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

Overview

- 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 LabelImg 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
VGG16	- Simplicity - Strong Feature Extraction	- Computationally Intensive - Not Specialized for Detectio
ResNet50	- Residual Connections - High Accuracy - Scalability	- Complexity - Resource Intensive
YOLO	- Real-Time Performance - High Accuracy - Unified Architecture	- Complexity - Resource Intensive

Architecture

- **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.

Data Processing

- **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 LabelImg or CVAT.
  - **Dataset Preparation**: Ensure balanced representation of various facial features and expressions.

Data Processing

- **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.

Dataset

- **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.

Dataset

- **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:
- **YAML Configuration**:
  - Defined paths for training and validation data in a `data.yaml` file.
  - Included class names and counts for model reference.

Evaluation Metrics

- **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.

Evaluation Metrics

- **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

- **VGG16**: AP, AP@0.5 results
- **ResNet50**: AP, AP@0.5 results
- **YOLO**: AP, AP@0.5 results

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**: 448x640 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

Comparison with Other Models

- **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.

Test Environment

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

Examples of Test Results

- **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
  - Example 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

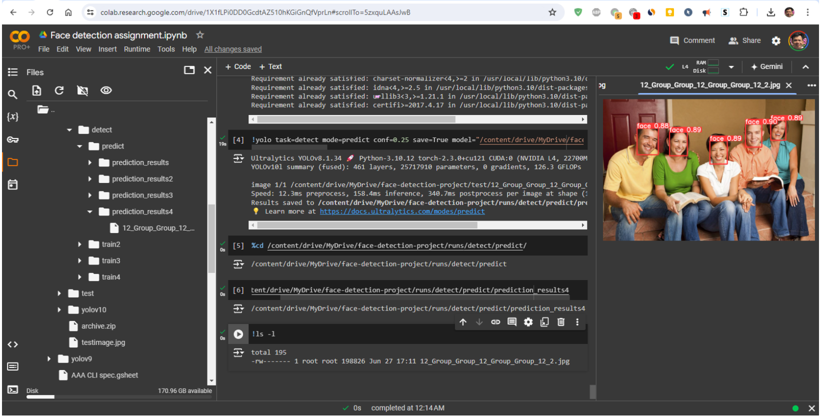
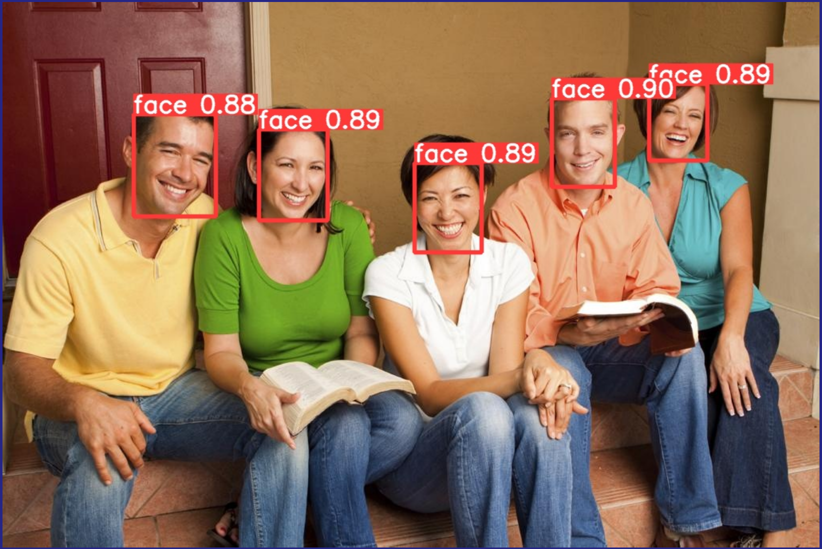


Figure 1: Google Colab workspace

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

References

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