Project Structure

```
Animal_data/
■■■ train/
■ ■■■ [class_1]/ (e.g. cats/)
■ ■■■ [class_2]/ (e.g. dogs/)
■■■ validation/
■ ■■■ [class_1]/ (e.g. cats/)
■ ■■■ [class_2]/ (e.g. dogs/)
■■■ animal_classifier_model.h5
■■■ predict_and_show.py
■■■ main_training.py
■■■ README.md
```

Features

- Data Augmentation for better generalization
- CNN-based image classification
- Live prediction for individual images
- Model evaluation using accuracy, confusion matrix, and classification report
- Visualization of predictions using Matplotlib

Model Architecture

```
model = Sequential([
Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
MaxPooling2D(2, 2),
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Conv2D(128, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Flatten(),
Dropout(0.5),
Dense(512, activation='relu'),
Dense(train_generator.num_classes, activation='softmax')
])
```

How to Run

1. Install dependencies:

```
pip install tensorflow numpy matplotlib scikit-learn
```

2. Train the Model:

```
python main_training.py
```

3. Make a Prediction:

```
python predict_and_show.py
```

Sample Prediction

```
img_path = r"C:\path\to\validation\cats\cat.119.jpg"
print(f"Predicted animal: {predict_animal(img_path)}")
```

Output:

```
Predicted animal: dolphin
```

Result

```
Accuracy: 1.0
Confusion Matrix:
[[2500]]
Classification Report:
precision recall f1-score support
cats 1.00 1.00 1.00 2500
accuracy 1.00 2500
macro avg 1.00 1.00 1.00 2500
weighted avg 1.00 1.00 1.00 2500
```

Libraries Used

- TensorFlow
- Keras
- NumPy
- Matplotlib
- Scikit-learn

Author

Roshani Patharkar