# Comparative Evaluation of Advanced Face Detection Models

**Sotirios Damas**
Concordia University
Montreal, QC, Canada

**Visakan Nambirajan**
Concordia University
Montreal, QC, Canada

## Abstract

Face detection is a key component of computer vision and serves as the foundation for many different applications, such as augmented reality, security systems, camera-based applications, biometric authentication, and embedded systems. Over the years, the field has progressed from simple feature matching techniques to complex deep learning models that provide high accuracy and efficiency. Despite these advancements, real-world factors such as occlusions, changing lighting, clarity, and positional diversity hamper the performance of these models. The main aim of this project is to thoroughly assess the most advanced face detection models: *Retina Face, MTCNN, YOLO, and OpenCV*. By using the benchmark dataset *WIDER FACE*, a custom subset of images is created that are split based on specific image properties. The model's performances are analyzed across critical metrics such as precision, recall, and inference speed. This will offer useful insights into the advantages and disadvantages of these models, determining their applicability for certain scenarios. The results are meant to help practitioners choose the best face detection models for their particular use case and support ongoing research to further improve the robustness and effectiveness of these models.

## 1 Literature Review

Over several decades, face detection technology has undergone many distinct phases of evolution, each distinguished by improvements in computational power and methodologies. This review highlights significant innovation and their effects on the field.[213] It highlights the key breakthroughs of face detection techniques over time.

### 1.1 1960 - 1980: Feature Mapping Techniques

Manual or semi-automated feature mapping techniques were a major component of early face detection algorithms. These techniques used geometric mappings to identify facial characteristics such as the mouth, nose, and eyes. Although they established the foundation for automatic face detection, these methods were extremely sensitive to changes in facial structure and posture and required a significant amount of manual input. Their precision and scalability were further constrained by a lack of processing resources. [1]

### 1.2 1990-2000: Template Matching and Statistical Models

The introduction of template-based techniques in the 1990s allowed for the analysis of correlations between facial features using pre-made templates. Some of the notable advancements include the following.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

**Eigenfaces** This technique, which was first presented as a dimensionality reduction strategy utilizing *Principal Component Analysis (PCA)*, successfully recorded fluctuations in facial features for face detection and recognition. It was among the first applications of statistical models in the field.[3]

**Feature-Based Techniques** These enhanced the detection's resilience by examining the spatial correlations between important face landmarks and matching them with preset templates. However, these models have difficulty with occlusion, lighting, and variations in facial expressions.

### 1.3 2000 – 2015: Machine Learning

Machine learning entered the picture at the start of the new millennium. Important turning points include the following.

**Viola Jones Algorithm** With the introduction of *Haar*-like features and an *AdaBoost*-based cascade classifier, this revolutionized real-time face detection. It was widely used in real-world applications, such as digital cameras and surveillance systems, because of its speedy and somewhat accurate face detection capabilities.[4,11]

**Support Vector Machine (SVM)** To increase the accuracy and resilience of frontal face detection, *Max-Margin Object Detection (MMOD)* integrated *SVMs* with enhanced feature extraction methods. Although these techniques improved detection skills, their reliance on manually created features limited their ability to be applied to intricate real-world situations.[5]

### 1.4 2015 – Present: Deep Learning and Convolutional Neural Networks

Deep learning revolutionized face detection by automating feature extraction through *Convolutional Neural Networks (CNNs)*. Key advancements in this era include the following.

**Muti-Task Cascaded Neural Networks (MTCNN)** The *MTCNN* model combines facial landmark localization and face detection in a multi-step procedure. It has established a standard in the sector thanks to its exceptional precision in handling occlusions, stance changes, and lighting situations.[6,14]

**Single Shot Multi-Box Detector (SSD)** *SSD* introduced a single-pass architecture that detects objects at multiple scales using convolutional feature pyramids. It achieves real-time speed and good accuracy, making it ideal for applications requiring efficiency. However, it struggles with detecting small or occluded faces in complex scenes.[7]

**You Only Look Once (YOLO)** *YOLO* simplified detection by predicting bounding boxes and class probabilities in a single evaluation, enabling fast inference. *YOLOv5*, tailored for face detection, balances speed and accuracy, making it suitable for real-time applications. However, it may face challenges with small or tightly packed faces due to its grid-based prediction approach.

**Feature Pyramid Models** These models are useful in busy and complex contexts because they combine contextual information with multi-scale feature extraction to improve the recognition of small and obscured faces. *Pyramid-Box* and *RetinaFace* are models that use this architecture.[8,9]

### 1.5 2020s: Efficiency and Scalability

In recent years, there has been a focus on improving models for scalability and efficiency in practical situations. The important advancements are what follows.

**Scale-Invariant Models (S3FD)** These models, which are made to deal with faces of different sizes and distances, have proven to have strong detection abilities in uncontrolled settings.

**Distributed Architectures** These allow for real-time deployment in environments with limited resources by enhancing inference speed and lowering computing overhead.[10]

### 1.6 Challenges in the Current Landscape

**Occlusion and Background Complexity** When faces are partially hidden or displayed against crowded backdrops, the detection accuracy often suffers.

**Generalization** Although many models perform well on benchmark datasets, they behave poorly in a variety of real-world scenarios.

**Real-Time Applications** For applications that demand low-latency responses, striking a balance between inference speed and detection accuracy continues to be a crucial problem.

## 2 Methodology

### 2.1 Dataset Description

For this study, the *WIDER FACE* dataset was used as the primary benchmark to evaluate the performance of face detection models. This dataset is widely recognized for its diversity, containing images with variations in face count, occlusion, blur, pose, and illumination. The validation set was used for this project, as the testing subset does not contain annotations for the ground truths and the image features. This facilitated the direct evaluation and comparison of the performance of different models.

#### 2.1.1 Splitting Criteria

The validation set was further split into subsets based on different image characteristics to enable a detailed analysis of model performance:

- **Splitting based on the number of faces:**
  - *Easy:* Images containing only 1 face.
  - *Medium:* Images containing 2 to 6 faces.
  - *Hard:* Images containing more than 6 faces.
- **Splitting based on Occlusion:**
  - *0 (Easy):* No occlusion.
  - *1 (Medium):* Partial occlusion.
  - *2 (Hard):* Significant occlusion, where the majority of the face is obscured.
- **Splitting based on Blur:**
  - *0 (Easy):* Sharp and clear images.
  - *1 (Medium):* Moderately blurry images.
  - *2 (Hard):* Heavily blurry images.
- **Splitting based on Illumination:**
  - *0 (Normal):* Images with balanced lighting.
  - *1 (Extreme):* Images with poor and high-contrast lighting conditions.

#### 2.1.2 Purpose of Splitting

A thorough assessment of the models' performance under specific and challenging circumstances is made possible by these subcategories. This research provides a comprehensive understanding of each model's strengths and weaknesses by isolating the effects of factors such as crowding, occlusion, blur, and illumination.

### 2.2 Evaluation

To comprehensively evaluate the performance of the selected face detection models, several metrics are employed. These metrics provide insights into the models' accuracy, efficiency, and robustness under various conditions.

**Precision** Measures the proportion of correctly detected faces out of all detections made by the model. A high precision indicates fewer false positives.

**Recall** Measures the proportion of correctly detected faces out of the total number of actual faces in the image. A high recall indicates fewer missed faces.

**F1 Score** The harmonic mean of precision and recall, providing a single metric to balance the trade-off between precision and recall. It is particularly useful when both metrics are equally important.

**Inference Time** The inference time, measured in seconds, represents the time taken by each model to process a single image. This metric is critical for evaluating the suitability of models for real-time applications where low latency is essential.

### 2.3 Face Detection Models

In this project, we evaluated four state-of-the-art face detection models. These models were chosen for their distinct strengths, enabling a comprehensive evaluation across diverse scenarios.

**RetinaFace** It based on the *Retina-Net* architecture. It is a single-stage detector that leverages a *Feature Pyramid Network (FPN)* for multi-scale detection and integrates facial landmark localization, making it robust to occlusion, pose variations, and small faces.[12]

**Multi-Task Cascaded Convolutional Networks** *MTCNN* employs a three-stage cascaded convolutional network architecture comprising a *Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net)*. This model excels in combining face classification, bounding box regression, and facial landmark localization, providing efficient and lightweight performance, though it struggles with detecting small faces.

**OpenCV** It is a widely used open-source library for computer vision and image processing tasks. *OpenCV* includes built-in methods such as *Haar* cascades and *DNN*-based detectors for face detection, which are both efficient and customizable. It was utilized for image preprocessing, real-time face detection, and as a benchmarking tool to evaluate the performance of the models under different conditions.

**YOLOv5** An adaptation of the *YOLOv5* object detection framework, is designed for real-time face detection, prioritizing speed, and efficiency. While it delivers fast inference, its precision can be impacted by extreme occlusions or challenging conditions.

## 3 Results

### 3.1 Evaluation on the Number of Faces

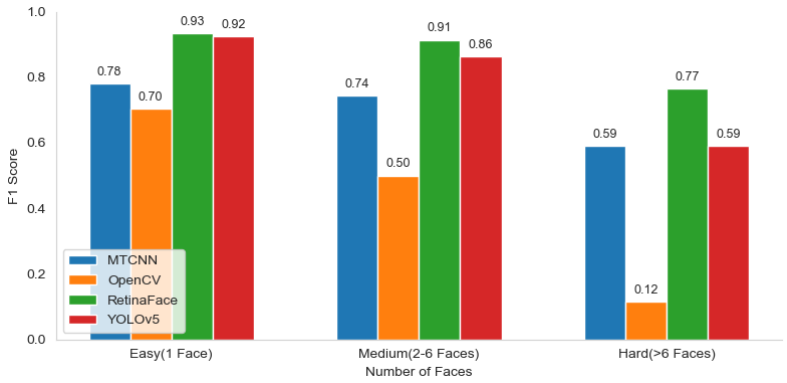


Figure 1: F1 Scores for the Number of Faces

Table 1: Model Performance (Precision and Recall) under the Number of Faces Condition

| Difficulty | MTCNN | | OpenCV | | RetinaFace | | YOLOv5 | |
|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Easy | 0.76 | 0.81 | 0.70 | 0.71 | 0.93 | 0.94 | 0.91 | 0.95 |
| Medium | 0.84 | 0.71 | 0.62 | 0.45 | 0.97 | 0.89 | 0.91 | 0.86 |
| Hard | 0.90 | 0.50 | 0.34 | 0.08 | 0.99 | 0.67 | 0.89 | 0.52 |

According to the statistics, *RetinaFace* is the most dependable model for identifying many or crowded faces since it consistently performs better than other models at all difficulty levels, with excellent precision (0.93+) and strong recall. *YOLOv5* also does well, particularly in Easy and Medium instances, but its overall efficacy is diminished in Hard situations due to a sharp decline in memory. *MTCNN* performs moderately well, striking a balance between recall and precision in easier scenarios while faltering in harder ones with a lower recall. Although *OpenCV* maintains a respectable level of precision in Easy cases, its low recall causes it to perform the worst, especially in Medium and Hard circumstances.

### 3.2 Evaluation on Blur

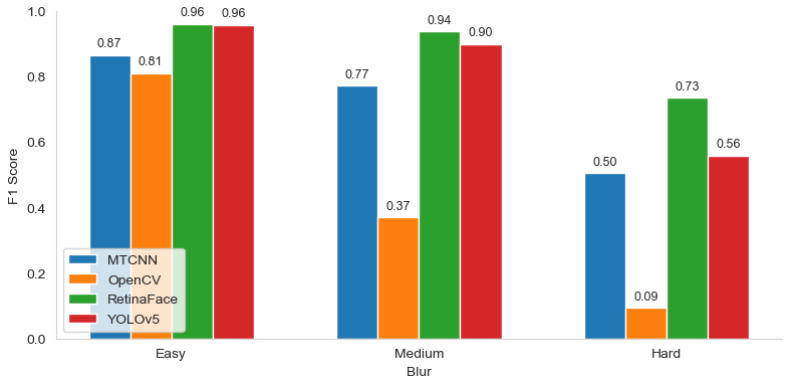


Figure 2: F1 Scores for Blur

Table 2: Model Performance (Precision and Recall) under the Blur Condition

| Difficulty | MTCNN | | OpenCV | | RetinaFace | | YOLOv5 | |
|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Easy | 0.86 | 0.89 | 0.84 | 0.80 | 0.96 | 0.97 | 0.95 | 0.98 |
| Medium | 0.82 | 0.76 | 0.50 | 0.33 | 0.96 | 0.93 | 0.92 | 0.90 |
| Hard | 0.81 | 0.41 | 0.28 | 0.06 | 0.97 | 0.64 | 0.84 | 0.49 |

*RetinaFace* is the most dependable model, even under difficult circumstances, according to the data, continuously delivering the best performance across all blur levels with the highest precision (0.95+) and strong recall. Next in line is *YOLOv5*, which does well in Easy and Medium instances but has a sharp decline in recall in Hard situations (0.48), which lowers its overall F1 score. In Easy and Medium scenarios, *MTCNN* performs quite well, striking a balance between precision and recall; nevertheless, in Hard circumstances, its recall drastically decreases (0.40), which compromises its dependability. *OpenCV* is not suited for properly handling blur because it performs poorly at all levels of blur, especially in Medium and Hard situations when its recall is quite low (0.32 and 0.06, respectively).

### 3.3 Evaluation on Occlusion

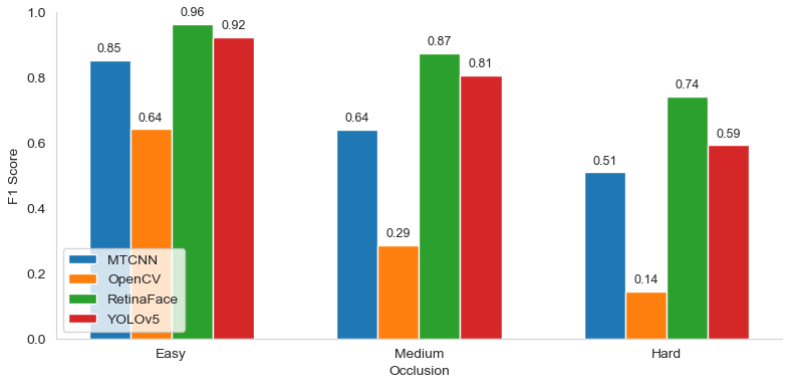


Figure 3: F1 Scores for Occlusion

Table 3: Model Performance (Precision and Recall) under the Occlusion Condition

| Difficulty | MTCNN | | OpenCV | | RetinaFace | | YOLOv5 | |
|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Easy | 0.87 | 0.94 | 0.69 | 0.62 | 0.97 | 0.97 | 0.94 | 0.94 |
| Medium | 0.76 | 0.60 | 0.44 | 0.25 | 0.94 | 0.84 | 0.88 | 0.79 |
| Hard | 0.81 | 0.41 | 0.36 | 0.11 | 0.98 | 0.64 | 0.87 | 0.52 |

According to the data, *RetinaFace* is the most dependable model even in cases of severe occlusion, continuously outperforming other models at all occlusion levels and achieving the highest precision (0.94+) and strong recall. With strong F1 ratings of 0.92 and 0.81 in Easy and Medium instances, respectively, *YOLOv5* likewise performs well; however, its recall sharply declines in Hard cases (0.52), reducing its total performance. In Easy instances, *MTCNN* performs rather well, balancing precision and recall. However, in Medium and Hard conditions, it performs poorly, especially in recall, yielding F1 scores of 0.64 and 0.50, respectively. With extremely low recall (0.24 and 0.10) and weak F1 scores (0.28 and 0.14), *OpenCV* performs the worst, particularly in Medium and Hard instances, making it inappropriate for handling occlusion.

### 3.4 Evaluation on Illumination

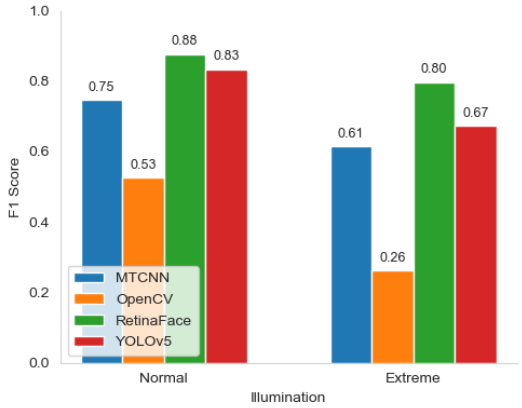


Figure 4: F1 Scores for Illumination

Table 4: Model Performance (Precision and Recall) under the Illumination Condition

| Difficulty | MTCNN | | OpenCV | | RetinaFace | | YOLOv5 | |
|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Normal | 0.84 | 0.72 | 0.63 | 0.50 | 0.95 | 0.85 | 0.92 | 0.83 |
| Extreme | 0.84 | 0.55 | 0.42 | 0.24 | 0.96 | 0.73 | 0.88 | 0.62 |

*RetinaFace* is the best model based on the data, outperforming other models in both normal and extreme illumination situations. It achieves the highest precision (0.95+ for normal, 0.96+ for extreme) and strong recall. With an F1 score of 0.83 under typical lighting, *YOLOv5* performs well; but, under harsh lighting, its recall falls to 0.62 (F1 score: 0.67). With an F1 score of 0.75 in typical illumination, *MTCNN* performs quite well; nevertheless, in extreme conditions, its efficacy is reduced (F1 score: 0.61). With subpar recall and F1 scores in both instances, *OpenCV* performs the worst and is hence inappropriate for different lighting conditions.

### 3.5 Evaluation on Inference Time

Table 5: Model Inference Time for Each Image

| | MTCNN | OpenCV | RetinaFace | YOLOv5 |
|---|---|---|---|---|
| Inference Time | 0.5 s | 0.028 s | 1.59 s | 0.17 s |

*OpenCV* performed the fastest inference time with 0.028 seconds per image, making it particularly suitable for scenarios requiring low latency. In contrast, *MTCNN* showed a significantly higher inference time of 0.5 seconds, which may limit its applicability in real-time environments. *YOLOv5's* achieved a balanced performance with an inference time of 0.17 seconds, offering a compromise between speed and detection accuracy.*RetinaFace*recorded the longest inference time at 1.59 seconds per image, potentially limiting its use in environments where rapid processing is highly important.

## 4 Conclusion

*RetinaFace* showed the best precision and recall, because of its strong architecture and multi-level feature localization, doing exceptionally well in situations with severe occlusions and difficult illumination. But because of its higher computational complexity, which results in lengthier inference times, it is more appropriate for applications that require accuracy than for real-time systems. For real-time applications like live video surveillance, *YOLOv5's* quick inference speed is perfect since it strikes a compromise between effectiveness and tolerable detection accuracy. For less complicated situations, *MTCNN* successfully balances accuracy and efficiency with intermediate precision and recall; nevertheless, it performs poorly in highly dynamic or occluded scenarios. Even while *OpenCV*-based detectors are quick and lightweight, their precision and recall greatly fall short of those of deep learning models, therefore they are only appropriate for straightforward jobs requiring little processing power.

# 5 References


[1] Insaf, A., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. *Electronics*, *9*, 1188. https://doi.org/10.3390/electronics9081188

[2] Minaee, S., Luo, P., Lin, Z., & Bowyer, K. (2021). Going deeper into face detection: A survey. *arXiv preprint arXiv:1903.06084*.

[3] Turk, M. A., & Pentland, A. P. (1991). Face recognition using eigenfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 586–591). https://doi.org/10.1109/CVPR.1991.139758

[4] Wang, Y.-Q. (2014). An analysis of the Viola-Jones face detection algorithm. *Image Processing On Line*, *4*, 128–148.

[5] King, D. E. (n.d.). Max-Margin Object Detection. *Unpublished manuscript*.

[6] Xiang, J., & Zhu, G. (2017). Joint face detection and facial expression recognition with MTCNN. In *Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 424–427). https://doi.org/10.1109/ICISCE.2017.95

[7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. (2016). SSD: Single Shot MultiBox Detector. In *Lecture Notes in Computer Science* (Vol. 9905, pp. 21–37). https://doi.org/10.1007/978-3-319-46448-0_2

[8] Hu, P., & Ramanan, D. (2016). Finding tiny faces. *arXiv preprint arXiv:1612.04402*.

[9] Tang, X., Du, D., He, Z., & Liu, J. (2018). PyramidBox: A context-assisted single shot face detector. *arXiv preprint arXiv:1803.07737*.

[10] Guo, J., Deng, J., Lattas, A., & Zafeiriou, S. (2021). Sample and computation redistribution for efficient face detection. *arXiv preprint arXiv:2105.04714*.

[11] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137–154.

[12] Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). RetinaFace: Single-Shot Multi-Level Face Localization in the Wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5202–5211. https://doi.org/10.1109/CVPR42600.2020.00525.

[13] Ghazali, R., El Abbadi, N., & Dosh, M. (2022). A comprehensive survey on face detection techniques. *Webology*, *19*, 613–628. https://doi.org/10.14704/WEB/V19I1/WEB19044.

[14] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342.

[15] Arriaga, O., Valdenegro, M., & Plöger, P. (2017). Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv:1710.07557*.