# WILDLIFE MONITORING

## TECHNICAL DOCUMENTATION

## COMPUTER VISION

MOHAMED EMAD HUSSIEN
MOAMEN MOSTAFA GABALLAH

# Technical Documentation: Wildlife Monitoring System Using Machine Learning

## 1. Introduction

This document provides technical details about the implementation and evaluation of the wildlife monitoring system using machine learning.

## 2. Dataset Description

**Dataset:** Animals-10 Dataset

- **Classes:** Cane (Dog), Cavallo (Horse), Elefante (Elephant), Farfalla (Butterfly), Gatto (Cat), Mucca (Cow), Pecora (Sheep), Ragno (Spider), Scoiattolo (Squirrel).
- **Structure:** Images organized in class-wise directories.

## 3. Preprocessing Steps

1. **Image Loading:** Images are resized to 64x64 pixels.
2. **Normalization:** Pixel values are scaled to the range [0, 1].
3. **Data Splitting:**
    - Training Set: 80%
    - Test Set: 20%

## 4. Models Implemented

### Logistic Regression

- **Purpose:** Binary classification between two classes.
- **Parameters:** Learning rate = 0.1, Iterations = 438.

### Softmax Regression

- **Purpose:** Multi-class classification.
- **Parameters:** Learning rate = 0.1, Iterations = 438.

### Support Vector Machine (SVM)

- **Kernel:** RBF
- **Hyperparameters:** C = 1.0, Gamma = 'scale'

**K-Nearest Neighbors (KNN)**

- **Hyperparameters:** k = 3

**Feed-Forward Neural Network**

- **Layers:**
  - Input: Flattened images
  - Hidden: Dense(128, activation='relu')
  - Output: Dense(num_classes, activation='softmax')
- **Optimizer:** Adam, Learning rate = 0.001
- **Epochs:** 10

**Convolutional Neural Network (CNN)**

- **Architecture:**
  - Conv2D (32 filters, 3x3 kernel)
  - MaxPooling2D (2x2 pool)
  - Dropout (0.25)
  - Conv2D (64 filters, 3x3 kernel)
  - MaxPooling2D (2x2 pool)
  - Dropout (0.25)
  - Flatten
  - Dense (128, activation='relu')
  - Dropout (0.5)
  - Dense (num_classes, activation='softmax')
- **Optimizer:** Adam, Learning rate = 0.001
- **Epochs:** 10

**5. Evaluation Metrics**

1. **Accuracy:** Measures overall correctness.
2. **Classification Report:** Precision, Recall, F1-score.
3. **Confusion Matrix:** Visualized using heatmaps.

## 6. Results and Analysis

### Logistic Regression

- Accuracy: 73%
- Observations: Suitable for binary classification tasks.

### SVM

- Accuracy: 46%
- Observations: High accuracy but computationally expensive.

### CNN

- Accuracy: 64%
- Observations: Best performance due to hierarchical feature learning.

## 7. Future Enhancements

- Implement transfer learning using pre-trained models like VGG16 or ResNet.
- Extend the system to handle videos for real-time monitoring.
- Incorporate additional preprocessing techniques to handle noisy datasets.

## 8. Conclusion

This project demonstrates the potential of machine learning in automating wildlife monitoring tasks. The implemented models provide a strong foundation for further development and real-world deployment.