**” page for users to view brief information about the predicted dog breeds such as body size, lifespan and temperament, as well as sample images of the predicted dog breeds.

3.3 Evaluation criterion

Evaluation will be conducted on both the deep learning model and the mobile application. For the deep learning model, its **accuracy**, **model size** and **inference speed** will be compared with state-of-the-art performances on the Stanford Dogs dataset achieved in previous studies. For the mobile application, its performance will be evaluated against existing market-leading solutions elaborated in section 2.3 based on the same set of evaluation criterion listed above.

4 Methodology

4.1 Deep learning models

The deep learning models are developed in Python using Tensorflow Keras (“**Keras**”) and **PyTorch** frameworks, then converted into **CoreML packages** for incorporation into the iOS application using Apple’s Core ML Tools (Apple, n.d.-b). The training and conversion are done via Google Colab. Taking the limited time and computing resources into consideration, it is decided to develop the deep learning models with inductive transfer learning. Models pre-trained on ImageNet are used as base models for feature extraction, and new fully connected output layers are attached for classification. All input images are center-cropped to 224 x 224 pixels, and normalised according to the specific requirements of the model architecture. Simple augmentations with random horizontal flip and random rotation with a factor of 0.1 are also applied to all training images.

A two-phase training process is adopted for the training of all models. First, the models are trained with Adam optimiser with a learning rate of either $1e^{-2}$ or $1e^{-3}$. In the first phase, the

goal is to leverage the generic feature extraction capability of the pre-trained models, and only configure the newly added classification layers for the dog breed FGIC problem. Hence, all weights in the base model are frozen and only weights of the classification layers are updated during training. In the second phase, all layers in the models are unfrozen to fine-tune the entire model. This allows the whole model to fine-tune on the dog breed FGIC problem. To avoid suddenly huge changes to the weights that could destroy information gained in the previous phase, a small learning rate of $1e^{-5}$ for the Adam optimiser is used. For both phases, a batch size of 64 and early stopping with 5 epochs of tolerance on validation accuracy are applied, and the model with the best validation accuracy is returned at the end of training. The models after fine-tuning are then evaluated on the test set.

4.1.1 Dog detection model

A. Model training and selection criterion

The VNRecognizeAnimalRequest API provided in Apple's Vision Framework is first being considered for the dog detector as it does not increase the size of the application (Apple, n.d.-a). However, after some testing, it is found that false negative results are often observed, which is likely to cause user frustrations as no breed classification request will be processed if the model detects no dog in the image. Hence, it is decided to use a self-trained model instead.



Figure 4-1: (Dog detector model) Example of images where Apple's API gives false negative results

Considering that dog detection is a much simpler task than dog breed FGIC with only two classes, and the final application will have two DNN models embedded, MobileNetV2, which has the most lightweight architecture and the second-fastest inference time among all models available in the Keras framework (Keras, n.d.), is chosen to be the base model for this task in order to have the minimal impact on the final application size and processing speed. The architecture of the model is presented in Figure 4-3 below. The dataset used for training is a combination of the **Stanford Dogs** and **Caltech-256** datasets. The entire Stanford Dogs dataset of 20,580 images is used as the “**dog**” dataset, while 19,359 images from the Caltech-256 dataset are used as the “**no_dog**” dataset. Figure 4-2 shows the distribution of the two classes

in the training, validation and testing sets respectively. Stratified splitting is used when splitting the dataset to preserve a fairly equal proportion of the two classes. As some four-legged mammals in Caltech-256 can be visually very similar to certain breeds of dogs, they are removed to limit the possibility of false negative predictions made by the model to prevent the application from not proceeding even when a dog exists in image uploaded by user. The list of classes excluded are detailed in Appendix III.

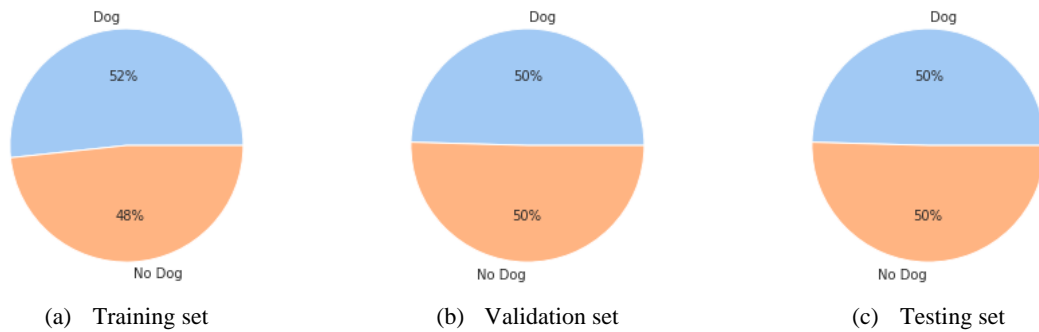


Figure 4-2: (Dog detector model) Distribution of samples in training, validation and testing sets respectively

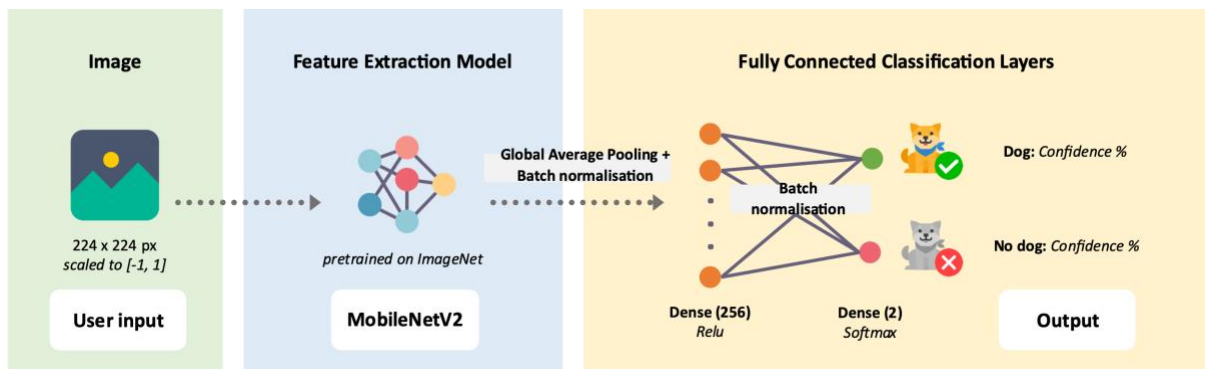


Figure 4-3: (Dog detector model) Architecture of the dog detection model. Global average pooling and batch normalisation layers are applied after the MobileNetV2 feature extractor. A fully connected layer with 256 nodes and an output layer with the “dog” and “no_dog” classes are then connected as the classification layers.

B. Result analysis

The test performance of the final model will be evaluated with the aid of a confusion matrix. Wrongly classified examples will be scrutinised using the visualisation methods of Activation Maximisation and Grad-CAM (Selvaraju et al., 2020). Grad-CAM produces a coarse localisation heatmap that shows the importance of regions in an image. These visualisation techniques aid the understanding of how the model makes the prediction and the possible reasons for misclassification.

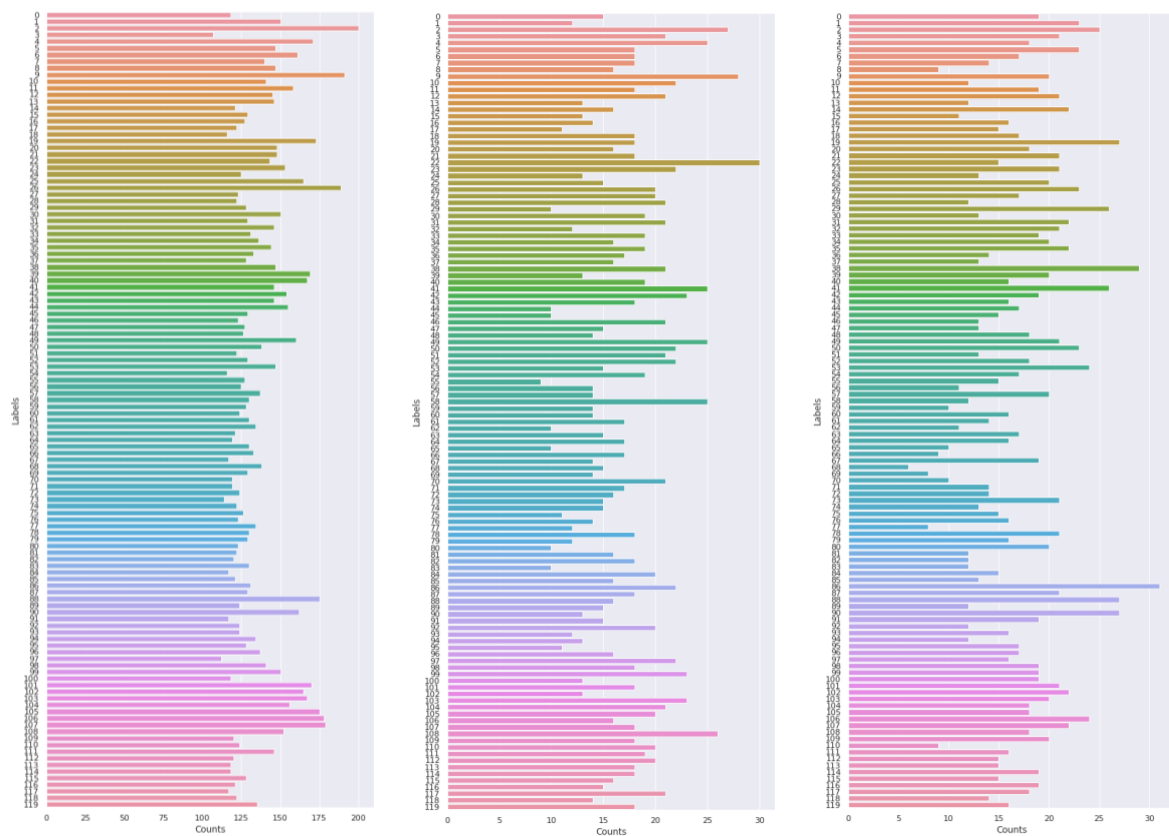
4.1.2 Dog breed classification model

Dog breed classification is the main feature of the application, and is a difficult FGIC problem. Hence, various state-of-the-art deep learning models, including CNN and ViT are explored with the aim to achieve good accuracy. A summary of the model architectures explored are listed in

Table 4-1. All pre-trained CNN models are downloaded from Keras, while the pre-trained ViT model is downloaded from Hugging Face (Hugging Face, 2021).

A. Model training and selection criterion

A train-validation-test split of 80-10-10 is applied to the Stanford Dogs dataset for the training of the breed FGIC model. A stratified split is used to ensure a similar proportion of classes exist in all three splits.



(a) Training set

(b) Validation set

(c) Testing set

Figure 4-4: (Breed FGIC model) Distribution of samples in training, validation and testing sets respectively

A total of 11 models are explored to select the best model architecture for the dog breed FGIC task. After The models are attached with an identical block of classification layers, and trained with the same settings and two-phase procedure mentioned in 4.1. Apart from ViT, all models

explored are CNN models. Table 4-1 ranks the models according to their size and number of parameters in ascending order.

MobileNetV2, DenseNet121 and EfficientNetB0 were the first set of models explored as they have a relatively light-weight architecture. EfficientNetB0 achieves the best performance during the first round of exploration, hence, other EfficientNet models are chosen for further exploration. Two state-of-the-art CNN models (ResNet50V2 and InceptionResNetV2) with larger model sizes are also explored for the purpose of understanding whether model size and accuracy are positively related. Lastly, the ViT model is included in the study for investigating the differences between the performances of CNN and transformer models.

Table 4-1: (Breed FGIC model) Overview of model architectures explored for the dog breed classification model.

Model architecture	Size (MB)	Parameters (Million)
MobileNetV2 (MNV2)	14	3.5
EfficientNetB0 (ENB0)	29	5.3
EfficientNetV2B0 (ENV2B0)	29	7.2
DenseNet121 (DN121)	33	8.1
EfficientNetV2B1 (ENV2B1)	34	8.2
EfficientNetV2B2 (ENV2B2)	42	10.2
EfficientNetV2B3 (ENV2B3)	59	14.5
EfficientNetV2S (ENV2S)	88	21.6
ResNet50V2 (RN50V2)	98	25.6
InceptionResNetV2 (InRNV2)	215	55.9
ViT	346	86

As the model is to be deployed on a mobile application which runs on devices with limited computing resources, the final model architecture is selected with consideration on not only accuracy, but also model size and inference time. As converting and deploying all models on an iOS device is time-consuming, the final model selection considers inference time of models by comparing the average time taken to run a single batch of 64 images during model testing. After the final model architecture is selected, hyperparameter tuning on the classification block architecture and the choice of optimiser are performed to further fine-tune the accuracy of the model.

B. Results analysis

The final model's performance will be evaluated from the perspectives of (i) **overall results** and **comparison with models in literatures**, (ii) **most confused breed pairs**, (iii) **most misclassified breeds**, and (iv) **worst predictions made by the model**. For the analysis on worst predictions, the test predictions will be ranked in descending order of the numerical difference between the confidences of the correct breed and the top-1 predicted breed. The

larger the difference, the worse prediction the model has made. In addition to Grad-CAM, Activation Maximisation visualisations on the penultimate dense layer are also used to gain insights into the decisioning of the final model by visualising the most representative patterns for a particular class (Mahendran & Vedaldi, 2016).

4.1.3 iOS mobile application

The iOS application is developed in Xcode, targeted to support devices with iOS 15.5 or later and tested on Xcode's iPhone 13 simulator and a physical iPhone 13. This project references the user input collection and image classification request creation in the sample code "Classifying images with Vision CoreML" released by Apple (Apple Developer, n.d.).

C. Image classification

The image classification is supported by Apple's Vision framework. When user selects an image to be classified, an image classification request will be created, and Vision will resize and crop the image according to the constraints of the model which the image is to be passed into. A prediction will be produced after the image is passed to and processed by the model and the results will be relayed back to the application for ranking, filtering, and display to users.

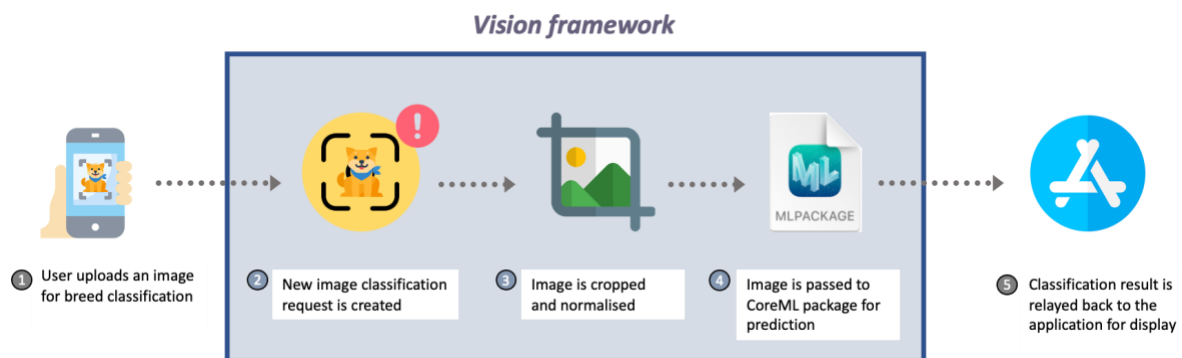


Figure 4-5: (Mobile app) Frontend-backend communication for an image classification request in the mobile application.

D. Breed information retrieval

Details of dog breeds are mainly obtained from a public service API, **The Dog API** (The Dog API, n.d.). It is a free service that provide properties of over 200 dog breeds. However, information about the dog breeds "Dingo" and "Dhole" in Stanford Dogs dataset are not available in the Dog API, hence their information is supplemented manually with information from other online sources. The breed information is incorporated into the mobile application in the form of a JSON file. A mapping JSON is also included in the application to map breeds

in Stanford Dogs dataset and the Dog API as some are named differently. When user initiates a request to display the information of a breed, the application will look for the ID of the breed in the Dog API through the mapping JSON, then retrieve the breed properties in the breed details JSON file.

Besides textual information, three images from each dog breed in the Stanford Dogs dataset are also stored in the application. These images are presented together with the breed properties described above to give users a better idea of the appearances and visual features of the breed.

5 Results

5.1 Deep learning models

5.1.1 Dog detection model

A. Model training and evaluation

The dog detection model built from MobileNetV2 architecture with ImageNet pretrained weights achieves over 99% of validation accuracy after the first epoch, which could be a result of the abundant samples for an easy task with only 2 classes. No major improvement in accuracy is observed during the fine-tuning phase with all model weights unfrozen, which implies that the model might start to overfit. Due to limited time and satisfactory accuracy, no further hyperparameter fine-tuning is performed. The final model chosen is the model after 2 epochs of training in the fine-tuning phase, which achieves the highest validation accuracy of 99.58%.

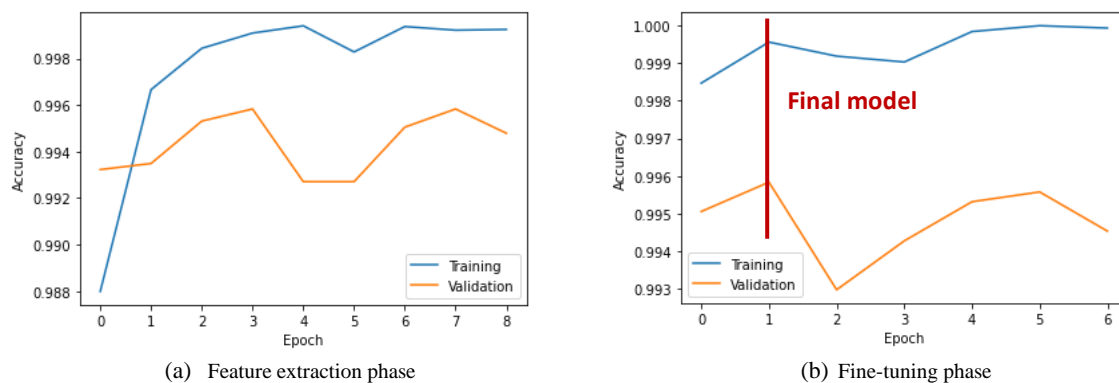


Figure 5-1: (Dog detector model) Top-1 accuracy during feature extraction and fine-tuning phase.

B. Results analysis

Overall results on test set

Figure 5-2 shows the confusion matrix of predictions made by the final model on the test set. It achieves an overall accuracy of 99.53%, a 0.32% false positive rate, and a 0.15% false negative rate. The performance is satisfactory as the accuracy is high, and false negative predictions are rare.

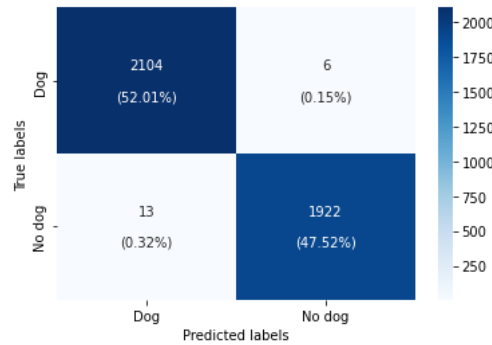


Figure 5-2: (Dog detector model) Confusion matrix

Analysis on false negative samples

Figure 5-3 shows the original images and Grad-CAM visualisations of the 6 false negative predictions made by the final model. There are three possible reasons for the misclassifications identified, which are *lack of posture samples*, *poor image quality* and *incorrect orientation*.



Figure 5-3: (Dog detector model) Original image and Grad-CAM of the false negative predictions

I. Lack of posture samples

For sample 1, by observing the Grad-CAM, the model is capable of locating the region of interest in the image. This misclassification is likely due to the lack of samples of dogs with jumping postures in the Stanford Dogs dataset. To remediate this classification limitation, a

reminder to user on choosing dog images with standing or sitting postures will be displayed before users upload an image.

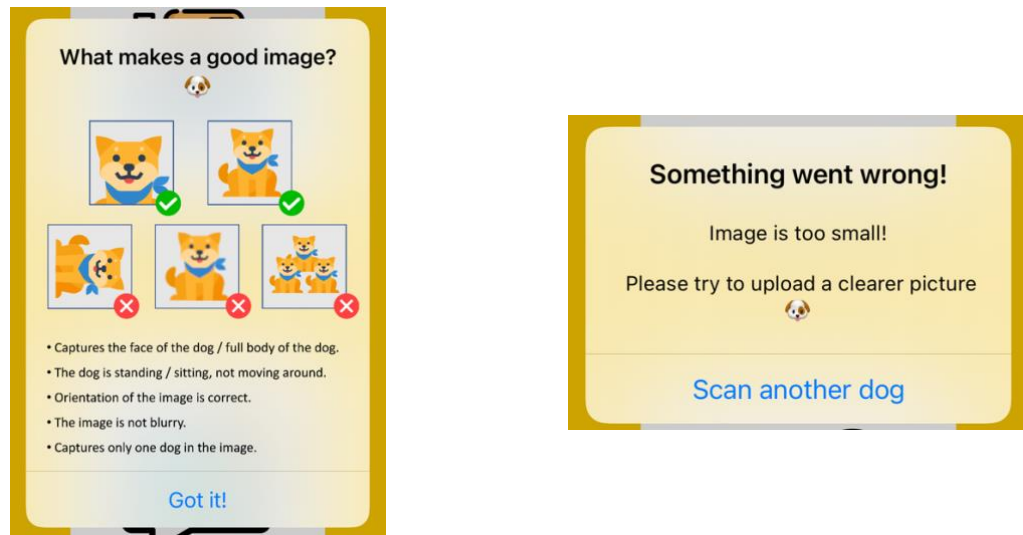
II. Poor image quality

For samples 2 and 6, their file sizes are 6.31 KB and 6.44 KB respectively, which are significantly lower than the average file size of correct classifications (38 KB). This implies that the resolution of the images are much lower than average. The low resolution images could prevent the model from effectively extracting clear and distinctive features from the image, and cause the model to perform classifications based on noises. As images captured with camera is unlikely to have resolution issues, a check is implemented for images uploaded by users through gallery to ensure that images uploaded by users are at least 38 KB.

For samples 4 and 5, there are irrelevant objects that are much larger than the dogs in the images, which could distract the model. From the Grad-CAM visualisations, it can be observed that the model fails to pay attention to the dog, instead, it focused on the tunnel in sample 4 and the car in sample 5 which occupy a much larger proportion in the image than the dogs. To ensure quality input from users, tips on taking good quality images will be displayed before user uploads an image.

III. Incorrect orientation

For sample 3, the misclassification is most likely caused by the incorrect orientation of the image. As CNNs are not rotational invariant, images with incorrect orientation will most likely lead to incorrect predictions. As a remediation, a reminder to users on uploading images in correct orientation will be displayed before user uploads an image.



(a) Tips on the criterion of a good image is shown before user selects an image (b) Images with file sizes too small will not be passed to the classification model for processing

Figure 5-4: (Dog detector model) New messages implemented in the application inspired from false negative samples analysis

Analysis on false positive samples

The false positive samples are mostly caused by *low resolution images* and *lack of training samples*. The full list of original images and the Grad-CAM visualisations of the false positive samples can be found in Appendix IV.

I. Low resolution images

Similar to the false negative samples, some false positive samples have very small file size, which leads to ineffective feature extraction by the model. 8 out of the 13 samples have file sizes under 30 KB. With the additional check on file size implemented in the final application, this type of misclassification should be greatly reduced.

II. Lack of training samples

Due to limited computational resources, the training set only contains 242 non-dog objects. When the model encounters a new object or objects that rarely appears in the dataset, the accuracy cannot be guaranteed. Figure 5-5 shows an example of the false positive prediction that is likely caused by the issue. From the Grad-CAM visualisation, it can be observed that the model's attention is mainly centred around the bride in the image. The misclassification might have been caused by patterns of human skin or clothing textures that are similar to dog patterns learned by the model as the Caltech-256 dataset does not contain many full-body images of humans.



Figure 5-5: (Dog detector model) False positive sample likely caused by the lack of variety of training samples

The lack of quantity of samples might also be a reason for the misclassifications. Some misclassifications are made on objects that are visually quite distinct from dogs, as shown in Figure 5-6. As these images are the only misclassified samples for these objects, this could imply that the model is capable of distinguishing the objects from dogs, yet it has not encountered enough samples to capture the variance of the objects. These misclassification could be reduced with heavier image augmentation, or with more samples introduced to the training set.



Figure 5-6: (Dog detector model) False positive sample likely caused by the lack of quantity of training samples

The lack of animal samples also caused some of the misclassifications. From the Grad-CAM visualisations in Figure 5-7, the model made the conclusion that dog exists in the images based on the hair patterns of the owl and the chimpanzee. To minimise false negative predictions made by the model which stops the application from continuing with the classification task, four-legged mammals are removed from the “no_dog” dataset. Although some animals are still kept in the dataset, the removal of four-legged mammals reduces the model’s ability to distinguish dogs from other animals which can demonstrate very similar features as dogs such as furry hair. However, as these misclassifications are relatively rare and inclusion of more samples would further increase the long model training time, the model is considered sufficient for the purpose of the application.

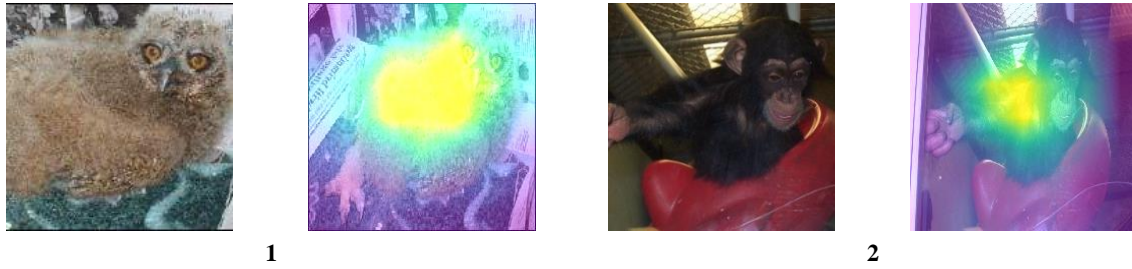


Figure 5-7: (Dog detector model) False positive sample likely caused by the lack of animal training samples

5.1.2 Dog breed classification model

A. Model training and evaluation

Accuracy

I. Feature extraction phase

During this phase, all CNN models are trained with a learning rate of $1e^{-2}$. The ViT model is trained with $1e^{-3}$ as no improvement in accuracy is shown with a learning rate of $1e^{-2}$. Figure 5-8 shows the percentage improvement in training and validation accuracies of the 11 models during this phase. The duller and taller bars indicate the percentage improvement in training accuracy, while the brighter and shorter bars indicate the percentage improvement in validation accuracy. The models are ranked by their model size in ascending order. It can be observed that models with a smaller model size generally have greater differences between their percentage improvements in training and validation accuracies. However, despite having a relatively large model size, ResNet50V2 shows the greatest difference between training and validation accuracies. The large difference in training and validation accuracies might indicate the models learned more noises than discriminative features from the training images. The history of training and validation accuracies for each of the 11 models during this phase can be found in Appendix V. From the plots, more obvious trends of increasing training accuracy and decreasing validation accuracy are observed for DenseNet121 and InceptionResNetV2, which could indicate the problem of overfitting.

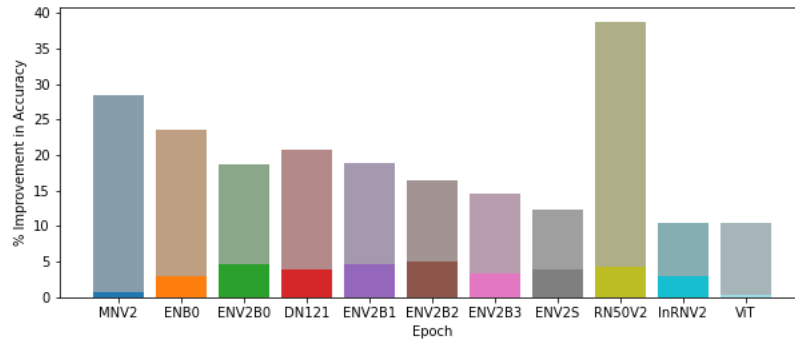


Figure 5-8: (Breed FGIC model) Percentage improvement in Top-1 accuracy of the 11 models during feature extraction

Figure 5-9 shows the history of training and validation accuracies of the 11 models during feature extraction. All models have steady improvement in training accuracy. However, in terms of validation accuracy, DenseNet121 begins to decline after the first epoch, this could indicate that the model begins to overfit. All models have a validation accuracy that is similar or lower than its training accuracy except for ViT, which may imply that the model is more resistant to the overfitting problem. Most models stop having an improvement in validation accuracy in less than 10 epochs except for MobileNetV2, EfficientNetV2S and ResNet50V2. ViT, the EfficientNetV2 models and InceptionResNetV2 have validation accuracies that are significantly better than the other models. At the end of feature extraction, ViT achieves the best validation accuracy, followed by EfficientNetV2B3 and EfficientNetV2B2.

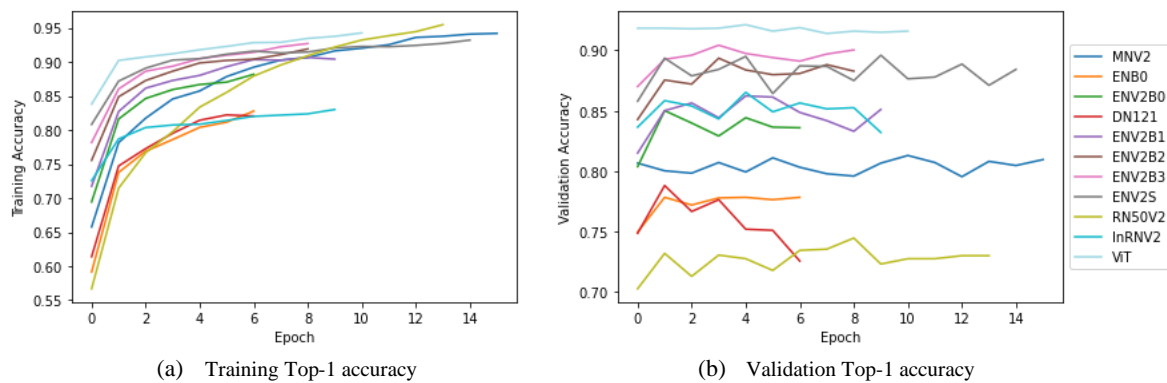


Figure 5-9: (Breed FGIC model) History of the training and validation Top-1 accuracies of the 11 models during feature extraction

II. Fine-tuning phase

During the fine-tuning phase, most models show only minimal improvement in validation accuracy even with all their weights unfroze. From Figure 5-10, EfficientNetB0 and DenseNet121 have the most improvement in validation accuracy. DenseNet121, EfficientNetB0 and InceptionResNetV2 have the largest differences in the increment of training and validation accuracies, which means that the model picked up more noises than

discriminative features during training. The history of training and validation accuracies for each of the 11 models during this phase can be found in Appendix VI. From the plots, MobileNetV2, DenseNet121 and EfficientNetV2B2 demonstrates more prominent signs of overfitting with increasing training accuracy and decreasing validation accuracy.

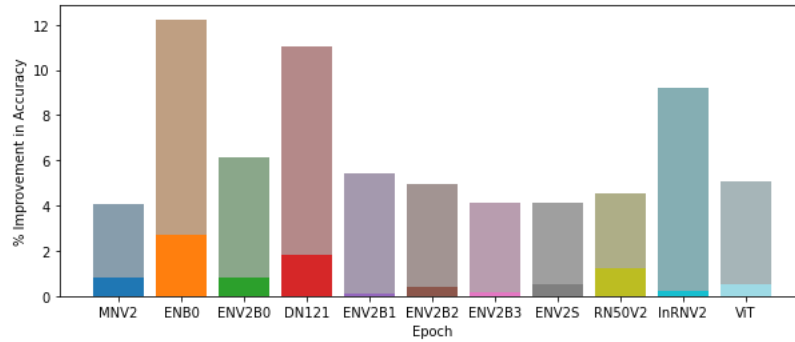


Figure 5-10: (Breed FGIC model) Percentage improvement in Top-1 accuracy of the 11 models during fine-tuning

From Figure 5-11, it can be observed that most trainings terminated at epoch 5 or 6, which means that there is no improvement in validation accuracy after the first or second epoch. By looking at the validation accuracy, the models can be coarsely divided into three groups. The “**Superior**” group being those achieving over 87%, the “**Mediocre**” being those between 80% to 85%, and the “**Poor**” group being those below 80%. ResNet50V2 is the only model in the “Poor” group. MobileNetV2, DenseNet121 and EfficientNetB0 which have comparatively smaller model sizes all fall into the “Mediocre” group. Despite having a small model size, EfficientNetV2B0 performs significantly better than the other small-sized models. All EfficientNetV2 models, InceptionResNetV2 and ViT belong to the “Superior” group. The same grouping applies to the testing accuracy of the best models of each of the 11 model architectures after fine-tuning illustrated in Figure 5-12. The best models of ViT, EfficientNetB2V3 and EfficientNetV2S achieve the best test accuracies of 93.2%, 92.14% and 90.43% respectively.

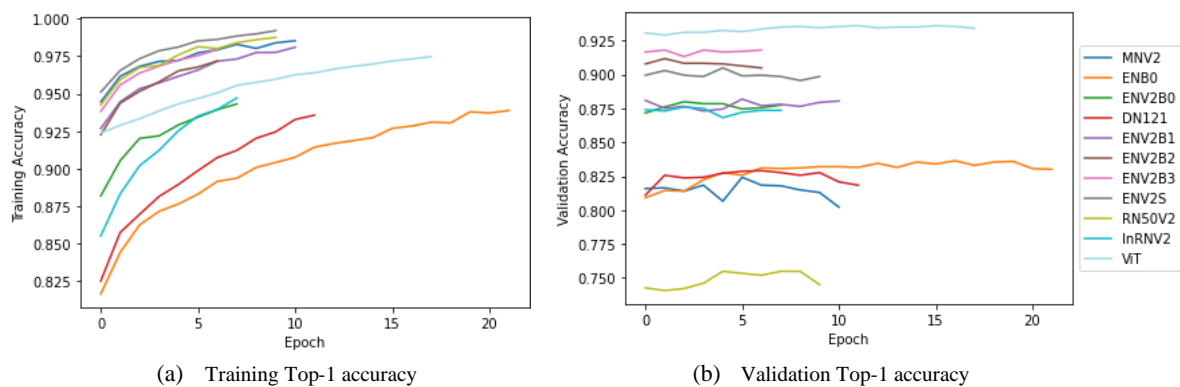


Figure 5-11: (Breed FGIC model) Training history of the 11 models during fine-tuning

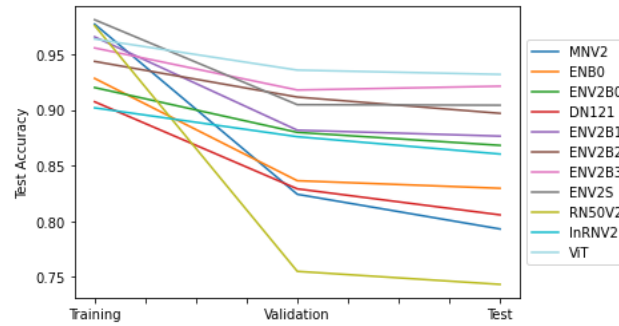


Figure 5-12: (Breed FGIC model) Training, validation and testing Top-1 accuracies of the best model for each of the 11 architectures

Model size and inference time

In terms of model size and inference time, ViT is significantly bigger and slower than the rest of the models, yet also the model with the highest testing accuracy, as illustrated in Figure 5-13. However, having a large model size does not necessarily imply good performance for the dog breed FGIC problem. InceptionResNetV2 has the second largest model size, yet all EfficientNetV2 models achieve higher test accuracy; ResNet50V2 has the third largest model size, yet it is the worst performing model. It is also observed that model size and inference time might not be positively related as ResNet50V2 has a faster inference time than two of the smaller models EfficientNetV2B3 and EfficientNetV2S. As expected, MobileNetV2 has the fastest inference time with its smallest model size and mobile-optimised architecture (Sandler et al., 2018). However, it is also the model with the second lowest test accuracy.

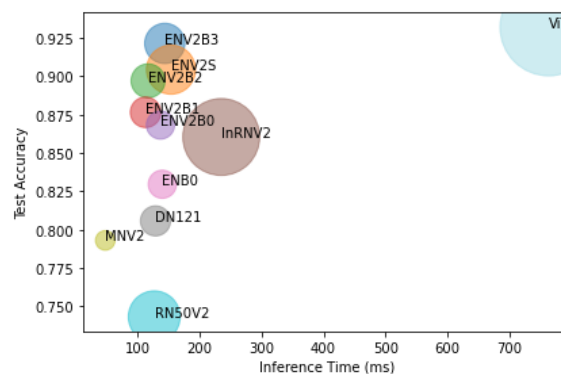


Figure 5-13: (Breed FGIC model) Top-1 accuracy, size and inference time of the 11 models

B. Final model architecture selection

The final model is selected with a balance between accuracy, model size and inference speed. As mentioned in section 4.1, ViT is unsuitable for mobile deployment due to its significantly larger model size and longer inference time. Its testing accuracy is only 1% better than the second-best model (EfficientNetV2B3), yet its model size is 4.8 times bigger, and its inference speed is 4 times slower. Comparing EfficientNetV2B3 with the third-best model (EfficientNetV2S), it is both smaller and faster, hence EfficientNetV2S is also eliminated.

EfficientNetV2B3 and EfficientNetV2B2 are both converted into CoreML packages to compare their inference speed on mobile devices. From testing, there is no significant difference between the inference speed of the models, hence, the more accurate EfficientNetV2B3 is chosen as the final model for the mobile application.

C. Hyperparameter tuning

Hyperparameter tuning on the classification block architecture and the choice of optimiser are performed with the aim to further improve the model accuracy. Table 5-1 lists all the structures and optimisers experimented, and the corresponding validation performances. Trials 1 to 5 focus on experimenting the number of dense layers and the number of nodes in dense layers. It is found that the classification block structure with two dense layers of 256 nodes achieves the best validation accuracy. In trial 6, the first Global Average Pooling layer is replaced with a Global Max Pooling layer, which results in a lower validation accuracy. Trial 6 leverages the same classification block structure as trial 5, yet the optimiser used in fine-tuning phase is switched to SGD with learning rate of $1e^{-4}$ and momentum of 0.9 instead of Adam with learning rate of $1e^{-5}$ in other trials. The validation accuracy achieved in trial 7 is the same as that in trial 5, however, the validation top 5 accuracy is slightly higher. Hence, the model trained in trial 7 is selected as the final model for inclusion in the mobile application.

Table 5-1: (Breed FGIC model) Experiment setups and results during hyperparameter tuning

Classification block structure	Trial						
	1	2	3	4	5	6	7
Global Pooling	Average					Max	Average
Batch Normalisation	✓	✓	✓	✓	✓	✓	✓
Dense layer (nodes)	1280	256	512	512	256	256	256
Batch Normalisation	✓	✓	✓	✓	✓	✓	✓
Dense layer (nodes)	-	-	-	256	256	256	256
Batch Normalisation	-	-	-	✓	✓	✓	✓
Dense layer (nodes)	120	120	120	120	120	120	120
Optimiser							
Fine-tuning phase	Adam						SGD
Results							
Validation Top-1 accuracy	0.9180	0.9194	0.9175	0.9194	0.9199	0.9165	0.9199
Validation Top-5 accuracy	0.9951	0.9946	0.9956	0.9927	0.9937	0.9937	0.9941

D. Results analysis

Overall results on test set

The final model achieves a test accuracy of 91.94%, and a top-5 accuracy of 99.32%. Comparing the result with current state-of-the-art results in literatures, the final model surpasses the performances of almost all CNN models except WS-DAN. However, the WS-DAN model has a backbone architecture that is much larger than the final model. The ViT model trained in this project achieves identical result as literature.

Table 5-2: (Breed FGIC model) Comparison of results of the final model and results in literatures




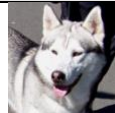



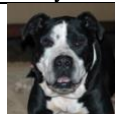












Type of model	Method	Backbone	Top-1 Accuracy
CNN	Pairwise Confusion (Dubey et al., 2018)	DenseNet-161	83.75%
CNN	SEF (Luo et al., 2020)	ResNet-50	88.8%
CNN	MPN-COV + SEB (Song et al., 2022)	EfficientNet-B5	93.0%
CNN	API-Net (Zhuang et al., 2020)	DenseNet-161	89.4%
CNN	API-Net (Zhuang et al., 2020)	ResNet-50	88.3%
CNN	API-Net (Zhuang et al., 2020)	ResNet-101	90.3%
CNN	WS-DAN (Imran & Athitsos, 2020)	Inception-V3	92.2%
Visual Transformer	ViT (Conde & Turgutlu, 2021)	ViT-B_16	93.2%
Visual Transformer	ViT-SAC (Do et al., 2022)	ViT-B_16	94.5%
Visual Transformer	TransFG (He et al., 2022)	ViT-B_16	92.3%
CNN	Final model	EfficientNetV2-B3	91.94%

Most confused pairs

Table 5-3 lists the 5 pairs of dog breeds that are most commonly misclassified by the final model. All 6 breed pairs are visually similar, and most have very high intra-class variances and small inter-class variances. For instance, a Collie with black and white coat can look very similar to a Border Collie with the same coat shade. Also, some breed pairs are more easily distinguishable if information about their body sizes are known, such as a Toy Poodle and a Miniature Poodle, also an Eskimo Dog and a Siberian Husky. However, the information is hard to be obtained via CNN as CNN models are limited in capturing global information. Besides, most images capture only the dog and has no other objects for size comparison, hence, little to no information about body size can be understood from the images.

The complete confusion matrix can be found in Appendix VII.

Table 5-3: (Breed FGIC model) Top 5 most confused breed pairs for the final model

#	Breed pair				Total misclassifications	Type of misclassification	
	Breed A		Breed B			A as B	B as A
1					11	9	2
	Eskimo Dog		Siberian Husky				
2					9	5	4
	Staffordshire Bullterrier		American Staffordshire Terrier				
3					8	8	0
	Collie		Border Collie				
4					7	3	4
	Walker Hound		English Foxhound				
5					6	4	2
	Toy Poodle		Miniature Poodle				

Most misclassified breeds

The top 5 most misclassified breeds are listed in Table 5-4. All of these breeds are those with very similar appearances as other breeds as mentioned above. Although the breeds have high top-1 misclassification rates, the true breed of the misclassified samples still rank within the top-5 predictions.

Table 5-4: (Breed FGIC model) Statistics on the top 5 most misclassified dog breeds

#	Breed	# Samples	# Misclassifications	% Wrong Top-1 Prediction	Average rank of correct prediction
1	Eskimo Dog	16	10	62.5	2
2	Collie	20	10	50.0	2
3	Walker Hound	11	5	45.5	3
4	Staffordshire Bullterrier	12	5	41.7	2
5	Miniature Poodle	19	6	31.6	2

Worst predictions

The worst predictions made by the final model are defined as the predictions with the largest differences between the rank of confidence value of the true breed and that of the top-1 predicted breed.

Table 5-5 lists an overview of the worst 5 predictions made by the model on the testing data. The Grad-CAM visualisations show that even though the dog breeds are misclassified, the model is able to identify the regions where the dogs are located at. Unfortunately, the activation maximisation visualisations do not give much insights into how the predictions are made as most are noisy. Although some body shapes, faces and eyes of dogs can be vaguely identified from the visualisations, it is hard to manually map these patterns back to the sample images with bare eyes. From these 5 samples, three reasons for misclassification can be observed, which are *poor image quality*, *wrongly annotated images*, and *insufficient samples of poses* in the training set.

I. Image quality

In terms of image quality, some images include multiple dogs in the image and some do not capture the dog well. In sample 1, there is a Whippet and a Briard in the image. As the model is designed to classify a single breed, and the Briard is bigger in the image, the model would tend to focus on the Briard instead of the whippet, as portrayed in the Grad-CAM visualisation. In samples 4 and 5 in Table 5-5, the photos are taken at a far distance from the dog, which causes the proportion of dog in the images to be small and with low resolution. The low

resolution obstructs the model from extracting disguisable low level features that can help in breed classification such as eyes, nose and ears.

II. Wrongly annotated images




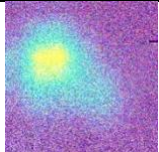

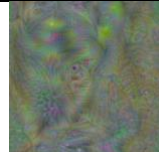



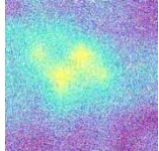





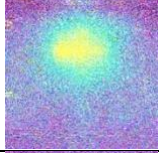




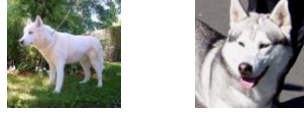
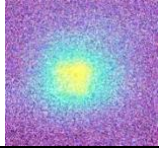



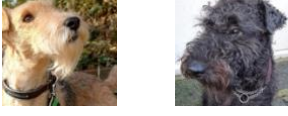
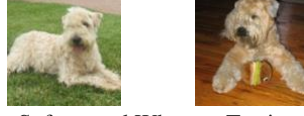
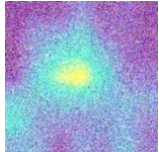

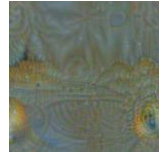
Some misclassifications are caused by wrongly annotated images. Taking sample 2 in Table 5-5 as an example, the sample image is more likely to be a French Bulldog instead of a Barbancon Griffon. According to the official standards published by the American Kennel Club (American Kennel Club, 1990; American Kennel Club, 2018), a Barbancon Griffon should have ears that are either cropped or semi-erected ears, while a French Bulldog has bat-like ears that are erected and with round top. Looking at the sample image, the ears of the dog assemble more closely to the description of a French Bulldog. Hence, the image is likely to be wrongly-annotated and the model prediction is actually accurate.

III. Insufficient samples of poses

Lastly, the insufficient samples for different poses also results in misclassifications. In sample 3 in Table 5-5, the Appenzeller is jumping and hence its ears appears to be erected instead of dropped like those in the illustrative images. After performing a scrutinisation of the dataset, it is found that the proportion of Border Collies with erected ears is much higher than the Appenzellers with erected ears. As other parts of the breeds are visually similar, the ears are likely to be one of the distinguishable feature for distinguishing the differences between an Appenzeller and a Border Collie, and the erected ears in the image might therefore misled the model. The lack of poses is hard to be compensated through image augmentation as the images are generated based on the existing images. Hence, more data with a wider variety of poses has to be collected in order to mitigate this cause of misclassification.

Mobile application for dog breed classification using deep learning and transfer learning

Table 5-5: (Breed FGIC model) Overview of the worst predictions made by the final model

#	Sample Image	Correct Breed (Rank of confidence value)	Top-1 Predicted Breed (Rank of confidence value)	# Samples in Training Set (% in Total Training Set)		Visualisation		
				Correct Breed	Top-1 Predicted Breed	Grad-CAM	Activation Maximisation (Correct Breed)	Activation Maximisation (Top-1 Predicted Breed)
1		 Whippet (87/120)	 Briard (1/120)	148 (0.90%)	126 (0.77%)			
2		 Brabancon Griffon (76/120)	 French Bulldog (1/120)	124 (0.75%)	134 (0.81%)			
3		 Appenzeller (37/120)	 Border Collie (1/120)	124 (0.75%)	122 (0.74%)			
4		 Labrador Retriever (20/120)	 Siberian Husky (1/120)	137 (0.83%)	150 (0.91%)			
5		 Lakeland Terrier (15/120)	 Soft-coated Wheaten Terrier (1/120)	147 (0.89%)	122 (0.74%)			

5.2 iOS mobile application

5.2.1 Comparison with current market leading products

Accuracy

The final mobile application is compared against current market leading products on AppStore on the same set of images used in the experiment in section 2.3. Figure 5-14 summarises the Top-1 and Top-5 accuracies, as well as average inference speed of all the three existing applications and the final application, tested on the same iPhone 13 device. The final application is superior over all the other three applications in terms of accuracy. It is also the only application that can achieve 100% Top-5 accuracy.

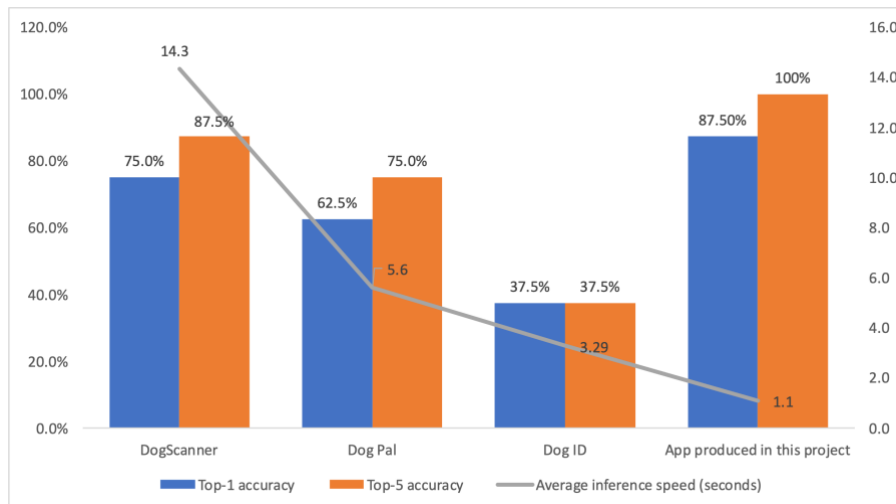


Figure 5-14: (Mobile app) Performance of the application produced in this project against market leading products

Storage requirement and inference time

In terms of storage requirement, the final application has the second largest size of 85.6 MB. Although Dog Pal and Dog ID have much smaller application sizes, their inference time are significantly slower than the final application. This might be a result of the choice of backbone model architecture, however, as the models behind the three existing products are not publicly available, analysis cannot be performed.

6 Conclusion and Future work

Conclusion

In this project, an iOS mobile application for the fine-grained dog breed classification problem is developed. A variety of CNN and transformer models pretrained on ImageNet are fine-tuned on the Stanford Dogs dataset and their performances are compared to select the best backbone model architecture for the application. From literatures and results of the fine-tuned models, transformer models are able to achieve better accuracy than CNN models on the Stanford Dogs dataset. However, their enormous model size make them a suboptimal choice for mobile deployment. With considerations on accuracy, model size and inference time, the final model architecture chosen for the application is EfficientNetV2B which achieves over 90% Top-1 accuracy.

To improve user experience and perform data validation on user's input, an additional model is developed to detect whether there is dog present in the image selected by user. Due to the simpler nature of the task, it is decided to leverage the lightweight MobileNetV2 model architecture pretrained on ImageNet, and fine-tune on a dataset with images from Stanford Dogs and Caltech256. The final model achieves high accuracy of over 99%, with rare cases of false negative predictions which are remediated with messages and checks on image quality in the application.

The final mobile application is developed in Xcode with the two models embedded and tested on a iPhone 13. The final application is superior over other market leading dog breed classification applications on AppStore in terms of accuracy, storage requirement and inference speed. However, it supports less breeds than one of the application due to the lack of publicly available data. The accuracy on cross-breed dogs is undetermined due to the same reason.

Future work

Future work can be done on exploring model compression techniques on transformer models. From the comparative study on the best performing model on the Stanford Dogs dataset, transformer model achieves state-of-the-art results. Although there have been works that successfully reduce the size of transformer models significantly while achieving better accuracies than CNN models by combining CNN and transformer models, the long inference time remains a major challenge for the practicality of deployment on resource-limited mobile

devices. By exploring model compression techniques on transformer or hybrid transformer models, the dog breed classification model might achieve better accuracy with acceptable sacrifice on efficiency.

Besides model compression, cloud deployment of deep learning models can also be considered to reduce model size and reduce storage burden on mobile devices. As mobile devices are resource-limited and highly accurate deep learning models are often computationally-intensive, offloading storing of deep learning models and inference tasks to powerful cloud servers would allow higher flexibility in the choice of model architecture and still achieve high accuracy and efficient inference. However, having a cloud architecture would increase the cost of maintain the application and the efficiency would be dependent on internet connection.

In terms of the variety of breeds supported in the application, further training of the model can be done by combining the Stanford Dogs dataset and the Tsinghua Dogs dataset. Incremental training on the deep learning models with user submitted images and feedback collected from user on the prediction results would also allow continuing improvement in the variety of dog breeds supported and accuracy of prediction.

Lastly, more fine-tuning on the dog detector model can be performed to further improve the classification accuracy between dogs and other objects. For instance, the proportion of object samples in the training data might be fine-tuned based on their similarity with dogs. For objects with more similar features as dogs, a larger proportion of images could be included, vice versa. This might help improve accuracy of the model with smaller impact on training time. Other lightweight model architectures such as EfficientNet and DenseNet can also be explored.

(Word count: 9855)

7 Bibliography

- Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). KAZE Features. *Computer Vision – ECCV 2012*, 214–227. [online] Available at: https://doi.org/10.1007/978-3-642-33783-3_16 [Accessed 17 Aug. 2022].
- American Kennel Club. (1990). *Official Standard of the Brussels Griffon*. [online] Available at: <https://bit.ly/3JZwR3C> [Accessed 17 Aug. 2022].
- American Kennel Club. (2018). *Official Standard of the French Bulldog*. [online] Available at: <https://bit.ly/3QyweAG> [Accessed 17 Aug. 2022].
- American Kennel Club (n.d.). *American Kennel Club*. American Kennel Club. [online] Available at: <https://www.akc.org/> [Accessed 17 Aug. 2022].
- Anwar, S., Barnes, N. and Petersson, L. (2021). *A Systematic Evaluation: Fine-Grained CNN vs. Traditional CNN Classifiers*. [online] Available at: <https://arxiv.org/pdf/2003.11154.pdf> [Accessed 17 Aug. 2022].
- App Store. (n.d. -a). *Dog Pal: Dog Scanner, Breed ID*. [online] Available at: <https://apple.co/3Ly5PjI> [Accessed 17 Aug. 2022].
- App Store. (n.d. -b). *Dog Scanner*. [online] Available at: <https://apple.co/3IDYoFp> [Accessed 17 Aug. 2022].
- App Store. (2017). *Dog ID - Dog Breed Identifier*. [online] Available at: <https://apple.co/3MKftQF> [Accessed 17 Aug. 2022].
- Apple. (n.d.-a). *Apple Developer Documentation*. Developer.apple.com. [online] Available at: <https://apple.co/3A23N6X> [Accessed 17 Aug. 2022].
- Apple. (n.d.-b). *Core ML Tools*. GitHub. [online] Available at: <https://github.com/apple/coremltools> [Accessed 17 Aug. 2022].
- Apple Developer. (n.d.). *Classifying Images with Vision and Core ML*. Developer.apple.com. [online] Available at: <https://apple.co/2JhbBWv> [Accessed 17 Aug. 2022].
- Ba, L.J. and Caruana, R. (2013). *Do Deep Nets Really Need to be Deep?* [online] Available at: <https://bit.ly/3OONscz> [Accessed 17 Aug. 2022].

- Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, [online] 110(3), pp.346–359. doi:10.1016/j.cviu.2007.09.014.
- Bosch, A., Zisserman, A. and Munoz, X. (2007). Image Classification using Random Forests and Ferns. *2007 IEEE 11th International Conference on Computer Vision*. doi:10.1109/iccv.2007.4409066.
- Castanyer, R., Martínez-Fernández, S. and Franch, X. (2021). *Integration of Convolutional Neural Networks in Mobile Applications*. [online] Available at: <https://arxiv.org/pdf/2103.07286.pdf> [Accessed 17 Aug. 2022].
- Chandrasekaran, R. (2019). *Dog Breed Identification using Deep Learning*. Medium. [online] Available at: <https://bit.ly/3iBOidQ> [Accessed 17 Aug. 2022].
- Chen, W.-C., Chang, C.-C., Lu, C.-Y. and Lee, C.-R. (2018). *Knowledge Distillation with Feature Maps for Image Classification*. [online] Available at: <https://arxiv.org/pdf/1812.00660.pdf> [Accessed 17 Aug. 2022].
- Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2020). A Survey of Model Compression and Acceleration for Deep Neural Networks. *IEEE Signal Processing Magazine, Special issue on deep learning for image understanding*. [online] Available at: <https://arxiv.org/pdf/1710.09282.pdf> [Accessed 17 Aug. 2022].
- Chennupati, K. and Cheng, C. (2021). *Adaptive Distillation: Aggregating Knowledge from Multiple Paths for Efficient Distillation*. [online] Available at: <https://bit.ly/37SW6FH> [Accessed 17 Aug. 2022].
- Conde, M. V., & Turgutlu, K. (2021). Exploring Vision Transformers for Fine-grained Classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2021 - FGVC8*. [online] Available at: <https://bit.ly/3zYZk53> [Accessed 17 Aug. 2022].

- Croy, K.C., Levy, J.K., Olson, K.R., Crandall, M. and Tucker, S.J. (2012). What kind of dog is that? Accuracy of dog breed assessment by canine stakeholders. *5th Annual Maddie's Shelter Medicine Conference, Orlando, USA*. [online] Available at: <https://bit.ly/37ZCkZn> [Accessed 17 Aug. 2022].
- Csurka, G., Dance, C. R., Fan, L., & Willamowski, J. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*.
- Dubey, A., Gupta, O., Guo, P., Raskar, R., & Farrell, R. (2018). Pairwise Confusion for Fine-Grained Visual Classification. *Springer International Publishing*. [online] Available at: <https://bit.ly/3w9uxBo> [Accessed 17 Aug. 2022].
- Dai, Z., Liu, H., Le, Q. and Tan, M. (2021). *CoAtNet: Marrying Convolution and Attention for All Data Sizes*. [online] Available at: <https://arxiv.org/pdf/2106.04803.pdf> [Accessed 17 Aug. 2022].
- Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1.
- de Campos, T., Csurka, G. and Perronnin, F. (2012). Images as sets of locally weighted features. *Computer Vision and Image Understanding*, 116(1), pp.68–85.
doi:10.1016/j.cviu.2011.07.011.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2020). *An image is worth 16X16 words: Transformer for image recognition at scale*. [online] Available at: <https://arxiv.org/pdf/2010.11929.pdf> [Accessed 17 Aug. 2022].
- d'Ascoli, S., Touvron, H., Leavitt, M., Morcos, A., Biroli, G. and Sagun, L. (2021). *ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases*. [online] Available at: <https://arxiv.org/pdf/2103.10697.pdf> [Accessed 17 Aug. 2022].
- Gao, H., Chen, W., & Dou, L. (2015). *Image classification based on support vector machine and the fusion of complementary features*. [online] Available <https://arxiv.org/pdf/1511.01706.pdf> [Accessed 17 Aug. 2022].

- Gou, J., Yu, B., Maybank, S.J. and Tao, D. (2021). Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* [online] Available at: <https://arxiv.org/pdf/2006.05525.pdf> [Accessed 17 Aug. 2022].
- Guo, J., Han, K., Wu, H., Xu, C., Tang, Y., Xu, C. and Wang, Y. (2021). *CMT: Convolutional Neural Networks Meet Vision Transformers.* [online] Available at: <https://arxiv.org/pdf/2107.06263.pdf> [Accessed 17 Aug. 2022].
- Gunter, L.M., Barber, R.T. and Wynne, C.D.L. (2018). A canine identity crisis: Genetic breed heritage testing of shelter dogs. *PLOS ONE*, 13(8), p.e0202633.
- He, K., Zhang, X., Ren, S. and Sun, J. (2015). *Deep Residual Learning for Image Recognition.* [online] Available at: <https://arxiv.org/pdf/1512.03385.pdf> [Accessed 17 Aug. 2022].
- He, J., Chen, J.-N., Liu, S., Kortylewski, A., Yang, C., Bai, Y., & Wang, C. (2022). TransFG: A Transformer Architecture for Fine-Grained Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1), 852–860. [online] Available at: <https://doi.org/10.1609/aaai.v36i1.19967> [Accessed 17 Aug. 2022].
- Hsu, S.-C., Chen, I-Chieh. and Huang, C.-L. (2015). Image classification using pairwise local observations based Naive Bayes classifier. *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. doi:10.1109/apsipa.2015.7415311.
- Hinton, G., Vinyals, O. and Dean, J. (2015). *Distilling the Knowledge in a Neural Network.* [online] Available at: <https://arxiv.org/pdf/1503.02531v1.pdf> [Accessed 17 Aug. 2022].
- Hugging Face. (2021, March 24). *google/vit-base-patch16-224-in21k* · Hugging Face. Huggingface.co. [online] Available at: <https://bit.ly/3c2awG6> [Accessed 17 Aug. 2022].
- Imran, A. and Athitsos, V. (2020). *Domain Adaptive Transfer Learning on Visual Attention Aware Data Augmentation for Fine-grained Visual Categorization.* [online] Available at: <https://bit.ly/374kmEY> [Accessed 17 Aug. 2022].
- Keras. (n.d.). *Keras documentation: Keras Applications.* Keras.io. [online] Available at: <https://keras.io/api/applications/> [Accessed 17 Aug. 2022].

- Jain, R., Singh, A., Jain, R. and Kumar, P. (2020). Dog Breed Classification Using Transfer Learning. *Proceedings of the Third International Conference on Computational Intelligence and Informatics*, pp.579–590.
- Khan, A., Sohail, A., Zahoor, U. and Qureshi, A.S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*.
- Khosla, A., Jayadevaprakash, N., Yao, B. and Li, F.-F. (2012). *Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs*. [online] Available at: <https://bit.ly/3LwSVCJ> [Accessed 17 Aug. 2022].
- Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp.84–90.
- LaRow, W., Mittl, B. and Singh, V. (2016). Dog Breed Identification. [online] Available at: <https://stanford.io/3wBmX3m> [Accessed 17 Aug. 2022].
- Lazebnik, S., Schmid, C., & Ponce, J. (2006). *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. [online] Available at: https://inc.ucsd.edu/mplab/users/marni/Igert/Lazebnik_06.pdf [Accessed 17 Aug. 2022].
- Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp.2278–2324.
- Lin, T.-Y., Roychoudhury, A. and Maji, S. (2015). *Bilinear CNN Models for Fine-grained Visual Recognition*. [online] Available at: <https://bit.ly/3Lupz7S> [Accessed 17 Aug. 2022].
- Liu, T., Alibhai, S., Wang, J., Liu, Q., He, X. and Wu, C. (2019). *Exploring Transfer Learning to Reduce Training Overhead of HPC Data in Machine Learning*. IEEE Xplore. [online] Available at: <https://bit.ly/3vQT90J> [Accessed 17 Aug. 2022].
- Lowe, D. (1999). *Object Recognition from Local Scale-Invariant Features*. [online] Available at: <https://www.cs.ubc.ca/~lowe/papers/iccv99.pdf> [Accessed 17 Aug. 2022].
- Luo, W., Zhang, H., Li, J., & Wei, X.-S. (2020). Learning Semantically Enhanced Feature for Fine-Grained Image Classification. *IEEE Signal Processing Letters*, 27, 1545–1549. [online] Available at: <https://doi.org/10.1109/lsp.2020.3020227> [Accessed 17 Aug. 2022].

- Mahendran, A., & Vedaldi, A. (2016). Visualizing Deep Convolutional Neural Networks Using Natural Pre-images. *International Journal of Computer Vision*, 120(3), 233–255. [online] Available at: <https://doi.org/10.1007/s11263-016-0911-8> [Accessed 17 Aug. 2022].
- Mansoori, N. S., Nejati, M., Razzaghi, P., & Samavi, S. (2013). Bag of visual words approach for image retrieval using color information. *2013 21st Iranian Conference on Electrical Engineering (ICEE)*. [online] Available at: <https://doi.org/10.1109/iranianicee.2013.6599562> [Accessed 17 Aug. 2022].
- Mehta, S., & Rastegari, M. (2021). MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. *ArXiv*. [online] Available at: <https://doi.org/10.48550/ARXIV.2110.02178> [Accessed 17 Aug. 2022].
- Menghani, G. (2021). Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. *ArXiv*. [online] Available at: <https://doi.org/10.48550/ARXIV.2106.08962> [Accessed 17 Aug. 2022].
- Mishra, R., Gupta, H. P., & Dutta, T. (2020). A Survey on Deep Neural Network Compression: Challenges, Overview, and Solutions. *ArXiv*. [online] Available at: <https://doi.org/10.48550/ARXIV.2010.03954> [Accessed 17 Aug. 2022].
- NowGaming. (n.d.). *Dog ID - Dog Breed Identifier by Allen Tom*. [online] Available at: <https://bit.ly/3y3OH4> [Accessed 17 Aug. 2022].
- O'Hara, S., & Draper, B. A. (2011). Introduction to the Bag of Features Paradigm for Image Classification and Retrieval. *ArXiv*.
- Pan, S.J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, [online] 22(10), pp.1345–1359. [online] Available at: <https://bit.ly/3s10MjY> [Accessed 17 Aug. 2022].
- Ripley, K. (2017). *Can You Tell These Dog Breed Look-Alikes Apart?* [online] American Kennel Club. [online] Available at: <https://bit.ly/3JzKci2> [Accessed 17 Aug. 2022].
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, 128(2), 336–359. [online] Available at: <https://doi.org/10.1007/s11263-019-01228-7> [Accessed 17 Aug. 2022].
- Sitaula, C. and Aryal, S. (2021). New bag of deep visual words based features to classify chest x-ray images for COVID-19 diagnosis. *Health Information Science and Systems*, 9(1). doi:10.1007/s13755-021-00152-w.
- Siwalu Software. (n.d.). *Siwalu - AI-based image recognition to identify animals*. [online] Available at: <https://siwalusoftware.com/> [Accessed 27 Apr. 2022].
- Song, Y., Sebe, N., & Wang, W. (2022). On the Eigenvalues of Global Covariance Pooling for Fine-grained Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1. [online] Available at: <https://doi.org/10.1109/tpami.2022.3178802> [Accessed 17 Aug. 2022].
- Stanford Vision and Learning Lab. (2011). *Stanford Dogs dataset for Fine-Grained Visual Categorization*. [online] Available at: <http://vision.stanford.edu/aditya86/ImageNetDogs/> [Accessed 26 Apr. 2022].
- The Dog API. (n.d.). *The Dog API - Dogs as a Service*. Thedogapi.com. [online] Available at: <https://thedogapi.com/>
- Tu, X., Lai, K. and Yanushkevich, S. (2018). *Transfer Learning on Convolutional Neural Networks for Dog Identification*. [online] IEEE Xplore. Available at: <https://bit.ly/3iCPYDP> [Accessed 24 Mar. 2022].
- Uno, M., Han, X.-H. and Chen, Y.-W. (2018). *Comprehensive Study of Multiple CNNs Fusion for Fine-Grained Dog Breed Categorization*. [online] IEEE Xplore. Available at: <https://bit.ly/3NmMuDC> [Accessed 24 Mar. 2022].
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Brain, G., Research, G., Jones, L., Gomez, A., Kaiser, L. and Polosukhin, I. (2017). *Attention Is All You Need*. [online] Available at: <https://arxiv.org/pdf/1706.03762.pdf> [Accessed 17 Aug. 2022].

- Wang, R., Ding, K., Yang, J. and Xue, L. (2016). A novel method for image classification based on bag of visual words. *Journal of Visual Communication and Image Representation*, 40, pp.24–33. doi:10.1016/j.jvcir.2016.05.022.
- Wang, Y., Wang, J., Zhang, W., Zhan, Y., Guo, S., Zheng, Q. and Wang, X. (2021). A survey on deploying mobile deep learning applications: A systemic and technical perspective. *Digital Communications and Networks*.
- Wang, Y. and Wang, Z. (2019). A survey of recent work on fine-grained image classification techniques. *Journal of Visual Communication and Image Representation*, 59, pp.210–214.
- Wei, X.-S., Song, Y.-Z., Aodha, O.M., Wu, J., Peng, Y., Tang, J., Yang, J. and Belongie, S. (2021). Fine-Grained Image Analysis with Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] pp.1–1. Available at: <https://arxiv.org/pdf/2111.06119.pdf> [Accessed 27 Apr. 2022].
- Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L. and Zhang, L. (2021). CvT: Introducing Convolutions to Vision Transformers. [online] Available at: <https://arxiv.org/pdf/2103.15808.pdf> [Accessed 29 Apr. 2022].
- Xie, E., Li, G. and Liu, W. (2018). *Improving Fine-Grained Object Classification Using Adversarial Generated Unlabelled Samples*. [online] IEEE Xplore. Available at: <https://bit.ly/3OJhpug> [Accessed 29 Apr. 2022].
- Yang, S., Bo, L., Wang, J. and Shapiro, L. (2012). *Unsupervised Template Learning for Fine-Grained Object Recognition*. [online] Available at: <https://bit.ly/3xZSr49> [Accessed 28 Apr. 2022].
- Yao, B., Khosla, A. and Li. (2011). *Combining Randomization and Discrimination for Fine-Grained Image Categorization*. [online] Available at: <https://stanford.io/3OYVEHI> [Accessed 28 Apr. 2022].
- Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014). *How transferable are features in deep neural networks?* [online] Available at: <https://arxiv.org/pdf/1411.1792.pdf> [Accessed 17 Aug. 2022].

- Zhao, B., Feng, J., Wu, X. and Yan, S. (2017). A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2), pp.119–135.
- Zhao, Z.-Q., Zheng, P., Xu, S.-T. and Wu, X. (2018). *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS FOR PUBLICATION 1 Object Detection with Deep Learning: A Review*. [online] Available at: <https://arxiv.org/pdf/1807.05511.pdf> [Accessed 17 Aug. 2022].
- Zheng, M., Li, Q., Geng, Y., Yu, H., Wang, J., Gan, J. and Xue, W. (2018). *A Survey of Fine-Grained Image Categorization*. [online] IEEE Xplore. Available at: <https://bit.ly/36ZNcWW> [Accessed 28 Apr. 2022].
- Zhuang, P., Wang, Y., & Qiao, Y. (2020). Learning Attentive Pairwise Interaction for Fine-Grained Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07), 13130–13137. [online] Available at: <https://doi.org/10.1609/aaai.v34i07.7016> [Accessed 17 Aug. 2022].
- Zou, D.-N., Zhang, S.-H., Mu, T.-J. and Zhang, M. (2020). A new dataset of dog breed images and a benchmark for finegrained classification. *Computational Visual Media*, 6(4), pp.477–487.

Appendix

Appendix I: Results of experiment on the existing mobile applications

As Stanford Dogs dataset is a popular fine-grained dog breed classification dataset, it is believed that the dataset was used in the development of these mobile applications, hence, the following images from the American Kennel Club website and personal sources are used instead to better reflect the mobile applications' performances on unseen test data.

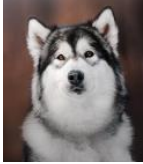





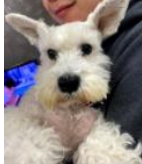

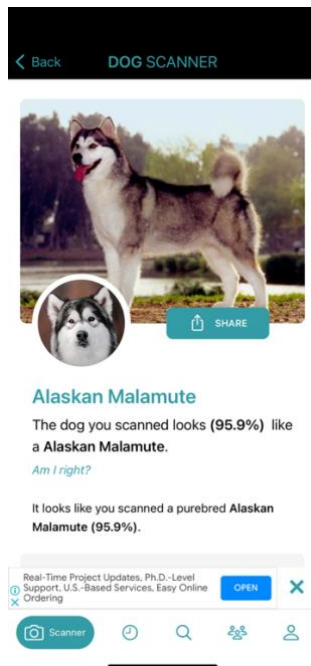
	Image 1: Alaskan Malamute	Image 2: Siberian Husky	Image 3: Whippet	Image 4: Italian Greyhound	Image 5: Golden Labrador	Image 6: Brown Labrador	Image 7: Miniature Schnauzer	Image 8: Miniature Schnauzer with longer hair
								
DogScanner								
Time taken (sec)	14	14.5	14.5	14	14.5	14.5	14.35	14.3
Result 1	Alaska Malamute	Siberian Husky	Whippet	Whippet	Labrador Retriever	Labrador Retriever	Miniature Schnauzer	Silky Terrier
Confidence %	95.9	99	99	79.1	85.8	90.5	96.5	36.6
Result 2	-	-	-	Italian Greyhound	Golden Retriever	Cão de Castro Laboreiro	-	Yorkshire Terrier
Confidence %	-	-	-	11.4	9.8	6.5	-	24.9
Result 3	-	-	-	-	-	-	-	Portuguese Podengo
Confidence %	-	-	-	-	-	-	-	18.2
Dog Pal								
Time taken (sec)	5	6	5	5	5	5	7	6.9
Result 1	Alaska Malamute	Siberian Husky	Whippet	Whippet	Labrador Retriever	Labrador Retriever	West Highland White Terrier	West Highland White Terrier
Result 2	Siberian Husky	-	-	Poodle	-	-	Schnauzer	Maltese Dog
Result 3	Labrador Retriever	-	-	Mastiff	-	-	Maltese Dog	Bichon frise
Lookalike	-	Alaskan Klee Kai	Italian greyhound	-	Flat-coated retriever	Flat-coated retriever	-	-
Dog ID								
Time taken (sec)	4.88	2.67	2.99	3.16	3	3.65	3	3
Result 1	Husky	Husky	Great Dane	Hound	Labrador Retriever	Labrador Retriever	Terrier	Terrier
Confidence %	99.7	99	99.9	56.9	98	99.1	80.7	86.4
Result 2	-	-	-	-	-	-	Poodle	Puppy
Confidence %	-	-	-	-	-	-	58.3	70.6
Result 3	-	-	-	-	-	-	-	Poodle
Confidence %	-	-	-	-	-	-	-	58.5

Figure A1: Images uploaded for the experiment and the corresponding classification results of the three applications tested.

Appendix II: Screenshots of result pages of the existing mobile applications

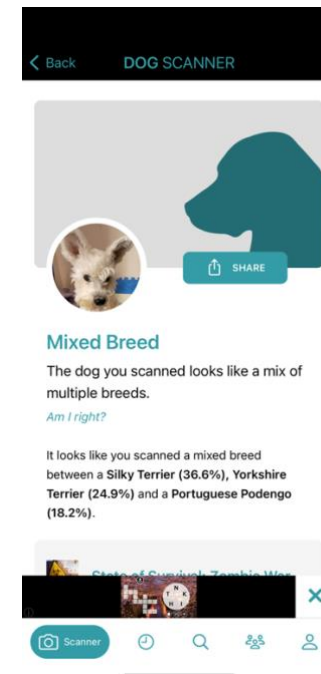
DogScanner



(a) One prediction.



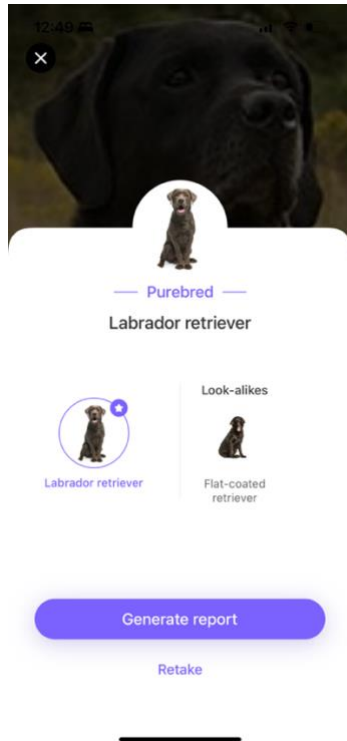
(b) More than one predictions.



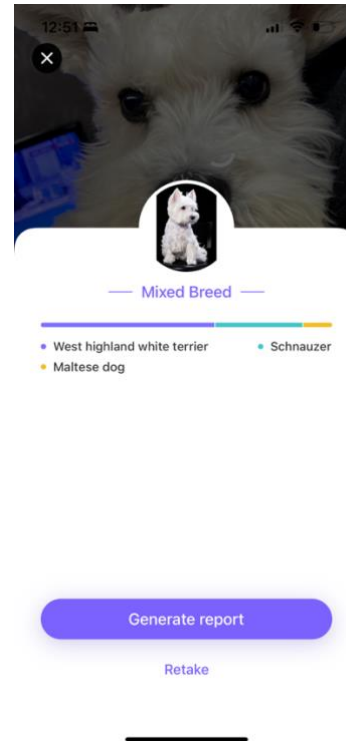
(c) More than two predictions
i.e. a mixed breed prediction

Figure A2: Result pages of DogScanner.

Dog Pal



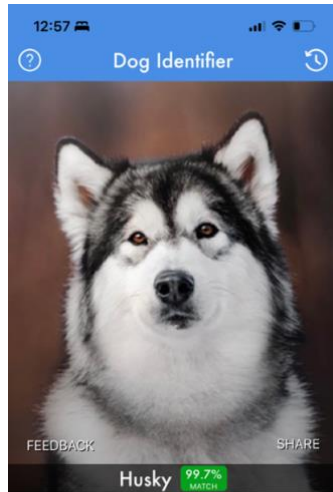
(a) Purebred prediction.



(b) Mixed breed prediction.

Figure A3: Result pages of Dog Pal.

Dog ID



Choose Another Photo

(a) One prediction.



Choose Another Photo







(b) More than one predictions.


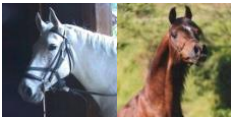

Figure A4: Result pages of Dog ID.







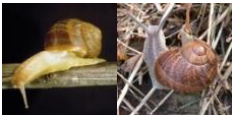
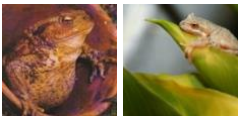
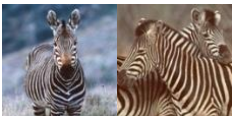
Appendix III: (Dog detector model) List of land animals kept or removed in the final dataset

Table A1 lists all land animals in the Stanford Dogs dataset (). All four-legged mammals are excluded from the final “no_dog” dataset as some might be visually very similar to certain breeds of dogs and is likely to cause false negative predictions, which deteriorates the user experience more than false positive predictions as the classification will not take place if the dog detector detects no dog.

Table A1: (Dog detector model) List of land animals in the Stanford Dogs dataset

Class ID	Animal	Excluded from “no_dog” dataset?
007	Bat	No
		
009	Bear	Yes
		
028	Camel	Yes
		
038	Chimp	No
		
056	Dog	Yes
		
064	Elephant	Yes
		

	Elk	
065		Yes
	Frog	
080		No
	Giraffe	
084		Yes
	Goat	
085		Yes
	Gorilla	
090		No
	Greyhound	
254		Yes
	Horse	
105		Yes
	Iguana	
116		No
	Kangaroo	
121		Yes

	Llama		
134		Yes	
	Leopards		
129		Yes	
	Porcupine		
164		No	
	Raccoon		
168		Yes	
	Skunk		
186		Yes	
	Snake		
190		No	
	Snail		
189		No	
	Toad		
256		No	
	Zebra		
250		Yes	

Appendix IV: (Dog detector model) Original images and Grad-CAM visualisations of false positive predictions

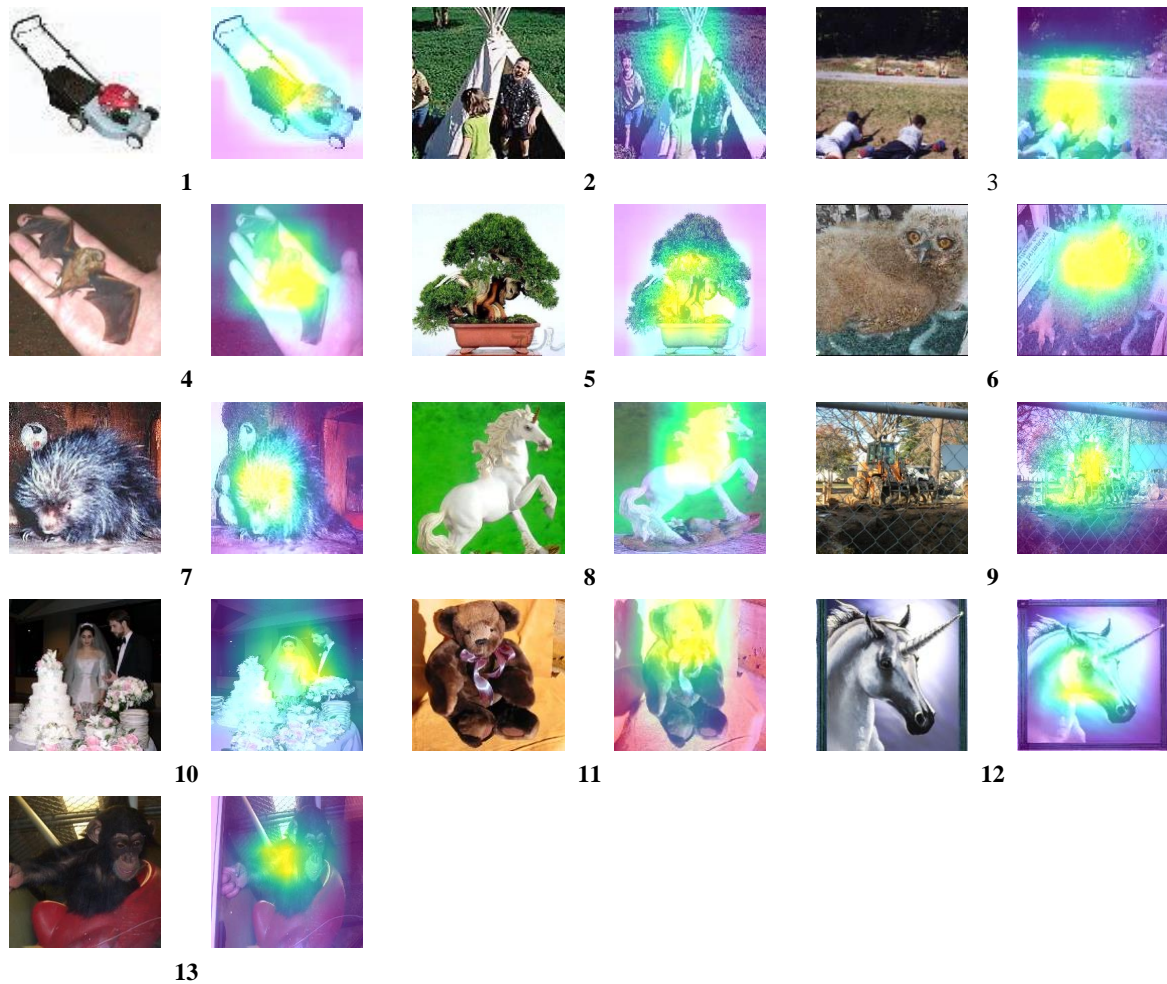


Figure A5: (Dog detector model) Original images and Grad-CAM visualisations of false positive samples

Appendix V: (Breed FGIC Model) Feature extraction training history of the 11 models explored

Comparison of Top-1 and Top-5 accuracies on the 11 models

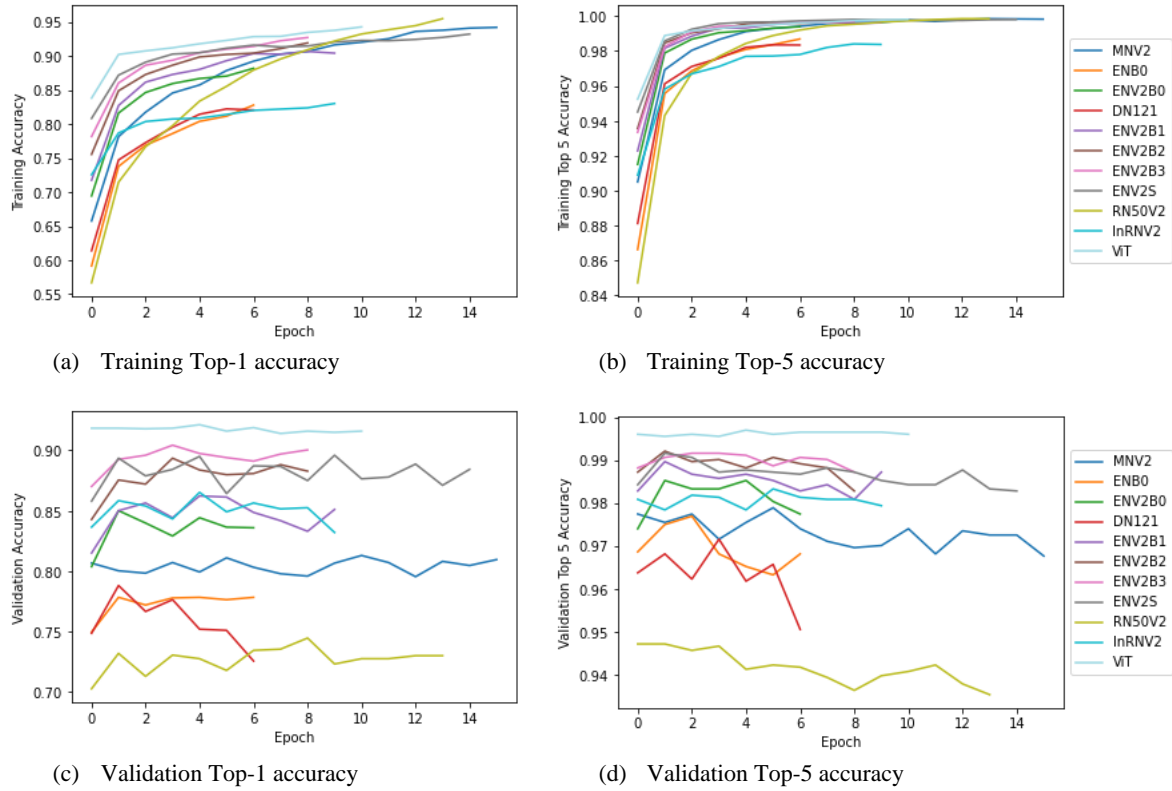
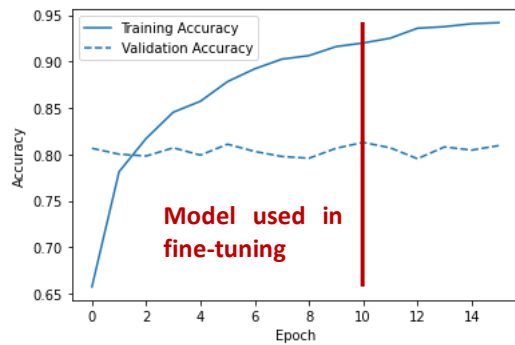
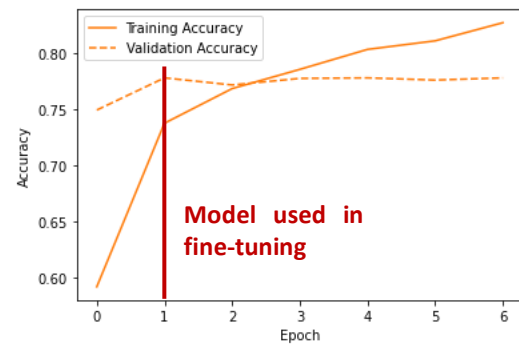


Figure A6: (Breed FGIC Model) Comparison of the Top-1 and Top5 training and validation accuracies on the 11 models during feature extraction

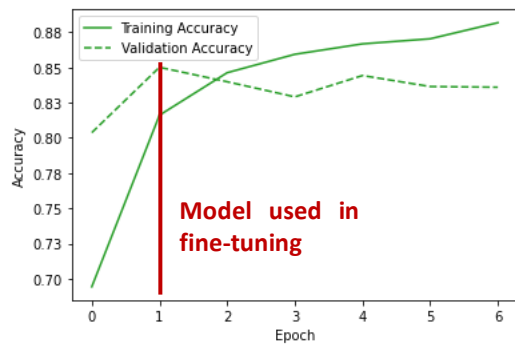
Training and validation accuracies of the 11 models respectively



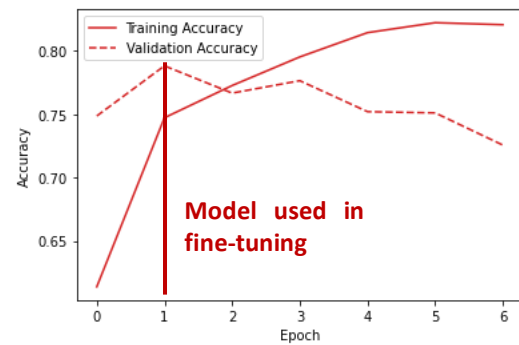
(a) MobileNetV2



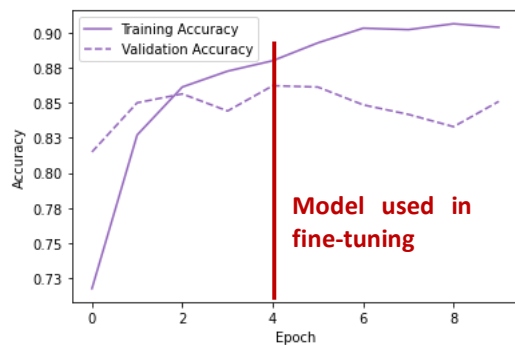
(b) EfficientNetB0



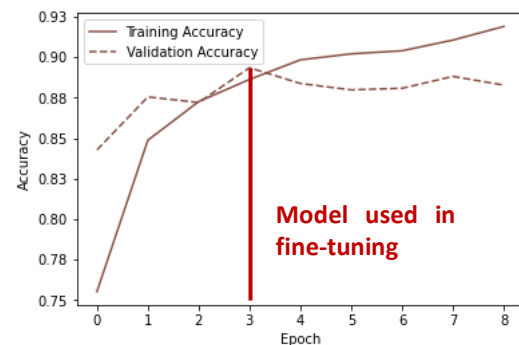
(c) EfficientNetV2B0



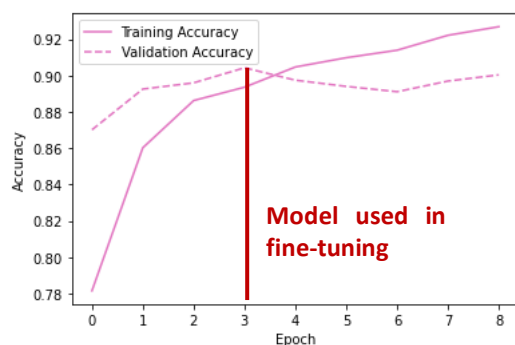
(d) DenseNet121



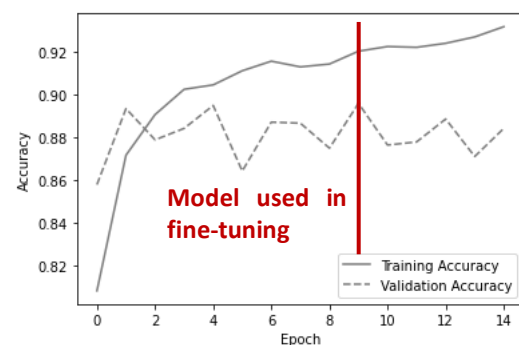
(e) EfficientNetV2B1



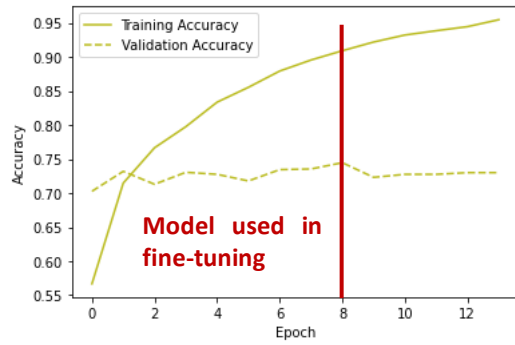
(f) EfficientNetV2B2



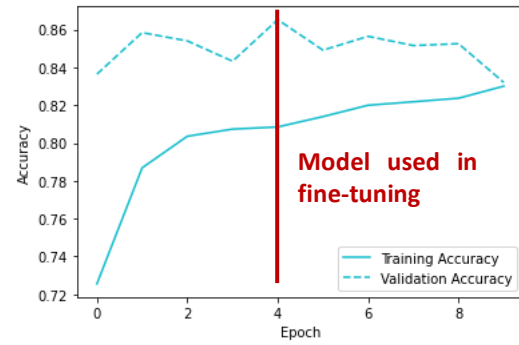
(g) EfficientNetV2B3



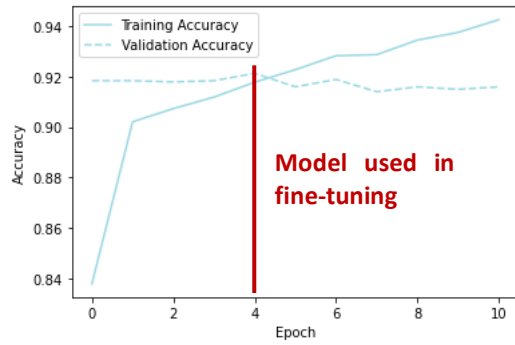
(h) EfficientNetV2S



(i) ResNet50V2



(j) InceptionResNetV2



(k) ViT

Figure A7: (Breed FGIC Model) History of training and validation accuracies of the 11 models respectively during feature extraction

Appendix VI: (Breed FGIC Model) Fine-tuning training history of the 11 models explored

Comparison of Top-1 and Top-5 accuracies on the 11 models

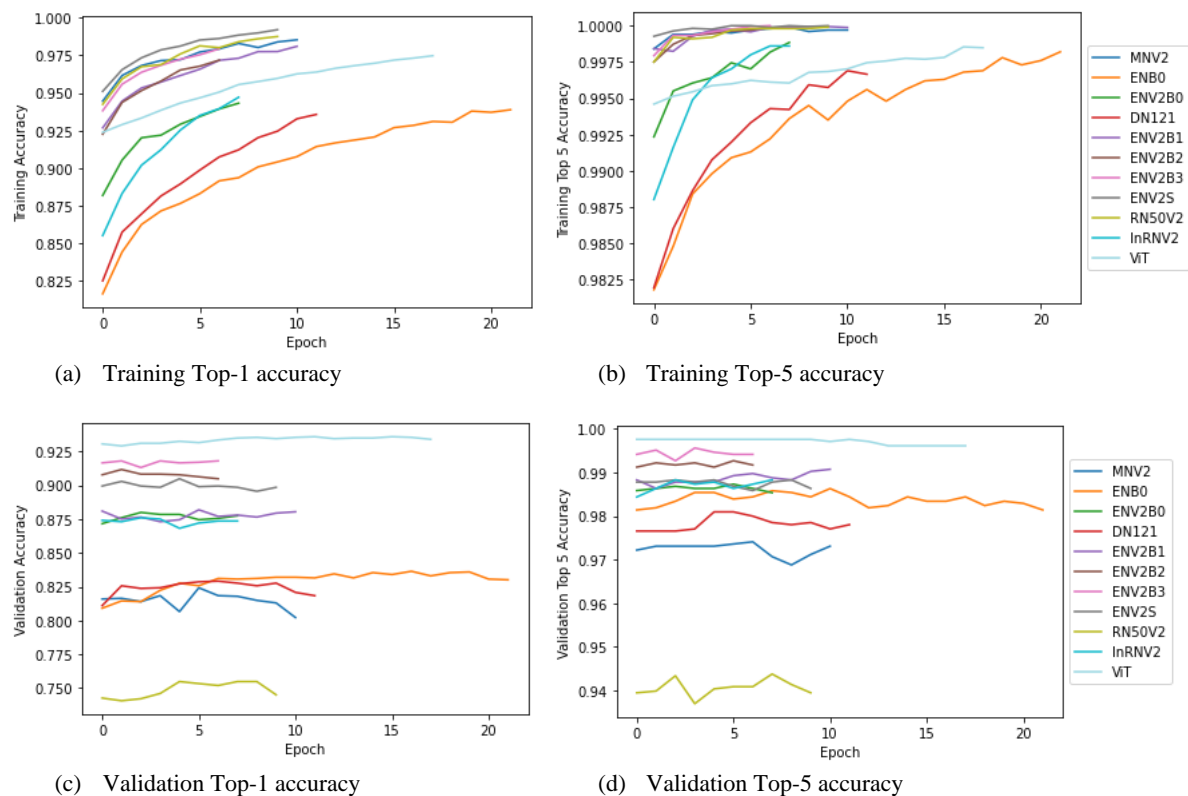
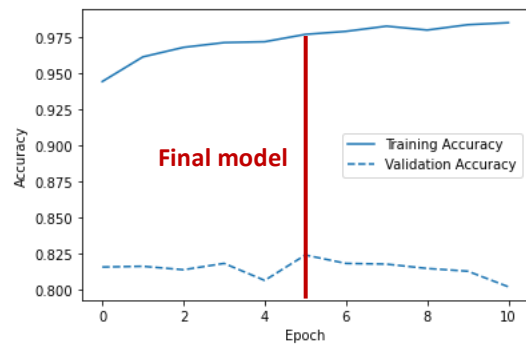
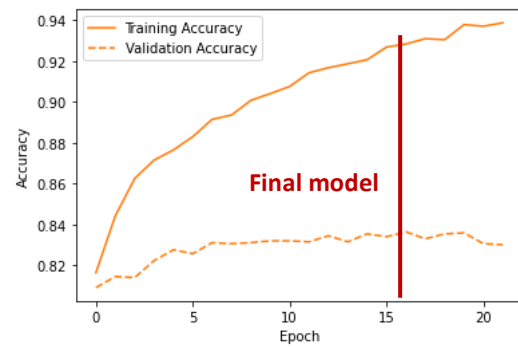


Figure A8: (Breed FGIC Model) Comparison of the Top-1 and Top5 training and validation accuracies on the 11 models during fine-tuning

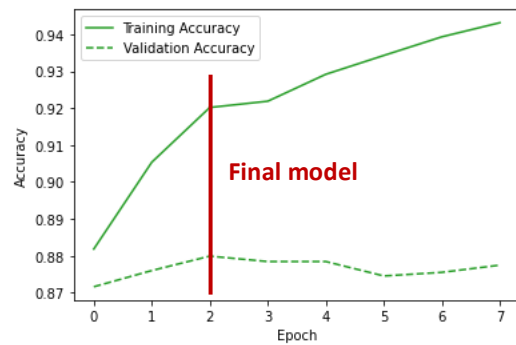
Training and validation accuracies of the 11 models respectively



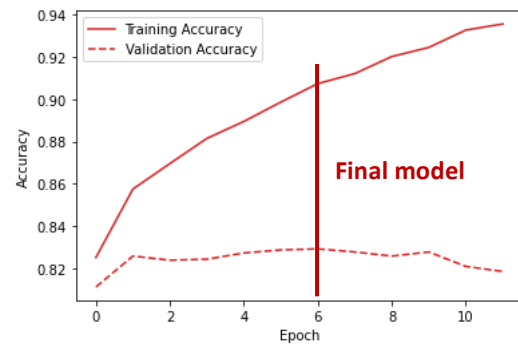
(a) MobileNetV2



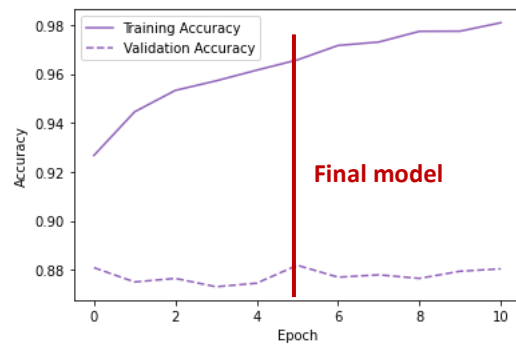
(b) EfficientNetB0



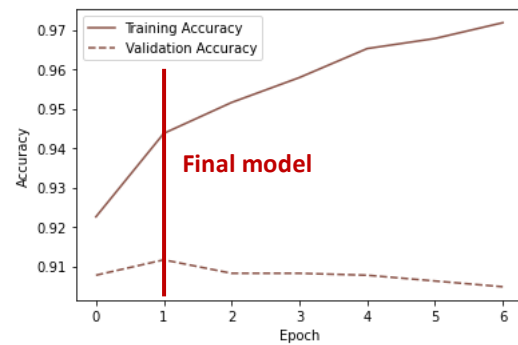
(c) EfficientNetV2B0



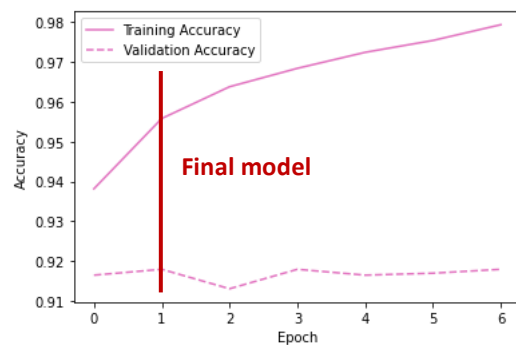
(d) DenseNet121



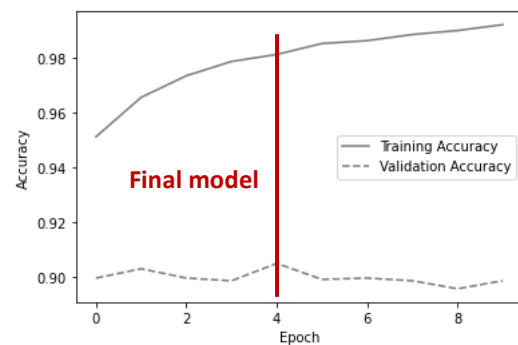
(e) EfficientNetV2B1



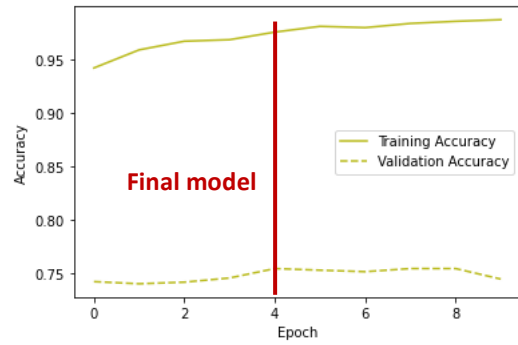
(f) EfficientNetV2B2



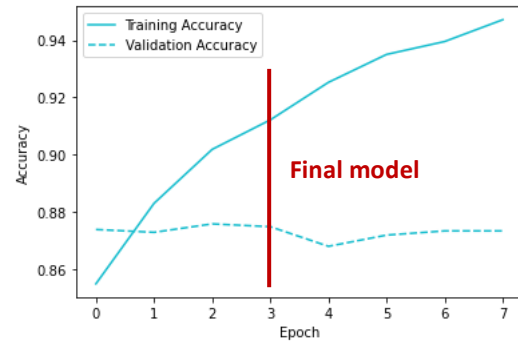
(g) EfficientNetV2B3



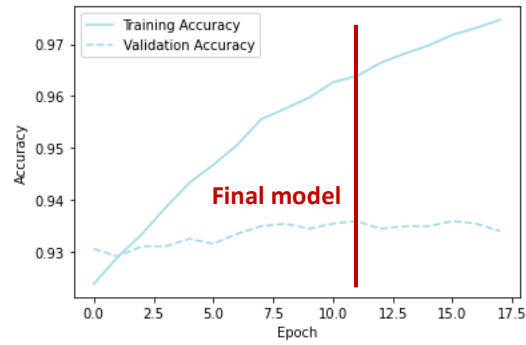
(h) EfficientNetV2S



(i) ResNet50V2



(j) InceptionResNetV2



(k) ViT

Figure A9: (Breed FGIC Model) History of training and validation accuracies of the 11 models respectively during fine-tuning

Appendix VII: (Breed FGIC Model) Confusion matrix of the final model on test set

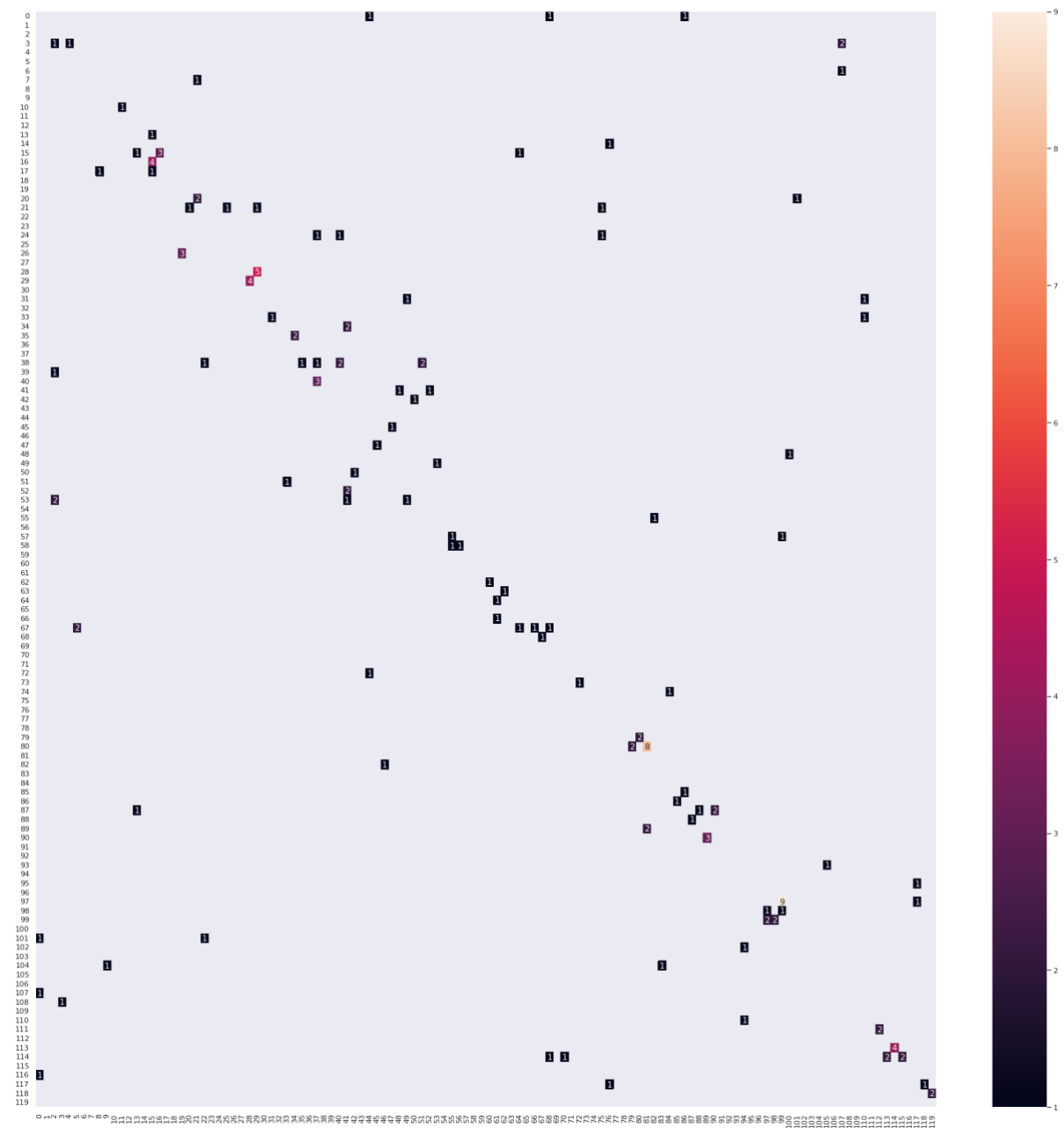


Figure A10: (Breed FGIC Model) Confusion matrix of the final model on test set

Table A2: (Breed FGIC Model) List of breeds

ID	Breed Name	ID	Breed Name
0	briard	60	bluetick
1	french_bulldog	61	weimaraner
2	bouvier_des_flandres	62	english_springer
3	silky_terrier	63	bedlington_terrier
4	shetland_sheepdog	64	australian_terrier
5	collie	65	giant_schnauzer
6	irish_terrier	66	miniature_schnauzer
7	newfoundland	67	eskimo_dog
8	bull_mastiff	68	border_collie
9	wire-haired_fox_terrier	69	appenzeller
10	boston_bull	70	pug
11	papillon	71	pomeranian
12	welsh_springer_spaniel	72	redbone
13	old_english_sheepdog	73	leonberg
14	german_shepherd	74	shih-tzu
15	airedale	75	lhasa
16	great_dane	76	yorkshire_terrier
17	pembroke	77	walker_hound
18	golden_retriever	78	kerry_blue_terrier
19	soft-coated_wheaten_terrier	79	blenheim_spaniel
20	staffordshire_bullterrier	80	siberian_husky
21	schipperke	81	dingo
22	brabancon_griffon	82	ibizan_hound
23	chesapeake_bay_retriever	83	miniature_poodle
24	greater_swiss_mountain_dog	84	japanese_spaniel
25	german_short-haired_pointer	85	beagle
26	cardigan	86	whippet
27	sealyham_terrier	87	toy_poodle
28	labrador_retriever	88	malinois
29	american_staffordshire_terrier	89	saluki
30	boxer	90	english_setter
31	saint_bernard	91	tibetan_mastiff
32	lakeland_terrier	92	maltese_dog
33	otterhound	93	border_terrier
34	norfolk_terrier	94	great_pyrenees
35	affenpinscher	95	bloodhound
36	pekinese	96	keeshond
37	vizsla	97	basenji
38	curly-coated_retriever	98	mexican_hairless
39	norwich_terrier	99	bernese_mountain_dog
40	flat-coated_retriever	100	dandie_dinmont

41	cocker_spaniel	101	kuvasz
42	rhodesian_ridgeback	102	sussex_spaniel
43	borzoi	103	miniature_pinscher
44	tibetan_terrier	104	scottish_deerhound
45	african_hunting_dog	105	standard_poodle
46	chihuahua	106	doberman
47	rottweiler	107	kelpie
48	malamute	108	groenendael
49	italian_greyhound	109	komondor
50	chow	110	irish_water_spaniel
51	irish_setter	111	gordon_setter
52	entlebucher	112	west_highland_white_terrier
53	cairn	113	samoyed
54	basset	114	english_foxhound
55	afghan_hound	115	brittany_spaniel
56	toy_terrier	116	black-and-tan_coonhound
57	scotch_terrier	117	norwegian_elkhound
58	irish_wolfhound	118	clumber
59	standard_schnauzer	119	dhole