

Enhancing Real-Time Face Detection Using Deep Learning Architectures for Edge Devices

Problem Statement:

Face detection and recognition have become important components of modern security, authentication and mobile systems. Deep learning, especially the integration of the Convolutional Neural Network (CNN) has greatly improved accuracy and real-time capabilities in recent years (Zhang et al., 2016; Deng et al., 2020). However, many state-of-the-art models still require high computational resources, making them impractical for deployment on resource-constrained devices such as smartphones and edge devices. Models such as RetinaFace and Single stage headless (SSH) provide outstanding performance under benchmark conditions by taking advantage of dense anchor strategies and multi-scale feature maps (Deng et al., Najibi et al., 2017). However, it is necessary to effectively operate the powerful GPU. In contrast, light models such as FaceBoxes (Wang et al., 2021) are adapted to the CPU and offer real-time performance, but underperforms when dealing with small, occluded, or low-resolution faces. This performance resource trade-off creates a difference in current systems, especially in public safety, embedded vision and mobile applications, where computational resources are limited. While many detection models address parts of the problem or emphasize accuracy or efficiency. Few offer a balanced solution that is both strong and deployable on low-power devices. Current literature mainly focuses on performance in controlled settings, with a limited evaluation in the actual environment (Li et al., 2020; Zhang et al., 2021). Therefore, there is a need to develop a structure that integrates architectural innovations from both a high-accuracy and light model, capable of detecting a reliable face in various hardware and environments.

Research Objectives:

The primary goal of this research is to design and evaluate a deep learning-based face detection model that achieves balance between accuracy and computational efficiency, which makes it suitable for both high performance and low-resource devices. To achieve this goal, the study aims:

1. To analyze the performance characteristics of the existing face detection models in terms of accuracy, speed and hardware compatibility.
2. To identify architectural components—multi-level features and light modules — which increase the strength and efficiency of detection.

3. To develop a hybrid face detection model that integrates the strength of both high-performance and lightweight architecture.
4. To evaluate the performance of models proposed under both controlled (benchmarks) and real-world conditions in various hardware platforms (CPU, GPU, mobile).
5. To provide practical deployment guidelines to apply models in the edge-computing environment.

Literature Review:

Recent advances in deep learning have significantly improved the accuracy and efficiency of face detection systems. Convolutional Neural Networks (CNNs) have emerged as the core technology behind these developments, enabling real-time detection across a range of environments. However, despite these strides, balancing model precision with computational efficiency—especially for deployable on edge devices—remains an ongoing challenge. This literature review synthesizes findings from six key studies to explore performance trade-offs, architectural innovations, and practical limitations in contemporary face detection systems.

Performance Trade-offs and Model Efficiency:

Performance trade-offs and model efficiency vary in modern facial detection approaches to how they adapt to either accuracy or speed. RetinaFace (Deng et al., 2020) and SSH (Najibi et al., 2017) prefer high identification accuracy using deep feature representations and dense anchor-based strategies. These models achieve state-of-the-art results on benchmark datasets, such as by increasing sensitivity to WIDER FACE dataset, small or surrounded faces. However, their dependence on complex backbones and multi-scale features imposes high computational costs on feature extraction, making them less suitable for resource-constrained settings. Conversely, models such as FaceBoxes (Wang et al., 2021) and YOLO-Face (Zhang et al., 2021) are specially designed for real-time applications on low-power hardware platforms. FaceBoxes employs a lightweight CNN structure adapted to CPU performance, while the YOLO-Face uses neural architecture search to optimize both speed and accuracy. Although these models excel in terms of time and compatibility with edge equipment, their detection performance leads to a decline in performance under extreme poses or visually complex settings.

Architectural Design: Single-stage vs. Multi-stage Pipelines:

The trade-off between accuracy and speed revolves around the decision between single stage and multi stage pipelines. RetinaFace, SSH, and YOLO-Face are examples of single

stage detectors that detect in a single pass and enable quick estimations. These systems integrate modules, multidimensional characteristics, and enhance detection quality without initiating a delay. However, multistage frameworks that focus on refinement through processing through numerous layers or stages include CNN Cascade (Li et al., 2020) and MTCNN (Zhang et al., 2016). For joint facial identification and alignment, MTCNN employs a three-stage CNN cascade. In a similar vein, CNN's cascade architecture increases processing overheads while filtering false positives in layers to boost dependability.

Facial Landmark Detection and Application Scope:

Facial landmark detection is another key factor that sets some of these models apart. Models like MTCNN and RetinaFace not only detect bounding boxes but also provide precise landmark localization, which makes them especially useful for follow up tasks like identity verification, face alignment, and expression recognition. On the other hand, models such as CNN Cascade, FaceBoxes, SSH, and YOLO-Face focus mainly on bounding box accuracy and don't include landmark detection. While this design choice speeds up inference and simplifies deployment, it also limits the model's flexibility, especially in more complex facial recognition pipelines.

Generalizability and Real-world Constraints:

While all six models perform well on benchmark tests, the situation is very different when used in real world environments. Most of these models were trained on standardized datasets such as FDDB, AFW, and Broad Faces, which are useful for comparison but do not accurately reflect real life conditions, such as different lighting, unusual camera angles, or unexpected user movement. The SSH model, for example, is excellent at detecting small faces, but it is too heavy to run in real time on mobile devices. The FaceBoxes model performs well under ideal conditions, but its accuracy drops significantly when faces are at unusual angles or are partially occluded. Therefore, although these models excel in controlled environments, they struggle to perform as conditions become more complex. For this reason, there is a growing need for models that can adapt to diverse and unexpected inputs without relying on expensive hardware.

Implications for Future Work:

One of the biggest challenges is finding a balance between real time speed and accuracy in real world conditions. One possible idea is to combine the strengths of different models. For example, MTCNN, FaceBoxes, and YOLO-Face models are known for their fast performance, while RetinaFace is known for its high accuracy. Combining these models could provide the best of both worlds: speed and accuracy. Looking ahead, it is also important to focus on testing these systems in real-world environments (not just in labs)

and designing them to be adaptable depending on the power of the device or the complexity of the surrounding environment.

References:

Deng, J., Guo, J., & Zafeiriou, S. (2020). RetinaFace: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*. <https://doi.org/10.48550/arXiv.1905.00641>

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2020). A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Najibi, M., Samangouei, P., Chellappa, R., & Davis, L. S. (2017). SSH: Single stage headless face detector. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Zhang, H., Luo, Y., & Liu, X. (2021). YOLO-Face: A real-time face detector optimized by neural architecture search. *Pattern Recognition Letters*.

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>

Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). FaceBoxes: A CPU real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (pp. 1–9). <https://doi.org/10.1109/BTAS.2017.8272695>