

Face Detection in the Practical Conditions: Evaluating MTCNN Performance, Transfer Learning, and Hardware Efficiency on the WiderFace Dataset

ARSENIY TOKAREV, University of Twente, The Netherlands

Pre-trained face detection models such as MTCNN achieve strong results on standard benchmarks, however, it remains unclear how their performance holds up under varying real world conditions. This paper evaluates MTCNN on the WiderFace dataset across three difficulty levels of images, finding that precision remains stable, dropping from 0.93 to 0.88, while recall declines sharply from 0.94 to 0.33. This degradation is explained by a larger presence of small, overlapping faces in crowded scenes as image complexity increases. A transfer learning experiment was conducted using MobileNetV2 to explore whether a general purpose pre-trained model can be adapted to a specific scene classification task, which scored 68% for test accuracy. An interesting methodological finding revealed that whole images substantially outperformed cropped faces as model input, due to the loss of contextual cues and environmental details that cropping caused. Data augmentation, similarly, was found to decrease performance on the small dataset instead of improving it. Cross demographic robustness experiments revealed that models trained on one demographic group generalise poorly to others, emphasizing the importance of both data diversity and data size in robustness of a model. Finally, inference and training speed are measured and compared between CPU and GPU on Apple's M1 hardware, showing comparable performance. Collectively these findings illustrate the limitations of deploying pre-trained face detection models under real world conditions and the trade-offs involved in adapting them through transfer learning.

Additional Key Words and Phrases: face detection, MTCNN, WiderFace, transfer learning, MobileNetV2, CPU/GPU performance, model robustness

1 Introduction

In recent years, deep learning has made significant progress in the field of computer vision, with face detection becoming one of the most extensively researched practical applications [1]. Face detection, which consists of localizing human faces within images, is a fundamental step for practical systems such as security surveillance, document and access control, and biometrics [2]. As these systems are increasingly being deployed for use in real-world applications, the need to reliably detect faces under diverse and difficult conditions becomes essential [3].

Existing pre-trained models, such as MTCNN (Multi-task Cascaded Convolutional Networks) displayed competent performance when tested on standard benchmarks. However, the systematic evaluation of how their performance declines with varying levels of image difficulty remains underexplored [4]. Understanding this degradation is important because real-world images often differ significantly from controlled benchmark conditions. The WiderFace

dataset contains a structured framework for such analysis and categorizes more than 32,000 images into Easy, Medium, and Hard difficulty levels based on factors such as face size, occlusion, pose variation, and scene complexity [5]. Evaluating MTCNN across these three categories allows us to determine where the model fails and also when and why that happens.

Another well-documented concern in face detection is demographic bias, which can undermine robustness. Models trained on non diverse datasets perform significantly worse on underrepresented groups. Early face detection systems, for example, worked reliably mainly on pale-skinned male faces and failed for other demographics [6]. This issue is relevant to this day, as models that are fine-tuned on certain demographics may underperform in generalizing for others. Experimentally evaluating this lack of robustness, by first intentionally training on one group and testing on another one, offers concrete evidence of how the diversity of training data influences performance in real-world applications.

Moreover, the computational cost of running face detection is becoming increasingly relevant as models today are being deployed on a wide variety of hardware platforms. The introduction of Apple Silicon's unified memory architecture, where CPU and GPU share the same memory pool instead of operating independently, challenges the conventional assumption that GPU inference is considerably faster than CPU [7]. Therefore, this makes the empirical comparison of performance on this hardware a relevant and practical contribution.

This paper confronts these gaps by evaluating the MTCNN model across different difficulty categories in the WiderFace dataset and using transfer learning with MobileNetV2 for a specific category, in our case soldiers versus civilians. It also compares inference and training speeds on Apple's M1 processor between CPU and GPU. Lastly, robustness experiments will be conducted to explore how models generalize across demographic groups.

The remainder of the paper's structure contains the problem statement and relevant research questions (Section 2). Section 3 covers related work; the fourth section will describe the methodology, and Section 5 displays the results. The last sections will discuss the findings and conclude the paper.

2 Problem Statement

Although pre-trained face detection models are commonly used, there is still a notable absence of comprehensive assessments of how their performance differs amongst various levels of image difficulty, demographics, and hardware environments in a combined study. This paper addresses this gap by evaluating MTCNN on the WiderFace dataset, investigating limitations of pre-trained face detection models and attempts to improve it using transfer learning in practical conditions.

45th Twente Student Conference on IT, July 4, 2026, Enschede, The Netherlands

© 2026 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2.1 Research questions

This leads to the following research question:

How well does MTCNN perform as a face detector for various real world conditions, and could transfer learning improve the performance on a specific category of scenes?

2.1.1 Sub research questions

RQ1: How does precision and recall vary across different difficulty images in the WiderFace dataset?

RQ2: How does inference speed compare between CPU and GPU on Apple's M1 hardware for the MTCNN face detection model?

RQ3: To what extent does transfer learning with MobileNetV2 improve specific scene category classification, and how does training and speed compare between CPU and GPU?

RQ4: How robust is a fine-tuned model when trained under one specific demographic group and tested on another?

3 Related Work

Tariyal et al. compared four of the most popular face detection algorithms, namely Viola-Jones, MTCNN, SSD and Yolo, evaluating the trade-offs between speed and accuracy [8]. The findings revealed that especially for facial landmark detection, MTCNN performed with high accuracy, but faced challenges in meeting real-time speed requirements when compared to other models such as YOLO and SSD. Despite this limitation, it is still a widely popular model used for baseline face detection, mainly due to multi-task capability which combines detection and landmark localisation in a single pipeline. Viola-Jones was considered a pioneer in the field, but was outperformed by more modern deep learning approaches when benchmarked for speed and accuracy. These findings emphasise that even though MTCNN performed with high accuracy for face detection, its computational requirements are also an important factor, which motivates us to explore the CPU and GPU speed comparison.

Yang et al. introduced the WIDER FACE dataset to overcome the drawbacks of existing face detection benchmarks, which at that time typically only had a few thousand faces with limited pose, scale and occlusion variations [5]. The dataset contains 32,203 images across 60 event categories, and 393,703 of them have labelled faces. This makes it ten times larger than datasets available previously. Moreover, images are categorized into three difficulty categories: Easy, Medium and Hard, which are mainly based on detection rate with an average recall rates of 92%, 76%, 34%. The research displayed that every one of the four baseline detection methods tested revealed substantial performance drops going from Easy to Hard subsets, with the average precision falling below 30% on the Hard set. This confirms that face detection performance degrades substantially as complexity of scenes in the images increases, such as inclusion of crowds and smaller faces.

One approach for improving face detection for certain types of scenes is transfer learning, which allows already pre-trained models to be adapted to new domains with limited data. Hussain et al. exhibited that MobileNetV2 using transfer learning scored 98-99% for classification accuracy on face mask detection datasets, with image quantities ranging between 2500-4436. This was a performance improvement, even over a custom Deep Convolutional Neural Network that achieved 97% accuracy for the same data [9]. The discovered

results confirm that MobileNetV2's pre-trained ImageNet weights show strong generalization even if just the final layers of classification are retrained for a new domain. This supports the use of MobileNetV2 as a base for our transfer learning in face-related classification.

4 Methodology

4.1 Experimental Setup

This subsection describes the dataset and exploratory data analysis that form the groundwork for the experiments mentioned in Section 4.2.

4.1.1 Dataset

This study uses WiderFace validation set as the main dataset for all conducted experiments. The validation set was chosen because it contains ground truth box annotations which are available to the public and allow us to perform direct evaluation of the detection performance. The test set annotations are not publicly released, which is why for our experiments the standard choice was the validation set. This dataset contains images that are organized into 61 event categories such as sporting events, military scenes and celebrations. For every image there are annotations for the bounding box for every visible face in the image.

Every image in the WiderFace dataset has a label corresponding to the difficulty level, which are Easy, Medium and Hard that are based on factors such as scene complexity, face sizes and pose variations. To conduct our experiment at each difficulty level, 20 images were randomly selected from every difficulty category, for a total of 60. A fixed random seed of 42 was chosen for sampling for reproducibility purposes.

In the transfer learning experiment, images from soldier event categories were picked, namely Soldier Patrol, Soldier Drilling, and Soldier Firing for a total of 120 images. An equal amount of civilian images were also sampled from Interview, Couple, Spa and Hand-shaking categories. The balanced dataset of 240 images was equally split into two classes (soldier and civilian images) of 120 images each, with each class further split into 80 for training, 20 for validation, and 20 for testing.

4.1.2 Data Analysis

In order to better understand how the sampled images are composed, an exploratory data analysis (EDA) was conducted on all 60 of the images. Every image was manually labelled for each of the three attributes: gender (male, female or mixed), setting (indoor versus outdoor), and scene type (single person, couple, or crowd). This was done manually rather than automatically due to a smaller sample size, which would make visualization more reliable and feasible.

The labels that resulted were then plotted on bar charts to visualize how every attribute varies across the three difficulty levels. The first purpose was to provide a descriptive overview of the types of images at each difficulty level to better identify patterns before running any detection experiments. Secondly, it provides context to interpret detection results, that is, which factors contribute to the degradation in performance.

The manual labels were also used in robustness experiments described in Section 4.2.4, where the models are trained on one

demographic group and tested against another to explore how well generalization works across different types of images.

4.2 Experimental Execution

In this section, the experiments conducted to answer the research questions are explained, building on the setup described in Section 4.1.

4.2.1 RQ1: MTCNN Precision and Recall For Varying Difficulty Levels

To explore how MTCNN’s detection precision and recall change as image difficulty increases, MTCNN (Multi-task Cascaded Convolutional Networks) has been chosen to be used for face detection baseline evaluation. This model operates in a three staged cascaded pipeline: first, the Proposal Network (P-Net), then the Refinement Network (R-Net), and lastly output the Output Network (O-Net) [10]. Every stage progressively refines candidates for a potential face and outputs bounding boxes along with landmark coordinates.

The model was evaluated on all 60 of the sampled images without any fine tuning or additional training. The detection accuracy was assessed by calculating Intersection over Union (IoU) with a threshold of 0.5, defined as:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (1)$$

This will be calculated as follows:

Given two bounding boxes, one produced by the model (A) and the other from annotations as the ground truth box (B), we can annotate them as $A = (x_1, y_1, x_2, y_2)$ and $B = (x_1', y_1', x_2', y_2')$, where (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates of the box.

Next step is to calculate the intersection area, which is done with:

$$\text{Intersection} = \max(0, \min(x_2, x_2') - \max(x_1, x_1')) \times \max(0, \min(y_2, y_2') - \max(y_1, y_1')) \quad (2)$$

This computes the width and height of the region that is overlapping between the two boxes, and we include $\max(0, x)$ to make sure if there is no overlapping, the intersection is 0 instead of negative.

After that, IoU is calculated with the following formula

$$IoU = \frac{\text{Intersection}}{(x_2 - x_1)(y_2 - y_1) + (x_2' - x_1')(y_2' - y_1') - \text{Intersection}} \quad (3)$$

The $(x_2 - x_1)(y_2 - y_1)$ is box A area, and $(x_2' - x_1')(y_2' - y_1')$ area of box B. The intersection is subtracted from their sum to not count the overlapping area twice. If IoU is larger than 0.5 then we count the detection as a true positive; otherwise, it counts as a false positive. Once every detection has been classified this way, precision is calculated as the number of true positives divided by the total number of detections the model made, showing how many of its predictions were actually correct. Recall is calculated as the number of true positives divided by the total number of real faces in the image, showing how many of the faces that were actually there got found. We expect recall to decline more sharply than precision

since Hard images contain a lot more smaller, overlapping faces in crowded scenes, which are more likely to be missed than mislocated.

4.2.2 RQ2: CPU vs GPU Speed Comparison

To determine whether GPU acceleration provides a speed advantage over CPU on Apple’s M1 hardware, I recorded MTCNN’s inference time separately on CPU and GPU for each of the 60 sampled images using Python’s time module. The same comparison was repeated for training time during the transfer learning experiment, which allowed both inference and training speed to be compared across the hardware. Because Apple’s M1 chip uses unified memory architecture, where CPU and GPU share the same physical memory pool, we expect the difference to be smaller than what is typically reported on traditional hardware with separate memory.

4.2.3 RQ3: Transfer Learning For Scene Classification

In order to investigate whether transfer learning can improve face detection performance on specific types of scenes, we conducted a binary classification experiment using MobileNetV2 as the base model. It is a lightweight convolutional neural network that is pre-trained on the ImageNet dataset that contains more than 1.4 million images and over 1000 categories [11]. The reason for selecting this model was efficiency and powerful generalisation capability, and due to a wide usage within transfer learning for image classification [9].

The classification task mainly consisted of differentiating soldier and civilian scenes apart, which are taken from the WiderFace validation set. Soldier images were sampled from three different event categories, namely: Soldier Patrol, Soldier Drilling and Soldier Firing, totalling 120 images. The other set contained the same number of images, from the following categories: Interview, Couple, Spa and Handshaking, which formed a binarily split dataset of 240 images in total. Additionally, in order to match MobileNetV2’s input requirements and normalize to the range of [0,1], each image was resized to 160x160 pixels. As described in Section 4.1.1, the dataset was split into 80 training, 20 validation, and 20 testing images per class, with accuracy measured as the proportion of test images correctly classified as soldier or civilian.

Afterwards, the pre-trained ImageNet weights were loaded into the MobileNetV2 base and its layers were frozen in order to preserve the existing learning and avoid overwriting it by training on a small dataset. Two layers were then added on top, a GlobalAveragePooling2D layer and a single Dense output layer with sigmoid activation, which is a standard approach for transfer learning when training is done on a lightweight classification head instead of the entire network. This resulted in only 1,281 trainable parameters from a total of 2,259,265. Compilation for the model was done by the Adam optimiser with a learning rate of 0.0001 and binary cross-entropy loss. The training was done in 10 epochs with a batch size of 32. Initially, faces were cropped for input but this resulted in poor performance as cropping faces removed contextual information that could help the model learn, such as uniforms or specifics of the environment. Switching to using whole images significantly improved the accuracy.

Moreover, I conducted data augmentation experiments, such as rotations up to 20 degrees, zoom up to 20% and random horizontal flips, to investigate if this improved performance on the small dataset.

Given the small dataset, we expect the accuracy to be moderate and potentially prone to overfitting, and it is not certain whether augmentation will help or hurt the performance at this scale.

4.2.4 RQ4: Cross-Demographic Robustness

To measure how well a model trained on one demographic group generalises to a contrasting group, I trained MobileNetV2 on one group and tested it on a contrasting one within the same attribute category, using setting (Outdoor vs Indoor), gender (Male vs Mixed), and scene type (Crowd vs Single); Female and Couple were excluded for insufficient sample size (10 and 2 images). For each comparison, the model used the same frozen-base configuration, trained for 10 epochs on one group, and was evaluated on the contrasting group, with training images labeled positive and testing images labeled negative, so accuracy reflects how distinguishable the two groups were to the model rather than a direct classification of the attribute itself. Group sizes ranged from 18 to 40 images, limited by the 60 manually labeled images available. Six experiments were conducted (Indoor/Outdoor, Male/Mixed, Crowd/Single, each direction), with accuracy recorded to assess generalisation across demographic distributions. We expect accuracy to drop substantially given the mismatched demographics and small group sizes.

5 Results

5.1 Exploratory Data Analysis

Sample images from each of the three difficulty categories are provided in Appendix A for reference, as their content is illustrative rather than directly relevant to the quantitative results discussed below.

After manually labeling each of the 60 sampled images across gender, setting and scene type, the resulting distributions display a noticeable difference between difficulty categories, as shown in Figure 1.

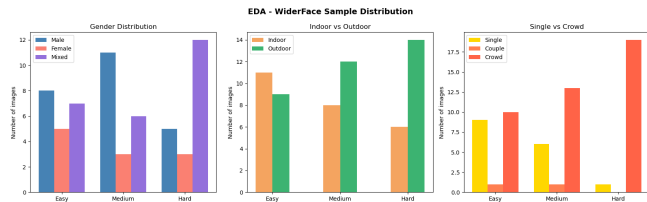


Fig. 1. EDA distribution chart — gender, indoor/outdoor, single/crowd bar charts

The easy image set was relatively balanced for all three attributes, showing a roughly even split between indoor (11) and outdoor (9) scenes, and a mix of single subjects (9) and crowds (10). Medium images tended to favor outdoor scenes (12) over (8) that were indoors and usually had larger groups of people, 13 crowd images versus 6 that had a single subject. Lastly, the hard image group skewed heavily towards crowd scenes, 19 out of 20, had a substantial amount of outdoor images (14) and 12 mixed gender groups.

5.2 MTCNN Baseline Evaluation

Table 1 displays the summaries of MTCNN’s precision, recall, and inference speed across the three difficulty categories.

Table 1. MTCNN precision, recall and inference speed

Category	Precision	Recall	CPU Time (s)	GPU Time (s)
Easy	0.93	0.94	0.385	0.365
Medium	0.72	0.54	0.263	0.265
Hard	0.88	0.33	0.353	0.340

Precision stayed relatively high throughout the three categories (0.93, 0.72, 0.88), while recall dropped from 0.94 on Easy images down to 0.33 on Hard images, as displayed in Figure 2.

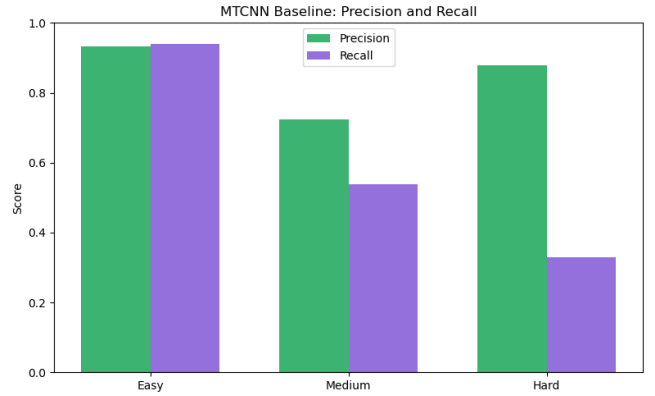


Fig. 2. Precision/Recall bar chart

Now, as for the inference speed on CPU and GPU, it was nearly identical for each of the categories, as displayed in Figure 3.

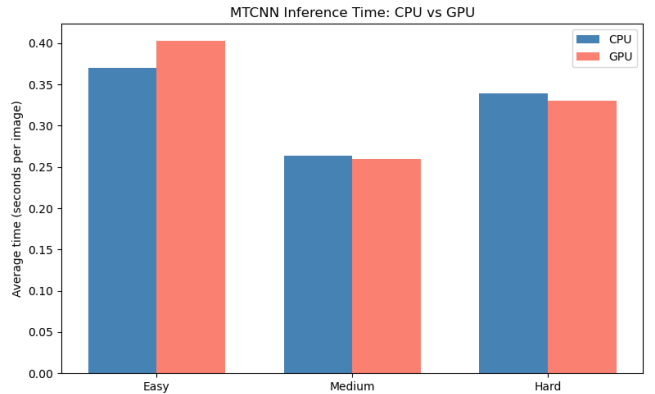


Fig. 3. CPU vs GPU inference speed bar chart

5.3 Transfer Learning: Soldier vs Civilian Classification

When testing the fine tuned MobileNetV2 model, the test accuracy without data augmentation was 68% and with augmentation it dropped down to 55%, shown in Table 2.

Table 2. MobileNetV2 fine-tuning results

Configuration	Test Accuracy	CPU Training (s)	GPU Training (s)
No Augmentation	68%	16.1	13.7
With Augmentation	55%	22.7	28.9

Comparing training speed, GPU was slightly faster than CPU without augmentation (13.7s vs 16.1s). With augmentation, however, this was reversed, with CPU training (22.7s) faster than GPU (28.9s).

The training and validation accuracy and loss curves for the model without augmentation (the more accurate one) are shown in Figure 4.

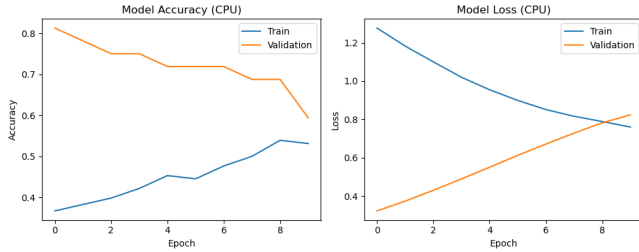


Fig. 4. Training and validation accuracy/loss curves

Training accuracy gradually increased over the 10 epochs from roughly 37% to 54%, while the validation accuracy actually decreased from approximately 81% to 59% for the same period. Training loss declined from about 1.27 to 0.76, while validation loss increased from 0.32 to 0.81, with the two curves crossing around epoch 8.

The examples of the augmented training images that were used in the augmentation experiment are shown in Figure 5.



Fig. 5. Augmentation example images

5.4 Cross Demographic Robustness

Table 3 below reveals the results from the six robustness experiments, where MobileNetV2 was trained on one demographic group and tested against a contrasting one.

Each of the six experiments produced accuracies below random chance (50%), with Male and Mixed performing worst at 5%, and Single and Crowd performing best at 38%.

Table 3. Cross-demographic robustness results

Experiment	Test Accuracy
Indoor → Outdoor	37%
Outdoor → Indoor	32%
Male → Mixed	5%
Mixed → Male	14%
Crowd → Single	28%
Single → Crowd	38%

6 Discussion

6.1 RQ1: MTCNN Precision and Recall Across Difficulty Levels

The results reveal that MTCNN’s precision stays relatively high for all three difficulty categories (0.93, 0.72 and 0.88). However, recall drops rapidly from 0.94 on Easy images to 0.33 on Hard images. This reveals that MTCNN does not get less confident or less accurate when it detects a face, instead it simply fails attempting to detect on a large proportion of faces in Hard images.

This is consistent with the EDA findings displayed in Section 5, where Hard images contained predominantly outdoor crowd scenes, where 19 of the 20 sampled images had crowds. The crowded scenes typically contain a larger amount of small and overlapping (sometimes occluded) faces which are inherently more challenging for the detection model to localise. This also aligns with the findings of Yang et al., who observed that face detection performance declines significantly as scene complexity increases [5]. The average precision falls below 30% on the WiderFace Hard subset for all four of the baseline methods they assessed.

Our results also align with Tariyal et al., who similarly discovered that MTCNN scores high on accuracy for face landmark detection, but may struggle to meet specific performance demands when compared to more modern detectors, such as YOLO and SSD [8]. The recall decline observed for these images suggests that MTCNN’s cascaded detection pipeline is quite accurate when faces are clearly visible, but it is not well suited to detecting a high density of small faces in crowd scenes.

6.2 RQ2: CPU vs GPU Speed Comparison

Inference speed measurements between CPU and GPU displayed minimal difference across all of the three difficulty categories, where times ranged between 0.263 and 0.385 seconds on CPU and between 0.265 and 0.365 seconds on GPU. This is a contrasting finding to the popular expectation that GPUs provide a significantly faster speed advantage for deep learning inference. However, it is an assumption that holds only for traditional hardware where CPUs and GPUs have separate dedicated memory.

This result can be explained by the unified memory architecture implemented for the Apple M1 chip used in this study, where CPU and GPU share the same physical memory pool instead of operating independently [7]. On common and traditional systems, the transfer of data between separate CPU and GPU memory introduces overhead that GPUs then compensate for through parallel processing,

which then results in a net speed advantage [12]. On M1 hardware, this overhead transfer is reduced due to the unified memory, which explains why there was no substantial speed difference observed [7]. In the transfer learning experiment (Section 5), the CPU and GPU training times were also close which reveals that this finding is consistent with a broad pattern seen in our experiments, and in one case CPU even outperformed GPU.

6.3 RQ3 Transfer Learning for Scene Classification

MobileNetV2 scored a 68% test accuracy on the soldier versus civilian classification task without data segmentation, while augmenting the data dropped the accuracy to 55%. These results suggest that transfer learning can in fact be applied to scene classification tasks, but performance is also highly sensitive to dataset size and pre-processing choices.

Another important finding from this experiment was the difference in performance when faces were cropped and when whole images were used as input. The initial experiments with cropped faces scored only 43% for accuracy, while switching to whole images improved it to 68%, suggesting that the contextual information around the face, such as uniform, equipment, and environmental specifics, carries discriminative information than just the face. Intuitively, differentiating a soldier from a civilian relies mainly on visual cues outside just the face, which are removed after it has been cropped.

The training curve shown in Section 5 reveals a clear pattern of overfitting, where training accuracy is gradually increasing while validation accuracy declines over the same period. This is a well documented drawback of training on small datasets [13]. This aligns with the fact that the dataset used for this experiment was small, consisting of only 160 training images, which is considerably smaller than the thousands of images typically used in transfer learning experiments. The relatively high initial validation accuracy of about 81% at epoch 0 suggests that MobileNetV2’s already pre-trained ImageNet features already contained useful information before fine-tuning, which then was degraded as training resumed on a smaller noisy dataset.

The reduction in accuracy with data augmentation, from 68% to 55%, supports the conclusion that dataset size was a limiting factor. Despite the common expectation that data augmentation generally improves generalisation by exposing the model to more variety of training examples, its effectiveness relies on having a sufficiently large dataset to begin with. With only 160 images, the additional variation introduced by augmentation made it more challenging to learn consistent decision boundaries instead of improving generalisation.

6.4 RQ4 Cross-Demographic Robustness

The robustness experiments displayed that MobileNetV2 showed poor results when distinguishing one demographic group from another, with all six experiments producing test accuracies lower than random chance of 50%. Accuracies ranged from 5% (Male vs Mixed) to 38% (Single vs Crowd), clearly displaying that the model performed worse than random guessing in every comparison.

This aligns with the broader concern that models trained on narrow or non-diverse data perform worse when applied to different distributions, a pattern also seen in demographic bias research in face-related machine learning tasks [6]. The consistently poor performance for each of the six group comparisons, regardless of which attribute was used, suggests that it is not a sole issue specific to one type of demographic split but rather a general pattern when training and test distributions differ substantially.

The results highlight the important role that the size of the dataset plays in affecting model robustness. Even with the benefit of using pre-trained ImageNet features, training on as few as 18 to 40 images was insufficient to produce a model that was capable of distinguishing the contrasting group reliably. This suggests that dataset size could be just as important and influential as the choice of attribute itself in determining how well the model performs. A model that was trained on a small homogeneous sample tends to learn narrow specific patterns instead of transferable features, regardless of which attribute is changed. This reinforces the importance of both data diversity and sufficient sample size as complementary requirements for building a robust system for face related classification.

7 Conclusion

This study explored the practical limitations of using pre-trained face detection models in real world conditions, evaluating MTCNN on the WiderFace dataset as a case study. Four experiments were carried out: a baseline evaluation of MTCNN’s precision and recall for three image difficulty levels, a comparison of CPU and GPU inference and training speed on Apple’s M1 hardware, a transfer learning experiment using MobileNetV2 for soldier-against-civilian scene classification, and a series of cross-demographic robustness experiments.

The results reveal that MTCNN displayed a high precision rate for all difficulty levels but suffered a drop in recall for the Hard image category, from 94% on Easy to 33% on Hard, which can be explained by a predominance of outdoor crowd scenes in the Hard category. CPU and GPU inference and training speeds were both found to be comparable on Apple’s M1 hardware, attributed to the unified memory in the chip’s architecture. Transfer learning using MobileNetV2 achieved 68% test accuracy for the classification of soldier vs civilian scenes without data augmentation. However, when data was augmented the accuracy dropped to 55%, the reason being that the added variation on a small dataset introduced more noise instead of useful generalisation. An important methodological finding was that whole images outperformed cropped face images, with the main reason being that cropped images lost contextual information and environmental specifics. Finally, robustness experiments showed that models that were trained only on one demographic group performed poorly when distinguishing that group from others, with all six comparisons producing accuracies below random chance, revealing an important finding about the critical importance of dataset size in model robustness alongside data diversity. Future work with larger and more diverse datasets would aid in clarifying how much of the robustness problem comes from the dataset size versus lack of diversity.

Together, these findings highlight the practical limitations of deploying pre-trained face detection models in practical conditions and display that transfer learning, while a useful tool for specific domain classification, requires careful consideration for the dataset size and composition to be effective.

References

- [1] I. D. Mienye and T. G. Swart. 2024. A comprehensive review of deep learning: Architectures, recent advances, and applications. *Information* 15, 12 (2024), 755. <https://doi.org/10.3390/info15120755>
- [2] S. K. Rath and S. S. Rautaray. 2014. A survey on face detection and recognition techniques in different application domain. *International Journal of Modern Education and Computer Science* 6, 8 (2014), 34–44. <https://doi.org/10.5815/ijmecs.2014.08.05>
- [3] M. Hassaballah and S. Aly. 2015. Face recognition: Challenges, achievements and future directions. *IET Computer Vision* 9, 4 (2015), 614–626. <https://doi.org/10.1049/iet-cvi.2014.0084>
- [4] N. Singhal, N. Gupta, and A. K. Singh. 2025. Facial detection in computer vision: Bridging gap between CNN, haar cascade and MTCNN. *Proceedings on Engineering Sciences* 7, 1 (2025), 427–436. <https://doi.org/10.24874/PES07.01C.009>
- [5] S. Yang, P. Luo, C. C. Loy, and X. Tang. 2016. WIDER FACE: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5525–5533. <https://doi.org/10.1109/CVPR.2016.596>
- [6] Y. Yang, A. Gupta, J. Feng, P. Singhal, V. Yadav, Y. Wu, P. Natarajan, V. Hedau, and J. Joo. 2022. Enhancing fairness in face detection in computer vision systems by demographic bias mitigation. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society (AI/ES '22)*. Association for Computing Machinery, 813–822. <https://doi.org/10.1145/3514094.3534153>
- [7] A. Benazir and F. X. Lin. 2025. Profiling large language model inference on Apple Silicon: A quantization perspective. *arXiv preprint arXiv:2508.08531* (2025). <https://arxiv.org/abs/2508.08531>
- [8] S. Tariyal, R. Chauhan, Y. Bijalwan, R. Rawat, and R. Gupta. 2024. A comparative study of MTCNN, Viola-Jones, SSD and YOLO face detection algorithms. In *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. 1–7. <https://doi.org/10.1109/iitcee59897.2024.10467445>
- [9] S. Hussain et al. 2022. Face mask detection using deep convolutional neural network and MobileNetV2-based transfer learning. *Wireless Communications and Mobile Computing* 2022 (2022), Article 1536318. <https://doi.org/10.1155/2022/1536318>
- [10] Ning Zhang, Wuqi Gao, and Junmin Luo. 2020. Research on face detection technology based on MTCNN. In *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. IEEE, 154–158. <https://doi.org/10.1109/ICCNEA50255.2020.00040>
- [11] Qian Xiang, Guoling Zhang, Xiaodan Wang, Jie Lai, Rui Li, and Qingshuang Hu. 2019. Fruit image classification based on MobileNetV2 with transfer learning technique. In *Proceedings of the 2019 3rd International Conference on Computer Science and Application Engineering (CSAE 2019)*. Association for Computing Machinery, Article 17. <https://doi.org/10.1145/3331453.3361658>
- [12] T. Dettmers. 2015. 8-bit approximations for parallelism in deep learning. *arXiv preprint arXiv:1511.04561* (2015). <https://arxiv.org/abs/1511.04561>
- [13] S. Somvanshi, S. Das, S. A. Javed, G. Antariksa, and A. Hossain. 2024. A survey on deep tabular learning. *arXiv preprint arXiv:2410.12034* (2024). <https://arxiv.org/abs/2410.12034>

Appendix A Sample Images by Difficulty Category

This appendix presents sample images from each of the three WiderFace difficulty categories used in the exploratory data analysis described in Section 5.



Fig. 6. Easy images sample grid

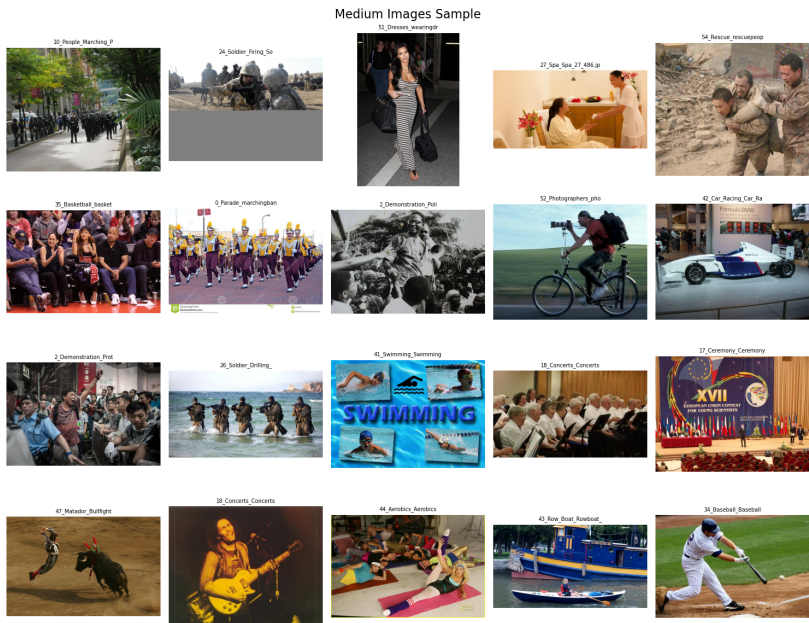


Fig. 7. Medium images sample grid

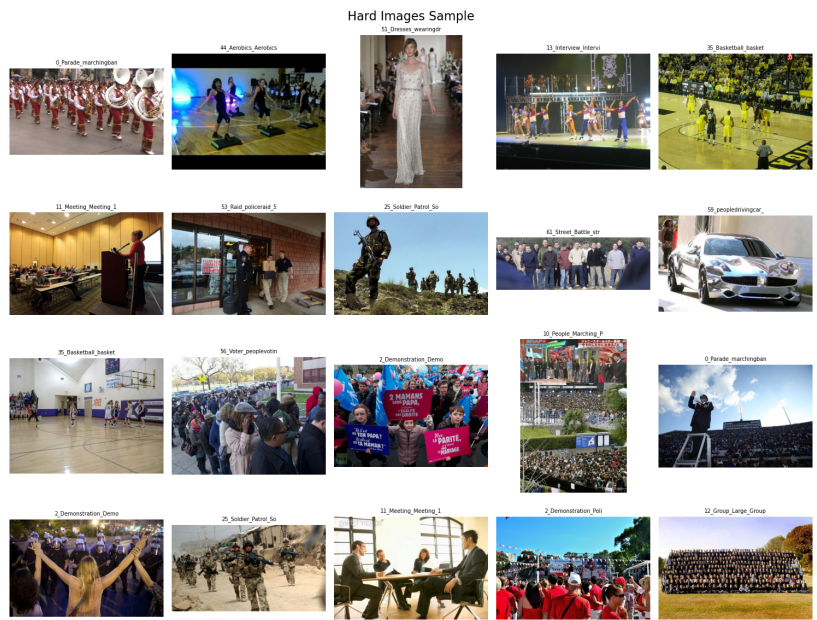


Fig. 8. Hard images sample grid