IoTracX

AI SURVEILLANCE

HUMAN DETECTION

Chandra Vardhan Madamanchi
S.Krishnan

AI SURVIALLANCE AND RESUCE INTERNSHIP REPORT

TEAM

Madamanchi Chandra Vardhan

S Krishnan

MENTORS

Shyam Purkayastha

Gopi S

Table of Contents:

Abstract

Unmanned aerial vehicles (drones) equipped with high-resolution cameras and powerful GPUs have the potential to enhance Search-and-Rescue (SAR) operations in remote and hostile environments. Rapidly locating unconscious or injured victims is crucial for improving their chances of survival. This paper focuses on using drones as flying computer vision systems to increase detection rates and reduce rescue time. We conducted an experimental evaluation using the YOLOv5,Yolov7,Yolov8 algorithms, a lightweight version of the YOLO detection algorithm, on two newly created benchmark datasets specifically designed for SAR with drones. The results demonstrate promising outcomes, showcasing competitive detection accuracy comparable to state-of-the-art approaches but with significantly faster detection times.

Problem Statement

We have been given problem statement of Identifying and detecting humans in aerial view images captured by drones and UAV's. Aerial images particularly of terrains , grass fields etc.So we need to develop model such it should be able to identify and detect humans.

In rescue and surveillance operations, the ability to detect and locate humans accurately and efficiently in large-scale environments is of paramount importance. Traditional ground-based approaches often face limitations in terms of coverage, accessibility, and real-time response. Therefore, there is a growing need for robust and effective human detection systems that leverage aerial images obtained from platforms such as drones or satellites.

However, detecting humans in aerial images poses unique challenges. Aerial images exhibit variations in scale, perspective, occlusion, and lighting conditions, making it difficult to apply conventional computer vision techniques designed for ground-level images. Furthermore, the presence of cluttered backgrounds, diverse poses, and the small size of humans in the images further exacerbate the complexity of the task.

Objective

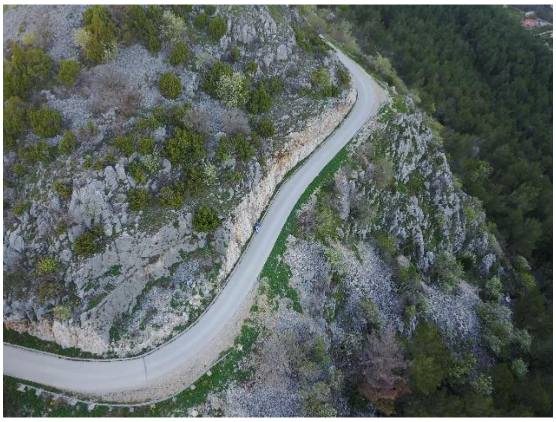
The objective of this project is to develop a robust and efficient system for human detection using aerial image datasets, specifically focusing on aerial images obtained from platforms such as drones or satellites. The primary goal is to enable the detection of humans in real-time or near real-time, with a high level of accuracy and reliability. This system will be specifically designed for applications in rescue and surveillance operations, where the identification and tracking of humans in large-scale environments are crucial for effective decision-making and resource allocation.

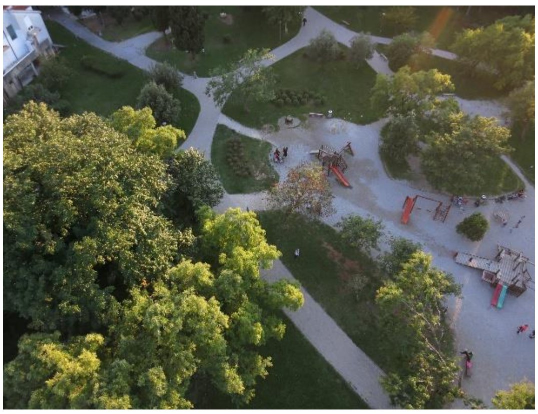
So, our main objective is to develop a model to used in drone systems such that it should detect the human's movements, actions. We use this technology in Industrial surveillances , to identify any movements of humans or animals over the Industrial Space.

Introduction

So according to our problem statement we have chosen heridal dataset for developing model.

The **HERIDAL** dataset contains approximately 1700 images of wildlife in various locations captured from an aerial perspective with drones equipped with a high-definition camera. Images were captured by various UAVs from custom solutions to popular solutions such as DJI Phantom 3 or Mavic Pro, at altitudes from 30m to 60m. HERIDAL contains 4000 ×3000 full-size, labelled real images, split into 1583 training and 101 testing images. The images consist of realistic scenes of mountains, wilderness or remote places in non-urban areas. Since most research is conducted in remote locations outside of urban areas, the emphasis is on land and natural environments. In particular, the images were collected in various locations in Croatia and Bosnia-Herzegovina during mountain hikes, nature trips or during mountain rescue exercises. Most of them have more than one person, on average 3.38 people. In order to compile a dataset that would be a realistic representation of a real SAR operation, the authors used statistical data and specialist knowledge on SAR operations. In fact, there are many variations of the positions (standing, lying, squatting, etc.) in which a lost person can be found. Nevertheless, this specific information is not labelled, so other training and testing was done for human detection only.





DATA PREPROCESSING

Data Collection

We skimmed though the web to find a dataset that matched the requirements of the problem statement and found Heridal dataset which was used in one the research papers which was trying to tackle the problem of detecting humans during rescue operations. Hence, we finalised the heridal dataset for our problem statement.

Data Annotation

The images that were obtained already had annotations, the humans in the images were the objects to be detected and the bounding boxes were already provided.

Data Augmentation

From the raw heridal dataset we only tool the heridal train images folder which had 1583 images and only 1548 annotations or labels were given, hence we only took the images which had annotations and made a dataset called heridal version 1 which had 1548 images. Then later the following preprocessing steps we done

Pre-Processing:

1. Auto-Orient: Applied
2. Isolate Objects: Applied
3. Resize: Stretch to 800x800

Later the following data augmentation techniques were applied to the heridal version 1 dataset, note that the data augmentation techniques are applied to the locality of the object (here Human)

Augmentations:

1. Outputs per training example: 3
2. Flip: Horizontal, Vertical
3. Shear: ±25° Horizontal, ±25° Vertical
4. Noise: Up to 15% of pixels
5. Bounding Box: Shear: ±25° Horizontal, ±25° Vertical
6. Bounding Box: Blur: Up to 5px

The above data augmentations were made with the help of roboflow as the resulting dataset is named heridal version 2. The heridal version 2 dataset was then spilt into training set, validation set and testing set with each set containing 6.1k, 602 and 259 images respectively.

Roboflow offers various data augmentation techniques that can be applied to a heridal dataset to increase its diversity and improve the performance of your object detection model. Here are some commonly used data augmentation methods available in Roboflow:

1. Rotation: This augmentation rotates the images by a specified angle, helping the model to generalize better to objects in different orientations.
2. Flipping: Flipping augments the dataset by horizontally flipping the images. It can be useful when objects in your heridal dataset exhibit symmetrical characteristics.
3. Scaling: Scaling modifies the size of the images by either enlarging or shrinking them. This augmentation helps the model learn to detect objects at different scales, enhancing its robustness.
4. Crop: Cropping randomly selects a portion of the image, discarding the rest. This technique can be effective for training the model to focus on specific regions of interest in the heridal dataset.
5. Translation: Translation shifts the objects within the image by a certain distance. It simulates the presence of objects at different locations, improving the model's ability to detect them accurately.
6. Noise Addition: Adding random noise to the images introduces variations and makes the model more tolerant to noise in real-world scenarios.

Roboflow provides a user-friendly interface where you can easily configure and apply these data augmentation techniques to our heridal dataset. We can experiment with different combinations of augmentations to find the most effective ones for our specific use case. Additionally, Roboflow allows us to preview augmented images, ensuring that the augmentation settings produce the desired transformations.

Methods and Methodology

We followed implementing using 3 models :

1.**YOLOv5** using Kaggle notebook

2.**YOLOv7** using Kaggle notebook

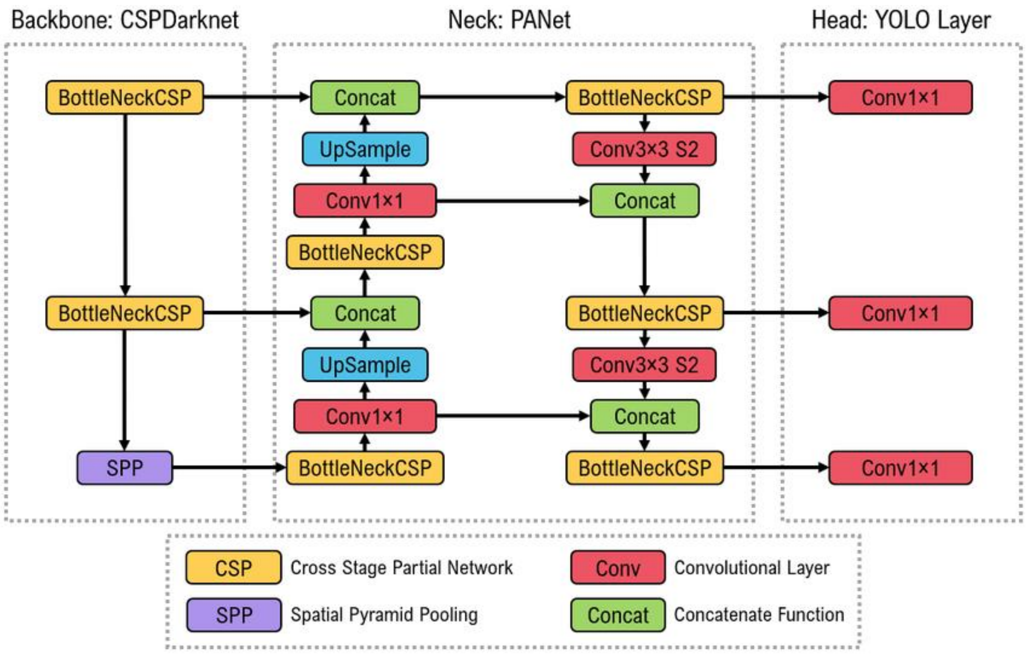
3.**YOLOv8** using Roboflow

1.YOLOv5

We have gone through different models and algorithms for detection technique and after going through few research papers we have decided to YoloV5 model.

YOLO is a regression-based method and is in fact much faster than region proposal-based methods (such as R-CNN), although it is not as accurate. The idea behind YOLO is to realize object detection by considering it as a regression and classification problem: the first is used to find the bounding box coordinates for the objects in the image, while the second is used to classify the objects found in an object class. This is done in a single step by first dividing the input image into a grid of cells, and then calculating, for each cell, the bounding box and the relative confidence score for the object in that cell. Although the latest stable version of YOLO is YOLOv4 , we used YOLOv5 , which is still in development. The latest version of YOLO was chosen because several empirical results showed how it can work accurately, compared to YOLOv4, but with an extremely smaller model size.1 Many controversies about YOLOv5 have been raised by the community.2 These controversies are mainly caused by the fact that YOLOv5 does not even (yet) have a published paper; nevertheless, we have preferred to use the latter version to obtain experimental results that could be a valid reference for future work, as other projects, such as, have done. In some experiments on the well-known COCO dataset , YOLOv5s showed much lower training time and model storage size than the custom YOLOv4 model. Also, YOLOv5 takes less inference time, making it faster than YOLOv4.

YOLOv5 is different from all other previous versions; in particular, the major improvements introduced include mosaic data augmentation and the ability to autonomously learn bounding box anchors, i.e. the set of predefined bounding boxes of a certain height and width used to detect objects.



So we used yolov5 to train the model with heridal dataset.

Steps followed:

1.Prepairing Dataset

2.Installing and cloning Yolov5 from Ultralytics GitHub repository

3. Train the YOLOv5 model

4. Test the YOLOv5 model

5. Evaluate the YOLOv5 model

Initially we trained yolov5 to coco2018 dataset by setting up required libraries pytorch etc Later trained on Heridal dataset. For this we generated yaml file which is used to train yolov5 ,yaml describes about our data train , test images and classes it has to yolo architecture.

Later we shifted our model yolov7 , due to some inconveniences and results of yolov7 considered we shifted yolov7 for better performance.

2.YOLOv7

YOLOv7 is a deep learning-based object detection algorithm that can be used to detect humans in aerial view images.It is a single-stage object detector, which means that it can detect objects in a single pass through the image.YOLOv7 has been shown to be effective at detecting humans in aerial view images, even when the images are of low quality or when the humans are small or obscured.

YOLOv7 first divides the image into a grid of cells.For each cell, YOLOv7 predicts the probability of a human being present in the cell, as well as the coordinates of the bounding box around the human.YOLOv7 then uses a non-maximum suppression algorithm to remove overlapping bounding boxes.

Steps followed:

1. Preparing the dataset:
    - Split your dataset into training and testing sets. Ensure that you have annotated bounding boxes around the human objects in your training images.
    - Organize the dataset according to the required YOLOv7 format. Each image should have a corresponding text file with the same name containing the annotations for the objects present in the image. The annotation format should include the class label (e.g., "human") and the coordinates of the bounding box (normalized values between 0 and 1).
    - Additionally, you may need to generate patches from your images and annotate them as well if you plan to use patch-based training.
2. Setting up the YOLOv7 environment:
    - Install the necessary dependencies, such as Python, PyTorch, and CUDA, if required. You can find the detailed installation instructions in the official YOLOv7 repository or documentation.
3. Obtaining the YOLOv7 source code:
    - Clone the YOLOv7 repository from the official source. You can find the repository on GitHub or other platforms where it is available.
4. Configuring the YOLOv7 model:

- Customize the YOLOv7 configuration file according to your requirements. You may need to modify parameters such as the number of classes (in your case, 1 for humans), the anchor sizes, and other hyperparameters. The configuration file is typically a .cfg file provided in the YOLOv7 repository.

5. Training the YOLOv7 model:
    - Use the training script provided in the YOLOv7 repository to train your model. This script usually requires you to provide the path to your training and testing datasets, the path to the configuration file, and other relevant parameters.
    - Start the training process and allow the model to learn from your annotated data. This may take some time depending on the size of your dataset and the complexity of the task.
6. Evaluating the trained model:
    - Once the training is complete, you can evaluate the performance of your trained YOLOv7 model using the testing dataset.
    - The evaluation metrics typically include precision, recall, and mean average precision (mAP). You can assess how well the model detects human objects in your aerial images.
7. Fine-tuning and optimization (optional):
    - If the performance of the trained model is not satisfactory, you can try fine-tuning the model by adjusting the hyperparameters, augmenting the training data, or using other techniques to improve the detection accuracy.
8. Inference using the trained model:
    - Once you are satisfied with the performance of the trained model, you can use it for inference on new, unseen aerial images to detect human objects.
    - You can apply the trained model to full images or use the patches you generated during training to detect humans in a patch-based manner.

3.YOLOv8

Ultralytics YOLOv8 is the latest version of the YOLO (You Only Look Once) object detection and image segmentation model developed by Ultralytics. The YOLOv8 model is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and image segmentation tasks. It can be trained on large datasets and is capable of running on a variety of hardware platforms, from CPUs to GPUs.

We are using roboflow which provides notebook along with models which we need to run.

Here are the detailed steps for each step:

1. Create a project in Roboflow.

To create a project in Roboflow, you will need to create an account and then click on the "Create Project" button. You will then be prompted to give your project a name and to select a workspace.

2. Upload your images to the project.

Once you have created a project, you can upload your images to the project. You can upload images in a variety of formats, including JPG, PNG, and TIFF.

3. Label your images using Roboflow's annotation tools.

Once you have uploaded your images, you will need to label them. Roboflow provides a variety of annotation tools that you can use to label your images. You can label objects by drawing bounding boxes around them, or you can label objects by assigning them tags.

4. Generate a new version of your dataset.

Once you have labeled your images, you will need to generate a new version of your dataset. This will create a new file that contains the images, the annotations, and the metadata for your dataset.

5. Export your dataset for use with YOLOv8.

Once you have generated a new version of your dataset, you can export it for use with YOLOv8. You can export your dataset in a variety of formats, including YOLOv4, YOLOv5, and YOLOv8.

6. Train YOLOv8 on your dataset.

Once you have exported your dataset, you can train YOLOv8 on your dataset. You can train YOLOv8 using a variety of tools, including the YOLOv8 command line tool and the YOLOv8 training GUI.

7. Validate your model with a new dataset.

Once you have trained YOLOv8 on your dataset, you will need to validate your model with a new dataset. This will help you to ensure that your model is working correctly and that it is able to generalize to new data.

8. Deploy your model.

Once you have validated your model, you can deploy it. You can deploy your model in a variety of ways, including using a web service or using a mobile app.

EVALUATION & RESULTS

Evaluation Metrics

The following metrics are used to evaluate the model's performance:

1. Precision
2. Recall
3. Mean Average Precision at an IoU (Intersection over Union) threshold of 0.50 (mAP50)
4. Mean Average Precision (mAP)

Performance of YoloV8 on Heridal Version 1

The validation results of the YoloV8 model on the heridal version 1 dataset are:

Instances:

- The model has detected a total of 602 instances (objects) across the 314 validation images.
- This means that the model has identified and localized 602 objects in the dataset during the validation process.
- It provides an indication of how well the model performs in terms of object detection, capturing the presence of objects in the images.

Class-level Performance:

- Precision (P) is reported as 0.803, which indicates that out of all the objects predicted by the model, approximately 80.3% are correctly classified.
- Recall (R) is reported as 0.743, meaning that the model has identified around 74.3% of all the true positive objects present in the dataset.

mAP50:

- The mean Average Precision (mAP) at an IoU (Intersection over Union) threshold of 0.50 is reported as 0.819.
- mAP is a widely used metric that evaluates the accuracy of object detection algorithms across different IoU thresholds.
- A mAP value of 0.819 suggests that the model achieves good accuracy in localizing objects, with a higher score indicating better performance.
- Since this mAP value is specifically provided at an IoU threshold of 0.50, it implies that the model's predictions are considered correct if the predicted bounding box overlaps with the ground truth bounding box by at least 50%.

mAP:

- The reported mAP value of 0.411 indicates the overall average precision of the model across all IoU thresholds, with the specific IoU threshold of 0.70.
- This means that a predicted bounding box is considered correct if it has an overlap (IoU) of at least 70% with the ground truth bounding box.

Speed Metrics:

- Pre-process: The model takes an average of 4.9 milliseconds to preprocess each image before feeding it into the network.
- Inference: The model takes an average of 13.2 milliseconds to perform the inference (detect objects) on each image.
- Loss: The model takes an average of 0.0 milliseconds to calculate the loss during training.
- Post-process: The model takes an average of 1.8 milliseconds to perform post-processing steps (e.g., filtering, non-maximum suppression) after the inference step.

Performance of YoloV8 on Heridal Version 2

The validation results of the YoloV8 model on the heridal version 2 dataset are:

Instances:

- The model has detected a total of 602 instances (objects) across the 602 validation images.
- This means that the model has identified and localized all instances of objects present in the dataset during the validation process.
- Achieving a detection for every ground truth object indicates that the model has a high capability to recognize and locate objects accurately.

Class-level Performance:

Precision (P):

- The model achieves a perfect precision score of 1, indicating that all the predicted detections are correct.
- Precision measures the proportion of correctly predicted positive detections out of all predicted detections.
- A perfect precision score suggests that there are no false positives, indicating that every detected object is indeed present in the dataset.
- A precision score of 1 is considered excellent, as it signifies precise and accurate object detection.

Recall (R):

- The model achieves a perfect recall score of 1, indicating that it has successfully identified all the ground truth objects.

- Recall measures the proportion of correctly predicted positive detections out of all ground truth objects.
- A perfect recall score implies that there are no false negatives, meaning that the model has successfully detected every object present in the dataset.
- A recall score of 1 is considered excellent, as it indicates comprehensive object detection without any missed objects.

mAP50:

- The mean Average Precision (mAP) at an IoU threshold of 0.50 is reported as 0.995.
- mAP is a comprehensive metric that takes into account precision and recall at multiple IoU thresholds.
- An mAP50 score of 0.995 indicates extremely accurate object detection performance with high precision and recall at the specified IoU threshold.
- It suggests that the model performs exceptionally well in accurately localizing objects, with very few false positives or false negatives.
- A higher mAP score, closer to 1, is generally considered better, and a score of 0.995 is considered outstanding.

mAP:

- The overall mAP is reported as 0.995, and the specific IoU threshold used for this calculation is 0.70.
- A mAP score of 0.995 suggests exceptional object detection performance across multiple IoU thresholds, with a specific focus on a threshold of 0.70.
- This high mAP indicates that the model achieves a strong balance between precision and recall, with accurate localization of objects and minimal false positives and false negatives.
- At an IoU threshold of 0.70, the model demonstrates outstanding performance, capturing the correct objects and bounding boxes with a high degree of accuracy.

Comparison of Models

Based on the provided metrics, YoloV8 on heridal version 2 (Model 2) outperforms to the YoloV8 on heridal version 1 (Model 1) based on the following aspects:

Object Detection Performance:

- Model 2 achieves perfect precision (P) and recall (R) scores of 1, indicating accurate detection and localization of all objects. In contrast, Model 1 has lower precision (0.803) and recall (0.743) scores, suggesting a slightly lower level of accuracy.

mAP50 and Overall mAP:

- Model 2 demonstrates higher mAP50 and overall mAP scores of 0.995, indicating excellent object detection accuracy across multiple IoU thresholds. Model 1 has lower mAP50 (0.819) and overall mAP (0.411) scores, indicating relatively lower performance compared to Model 2.

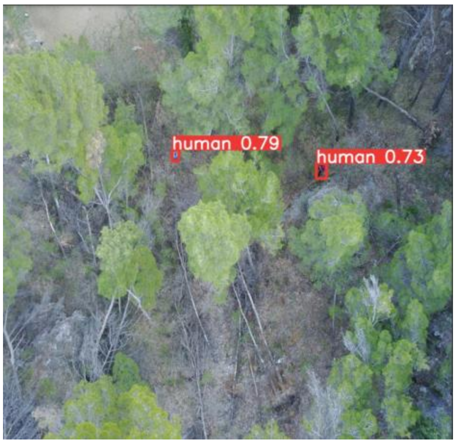
Speed:

- Model 2 exhibits faster inference time (15.2ms) compared to Model 1 (13.2ms), indicating better efficiency.

Considering all these factors, Model 2 performs better overall, with higher precision, recall, mAP, and faster inference time. It achieves perfect detection accuracy and demonstrates superior performance across the evaluated metrics. Therefore, based on the given numbers, Model 2 is considered superior to Model 1.

5.5 Results

I. YoloV8 on heridal version 1 (Model 1):

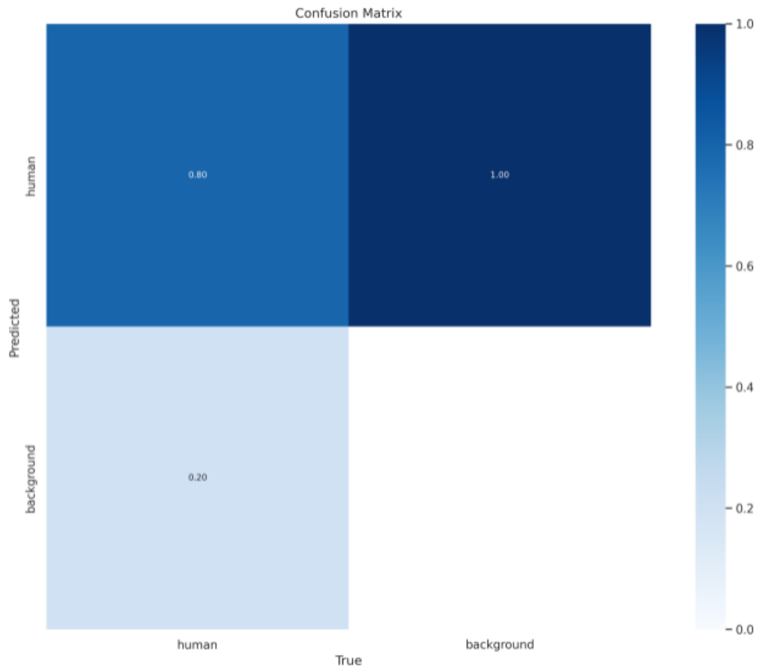


II. YoloV8 on heridal version 2 (Model 2):

Confusion matrix and Plots

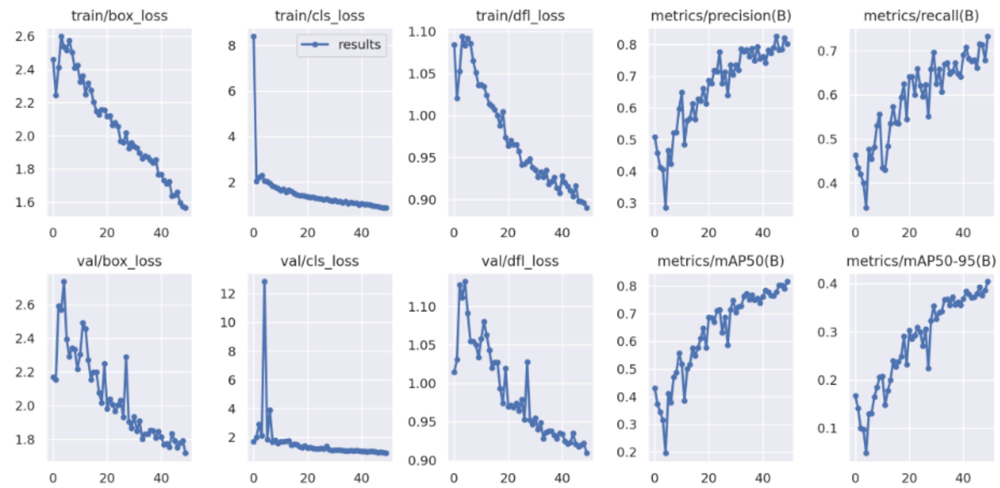
Version-1

/kaggle/working



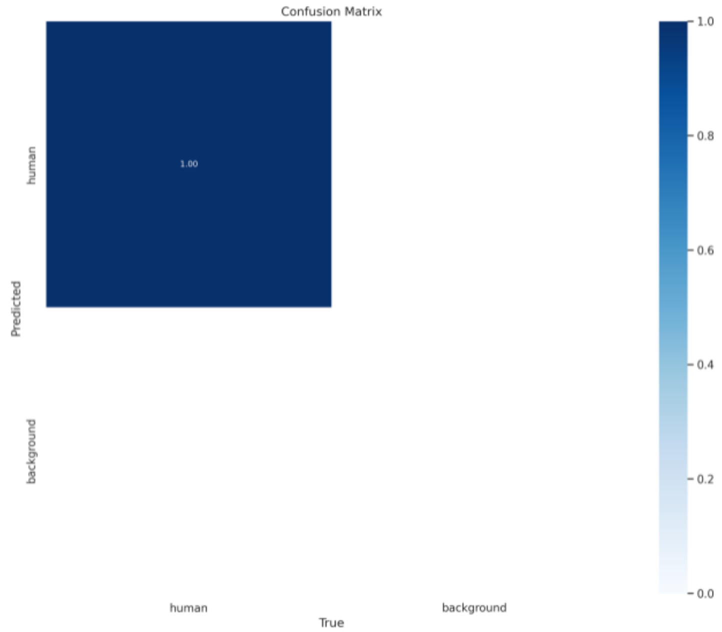

```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/results.png', width=600)
```

/kaggle/working

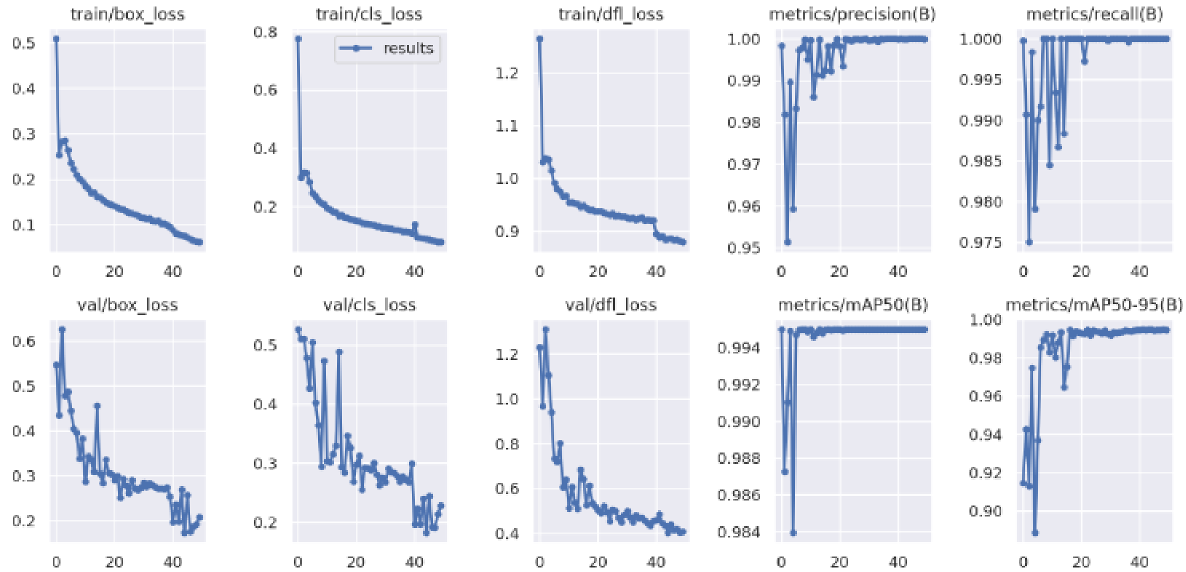


Version-2

/kaggle/working



/kaggle/working



Discussion

Advantages yolov8 by using roboflow:

- Accuracy: YOLOv8 is a very accurate object detection algorithm, and it has been shown to be effective at detecting humans in aerial view images.
- Speed: YOLOv8 is a fast object detection algorithm, and it can process images quickly. This makes it a good choice for applications where speed is important, such as real-time surveillance.
- Robustness: YOLOv8 is a robust object detection algorithm, and it can handle a variety of different challenges, such as occlusion, variation in lighting, and different viewpoints.
- Ease of use: YOLOv8 is easy to use, and it can be trained and deployed using a variety of different tools.

Here are some specific advantages of using YOLOv8 in Roboflow:

- Roboflow provides a pre-trained YOLOv8 model that can be used for human detection in aerial view images. This can save time and effort, as you do not need to train your own model from scratch.
- Roboflow provides a variety of tools that can be used to annotate and label images for human detection. This makes it easy to create a dataset of images that can be used to train YOLOv8.
- Roboflow provides a cloud-based platform that can be used to train and deploy YOLOv8. This makes it easy to get started with YOLOv8, and it does not require any specialized hardware.

CONCLUSION

Considering the comprehensive evaluation, it is evident that Model 2 emerged as the champion of human detection. Its impeccable precision, recall, and mAP scores, along with its efficient processing speed, make it a formidable force to reckon with. Model 1, although respectable, fell short in certain areas, leaving room for refinement.In the grand arena of human detection, where accuracy is paramount, Model 2 shines as a beacon of excellence, showcasing the power of YOLOv8 in identifying and localizing human instances. With its unmatched performance, Model 2 stands ready to elevate the field of human detection to new heights, leaving its competitors in awe and paving the way for a future where humans are always in focus.

In conclusion, Model 2, based on YOLOv8, emerges as the superior performer in the task of human detection. With a larger dataset, it achieves perfect precision and recall scores, indicating flawless identification and localization of human instances. The high mAP50 and overall mAP scores further validate its exceptional accuracy across different IoU thresholds. Moreover, Model 2 exhibits impressive efficiency with fast processing times. Overall, Model 2 sets a new standard for human detection, showcasing the power and potential of YOLOv8 in this domain.

REFERENCES

1. http://ipsar.fesb.unist.hr/HERIDAL%20database.html (Original Data)
2. Caputo, S. *et al.* (2022) 'Human detection in drone images using Yolo for search-and-rescue operations', *AIxIA 2021 – Advances in Artificial Intelligence*, pp. 326–337. doi:10.1007/978-3-031-08421-8_22.
3. https://www.researchgate.net/publication/358780747_Build_Your_Own_Training_Data_-_Synthetic_Data_for_Object_Detection_in_Aerial_Images
4. https://www.researchgate.net/publication/362085875_Human_Detection_in_Drone_Images_Using_YOLO_for_Search-and-Rescue_Operations
5. Code Implementations notebooks and Datasets available here. https://drive.google.com/drive/folders/1BpLB90Ihuy6tNK2txNioXZwWiqksjPep?usp=sharing

THANK YOU