Project Report Format

1. **INTRODUCTION**
   1.1 Project Overview
   1.2 Purpose
2. **LITERATURE SURVEY**
   2.1 Existing problem
   2.2 References
   2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   3.1 Empathy Map Canvas
   3.2 Ideation & Brainstorming
4. **REQUIREMENT ANALYSIS**
   4.1 Functional requirement
   4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   5.1 Data Flow Diagrams & User Stories
   5.2 Solution Architecture
6. **PROJECT PLANNING & SCHEDULING**
   6.1 Technical Architecture
   6.2 Sprint Planning & Estimation
   6.3 Sprint Delivery Schedule
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   7.1 Feature 1
   7.2 Feature 2
   7.3 Database Schema (if Applicable)
8. **PERFORMANCE TESTING**
   8.1 Performace Metrics
9. **RESULTS**
   9.1 Output Screenshots
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
   Source Code
   GitHub & Project Demo Link

PROJECT REPORT

1.INTRODUCTION

1.1 Project Overview

This project aims to develop a machine learning model that accurately identifies the breed of a dog from an image. Convolutional neural networks (CNNs) will be employed due to their effectiveness in image classification tasks. The dataset, comprising over 20,000 images of dogs from 120 distinct breeds, will be utilized. The model's performance will be evaluated using a test set of images, with an anticipated accuracy of over 90%. This project has the potential to impact animal shelters, pet owners, and researchers, and mitigation strategies have been identified to address potential risks. 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.

1.2 Purpose

Using transfer learning to identify dog breeds helps to comprehend breed diversity, 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 obstacles to for dog breed identification are:

- **Data Collection and Quality**: Acquiring a large and diverse dataset of dog images with accurate labels is difficult.
- **Data Preprocessing and Augmentation**: Properly preprocessing and augmenting the dataset to improve model generalization is crucial.
- **Model Selection and Optimization**: Choosing the right CNN architecture and hyperparameters is essential for high accuracy.
- **Overfitting and Underfitting**: Balancing training to avoid overfitting and underfitting is critical for good performance.
- **Data Bias and Fairness**: Addressing potential biases in the training data is essential for fairness.

2.2 References

The reference of this project are from smart internz website where this project is mentioned also many documentation were followed to complete this project

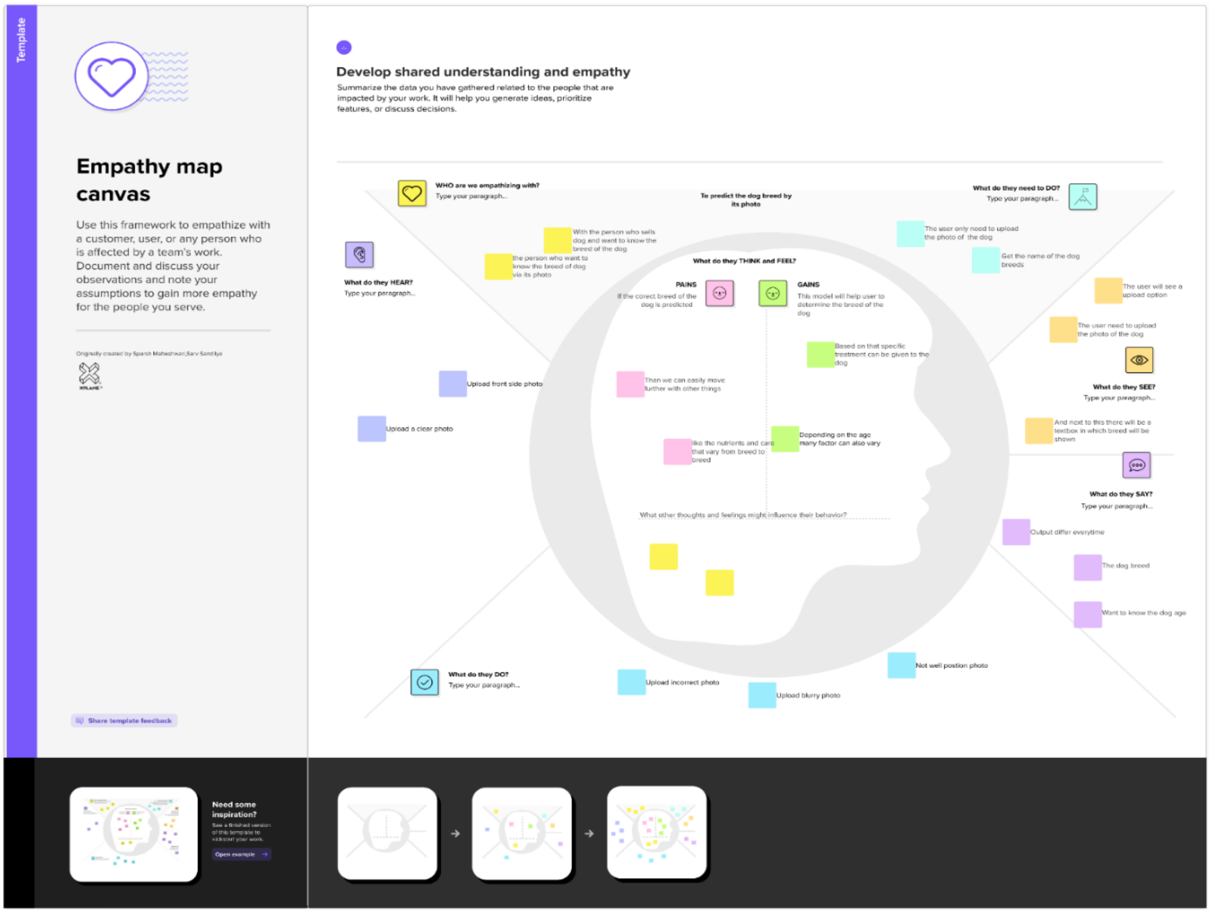
2.3 Problem Statement Definition

To engender user trust in the accuracy of dog breed predictions produced by a transfer learning-based model, we must address issues with breed misclassification, bias, and model generalization. This necessitates implementing robust assessment metrics, ensuring transparent model performance, and employing strategies to manage prediction uncertainty, thereby providing users with reliable and understandable information about the model's capabilities and limitations.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps teams better understand their users.



Template

Empathy map canvas

Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.

Originally created by Sparsh Maheshwari,Sarv Sandilya

XPLANE

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Develop shared understanding and empathy

Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.

WHO are we empathizing with?
Type your paragraph...

To predict the dog breed by its photo

With the person who sells dog and want to know the breed of the dog

the person who want to know the breed of dog via its photo

What do they HEAR?
Type your paragraph...

Upload front side photo

Upload a clear photo

What do they THINK and FEEL?

PAINS

If the corect breed of the dog is predicted

Than we can easily move further with other things

like the nutrients and care that vary from breed to breed

GAINS

This model will help user to determine the breed of the dog

Based on that specific treatment can be given to the dog

Depending on the age many factor can also vary

What other thoughts and feelings might influence their behavior?

What do they need to DO?
Type your paragraph...

The user only need to upload the photo of the dog

Get the name of the dog breeds

The user will see a upload option

The user need to upload the photo of the dog

What do they SEE?
Type your paragraph...

And next to this there will be a textbox in which breed will be shown

What do they SAY?
Type your paragraph...

Output differ everytime

The dog breed

Want to know the dog age

What do they DO?
Type your paragraph...

Upload incorrect photo

Upload blurry photo

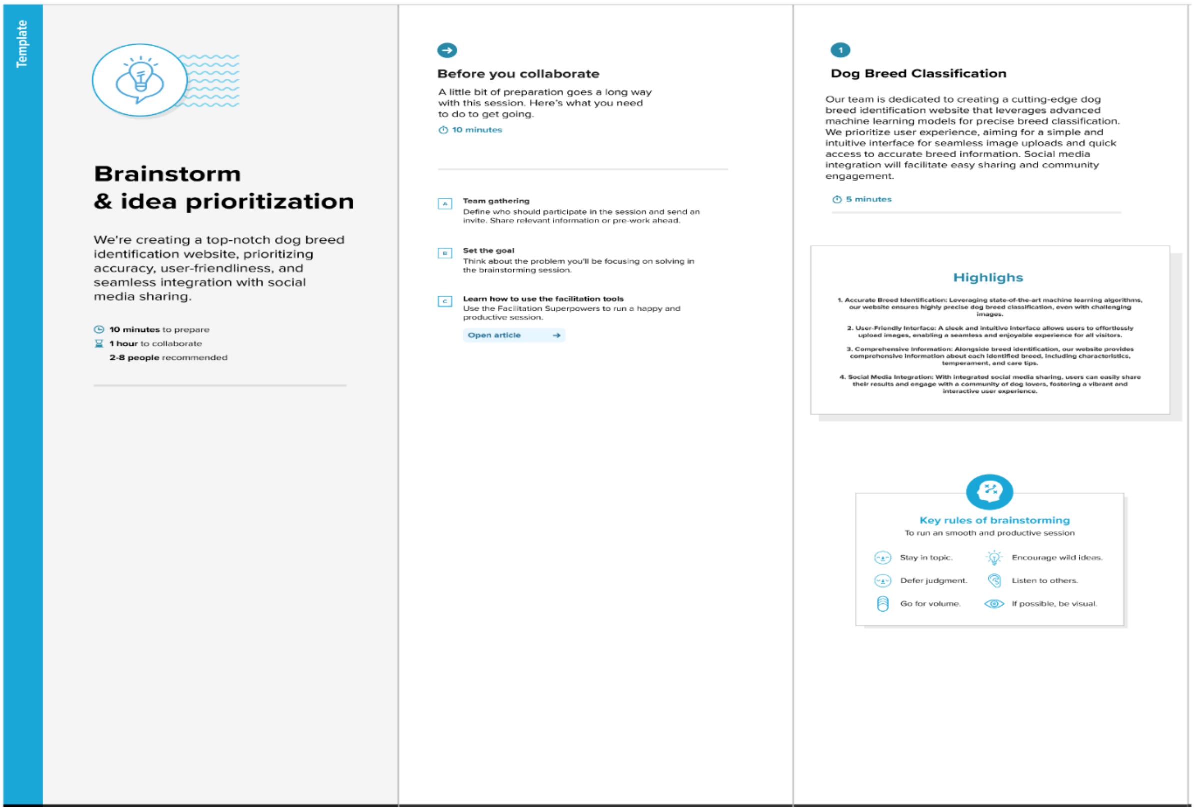
Not well postion photo

LINK:

https://app.mural.co/t/codingwaves2641/m/codingwaves2641/1698678026378/b33cb76c61b143210456ef53b27cfcd30200bcbb?sender=u633aaf89ece26fc5a47b6272

3.2 Ideation & Brainstorming

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions.



Template

Brainstorm & idea prioritization

We're creating a top-notch dog breed identification website, prioritizing accuracy, user-friendliness, and seamless integration with social media sharing.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Dog Breed Classification

Our team is dedicated to creating a cutting-edge dog breed identification website that leverages advanced machine learning models for precise breed classification. We prioritize user experience, aiming for a simple and intuitive interface for seamless image uploads and quick access to accurate breed information. Social media integration will facilitate easy sharing and community engagement.

5 minutes

Highlighs

1. Accurate Breed Identification: Leveraging state-of-the-art machine learning algorithms, our website ensures highly precise dog breed classification, even with challenging images.

2. User-Friendly Interface: A sleek and intuitive interface allows users to effortlessly upload images, enabling a seamless and enjoyable experience for all visitors.

3. Comprehensive Information: Alongside breed identification, our website provides comprehensive information about each identified breed, including characteristics, temperament, and care tips.

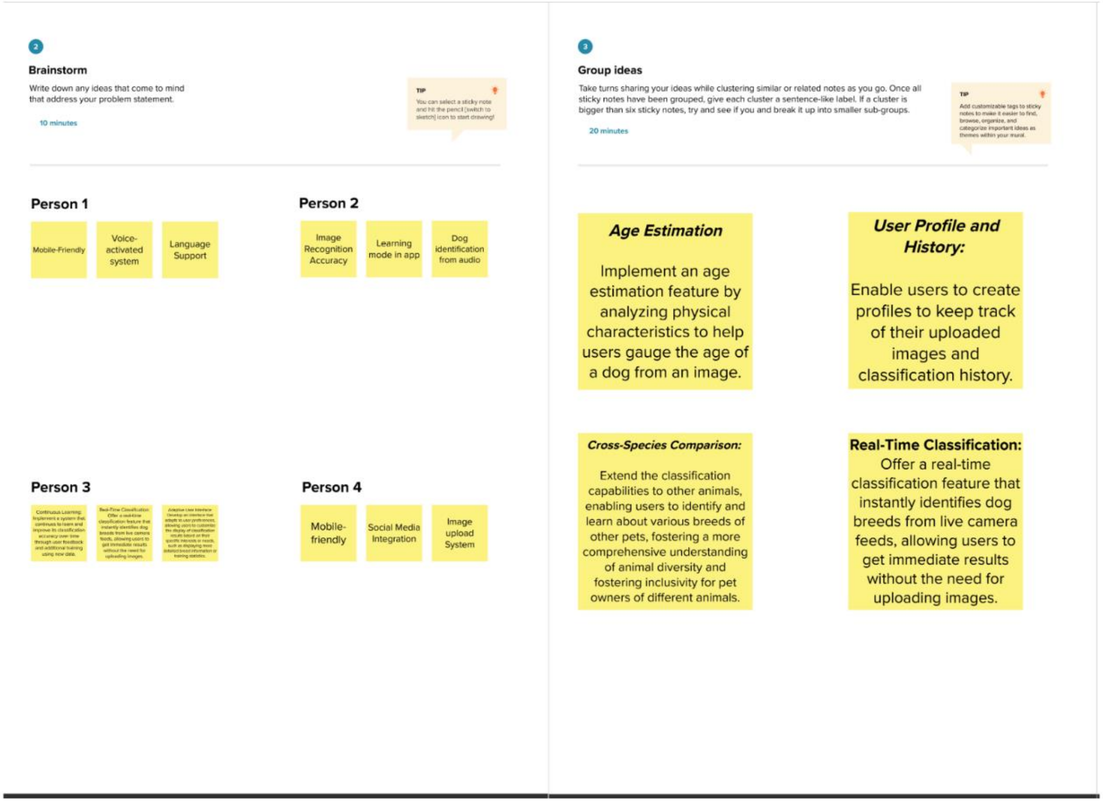
4. Social Media Integration: With integrated social media sharing, users can easily share their results and engage with a community of dog lovers, fostering a vibrant and interactive user experience.

Key rules of brainstorming

To run an smooth and productive session

Stay in topic. | Encourage wild ideas.
Defer judgment. | Listen to others.
Go for volume. | If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Mobile-Friendly
Voice-activated system
Language Support

Person 2

Image Recognition Accuracy
Learning mode in app
Dog identification from audio

Person 3

Continuous Learning: Implement a system that continues to learn and improve its classification accuracy over time through user feedback and additional training using new data.

Real-Time Classification: Offer a real-time classification feature that instantly identifies dog breeds from live camera feeds, allowing users to get immediate results without the need for uploading images.

Adaptive User Interface: Develop an interface that adapts to user preferences, allowing users to customize the display of classification results based on their specific interests or needs, such as displaying more detailed breed information or training statistics.

Person 4

Mobile-friendly
Social Media Integration
Image upload System

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Age Estimation

Implement an age estimation feature by analyzing physical characteristics to help users gauge the age of a dog from an image.

User Profile and History:

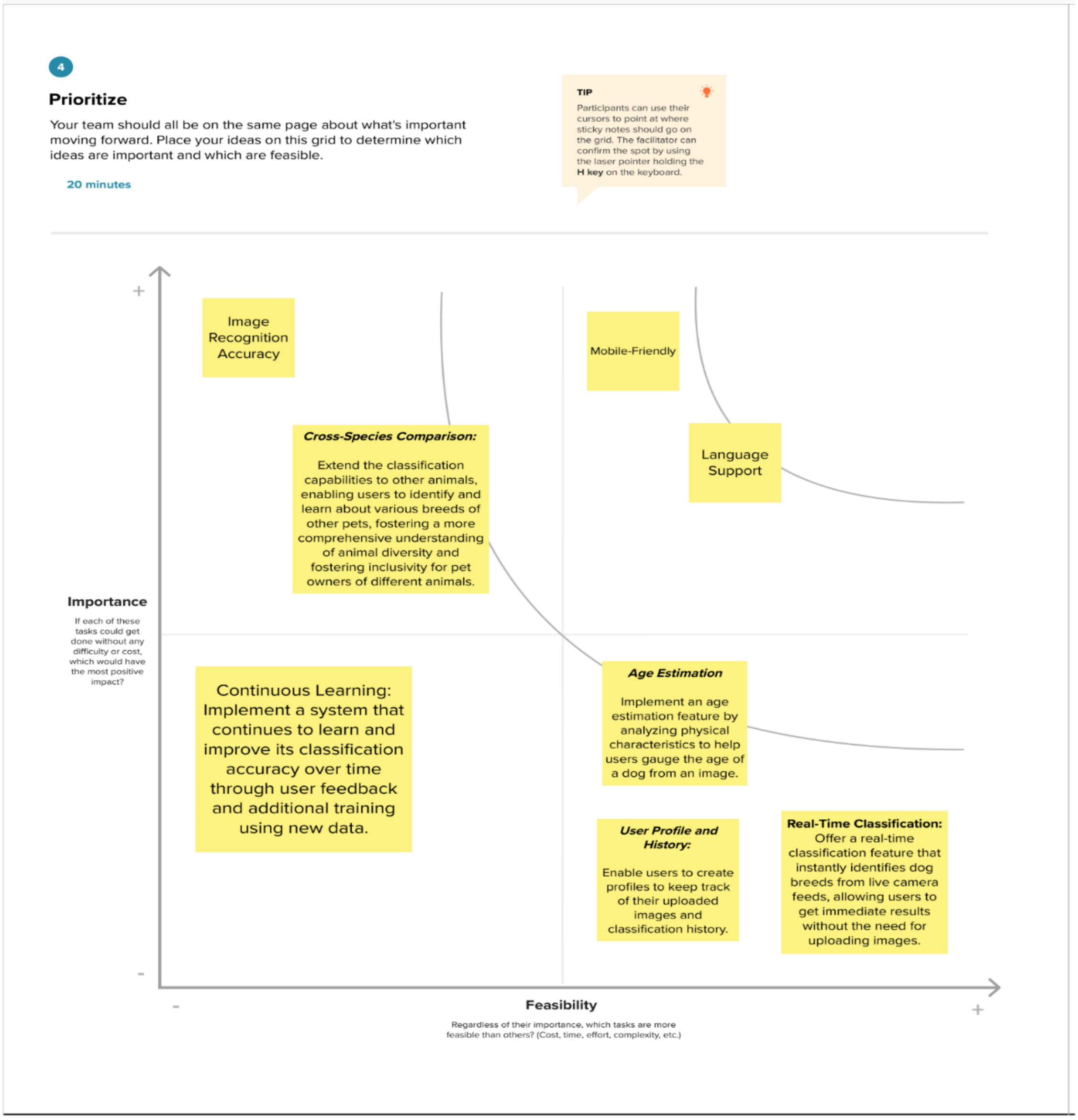
Enable users to create profiles to keep track of their uploaded images and classification history.

Cross-Species Comparison:

Extend the classification capabilities to other animals, enabling users to identify and learn about various breeds of other pets, fostering a more comprehensive understanding of animal diversity and fostering inclusivity for pet owners of different animals.

Real-Time Classification: Offer a real-time classification feature that instantly identifies dog breeds from live camera feeds, allowing users to get immediate results without the need for uploading images.

Step-3: Idea Prioritisation



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Image Recognition Accuracy

Cross-Species Comparison:

Extend the classification capabilities to other animals, enabling users to identify and learn about various breeds of other pets, fostering a more comprehensive understanding of animal diversity and fostering inclusivity for pet owners of different animals.

Mobile-Friendly

Language Support

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Continuous Learning: Implement a system that continues to learn and improve its classification accuracy over time through user feedback and additional training using new data.

Age Estimation

Implement an age estimation feature by analyzing physical characteristics to help users gauge the age of a dog from an image.

User Profile and History:

Enable users to create profiles to keep track of their uploaded images and classification history.

Real-Time Classification: Offer a real-time classification feature that instantly identifies dog breeds from live camera feeds, allowing users to get immediate results without the need for uploading images.

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

- Image Input
- Image Processing
- Model to predict breed
- Explain model decisions
- Visualize model predictions
- Maintenance
- Feedback

4.2 Non-Functional requirements

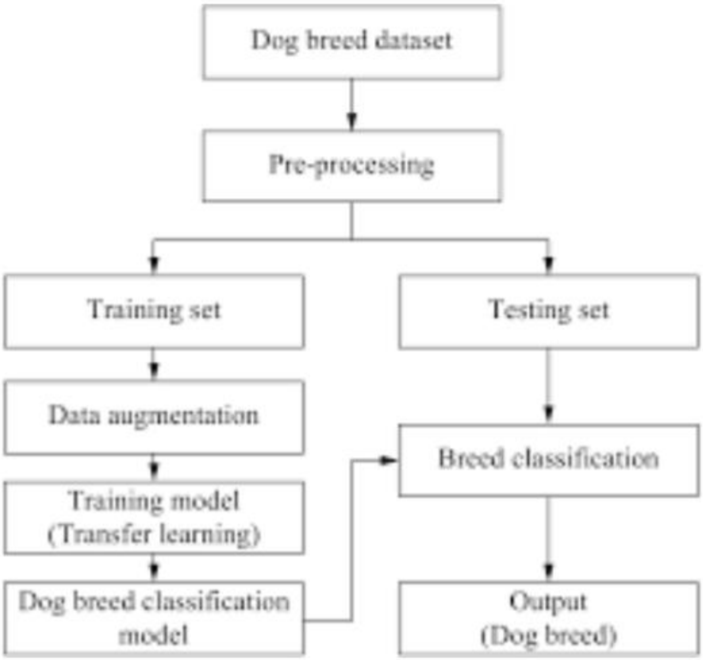
- Fairness
- Explainability
- Generalization
- Reliability
- User Friendliness
- Performance
- Privacy

5. PROJECT DESIGN

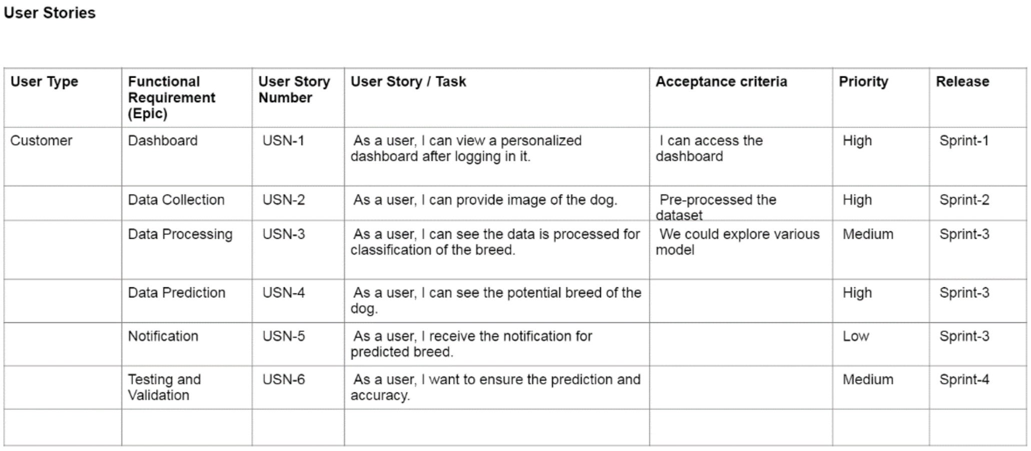
5.1 Data Flow Diagrams & User Stories

Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories :



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Dashboard	USN-1	As a user, I can view a personalized dashboard after logging in it.	I can access the dashboard	High	Sprint-1
	Data Collection	USN-2	As a user, I can provide image of the dog.	Pre-processed the dataset	High	Sprint-2
	Data Processing	USN-3	As a user, I can see the data is processed for classification of the breed.	We could explore various model	Medium	Sprint-3
	Data Prediction	USN-4	As a user, I can see the potential breed of the dog.		High	Sprint-3
	Notification	USN-5	As a user, I receive the notification for predicted breed.		Low	Sprint-3
	Testing and Validation	USN-6	As a user, I want to ensure the prediction and accuracy.		Medium	Sprint-4

5.2 Solution Architecture

1. Data Collection and Preprocessing:

• Gather a large dataset of dog images with labelled breeds.

• Preprocess the data by resizing, normalising, and augmenting the images to ensure diversity and prevent overfitting.

2. Model Selection and Transfer Learning:

• Choose the VGG19 pre-trained model as the base architecture.

• Remove the fully connected layers from VGG19 and keep the convolutional base.

• Freeze the convolutional layers to prevent their weights from being updated during training.

• Add custom fully connected layers on top of the VGG19 architecture to adapt it to the specific classification task.

3. Model Training and Evaluation:

• Split the dataset into training, validation, and testing sets.

• Train the model on the training set and fine-tune the parameters using the validation set.

• Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score.

4. Deployment and Web Development:

• Build a web application for users to upload images of dogs for classification.

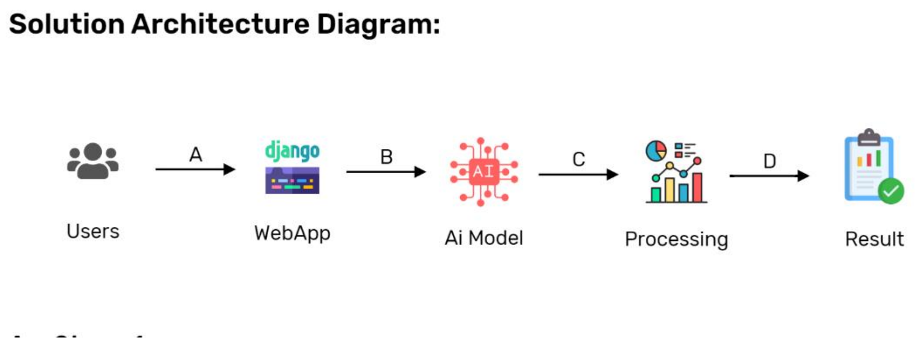
• Develop an intuitive user interface for easy interaction.

• Integrate the trained model into the web application for real-time breed classification. • Ensure scalability and responsiveness for different screen sizes and devices.

5. User Experience and Accessibility:

• Prioritise user experience by creating an intuitive and visually appealing interface.

• Ensure accessibility for users with disabilities by following accessibility guidelines and standards.



Solution Architecture Diagram:

A. Step - 1: •

The user will give an image as input and will upload jpg, png, jpeg etc. files.

• The image will be uploaded to the web app, which will be designed in Django.

B. Step - 2:

• The data will be sent to Ai Model, which will be embedded in the Django Web app.

• The image will be processed and evaluated.

C. Step - 3:

• The data will be processed and converted to Tensors and further evaluated.

D. Step - 4:

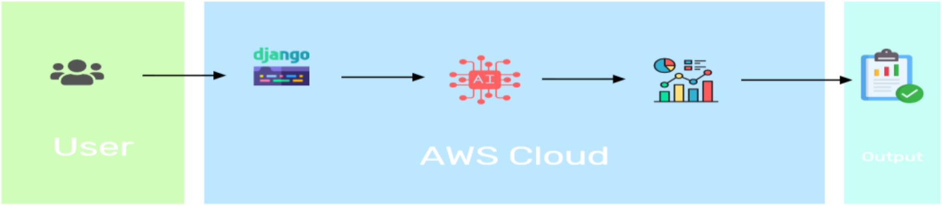
• The user will give an image as input and will upload jpg, png, jpeg etc. files.

• The image will be uploaded to the web app, which will be designed in Django. Result: Here the processed data and the result will be displayed to the user in the website UI.

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

The Deliverable shall include the architectural diagram below and the information as per the table1 & table 2



1. **AI Model Build using Transfer Learning Process:**

   - **Data Collection**: Gather relevant data for the target domain.
   - **Preprocessing**: Clean, preprocess, and augment the data to improve model performance.
   - **Model Selection**: Choose a pre-trained model suitable for transfer learning.
   - **Feature Extraction**: Extract relevant features from the pre-trained model.
   - **Fine-tuning**: Fine-tune the pre-trained model using the domain-specific dataset.
   - **Evaluation**: Assess the model's performance and make necessary adjustments. - **Deployment**: Prepare the model for deployment within the Django website.

2. **Infrastructural Demarcation:**

   - **Local Infrastructure**: Utilize local servers or machines for model training and inference.
   - **Cloud Infrastructure**: Consider using cloud platforms like AWS, GCP, or Azure for scalability and efficient resource management.

3. **External Interfaces:**

   - **Django REST API**: Expose the AI model's functionalities through a Django REST API for seamless integration with the website.
   - **Third-Party APIs**: Integrate third-party APIs for additional functionalities, such as payment gateways or authentication services.

4. **Data Storage Components/Services:**

   - **Database Management System**: Utilize Django's built-in ORM for efficient data storage and retrieval.
   - **Cloud Storage**: Employ cloud-based storage services for managing large datasets and model artefacts.

5. **Interface to Machine Learning Models:**

   - **Django Model Integration**: Incorporate the AI model within the Django application's backend logic.

- **REST API Endpoints**: Create dedicated endpoints within the Django app to handle requests and responses related to the AI model.

- **Model Inference**: Implement model inference logic to process incoming data and generate predictions or classifications.

- **Data Visualization**: Integrate tools for visualizing model outputs or predictions within the website's front end for user interaction.

6. Table-1 : Components & Technologies:

SI No.	Component	Description	Technology
1	User interface	We're looking to build UIs using Django. This will be basically the web app deployed on AWS servers fro users to interact with	HTML, CSS, Django Templates
2	Machine Learning Model	We'll build machine learning model by using the technique called transfer learning	Python, Tensorflow, VGG19
3	AWS Cloud host	We'll use AWS Cloud to deploy our app and host the website	AWS S3
4	AWS Compute Engine	We'll use AWS EC2 instance for the deployment of ML Model	AWS EC2

7. Table-2: Application Characteristics:

SI No.	Component	Description	Technology
1	Open-Source Frameworks	We'll use Django as open source fromework to build our webapp	Django
2	Security Implementations	We'll enable security feature for the to and fro data sharing of the user like image used for prediction as it may be of his/her own dog picture	SHA-256, Encryptions,
3	Scalable Architecture	Architecture that is used will be scalable, will be build as modules and all the modules will be loosely coupled for better maintenance and scalable	Clean Architecture
4	Availability	To ensure availability w'll use AWS Cloud service, that is also know for the robustness and reliability	AWS
5	Performance	We'll leverage all the better features from the AWS Cloud, like caching region based S3 buckets and more.	AWS CDN Services

8.

6.2 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collecting data	USN-1	As a developer we first need to get the data	1	Low	1
Sprint-1	Clean data	USN-2	As a developer we first need to clean and process the data that we have got and have to make it usable	2	High	2
Sprint-2	Train model	USN-3	Now we need to train the model using transfer learning VGG19	4	High	2
Sprint-2	Test the model	USN-4	Before using the model we need to test it against the data set to make sure that it is trained properly	1	Medium	1
Sprint-3	Design a web page	USN-5	As out model is ready so we need to develop the website where we will be using our model	4	High	3
Sprint-3	Integrate the model	USN-6	Now we need to integrate the model that we have created with our web page	2	High	2
Sprint-4	Host the project	USN-7	Now we need to deploy our this project on the web so that it can be accessible to normal people and they can use it	1	Medium	1
Sprint-4	Spread it	USN-8	Now as we have hosted our project on web and it is functioning and live so now we need to share our this project to people so that they can use it	1	Low	4

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	3	2 Days	9 Nov 2023	10 Nov 2023	3	
Sprint-2	5	3 Days	11 Nov 2023	13 Nov 2023	5	
Sprint-3	6	4 Days	14 Nov 2023	17 Nov 2023	6	
Sprint-4	2	1 Days	18 Nov 2023	18 Nov 2023	2	

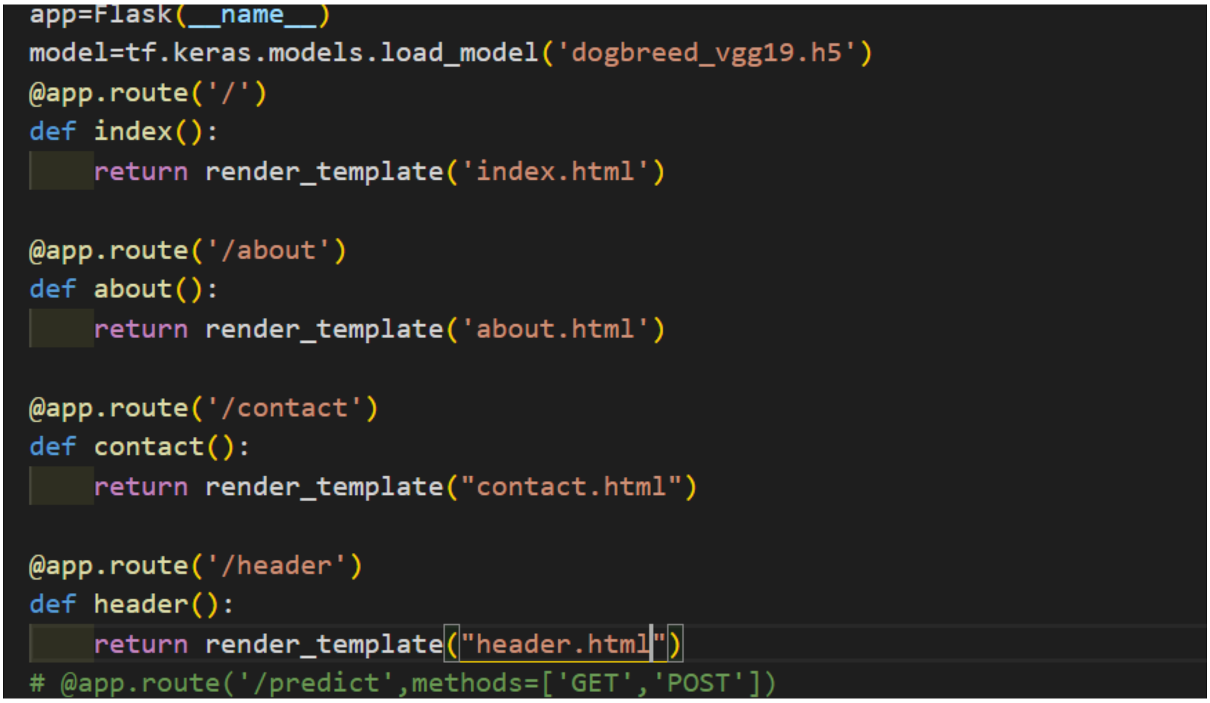
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

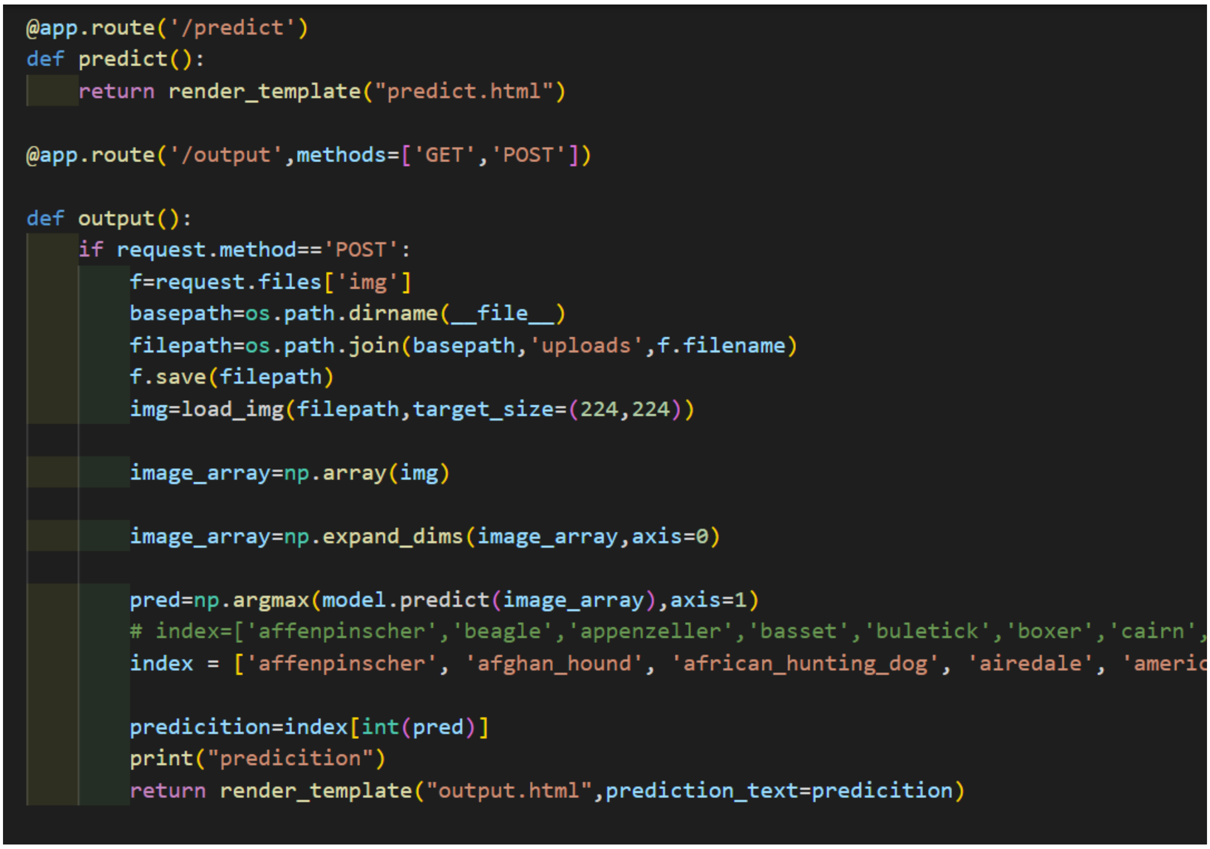
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

7. CODING & SOLUTIONING

Web App



```
app=Flask(__name__)
model=tf.keras.models.load_model('dogbreed_vgg19.h5')
@app.route('/')
def index():
    return render_template('index.html')

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

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

@app.route('/header')
def header():
    return render_template("header.html")
# @app.route('/predict',methods=['GET','POST'])
```

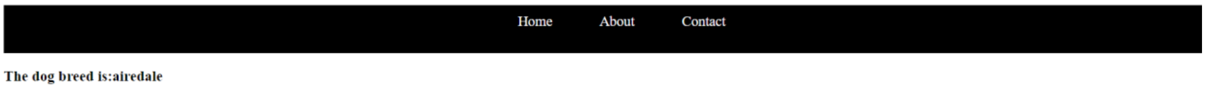


```
@app.route('/predict')
def predict():
    return render_template("predict.html")

@app.route('/output',methods=['GET','POST'])

def output():
    if request.method=='POST':
        f=request.files['img']
        basepath=os.path.dirname(__file__)
        filepath=os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)
        img=load_img(filepath,target_size=(224,224))

        image_array=np.array(img)

        image_array=np.expand_dims(image_array,axis=0)

        pred=np.argmax(model.predict(image_array),axis=1)
        # index=['affenpinscher','beagle','appenzeller','basset','buletick','boxer','cairn',
        index = ['affenpinscher', 'afghan_hound', 'african_hunting_dog', 'airedale', 'americ

        predicition=index[int(pred)]
        print("predicition")
        return render_template("output.html",prediction_text=predicition)
```

8. PERFORMANCE TESTING

The app is functioning as expected and the result is also coming as per expectation

9. RESULTS



The dog breed is:airedale

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Reduced Development Time and Cost: Transfer learning allows for leveraging pre-trained models, reducing the time and resources required for training from scratch.

- Improved Accuracy: Transfer learning can achieve higher accuracy compared to training models from scratch, especially for complex tasks like dog breed classification.

- Domain Adaptation: Transfer learning can adapt to specific domains, such as dog breed classification, by fine-tuning the pre-trained model on relevant data.

- Reduced Data Requirements: Transfer learning can reduce the amount of data required for training, which is particularly beneficial for resource-constrained applications.

Disadvantages:

- Potential Bias: Transfer learning models may inherit biases from the pre-trained models, which can affect the fairness and accuracy of predictions.

- Limited Flexibility: Transfer learning models may not be as flexible as models trained from scratch, making it challenging to customize the model for specific requirements.

- Interpretability Challenges: Transfer learning models can be less interpretable compared to simpler models, making it difficult to understand their decision-making process.

- Overfitting Risk: If not carefully fine-tuned, transfer learning models can overfit to the specific training data, reducing their generalization ability.

- Computational Requirements: Transfer learning models can still be computationally expensive, requiring sufficient hardware resources for training and inference.

11. CONCLUSION

Dog breed classification application utilizes machine learning to reliably identify dog breeds from images. Its accuracy stems from extensive testing using a diverse dataset of dog photographs. Currently, the application provides basic information about each recognized dog breed, gathered from various sources. The model's underlying learning mechanism employs Convolutional Neural Networks (CNNs), widely recognized for their effectiveness in image classification tasks. The CNN-based deep learning method is specifically tailored to predict dog breeds from input images. Leveraging transfer learning, the model is capable of classifying a wide range of dog breeds with high accuracy.

12. FUTURE SCOPE

combining transfer learning with techniques like ensemble approaches, fine-tuning, and advancements in neural network architectures can enhance the accuracy of identifying subtle breed-specific characteristics. Furthermore, progress in explainable AI can illuminate how these models make decisions, fostering their usability and trustworthiness in practical applications such as veterinary care, pet care, and breeding. By leveraging these advancements, transfer learning holds the potential to revolutionize the future of dog breed identification systems, making them increasingly precise, reliable, and widely applicable.

13. APPENDIX

Source Code

Git Hub and Project Demo Link

Demo Link:

Git Hub: Git Hub link