**POND FISH DETECTION MODEL**

**Team details:**

Team name: STATIS

Team members: Thanush A A(team leader), Dilli B

**Summary:**

We propose two different pond fish detection algorithms (MSR-yolov8s and CLAHE-yolov8s) as a part of the DePondFi’23 challenge. The performance of both the proposed models are similar, with CLAHE-yolov8s being slightly more accurate than MSR-yolov8s. However, the image processing speed of CLAHE-yolov8s model was found to be faster than the MSR-yolov8s model. Hence, both of the models were proposed so that the users could choose the model based on their preferences. A python script has been developed through which users can interact in a command-line interface to choose the type of object detection model and get the bounding box coordinates by providing the required set of inputs. A Detailed explanation of the proposed models, dependencies required, and instructions to get the bounding box coordinates are provided in the upcoming sections.

**FISH DETECTION MODELS:**

The models proposed in this project are:

(i) MSR-YOLOv8s

(ii) CLAHE-YOLOv8s

The models were trained using the training dataset provided in the DePondFi challenge.

**1. MSR-YOLOv8s:**

Multi Scale Retinex (MSR) is an image enhancement algorithm based on the Retinex theory, which explains human colour perception. The theory is based on the following assumptions:

i. The colour perceived by the eyes is a result of the interaction of light and matter

ii. Each colour region is composed of three primary colours of a given wavelength: red, green, and blue

iii. The three primary colours determine the colour of each unit area

According to retinex theory, the colour of an object is not affected by the intensity of the reflected light, but on the constancy of colour perception.

The MSR-YOLOv8s model processes the input images using MSR algorithm before feeding it to the yolov8s architecture for fish detection.

**2. CLAHE-YOLOv8s:**

Contrast Limit Adaptive Histogram Equalization (CLAHE) is an image enhancement algorithm that is a variation of the adaptive image enhancement (AHE) algorithm. The CLAHE algorithm consists of three parts: The tile generator, Histogram equalizer, and the bilinear interpolator. The given input image is split into smaller units known as tiles. These tiles undergo histogram equalization, which is a five-step computation process. The resultant tiles are then joined using bilinear interpolation. The algorithm can generate output images with increased contrast.

The CLAHE-YOLOv8s model processes the input images using CLAHE algorithm before feeding it to the yolov8s architecture for fish detection.

A comparison of the performance of the models are provided in the table below. The mean average precision values were based on the validation dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | mAp50 | F1 score | Inference time per image frame (seconds) |
| MSR-YOLOv8s | 0.9709 | 0.932 | 38.44 |
| CLAHE-YOLOv8s | 0.8612 | 0.842 | 25.88 |

NOTE: inference time includes the time for processing the image and the time for detection

**DEPENDENCIES REQUIRED:**

Python 3.9.7 or higher

Pip 23.0.1 or higher

**Python modules required (can be installed using pip):**

matplotlib>=3.2.2

opencv-python>=4.6.0

Pillow>=7.1.2

PyYAML>=5.3.1

requests>=2.23.0

scipy>=1.4.1

torch>=1.7.0

torchvision>=0.8.1

tqdm>=4.64.0

numpy>=1.21.6

ultralytics (latest version)

natsort>=8.3.1

xlwt>=1.3.0

**HOW TO GENERATE THE BOUNDING BOX COORDINATES:**

1. Open the command line interface of your Operating System in the root directory of the project

2. Type “python image\_text\_annotation.py”

3. A prompt will appear on the image enhancement algorithm to be applied (MSR/CLAHE). Type the algorithm that you want to use

4. Type the directory containing the image(s) using ‘/’ to separate the folders instead of ‘\’. Make sure that the location specified in the prompt ends with ‘/’ to avoid errors

5. Wait for the enhancement and detection to complete

6. The coordinates would be saved in the results folder in yolov5 annotations format

**HOW TO GENERATE IMAGE WITH BOUNDING BOX OUTPUT:**

1. Open the command line interface of your Operating System in the root directory of the project

2. Type “python image\_detection.py”

3. A prompt will appear on the image enhancement algorithm to be applied (MSR/CLAHE). Type the algorithm that you want to use

4. Type the directory containing the image(s) using ‘/’ to separate the folders instead of ‘\’. Make sure that the location specified in the prompt ends with ‘/’ to avoid errors

5. Wait for the enhancement and detection to complete

6. The directory containing image(s) with bounding boxes would be saved in the ‘runs/detect/’ folder

**HOW TO GENERATE VIDEO WITH BOUNDING BOX OUTPUT:**

1. Open the command line interface of your Operating System in the root directory of the project

2. Type “python video\_detection.py”

3. A prompt will appear on the image enhancement algorithm to be applied (MSR/CLAHE). Type the algorithm that you want to use

4. Type the location of the video file

(e.g.): path/to/video.extension

5. Wait for the enhancement and detection to complete

6. The directory containing image(s) with bounding boxes would be saved in the ‘runs/detect/’ folder

7. Make sure that the video in the detections directory is deleted before running the code again

**TEST DATASET RESULTS AND ANNOTATIONS:**

The test dataset results of the individual models containing the bounding box images and annotations are available in the test\_results directory