AUTHOR :- KANSARA RUSHABH BHAVESH TECHNOLOGY/DOMAIN :- MACHINE LEARNING

INTRODUCTION

→ This project focuses on developing a machine learning pipeline to classify images into specific animal categories. The primary objective is to train a deep learning model that can accurately identify distinct animal species from input images. The dataset used for this project consists of 15 different classes, including Bear, Bird, Cat, Cow, Deer, Dog, Dolphin, Elephant, Giraffe, Horse, Kangaroo, Lion, Panda, Tiger, and Zebra. The workflow involves data loading, visualization, and leveraging pre-built neural network architectures to achieve classification tasks.

SOFTWARE REQUIREMENT SPECIFICATION

LIST	DESCRIPTION
Python	Version 3.7 or higher The syntax and libraries used (especially the f-strings or newer Pandas features if you use them later) require a modern Python version.
Development IDE	Jupyter Notebook

MODEL USED

→ **DESCRIPTION :-** This is a Convolutional Neural Network (CNN) architecture designed for mobile and resource-constrained environments. It is imported from tensorflow.keras.applications in the project.

→ **RESAON :-** MobileNetV2 is widely used in image classification tasks because it is lightweight and computationally efficient while maintaining high accuracy. In this project, it likely serves as a feature extractor or a base model for transfer learning, allowing the system to leverage pre-learned patterns from large datasets (like ImageNet) to classify the specific animals in this dataset.

CORE IDEA

→ The core idea of this project is Transfer Learning for Image Classification.

→ Instead of building a complex Artificial Intelligence "brain" from scratch, this project leverages a pre-existing, highly advanced model (MobileNetV2) that has already "learned" to see patterns in millions of images. You are essentially taking this expert model and fine-tuning it to become a specialist in identifying your 15 specific animal categories.

1. The “Expert ” foundation (Feature extraction)

→ **Concept :-** The project uses MobileNetV2 as the foundation. This model already knows how to detect edges, textures, shapes, and complex patterns because it was trained on a massive dataset (ImageNet).

→ **Application :-** By passing your animal images through this model, you extract meaningful features (like the stripes of a zebra or the trunk of an elephant) without having to teach the computer what a "stripe" or "trunk" is from square one.

2. The "Specialist" Layer (Classification Head)

→ **Concept :-** You freeze the "brain" of MobileNetV2 (so it doesn't forget what it knows) and add your own custom layers on top.

→ **Application :-** These new layers are the only parts you train extensively. They take the features detected by MobileNetV2 and learn to map them specifically to your 15 classes (Bear, Lion, Panda, etc.).

3. Data Efficiency

→ **Concept :-** Deep learning usually requires massive amounts of data and computing power.

→ **Application :-** Because you are using Transfer Learning, you can achieve high accuracy with a relatively small dataset (around 120-137 images per class) and much faster training times compared to building a custom Convolutional Neural Network (CNN) from scratch.

LIST OF LIBRARIES AND FRAMEWORKS USED IN PROJECT.

LIBRARIES	USE
Numpy	For numerical operations, specifically for handling image as numerical arrays (tensors) that the model can process.
Tensorflow and Keras	These are the core frameworks for building and training the deep learning model. MobileNetV2 :- Pre-trained model for transfer learning ImageDataGenerator :- Used for loading images from directories and likely performing data augmentation
Matplotlib	Used for visualization, specifically to plot the bar chart showing the distribution of animal classes in the dataset
OS	A standard python library used to interact with the operating system, allowing the code to navigate directories and file paths to locate the image dataset
Pillow (PIL)	The python imaging library, used for opening, manipulating and saving image files.

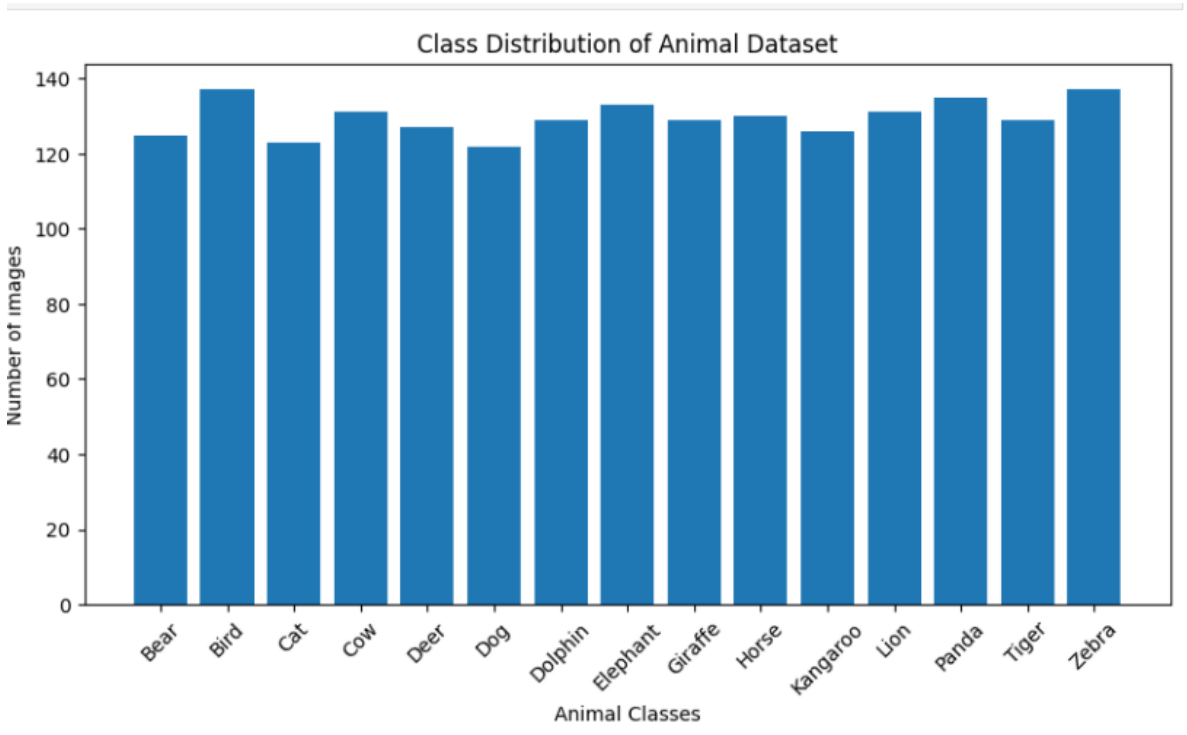
DIAGRAMS AND VISUALIZATIONS

1. BAR CHART.

→ **DESCRIPTION :-** A bar graph that visualizes the number of images available for each of the 15 animal classes.

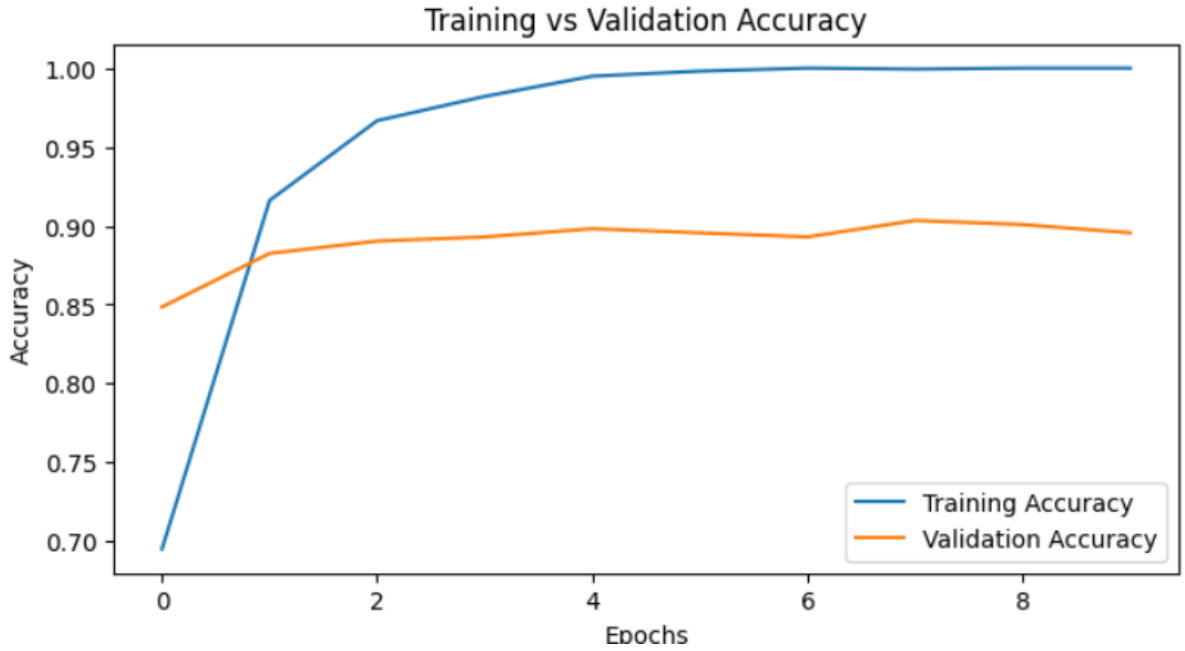
→ **REASON :-** This diagram is critical for analysing the dataset's structure. It helps identify class imbalances—situations where one animal type might have significantly more or fewer images than others.

→ **IMPORTANCE :-** Ensuring a balanced dataset is vital for model performance. If the data is imbalanced, the model might become biased, performing well on common classes (e.g., Birds with 137 images) but poorly on others (e.g., Dogs with 122 images). The visualization confirms that the dataset is relatively balanced, with image counts ranging between roughly 120 and 137 per class.



2. LINE CHART

→ Shows trends over time.
→ Used for continuous data.
→ Example :- training vs validation accuracy, stock prices, temperature change.



3. IMAGE GRID

→ Displays sample images from each animal class.
→ Help us to verify image quality, correct folder labeling and Variations in images.

FUTURE ENHANCEMENT

1. Add more animal classes.

→ Right now there are limited number of animal categories.
→ In future there will be more animal classes like Elephant, Giraffe, Zebra etc.

2. Increase Dataset Size

→ The current status is Medium Size which is Risk of Overfitting.
→ There will more images per class for removing the problem of "Overfitting".
→ This will led us for the better validation accuracy.

3. Apply Data Augmentation

→ Model trained on original images only.
→ The Future Updation will be to adding rotation, flipping, zoom, brightness variation.

4. Add Confusion Matrix and Classification Report.

→Currently :- Only accuracy and loss curves
→Future Updation :- Precision, Recall , F1-score.

5. Implement Model saving and Reloading.

→ Updation explanation :- Save the trainded model in ".h5" or ".keras" file.
→ Reload model for interface.