Oklahoma State University Test Plan Document
Project 2 Animal Detection and Classification System.

December 4, 2024

Mohamad Ali

Contents

Introduction 3
Unit Test 3
File Upload 3
YOLO Object Detection 3
Image cropping 3
Roboflow Classification 3
Image Processing 3
Video Processing 3
GUI 3
Integration Test 4
Image Processing 4
Video Processing 4
Classification Handling 4
GUI Interaction 4
Error Handling 5

Introduction

This document provides a Test Plan for the Animal Detection and Classification System, a Python code using YOLO for object detection and Roboflow for object classification. The application lets users upload images and videos, processes them to detect and classify animals, and outputs annotated images and videos with bounding boxes and labels. The system includes a user interface (GUI) for user interaction.

Unit Test

File Upload

The application correctly handles different file types (image, video) and users can browse and upload images and videos

YOLO Object Detection

Bounding boxes and confidence scores are correctly returned.

Image cropping

A cropped image file is saved in the Detected Animals directory with the expected dimensions.

Roboflow Classification

Class names and confidence scores are correctly returned for valid images.

Image Processing

cropping and annotation of detected objects in images are correct and as expected

Video Processing

Test that each frame of a video is read and processed correctly. Frames are annotated correctly with bounding boxes and labels. progress bar updates during video processing the progress bar reflects the percentage of frames processed.

GUI

Verify detection and classification threshold sliders. Adjusting slides dynamically affects thresholds without errors. Processed images appear on the GUI

Integration Test

Image Processing

1. Upload a sample image.
2. Detect and classify objects.
3. Display the annotated image in the GUI.

Expected results

Detected objects are annotated with bounding boxes and labels (red box for "coyote" and green for all other classes).

Video Processing

1. Upload a sample video.

2. Process frame-by-frame.

3. Progress bar advance.

4. Save the annotated video in the Processed_Videos directory.

Expected results

Annotated video is saved, with bounding boxes and labels on detected objects. (red box for "coyote" and green for all other classes).

Classification Handling

1. Detect an object.
2. Crop and classify the object using Roboflow.

Expected results

Correct classification labels are applied to bounding boxes, and the label matches the Roboflow response.

GUI Interaction

1. Adjust detection and classification thresholds using sliders.
2. Upload and process a file.
3. Monitor changes in detection results.

Expected results

Threshold changes and affect detection results.

Error Handling

1. Upload unsupported file types.
2. Test incomplete or invalid Roboflow responses.
3. Internet connection loss

Expected results

Errors are handled without crashing the program, with messages shown to the user.