

UNDERGRADUATE PROJECT PROGRESS REPORT

|  |  |
| --- | --- |
| **Project Title:** | **Ensemble Model for the Classification of Dog Breed** |
| **Surname:** | **Qiao** |
| **First Name:** | **Josephine** |
| **Student Number:** | **201918020201** |
| **Supervisor Name:** | **Dr. Grace Ugochi Nneji** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **May 5, 2023** |

**Chengdu University of Technology Oxford Brookes College**

**Chengdu University of Technology**

**BSc (Single Honours) Degree Project**

Programme Name: Projects

Module No.: CHC 6096

Surname: Qiao

First Name: Josephine

Project Title: Ensemble Model for the Classification of Dog Breed

Student No.: 201918020201

Supervisor: Dr. Grace Ugochi Nneji

Date submitted: May 5, 2023

*A report submitted as part of the requirements for the degree of BSc (Hons) in Software Engineering*

*At*

**Chengdu University of Technology Oxford Brookes College**

# **Declaration**

**Student** **Conduct** **Regulations**:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on

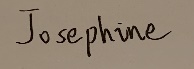
www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed on-line at

<https://www.brookes.ac.uk/students/sirt/student-conduct/>

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

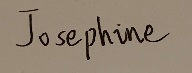
**I** **declare** **that** **I** **have** **read** **and** **understood** **Regulations** **C1.1.4** **of** **the** **Regulations** **governing** **Academic** **Misconduct,** **and** **that** **the** **work** **I** **submit** **is** **fully** **in** **accordance** **with** **them**.

Signature  Date: 05/05/2023

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Library.

**|** **agree** **that** **this** **dissertation** **may** **be** **available** **for** **reading** **and** **photocopying** **in** **accordance** **with** **the** **Regulations** **governing** **use** **of** **the** **Library.**

Signature  Date: 05/05/2023

# **Acknowledgment**

I want to express my heartfelt gratitude to my Supervisor, Dr. Grace Ugochi Nneji, for her guidance and support throughout my final undergraduate project. Her expertise, patience, and encouragement have been invaluable in shaping this project. Also, I would like to thank the module leader Joojo Walker and other teachers throughout my undergraduate study time for their teaching and advice.

More so, I would like to acknowledge the resources and facilities provided by Oxford Brookes University in collaboration with the Chengdu University of Technology for an outstanding opportunity.

And to my family and friends, your unending love and encouragement have been with me. I will always be grateful.

# **Table of Contents**

[**Declaration** i](#_Toc133857920)

[**Acknowledgment** ii](#_Toc133857921)

[**Table of Contents** iii](#_Toc133857922)

[**Table of Figures** vi](#_Toc133857923)

[**Table of Tables** ix](#_Toc133857924)

[**Abstract** x](#_Toc133857925)

[***Keywords*** x](#_Toc133857926)

[**Abbreviations** xi](#_Toc133857927)

[**Glossary** xii](#_Toc133857928)

[**Chapter 1 Introduction** 1](#_Toc133857929)

[1.1 Background 1](#_Toc133857930)

[1.2 Aim 1](#_Toc133857931)

[1.3 Convolutional neural network 2](#_Toc133857932)

[1.3.1 Input layer 2](#_Toc133857933)

[1.3.2 Convolutional layer (Conv layer) 2](#_Toc133857934)

[1.3.3 Pooling layer 3](#_Toc133857935)

[1.3.4 Fully-Connected layers (FC layers) 4](#_Toc133857936)

[1.3.5 Activation functions 5](#_Toc133857937)

[1.3.6 Loss functions 5](#_Toc133857938)

[1.4 Objectives 6](#_Toc133857939)

[1.5 Project Overview 6](#_Toc133857940)

[1.5.1 Scope 6](#_Toc133857941)

[1.5.2 Audience 7](#_Toc133857942)

[**Chapter 2 Background Review** 8](#_Toc133857943)

[2.1.1 CNN & Transfer learning 8](#_Toc133857944)

[2.1.2 Ensemble models 8](#_Toc133857945)

[**Chapter 3 Methodology** 10](#_Toc133857946)

[3.1 Dataset 10](#_Toc133857947)

[3.1.1 About Dataset 10](#_Toc133857948)

[3.1.2 Data Resizing 11](#_Toc133857949)

[3.1.3 Data labeling 11](#_Toc133857950)

[3.1.4 Data Augmentation 12](#_Toc133857951)

[3.2 Model Architecture 14](#_Toc133857952)

[3.3 Testing and Evaluation 17](#_Toc133857953)

[3.3.1 Data testing: 17](#_Toc133857954)

[3.3.2 Performance metrics 17](#_Toc133857955)

[3.4 Technology 19](#_Toc133857956)

[3.5 Project Version Management 19](#_Toc133857957)

[**Chapter 4 Experiments and results** 20](#_Toc133857958)

[4.1 The first stage of the experiment 20](#_Toc133857959)

[4.1.1 Implementation design for each model 20](#_Toc133857960)

[4.1.2 Results and analysis 25](#_Toc133857961)

[4.2 The second stage of the experiment 32](#_Toc133857962)

[4.2.1 Experimental results for two different data augmentation parameters 32](#_Toc133857963)

[4.2.2 Summary of data augmentation experiments 36](#_Toc133857964)

[4.3 The third stage of the experiment 38](#_Toc133857965)

[4.3.1 70 dog breeds VS 120 dog breeds 38](#_Toc133857966)

[4.3.2 The best model’s detailed results 40](#_Toc133857967)

[4.4 The deployment of the best model 43](#_Toc133857968)

[**Chapter 5 Professional Issues** 46](#_Toc133857969)

[5.1 Project Management 46](#_Toc133857970)

[5.1.1 Activities 46](#_Toc133857971)

[5.1.2 Schedule 47](#_Toc133857972)

[5.1.3 Project Data Management 47](#_Toc133857973)

[5.1.4 Project Deliverables 48](#_Toc133857974)

[5.2 Risk Analysis 48](#_Toc133857975)

[5.3 Professional Issues 49](#_Toc133857976)

[**Chapter 6 Conclusion** 52](#_Toc133857977)

[**References** 53](#_Toc133857978)

# **List of Figures**

* + [Figure 1‑1: An example of dog breed identification 2](#_Toc133853535)
  + [Figure 1‑2: An example of operations in the Conv layer 3](#_Toc133853536)
  + [Figure 1‑3: The operation example of Max-pooling 4](#_Toc133853537)
  + [Figure 1‑4: The operation example of Mean-pooling 4](#_Toc133853538)
  + [Figure 1‑5: The Rectified Linear Unit (ReLU) activation function produces 0 as an output when x < 0, and then produces a linear with slope of 1 when x > 0[11] 5](#_Toc133853539)
  + [Figure 3‑1: Distribution of the images stored in each class of the dataset 10](#_Toc133853540)
  + [Figure 3‑2: Information on the new version "Dog Breed Identification" dataset 11](#_Toc133853541)
  + [Figure 3‑3: The labeling of the "70 Dog Breeds-Image Data Set" 12](#_Toc133853542)
  + [Figure 3‑4: The labeling of the new version "Dog Breed Identification" dataset 12](#_Toc133853543)
  + [Figure 3‑5[26]: Residual learning: a building block 14](#_Toc133853544)
  + [Figure 3‑6[14]: A standard VGG16 model’s architecture 15](#_Toc133853545)
  + [Figure 3‑7[29]: The InceptionV3 model's architecture. The detail of blocks 1∼4 are shown in Figure 3-7(a), Figure 3-7(b), Figure 3-7(c) and Figure 3-7(d). 15](#_Toc133853546)
  + [Figure 3‑8: The proposed architecture of this project 16](#_Toc133853547)
  + [Figure 4‑1: VGG16 model process 24](#_Toc133853548)
  + [Figure 4‑2: ResNet50 model process 25](#_Toc133853549)
  + [Figure 4‑3: InceptionV3 model process 25](#_Toc133853550)
  + [Figure 4‑4 a) and b) show the results of InceptionV3(224\*224) model 25](#_Toc133853551)
  + [Figure 4‑5 a) and b) show the results of InceptionV3 (299\*299) model(Epoch = 20) 26](#_Toc133853552)
  + [Figure 4‑6 a) and b) show the results of Ensemble Stacking(Version 1) model 27](#_Toc133853553)
  + [Figure 4‑7 a) and b) show the results of Ensemble Stacking(Version 2) model 28](#_Toc133853554)
  + [Figure 4‑8 a) and b) show the comparative results of the two models’ fine-tuning 29](#_Toc133853555)
  + [Figure 4‑9 a) and b) show the results of VGG16 model(Epoch = 20) 29](#_Toc133853556)
  + [Figure 4‑10 a) and b) show the results of VGG16 model(Epoch = 50) 29](#_Toc133853557)
  + [Figure 4‑11 a) and b) show the results of ResNet50 model(Epoch = 20) 30](#_Toc133853558)
  + [Figure 4‑12 a) and b) show the results of ResNet50 model(Epoch = 50) 30](#_Toc133853559)
  + [Figure 4‑13 a) and b) show the results of InceptionV3(299\*299) model(Epoch = 50) 31](#_Toc133853560)
  + [Figure 4‑14: VGG16 comparative results: (a) and (b) shows train loss and validation loss, (c) and (d) shows train accuracy and validation accuracy 33](#_Toc133853561)
  + [Figure 4‑15: ResNet50 comparative results: (a) and (b) shows train loss and validation loss, (c) and (d) shows train accuracy and validation accuracy 34](#_Toc133853562)
  + [Figure 4‑16: InceptionV3 comparative results: (a) and (b) shows train loss and validation loss, (c) and (d) shows train accuracy and validation accuracy 35](#_Toc133853563)
  + [Figure 4‑17: InceptionV3(DA version 2) results: (a) shows loss results, (b) shows accuracy results 36](#_Toc133853564)
  + [Figure 4‑18: InceptionV3(DA version 2 + Early stopping) results: (a) shows loss results, (b) shows accuracy results 36](#_Toc133853565)
  + [Figure 4‑19: Performance comparison of the four models on two different datasets 39](#_Toc133853566)
  + [Figure 4‑20: The ensemble stacking’s training information on “Dog breed identification” dataset: (a)-(b)shows loss and accuracy, (c)-(d) shows AUC, TP and FP, (e)-(f) shows precision and recall 40](#_Toc133853567)
  + [Figure 4‑21: The best model’s testing accuracy, f1-score, recall, precision value 41](#_Toc133853568)
  + [Figure 4‑22: The best model’s confusion matrix 41](#_Toc133853569)
  + [Figure 4‑23: The best model’s ROC graph 41](#_Toc133853570)
  + [Figure 4‑24: The website index page 42](#_Toc133853571)
  + [Figure 4‑25: The six most popular dogs’ introduction page 43](#_Toc133853572)
  + [Figure 4‑26: Dog breed identification page 43](#_Toc133853573)
  + [Figure 5‑1: The project schedule 45](#_Toc133853574)
  + [Figure 5‑2: The local folder structure 46](#_Toc133853575)

# **List of Tables**

* [Table 2‑1: Comparison of different models' performance 9](#_Toc133868038)
* [Table 3‑1: Two different data augmentation versions 13](#_Toc133868039)
* [Table 3‑2: A summary of project datasets 14](#_Toc133868040)
* [Table 3‑3: The technologies of the project 19](#_Toc133868041)
* [Table 3‑4: The versions’ introduction 19](#_Toc133868042)
* [Table 4‑1: The same parameters setting of the four project models 21](#_Toc133868043)
* [Table 4‑2: The different parameters setting of the four project models 22](#_Toc133868044)
* [Table 4‑3: The project four models’ FC layer’s information 23](#_Toc133868045)
* [Table 4‑4: VGG16 model summary 23](#_Toc133868046)
* [Table 4‑5: ResNet50 model summary 24](#_Toc133868047)
* [Table 4‑6: InceptionV3 model summary 24](#_Toc133868048)
* [Table 4‑7: The results of InceptionV3 model’s two different input size 26](#_Toc133868049)
* [Table 4‑8: The two versions of the ensemble model 27](#_Toc133868050)
* [Table 4‑9: The two versions’ results of the ensemble model 28](#_Toc133868051)
* [Table 4‑10: Epoch = 20 VS Epoch = 50 31](#_Toc133868052)
* [Table 4‑11: The final epoch setting of each model 32](#_Toc133868053)
* [Table 4‑12: A summary of data augmentation experiments’ results 38](#_Toc133868054)
* [Table 4‑13: Comparison Analysis with Other State-of-the-art models 39](#_Toc133868055)
* [Table 5‑1: The complete/uncompleted tasks for each objective 47](#_Toc133868056)
* [Table 5‑2: Risk Analysis 49](#_Toc133868057)

# **Abstract**

The Knowledge of recognizing the breed of dogs, their conditions, behaviors, and natural instincts become a difficult challenge in our society today. Even, the traditional methods for identifying dog breeds, such as expert assessment or genetic testing, are often inefficient and costly. Therefore, there is a need to use an automated system for the recognition of dog breeds. The application of deep learning techniques has been inculcated in handling the aforementioned challenge. However, this project found that the deep learning techniques used in this field mainly rely on a single CNN model. If the classification problem becomes more complex, the performance of the single CNN model will be less.

To address and validate this problem, this project will use an ensemble model to provide a more accurate and robust network for dog breed recognition. The model will utilize the stacking ensemble algorithm to merge three pre-trained CNN models obtained through transfer learning: VGG16, ResNet-50, and InceptionV3, into a stacked ensemble model. Then evaluate the performance of these four models on two dog breed classification datasets, which are at different levels of complexity. During this process, fine-tuning, data augmentation, and early stopping techniques will be utilized to improve the models' performance. Finally, the experimental results illustrate that the proposed ensemble stacking model performs the best on the more complex dataset and has the least decline in model performance as the classification problem becomes more complicated among the four models. This outstanding performance proves the problem discovered in this project and provides a better solution for the increasingly complex dog breed classification area.

***Keywords:*** *Dog Breed Classification, Ensemble Model, Convolutional Neural Networks, Data Augmentation, Transfer Learning*

# **Abbreviations**

CNN Convolutional Neural Networks

VGG Visual Geometry Group

ResNet Residual Network

FC Fully Connected

GAP Global Average Pooling

BN Batch Normalization

TN True Negative

TP True Positive

FP False Positive

FN False Negative

M-PRC Macro-precision

M-REC Macro-recall

M-F1 Macro-F1 Score

ReLU Rectified Linear Unit

PCA Principal component analysis

LBP Local Binary Pattern

HOG Histogram of Oriented Gradients

# **Glossary**

**Dog Breed Recognition** is getting dog breeds by their faces' distinct features

**Ensemble Model is the process of synthesizing a final forecast by aggregating the forecasts of at least two different basic models**

**Stacking**is an ensemble learning algorithm to study how to aggregate the base models in an ensemble model

**Averaging is an ensemble learning technique in which each base model prediction will be used equally in the final predication of an ensemble model**

**Convolutional Neural Networks** are a kind of network architecture for deep learning algorithms and are used primarily for computer vision and processing

**Visual Geometry Group-16 is a standard deep convolutional neural network (CNN) architecture with 16 layers for image recognition**

**Residual Network-50 is a kind of convolutional neural network which is built by 48 convolutional layers, one MaxPool layer, and one average pool layer**

**InceptionV3 is a popular image recognition model which used a kind of deep learning convolutional neural networks architectures**

**Transfer Learning is in new tasks that apply acquired knowledge and skills which came from previously related tasks to increase the efficiency of task solving or improve the quality of task completion**

**Pre-trained Model is a model that has been previously trained on a large dataset, and the model's solution domain covers the task to be solved**

**Fine-tuned is a process of applying transfer learning, where the weights of an already trained network are used as starting values for training a new network**

**Data Augmentation** is a technique for increasing the amount of data in an existing dataset by making artificial changes to it

# **Introduction**

## Background

The canine is the animal most closely associated with humans. It brings many benefits to humans, such as emotional companionship, health benefits[1], and the fulfillment of specific tasks such as drug detection dogs, guide dogs for the blind, etc. Some dogs have different breeds with different habits and genetic disorders, but their appearance is too similar to be distinguished which has always been a challenge. Unfortunately, some issues are specific to particular breeds of dogs, such as the prevention of disease outbreaks such as rabies, regulation of vaccinations, and legal ownership. Workers face great difficulties in solving the above problems without recognizing breed details. With the number of dog breeds reaching 20,580 in 2021[2], it is challenging and time-consuming to identify dog breeds through expert identification or genetic testing, which are traditional methods of canine differentiation. However, applying image recognition techniques related to deep learning to dog breed identification and classification in obtaining accurate results can improve the understanding of dogs, promote animal science development, and avoid wasting human resources and some errors caused by humans has become this project goal.

## Aim

The ensemble model is a learning technology that combines many network models to solve a problem[3]. Compared to any separate image classification model, the benefits of the ensemble model, which obviously improves the classification performance are massive[4]. This project aims to utilize an ensemble model for the efficient identification of dog breeds[5].



Figure 1‑1: An example of dog breed identification

## Convolutional neural network

Convolutional neural network (CNN) is a kind of deep network architecture for deep learning algorithms and is used primarily for computer vision and processing. Because CNN not only can simplify complex problems by reducing a large number of parameters into a small number of parameters before processing, but also it is able to effectively retain image features, which is crucial for processing images. Furthermore, in this network, the output of the former layer is the input of the latter layer, whose mathematical expression is:

(1-1)

represents the output feature’s number, represents the input feature number, k means the kernel size, p is the padding size and s is the stride.

A typical CNN consists of 3 parts, convolutional layers, pooling layers, and fully connected layers.

### Input layer

In CNNs, the input layer is the first and essential layer because it processes the raw input image and forms the foundation for the rest of the network's feature extraction and classification capabilities. In image recognition, the raw Red-Green-Blue (RGB) pixel values, possibly zero-centered and/or normalized, are usually used as input[6] In other words, the input layer requires a three-dimensional tensor (height, width, depth), where height and width represent the spatial dimensions of the image and depth represents the number of color channels.

### Convolutional layer (Conv layer)

The convolutional layer consists of a set of learnable filters, also known as kernels, that slide or convolve over the input data to extract local features. As shown in [Figure 1-2](#figure_1_2), each filter is a small matrix of learnable weights that are dotted between its weights and a small region of the input. And the process of filter convolution on the input data creates a set of feature maps, each of which corresponds to the output of a particular filter[7]. This can be understood by using a filter (convolution kernel) to filter individual small regions of the image to obtain the feature values of these small regions. The output of this operation is a single value corresponding to the presence or absence of a particular feature in that region of the input.

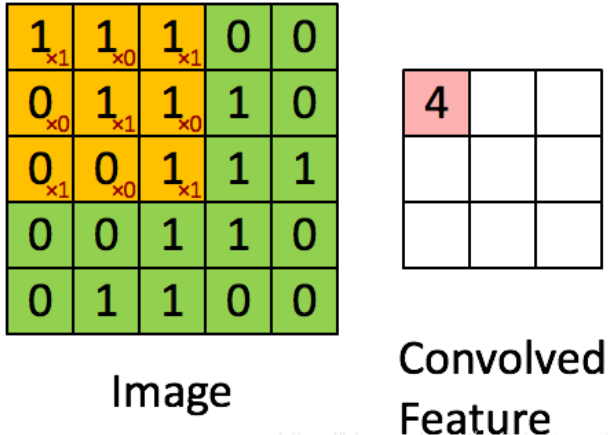


Figure 1‑2: An example of operations in the Conv layer

The size of the feature map can be controlled by adjusting the span of the convolution layer, the padding, and the filter size. The padding value is usually set to 0[8]. And the value is calculated as follows:

Padding = (1-2)

Additionally, the expression for the computation of the convolution layer can be:

(1-3)

P means padding and the k expresses the kernel size value.

### Pooling layer

The pooling layer will filter feature values. It means significantly reducing the parameter magnitude (dimensionality reduction) to greatly reduce computation and effectively avoid overfitting. There are two common ways of pooling, one is max pooling, and the other is mean pooling. [Figure 1-3](#figure_1_3) illustrates the operation of max-pooling in detail. By doing so, the height and width of the output image can be halved, and the number of channels remains the same.

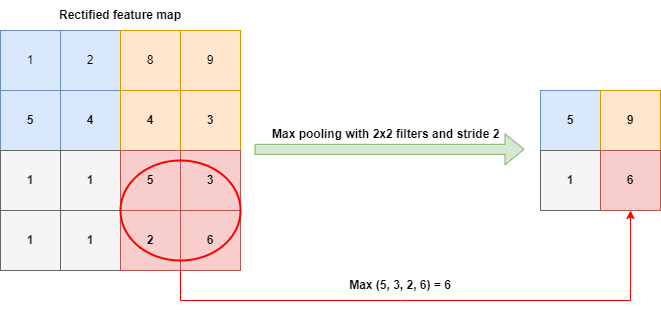


Figure 1‑3: The operation example of Max-pooling

Moreover, the mean-pooling, slightly different from max-pooling, is to the average value of a region instead of taking the maximum value, as shown in [Figure 1-4](#figure_1_4).

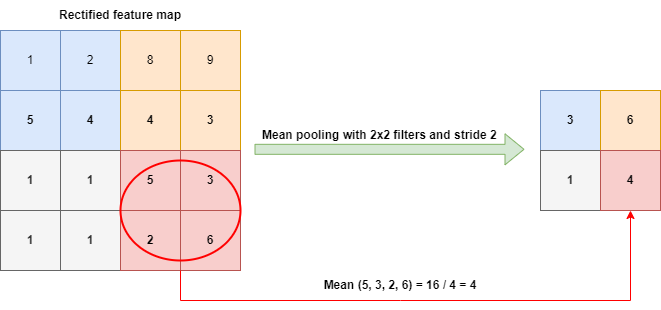


Figure 1‑4: The operation example of Mean-pooling

### Fully-Connected layers (FC layers)

After several convolutions and pooling, the feature map is sent to the fully-connected layer for classification. The fully connected layer can integrate the class-discriminative local features extracted from the convolutional and pooling layers to obtain a global representation of the input image and make the final classification prediction. In the overall architecture of CNN, the convolution and pooling steps can be considered the feature extraction process and the final fully connected layer as a classifier. The last fully connected layer is the predicted classes, i.e., the output layer[9], which can be classified using softmax as the activation function, and this layer can also be called the softmax layer[10]. And to improve the performance of CNN networks, the activation function of each neuron in the fully connected layer generally adopts the ReLU function.

### Activation functions

Without using the activation function, each layer of the CNN just does a linear transformation, and the multi-layer inputs are still linearly transformed when superimposed. Unfortunately, the expressiveness of linear models is usually insufficient. However, the activation function can introduce nonlinear factors that improve the expressiveness of the model. [Figure 1-5](#figure_1_5) shows a common activation function used for the hidden layer of the FC layer in CNN is ReLU[11]. And the softmax activation function is generally used in the classification layer.

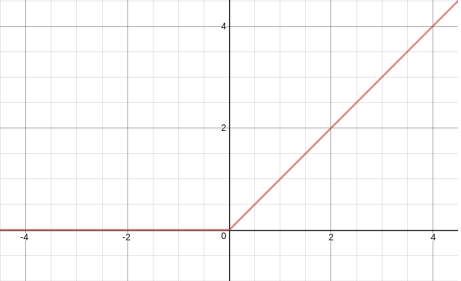


Figure 1‑5: The Rectified Linear Unit (ReLU) activation function produces 0 as an output when x < 0, and then produces a linear with slope of 1 when x > 0[11]

### Loss functions

The loss function is essentially a function that calculates the difference between a model's predicted value for a sample and the true value and is used to evaluate the effectiveness of the model[12]. To be detailed, the smaller the loss value, the closer the true value is to the predicted value, and the better the model performance is. Additionally, the loss function can be broadly classified into two categories: classification loss and regression loss, where the classification loss can be subdivided into binary cross-entropy loss and multi-class.

## Objectives

The project will be divided into six phases:

* Training three CNN models using transferring learning on a dataset with 120 dog breeds.
* Use the stacking ensemble algorithm to combine the three CNN models implemented above to obtain an ensemble model and train it on the same dataset.
* Improve the performance of the four models as much as possible by tuning the hyperparameters reasonably and doing the data augmentation.
* Doing a comparative analysis between the three CNN models and the project-proposed ensemble model on the same dataset.
* Doing a comparative analysis of the four models between two different breeds datasets to get the best model.
* Deploy the best model to a web application.

## Project Overview

### Scope

CNN is popular in the field of image classification due to its excellent feature extraction ability[13]–[15]. But a CNN whose output is ideal possibly will have an overfitting situation because of high complexity[3] Individual CNN models are prone to the drawback of overfitting. In contrast, using the ensemble network can combine the advantages of different CNN models while compensating for each other's drawbacks to achieve a relatively desirable result for species classification and recognition. The excellent performance of ensemble learning has led to an increasing tendency to combine various CNN models[16], [17]. Therefore, the purpose of the study is to put together the advantages of individual networks for robust performance for the classification of dog breeds.

The significances of this study are as follows:

* To help promote animal science development.
* Reduces time and human efforts.
* For easy identification of dog breeds and genetic testing.
* For educational purpose
* For legal ownership
* To promote responsible pet ownership and greatly benefit animal welfare
* Improve public safety

### Audience

This project has several potential benefits for different stakeholders.

For the government, it can aid in studying the common features of strongly aggressive dogs that should not be kept in the city, which can inform policy decision-making related to dog ownership and access to public spaces. Access control systems can also use this technology to monitor whether people own legal dogs and are walking them in permitted areas, reducing the need for manual monitoring and associated costs.

For residents, this project can improve the safety of the public environment to some extent by reducing the likelihood and severity of dog attacks. This is because accurate identification of dogs that may pose a threat, such as large and aggressive breeds, can help to prevent incidents before they occur. Additionally, in the event of an attack, knowing the attacking dog's breed can help medical professionals to provide timely and appropriate treatment, which may reduce the severity of the injuries sustained.

Moreover, as mentioned at the beginning, it also greatly helps veterinarians in their work, such as accurate breed identification can help reduce the risk of misdiagnosis and ensure appropriate treatment plans are put in place, improving consultation efficiency to allow veterinarians to spend more time with their patients instead of identifying a dog's breed, and assist disease prevention and curing by provide the genetic predispositions and health risks associated with specific dog breeds.

Prospective or current dog owners: This project can be used to accurately identify the breed of a dog they are considering purchasing or already own. By doing so, they can learn about the specific health concerns and dietary needs associated with that breed. This information can help them make more informed purchasing decisions and provide better care for their pets, potentially improving their overall health and well-being.

Animal shelters and rescue organizations can benefit greatly from this project as it aids in identifying dogs and providing accurate information on breed traits and health risks to potential adopters. This can lead to increased adoption rates and better outcomes for dogs in their care, ultimately reducing the likelihood of dogs being returned to the shelter or rescue organization.

# **Background Review**

This chapter shows some related research about CNN models, transferring learning, and ensemble models in the dog breed identification area.

### CNN & Transfer learning

There are numerous pieces of research in the dog breed identification domain using deep learning techniques. Traditional methods like Chanvichitkul M et al. [18]employed the template matching technique and Principle Component Analysis (PCA) to classify 35 dog breeds in 700 images and get approximately 93% accuracy. However, for a bigger dataset, using CNN is a wise way. Varshney A compared two CNN models' classification results, InceptionV3 and VGG16[2]. And better accuracy (InceptionV3) is 85% in 2050 images. Furthermore, Rishita M et al.[19] and Shah B et al.[20], respectively, used the ResNet-50 model and CNN to achieve 87.42% and 95.5% accuracy in 7515 and 8351 image dog classification. Then, Borwarginn P et al.[21] compare a conventional method that uses Local Binary Pattern and Histogram of Oriented Gradient and a deep learning way that uses CNN and transfer learning. The better one is the deep learning method, with an accuracy is 96.75% in 8351 images. Moreover, the research analyzes VGG, ResNetV2, DenseNet, and InceptionResNetV2 models’ effects and different transfer learning algorithms in 10000 images[22]. The best one is InceptionResnetV2 plus AdaDelta to achieve around 91% accuracy.

### Ensemble models

Finally, the more breeds there are, the larger the number of classifications required, and the less well a single CNN model performs. The ensemble model, by combining multiple CNN models, can remove its weaknesses and take its strengths, making it more suitable for increasingly complex identification purposes. Ensemble models have achieved outstanding performance in other fields. For example, research using VGG16 and ResNet50 to get an ensemble model for four sheep breed classifications and got an excellent accuracy, of 97.32%[23]. But for the dog breed classification area, it is hard to find research about using ensemble models. This project only found research that used VGG16, ResNet50, and InceptionV3 models with ensemble learning in 10222 images and 120 breeds[24]. It used averaging to merge the three models, and the accuracy was 67.17%, which is not an ideal performance. Therefore, for the dog breed identification area, the potential of ensemble models is not fully explored. It is certainly one of the research objectives of this project, to obtain a better performance of dog breed recognition using a proposed ensemble model to make up for the deficiency that the popular single CNN model is used for dog breed recognition and give a powerful solution.

[Table 2-1](#Table_2_1) displays a summary of the related research above.

|  |  |  |
| --- | --- | --- |
| **Researchers** | **Techniques** | **Performance** |
| Chanvichitkul M et al. [18] | Coarse Classification and PCA | Accuracy≈93%  (700 images, 35 breeds) |
| Varshney A et al. [2] | InceptionV3, VGG16 | Accuracy=85%  (2050 images, 120 breeds) |
| Rishita M et al. [19] | ResNet-50 | Accuracy=87.42%  (7515 images, 133 breeds) |
| Shah B et al. [20] | CNN | Accuracy=95.5%  (8351 images,103 breeds) |
| Borwarnginn P et al. [21] | LBP and HOG  CNN + Transfer learning | Accuracy=96.75%  (8351 images,133 breeds) |
| Agarwal A et al. [22] | VGG, ResNetV2, DenseNet, and InceptionResNetV2, AdaDelta | Accuracy≈91%  (10000 images,70 breeds) |
| Agrawal D et al. [23] | Ensemble  (VGG16 + ResNet50) | Accuracy=97.32%  (4 sheep breeds) |
| Liang B et al. [24] | Ensemble Averaging  (VGG16, ResNet50 and InceptionV3) | Accuracy=67.17%,  (10222 images,120 breeds) |

Table 2‑1: Comparison of different models' performance

# **Methodology**

This section detailly describes the project datasets, what operations were performed on the datasets, the CNN model used in the project, the ensemble algorithm, the ensemble model architecture proposed in the project, and the model evaluation matrix.

## Dataset

This subsection will introduce the phases carried out in the preparation of the dataset which are data resizing, data labeling, and data augmentation.

### About Dataset

The project uses two dog breed classification datasets which are available on the Kaggle public platform.

The first dataset is the "Dog Breed Identification" dataset which is an imbalanced dataset [(Figure 3-1)](#figure_3_1) and has 120 dog breeds. The dataset only provides the training dataset which has 10222 total images. During the experiment stage 1-2, the project uses the split ratio of 20:80 to divide them into 8,178 training dataset images and 2,044 testing dataset images. However, to choose the best model, the project moves each breed’s ten images from the training dataset to consist of the testing dataset. And for this new version of the "Dog Breed Identification" dataset, has 7218 training images, 1804 validation images, and 1200 testing images [(Figure 3-2)](#figure_3_2).

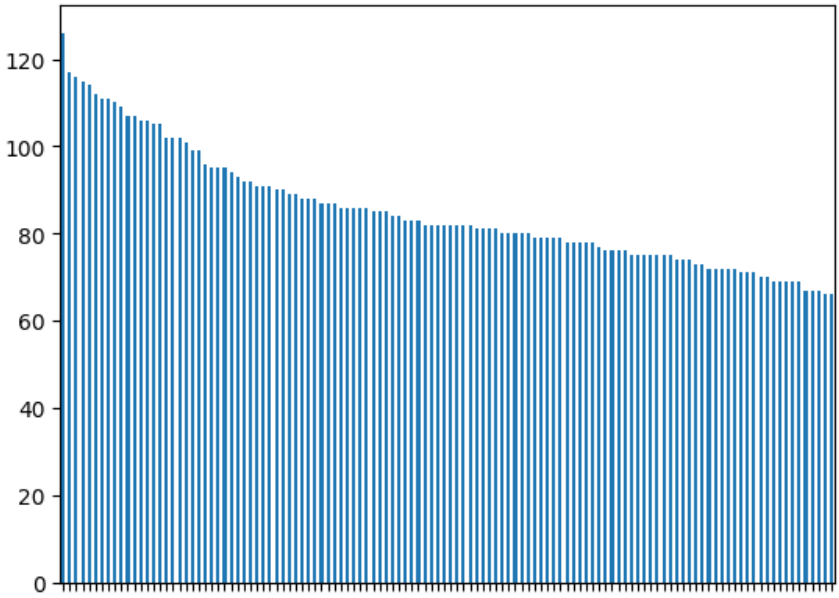


Figure 3‑1: Distribution of the images stored in each class of the dataset

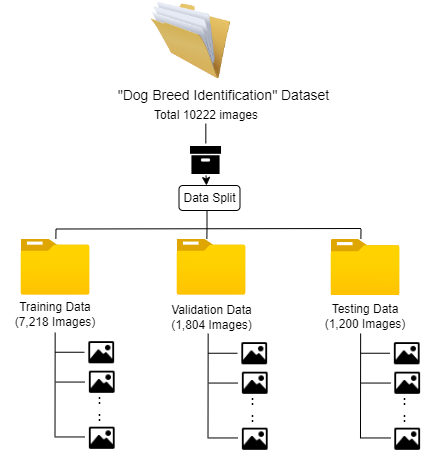
******

Figure 3‑2: Information on the new version "Dog Breed Identification" dataset

And the second dataset is the "70 Dog Breeds-Image Data Set". It has a total of 9346 images with 70 dog breeds, of which the training dataset is 7946, the validation dataset is 700 and the testing dataset is 700.

### Data Resizing

To comply with the input image size requirements of the CNN model utilized in the project, the "Dog Breed Identification" dataset images are resized to versions 224\*224 and 299\*299. Conversely, since the "70 Dog Breeds-Image" dataset offers default images of size 224\*224, this project solely resizes the images of this dataset once to obtain the 299\*299 version.

### Data labeling

This section presents the structure in which the images of the two datasets are stored, as illustrated in [Figures 3-3](#Figure_3_3) and [3-4](#Figure_3_4).

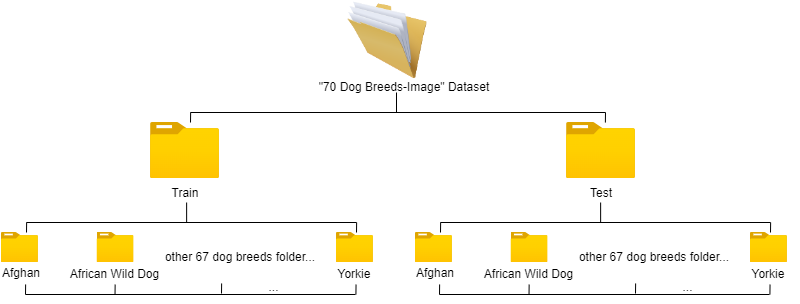
******

Figure 3‑3: The labeling of the "70 Dog Breeds-Image Data Set"

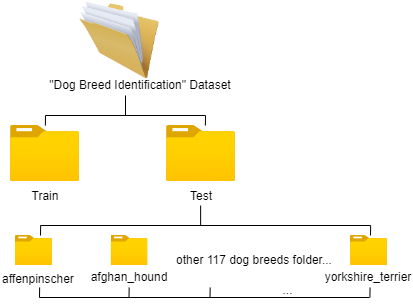
******

Figure 3‑4: The labeling of the new version "Dog Breed Identification" dataset

### Data Augmentation

Data augmentation refers to the process of artificially increasing the size of a dataset by generating new training examples through transformations of the existing ones while preserving their original labels. Data augmentation is widely used in deep learning to improve the generalization ability and robustness of models, as it enables models to learn from a diverse set of training examples and better handle variations and perturbations in the input data. Data augmentation means it can create a more diverse and balanced training dataset, which can lead to improved model performance and better generalization to new and unseen data.

And as shown in [Figure 3-1](#figure_3_1), in the case of the "Dog Breed Identification" dataset, some breed categories may have significantly fewer images than others, leading to class imbalance. Class imbalance refers to the situation where the number of samples in one or more classes is significantly smaller than the number of samples in the other classes. This can negatively impact the network's training process because the model will be biased towards the majority class and may have difficulty learning the features of the minority class. As a result, the model may have poor performance when it comes to predicting the underrepresented class, which can be problematic in real-world applications. To address this issue, and enhance the generalization and robustness of the deep learning models, employed in this project, various data augmentation techniques will be utilized, particularly considering they are for the dog breeds multiclass identification task: Random rotation will be applied to increase the variation and diversity of the dataset, as different dog breeds may have significant differences in shape and posture. Additionally, random translation will be utilized to introduce diversity in the dataset, as different dog breeds may appear at different locations in the images. Random zoom will also be applied to increase the variation in size among different dog breeds. Furthermore, random shear will add diversity to the dataset, as different dog breeds may have significant differences in shape. Random brightness adjustment will address the variation in image brightness due to different shooting environments. Finally, horizontal flipping will be employed to add diversity to the dataset, as most dogs exhibit left-right symmetry.

Therefore, a reasonable data augmentation setting for this project is version 2 in [Table 3-1](#Table_3_1). However, to prove it, the project designed version 1 to do the comparative analysis.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **DA Version** | **Rotation** | **Translation** | **Zoom** | **Shear** | **Brightness** | **Horizontal**  **Flipping** | **Vertical**  **Flipping** |
| 1 | 20 degrees | 20% | 20% | 0% | 1.0 | True | True |
| 2 | 5 degrees | 10% | 10% | 5% | 0.8-1.2 | True | False |

Table 3‑1: Two different data augmentation versions

**Finally,** [Table 3-2](#Table_3_2) **is a summary of project datasets.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Training images** | **Validation images** | **Testing images** | **Total** |
| "Dog Breed Identification" | 8178 | 2044 | 0 | 10222 |
| New "Dog Breed Identification" | 7218 | 1804 | 1200 | 10222 |
| "70 Dog Breeds-Image" | 7946 | 700 | 700 | 9346 |

Table 3‑2: A summary of project datasets

## Model Architecture

***ResNet(Residual Network):***Degradation problem: using shallow layers stacked directly into deep networks not only makes it difficult to exploit the powerful feature extraction capabilities of deeper networks but also decreases in accuracy, and not due to overfitting[25]. ResNet uses identity mapping to pass the previous layer's output directly to the later layer, making the network layers very deep, and the final classification results are excellent[26]. The basic structure of the residual network is shown in [Figure 3-5](#Figure_3_5).

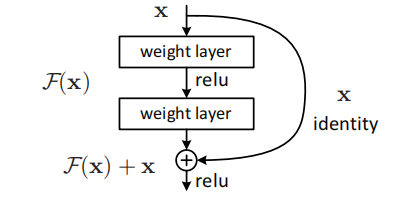


Figure 3‑5[26]: Residual learning: a building block

***VGG16(Visual Geometry Group)*:** VGG-16 is a convolutional neural network that is 16 layers which consist of 13 convolutional layers and 3 fully-connected layers. VGG is a famous and popular CNN architecture that is first represented using a deep network and small convolutional filters(3 x 3) to recognize images with high accuracy likelihood[14].

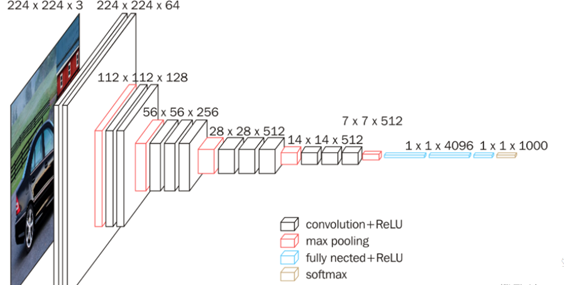


Figure 3‑6[14]: A standard VGG16 model’s architecture

***Inceptionv3*:** To improve neural networks' performance, InceptionNet has been created to increase the depth and width of the neural network. The network uses convolutional kernels of different sizes, allowing the existence of perceptual fields of different sizes, and finally achieving splicing to achieve the fusion of features at different scales[27]. And one of the most popular ones in InceptionNet is InceptionV3[28].

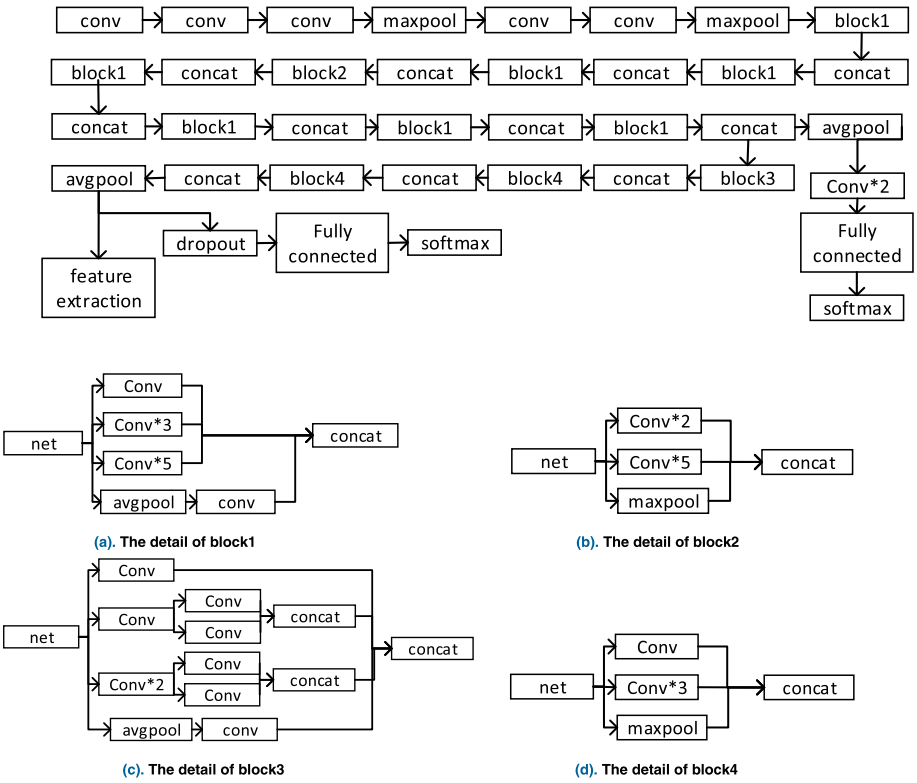


Figure 3‑7[29]: The InceptionV3 model's architecture. The detail of blocks 1-4 are shown in Figure 3-7(a), Figure 3-7(b), Figure 3-7(c), and Figure 3-7(d).

***Ensemble models*:** Ensemble models combine multiple weakly supervised models to obtain a better, more comprehensive, strongly supervised model[4]. In other words, if a weak classifier gets an incorrect prediction, the other weak classifiers can correct the error.

***Stacking ensemble algorithm:*** It takes the outputs of base models as input and attempts to learn how to best combine the input predictions to make a better output prediction.

Detailed information on the project’s architecture is shown in [Figure 3-8](#Figure_3_8).

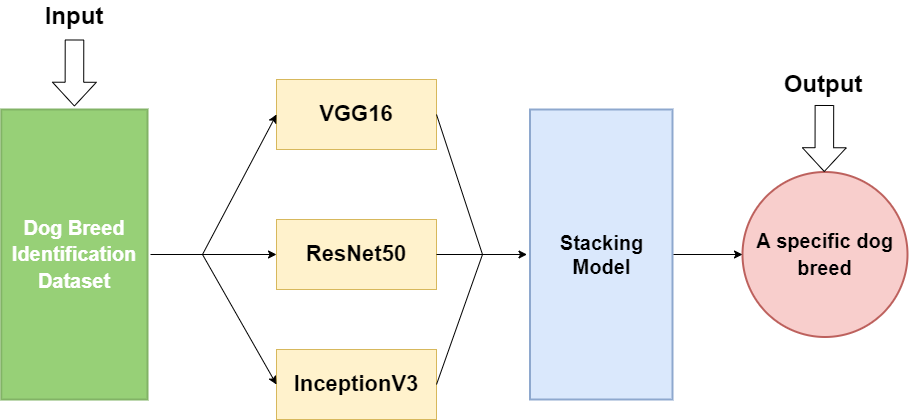


Figure 3‑8: The proposed architecture of this project

This project will use Batch Normalization(BN) and GlobalAveragePooling2d methods to implement the project CNN models.

***Batch Normalization:*** Using BN could make the distribution of eigenvalues in this layer back to a standard normal distribution with a mean of 0 and a variance of 1[30]. The features' values will fall in the interval where the activation function is more sensitive to the input. Small changes in the input can lead to larger changes in the loss function, making the gradient larger, avoiding the vanishing gradient, and speeding up convergence[31]. Meanwhile, it could improve the network generalization ability.

**GlobalAveragePooling2d** does the global average pooling(GAP) to simplify enormous parameter operations and prevent overfitting at this layer. And because it integrates global spatial information, the spatial translation is more robust.

## Testing and Evaluation

This section discusses how the project is to do tests and evaluations.

### Data testing:

* This project will check the dataset’s images’ state to ensure that the corresponding data augmentation operation is working.
* This project will check the images are resized to the desired size and that the corresponding total number of resizing images are same as the original dataset to ensure the resizing images' size operation is correct.
* This project will check the total images' numbers between the validation and training datasets meet the designed division ratio and that the sum of them is equal to the original training dataset images' total number to ensure the division of the validation dataset from the training dataset process is correct.

### Performance metrics

**Accuracy** (ACC) calculates the ratio of correctly classified predictions to the total number of predictions. However, accuracy is not a good metric for unbalanced datasets. That is, when a class x in the dataset is the majority class, even if the model's predicted results are only "class x", a high accuracy score can be obtained.

**Recall:** the higher the recall, the more positive samples are correctly predicted by the model, and the better the model is.

**Precision:** the proportion of samples identified as positive by the model that is actually positive.

The **F1-score** is based on a harmonic mean of the Recall and Precision.

For multi-classification, precision, recall, and F1-score represent the model's performance in one class, while M-PRC, M-REC, and M-F1 are the criteria to judge the model's overall classification performance.

And Equations (3-1)-(3-6) show how to calculate the metrics.

Accuracy = (3-1)

Precisioni = (3-2)

Precisionmacro = (3-3)

Recalli = (3-4)

Recallmacro = (3-5)

F1-scoremacro = (3-6)

True and False indicate correct and incorrect prediction results, and Positive and Negative indicate positive and negative examples of the actual sample.

TN, TP, FP, and FN represent true negative, true positive, false positive, and false negative, respectively.

In addition, the **ROC (Receiver Operating Characteristic) curve** is a graphical representation of the relationship between the true positive rate (TPR) and false positive rate (FPR) of a binary classifier at different classification thresholds. TPR represents the proportion of positive samples that are correctly classified, while FPR represents the proportion of negative samples that are incorrectly classified as positive. The ROC curve is advantageous because it considers the classifier's performance at different classification thresholds, rather than just one fixed threshold. Plotting the ROC curves to visualize the model's trade-off between the true positive rate and the false positive rate, provides an intuitive understanding of its performance.

**AUC (Area Under the ROC Curve)** is the area under the ROC curve and its value ranges from 0 to 1. In a binary classification problem, AUC measures the model's ability to distinguish between positive and negative samples. In a multi-class classification problem, AUC can be used as a metric to evaluate the overall performance of the model for all classes. ROC curve and AUC can be used to evaluate the performance of a classification model. Generally, a good classifier's ROC curve should be as close to the upper left corner as possible, and AUC should be closer to 1, indicating better classification performance.

To evaluate the performance of various models in the challenging 120 dog breeds multi-class classification problem, a comprehensive set of evaluation metrics, including accuracy, macro-precision(M-PRC), macro-recall(M-REC), macro-F1 score(M-F1), ROC graph, and AUC, were employed. By leveraging these metrics and visualization techniques, the project was able to gain a deep understanding of the strengths and weaknesses of each model in accurately classifying the data for a comprehensive evaluation of the model's performance.

## Technology

The part information is shown in [Table 3-3](#Table_3_3).

|  |  |  |
| --- | --- | --- |
| Software | Framework | TensorFlow, Flask, Bootstrap 4 |
| Language | Python, HTML5, CSS3, JavaScript, Ajax |
| Libraries | Numpy, Scikit learn, Pandas, OpenCV, Scipy, Keras, Pathlib, Matplotlib |
| Version management plan | Git repository |
| Hardware | Central processing unit(CPU) | Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz |
| Graphic Processing Unit(GPU) | NVIDIA GeForce GTX 1660Ti |
| Cloud-based Jupyter notebook environment | Google Colaboratory |

Table 3‑3: The technologies of the project

## Project Version Management

This part will use Github to manage four versions. The versions’ details are in [Table 3-4](#Table_3_4).

And the Github repository link is: <https://github.com/JosephineQiao/Project-Ensemble-Model-for-the-Classification-of-Dog-Breed>

|  |  |
| --- | --- |
| Version 1 | Implement three different CNN models on the “Dog breed identification” dataset. |
| Version 2 | Implement combining all single CNN models into a model to fit the “Dog breed identification” dataset by using the stacking combination strategy. |
| Version 3 | Training all models after doing data augmentation. |
| Version 4 | Implement the four models on the “70 Dog Breeds-Image Data Set”. |
| Version 5 | Deploy the best model by Flask to a mini web application. |

Table 3‑4: The versions’ introduction

# **Experiments and results**

The experiments are divided into three phases:

* Implements VGG16, ResNet50, InceptionV3, and ensemble stacking models. Some hyperparameters will be tuned for InceptionV3 and ensemble stacking models so that their performance can be improved as much as possible.
* Implements data augmentation and early stopping techniques to maximize the performance of each model.
* Obtains additional evaluation metrics for the four models on the "Dog breed identification" dataset. Then, training the four models on another dataset to verify whether a single CNN model performs less well as the classification problem becomes more complex, while the ensemble model is conversely.

## The first stage of the experiment

During the experimental phase, the three CNN models were trained on the ImageNet dataset to obtain relevant pre-trained neural networks. Furthermore, these pre-trained models were further utilized in a stacking algorithm to obtain an ensemble model. The performance of each individual model on the "Dog breed identification" dataset was evaluated, and certain parameter values of the InceptionV3 model and ensemble stacking model were adjusted in an attempt to enhance their performance. Moreover, the effects of setting epoch values to 20 and 50 on the models were investigated. And give the final parameters selection was determined based on the results.

### Implementation design for each model

**4.1.1.1 Hyperparameters in each model**

This subsection shows the specifics of the hyperparameters for each model in the experiment. [Table 4-1](#Table_4_1) shows the common hyperparameter settings for the four models implemented in the project, while [Table 4-2](#Table_4_2) shows the different hyperparameter settings.

|  |  |
| --- | --- |
| Parameters | Values |
| Epochs | 20 |
| 50 |
| Batch size | 16 |
| Output | 120 |
| 70 |
| Optimizer | Adam |

Table 4‑1: The same parameters setting of the four project models

|  |  |  |
| --- | --- | --- |
| Model | Parameters | Values |
| VGG16 | Input size | 224 \* 224 \* 3 |
| Activation function | ReLU + Softmax |
| Learning rate | 0.001 |
| FC Layers number | 3 |
| ResNet50 | Input size | 224 \* 224 \* 3 |
| Activation function | ReLU + Softmax |
| Learning rate | 0.0001 |
| FC Layers number | 2 |
| InceptionV3 | Input size | 224 \* 224 \* 3 |
| 299 \* 299 \* 3 |
| Activation function | ReLU + Softmax |
| Learning rate | 0.0001 |
| FC Layers number | 2 |
| Ensemble stacking | Input size | 224 \* 224 \* 3, 224 \* 224 \* 3, 224 \* 224 \* 3 |
| 224 \* 224 \* 3, 224 \* 224 \* 3, 299 \* 299 \* 3 |
| Activation function | ReLU + Softmax |
| Leaky ReLU + Softmax |
| FC Layers number | 2 |

Table 4‑2: The different parameters setting of the four project models

[Table 4-2](#Table_4_2) illustrates how InceptionV3 and ensemble stacking models will change hyperparameters for fine-tuning experiments. Their experimental results and analysis can be seen in **Chapter 4.1.2.1-4.1.2.2.**

**4.1.1.2 FC situation**

[Table 4-3](#Table_4_3) shows the project's design for the fully connected(FC) layer section of the four project models.

|  |  |  |
| --- | --- | --- |
| Before FC | Model | FC layers information |
| Corresponding pre-training model,BN, GAP, 0.5 dropout. | VGG16 | FC layer 1: Relu activation function, neuron number: 1024;  Then, 0.5 dropout.  FC layer 2: Relu, neuron number: 256;  Then, 0.5 dropout.  FC layer 3: Softmax, 120.  0.5 dropout. |
| ResNet50 | FC layer 1: Relu, 2048;  Then, 0.5 dropout.  FC layer 2: Softmax, 120. |
| InceptionV3 | FC layer 1: Relu, 1024;  Then, 0.5 dropout.  FC layer 2: Softmax, 120. |
| Create a larger tensor that combines the features learned by the three project CNN models, BN | Ensemble stacking | FC layer 1: Relu, 1024;  Then, 0.5 dropout.  FC layer 2: Softmax, 120. |

Table 4‑3: The project four models’ FC layer’s information

[Table 4-4](#Table_4_4) to [4-6](#Table_4_6) provides detailed depiction information of the implementation of the three CNN models, while [Figure 4-1](#figure_4_1) to [4-3](#figure_4_3)illustrates the composition of these models visually.

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Parameters |
| VGG16 (Functional) | (None, 7, 7, 512) | 14714688 |
| Batch Normalization | (None, 7, 7, 512) | 2048 |
| GlobalAveragePooling2d | (None, 512) | 0 |
| Dropout | (None, 512) | 0 |
| Dense | (None, 1024) | 525312 |
| Dropout 1 | (None, 1024) | 0 |
| Dense 1 | (None, 256) | 262400 |
| Dropout 2 | (None, 256) | 0 |
| Dense 2 | (None, 120) | 30840 |
| Total parameters: 15,535,288  Trainable parameters: 819,576  Non-trainable parameters: 14,715,712 | | |

Table 4‑4: VGG16 model summary

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Parameters |
| ResNet50 (Functional) | (None, 7, 7, 2048) | 23587712 |
| Batch Normalization | (None, 7, 7, 2048) | 8192 |
| GlobalAveragePooling2d | (None, 2048) | 0 |
| Dropout | (None, 2048) | 0 |
| Dense | (None, 2048) | 4196352 |
| Dropout 1 | (None, 2048) | 0 |
| Dense 1 | (None, 120) | 245880 |
| Total parameters: 28,038,136  Trainable parameters: 4,446,328  Non-trainable parameters: 23,591,808 | | |

Table 4‑5: ResNet50 model summary

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Parameters |
| InceptionV3 (Functional) | (None, 5, 5, 2048) | 21802784 |
| Batch Normalization | (None, 5, 5, 2048) | 8192 |
| GlobalAveragePooling2d | (None, 2048) | 0 |
| Dropout | (None, 2048) | 0 |
| Dense | (None, 1024) | 2098176 |
| Dropout 1 | (None, 1024) | 0 |
| Dense 1 | (None, 120) | 123000 |
| Total parameters: 24,032,152  Trainable parameters: 2,225,272  Non-trainable parameters: 21,806,880 | | |

Table 4‑6: InceptionV3 model summary



Figure 4‑1: VGG16 model process



Figure 4‑2: ResNet50 model process



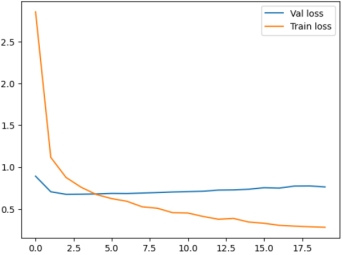
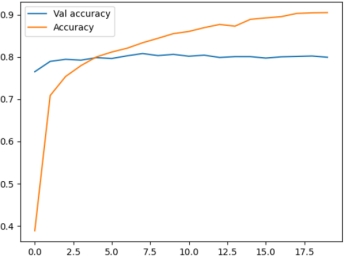
Figure 4‑3: InceptionV3 model process

### Results and analysis

This section will initially present the outcomes of the implemented InceptionV3 model and ensemble stacking model, along with the results and analysis of their corresponding fine-tuning experiments. Subsequently, the results of the three CNN models with epoch values set to 20 and 50 will be presented, and decide the ultimate choice of the epoch values.

**4.1.2.1** **InceptionV3 implementation and fine-tuning results and analysis.**

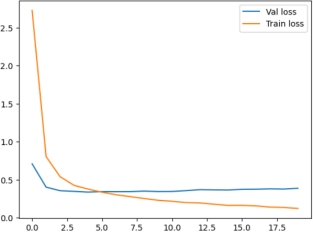
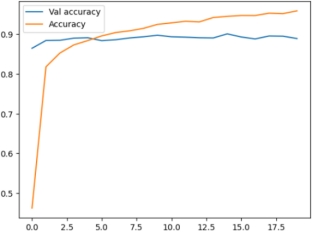
The present experiment examines the impact of two distinct input sizes on the performance of the InceptionV3 model. Specifically, [Figure 4-4](#figure_4_4) depicts the model's performance with an input size of 224\*224, while [Figure 4-5](#figure_4_5) demonstrates the model's performance with an input size of 299\*299.

a) The validation loss and training loss b) The validation and training accuracy

of the InceptionV3(224\*224) model for the InceptionV3(224\*224) model

Figure 4‑4 a) and b) show the results of the InceptionV3(224\*224) model

a) The validation loss and training loss b) The validation and training accuracy

of the InceptionV3(299\*299) model for the InceptionV3(299\*299) model

Figure 4‑5 a) and b) show the results of the InceptionV3 (299\*299) model(Epoch = 20)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Train\_Accuracy | Val\_Accuracy | Train\_Loss | Val\_Loss |
| InceptionV3(224\*224) | 90.47% | 79.89% | 0.2793 | 0.7622 |
| InceptionV3(299\*299) | 95.85% | 88.89% | 0.1234 | 0.3888 |

Table 4‑7: The results of InceptionV3 model’s two different input size

With changing the input size of the InceptionV3 model to 299\*299\*3, both the gap was lower between the loss and accuracy of its validation and training datasets, as visualized in [Figures 4-4](#figure_4_4) to [4-5](#figure_4_5), and the data in [Table 4-7](#Table_4_7), which accurately shows that the train accuracy and validation accuracy were improved. These prove that the best Input size of the InceptionV3 model for this project is 299\*299\*3.

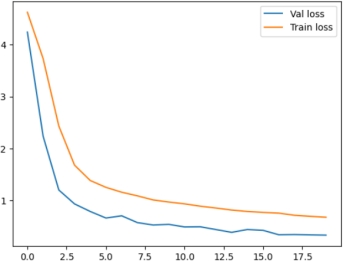
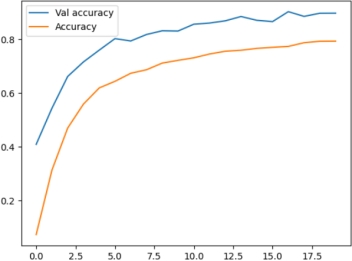
**4.1.2.2 The ensemble model’s implementation and fine-tuning results and analysis**

As the InceptionV3 model is an integral component of the ensemble stacking model, it is imperative to adjust the input image size of the InceptionV3 model for improved model performance. Furthermore, if using the ReLU, when x<0, the derivative of the ReLU function is 0, and the corresponding weights cannot be updated[32]. By replacing the activation function from ReLU to Leaky ReLU, the "dying ReLU" problem is solved[32]. It could improve the performance of the "Ensemble Averaging" model. In summary, the ensemble stacking model will conduct the experiments outlined in [Table 4-8](#Table_4_8) for both versions.

|  |  |  |
| --- | --- | --- |
|  | Input size | Activation function |
| Version 1 | 224 \* 224 \* 3, 224 \* 224 \* 3, 224 \* 224 \* 3 | ReLU + Softmax |
| Version 2 | 224 \* 224 \* 3, 224 \* 224 \* 3, 299 \* 299 \* 3 | Leaky ReLU + Softmax |

Table 4‑8: The two versions of the ensemble model

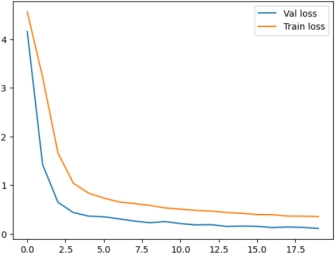
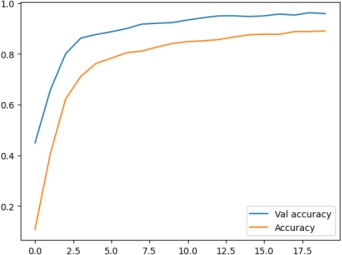
[Figure 4-6](#figure_4_6) shows the performance of the ensemble stacking model (version 1), while [Figure 4-7](#figure_4_7) shows the performance of the ensemble stacking model (version 2).

a) The validation and training loss of the b) The validation and training accuracy for

Ensemble Stacking(Version 1)model the Ensemble Stacking(Version 1) model

Figure 4‑6 a) and b) show the results of the Ensemble Stacking(Version 1) model

a) The validation and training loss of the b) The validation and training accuracy for

Ensemble Stacking(Version 2) model the Ensemble Stacking(Version 2) model

Figure 4‑7 a) and b) show the results of the Ensemble Stacking(Version 2) model

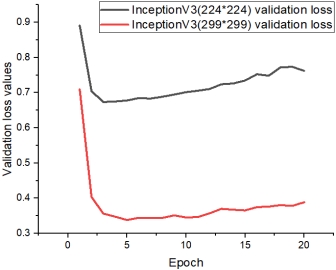
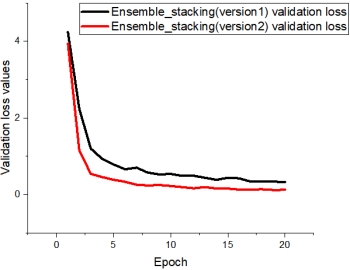
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Train\_Accuracy | Val\_Accuracy | Train\_Loss | Val\_Loss |
| Ensemble Stacking (Version 1) | 79.38% | 89.82% | 0.6777 | 0.3331 |
| Ensemble Stacking (Version 2) | 89.99% | 95.97% | 0.3215 | 0.1255 |

Table 4‑9: The two versions’ results of the ensemble model

As illustrated in [Figures 4-6](#figure_4_6) to [4-7,](#figure_4_7) and [Table 4-9](#Table_4_9), the ensemble stacking model displays a marked enhancement in Version 2 when compared to Version 1. Specifically, the training accuracy of Version 2 attains 89.99%, indicating a 10.61% increase over Version 1, while the validation accuracy exhibits a 6.15% improvement. Moreover, Version 2 shows a significant reduction in training and validation loss, with decreases of 0.3562 and 0.2076, respectively, in comparison to Version 1. These results indicate that the improvements in the input size and activation function adopted in Version 2 have had a positive impact, resulting in improved model performance.

**4.1.2.3 Summary of fine-tuning**

Based on [Figure 4-8](#figure_4_8), the InceptionV3(299\*299) and ensemble stacking(version2) are selected as the parameters for the final two models because of their smaller loss in the validation set meaning better performance.

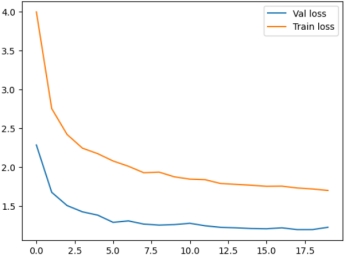
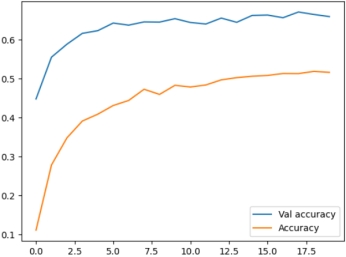
a) The validation loss of different input size’s b) The validation loss of different

in the InceptionV3 model versions in the ensemble stacking model

Figure 4‑8 a) and b) show the comparative results of the two models’ fine-tuning

**4.1.2.4 The results of the four models at different epochs**

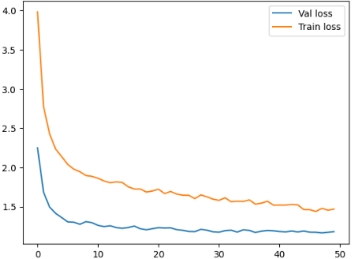
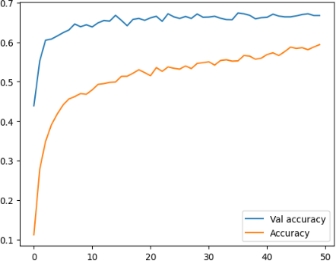
[Figure 4-5](#figure_4_5), and [Figure 4-9](#figure_4_9) to [4-13](#figure_4_13) represent the training results of the three CNN models proposed in the project at epochs 20 and 50. Due to the time limitation of the experiment and the fact that ensemble stacking is composed of these three CNN models, this experiment decided to analyze the performance of epoch on these three CNN models to determine the Epoch value of this model.

a) The validation loss and training loss b) The validation accuracy and training

of the VGG16 model accuracy for the VGG16 model

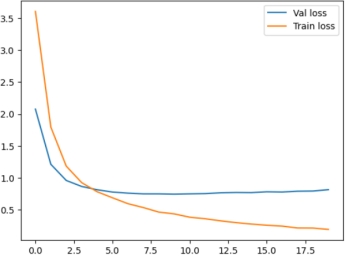
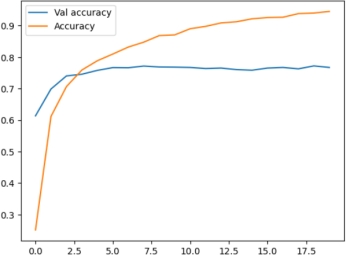
Figure 4‑9 a) and b) show the results of the VGG16 model(Epoch = 20)

a) The validation loss and training loss b) The validation accuracy and training

of the VGG16 model accuracy for the VGG16 model

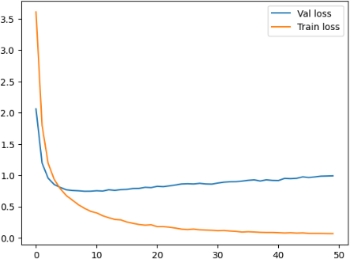
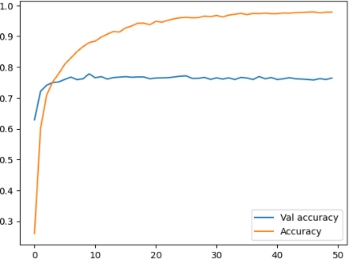
Figure 4‑10 a) and b) show the results of the VGG16 model(Epoch = 50)

a) The validation loss and the training loss b) The validation accuracy and training

of the ResNet50 model accuracy for the ResNet50 model

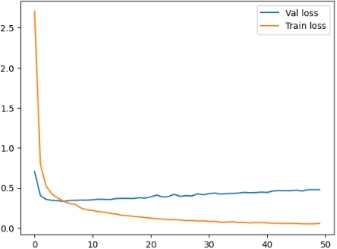
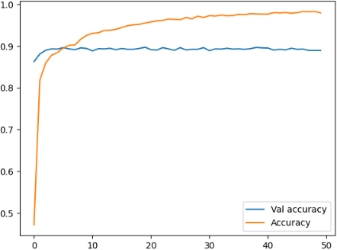
Figure 4‑11 a) and b) show the results of the ResNet50 model(Epoch = 20)

a) The validation loss and the training loss b) The validation accuracy and training

of the ResNet50 model accuracy for the ResNet50 model

Figure 4‑12 a) and b) show the results of the ResNet50 model(Epoch = 50)

a) The validation loss and training loss b) The validation accuracy and training of the InceptionV3(299\*299) model accuracy for the InceptionV3(299\*299) model

Figure 4‑13 a) and b) show the results of the InceptionV3(299\*299) model(Epoch = 50)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Epoch | Train\_Accuracy | Val\_Accuracy | Train\_Loss | Val\_Loss |
| VGG16 | 20 | 51.64% | 65.95% | 1.7018 | 1.2282 |
| 50 | 59.37% | 66.73% | 1.4687 | 1.1815 |
| ResNet50 | 20 | 94.51% | 76.71% | 0.1928 | 0.8149 |
| 50 | 97.86% | 76.47% | 0.0692 | 0.9926 |
| InceptionV3(299\*299) | 20 | 95.85% | 88.89% | 0.1234 | 0.3888 |
| 50 | 97.97% | 88.94% | 0.0585 | 0.4779 |

Table 4‑10: Epoch = 20 VS Epoch = 50

[Table 4-10](#Table_4_10) displays the training accuracy of VGG16 improves by about 8% at 50 epochs, the validation accuracy improves by about 0.8%, and the loss value decreases by about 0.5 points after training for 50 epochs. This suggests that VGG16 can benefit from longer training, so setting the epoch to 50 may be meaningful.

However, for the ResNet50 and InceptionV3 models, after training 50 epochs, their training accuracy and validation accuracy basically did not improve, while the validation loss value increased slightly. This suggests that both models may have been overfitted after 50 epochs, and thus continued training may not produce better results. Therefore, setting the epoch to 50 may not be a good choice.

Based on the data in [Table 4-10](#Table_4_10), it can be seen that in each model, a larger epoch does not necessarily lead to better results, as the risk of overfitting also increases. For the ensemble model, which is composed of several different models, the epoch of the ensemble model for this experiment was set to 20 for training. In summary, the Epoch settings for the four models are shown in [Table 4-11](#Table_4_11) below.

|  |  |
| --- | --- |
| Model | Epoch |
| VGG16 | 50 |
| ResNet50 | 20 |
| InceptionV3 | 20 |
| Ensemble stacking | 20 |

Table 4‑11: The final epoch setting of each model

## The second stage of the experiment

This phase of the experiment will first implement two different data augmentation versions as described in **Chapter 3.1.4** to verify that the second data augmentation strength is more appropriate for the project, and then discuss whether the data augmentation is appropriate for the project. Finally, the four models are trained on the "Dog breed identification" dataset with a divided test set to obtain additional evaluation metrics for each model: testing accuracy, f1-score, recall, precision, confusion matrix, and ROC to select the best model.

### Experimental results for two different data augmentation parameters

[Figures 4-14](#figure_4_14) to [4-16](#figure_4_16) show each model's comparison results of their two data augmentation and the original without data augmentation.

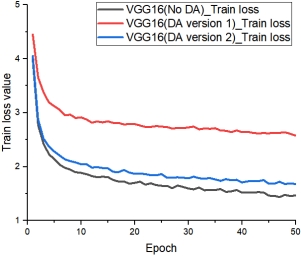
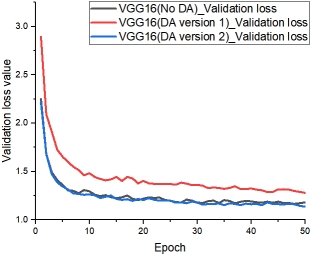
 

Figure 4-14: (a) Figure 4-14: (b)

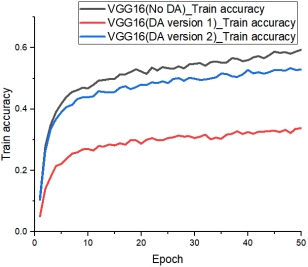
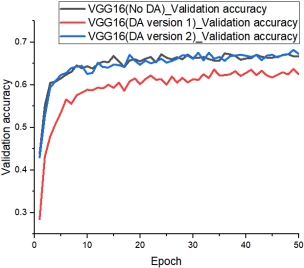
 

Figure 4-14: (c) Figure 4-14: (d)

Figure 4‑14: VGG16 comparative results: (a) and (b) shows train loss and validation loss, (c) and (d) shows train accuracy and validation accuracy

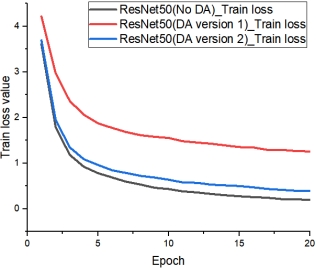
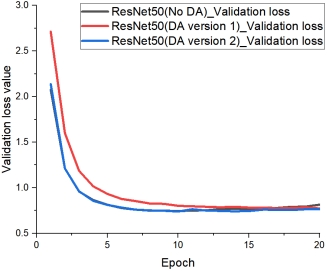
 

Figure 4-15: (a) Figure 4-15: (b)

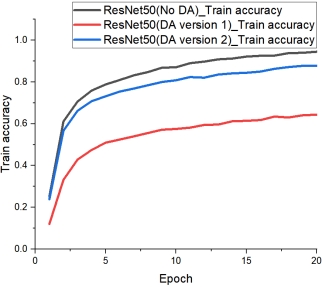
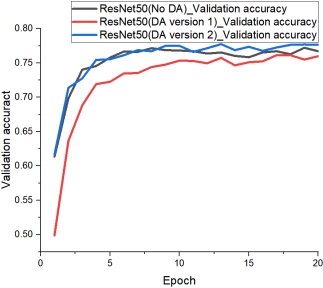
 

Figure 4-15: (c) Figure 4-15: (d)

Figure 4‑15: ResNet50 comparative results: (a) and (b) shows train loss and validation loss, (c) and (d) shows train accuracy and validation accuracy

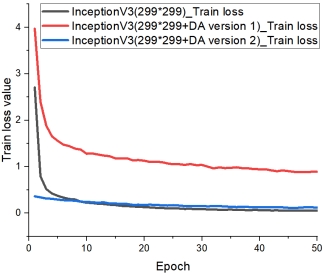
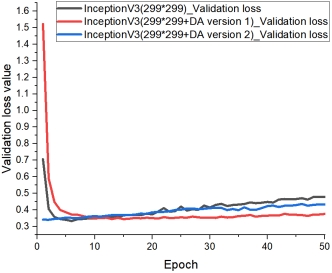
 

Figure 4-16: (a) Figure 4-16: (b)

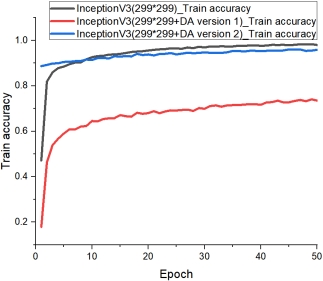
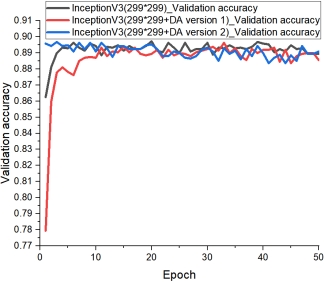
 

Figure 4-16: (c) Figure 4-16: (d)

Figure 4‑16: InceptionV3 comparative results: (a) and (b) shows train loss and validation loss, (c) and (d) shows train accuracy and validation accuracy

Besides, as shown in [Figure 4-17](#figure_4_17), during the implementation of the version 2 data augmentation experiments, for example, there was an oscillation after implementing the data augmentation version 2 for InceptionV3, which occurred during the training of the model. Therefore, the experiments used the early stopping technique to avoid it and successfully made the training process of InceptionV3 and other models smooth as shown in [Figure 4-18](#figure_4_18).

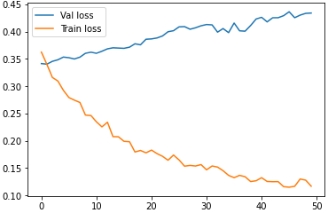
 

Figure 4-17: (a) Figure 4-17: (b)

Figure 4‑17: InceptionV3(DA version 2) results: (a) shows loss results, (b) shows accuracy results

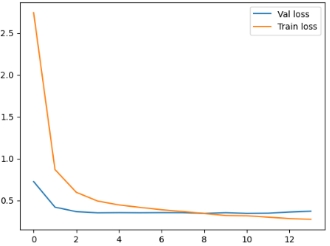
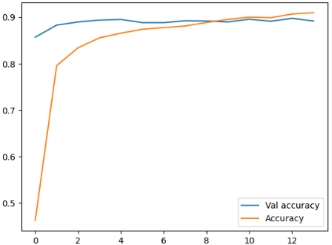
 

Figure 4-18: (a) Figure 4-18: (b)

Figure 4‑18: InceptionV3(DA version 2 + Early stopping) results: (a) shows loss results, (b) shows accuracy results

### Summary of data augmentation experiments

This section will show in detail the experimental versions of each model and the corresponding results, giving whether and which version of data augmentation is used and whether Early stopping is used.

Combining the above images, and the [Table 4-12](#Table_4_12) data, it can be seen that for VGG16, ResNet50, and InceptionV3, although the performance on the training dataset decreases with data augmentation, the performance on the validation dataset, which is used to truly evaluate the performance of the models, improves to a greater or lesser extent. At the same time, after comparison, there is no doubt that the data augmentation version 2 improves the performance of the model too much and affects the training process of the model less, which is more suitable for this project. And the data augmentation version 2 speeds up the training of InceptionV3, which makes the model get a good accuracy rate at the beginning. The addition of early stopping not only reduces the number of training rounds but also saves training time, while the performance of the three models does not deteriorate or even improve slightly.

Therefore, for the three CNN models in this project, the data augmentation and early stopping of version 2 will be adopted.

However, for the ensemble stacking model, the data augmentation and early stopping of version 2 were adopted directly after combining the effects of the three previous CNN models. The validation accuracy is slightly improved, but the most important validation loss is 4 times larger, which seriously reduces the performance of the model. Using early stopping based on DA version 2, the model does not perform as well as it should, even though the number of training rounds is reduced. Therefore, the data augmentation and early stopping techniques are not suitable for the ensemble stacking model in this project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Epoch | Train Accuracy | Validation Accuracy | Train Loss | Validation  Loss |
| VGG16(No DA) | 50 | 59.37% | 66.73% | 1.4687 | 1.1815 |
| VGG16(DA version 1) | 50 | 33.69% | 62.52% | 2.5762 | 1.2790 |
| VGG16(DA version 2) | 50 | 52.89% | 67.27% | 1.6787 | 1.1365 |
| VGG16(DA version2+EarlyStopping) | 32 | 50.44% | 65.12% | 1.7802 | 1.1992 |
| ResNet50(No DA) | 20 | 94.51% | 76.71% | 0.1928 | 0.8149 |
| ResNet50(DA version 1) | 20 | 64.36% | 75.98% | 1.2588 | 0.7738 |
| ResNet50(DA version 2) | 20 | 87.70% | 77.64% | 0.3963 | 0.7670 |
| ResNet50(DA version2+EarlyStopping) | 16 | 85.64% | 77.69% | 0.4613 | 0.7312 |
| InceptionV3(299\*299) | 50 | 95.85% | 88.89% | 0.1234 | 0.3888 |
| InceptionV3(299\*299+DA version 1) | 50 | 73.51% | 88.55% | 0.8895 | 0.3754 |
| InceptionV3(299\*299+DA version 2) | 50 | 95.83% | 89.09% | 0.1170 | 0.4339 |
| InceptionV3(299\*299+DA version 2 + Early Stopping) | 14 | 90.94% | 89.14% | 0.2722 | 0.3697 |
| Ensemble stacking(version2) | 20 | 89.99% | 95.97% | 0.3215 | 0.1255 |
| Ensemble stacking(version 2+DA version 2) | 20 | 86.28 | 87.33% | 0.4409 | 0.5576 |
| Ensemble stacking(version 2+DA version 2 + Early Stopping) | 12 | 85.09% | 86.25% | 0.4908 | 0.5875 |

Table 4‑12: A summary of data augmentation experiments’ results

## The third stage of the experiment

This phase of the experiment will train the four models in the "Dog breed identification" dataset which has the test dataset version and the "70 Dog Breeds-Image" Dataset. The corresponding testing accuracy, f1-score, recall, and precision will be used not only to select the best model but also to validate the idea that the ensemble model performs better in more complex dog breed classification problems.

### 70 dog breeds VS 120 dog breeds

[Figure 4-19](#figure_4_19) shows in detail the model performance of these four models for the two datasets with the number of breeds of 70 and 120. When the dog classification problem is simpler, i.e., 70 dog breeds, there is no doubt that all four models perform well, with ResNet50 performing best and InceptionV3 trailing by a small margin, followed by ensemble stacking and VGG16 models. However, as the classification problem becomes more complex, the performance of all four models decreases in varying degrees. The ensemble model, however, not only ranks first with 90.2% accuracy and other evaluation values but also has the least degradation in performance. The other three CNN models showed significant performance degradation, especially the ResNet50 and VGG16 models, which lost almost 20% to 30% of their original performance.

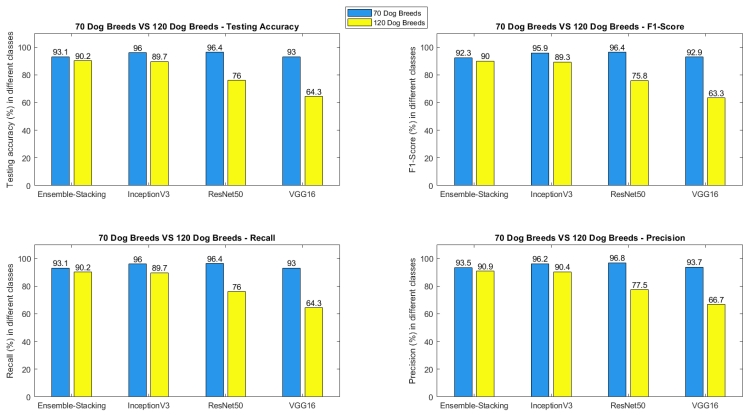


Figure 4‑19: Performance comparison of the four models on two different datasets

To sum up, the ensemble stacking is the best model for this project, providing a more robust, accurate, and robust model for the dog breed classification domain.

|  |  |  |  |
| --- | --- | --- | --- |
| **Researchers** | **Class Number** | **Total Images** | **Performance** |
| Chanvichitkul M et al. [18] | 35 | 700 | Accuracy≈93% |
| Varshney A et al. [2] | 120 | 2050 | Accuracy=85% |
| Rishita M et al. [19] | 133 | 7515 | Accuracy=87.42% |
| Shah B et al. [20] | 103 | 8351 | Accuracy=95.5% |
| Borwarnginn P et al. [21] | 133 | 8351 | Accuracy=96.75% |
| Agarwal A et al. [22] | 70 | 10000 | Accuracy≈91% |
| Liang B et al. [24] | 120 | 10222 | Accuracy=67.17%, |
| Project Proposed ensemble model | 120 | 10222 | Accuracy=90.02% |

Table 4‑13: Comparison Analysis with Other State-of-the-art models

Referring to the research mentioned in [Table 4-13](#Table_4_13), the proposed ensemble model achieved a testing accuracy of 90.02% on a dataset of 10222 images and 120 dog breeds, which is comparable to or higher than models that used a single CNN architecture. For instance, Chanvichitkul M et al. [18] obtained about 93% accuracy with 700 images and 35 dog breeds, while Varshney A et al. [2] achieved an 85% accuracy on 2060 images and 120 dog breeds. And Rishita M et al. [19] got 87.42% accuracy on the 133 classes with 7515 images. Similarly, Agarwal A. et al. [22] achieved an accuracy rate of almost 91% with 10000 images and 70 dog breeds. These findings indicate that combining multiple CNN models with the proposed ensemble technique can enhance classification accuracy. Nonetheless, Shah B et al [20] attained a 95.5% accuracy on 8351 images and 103 dog breeds, while Borwarnginn P et al [21] achieved 96.75% accuracy on 8351 images and 133 breeds, surpassing the proposed model's performance. However, the proposed model utilized a dataset of over 1.2 times the amount of data used in these studies. Therefore, it is not appropriate to conclude that the proposed model is inferior to these models. These indicate that combining other CNN models can boost performance.

Regarding the ensemble technique, Liang B et al [24] achieved an accuracy rate of 67.17% using the same base models but a different ensemble algorithm for the same dataset. The proposed ensemble model attained significantly higher accuracy. The improvement in performance can be attributed to the stacking technique, which is better suited to the dog breed classification problem, enabling the model to learn more complex features and patterns.

In summary, the proposed ensemble model demonstrates competitive accuracy compared to other state-of-the-art models. In the future, implementing techniques that can enhance accuracy before inputting the images into the network, selecting better CNN models to combine, and employing different ensemble techniques can further improve the proposed model's performance.

### The best model’s detailed results

This section shows all the results of the ensemble stacking model on the latest "Dog breed identification" dataset. [Figure 4-20](#figure_4_20) shows the details of the training process of the model. [Figures 4-21](#figure_4_21) to [4-23](#figure_4_23) show the performance of the model on the testing dataset.

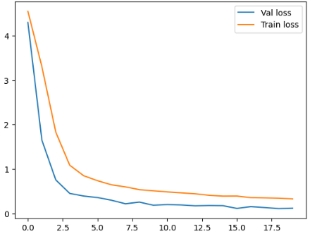
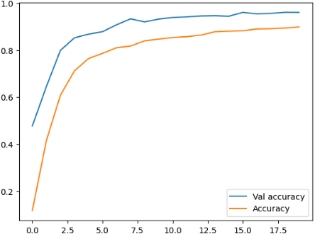
 

Figure 4-20: (a) Figure 4-20: (b)

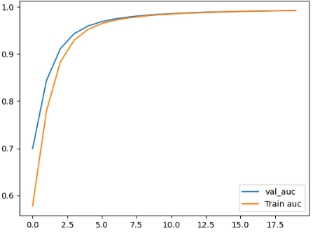
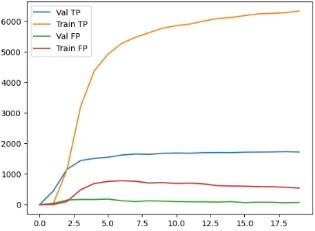
 

Figure 4-20: (c) Figure 4-20: (d)

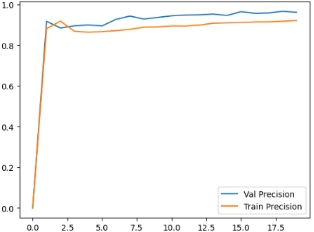
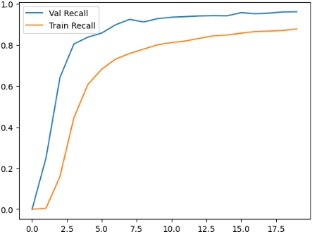
 

Figure 4-20: (e) Figure 4-20: (f)

Figure 4‑20: The ensemble stacking’s training information on the “Dog breed identification” dataset: (a)-(b)shows loss and accuracy, (c)-(d) shows AUC, TP, and FP, (e)-(f) shows precision and recall

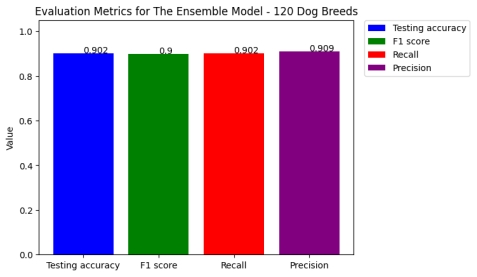


Figure 4‑21: The best model’s testing accuracy, f1-score, recall, precision value

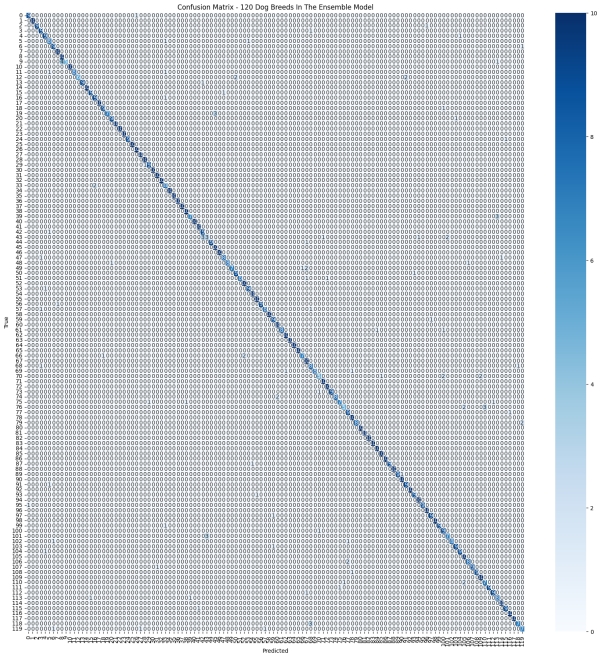


Figure 4‑22: The best model’s confusion matrix

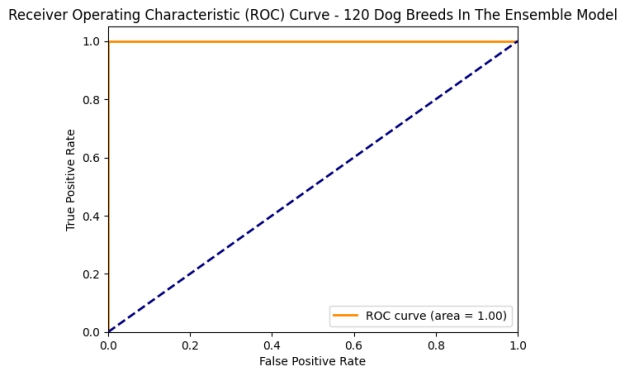


Figure 4‑23: The best model’s ROC graph

## The deployment of the best model

Using the flask framework, the ensemble stacking model is deployed to a mini web application for popularizing dog breeds' information and identifying what breeds they belong to, which not only solves the previously mentioned problem but also provides a powerful aid to the previously mentioned audience. There are also the following benefits.

* Increased usage and testing -> Collecting more data -> Improve the effectiveness

and reliability of the model.

* Model performance becomes more intuitive.
* Gathering user feedback to improve the model's performance -> Identifying and
* solving problems with the model in practice -> Optimising the model and algorithms

The following images show each page of the website.



Figure 4‑24: The website index page

After clicking "GET IN", users will be redirected to the page introducing the six most popular dog breeds ([Figure 4-25](#figure_4_25))

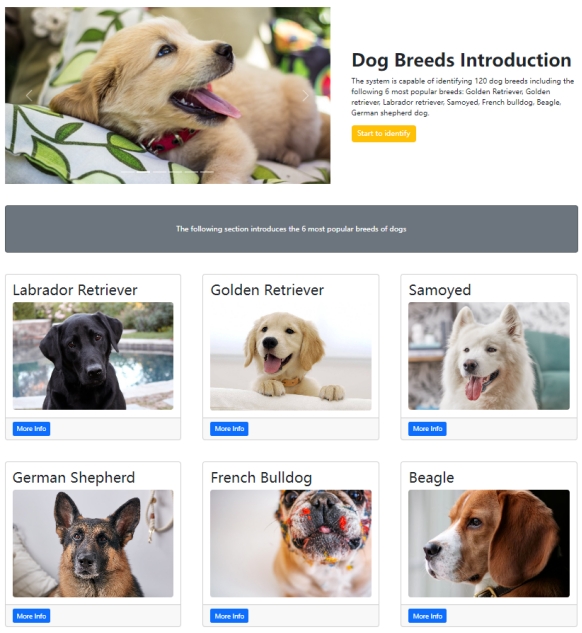


Figure 4‑25: The six most popular dogs’ introduction page

Clicking on "Start to identify" will take users to the page where users can identify the breed of the dog they uploaded ([Figure 4-26](#figure_4_26))

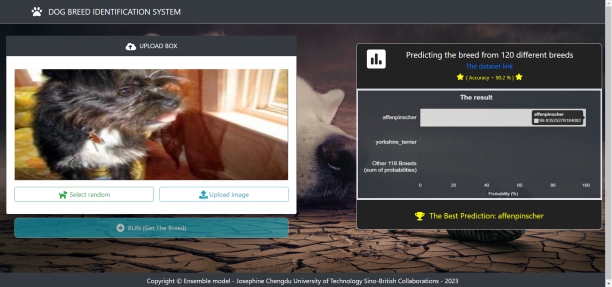


Figure 4‑26: Dog breed identification page

# **Professional Issues**

## Project Management

### Activities

The part information is shown in [Table 5-1](#Table_5_1).

|  |  |
| --- | --- |
| Tasks | State |
| A. Conduct an image classification techniques research | Complete |
| B. Conduct a fine-classifier techniques research | Complete |
| C. Learn three or more CNN models | Complete |
| D. Conduct single model, dual models, and ensemble models in the image classification domain’s research and learn ensemble algorithms | Complete |
| E. Search suitable ensemble modes for fine-classifier image classification codes | Complete |
| F. Study and understand the codes. | Complete |
| G. Implement different CNN models to fit the project dataset separately. | Complete |
| H. Designing and doing the comparative analysis, then choosing the best single CNN model. | Complete |
| I. Combine all single CNN models into a network to fit the project dataset by using the stacking combination strategy. | Complete |
| J. Doing the comparative analysis in all implemented project models. | Complete |
| K. Based on all analyses, choose the best model. | Complete |
| L. Doing the data augmentation and training all models again. | Completed |
| M. Doing the analysis of the changes that data augmentation brings. | Completed |
| N. Training these four models on another dataset | Completed |
| O. Doing the comparative analysis of the models’ performances on the two datasets, choose the best model again. | Completed |
| P. Designing a Graphical User Interface to more intuitive showing the project results. | Completed |
| Q. Implementing the Graphical User Interface. | Completed |

Table 5‑1: The complete/uncompleted tasks for each objective

### Schedule

The schedule is represented by Gantt as below:

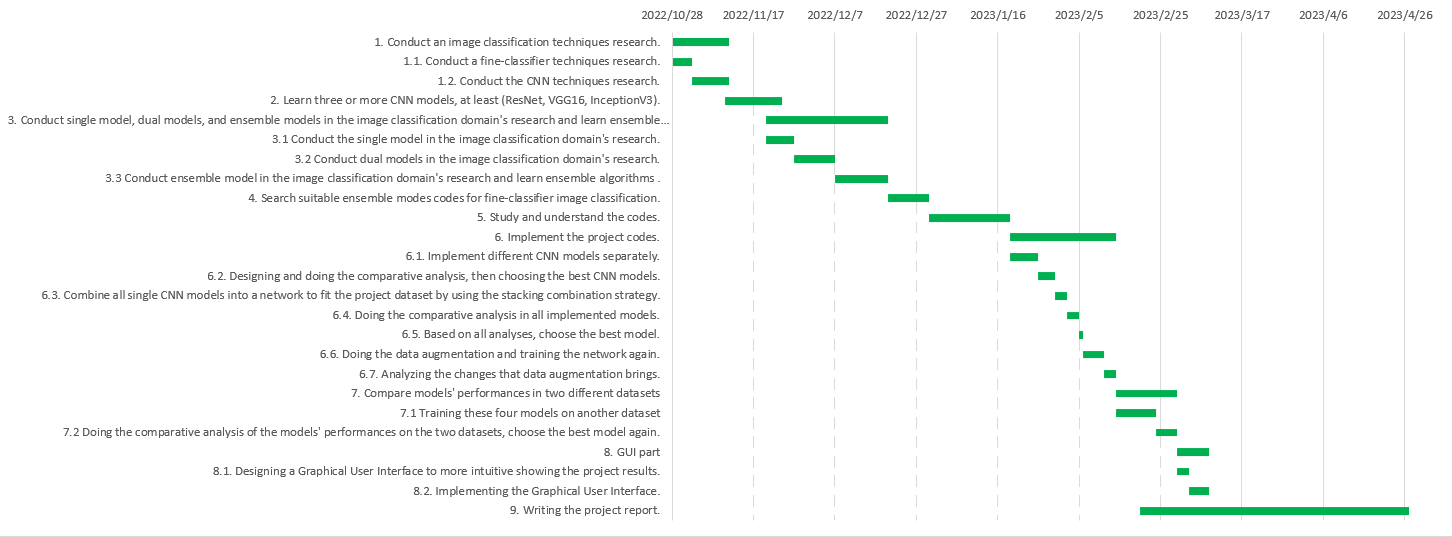


Figure 5‑1: The project schedule

### Project Data Management

1. All deliverables will be stored in a local file (as shown in [Figure 5-2](#Figure_5_2)) and will upload to Baidu Drive except codes.
2. Upload the project codes on GitHub.
3. Using Mendeley to manage references, save all pdf files in a local folder.

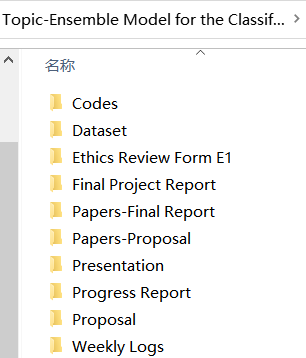


Figure 5‑2: The local folder structure

### Project Deliverables

1. The project proposal
2. Ethics Review Form
3. Weekly reports
4. Progress Report
5. Final Project Report
6. Project codes
7. Project presentation’s ppt
8. Project poster
9. Project presentation
10. The website link to the project dataset

## Risk Analysis

The part information is shown in [Table 5-2](#Table_5_2).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Potential Risk | Potential Causes | Severity | Likelihood | Risk | Mitigation |
| Miss the deadline | Unrealistic schedules | 4 | 2 | 8 | Need better project control |
| Developing the wrong functions to waste time | 5 | 1 | 5 | Follow the functional requirements at the start. |
| Poor time management | 4 | 3 | 12 | Make a Gantt plan first |
| The time to train models is so long because of the poor local environment | 2 | 3 | 6 | Using the GPU cloud service to train models |
| The model gets terrible classified accuracy in real life. | The training dataset provides not enough training samples to learn. | 3 | 3 | 9 | Using Data Augmentation to add more training samples. |
| The model section's criteria are just one, too partial to choose a real best model. | 4 | 3 | 12 | Add other criteria to select models |
| The model is overfitting | 4 | 2 | 8 | Using Dropout |
| Problems with the Data testing section | 3 | 1 | 3 | Following the data testing guideline before training models |
| The application is not working correctly | The app has been maliciously attacked | 5 | 3 | 15 | Analyze the corresponding malicious attack and find the corresponding solution. |

Table 5‑2: Risk Analysis

## Professional Issues

Identification and discussion of relevant legal, social, ethical, and environmental issues in the context of the project, with reference to professional codes of conduct, e.g. BCS, ACM:

The use of deep learning models to identify dog breeds raises legal concerns, particularly in countries or regions with breed-specific legislation or restrictions on certain breeds of dogs. Misidentifying a banned breed as a different breed could lead to legal issues for the dog owner, and misidentifying a non-banned breed as a banned breed could result in false accusations and legal consequences. Therefore, it is crucial to ensure that the deep learning models used in this project are accurate and unbiased to avoid legal issues.

Deep learning models can perpetuate stereotypes and discrimination against certain dog breeds, which can have a negative impact on the public perception of these breeds and lead to unfair treatment or even mistreatment of these dogs. For example, misclassifying a docile breed as a fierce dog could result in dogs of that breed being viewed as dangerous and subject to breed-specific legislation, even though they pose no real threat. Therefore, it is essential to consider the social impact of the project's findings and ensure that they are not perpetuating harmful stereotypes.

Using animals in research raises ethical considerations, and it is important to treat animals with respect and care. In this project, it is crucial to ensure that the dogs used in the Dog Breed Identification dataset are treated ethically and that their welfare is not compromised in any way. Additionally, misidentifying a dog's breed could have serious ethical implications for their health and well-being, as different breeds have different nutritional requirements and may be predisposed to certain health conditions. Therefore, it is essential to consider the ethical implications of the project's findings and ensure that they do not harm the dogs involved.

Although this project is primarily focused on identifying dog breeds, it is worth considering its potential impact on the environment. Breeding and caring for dogs requires significant resources, including food, water, and land, which can have negative environmental consequences. Furthermore, dogs can have an impact on wildlife and ecosystems. And breeding dogs without proper knowledge of their breed can result in ineffective breeding, leading to a greater waste of resources and further environmental harm. Therefore, it is important to be mindful of the environmental impact of dog breeding and ownership and take measures to mitigate it.

The British Computer Society (BCS) and the Association for Computing Machinery (ACM) provide professional codes of conduct that offer guidance on ethical and professional practices when working with deep learning models. These codes emphasize the importance of ensuring transparency and accountability in the model's design and use, protecting the privacy and security of data, and avoiding harm to individuals or groups. It is essential to adhere to these codes of conduct when working on this project to ensure that it is conducted in an ethical and professional manner.

The project's findings have practical applications in animal hospitals and the development of more accurate and unbiased breed identification tools. Misidentifying a dog's breed in an animal hospital could result in incorrect prescriptions and potentially serious consequences for the dog's health. Therefore, accurate and unbiased breed identification tools are essential for ensuring that dogs receive appropriate care. The project’s findings could also be used to develop more effective and ethical breed identification tools that avoid perpetuating harmful stereotypes and discrimination against certain breeds. The project is based on an ensemble model which utilized VGG16, ResNet 50, and Inceptionv3 models, and the Dog Breed Identification dataset and 70 Dog Breeds-Image Data Set, which are open-source datasets, are used to evaluate the performance of the proposed model for this project. The project code is publicly available on Github allows for transparency and scrutiny of the methods and results by the wider community.

Overall, the legal, social, ethical, and environmental implications of this project are multifaceted and require careful consideration. Adherence to professional codes of conduct and a commitment to ethical and responsible practices are essential for ensuring that the project's findings have a positive impact and avoid perpetuating harmful stereotypes and practices.

# **Conclusion**

This project explored the use of various CNN models and a stacking ensemble model for the classification of 120 different dog breeds. The project results show that while individual CNN models like InceptionV3, ResNet50, and VGG16 are effective for simpler classification tasks, they may not perform as well in more complex problems. However, the project-proposed ensemble stacking model which is the best model, achieved a testing accuracy of 90.02%, F1-score of 90.00%, recall of 90.20%, and precision of 90.90% on the 120 dog breeds classification dataset, demonstrating its superiority over individual models. Furthermore, the project analysis showed that when changing the classes number 70 to 120, the model's performance dropped with only 2.9% of test accuracy, 2.3% of F1-score, 2.9% of recall, and 2.6% of precision, which are small fluctuations. Whereas, the InceptionV3, ResNet50, and VGG16 models decreased by about 6%, 21%, and 31% of their performance compared to their performance on the "70 dog breeds" dataset. This finding highlights the importance of using ensemble models for complex classification tasks, especially in real-world applications such as dog policies, resident safety, and veterinary work efficiency.

As for future work, implementing an object detection algorithm before feeding images into the model may further improve the performance of the ensemble model. Additionally, choosing other better CNN models to combine and exploring other ensemble algorithms, such as boosting could lead to even better results. Overall, this project provides a cost-effective, efficient, and powerful tool for the classification of multiple dog breeds, with potential applications in various industries.

# **References**

[1] J. T. Teo, S. J. Johnstone, S. S. Römer, and S. J. Thomas, “Psychophysiological mechanisms underlying the potential health benefits of human-dog interactions: A systematic literature review,” *International Journal of Psychophysiology*, vol. 180. Elsevier B.V., pp. 27–48, Oct. 01, 2022. doi: 10.1016/j.ijpsycho.2022.07.007.

[2] A. Varshney, A. Katiyar, A. K. Singh, and S. S. Chauhan, “Dog Breed Classification Using Deep Learning,” in *2021 International Conference on Intelligent Technologies, CONIT 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021. doi: 10.1109/CONIT51480.2021.9498338.

[3] X. Cheng and H. Lei, “Remote Sensing Scene Image Classification Based on mmsCNN–HMM with Stacking Ensemble Model,” *Remote Sens (Basel)*, vol. 14, no. 17, Sep. 2022, doi: 10.3390/rs14174423.

[4] Z.-H. (Computer scientist) Zhou, *Ensemble methods : foundations and algorithms*. 2012.

[5] G. R. You, Y. R. Shiue, C. T. Su, and Q. L. Huang, “Enhancing ensemble diversity based on multiscale dilated convolution in image classification,” *Inf Sci (N Y)*, vol. 606, pp. 292–312, Aug. 2022, doi: 10.1016/j.ins.2022.05.064.

[6] A. Ramirez, E. Ohn-Bar, and M. M. Trivedi, “Go with the Flow: Improving Multi-view Vehicle Detection with Motion Cues,” in *2014 22nd International Conference on Pattern Recognition*, IEEE, Aug. 2014, pp. 4140–4145. doi: 10.1109/ICPR.2014.709.

[7] P. Domingos, “A few useful things to know about machine learning,” *Commun ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012, doi: 10.1145/2347736.2347755.

[8] M. Rezaei and M. Terauchi, “Vehicle Detection Based on Multi-feature Clues and Dempster-Shafer Fusion Theory,” 2014, pp. 60–72. doi: 10.1007/978-3-642-53842-1\_6.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[10] K. A. Rudakov, A. S. Pikalov, and D. A. Yudin, “Analysis of methods of tracking objects on a sequence of images,” *International conference “Actual problems of robotics and automation"*, Belgorod, pp. 154–158, 2015.

[11] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” Mar. 2018.

[12] V. D. Kustikova, *Methods and algorithms for analysis of motion trajectories in solving the problem of video detection of vehicles*. The Nizhny Novgorod state. University of. Ni Lobachevsky, Nizhny Novgorod, 2015.

[13] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” Jun. 2014, [Online]. Available: http://arxiv.org/abs/1408.5093

[14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[15] C. Szegedy *et al.*, “Going Deeper with Convolutions,” Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.4842

[16] D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, “Image-Based malware classification using ensemble of CNN architectures (IMCEC),” *Comput Secur*, vol. 92, May 2020, doi: 10.1016/j.cose.2020.101748.

[17] R. Minetto, M. Pamplona Segundo, and S. Sarkar, “Hydra: An ensemble of convolutional neural networks for geospatial land classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6530–6541, Sep. 2019, doi: 10.1109/TGRS.2019.2906883.

[18] M. Chanvichitkul, P. Kumhom, and K. Chamnongthai, “Face recognition based dog breed classification using coarse-to-fine concept and PCA,” in *2007 Asia-Pacific Conference on Communications, APCC*, 2007, pp. 25–29. doi: 10.1109/APCC.2007.4433495.

[19] M. V. S. Rishita and T. A. Harris, “Dog Breed Classifier using Convolutional Neural Networks,” *International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, 2018.

[20] B. K. Shah, A. Kumar, and A. Kumar, “Dog breed classifier for facial recognition using convolutional neural networks,” in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 508–513. doi: 10.1109/ICISS49785.2020.9315871.

[21] P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn, and W. Kusakunniran, “Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning,” *11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2019.

[22] A. K. Agarwal, V. Kiran, R. K. Jindal, D. Chaudhary, and R. G. Tiwari, “Optimized Transfer Learning for Dog Breed Classification,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 2022, no. 1s, pp. 18–22, 2022, [Online]. Available: https://www.researchgate.net/publication/364777086

[23] D. Agrawal, S. Minocha, S. Namasudra, and S. Kumar, “Ensemble Algorithm using Transfer Learning for Sheep Breed Classification,” in *SACI 2021 - IEEE 15th International Symposium on Applied Computational Intelligence and Informatics, Proceedings*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 199–204. doi: 10.1109/SACI51354.2021.9465609.

[24] B. Liang, Z. Wang, and G. Huang, “Dog Breed Classification using Fine-tuned pretrained Models and Ensemble Models,” 2022. [Online]. Available: https://github.com/NicholasL4/Dog-Breed-Classification-using-Ensemble-Pretrained-Models

[25] K. He and J. Sun, “Convolutional Neural Networks at Constrained Time Cost,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.1710

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway Networks,” May 2015, [Online]. Available: http://arxiv.org/abs/1505.00387

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Dec. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[28] W. Li *et al.*, “Classification of High-Spatial-Resolution Remote Sensing Scenes Method Using Transfer Learning and Deep Convolutional Neural Network,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 13, pp. 1986–1995, 2020, doi: 10.1109/JSTARS.2020.2988477.

[29] C. Wang *et al.*, “Pulmonary image classification based on inception-v3 transfer learning model,” *IEEE Access*, vol. 7, pp. 146533–146541, 2019, doi: 10.1109/ACCESS.2019.2946000.

[30] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *Proceedings of the 32 nd International Conference on Machine Learning*, vol. 1, pp. 448–456, 2015.

[31] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, “Understanding Batch Normalization,” 2018.

[32] B. H. Nayef, S. N. H. S. Abdullah, R. Sulaiman, and Z. A. A. Alyasseri, “Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks,” *Multimed Tools Appl*, vol. 81, no. 2, pp. 2065–2094, Jan. 2022, doi: 10.1007/s11042-021-11593-6.