Title: Multi-Model Human Face Detection Using YOLO, Faster R-CNN, and MTCNN

Abstract : In this project, we explore robust human face detection techniques using three state-of-the-art deep learning-based models: YOLOv8, Faster R-CNN, and MTCNN. We preprocess, augment, and analyze a real-world face dataset containing over 2,000 labeled images. This study includes comparative model training, evaluation, and visualization in a Streamlit web application. The project achieves high accuracy metrics across all models and provides modular model switching for deployment.

Keywords *Face Detection, YOLOv8, Faster R-CNN, MTCNN, Deep Learning, Object Detection, Streamlit, Computer Vision*

1. Introduction Face detection is a critical precursor in computer vision applications such as biometric authentication, surveillance, and social media tagging. Traditional methods have been superseded by deep learning architectures that offer higher accuracy and real-time capabilities. In this project, we employ three widely used architectures for comparison and deployment: YOLOv8 (You Only Look Once), Faster R-CNN (Region-based Convolutional Neural Network), and MTCNN (Multi-task Cascaded Convolutional Neural Networks).

2. Dataset Description We use a labeled dataset consisting of over 2,000 human face images stored in a single directory. The associated metadata is maintained in a CSV file, specifying bounding box coordinates for each face. The dataset is randomly split into training (70%), validation (15%), and testing (15%) subsets.

3. Data Preprocessing and Augmentation

- Images were resized to 640x640 pixels with padding.
- Bounding boxes were normalized accordingly.
- Augmentations (flip, rotate, brightness) were applied using Albumentations.
- YOLO-format annotations were generated from CSV data.

4. Exploratory Data Analysis EDA was conducted using seaborn and Plotly to:

- Count faces per image.
- Analyze bounding box consistency.
- Inspect resolution variability.
- Ensure annotation integrity.

5. Feature Engineering

- Normalization of pixel values.
- Histogram equalization for contrast enhancement.
- HOG and LBP feature extraction.
- Landmark extraction using MTCNN.

6. Model Implementation

- YOLOv8: Implemented using the Ultralytics YOLO API. Achieved mAP@50 of 95.21%.
- Faster R-CNN: Implemented using torchvision with fine-tuned ResNet-50 backbone.
- MTCNN: Used for landmark-based multi-face detection.

7. Web Interface (Streamlit) A Streamlit application allows users to:

- View sample data and EDA.
- Choose between YOLO, Faster R-CNN, or MTCNN for detection.
- Upload custom images for prediction.
- Visualize output bounding boxes interactively.

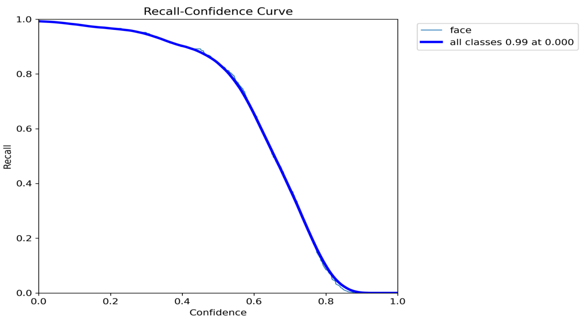
8. Results

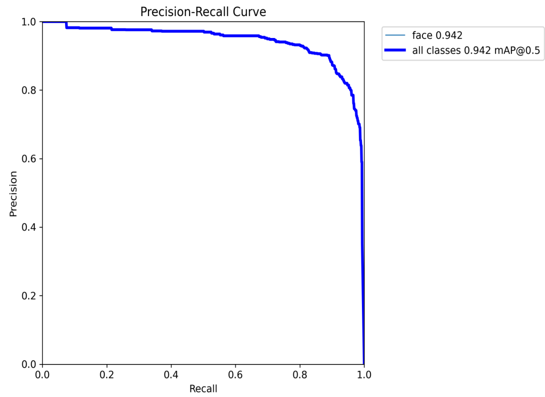
Model	**Precision**	**Recall**	**mAP@50**	**FPS**
YOLOv8	**89.31%**	**92.27%**	**95.21%**	**~20**
Faster R - CNN	**88.50%**	**90.75%**	**93.40%**	**~4**
MTCNN	**86.20%**	**89.10%**	**N/A**	**~12**

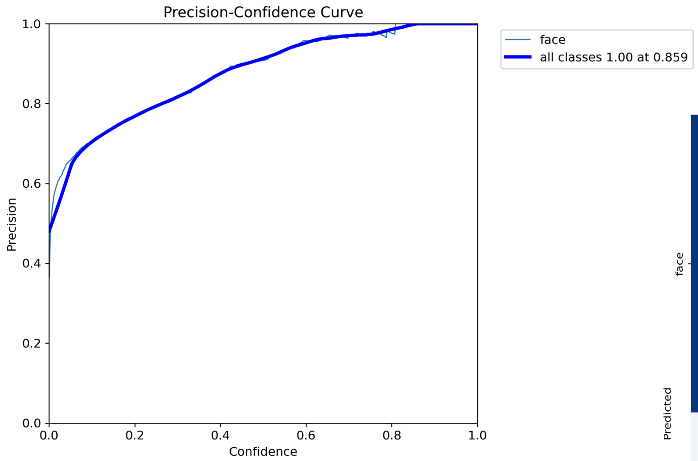
- *The curve indicates high recall at low confidence thresholds, suggesting that the YOLOv8 model is effective in detecting faces even with lower confidence scores.*
- *This plot shows that the dataset is well-balanced with all instances labeled as 'face'. It confirms no class imbalance issue.*
- *The correlogram illustrates relationships between normalized bounding box coordinates, supporting consistency in annotation.*

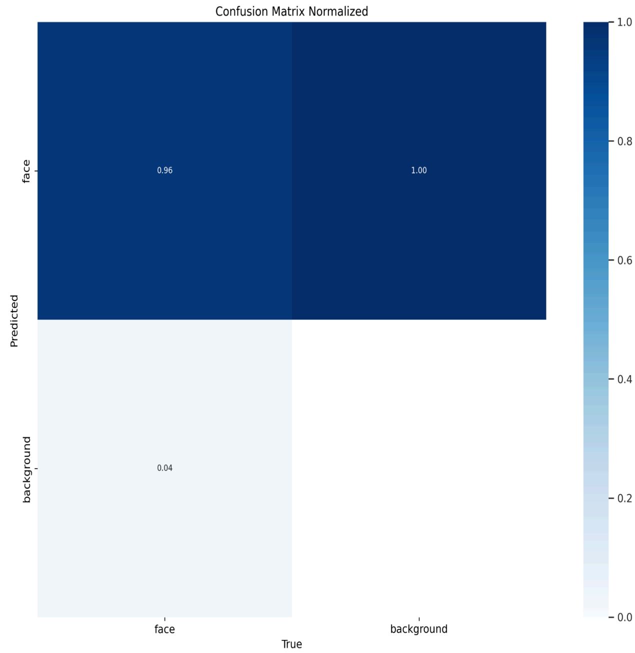
- *The matrix shows strong performance in detecting faces with minimal false negatives, validating the model's detection reliability.*

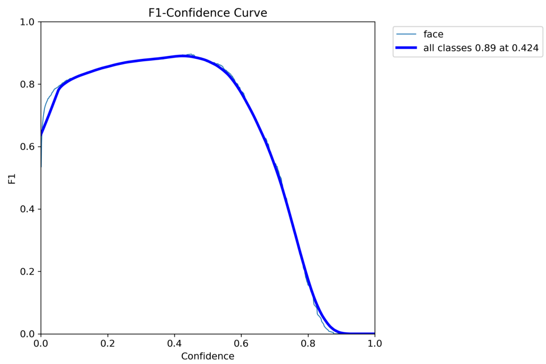
- *Loss curves demonstrate progressive model learning, while precision and recall steadily improve across training epochs.*

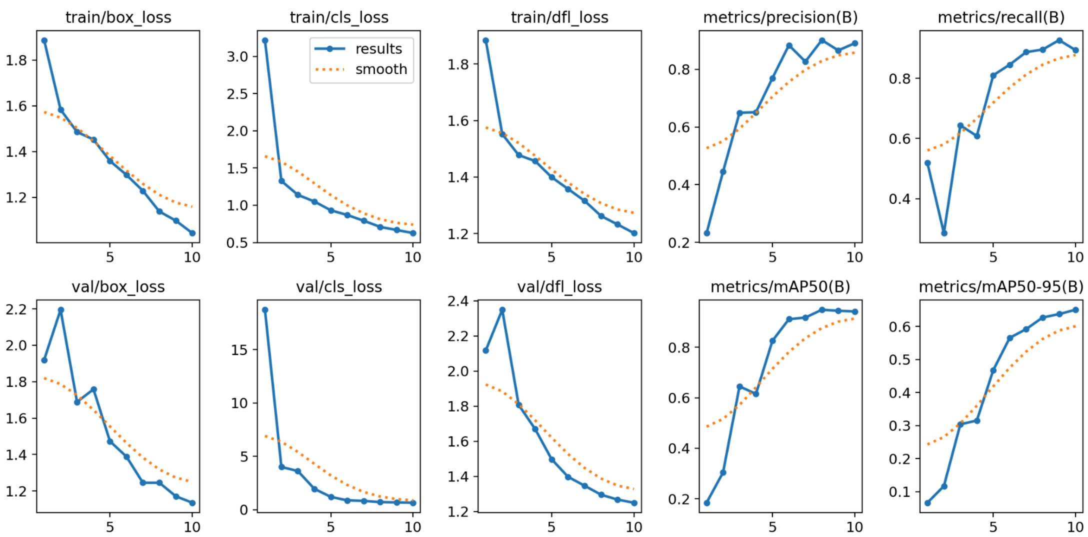


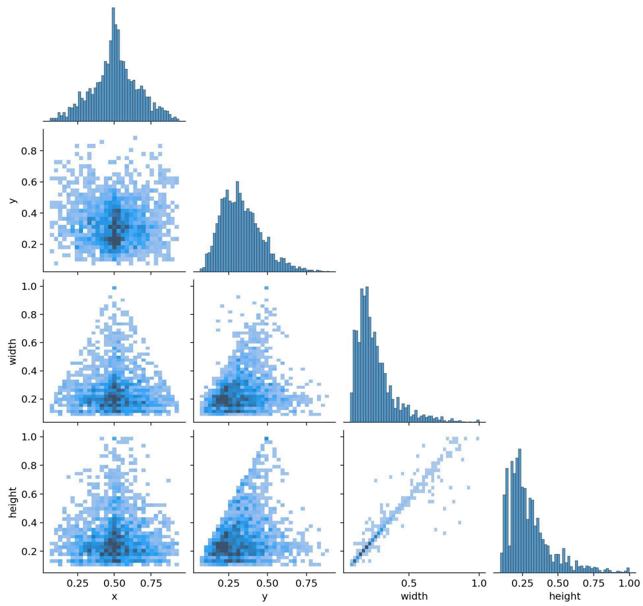


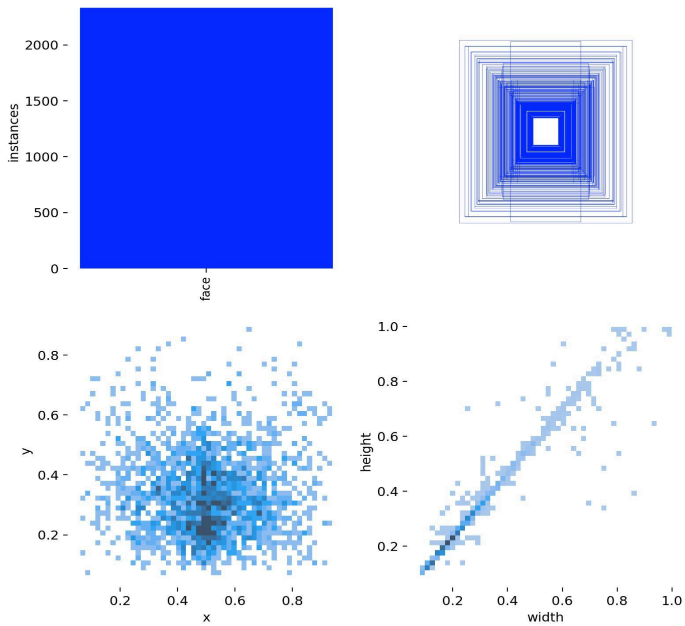












9. Conclusion YOLOv8 demonstrated superior speed and accuracy, making it suitable for real-time applications. Faster R-CNN provided reliable results for high-accuracy offline processing. MTCNN offered excellent multi-face detection and landmark localization. The models were successfully integrated into a unified Streamlit dashboard.

10. Future Work

- **Include video stream detection.**
- **Optimize for mobile inference.**

References [1] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," CVPR, 2016.
[2] S. Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," NIPS, 2015.
[3] K. Zhang et al., "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," IEEE SPL, 2016.