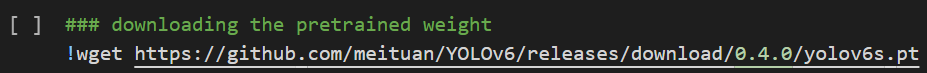
Object Detection Using YOLO and Overlaying of Tattoo

Aim - to overlay an external object onto the hand of a person using object detection frameworks like YOLOv5-7.

Framework chosen for object detection - YOLOv6. The rationale behind using this OD framework is that it is one of the most stable versions in YOLO series, thoroughly tested and improved over the past 1.5 years. The Github repository has seen 3 major changes (current version 0.4.0) and improvements after once it was launched.

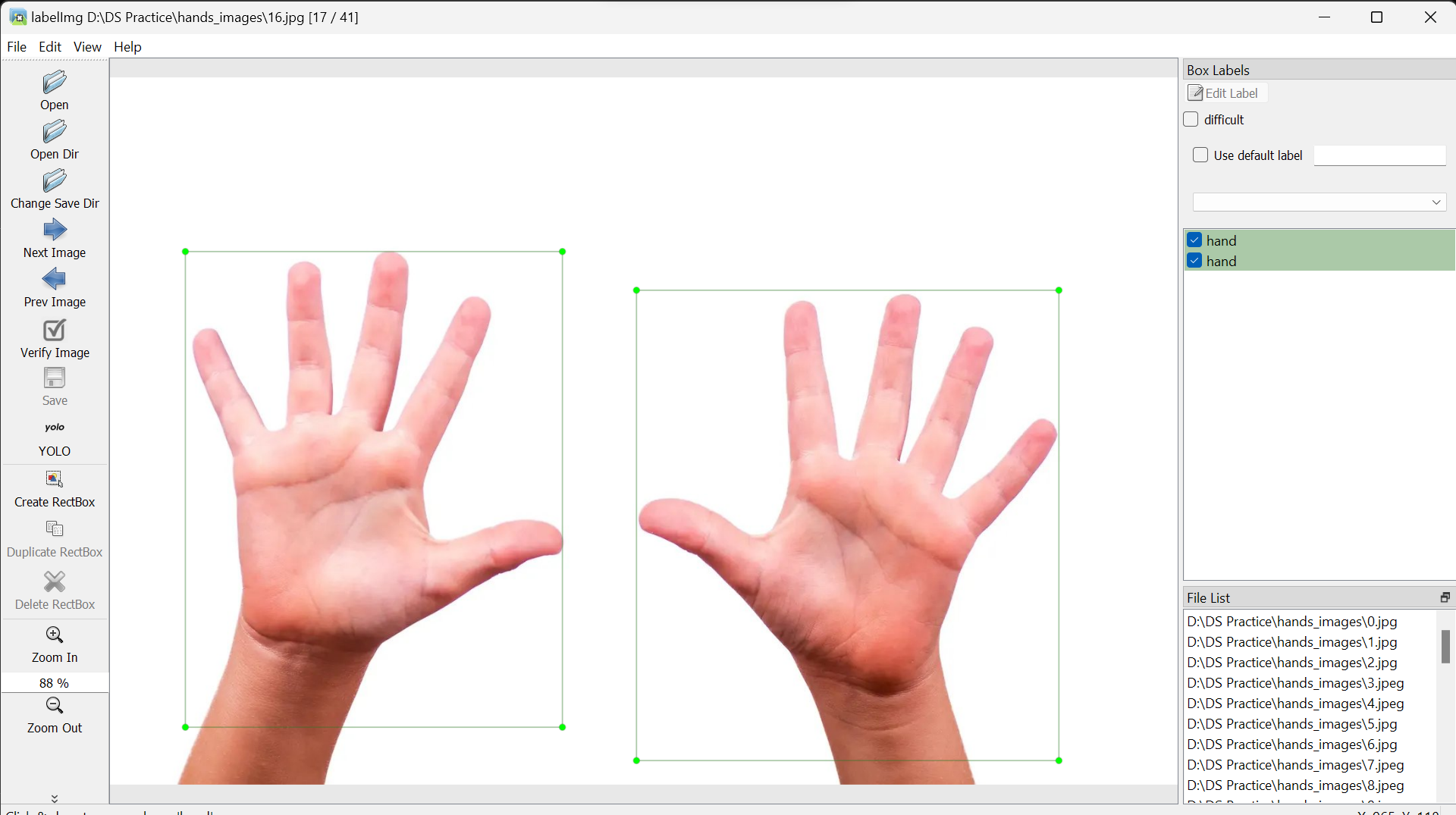


Object chosen for overlay - a png image of a tattoo with transparent background.

Model development and training environment - Google Colab, used to leverage the free GPU runtime using T4 GPU for a quicker model training.

Dataset - collected 41 images of hands of varying shapes, sizes, and colour, of different age groups, to bring in variety into the dataset for a more stable and generalised model training.

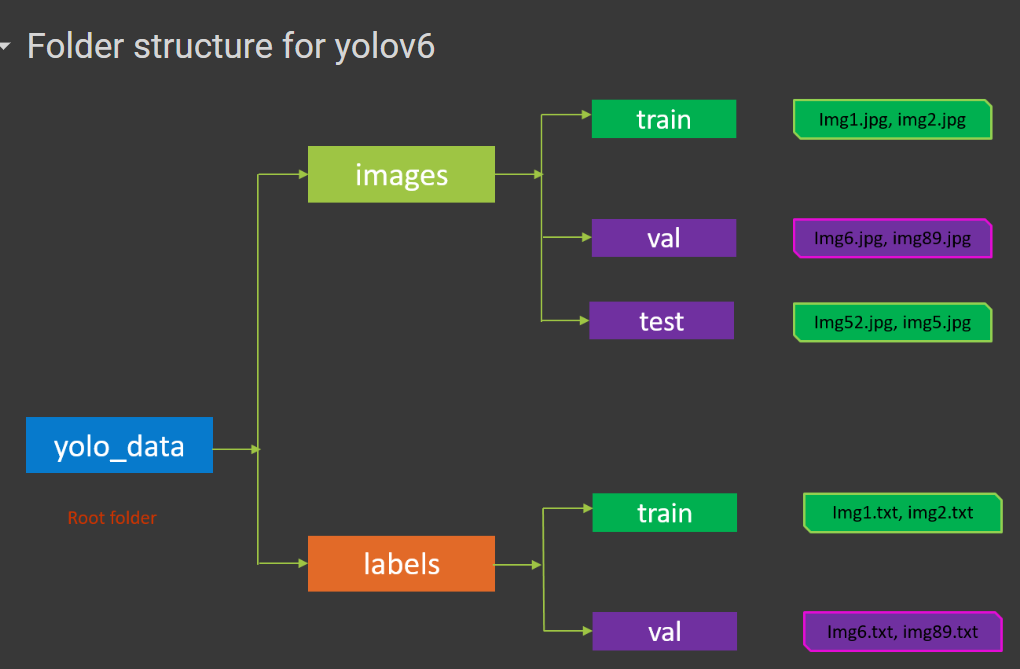
Image annotation - images were annotated using Labelimg, which is a python application which can be installed externally as a package. The class name chosen is 'hand'.



The model was trained on 100 epochs, with a batch size of 4, from which the best weights file was exported with the name 'best\_ckpt.pt' (file size: 38 Mb).



For model training, the images dataset was split into training (80%) and validation (20%).



The class name was changed in the **'dataset.yaml'** file to 'hand'.

A computer screen shot of a program

Description automatically generated

The pretrained weights downloaded for model training was yolov6s.pt where 's' stands for small. The smaller version of YOLOv6 is a bit less accurate but much faster while making inferences than the larger versions.

The tattoo overlaying pipeline was developed using VSCode, by cloning the YOLOv6 repository into the project environment. The pipeline is created in the **‘run.ipynb’** ipython notebook.

A screenshot of a computer program

Description automatically generated

The project repository structure

A screen shot of a computer program

Description automatically generated

Importing all the required libraries

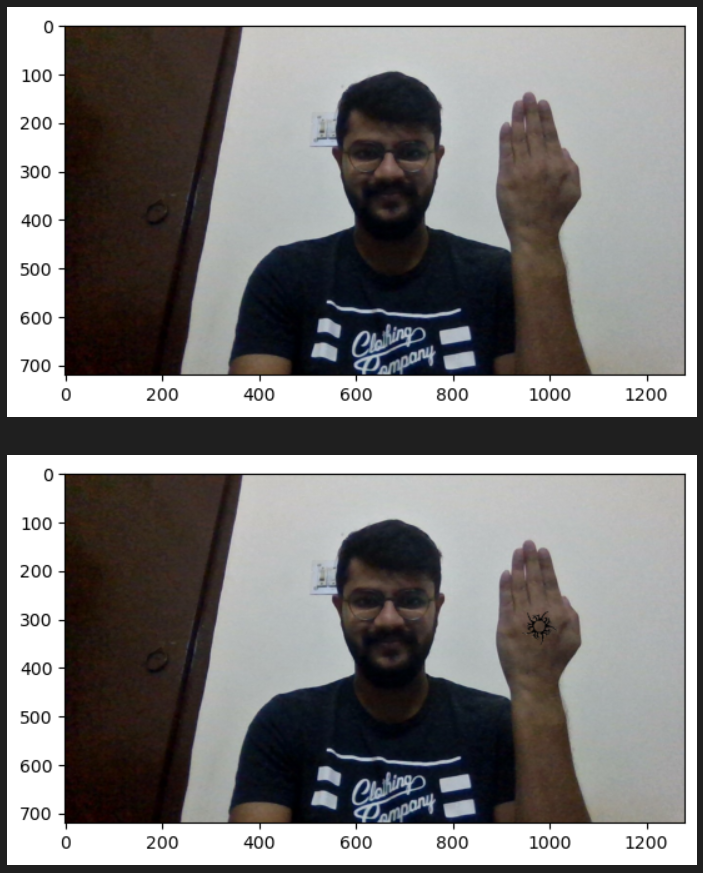
A screenshot of a computer screen

Description automatically generated

PNG image of the tattoo chosen



Executing the final function



The final output on left hand

A person holding his hand up

Description automatically generated

The final output on right hand

Conclusion – given sufficient time, the pipeline can be further fine-tuned and made more robust for better object detection and tattoo overlaying on the hand area. Adding more images to the dataset should improve the accuracy of the OD model.

Completed and submitted by –

Ujjawal Roy (Senior Associate Data Scientist)

Ph. No. - +919321398895