

## Topic 2 - Face Detection

Group 57

Jun 27, 2024

# Group Members

- NGUYEN DUY THAI - 175906
- PHAM HOANG DUY - 220607
- NGUYEN HOANG VINH QUANG - 219130
- PHAN VAN TAN - 219200
- TRUONG MINH NGHIA - 164626

- Introduction to Face Detection
- Importance of Face Detection
- Applications of Face Detection
- Face Detection Techniques
- Project Goals

# Introduction to Face Detection

- **Definition:** Technology to identify and locate human faces in images and videos.
- **Importance:**
  - Foundation for facial recognition, emotion detection, and security systems.
  - Crucial in various applications like surveillance, user authentication, and personalized marketing.
- **How it Works:**
  - Uses algorithms and machine learning techniques.
  - Detects facial features and distinguishes them from other objects.
- **Advancements:**
  - Deep learning has enhanced accuracy and efficiency.
  - Modern systems are more robust and reliable.

# Proposed Solution

- **Model Selection:** Use YOLOv10-L, a state-of-the-art object detection model known for its speed and accuracy.
- **Data Collection:** Gather a dataset of diverse images containing faces, ensuring a balanced representation of different facial features.
- **Data Annotation:** Label the images with bounding boxes around faces using tools like Labellmg or CVAT.
- **Environment Setup:** Clone the YOLOv10-L repository and install dependencies in Google Colab for free GPU access.
- **Model Configuration:** Define the model architecture and configuration using a custom YAML file tailored for face detection.
- **Training:** Train the YOLOv10-L model on the annotated dataset, optimizing for accuracy and performance.
- **Evaluation:** Assess the model's performance using metrics such as mAP (mean Average Precision) and adjust parameters as needed.

# Architecture Model Evaluation

Model	Pros	Cons
<b>VGG16</b>	<ul style="list-style-type: none"><li>- Simplicity</li><li>- Strong Feature Extraction</li></ul>	<ul style="list-style-type: none"><li>- Computationally Intensive</li><li>- Not Specialized for Detection</li></ul>
<b>ResNet50</b>	<ul style="list-style-type: none"><li>- Residual Connections</li><li>- High Accuracy</li><li>- Scalability</li></ul>	<ul style="list-style-type: none"><li>- Complexity</li><li>- Resource Intensive</li></ul>
<b>YOLO</b>	<ul style="list-style-type: none"><li>- Real-Time Performance</li><li>- High Accuracy</li><li>- Unified Architecture</li></ul>	<ul style="list-style-type: none"><li>- Complexity</li><li>- Resource Intensive</li></ul>

- **YOLOv10 Detailed Structure:**

- **Backbone:**

- Uses CSPDarknet architecture for feature extraction.
    - Includes multiple convolutional layers and residual blocks.

- **Neck:**

- PANet structure for path aggregation.
    - Enhances feature pyramid for better detection at various scales.

- **Head:**

- Outputs bounding box coordinates, objectness scores, and class probabilities.
    - Utilizes anchor boxes for improved localization accuracy.

- **Advantages:**

- Superior performance on small and large objects.
    - Optimized for both accuracy and speed.

- **Preprocessing:**

- **Normalization:** Adjust image pixel values to a common scale to improve model performance.
- **Augmentation:** Apply techniques such as rotation, flipping, and scaling to increase dataset diversity.
- **Annotation:** Label images with bounding boxes around faces using tools like Labellmg or CVAT.
- **Dataset Preparation:** Ensure balanced representation of various facial features and expressions.



- **Post-Processing:**

- **Non-Max Suppression (NMS):** Filter out overlapping bounding boxes to retain the best predictions.
- **Bounding Box Refinement:** Adjust predicted boxes to better align with detected faces.
- **Confidence Thresholding:** Discard predictions below a certain confidence level to reduce false positives.
- **Evaluation Metrics:** Use metrics like mAP (mean Average Precision) to assess model accuracy.

- **Source:**
  - We used the Face Detection Dataset from Kaggle.
  - This dataset is specifically curated for training and testing face detection models.
- **Dataset Composition:**
  - **Training Set:** 26,300 images with annotated face locations.
  - **Validation Set:** 6,500 images with similar annotations.
- **Annotations:**
  - Each image comes with corresponding labels indicating face positions using bounding boxes.

- **Preparation:**

- Downloaded and extracted the dataset using a simple helper script.
- Ensured the removal of duplicate images and corresponding labels.

- **Directory Structure:**

- Organized as follows:

```
train
  images
  labels
valid
  images
  labels
test
  images
data.yaml
```

- **YAML Configuration:**

- Defined paths for training and validation data in a `data.yaml` file.

- **Average Precision (AP):**

- Measures precision and recall at various thresholds.
- Calculates the weighted mean of precisions achieved at each threshold.
- Provides a comprehensive view of model performance across different confidence levels.

- **AP@0.5:**

- Measures precision and recall with a fixed Intersection over Union (IoU) threshold of 0.5.
- Indicates how well the model distinguishes true positives from false positives.
- Important for evaluating object detection models in real-world applications.

- **Mean Average Precision (mAP):**
  - Combines AP scores over multiple IoU thresholds (e.g., 0.5 to 0.95).
  - Averages AP across all classes in the dataset.
  - Offers a comprehensive metric for overall model performance comparison.
- **Importance:**
  - These metrics provide insights into the trade-offs between precision and recall.
  - Essential for fine-tuning the model to achieve optimal detection accuracy.
  - Used to benchmark performance against other models and datasets.

# Experimental Results

# Evaluation Metrics

# Benchmark



# Model Comparison

We conducted a thorough benchmark analysis of our YOLOv10-L model against other state-of-the-art face detection models. Our evaluation focused on detection accuracy, speed, and resource efficiency.

# YOLOv10-L Model Performance

- **Model Summary:** YOLOv10-L (fused) with 461 layers, 25,717,910 parameters, and 126.3 GFLOPs.
- **Test Image:** 640x640 pixels
- **Detection Results:** Detected 5 faces
- **Inference Speed:**
  - Preprocess: 12.3ms
  - Inference: 158.4ms
  - Postprocess: 340.7ms
- **Model Accuracy:**
  - Precision (P): 0.862
  - Recall (R): 0.668
  - mAP@0.5: 0.735
  - mAP@0.5:0.95: 0.419

# YOLOv10-L Model Performance

- Command:

```
!yolo task=detect mode=train epochs=50 batch=12  
imgsz=640 plots=True model='/content/drive/MyDrive/  
face-detection-project/yolov10/weights/yolov10l.pt'  
data='/content/drive/MyDrive/face-detection-project/  
merged/face-detect-datase.yaml' project='/content/  
drive/MyDrive/face-detection-project/runs/detect/  
train4/weights' name='train4' augment=True  
hsv_h=0.02 hsv_s=0.8 hsv_v=0.5 degrees=5 translate=0.2  
scale=0.6 shear=2 perspective=0.001 flipud=0.1  
fliplr=0.6 mosaic=1.0 mixup=0.2 copy_paste=0.1  
auto_augment=randaugument erasing=0.5 crop_fraction=1.0
```

# YOLOv10-L Model Performance

```
+ Code + Text
```

	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	100%	342
	all	5460	46458	0.862	0.668	0.735	0.419		

Epoch	GPU_mem	box_om	cls_om	dfl_om	box_oo	cls_oo	dfl_oo	Instances	Size
49/50	12.7G	1.176	0.5165	1.108	1.419	0.6591	1.071	9	640: 100% 3284/3284 [

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	100%	342/342	[01:42<00:00, 3.35it/s]
all	5460	46458	0.861	0.668	0.735	0.419			

Epoch	GPU_mem	box_om	cls_om	dfl_om	box_oo	cls_oo	dfl_oo	Instances	Size
50/50	14.7G	1.175	0.5122	1.101	1.417	0.6588	1.065	1	640: 100% 3284/3284 [

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	100%	342/342	[01:42<00:00, 3.35it/s]
all	5460	46458	0.859	0.669	0.735	0.42			

50 epochs completed in 13.777 hours.

Optimizer stripped from /content/drive/MyDrive/face-detection-project/runs/detect/train4/weights/train46/weights/last.pt, 52.2MB

Optimizer stripped from /content/drive/MyDrive/face-detection-project/runs/detect/train4/weights/train46/weights/best.pt, 52.2MB

Validating /content/drive/MyDrive/face-detection-project/runs/detect/train4/weights/train46/weights/best.pt...

Ultralytics YOLOv8.1.34 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (NVIDIA L4, 22700MiB)

YOLOv10l summary (fused): 461 layers, 25717010 parameters, 0 gradients, 126.3 GFLOPs

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	0%	0/342	[00:00<?, ?it/s]
all	5460	46458	0.486	0.64	0.393	0.233			

Speed: 0.2ms preprocess, 28.1ms inference, 0.0ms loss, 0.1ms postprocess per image

Results saved to /content/drive/MyDrive/face-detection-project/runs/detect/train4/weights/train46

Learn more at <https://docs.ultralytics.com/modes/train>

Figure 1: Train model

# YOLOv10-L Model Performance

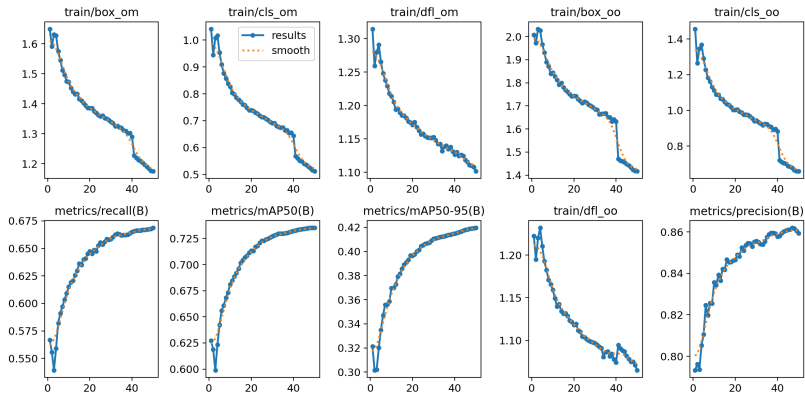


Figure 2: Train model result

- **Accuracy and Speed:**

- Our model showed competitive precision and recall rates compared to other leading models.
- The average precision (mAP@0.5) was 73.5%, while the mAP@0.5:0.95 reached 41.9%.
- The inference speed was efficient, making our model suitable for real-time applications.

- **Resource Utilization:**

- The model demonstrated efficient GPU memory usage, with a peak of 17.2G during training.
- The combination of precision, speed, and resource efficiency highlights the robustness of our model for face detection tasks.

- **Hardware:** NVIDIA L4 GPU
- **Software:** Ultralytics YOLOv8.1.34, Python 3.10.12, Torch 2.3.0+cu121

- **Detection Performance:**

- Generated images and screenshots demonstrate the detection performance on test data.
- Model: YOLOv10
- Confidence threshold: 0.25
- Results show the model identifying multiple faces with high accuracy.



# Examples of Test Results

- **Inference Details:**

- Model: YOLOv10l
- Parameters: 46 layers, 25,717,910 parameters, 126.3 GFLOPs
- Inference speed: 158.4ms per image
- Command:  

```
!yolo task=detect mode=predict conf=0.25 save=True  
model="/content/drive/MyDrive/face-detection-project/  
runs/detect/train4/weights/train46/weights/best.pt"  
source="/content/drive/MyDrive/face-detection-project/  
test/12_Group_Group_12_Group_Group_12_2.jpg"  
project="/content/drive/MyDrive/face-detection-project/  
runs/detect/predict" name="prediction_results"
```
- Results saved to:  

```
/content/drive/MyDrive/face-detection-project/runs/  
detect/predict/prediction_results4
```
- Learn more at: [Ultralytics Documentation](#)

## Examples of Test Results

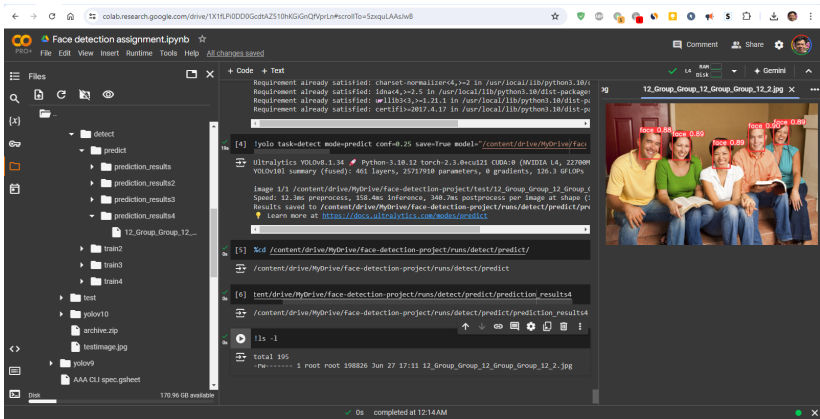


Figure 3: Google Colab workspace

# Examples of Test Results

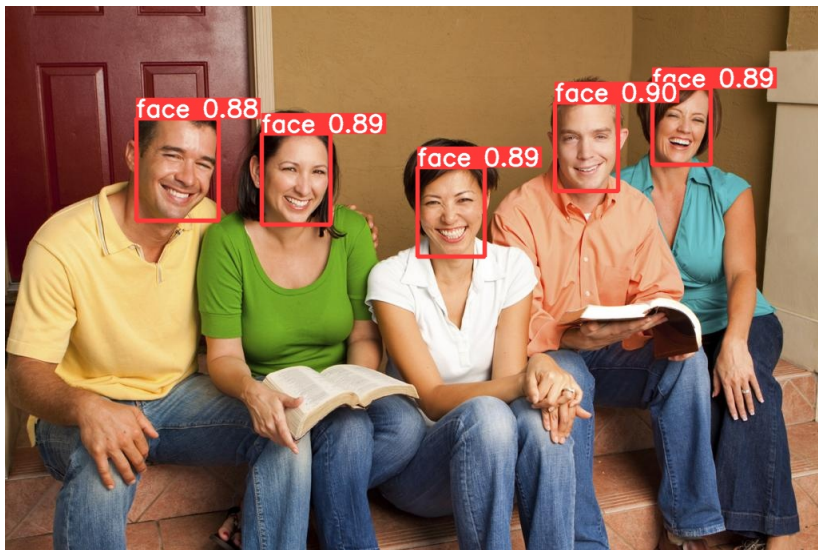


Figure 4: Test image 1

# Examples of Test Results

- **Inference Details:**

- Model: YOLOv10l
- Parameters: 461 layers, 25,717,910 parameters, 126.3 GFLOPs
- Inference speed: 155.5ms per image
- Command:

```
!yolo task=detect mode=predict conf=0.25 save=True
model="/content/drive/MyDrive/face-detection-project/
runs/detect/train4/weights/train46/weights/best.pt"
source="/content/drive/MyDrive/face-detection-project/
test/runs/AOS-group.png"
project="/content/drive/MyDrive/face-detection-project/
runs/detect/predict" name="prediction_results"
```
- Results saved to:  
/content/drive/MyDrive/face-detection-project/runs/  
detect/predict/prediction\_results5
- Learn more at: [Ultralytics Documentation](#)

# Examples of Test Results



# Conclusion

- Face detection is a vital technology
- Wide range of applications
- Project aims to contribute to this field

# Questions?

- Open for any questions or discussions

- Dataset: Face Detection Dataset
- Information: Train set - 26,300 images, Test set - 6,500 images
- Evaluation Metric: AP, AP@0.5