

Enhancing Real-Time Face Detection for Edge Devices using Hybrid Lightweight Deep Learning Models

Student name: Shahad AlHarbi

Contents

1. Abstract	3
2. Introduction.....	4
2.1. Background	4
2.2. Challenges.....	4
2.3 Problem statement	5
2.4. Research objectives.....	5
2.5. Research scope	6
3. Literature review.....	6
4. Research Methodology	7
4.5. Ethical Considerations	8
4.6. Limitations.....	8
4.7. Expected Outcomes	8
4.8. Gantt Chart	9
5. Reference:.....	10

1. Abstract

The widespread use of handheld and embedded devices has created a pressing need for face detection methods that can run within tight energy and processing budgets. Current deep neural networks yield excellent detection accuracy, but their size and complexity confine them to high performance platforms. This paper considers face detection within the broader scope of computer vision, offering a thorough literature survey and empirical comparison to illuminate where existing solutions fall short. On the basis of those insights, it lays out a step by step research program: reproduce and benchmark current models, design a hybrid lightweight detector that combines a pared down backbone with context sensitive feature fusion and multi task outputs, and test that design on standard datasets like WIDER FACE across processors ranging from desktop CPUs and GPUs to mobile chips. The goal is to deliver a face detection architecture that meets the constraints of edge computing, backed by comparative data and practical guidance for deploying such systems.

2. Introduction

Human faces encode a wealth of information about identity and behaviour, making face detection a fundamental problem in computer vision. Accurate localisation of facial bounding boxes and landmarks enables higher-level tasks such as recognition, expression analysis and driver monitoring. Early detectors, based on hand-crafted features like Haar cascades, performed poorly under variations in pose, illumination and occlusion. Advances in deep learning produced dramatic gains in robustness; for example, multi-task cascaded convolutional networks (MTCNN) introduced a unified pipeline for detection and landmark localisation[1]. However, many deep detectors assume access to dedicated GPUs. A real-time face recognition study notes that these models are computationally heavy and form the principal bottleneck in embedded systems[6]. Edge applications such as mobile authentication or unmanned-aerial-vehicle rescue therefore demand detectors that are fast, accurate and resource efficient.

2.1. Background

Developers who design face detectors tend to gravitate toward one of two routes. One route chains several neural networks together: an early pass quickly discards image patches that obviously are not faces, and later passes zero in on the correct bounding box and even pick out facial features such as the eyes and mouth. MTCNN and the CNN Cascade [4] are examples of this more elaborate, multi-stage pipeline[1]. It can be quite precise, but running several models back-to-back inevitably slows down the process. The other route tries to get everything done in one sweep. RetinaFace, SSH [5] and YOLO-Face, for instance, read the convolutional feature maps and decide where faces are in a single forward pass. RetinaFace combines a MobileNet-V1 backbone with a context module to build feature maps at several scales and then lays a dense grid of anchor boxes over every pixel[2]. That abundance of anchors produces lots of overlapping boxes that must later be merged[2]. Among the leanest options, FaceBoxes [3] employs a small convolutional network that can process about twenty frames per second on a standard CPU, though it has trouble with tiny or partially occluded faces. YOLO-Face adapts the familiar YOLO design through automated architecture search, giving up a bit of accuracy for greater speed. Selecting among these strategies is a matter of balancing precision against computational overhead and the limitations of the target hardware.

2.2. Challenges

Although deep learning approaches have improved face detection considerably, some problems persist. The most accurate models typically depend on substantial neural backbones and densely arranged anchor boxes, which place heavy demands on computation and memory. By contrast, stripped-down architectures run more quickly but may fail to pick up tiny faces or cope with occlusion and poor lighting. Laying down a dense grid of anchors also generates overlapping candidate boxes that must be weeded out via non-maximum suppression [2], further slowing the process. On top of that, training cutting-edge detectors calls for extensive labelled datasets and meticulous tuning of hyperparameters. These competing requirements become especially difficult to reconcile when the intended platform has limited power and processing capacity.

2.3. 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 [1][2]. 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 [2][5]. However, it is necessary to effectively operate a powerful GPU. In contrast, lightweight models such as FaceBoxes [3] are adapted to the CPU and offer real-time performance, but underperform 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 [6][7]. Therefore, there is a need to develop a structure that integrates architectural innovations from both a high-accuracy and lightweight model, capable of reliably detecting faces across diverse hardware and environments.

2.4. 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.

2.5. Research scope

This research is focused on the task of face detection only, without extending into recognition or anti-spoofing systems. The study will use publicly available datasets such as WIDER FACE, FDDB, and AFLW for training and evaluation, which ensures consistency with prior work and makes replication possible. The proposed model will be tested on three hardware settings: a desktop CPU, a GPU workstation, and a mobile device. By comparing performance across these platforms, the project seeks to understand how a single model design can adapt to different levels of computational power.

The scope does not cover broader social or ethical concerns such as demographic bias, fairness, or user privacy, although these are important topics for future research. Instead, the project remains technical in nature, aiming to balance accuracy with efficiency in a way that makes deployment on edge devices more practical.

3. Literature review

Early efforts in deep learning-based face detection laid the groundwork for today's methods but revealed persistent trade-offs between accuracy and efficiency. One of the earliest examples, MTCNN [1], demonstrated that a cascaded approach could simultaneously detect faces and estimate landmarks such as eye and mouth positions. Its multi-stage refinement improved localization, yet the sequential structure and large computational cost made it slow and unsuitable for deployment on mobile or embedded devices.

To address these limitations, subsequent models sought to streamline detection into a single pass. RetinaFace [2] integrated context modules and multi-scale features, achieving high accuracy and landmark localization within one pipeline. While RetinaFace quickly became a benchmark on datasets such as WIDER FACE, its dependence on GPU acceleration renders it impractical for CPU- or mobile-based systems. Similarly, SSH [5] adopted a single-stage design and incorporated multi-scale receptive fields to boost precision. However, like RetinaFace, its improvements came at the expense of higher resource consumption.

A different direction was pursued with FaceBoxes [3], which prioritized speed by employing lightweight convolutional layers optimized for CPUs. The model achieved real-time performance, making it attractive for edge devices. Its drawback, however, lies in reduced accuracy, particularly on small or occluded faces. The CNN Cascade [4] also followed a multi-stage refinement strategy, offering robust detection in unconstrained settings but requiring significant computation, which limited its practicality outside high-performance environments.

Growing interest in efficiency gave rise to ultra-lightweight approaches. The ULFG model [6], for example, compressed the network to around 1 MB while maintaining acceptable accuracy. It could run on low-power hardware such as Raspberry Pi, proving that extreme efficiency was possible. Yet, this came with reduced robustness, especially when detecting very small or low-quality faces. Another lightweight variant, LAFD [7], combined MobileNetV3 with deformable convolutions and attention mechanisms, producing accuracy gains over RetinaFace while

keeping the network compact. Nonetheless, sensitivity to input resolution changes limited its consistency in diverse applications.

Recent studies have explored hybrid designs to combine accuracy with efficiency. IFVMNet [8] introduced FasterNet modules and adaptive pruning into a YOLOv8 framework, reducing redundancy while maintaining competitive detection accuracy. Similarly, CSTDNet [9] integrated contextual and complementary feature modules, significantly improving the detection of small objects. Although these models bridged some of the gap, they often increased architectural complexity, which may hinder straightforward deployment. The YOLO-Face model [10] followed a similar philosophy, adapting the general YOLO framework for faces by refining anchor boxes and tuning for real-time inference. While it showed practical benefits, the added complexity echoed the same challenges faced by other hybrid detectors.

Taken together, the literature reveals a consistent pattern. High-accuracy detectors such as RetinaFace [2] and SSH [5] excel in benchmark conditions but require specialized hardware. Lightweight models like FaceBoxes [3] and ULFG [6] are practical on CPUs and mobile devices but suffer from reduced precision under challenging conditions. Hybrid solutions (e.g., CSTDNet [9], YOLO-Face [10]) suggest that both goals can be pursued simultaneously, yet they introduce design complexity that limits ease of adoption. These trends highlight a persistent research gap: the absence of a balanced detector that combines robustness with efficiency across diverse platforms. This gap provides the motivation for the present study.

4. Research Methodology

This project is organised into four phases. It begins with a review of existing research and ends with testing and deployment. The structure is intended to move step by step, starting from understanding what has already been done and finishing with a working model that can be deployed.

4.1. Phase 1 – Literature Review

The first phase covers a review of face detection studies and models. Examples include MTCNN [1], RetinaFace [2], FaceBoxes [3], YOLO-Face [10], CNN Cascade [4], SSH [5], ULFG [6], LAFD [7], IFVMNet [8], and CSTDNet [9]. The purpose is to identify where current methods succeed and where they fall short. In practice, this stage should highlight the well-known trade-off: more complex models often give better accuracy, while lightweight ones are faster but lose precision.

4.2. Phase 2 – Benchmarking Existing Models

The second phase benchmarks some of the well-known detectors. Where possible, pre-trained weights will be used. The WIDER FACE dataset will serve as the main testing ground. Key measurements include precision, recall, mean average precision (mAP), processing speed, and memory use. This step is meant to set a clear baseline so that later results can be compared fairly.

4.3. Phase 3 – Design and Training of the Proposed Model

The third phase focuses on building the new hybrid model. It will be based on an efficient backbone, such as MobileNet V3 or FasterNet, and will also include context-aware modules with multiple outputs for detection and landmarks. Training will be carried out in PyTorch using WIDER FACE. Optimisation methods like augmentation, pruning, and quantisation will be applied. The objective is not only to improve accuracy but also to ensure the model remains lightweight enough to run on different devices.

4.4. Phase 4 – Evaluation and Deployment

In the last phase, the model will be tested on FDDB and AFLW as well as across different hardware platforms (CPU, GPU, and mobile). Deployment frameworks such as ONNX Runtime and NCNN will be used to simulate real-world conditions on mobile devices. The aim is to check whether the model works in practice, not only in controlled benchmarks.

4.5. Ethical Considerations

The datasets used in this research (WIDER FACE, FDDB, and AFLW) are all publicly available. No personal or sensitive data will be collected. Since the focus is detection and not recognition, the work does not involve identifying individuals. Still, it is important to recognise that face detection in general can raise fairness and bias concerns. These issues are beyond the current scope but are acknowledged as important for future work.

4.6. Limitations

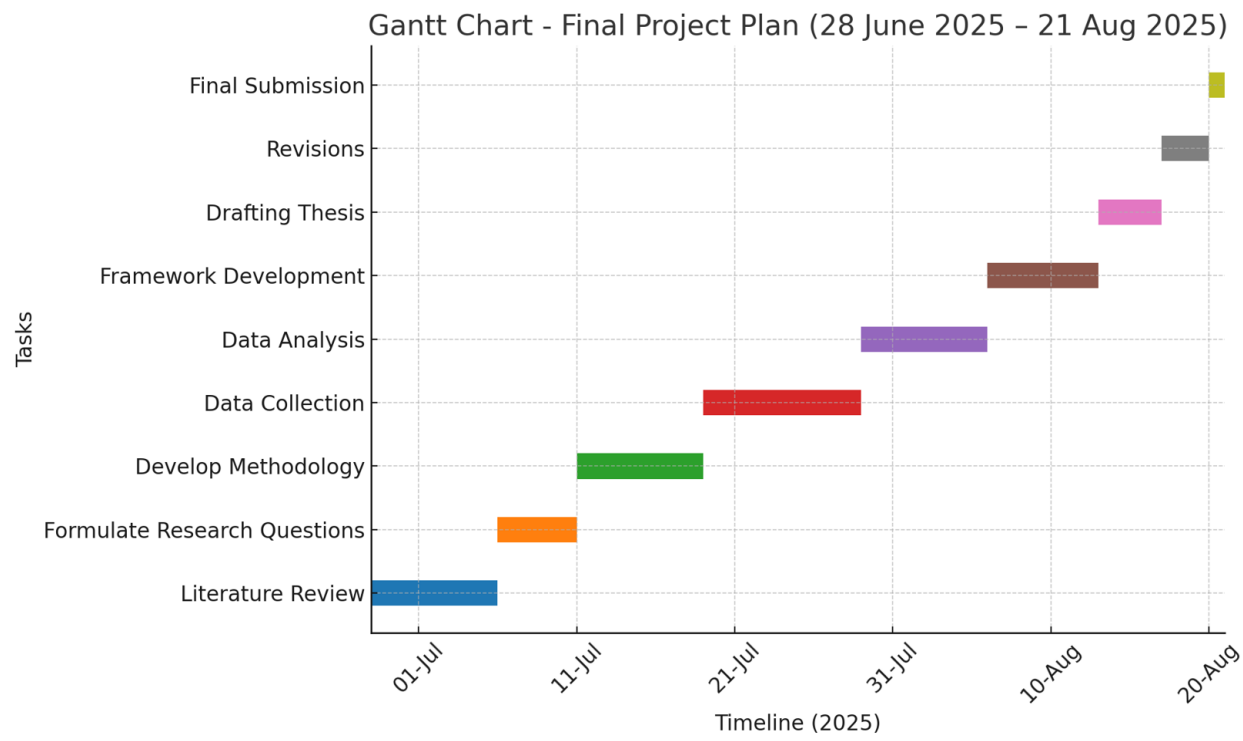
The project is limited to face detection only. It does not deal with recognition, spoofing, or demographic fairness. The experiments rely on a few public datasets, which may not represent every real-world situation. Hardware testing is also limited to one CPU, one GPU, and one mobile device, which means results may not generalize across all possible environments.

4.7. Expected Outcomes

The expected outcomes include:

- A review of current face detection approaches.
- Benchmarking results of existing detectors.
- Development of a lightweight hybrid model.
- Evaluation on multiple datasets and hardware platforms.
- Guidelines for deploying face detectors on edge devices.

4.8. Gantt Chart



5. References:

- [1] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [2] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [3] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2017.
- [4] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [5] S. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017.
- [6] L. Yu, H. Liang, and J. Dong, "Ultra-Light-Fast-Generic-Face-Detector," GitHub repository, 2019. [Online]. Available: <https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB> (accessed: Aug. 21, 2025).
- [7] A. Chandrasekar, "Light and accurate face detection using MobileNetV3 and improved RetinaFace," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 14, no. 2, pp. 330–338, 2023.
- [8] J. Yang, Z. Xu, and Q. Chen, "IFVMNet: Lightweight and efficient object detection based on FasterNet and YOLOv8," *Applied Sciences*, vol. 15, no. 2, Art. no. 12216, 2025.
- [9] Y. Wang, X. Li, L. Zhang, and C. Wu, "Contextualized small target detection network (CSTDNet)," *Computers and Electronics in Agriculture*, vol. 203, Art. no. 107115, 2023.
- [10] T. Li, H. Zhang, and Z. Sun, "YOLO-Face: Improved YOLO model for face detection," *J. Vis. Commun. Image Represent.*, vol. 78, Art. no. 103145, 2021.