Anonymization of Images for Privacy Protection on Embedded Systems

OLIVER HNAT, University of Twente, The Netherlands

As small, camera-equipped devices become increasingly common, they often capture images containing faces and other personal details. This raises significant privacy concerns, especially under regulations such as the European Union's General Data Protection Regulation (GDPR), which require that such sensitive data be anonymized. However, implementing strong privacy safeguards directly on these devices is challenging due to their limited computing power.

In our research, we explore the trade-off between computational efficiency and privacy protection by implementing three lightweight anonymization techniques, namely masking, pixelation, and blurring, directly on a resource-constrained embedded platform, the ESP32-P4. Each method is evaluated for its execution time and its effectiveness at protecting identity, the latter measured using cosine similarity scores derived from the DeepFace Python library. By analyzing the performance and privacy impact of these techniques, our work aims to uncover practical strategies for real-time, on-device anonymization. This enables privacy-preserving image capture at the edge, without the need for potentially insecure cloud processing.

Our findings show that all three anonymization techniques can be executed in under 2 microseconds on the ESP32-P4, making them highly suitable for real-time processing. The full anonymization pipeline, including face detection, operates at $\tilde{2}0$ milliseconds per frame, enabling throughput up to 50 FPS. While detection accuracy reached 59.2% across a diverse dataset of 200 images, the main performance bottleneck lies in the face detection model, not in the anonymization methods themselves. These results confirm the viability of real-time, edge-based visual anonymization on constrained embedded systems

Additional Key Words and Phrases: Embedded systems, image anonymization, privacy protection, pixelation, masking, blurring, ESP32-P4, resource-constrained devices

1 INTRODUCTION

The growing use of embedded devices such as Arduino-based cameras raise critical concerns about privacy. The captured raw images may contain personally identifiable information, including faces or other personal details. Under regulations such as the European Union's General Data Protection Regulation (GDPR), such data must be protected before storage, transmission, or analysis, as unauthorized use of such images can lead to identity theft, surveillance, or discrimination [3] [6].

Given the computational and memory constraints of embedded systems, implementing a privacy preserving mechanism on the device is challenging [2], while cloud-based systems offers high-performance processing, it poses additional privacy threats during data transmission, increases latency and may conflict with regulations that require sensitive data to be processed locally.

This research addresses the gap by developing and evaluating lightweight, on-device anonymization methods, suitable for embedded systems. By benchmarking masking, pixelation, and blurring methods directly on the ESP32-P4, this study aims to provide practical, regulatory-compliant solutions that enhance user privacy without affecting system performance.

2 PROBLEM STATEMENT

This research proposes to explore and evaluate simple anonymization algorithms that can be efficiently executed on low-power embedded devices such as ESP32-P4. The goal is to protect user privacy by ensuring that images captured by cameras no longer reveal identifiable information, while still keeping the body posture and rest of the image unchanged.

2.1 Research Question

The main research question that will lead the research is:

Which simple anonymization techniques (e.g. blurring, masking, pixelation) provide an optimal balance between privacy protection and computational efficiency for usage on resource-constrained embedded devices such as Arduino-based cameras?

This can be answered with the following sub questions:

(1) How does each anonymization technique impact execution time on the ESP32-P4?
(2) How effectively does each anonymization technique prevent visual identification of individuals?

In the spirit of open source and reproducible research, all code used in this project is available on GitHub.[1].

3 RELATED WORK

While there has not been much previous research on directly building anonymization strategies on embedded devices, there is a substantial body of work focused on anonymization techniques in general. These range from traditional image processing methods to advanced neural networks, each with a different ratio of anonymization and computational load.

Pixelation is one of the most widely used anonymization techniques due to its simplicity and the ability to preserve the general facial structure and motion cues in video. [12] It is commonly applied in both consumer products and surveillance contexts to provide visual context without compromising identity.

However, recent advances in deep learning have exposed critical flaws in this method. Research introduces a neural framework capable of reconstructing high-resolution face images from pixelated video. [12] This system is able to reverse coarse pixelation (as low as 16x16 or 8x8), recovering facial expressions and identifying features with accuracy. This work demonstrates that even heavily pixelated video does not guarantee anonymity and that pixelation alone is insufficient as a privacy-preserving method.

TScIT 43, July 4, 2022, Enschede, The Netherlands

[1] https://github.com/olinpin/image-anonymization-on-ES

Blurring is another commonly used anonymization strategy that reduces image detail by applying a Gaussian blur to facial regions. It balances privacy and visual context better than masking, as some facial layout and expressions remain partially visible, making it useful in surveillance footage, journalism, or broadcast media where identity suppression is needed without entirely obscuring the subject. However as with pixelation, blurred faces can be partially reconstructed using modern neural deblurring models [8], so blur alone may not provide sufficient protection against identification, particularly if the blur kernel is weak or small [8].

And last but not least, masking is one of the most straightforward anonymization techniques, typically implemented by overlaying a solid color (black in our case) rectangle over the detected face. Its simplicity makes it extremely efficient, especially on resource-constrained platforms, and unlike pixelation or blurring, it does not rely on visual transformation but rather on full occlusion of sensitive regions. As such, it offers strong guarantees against identity reconstruction attacks, since no facial information is preserved. This method has been widely used in privacy-sensitive applications such as pedestrian detection datasets [10]. User studies have shown that while masking provides the highest level of privacy protection, it is often perceived as visually intrusive and less informative compared to other techniques such as pixelation and blurring [1]. Despite this, masking remains the most secure among the lightweight anonymization methods, particularly when privacy is prioritized over visual fidelity.

On the other end of the spectrum, **DeepPrivacy2** [5], a full-body anonymization framework using generative adversarial networks. DeepPrivacy2 achieves anonymization using high visual realism through multiple specialized generative models, ensuring strong privacy even against re-identification attacks. While DeepPrivacy2 sets a good benchmark in realistic anonymization, its computational requirements make it unsuitable for low-power embedded platforms like ESP32-P4.

These contrasting approaches highlight the need for a middle ground: anonymization methods that are secure, yet feasible to deploy on resource-limited embedded systems. This thesis aims to evaluate such methods systematically on an Arduino-based platform.

Furthermore, privacy-preserving techniques on resource constrained hardware have been studied in the context of lightweight cryptography, which shares similar challenges with image anonymization on embedded devices.

While no standard exist for image anonymization on embedded devices, we aim to address this gap by comparing different methods. Recent benchmarking of lightweight cryptographic algorithms on microcontrollers, such as AES-128, SPECK, and ASCON, shows that authenticated encryption techniques can achieve security with low latency, minimal memory usage, and energy efficiency on resource constrained hardware like Arduino Nano and Micro [9]. The results highlight that SPECK and ASCON represent strong candidates for securing data on low-cost, low-power hardware. At the same time, NIST's selected ASCON as the official standard for lightweight cryptography, confirming its suitability for resource constrained hardware [7].

4 METHODOLOGY

The goal of this research is to evaluate the feasibility and effectiveness of simple, lightweight image anonymization techniques on resource-constrained embedded platforms. Namely, the ESP32-P4 environment is targeted, where memory and processing capabilities are limited, making real-time anonymization challenging.

4.1 Evaluation Framework

Success will be evaluated based on these criteria:

(1) **Privacy Effectiveness**: Assess how well each anonymization method prevents identity recognition using *Identity Similarity* (IS) from the `DeepFace` Python library [11]
(2) **Time Feasibility**: Measure execution time on the ESP32-P4 platform.

These criteria are used to determine whether a given method has a balance between preserving privacy and running on the system.

4.1.1 Dataset Selection. In order to evaluate the face detection and anonymization under realistic conditions, we use the Face Detection Framework (FDF) dataset [4]. The FDF dataset is widely used in the research community, publicly available, and free for academic purposes. It contains a diverse set of real-world face images across varying lighting conditions, skin tones, poses, and backgrounds, making it well-suited for stress-testing detection and anonymization techniques on embedded hardware. Furthermore, using a standard and openly accessible dataset enhances the reproducibility and comparability of the results, in line with the open-source goals of this project.

From the FDF dataset [4], 50 images were randomly selected for evaluation. To ensure meaningful evaluation, images where faces were completely unrecognizable, such as those turned away from the camera or heavily obscured by objects like scuba gear, were manually excluded and replaced with new randomly selected samples. This filtering ensured that the benchmark focused on images where a human observer would reasonably expect face detection and anonymization to be applicable.

4.2 Lightweight Techniques on Arduino

Three traditional anonymization techniques will be implemented directly on the ESP32-P4 platform:

- **Masking**: Replacing detected face regions with a solid black rectangle.
- **Pixelation**: Using different pixelation grid sizes (3x3, 5x5, 7x7), for each grid in the face, apply the center pixel's color over the rest of the grid.
- **Blurring**: Applying a simple Gaussian to obscure facial features. [13]

These techniques were chosen specifically for their combination of computational efficiency and real-time feasibility. Unlike advanced deep learning-based anonymization systems, which require GPUs or cloud processing, these methods are simple enough to be executed within milliseconds on the ESP32-P4 and feasible in the scope of this paper, yet still capable of substantially reducing face identifiability. Their use in prior work and public applications

further supports their relevance as benchmark techniques for embedded anonymization.

4.3 Measurements and Analysis

Each method will be evaluated based on the following metrics:

- **Execution Time**: Average time taken to detect a face and apply anonymization on a single frame.
- **Privacy Effectiveness**: `DeepFace` Python library [11] is used to compute identity similarity scores between original and anonymized images. DeepFace extracts facial embeddings and computes cosine similarity. Lower similarity values indicate stronger anonymization.

All anonymized images generated and recorded on ESP32-P4 will be saved and analyzed offline on a laptop for identity similarity using DeepFace. This provides a consistent privacy benchmark across all methods while remaining computationally feasible.

4.4 Hardware constraints

Given the computational constraints, the ESP32-P4 implementation is limited to frame-by-frame anonymization, without any temporal coherence or advanced synthesis. The ESP32-P4 is also constrained by available resources. Furthermore, unlike DeepPrivacy2, we do not attempt to protect against re-identification through body features, or clothing.

4.5 System Overview

Figure 1 shows the complete processing pipeline used in this project. It starts from the input image and proceeds through decoding, resizing, neural network inference, and ends with one of the anonymization methods. This flowchart corresponds directly with the technical implementation on the ESP32-P4 platform and provides context for the execution time breakdowns and processing stages discussed in the following sections.

5 RESULTS

This section presents the performance and privacy effectiveness of the three above mentioned anonymization techniques (masking, pixelation, and blurring) when deployed on the ESP32-P4 platform. Each method was evaluated using the execution time and identity similarity score as described in the methodology.

5.1 Face Detection Pipeline Breakdown

The face detection process on the ESP32-P4 consists of three stages:

- **Preprocessing:** The input image is resized and normalized to 120×120 pixels, matching the expected input of the neural network.
- **Inference:** The resized image is passed through a quantized neural network that predicts face bounding boxes and keypoints.
- **Postprocessing:** the system sets scaling factors to map coordinates back to the original image, filters out detections below confidence threshold of 0.5, and returns the final list of detected faces.

A breakdown of the face detection pipeline reveals the proportion of time spent in each stage. On average:

- Preprocessing took approximately **5,244 µs** ($\tilde{2}$5% of total)
- Model inference required **14,521 µs** ($\tilde{7}$4%)
- Postprocessing took just **174 µs** ($\tilde{1}$%)

This shows that most of the execution time is spent running the neural network itself, with preprocessing also contributing significantly due to the need to resize and normalize the input to 120×120 pixels. Postprocessing overhead is negligible.

5.2 Face Detection Time

Before anonymization, a lightweight face detection model was executed on the ESP32-P4 to locate the person's face. The model was evaluated on three different test images:

- **Image 1:** White male
- **Image 2:** Black male
- **Image 3:** Group photo with multiple faces

Each image was processed 50 times to obtain average execution times and standard deviations. Results are summarized in Table 1.

Table 1. Face Detection Execution Time (µs), Averaged over 50 Runs

Image	Resolution	Avg. Time (µs)	Std. Dev. (µs)
White male	522×526	20,091.41	11.92
Black male	433×350	19,939.98	11.16
Multiple faces	336×300	22,301.62	7.92

Face detection performed consistently across the single-face images, with average execution times around 20 milliseconds and very low variance. The group photo (336×300), despite containing four visible faces, required slightly more time (22.3 ms) and exhibited even lower variance. However, only two of the four faces were successfully detected. This reduced accuracy is likely due to the low resolution and small size of individual faces, which poses challenges for lightweight detection models deployed on embedded systems.

5.3 Anonymization Execution Time

After face detection, each image was anonymized using one of three techniques: masking, pixelation (grid sizes are chosen dynamically based on face size), or blurring. All three methods were implemented directly on the ESP32-P4 platform and executed immediately after face detection.

Each anonymization technique was applied to all three test images. Execution times were measured in microseconds using the built-in timing functionality on the ESP32-P4. The measurements were repeated multiple times and showed no observable variation across runs or image types. All anonymization algorithms were written in-place and modified each pixel in the detected face area once.

All three techniques completed in approximately 2 microseconds per frame. This indicates that their computational cost is negligible relative to face detection, which had a much larger per-frame processing time. The consistency across methods also suggests that

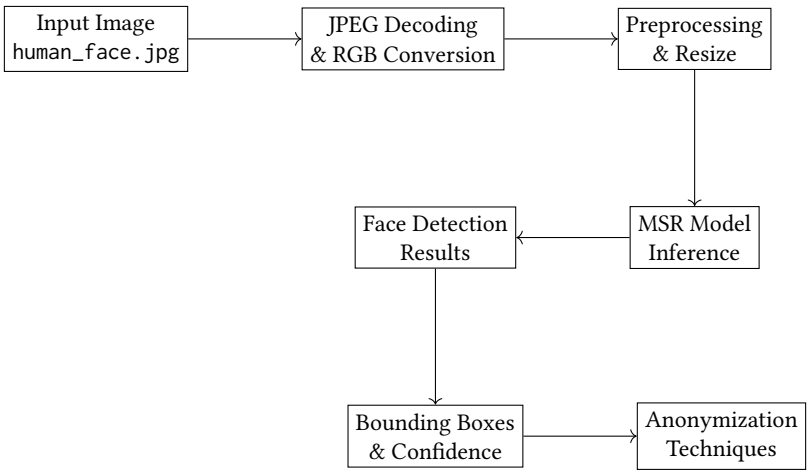

Fig. 1. System pipeline from image input to anonymized output.

Table 2. Anonymization Execution Time (µs)

Technique	Avg. Time (µs)	Std. Dev. (µs)
Masking	2	0
Pixelation	2	0
Blurring	2	0

image size or content had little impact on performance in this configuration.

5.4 Privacy Effectiveness

To find out how well each anonymization method protected identity, we employ face-level verification using the DeepFace library with the Facenet model. Cosine similarity is computed between the embeddings of faces in original and anonymized images. A face was considered *protected* if its similarity was below a threshold of 0.4, or if no corresponding face was detected in the anonymized image.

Three image categories were used:

- **White male** – 522×526 resolution, 1 face
- **Black male** – 433×350 resolution, 1 face
- **Multiple faces** – 336×300 resolution, 4 faces

Table 3 summarizes the protection rates for each technique:

Table 3. Percentage of Protected Faces per Anonymization Method

Image	Masking (%)	Pixelation (%)	Blurring (%)
White male	100.0	100.0	100.0
Black male	100.0	100.0	100.0
Multiple faces	50.0	50.0	50.0

All three techniques achieved **full protection (100%)** in the single-face images, with no faces successfully verified after anonymization. In the group image, which contained four visible faces, only **two faces were protected**, resulting in a 50% protection rate across all methods.

This reduction was due to the failure of the embedded face detection model, that was running directly on the ESP32-P4. It detected only two of the four faces in the original image. Then the anonymization methods were applied to only those two detected faces, leaving the remaining faces unmodified and recognizable. This highlights a limitation: the overall effectiveness of privacy protection is constrained by the accuracy and completeness of the face detection stage.

5.5 Batch Testing on FDF Dataset

To improve statistical robustness, the evaluation was expanded to a larger dataset. Four batches of 50 images (300x300 px) were selected from the public Face Detection Framework dataset (FDF) [4]. Each image was passed through the full anonymization pipeline using the ESP32-P4, and detailed timing metrics were recorded.

Table 4. Average Execution Time per Stage (ESP32-P4, 4x50 Images)

Stage	Time (ms)	Percentage
Preprocessing	4.88	24.4%
Inference	14.63	73.2%
Postprocessing	0.14	0.7%
Anonymization	0.30	1.6%
Total	**19.97**	**100%**

As Table 4 shows, the timing analysis over 4x50 (different) images confirms that the ESP32-P4 performs consistently across the anonymization pipeline, with an average time of 19.97 milliseconds per image. The majority of this time, approximately 73.2%, is spent on neural network inference, preprocessing steps such as image resizing and RGB conversion take 24.4% of total time. Postprocessing is minimal at just 0.7%, and the anonymization step adds only 1.6%. This shows that lightweight anonymization (masking,

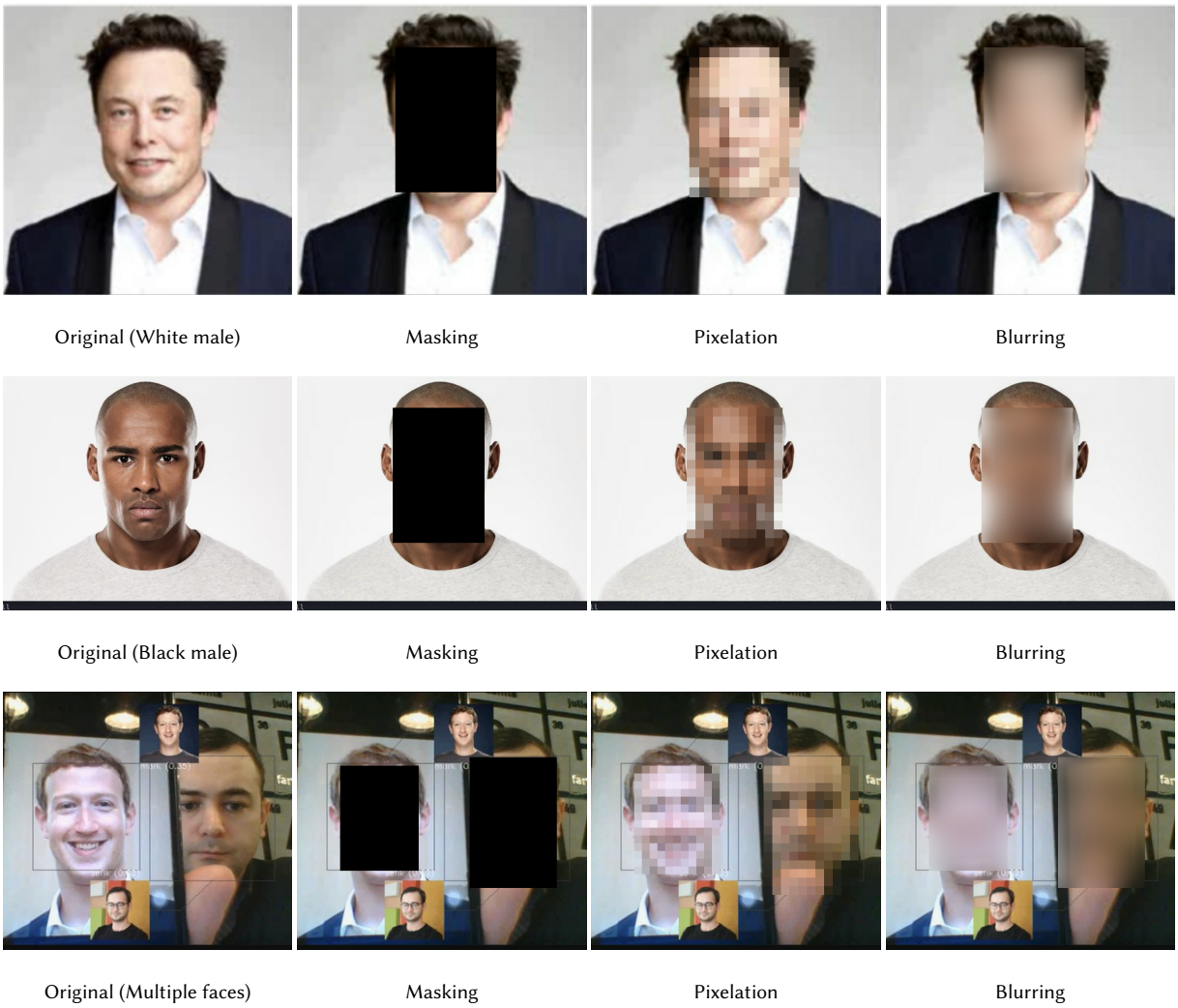

Fig. 2. Visual comparison of anonymization techniques applied to three test images. Top: White male; Middle: Black male; Bottom: group photo.

pixelation, blurring) is computationally inexpensive. Despite the limited resources of the embedded platform, the system maintained a throughput of $\tilde{5}0$ frames per second for the full batch, and $\tilde{6}8$ FPS when considering inference alone. The variance in total execution time across the 200 tests was only 2.1%, which indicatese stable and reliable performance. In terms of detection accuracy, 59.2% of the faces were successfully identified, resulting in a 59.2% detection rate. This highlights that while the system is highly optimized for speed, particularly on diverse or low-resolution images detection requires further improvement.

As shown in Table 5, the batch evaluation across 200 anonymized images reveals that masking achieved the highest privacy protection, successfully anonymizing 75.0% of detected faces. Pixelation followed closely with a protection rate of 65.0%, demonstrating moderate effectiveness. Blurring, however, performed significantly worse, protecting only 37.5% of faces, with over 60% still recognizable.

Table 5. Batch Privacy Protection Effectiveness (200 Images, ESP32-P4)

Method	Protected Faces (%)	Recognizable Faces (%)
Pixelation	65.0	35.0
Masking	75.0	25.0
Blurring	37.5	62.5

These results suggest that, despite its simplicity and visual intrusiveness, masking remains the most reliable lightweight anonymization method on embedded systems. Full facial obfuscation offers stronger resistance against identity recognition algorithms like DeepFace, especially when compared to partial obfuscation techniques such as pixelation or Gaussian blur.

5.6 Total Processing Time and Real-Time Feasibility

Combining the face detection time (approximately 20,000 µs) with the anonymization time (2 µs), the total per-frame processing time is around 20 milliseconds. This shows that the system could theoretically process up to 50 frames per second, making it suitable for real-time image anonymization on live camera input with frame rates up to approximately 50 FPS.

6 DISCUSSION

This study set out to evaluate the trade-offs between privacy protection and computational feasibility for image anonymization techniques on resource-constrained embedded systems. The results provide several key insights.

6.1 Anonymization Performance vs. Cost

This section will try to address the research question *How does each anonymization technique impact execution time on the ESP32-P4?*

All three anonymization techniques, masking, pixelation, and blurring, achieved nearly instantaneous execution on the ESP32-P4, each taking approximately 2 microseconds per frame. Despite differences in algorithmic complexity, their runtime was negligible compared to face detection, which required approximately 20 milliseconds per frame.

This finding confirms that basic anonymization methods are feasible for real-time execution on low-power devices. With a total processing time per frame of around 20 ms, the system can support camera frame rates of up to 45–50 FPS, meeting real-time performance standards for video streaming or live capture.

6.2 Privacy Effectiveness and Resolution Dependence

The following section will try to answer the research question *How effectively does each anonymization technique prevent visual identification of individuals?*

In the 200 test images, masking prevented identity recognition by the DeepFace model 75% of the time, which was the best out of all three methods, pixelation was closely behind with 65% and blurring performed the worst with only masking 37.5% of the faces. This highlights the effectiveness of using masking as a lightweight technique for on-device anonymization.

In the multiple-face image, only 50% of faces were protected across all methods. Importantly, this was not due to failures in the anonymization algorithms themselves. Instead, the embedded face detection model detected only 2 out of 4 faces in the group photo. Since anonymization was applied only to detected regions, the other faces remained untouched and were successfully recognized.

This highlights a critical finding: the overall effectiveness of privacy protection is highly dependent on the quality of face detection. Even flawless anonymization is insufficient if it is not applied to all sensitive regions.

6.3 Impact of Resolution and Face Size

Out of the 3 test images, the group image had the lowest resolution (336×300), and the undetected faces were notably smaller. This suggests that small or distant faces may not be reliably detected by lightweight models under constrained hardware settings. Thus, while anonymization performance was excellent in isolated cases, its robustness in complex, real-world scenes remains limited by face detection capability. Additionally, the face detection model requires all input images to be 120×120 pixels. So the images are scaled down during preprocessing. This downscaling is necessary but can severely reduce detail, especially for small or distant faces, making them harder to detect accurately.

6.4 Limitations

Several limitations influenced the validity and scope of this study. First of all a key limitation of the current implementation is the fixed input resolution of 120×120 pixels for the face detection model. All images must be downscaled to fit this input size before inference, which can result in loss of detail. This loss may affect the model's ability to extract facial features for accurate detection, especially when faces are not centered or occupy only a small portion of the image. So the preprocessing step may reduce detection reliability, affecting the overall success of anonymization.

Secondly, the face detection model for the ESP32-P4 is intentionally small to fit within the resource constraints. While this allows the system to operate without any outside (cloud) interactions, it also limits the face detection abilities. This constraint shows a larger limitation in AI in embedded systems: the balance between performance and resource efficiency.

Lastly, although the DeepFace framework [11] was used to grade the identity protection, the analysis was conducted using a single model (Facenet), without comparison to other identity verification systems. While Facenet is widely used, different recognition models vary in sensitivity to anonymization. It is possible that some anonymized faces deemed "protected" in this study may still be partially identifiable by more advanced or differently trained recognizers (such as specifically trained machine learning models [8] [12]. This limitation suggests that privacy protection results should be interpreted as a lower bound, and future work should consider multiple verification models to better estimate the strength of each anonymization technique.

6.5 Future Work

For future research there are several interesting areas to explore. This includes integrating more advanced or cascaded face detection models that offer better detection accuracy while maintaining low latency. The current model is highly optimized for speed, so it may miss subtle or less prominent faces, especially when they appear under challenging conditions or at varying scales. More sophisticated models could improve coverage without significantly increasing computational cost. This would help to ensure that all sensitive regions are identified and anonymized, enhancing overall system reliability.

Another area involves testing the anonymization pipeline on a wider range of image conditions. The current study used the FDF dataset [4], which provides a strong baseline for facial diversity, but future evaluations should include more complex scenarios such as surveillance footage, traffic camera data, protests or group gatherings. These conditions are more representative of real-world deployments and would help uncover limitations or failure modes

that are not evident in controlled testing. Expanding the evaluation scope would also inform how the system performs across domains with varying resolution, lighting, and movement patterns.

Finally, deploying and testing the anonymization system in real-time conditions—such as continuous camera streams or interactive embedded applications—would be a critical next step. While current evaluations simulate real-time performance through static image processing, actual deployment scenarios may introduce additional challenges, including buffer management, power constraints, and interaction with other system components. Testing the anonymization pipeline under these operational conditions would validate its robustness, guide hardware-specific optimizations, and potentially highlight integration issues not visible during offline benchmarking.

- Integrating more advanced or cascaded face detectors that maintain low latency but improve detection coverage.
- Testing anonymization under more diverse image conditions (e.g., surveillance, traffic, group events, occlusions).
- Testing anonymization in real-time conditions

6.6 Conclusion

The study demonstrates that lightweight anonymization methods such as masking, pixelation, and blurring are computationally viable for real-time use on embedded platforms. When face detection is accurate, these methods are capable of offering strong privacy protection. However, the overall system effectiveness is fundamentally limited by the quality of face detection.

Furthermore, prior research suggests that pixelation and blurring, although visually anonymized, may still be vulnerable to machine learning-based reconstruction attacks [12]. Masking, despite being visually more intrusive, offers a stronger privacy guarantee in high-risk scenarios.

REFERENCES

[1] Pascal Birnstill, Sibylle Haider, and Stefan Winkler. 2015. A User Study on Anonymization Techniques for Smart Video Surveillance. In *2015 International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–6. https://doi.org/10.1109/AVSS.2015.7301760

[2] Luigi Coppolino, Salvatore D'Antonio, Giuseppe Mazzeo, and Luigi Romano. 2019. A comprehensive survey of hardware-assisted security: From the edge to the cloud. *Internet of Things* 6 (2019), 100055. https://www.sciencedirect.com/science/article/pii/S2542660519300101

[3] Zekeriya Erkin, Michael Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. 2009. Privacy-preserving face recognition. In *Privacy Enhancing Technologies Symposium (PETS)*. Springer, 235–253. https://homepage.tudelft.nl/c7c8y/SSP/PrivacyPreservingFaceRecognition.pdf

[4] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. 2019. DeepPrivacy: A Generative Adversarial Network for Face Anonymization. In *Advances in Visual Computing*. Springer International Publishing, 565–578.

[5] Håkon Hukkelås and Frank Lindseth. 2023. DeepPrivacy2: Towards Realistic Full-Body Anonymization. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 1329–1338. https://doi.org/10.1109/WACV56688.2023.00138

[6] Mukuka Kangwa, Charles S. Lubobya, and Jackson Phiri. 2021. Protection of Personally Identifiable Information and Privacy via the use of Hardware and Software. In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*. Hong Kong. https://www.iaeng.org/publication/IMECS2021/IMECS2021_pp75-81.pdf

[7] Jasmin Kaur, Alvaro Cintas Canto, Mehran Mozaffari Kermani, and Reza Azarderakhsh. 2023. A Comprehensive Survey on the Implementations, Attacks, and Countermeasures of the Current NIST Lightweight Cryptography Standard. arXiv:2304.06222 [cs.CR] https://arxiv.org/abs/2304.06222

[8] Ryan McPherson, Reza Shokri, and Vitaly Shmatikov. 2016. Defeating Image Obfuscation with Deep Learning. *arXiv preprint arXiv:1609.00408* (2016).

[9] Indu Radhakrishnan, Shruti Jadon, and Prasad B. Honnavalli. 2024. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors* 24, 12 (2024). https://doi.org/10.3390/s24124008

[10] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. 2017. JAAD: Joint Attention in Autonomous Driving Dataset. Available: https://github.com/ykotseruba/JAAD.

[11] Sefik Ilkin Serengil and Alper Ozpinar. 2020. DeepFace: A Lightweight Face Recognition and Facial Attribute Analysis Framework for Python. https://github.com/serengil/deepface

[12] Maayan Shuvi, Noa Fish, Kfir Aberman, Ariel Shamir, and Daniel Cohen-Or. 2020. Neural Alignment for Face De-pixelization. arXiv:2009.13856 [cs.CV] https://arxiv.org/abs/2009.13856

[13] Yang Yang, Yiyang Huang, Ming Shi, Kejiang Chen, and Weiming Zhang. 2023. Invertible mask network for face privacy preservation. *Information Sciences* 629 (2023), 566–579. https://doi.org/10.1016/j.ins.2023.02.013