

Project Design Phase-I

Solution Architecture

Date	02 November 2023
Team Id	Team-593195
Project Name	Dog Breed Identification using Transfer Learning
Maximum Marks	

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.

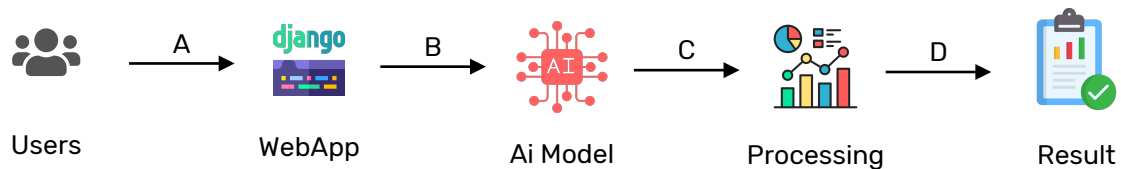
5. User Experience and Accessibility:

- Prioritise user experience by creating an intuitive and visually appealing interface.
- Ensure accessibility for users with disabilities by following accessibility guidelines and standards.

To implement the proposed solution to achieve the ***“Classification of Dog Breed”*** we’ll be using the below-described tech stacks, data-flow and tech-architecture.

- For front-end and user interaction, we'll be using **Django**.
- For Transfer-Learning we'll be using **VGG19 Arch**.

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.