Project Report

1. INTRODUCTION

1.1 Project Overview

Identification of dog breeds has several uses. It aids in comprehending the unique care requirements, behavioral patterns, and underlying health issues of dogs. It can help with training methods, finding dogs good owners, and making sure the environment is healthy for the dogs overall. It can also be useful in legal or regulatory contexts, particularly when it comes to circumstances in which particular breeds may be subject to particular rules or limitations. And transfer learning makes use of pre-trained models that have gleaned features from a sizable dataset, it is utilized to identify dog breeds. Using this pre-existing knowledge, dog breeds with smaller, more specialized datasets can be identified more quickly and accurately. It facilitates comprehension of a dog's behavior, underlying medical conditions, and particular care requirements.Because it uses pre-trained models that have learned features from a large dataset, transfer learning is used to identify dog breeds. With smaller, more specialized datasets, it is possible to identify dog breeds more quickly and accurately by leveraging the existing knowledge.With this method, breed identification tasks can be completed with high performance and a reduced requirement for vast quantities of labeled data and computing power.

In this project, a method for recognising dog breeds based on facial image recognition is presented. The suggested method uses a deep learning-based methodology to identify the breeds of these animals. Using the publicly available dog breed dataset, pre-trained Convolutional neural networks (CNNs) are first trained through a process known as transfer learning. Subsequently, different settings for image augmentation are also applied on the training dataset to enhance the classification performance.

1.2 Purpose

Using transfer learning to identify dog breeds helps to comprehend breed diversity, customize healthcare, place dogs with appropriate owners, maintain legal compliance, create efficient training plans, and encourage well-informed decisions regarding the welfare and management of dogs.

2. LITERATURE SURVEY

2.1 Existing problem

The following are existing obstacles to for dog breed identification:

I. Variations in learned features make it challenging to modify pre-trained models to correctly identify dog breeds.

II. Difficulties in achieving the ideal fine-tuning equilibrium and managing overfitting when adjusting the model.

III. Unbalanced datasets that favor some breeds over others and ignore others.

IV. Difficulties in correctly recognizing dogs that have features from more than one breed, or mixed breeds.

V. Restraints on the model's applicability to uncommon or uncommon breeds.

These problems lead to less accurate solutions.

2.2 References

I. [https://towardsdatascience.com/dog-breed-classification-using-cnns-and- transfer- learning-e36259b29925](https://towardsdatascience.com/dog-breed-classification-using-cnns-and-transfer-learning-e36259b29925)

II. https://link.springer.com/article/10.1007/s11633-020-1261-0

III. https://cvr.ac.in/ojs/index.php/cvracin/article/view/570

IV.https://dev.to/nicolasvallee/using-transfer-learning-and-tensorflow-to-identify-dog-breeds-from-images-5b4b

2.3 Problem Statement Definition

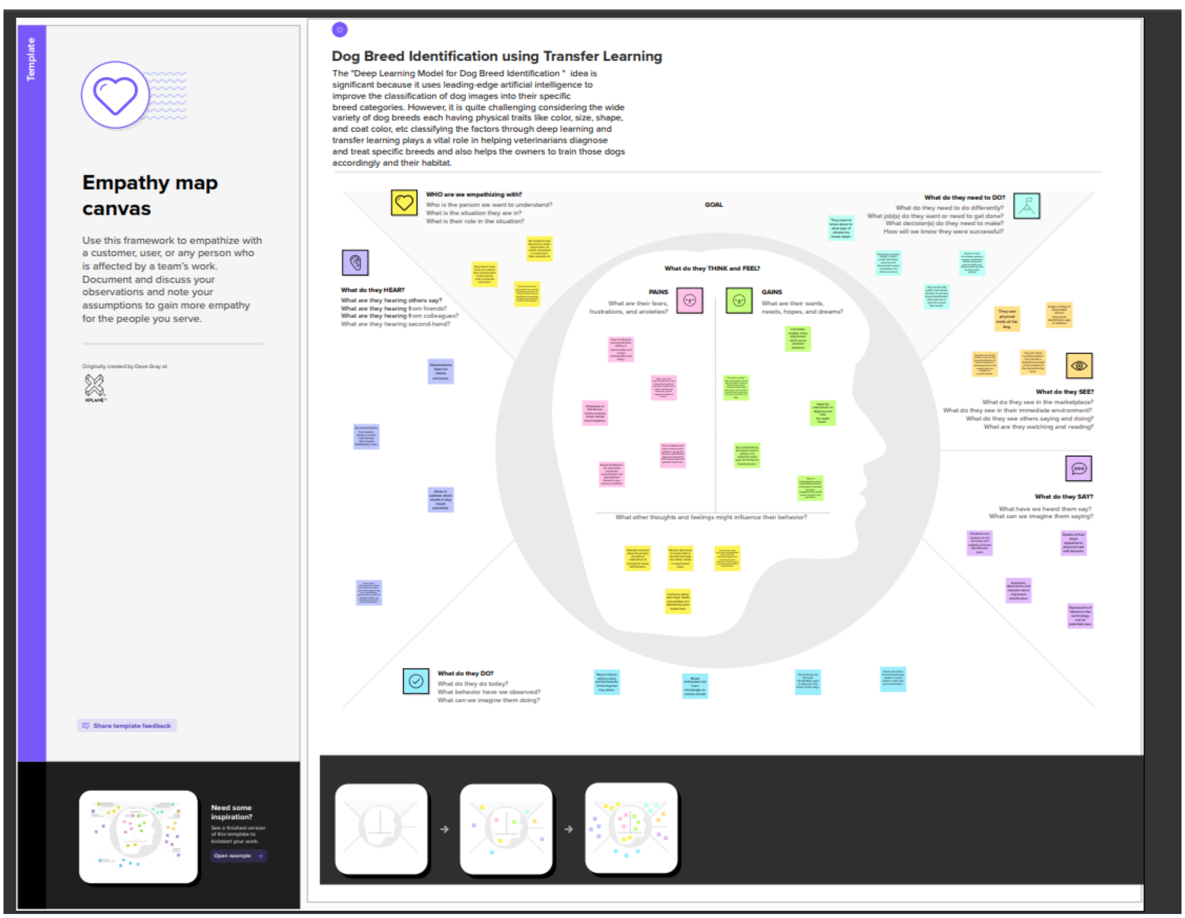
To address issues with breed misclassification, bias, and model generalization, and to build user confidence in the accuracy of dog breed predictions produced by a transfer learning-based model. This entails putting in place strong assessment metrics, transparent model performance, and strategies to manage prediction uncertainty in order to give users trustworthy and intelligible information about the model's strengths and weaknesses.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map helps to map what a design team knows about the potential audience. This tool helps to understand the reason behind some actions a user takes deeply.

<u>Dog Breed Identification Using Transfer Learning:</u>



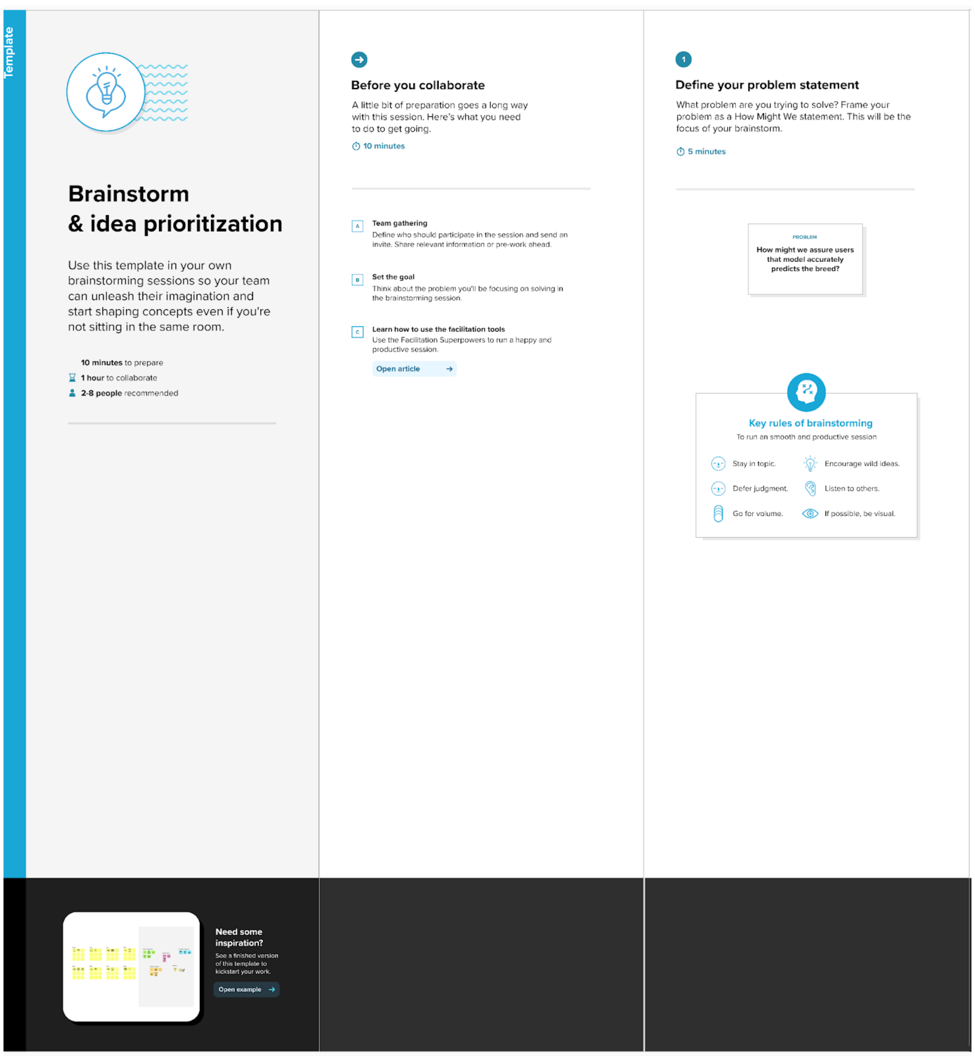
Link: https://app.mural.co/t/empathymapfordogbreedidentif4316/m/empathymapfordogbreedidentif4316/1698467317340/e893f3ba3f3fe1d54dfa0272bc5674e93f492c23?sender=ufbac058c0ed86d3f24a46657

3.2 Ideation & Brainstorming

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge

Link for Brainstorming: https://app.mural.co/t/empathymapfordogbreedidentif4316/m/empathymapfordogbreedidentif4316/1698593425671/992089fbf1068ea1c52e3be462edc99de26fd744?sender=ufbac058c0ed86d3f24a46657

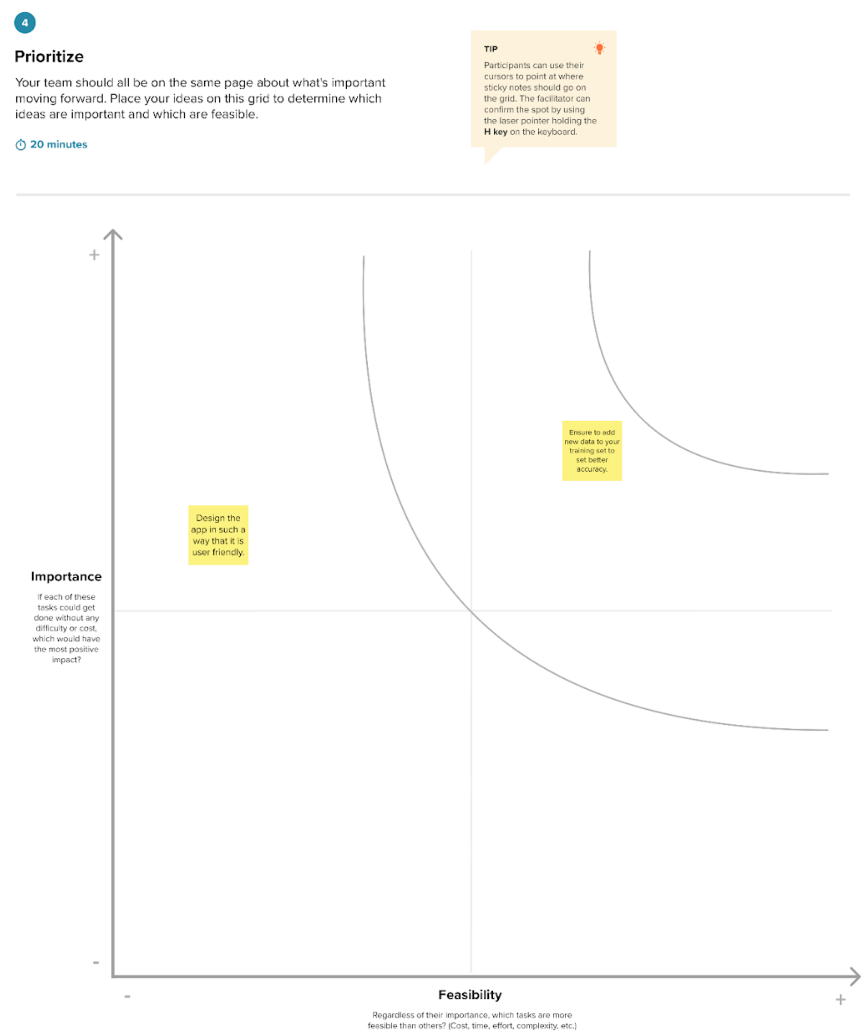
Step-1: Team Gathering, Collaboration and Select the Problem Statement

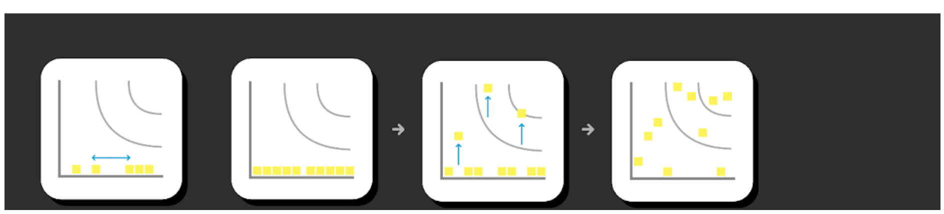


Step-2:

Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization





4. **REQUIREMENT ANALYSIS**

4.1 Functional requirement

i. Image Input

ii. Image Processing : Normalizing,resizing the image

iii. Breed recognition Algorithm : Build a model to predict breed of the image.(ImagetNetv2)

iv. New Breed should update in the data.

v. User Interface: We need an interface for users to interact.

vi. Maintenance of site.

vii. Feedback Mechanism

4.2 Non-Functional requirements

i. Performance of the system or model.

ii. Scalable.

iii. Traffic congestion of the website.

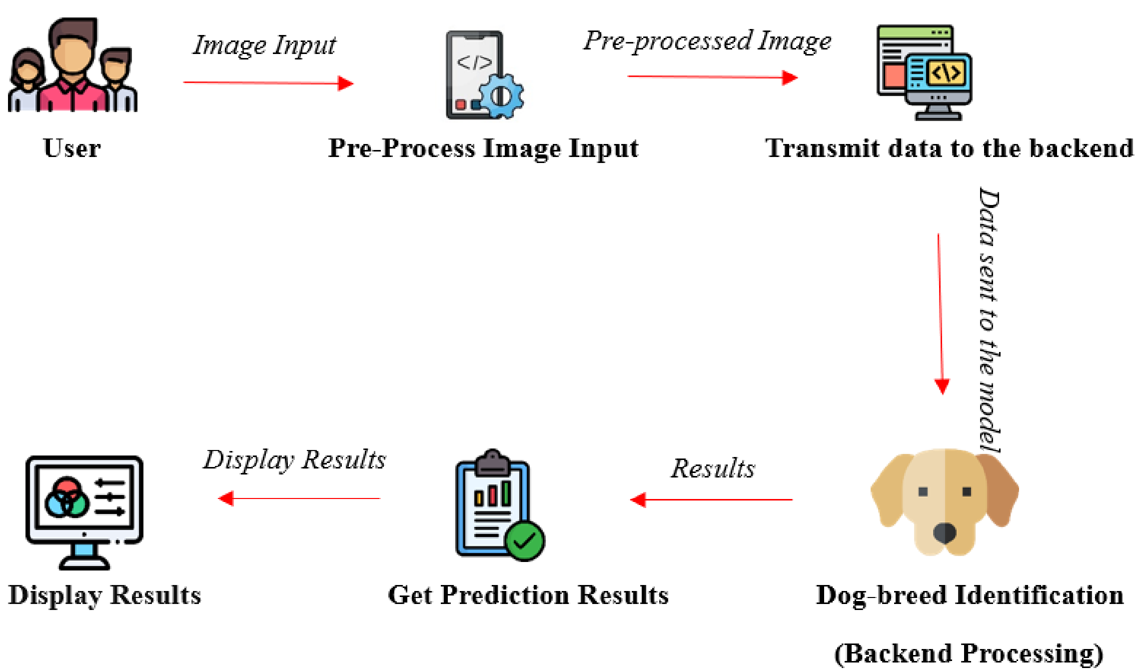
iv. Response Time.

v. Privacy.

vi. Support.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Users	Image Uploadation	USN-1	As a user, I want to upload pictures of a dog to identify the breed.	The system should allow the user to upload dog images in common formats (e.g., JPEG, PNG).	High	Sprint-2
	Dog-Breed Identification	USN-2	As a user, I want to view the prediction results and understand the basic behavior and history of dogs.	The user should be able to view a clear and understandable prediction report indicating the breed of the dog, with relevant details.	Medium	Sprint-2
AI/ML Developers	Model Development	USN-3	Improving Machine Learning Model for Dog-Breed Identification.	The ML developer should be able to access the model's source code, implement enhancements, and validate the changes through testing.	Medium	Sprint-3

5.2 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram

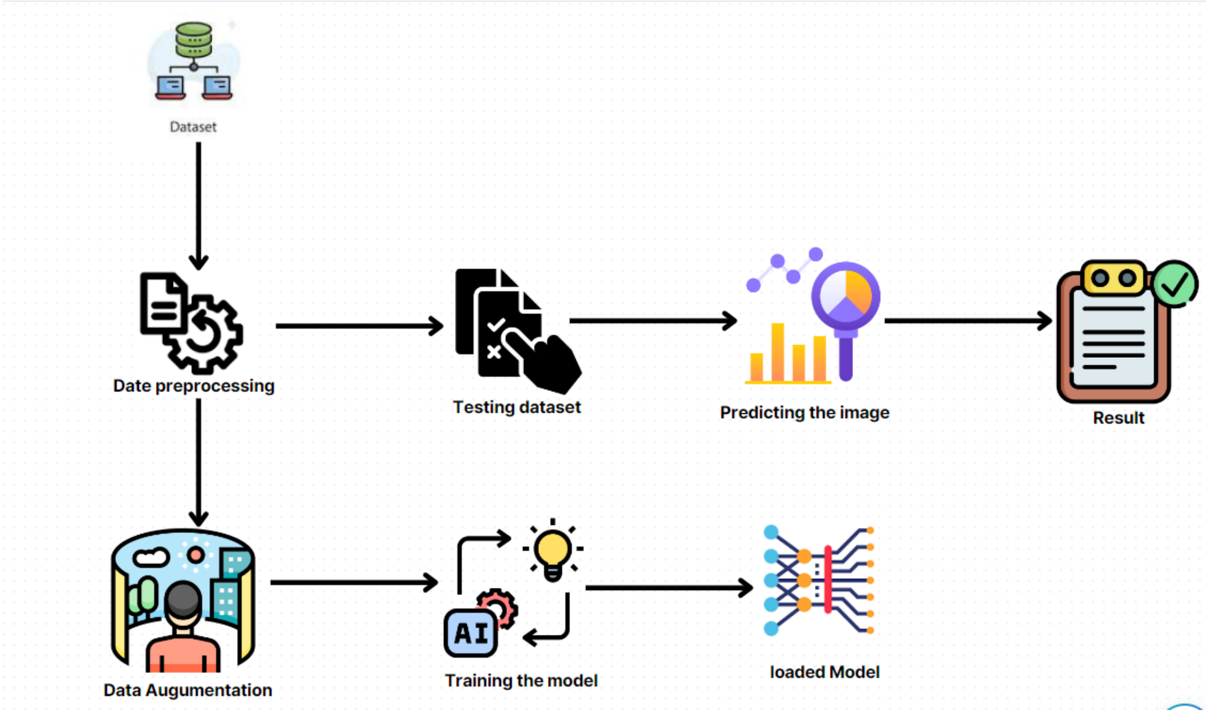


Figure 1: Architecture and data flow of the dog breed identification Model

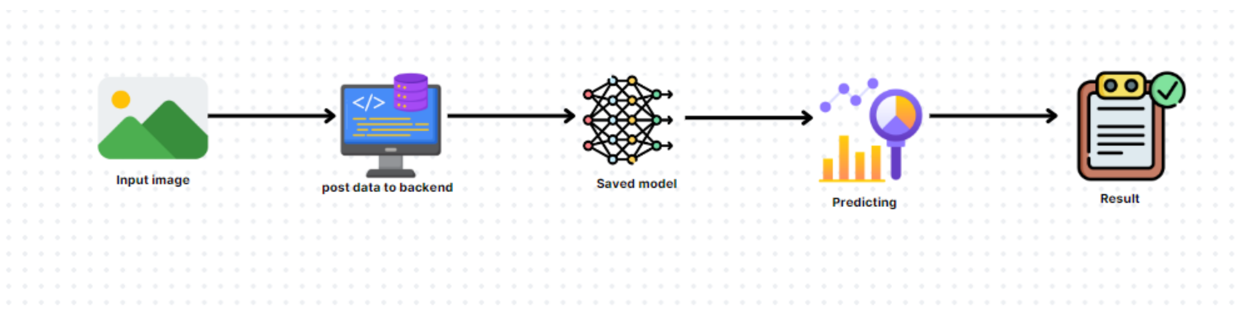
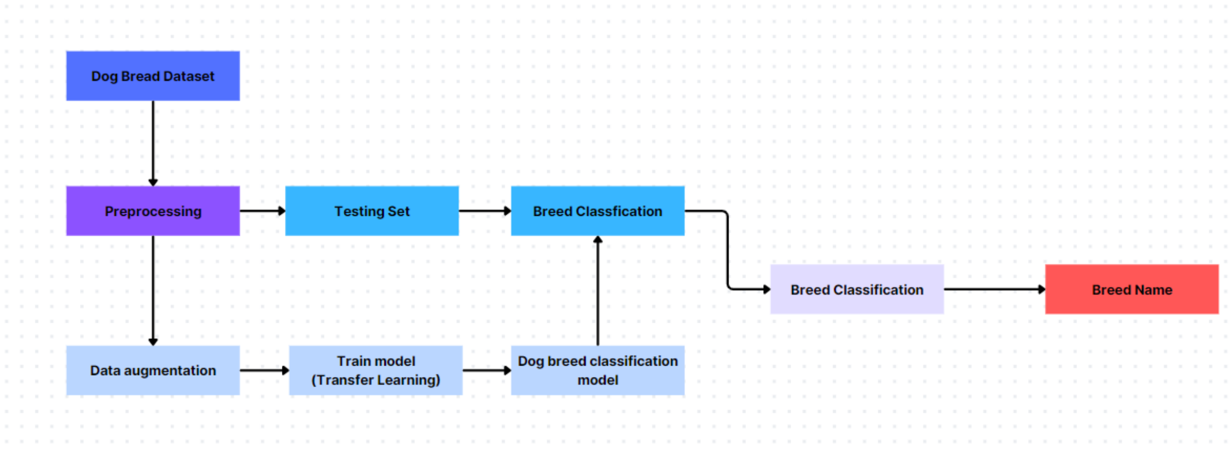


Figure 2: Architecture and data flow for the web application

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Gathering Problems Faced by Users through Empathy Mapping	USN-1	For designing a better website.	4	High	V.AKHIL M.Divya Sree A.Bharath Chandra Y.Devi Prasanna
Sprint-1	Collecting different Ideas	USN-2	Converging the divergent Ideas	4	High	V.AKHIL M.Divya Sree A.Bharath Chandra Y.Devi Prasanna
Sprint-1	Proposed Solution and Architecture for model	USN-3	For better service of the website	6	Medium	V.AKHIL M.Divya Sree A.Bharath Chandra
Sprint-1	Technical architecture and Some planning for work	USN-4	Data Flow diagram	4	Medium	M.Divya Sree A.Bharath Chandra Y.Devi Prasanna

Sprint-2	Tools and Project planning	USN-5	This is the most important step to choose the technologies and format.	12	High	V.AKHIL M.Divya Sree
Sprint-3	Training the model	USN-6	Checking various hyper tuning parameters to fit the best accuracy model.	8	High	V.AKHIL
Sprint-3	Building website	USN-7	User Interface	8	High	V.AKHIL M.Divya Sree
Sprint-3	Debugging	USN-8	Rectifying error by deploying it in localhost.	4	Low	V.AKHIL
Sprint-4	Project Documentation	USN-9	Documentation for coding.	20	Medium	V.AKHIL
Sprint-5	Solution Performance	USN-10	Model Metrics	10	High	V.AKHIL
Sprint-5	Project Report	USN-11	Report for the project.	10	High	Y.Devi Prasanna M.Divya Sree V.AKHIL A.Bharath Chandra
Sprint-5	Demonstration	USN-12	Exhibiting project to the user	10	High	V.AKHIL M.Divya Sree A.Bharath Chandra Y.Devi Prasanna

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	6 Days	27 Oct 2023	1 Nov 2023	18	1 Nov 2023
Sprint-2	12	6 Days	4 Nov 2023	9 Nov 2022	8	10 Nov 2023
Sprint-3	20	10 Days	07 Nov 2022	16 Nov 2022	20	16 Nov 2022
Sprint-4	25	2 Days	16 Nov 2022	17 Nov 2022	25	17 Nov 2022
Sprint-5	25	5 Days	18 Nov 2022	22 Nov 2022	25	21 Nov 2022

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Image Upload: Allow users to upload images of the dogs to the website.

```
// Upload Preview
function readURL(input) {
  if (input.files && input.files[0]) {
    var reader = new FileReader();
    reader.onload = function (e) {
      $("#img").css(
        "background-image",
        "url(" + e.target.result + ")"
      );
      $("#img").hide();
      $("#img").fadeIn(650);
    };
    reader.readAsDataURL(input.files[0]);
  }
}
$("#fil").change(function () {
  $(".image-section").show();
  $("#sub").show();
  $("#result").text("");
  $("#result").hide();
  $("#breed").text("");
  $("#breed").hide();
  $("#feat").text("");
  $("#feat").hide();
  $("#features").text("");
  $("#features").hide();
  readURL(this);
});
```

SMARTBRIDGE

7.2 Feature 2

Model Interface: Process the uploaded images through the pre-trained dog breed identification model to predict the breed.

```python
model = load_model("model_final2.h5",compile=False)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
        print("current path")
        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)
        image = read_image(filepath, 224)
        image = np.expand_dims(image, axis=0)
        pred = model.predict(image)[0]
        label_idx = np.argmax(pred)
        breed_name = id2breed[label_idx]
        print(breed_name)
    return breed_name
if __name__ == '__main__':
    app.run(debug = True)
```

8. PERFORMANCE TESTING

8.1 Performance Metrics

```
____________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
====================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3)]  0         []

 Conv1 (Conv2D)                 (None, 112, 112, 32)  864         ['input_1[0][0]']

 bn_Conv1 (BatchNormalizati     (None, 112, 112, 32)  128         ['Conv1[0][0]']
 on)

 Conv1_relu (ReLU)              (None, 112, 112, 32)  0           ['bn_Conv1[0][0]']

 expanded_conv_depthwise (D     (None, 112, 112, 32)  288         ['Conv1_relu[0][0]']
 epthwiseConv2D)

 expanded_conv_depthwise_BN     (None, 112, 112, 32)  128         ['expanded_conv_depthwise[0][0
  (BatchNormalization)                                            ]']

 expanded_conv_depthwise_re     (None, 112, 112, 32)  0           ['expanded_conv_depthwise_BN[0
 lu (ReLU)                                                        ][0]']

 expanded_conv_project (Con     (None, 112, 112, 16)  512         ['expanded_conv_depthwise_relu
 v2D)                                                             [0][0]']

 expanded_conv_project_BN (     (None, 112, 112, 16)  64          ['expanded_conv_project[0][0]'
 BatchNormalization)                                              ]

 block_1_expand (Conv2D)        (None, 112, 112, 96)  1536        ['expanded_conv_project_BN[0][
                                                                  0]']

 block_1_expand_BN (BatchNo     (None, 112, 112, 96)  384         ['block_1_expand[0][0]']
 rmalization)

 block_1_expand_relu (ReLU)     (None, 112, 112, 96)  0           ['block_1_expand_BN[0][0]']

 block_1_pad (ZeroPadding2D     (None, 113, 113, 96)  0           ['block_1_expand_relu[0][0]']
 )

 block_1_depthwise (Depthwi     (None, 56, 56, 96)   864          ['block_1_pad[0][0]']
 seConv2D)
```

```
block_1_depthwise_BN (Batc  (None, 56, 56, 96)          384       ['block_1_depthwise[0][0]']
hNormalization)

block_1_depthwise_relu (Re  (None, 56, 56, 96)          0         ['block_1_depthwise_BN[0][0]']
LU)

block_1_project (Conv2D)    (None, 56, 56, 24)          2304      ['block_1_depthwise_relu[0][0]
                                                                  ']

block_1_project_BN (BatchN  (None, 56, 56, 24)          96        ['block_1_project[0][0]']
ormalization)

block_2_expand (Conv2D)     (None, 56, 56, 144)         3456      ['block_1_project_BN[0][0]']

block_2_expand_BN (BatchNo  (None, 56, 56, 144)         576       ['block_2_expand[0][0]']
rmalization)

block_2_expand_relu (ReLU)  (None, 56, 56, 144)         0         ['block_2_expand_BN[0][0]']

block_2_depthwise (Depthwi  (None, 56, 56, 144)         1296      ['block_2_expand_relu[0][0]']
seConv2D)

block_2_depthwise_BN (Batc  (None, 56, 56, 144)         576       ['block_2_depthwise[0][0]']
hNormalization)

block_2_depthwise_relu (Re  (None, 56, 56, 144)         0         ['block_2_depthwise_BN[0][0]']
LU)

block_2_project (Conv2D)    (None, 56, 56, 24)          3456      ['block_2_depthwise_relu[0][0]
                                                                  ']

block_2_project_BN (BatchN  (None, 56, 56, 24)          96        ['block_2_project[0][0]']
ormalization)

block_2_add (Add)           (None, 56, 56, 24)          0         ['block_1_project_BN[0][0]',
                                                                   'block_2_project_BN[0][0]']

block_3_expand (Conv2D)     (None, 56, 56, 144)         3456      ['block_2_add[0][0]']

block_3_expand_BN (BatchNo  (None, 56, 56, 144)         576       ['block_3_expand[0][0]']
rmalization)

block_3_expand_relu (ReLU)  (None, 56, 56, 144)         0         ['block_3_expand_BN[0][0]']
```

```
block_3_pad (ZeroPadding2D  (None, 57, 57, 144)       0          ['block_3_expand_relu[0][0]']
)

block_3_depthwise (Depthwi  (None, 28, 28, 144)       1296       ['block_3_pad[0][0]']
seConv2D)

block_3_depthwise_BN (Batc  (None, 28, 28, 144)       576        ['block_3_depthwise[0][0]']
hNormalization)

block_3_depthwise_relu (Re  (None, 28, 28, 144)       0          ['block_3_depthwise_BN[0][0]']
LU)

block_3_project (Conv2D)    (None, 28, 28, 32)        4608       ['block_3_depthwise_relu[0][0]
                                                                  ']

block_3_project_BN (BatchN  (None, 28, 28, 32)        128        ['block_3_project[0][0]']
ormalization)

block_4_expand (Conv2D)     (None, 28, 28, 192)       6144       ['block_3_project_BN[0][0]']

block_4_expand_BN (BatchNo  (None, 28, 28, 192)       768        ['block_4_expand[0][0]']
rmalization)

block_4_expand_relu (ReLU)  (None, 28, 28, 192)       0          ['block_4_expand_BN[0][0]']

block_4_depthwise (Depthwi  (None, 28, 28, 192)       1728       ['block_4_expand_relu[0][0]']
seConv2D)

block_4_depthwise_BN (Batc  (None, 28, 28, 192)       768        ['block_4_depthwise[0][0]']
hNormalization)

block_4_depthwise_relu (Re  (None, 28, 28, 192)       0          ['block_4_depthwise_BN[0][0]']
LU)

block_4_project (Conv2D)    (None, 28, 28, 32)        6144       ['block_4_depthwise_relu[0][0]
                                                                  ']

block_4_project_BN (BatchN  (None, 28, 28, 32)        128        ['block_4_project[0][0]']
ormalization)

block_4_add (Add)           (None, 28, 28, 32)        0          ['block_3_project_BN[0][0]',
                                                                   'block_4_project_BN[0][0]']

block_5_expand (Conv2D)     (None, 28, 28, 192)       6144       ['block_4_add[0][0]']

block_5_expand_BN (BatchNo  (None, 28, 28, 192)       768        ['block_5_expand[0][0]']
rmalization)
```

```
 block_5_expand_relu (ReLU)  (None, 28, 28, 192)         0           ['block_5_expand_BN[0][0]']

 block_5_depthwise (Depthwi  (None, 28, 28, 192)         1728        ['block_5_expand_relu[0][0]']
 seConv2D)

 block_5_depthwise_BN (Batc  (None, 28, 28, 192)         768         ['block_5_depthwise[0][0]']
 hNormalization)

 block_5_depthwise_relu (Re  (None, 28, 28, 192)         0           ['block_5_depthwise_BN[0][0]']
 LU)

 block_5_project (Conv2D)    (None, 28, 28, 32)          6144        ['block_5_depthwise_relu[0][0]
                                                                     ']

 block_5_project_BN (BatchN  (None, 28, 28, 32)          128         ['block_5_project[0][0]']
 ormalization)

 block_5_add (Add)           (None, 28, 28, 32)          0           ['block_4_add[0][0]',
                                                                      'block_5_project_BN[0][0]']

 block_6_expand (Conv2D)     (None, 28, 28, 192)         6144        ['block_5_add[0][0]']

 block_6_expand_BN (BatchNo  (None, 28, 28, 192)         768         ['block_6_expand[0][0]']
 rmalization)

 block_6_expand_relu (ReLU)  (None, 28, 28, 192)         0           ['block_6_expand_BN[0][0]']

 block_6_pad (ZeroPadding2D  (None, 29, 29, 192)         0           ['block_6_expand_relu[0][0]']
 )

 block_6_depthwise (Depthwi  (None, 14, 14, 192)         1728        ['block_6_pad[0][0]']
 seConv2D)

 block_6_depthwise_BN (Batc  (None, 14, 14, 192)         768         ['block_6_depthwise[0][0]']
 hNormalization)

 block_6_depthwise_relu (Re  (None, 14, 14, 192)         0           ['block_6_depthwise_BN[0][0]']
 LU)

 block_6_project (Conv2D)    (None, 14, 14, 64)          12288       ['block_6_depthwise_relu[0][0]
                                                                     ']

 block_6_project_BN (BatchN  (None, 14, 14, 64)          256         ['block_6_project[0][0]']
 ormalization)
```

block_7_expand (Conv2D)	(None, 14, 14, 384)	24576	['block_6_project_BN[0][0]']
block_7_expand_BN (BatchNormalization)	(None, 14, 14, 384)	1536	['block_7_expand[0][0]']
block_7_expand_relu (ReLU)	(None, 14, 14, 384)	0	['block_7_expand_BN[0][0]']
block_7_depthwise (DepthwiseConv2D)	(None, 14, 14, 384)	3456	['block_7_expand_relu[0][0]']
block_7_depthwise_BN (BatchNormalization)	(None, 14, 14, 384)	1536	['block_7_depthwise[0][0]']
block_7_depthwise_relu (ReLU)	(None, 14, 14, 384)	0	['block_7_depthwise_BN[0][0]']
block_7_project (Conv2D)	(None, 14, 14, 64)	24576	['block_7_depthwise_relu[0][0]']
block_7_project_BN (BatchNormalization)	(None, 14, 14, 64)	256	['block_7_project[0][0]']
block_7_add (Add)	(None, 14, 14, 64)	0	['block_6_project_BN[0][0]', 'block_7_project_BN[0][0]']
block_8_expand (Conv2D)	(None, 14, 14, 384)	24576	['block_7_add[0][0]']
block_8_expand_BN (BatchNormalization)	(None, 14, 14, 384)	1536	['block_8_expand[0][0]']
block_8_expand_relu (ReLU)	(None, 14, 14, 384)	0	['block_8_expand_BN[0][0]']
block_8_depthwise (DepthwiseConv2D)	(None, 14, 14, 384)	3456	['block_8_expand_relu[0][0]']
block_8_depthwise_BN (BatchNormalization)	(None, 14, 14, 384)	1536	['block_8_depthwise[0][0]']
block_8_depthwise_relu (ReLU)	(None, 14, 14, 384)	0	['block_8_depthwise_BN[0][0]']
block_8_project (Conv2D)	(None, 14, 14, 64)	24576	['block_8_depthwise_relu[0][0]']
block_8_project_BN (BatchNormalization)	(None, 14, 14, 64)	256	['block_8_project[0][0]']

```
 block_8_add (Add)           (None, 14, 14, 64)           0         ['block_7_add[0][0]',
                                                                     'block_8_project_BN[0][0]']

 block_9_expand (Conv2D)     (None, 14, 14, 384)          24576     ['block_8_add[0][0]']

 block_9_expand_BN (BatchNo  (None, 14, 14, 384)          1536      ['block_9_expand[0][0]']
 rmalization)

 block_9_expand_relu (ReLU)  (None, 14, 14, 384)          0         ['block_9_expand_BN[0][0]']

 block_9_depthwise (Depthwi  (None, 14, 14, 384)          3456      ['block_9_expand_relu[0][0]']
 seConv2D)

 block_9_depthwise_BN (Batc  (None, 14, 14, 384)          1536      ['block_9_depthwise[0][0]']
 hNormalization)

 block_9_depthwise_relu (Re  (None, 14, 14, 384)          0         ['block_9_depthwise_BN[0][0]']
 LU)

 block_9_project (Conv2D)    (None, 14, 14, 64)           24576     ['block_9_depthwise_relu[0][0]
                                                                    ']

 block_9_project_BN (BatchN  (None, 14, 14, 64)           256       ['block_9_project[0][0]']
 ormalization)

 block_9_add (Add)           (None, 14, 14, 64)           0         ['block_8_add[0][0]',
                                                                     'block_9_project_BN[0][0]']

 block_10_expand (Conv2D)    (None, 14, 14, 384)          24576     ['block_9_add[0][0]']

 block_10_expand_BN (BatchN  (None, 14, 14, 384)          1536      ['block_10_expand[0][0]']
 ormalization)

 block_10_expand_relu (ReLU  (None, 14, 14, 384)          0         ['block_10_expand_BN[0][0]']
 )

 block_10_depthwise (Depthw  (None, 14, 14, 384)          3456      ['block_10_expand_relu[0][0]']
 iseConv2D)

 block_10_depthwise_BN (Bat  (None, 14, 14, 384)          1536      ['block_10_depthwise[0][0]']
 chNormalization)

 block_10_depthwise_relu (R  (None, 14, 14, 384)          0         ['block_10_depthwise_BN[0][0]'
 eLU)                                                               ]

 block_10_project (Conv2D)   (None, 14, 14, 96)           36864     ['block_10_depthwise_relu[0][0
```

```
                                                                 - -

 block_10_project_BN (Batch  (None, 14, 14, 96)          384       ['block_10_project[0][0]']
 Normalization)

 block_11_expand (Conv2D)    (None, 14, 14, 576)          55296     ['block_10_project_BN[0][0]']

 block_11_expand_BN (BatchN  (None, 14, 14, 576)          2304      ['block_11_expand[0][0]']
 ormalization)

 block_11_expand_relu (ReLU  (None, 14, 14, 576)          0         ['block_11_expand_BN[0][0]']
 )

 block_11_depthwise (Depthw  (None, 14, 14, 576)          5184      ['block_11_expand_relu[0][0]']
 iseConv2D)

 block_11_depthwise_BN (Bat  (None, 14, 14, 576)          2304      ['block_11_depthwise[0][0]']
 chNormalization)

 block_11_depthwise_relu (R  (None, 14, 14, 576)          0         ['block_11_depthwise_BN[0][0]'
 eLU)                                                                ]

 block_11_project (Conv2D)   (None, 14, 14, 96)           55296     ['block_11_depthwise_relu[0][0
                                                                     ]']

 block_11_project_BN (Batch  (None, 14, 14, 96)           384       ['block_11_project[0][0]']
 Normalization)

 block_11_add (Add)          (None, 14, 14, 96)           0         ['block_10_project_BN[0][0]',
                                                                      'block_11_project_BN[0][0]']

 block_12_expand (Conv2D)    (None, 14, 14, 576)          55296     ['block_11_add[0][0]']

 block_12_expand_BN (BatchN  (None, 14, 14, 576)          2304      ['block_12_expand[0][0]']
 ormalization)

 block_12_expand_relu (ReLU  (None, 14, 14, 576)          0         ['block_12_expand_BN[0][0]']
 )

 block_12_depthwise (Depthw  (None, 14, 14, 576)          5184      ['block_12_expand_relu[0][0]']
 iseConv2D)

 block_12_depthwise_BN (Bat  (None, 14, 14, 576)          2304      ['block_12_depthwise[0][0]']
 chNormalization)
```

SMARTBRIDGE

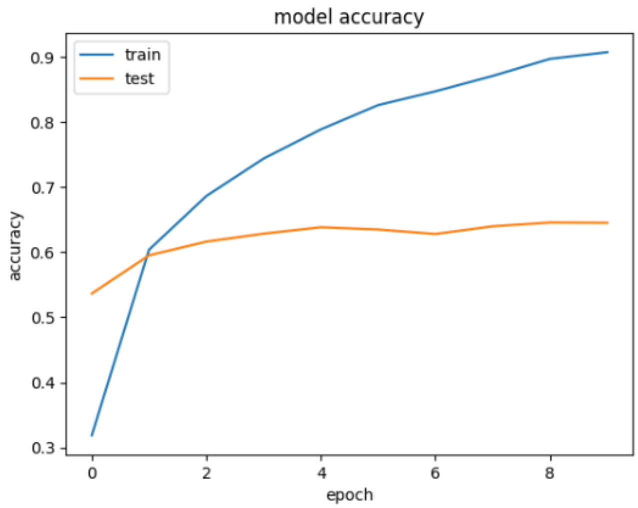
```
 block_12_depthwise_relu (R  (None, 14, 14, 576)         0         ['block_12_depthwise_BN[0][0]'
 eLU)                                                               ]

 block_12_project (Conv2D)   (None, 14, 14, 96)          55296     ['block_12_depthwise_relu[0][0
                                                                    ]']

 block_12_project_BN (Batch  (None, 14, 14, 96)          384       ['block_12_project[0][0]']
 Normalization)

 block_12_add (Add)          (None, 14, 14, 96)          0         ['block_11_add[0][0]',
                                                                     'block_12_project_BN[0][0]']

 block_13_expand (Conv2D)    (None, 14, 14, 576)         55296     ['block_12_add[0][0]']

 block_13_expand_BN (BatchN  (None, 14, 14, 576)         2304      ['block_13_expand[0][0]']
 ormalization)

 block_13_expand_relu (ReLU  (None, 14, 14, 576)         0         ['block_13_expand_BN[0][0]']
 )

 block_13_pad (ZeroPadding2  (None, 15, 15, 576)         0         ['block_13_expand_relu[0][0]']
 D)

 block_13_depthwise (Depthw  (None, 7, 7, 576)           5184      ['block_13_pad[0][0]']
 iseConv2D)

 block_13_depthwise_BN (Bat  (None, 7, 7, 576)           2304      ['block_13_depthwise[0][0]']
 chNormalization)

 block_13_depthwise_relu (R  (None, 7, 7, 576)           0         ['block_13_depthwise_BN[0][0]'
 eLU)                                                               ]

 block_13_project (Conv2D)   (None, 7, 7, 160)           92160     ['block_13_depthwise_relu[0][0
                                                                    ]']

 block_13_project_BN (Batch  (None, 7, 7, 160)           640       ['block_13_project[0][0]']
 Normalization)

 block_14_expand (Conv2D)    (None, 7, 7, 960)           153600    ['block_13_project_BN[0][0]']

 block_14_expand_BN (BatchN  (None, 7, 7, 960)           3840      ['block_14_expand[0][0]']
 ormalization)

 block_14_expand_relu (ReLU  (None, 7, 7, 960)           0         ['block_14_expand_BN[0][0]']
 )
```

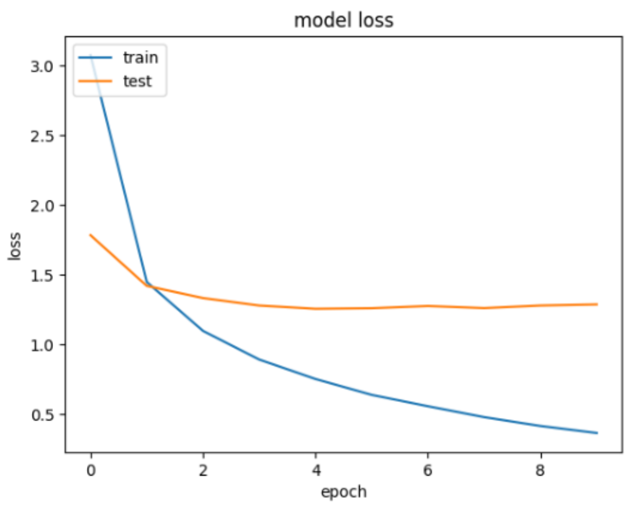
```
 block_14_depthwise (Depthw  (None, 7, 7, 960)           8640      ['block_14_expand_relu[0][0]']
 iseConv2D)

 block_14_depthwise_BN (Bat  (None, 7, 7, 960)           3840      ['block_14_depthwise[0][0]']
 chNormalization)

 block_14_depthwise_relu (R  (None, 7, 7, 960)           0         ['block_14_depthwise_BN[0][0]'
 eLU)                                                              ]

 block_14_project (Conv2D)   (None, 7, 7, 160)           153600    ['block_14_depthwise_relu[0][0
                                                                   ]']

 block_14_project_BN (Batch  (None, 7, 7, 160)           640       ['block_14_project[0][0]']
 Normalization)

 block_14_add (Add)          (None, 7, 7, 160)           0         ['block_13_project_BN[0][0]',
                                                                    'block_14_project_BN[0][0]']

 block_15_expand (Conv2D)    (None, 7, 7, 960)           153600    ['block_14_add[0][0]']

 block_15_expand_BN (BatchN  (None, 7, 7, 960)           3840      ['block_15_expand[0][0]']
 ormalization)

 block_15_expand_relu (ReLU  (None, 7, 7, 960)           0         ['block_15_expand_BN[0][0]']
 )

 block_15_depthwise (Depthw  (None, 7, 7, 960)           8640      ['block_15_expand_relu[0][0]']
 iseConv2D)

 block_15_depthwise_BN (Bat  (None, 7, 7, 960)           3840      ['block_15_depthwise[0][0]']
 chNormalization)

 block_15_depthwise_relu (R  (None, 7, 7, 960)           0         ['block_15_depthwise_BN[0][0]'
 eLU)                                                              ]

 block_15_project (Conv2D)   (None, 7, 7, 160)           153600    ['block_15_depthwise_relu[0][0
                                                                   ]']

 block_15_project_BN (Batch  (None, 7, 7, 160)           640       ['block_15_project[0][0]']
 Normalization)

 block_15_add (Add)          (None, 7, 7, 160)           0         ['block_14_add[0][0]',
                                                                    'block_15_project_BN[0][0]']

 block_16_expand (Conv2D)    (None, 7, 7, 960)           153600    ['block_15_add[0][0]']
```

```
 block_14_depthwise (Depthw  (None, 7, 7, 960)            8640      ['block_14_expand_relu[0][0]']
 iseConv2D)

 block_14_depthwise_BN (Bat  (None, 7, 7, 960)            3840      ['block_14_depthwise[0][0]']
 chNormalization)

 block_14_depthwise_relu (R  (None, 7, 7, 960)            0         ['block_14_depthwise_BN[0][0]'
 eLU)                                                                ]

 block_14_project (Conv2D)   (None, 7, 7, 160)            153600    ['block_14_depthwise_relu[0][0
                                                                    ]']

 block_14_project_BN (Batch  (None, 7, 7, 160)            640       ['block_14_project[0][0]']
 Normalization)

 block_14_add (Add)          (None, 7, 7, 160)            0         ['block_13_project_BN[0][0]',
                                                                     'block_14_project_BN[0][0]']

 block_15_expand (Conv2D)    (None, 7, 7, 960)            153600    ['block_14_add[0][0]']

 block_15_expand_BN (BatchN  (None, 7, 7, 960)            3840      ['block_15_expand[0][0]']
 ormalization)

 block_15_expand_relu (ReLU  (None, 7, 7, 960)            0         ['block_15_expand_BN[0][0]']
 )

 block_15_depthwise (Depthw  (None, 7, 7, 960)            8640      ['block_15_expand_relu[0][0]']
 iseConv2D)

 block_15_depthwise_BN (Bat  (None, 7, 7, 960)            3840      ['block_15_depthwise[0][0]']
 chNormalization)

 block_15_depthwise_relu (R  (None, 7, 7, 960)            0         ['block_15_depthwise_BN[0][0]'
 eLU)                                                                ]

 block_15_project (Conv2D)   (None, 7, 7, 160)            153600    ['block_15_depthwise_relu[0][0
                                                                    ]']

 block_15_project_BN (Batch  (None, 7, 7, 160)            640       ['block_15_project[0][0]']
 Normalization)

 block_15_add (Add)          (None, 7, 7, 160)            0         ['block_14_add[0][0]',
                                                                     'block_15_project_BN[0][0]']

 block_16_expand (Conv2D)    (None, 7, 7, 960)            153600    ['block_15_add[0][0]']

 block_16_expand_BN (BatchN  (None, 7, 7, 960)            3840      ['block_16_expand[0][0]']
 ormalization)

 block_16_expand_relu (ReLU  (None, 7, 7, 960)            0         ['block_16_expand_BN[0][0]']
 )

 block_16_depthwise (Depthw  (None, 7, 7, 960)            8640      ['block_16_expand_relu[0][0]']
 iseConv2D)

 block_16_depthwise_BN (Bat  (None, 7, 7, 960)            3840      ['block_16_depthwise[0][0]']
 chNormalization)

 block_16_depthwise_relu (R  (None, 7, 7, 960)            0         ['block_16_depthwise_BN[0][0]'
 eLU)                                                                ]

 block_16_project (Conv2D)   (None, 7, 7, 320)            307200    ['block_16_depthwise_relu[0][0
                                                                    ]']

 block_16_project_BN (Batch  (None, 7, 7, 320)            1280      ['block_16_project[0][0]']
 Normalization)

 Conv_1 (Conv2D)             (None, 7, 7, 1280)           409600    ['block_16_project_BN[0][0]']

 Conv_1_bn (BatchNormalizat  (None, 7, 7, 1280)           5120      ['Conv_1[0][0]']
 ion)

 out_relu (ReLU)             (None, 7, 7, 1280)           0         ['Conv_1_bn[0][0]']

 global_average_pooling2d (  (None, 1280)                 0         ['out_relu[0][0]']
 GlobalAveragePooling2D)

 dropout (Dropout)           (None, 1280)                 0         ['global_average_pooling2d[0][
                                                                    0]']

 dense (Dense)               (None, 1024)                 1311744   ['dropout[0][0]']

 dense_1 (Dense)             (None, 120)                  123000    ['dense[0][0]']

==================================================================================================
Total params: 3692728 (14.09 MB)
Trainable params: 1434744 (5.47 MB)
Non-trainable params: 2257984 (8.61 MB)
__________________________________________________________________________________________________
```

Accuracy and Loss:

```
Epoch 1/10
512/512 [==============================] - ETA: 0s - loss: 3.0769 - acc: 0.3188
Epoch 1: val_loss improved from inf to 1.78527, saving model to model_final2.h5
/opt/conda/lib/python3.10/site-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 f
ile via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `mode
l.save('my_model.keras')`.
  saving_api.save_model(
512/512 [==============================] - 140s 246ms/step - loss: 3.0769 - acc: 0.3188 - val_loss: 1.7853 - val_acc: 0.5364 -
lr: 1.0000e-04
Epoch 2/10
511/512 [============================>.] - ETA: 0s - loss: 1.4500 - acc: 0.6036
Epoch 2: val_loss improved from 1.78527 to 1.42178, saving model to model_final2.h5
512/512 [==============================] - 59s 115ms/step - loss: 1.4498 - acc: 0.6036 - val_loss: 1.4218 - val_acc: 0.5951 -
lr: 1.0000e-04
Epoch 3/10
511/512 [============================>.] - ETA: 0s - loss: 1.0990 - acc: 0.6860
Epoch 3: val_loss improved from 1.42178 to 1.33423, saving model to model_final2.h5
512/512 [==============================] - 61s 119ms/step - loss: 1.0989 - acc: 0.6861 - val_loss: 1.3342 - val_acc: 0.6161 -
lr: 1.0000e-04
Epoch 4/10
511/512 [============================>.] - ETA: 0s - loss: 0.8934 - acc: 0.7435
Epoch 4: val_loss improved from 1.33423 to 1.28071, saving model to model_final2.h5
512/512 [==============================] - 58s 113ms/step - loss: 0.8933 - acc: 0.7435 - val_loss: 1.2807 - val_acc: 0.6284 -
lr: 1.0000e-04
Epoch 5/10
511/512 [============================>.] - ETA: 0s - loss: 0.7540 - acc: 0.7883
Epoch 5: val_loss improved from 1.28071 to 1.25714, saving model to model_final2.h5
512/512 [==============================] - 58s 114ms/step - loss: 0.7539 - acc: 0.7883 - val_loss: 1.2571 - val_acc: 0.6381 -
lr: 1.0000e-04
Epoch 6/10
511/512 [============================>.] - ETA: 0s - loss: 0.6400 - acc: 0.8256
Epoch 6: val_loss did not improve from 1.25714
512/512 [==============================] - 57s 112ms/step - loss: 0.6400 - acc: 0.8256 - val_loss: 1.2611 - val_acc: 0.6347 -
lr: 1.0000e-04
Epoch 7/10
511/512 [============================>.] - ETA: 0s - loss: 0.5584 - acc: 0.8467
Epoch 7: val_loss did not improve from 1.25714
512/512 [==============================] - 54s 107ms/step - loss: 0.5583 - acc: 0.8468 - val_loss: 1.2773 - val_acc: 0.6279 -
lr: 1.0000e-04
Epoch 8/10
511/512 [============================>.] - ETA: 0s - loss: 0.4802 - acc: 0.8704
Epoch 8: val_loss did not improve from 1.25714
512/512 [==============================] - 56s 110ms/step - loss: 0.4801 - acc: 0.8704 - val_loss: 1.2621 - val_acc: 0.6396 -
lr: 1.0000e-04
Epoch 9/10
511/512 [============================>.] - ETA: 0s - loss: 0.4168 - acc: 0.8965
Epoch 9: val_loss did not improve from 1.25714
512/512 [==============================] - 41s 79ms/step - loss: 0.4168 - acc: 0.8965 - val_loss: 1.2808 - val_acc: 0.6455 - l
r: 1.0000e-04
Epoch 10/10
511/512 [============================>.] - ETA: 0s - loss: 0.3665 - acc: 0.9068
Epoch 10: val_loss did not improve from 1.25714
512/512 [==============================] - 48s 94ms/step - loss: 0.3665 - acc: 0.9068 - val_loss: 1.2886 - val_acc: 0.6450 - l
r: 1.0000e-04
```


model accuracy
train
test
accuracy
epoch


model loss
train
test
loss
epoch

9. RESULTS

9.1 Output Screenshots

epoch

```
In [29]: I="/kaggle/input/dog-breed-identification1/Dog Breed Identification using Transfer Learning/test/0a8d8dda0e354c0571c8d4760

In [30]: id2breed = {i: name for i, name in enumerate(breed)}

In [31]: import PIL
         PIL.Image.open(I)

Out[31]:
```

```
In [32]: image = read_image(I, 224)
         image = np.expand_dims(image, axis=0)
         pred = model.predict(image)[0]
         label_idx = np.argmax(pred)
         breed_name = id2breed[label_idx]
         print(breed_name)

1/1 [==============================] - 1s 849ms/step
pug

In [33]: I1="/kaggle/input/dog-breed-identification1/Dog Breed Identification using Transfer Learning/test/00a3edd22dc7859c487a64777-

In [34]: PIL.Image.open(I1)

Out[34]:
```

```
In [35]: image = read_image(I1, 224)
         image = np.expand_dims(image, axis=0)
         pred = model.predict(image)[0]
         label_idx = np.argmax(pred)
         breed_name = id2breed[label_idx]
         print(breed_name)

1/1 [==============================] - 0s 24ms/step
australian_terrier
```

10. ADVANTAGES & DISADVANTAGES

There are various benefits to using transfer learning to identify dog breeds. It saves both time and computational resources by utilizing pre-trained models which have picked up features from massive datasets. By applying knowledge from one domain—general image recognition, for example—to another—specific dog breed identification—transfer learning makes it possible to achieve results that are frequently more accurate with less training data.

But there are also certain drawbacks. Inaccuracies or misclassifications may result from transfer learning's inability to accurately capture breed-specific traits, particularly for obscure or mixed breeds. Furthermore, the results may not precisely match the subtleties of different dog breeds due to the biases and limitations of the pre-trained model derived from the original dataset.

11. CONCLUSION

This model aims to accurately identify dog breeds from images by using a machine learning classification tool. Results from the application are dependable and accurate because it has been thoroughly tested using a variety of dog photos. For now the application offers each recognized dog breed's basic information, which is scraped from various sources. The learning mechanism in this model is based on Convolutional neural networks (CNNs), which have become very popular for image classification tasks. The CNN-based deep learning method is especially designed to predict dog breeds from input images. The model is built to produce outputs classifying many different kinds of dogs through transfer learning.

12. FUTURE SCOPE

The accuracy of identifying minute breed-specific characteristics may be improved by combining transfer learning with other strategies like ensemble approaches, fine-tuning, and improvements in neural network architectures. Additionally, developments in explainable AI may contribute to our understanding of how these models decide, which will boost their usability and trustworthiness in practical applications like veterinary care, pet care, and breeding. All things considered, transfer learning may lead to the development of increasingly precise, dependable, and broadly useful dog breed identification systems in the future.

13. APPENDIX

Source Code

GitHub & Project Demo Link

Demo Link: Click here for demo link

Git hub : Click here for git repo