# EfficientViTGuard: Real-Time Face Detection and Blurring for Video Privacy

**Dev Chheda, Divya Nori, Anirudh Valiveru**
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{dchheda, divnor80, anirudhv}@mit.edu

## Abstract

In this paper we present ***EfficientViTGuard***, a system designed to blur faces in real-time using on-device inference to ensure privacy in recorded videos. Due to the wide proliferation of smartphones, daily risks against our personal privacy only increases with each passing day. With this in mind, *EfficientViTGuard* is a fully-integrated system meant to run directly on the camera's hardware, using *EfficientViT-SAM* with a pruned version of the *MTCNN* face-detection model to blur facial pixels at each time step throughout an input video. Results regarding the accuracy and time-efficiency of our method are included in this work.

## 1 Introduction

In an era dominated by high-resolution imaging and pervasive video recording, the concern of individual privacy looms larger than ever. The ubiquitous nature of surveillance cameras, coupled with the ease of sharing visual content on various platforms, poses a substantial risk to personal privacy. In scenarios where video content is being streamed or shared live, real-time face blurring is critical as a preventive measure against unauthorized sharing of sensitive information. By immediately obscuring faces, the likelihood of unintentional or malicious dissemination of identifiable information is minimized.

Recently, deep learning-based computer vision models have made strides in the accuracy of face detection and segmentation. For example, a model with a cascaded structure of carefully designed deep convolutional networks that predict face and landmark location in a coarse-to-fine manner achieved state-of-the-art accuracy on the task of face detection Zhang et al. [2016]. This approach, called MTCNN, can perform face detection and alignment in unconstrained environments. Additionally, a new class of computer vision models, Segment Anything Models (SAM), can be used to separate specific objects within an image given a general region. SAM models excel at segmenting objects within a complex visual scene, allowing for precise identification of faces for subsequent blurring.

However, the drawback lies in these models' time efficiency. For privacy applications, performance is critical because this anonymization should be done on edge instead of in the cloud, ensuring that peoples' identities are protected before the image data is transmitted online. In this project, we build a real-time face detection and blurring system, integrating state-of-the-art face detection and segmentation models. Most importantly, both models are optimized for performance on standard hardware, a key requirement for real-time privacy-preserving systems. We assess how well both models perform in terms of time efficiency and accuracy in various realistic video settings. In particular, we focus on quantifying the performance of these models in settings with occlusions and low illumination, conditions where previous real-time systems struggle.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

To our knowledge, this is the first application of SAM for real-time face blurring, offering significant accuracy boosts over existing systems in the challenging context of video frames.

## 2 Related Work

### 2.1 Face Detection

In recent years, deep neural networks have been applied ubiquitously for facial detection. The first successful application of neural networks for this task was reported in 1998, when 2-layered neural networks were trained to examine small windows of an image and decide whether each window contains a face Rowley et al. [1998]. However, like many other computer vision tasks, convolutional neural networks (CNNs) quickly rose above other deep learning architectures. Their advantage arises from the fact that all the receptive fields in a layer share weights, resulting in a significantly smaller number of parameters than fully- connected neural networks Minaee et al. [2021]. In 2015, a CNN-based model outperformed all existing baselines on facial detection tasks, even many existing convolutional models Li et al. [2015]. With a cascading structure, the model operates at multiple resolutions, quickly rejecting low-probability regions and carefully selecting between high-probability regions at the last stage. Impressively, the proposed method runs at 14 FPS on a single CPU core for VGA-resolution images.

A similar strategy was adopted to build Multi-task Cascaded Convolutional Networks (MTCNN), a deep convolutional network with three stages Zhang et al. [2016]. As shown in Figure 1, MTCNN contains a *P-net*, *R-net*, and *Q-net*. The *P-net*, or Proposal Network, generates several candidate bounding boxes via regression. These candidates are refined by he *R-net*, or Refinement Network. Finally, the *O-Net*, or Output Network, predicts a final bounding box and facial landmark positions.

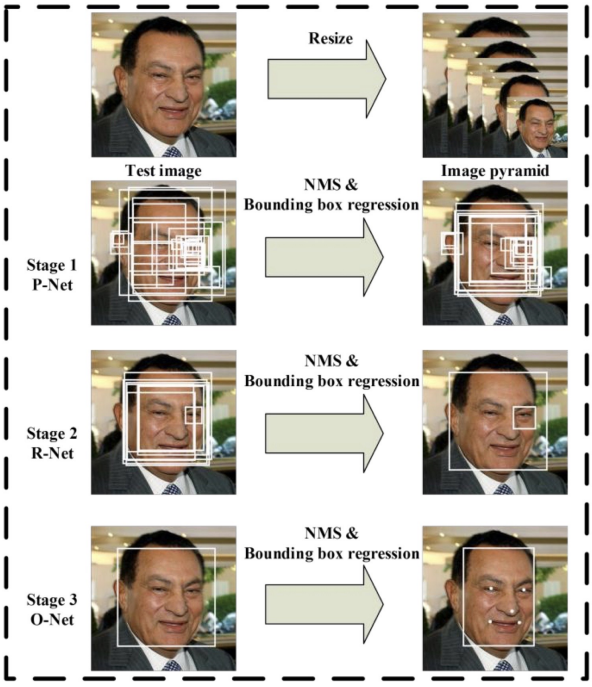


Figure 1: MTCNN contains a *P-net*, *R-net*, and *Q-net*. Figure from Zhang et al. [2016].

Because MTCNN is the state-of-the-art model for many face detection benchmarks, we evaluate this method for robustness to realistic video conditions. Additionally, we prune the network to improve performance of the full EfficientViTGuard system.

### 2.2 Face Segmentation

After detecting the approximate location of the face in a given video frame, the face object must be segmented from the frame for subsequent blurring. Segment Anything Models (SAM) have achieved state-of-the-art accuracy on the task of "cutting out" an object from an image given a click or bounding box prompt. Additionally, SAM has learned a general notion of what objects are, so this understanding enables zero-shot generalization to unfamiliar objects and images without requiring additional training Kirillov et al. [2023]. However, the original SAM model requires GPU compute to run efficiently. Therefore, we benchmark and integrate the Efficient Segment Anything (EfficientVit-SAM) model Cai et al. [2022] into EfficientVitGuard.

EfficientVit-SAM applies multi-scale linear attention to improve performance on mobile CPUs, the ideal hardware for privacy applications. As shown in Figure 2, after getting initial query, key, and value vectors from feed forward networks, multi-scale tokens are generated by performing convolutions. Most importantly, ReLU linear attention is then applied instead of heavy softmax attention, enabling time efficiency.

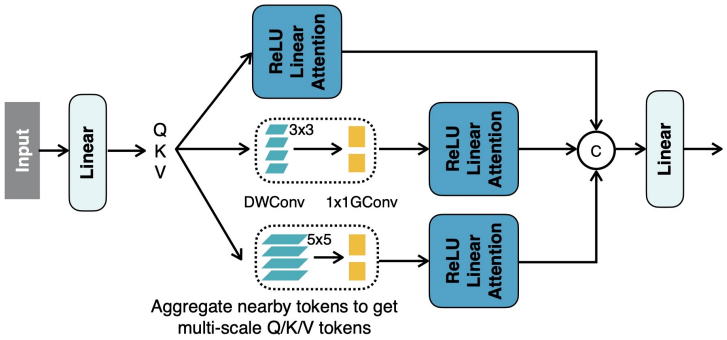


Figure 2: Multi-scale tokens are generated from query, key, and value vectors. Figure from Cai et al. [2022].

### 2.3 Deep Learning for Privacy Systems

Recently, a system named Blur & Track was developed for the purpose of real-time face blurring Jaichuen et al. [2023]. The system performs face detection using RetinaFace, a deep learning model that outperforms MTCNN on front-facing images Deng et al. [2019]. The predicting bounding box area is taken as the region of interest for blurring. While RetinaFace employs lightweight backbone networks and can therefore run real-time on a single CPU core for a VGA-resolution images, Blur & Track naively blurs pixels that do not need to be blurred, lowering the precision of face coverage. While this may not seem like an immediate concern, this leads to loss of background details, which news networks or content creators care about. Therefore, we add a segmentation step between face detection and blurring. Additionally, MTCNN performs more accurately across video conditions, so we experiment with this model.

## 3 Methods

### 3.1 Pruning MTCNN

Since our face segmentation model (EfficientViT-SAM) is already optimized for edge device performance, we focus on optimizing the face detection model, MTCNN. As mentioned, MTCNN uses a cascading series of CNNs to detect and localize faces in digital images or videos. Given that we attempt to optimize performance on non-specialized hardware, we implement channel-based pruning. we remove the channels whose weights are of smaller magnitudes (measured by Frobenius norm). Particularly, we remove $50\%$ of channels from all convolutional and associated activation (PReLu) layers, which reduces the overall size of the model by $20\%$. After pruning the model, we fine-tune on a subset of our dataset.

### 3.2 Experimental Set-Up

As mentioned previously, we benchmark the MTCNN face detection model and EfficientViTSAM segmentation model across various settings. For these experiments, we use the Wider Face benchmark dataset from HuggingFace Yang et al. [2016]. The dataset consists of $32,203$ images, containing $393,703$ labelled faces, with a high degree of variability in lighting scenarios, pose, and occlusion. Face labels are given as bounding boxes, listed as the upper left and bottom right box coordinates. Illumination is given as a binary label, either $0$ for normal illumination or $1$ for abnormal illumination. Occlusions are given as a multiclass label, ranging from $0$ for no occlusion to $2$ for highly occluded. We select $1000$ random images from this dataset for all evaluations, and the remaining images were used to fine-tune MTCNN after pruning.

Therefore, across the given lighting and occlusion scenarios, were $6$ possible image conditions. In this dataset, there were no examples of low illumination images with an occlusion level of $2$. Therefore, our images were categorized into the remaining $5$ classes.

We compare the accuracy of the MTCNN face detection model across these $5$ classes, where accuracy is measured by root mean squared error of the predicted bounding box with respect to the ground truth. We also compare time efficiency of MTCNN, measured as the number of seconds taken to detect a face in a single image on one CPU. Using the same method, we compare time efficiency of EfficientViTSAM, though we do not benchmark accuracy of the segmentation model due to lack of ground truth segmentation masks.

### 3.3 End-to-End System

We design and develop a prototype of a full-stack end-to-end system, EfficientViTGuard. Our prototype uses a React and Node.js frontend, where the user can upload an image of their choice. This image is then sent to the backend server via a Flask API, where the image is run through the MTCNN face detection model to generate a bounding box query. The original image and bounding box query are then fed into EfficientViT-SAM, which returns a segmentation mask for the face. These segmented pixels are replaced with blurred pixels by our backend OpenCV code, and finally, the image with blurred face is returned to the user. An example of this end-to-end system flow is shown in Figure 3.

This system is a prototype for our vision of the final end-to-end EfficientViTGuard, which would instead involve processing full video clips instead of individual images.

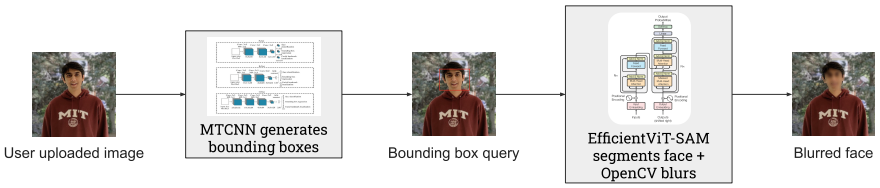


Figure 3: End-to-end example of EfficientViTGuard.

## 4 Results

### 4.1 Accuracy of facial detection is affected by occlusion

As shown in Figure 4, the error of bounding box predictions is significantly greater at high occlusion levels. The vertical axis shows the RMSE between predicted and ground-truth face bounding box, and the horizontal axis is split by illumination level. At each illumination level, we observe the distribution of RMSE values across our evaluation set, colored by occlusion level. With normal illumination (0), we observe that the interquartile ranges of images at $0$ and $1$ occlusion levels overlap. However, the majority ($> 75\%$) of images at an occlusion level of $2$ have a larger error than the median errors of images with lower occlusion levels. Therefore, we conclude that accuracy of facial detection is affected by occlusion.

For images with uneven illumination (1), we observe that the majority ($> 75\%$) of images at an occlusion level of 1 have a larger error than the median error of images with no occlusion. However, given that the spread of this distribution is very narrow, we cannot make a concrete conclusion about the joint effects of uneven illumination and occlusion.

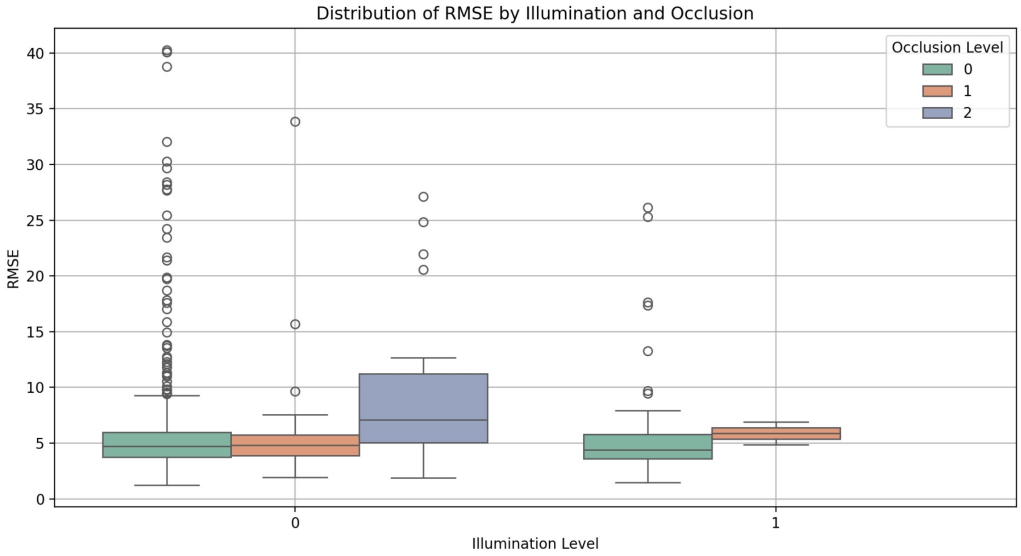


Figure 4: Impact of occlusion levels on bounding box prediction error. Higher occlusion levels exhibit increased errors, especially under normal illumination.

### 4.2 Time efficiency of detection and segmentation is generally consistent

As we can see in Figure 5, the occlusion and illumination levels of input images have a generally marginal effect on the overall time-efficiency of EfficientViTGuard, as measure in seconds per inference. The one exception to this rule that we observed was in the case where an image is both unevenly illuminated (category 1) and slightly occluded (category 1). In these cases, the bounding box prediction algorithm's inference increases significantly, which is likely a result of high input noise due to a hit in data quality. However, we find that EfficientViTGuard predicts bounding boxes quite quickly in *all* cases where the image is illuminated, regardless of the image's occlusion level.

Figure 6 shows that occlusion and illumination levels have a negligible effect on EfficientViT-SAM's inference-time efficiency when used to segment face pixels from the image. Although these image properties don't seem to effect time efficiency, other properties, such as the size of a bounding box or the semantic content of the image, might have an effect. While outside of the scope of this work, these factors may be analyzed in a more comprehensive study.

### 4.3 Full-stack system is an effective proof-of-concept

Our full-stack EfficientVitGuard system serves as a proof-of-concept that a real-time face blurring system can be built by combining EfficientViTSAM for face segmentation with a pruned version of MTCNN for face detection. A demo of our current prototype is available at `https://www.youtube.com/watch?v=VTCGm4N-_k8`. While our system only currently supports blurring for a single input image, it is easily extensible. By optimizing optical flow models like Recurrent All-Field Transform (RAFT)Teed and Deng [2020] for CPU hardware, EfficientViTGuard can take track the initial face segmentation across an entire video, thus protecting the face's identity across the entire video.

## 5 Conclusions

In this paper, we present *EfficientViTGuard*, a real-time and on-device face detection and blurring system that is designed to enhance privacy in video content. As such, our work integrates a pruned

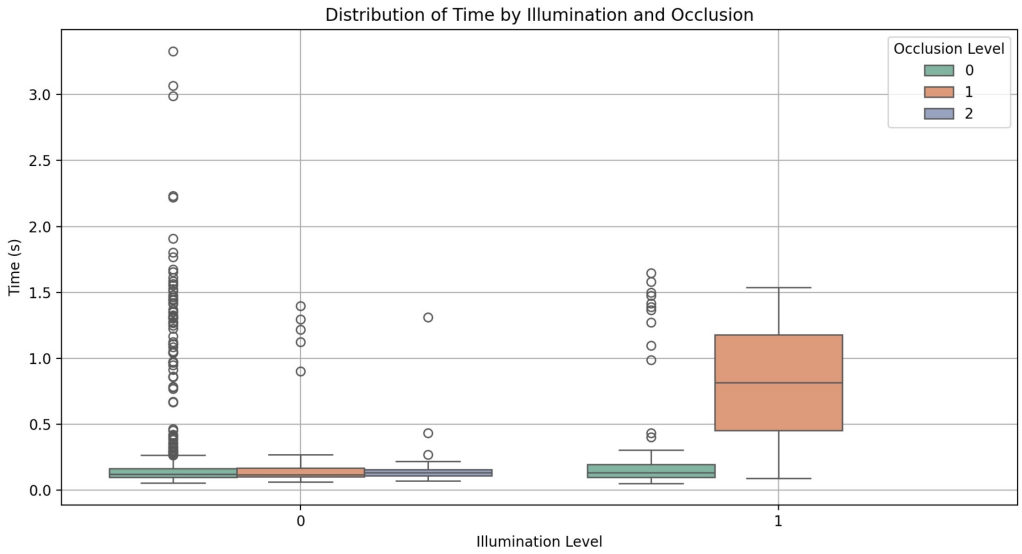


Figure 5: Impact of illumination and occlusion on bounding box prediction time efficiency. Under high occlusion and low illumination, time efficiency degrades significantly.

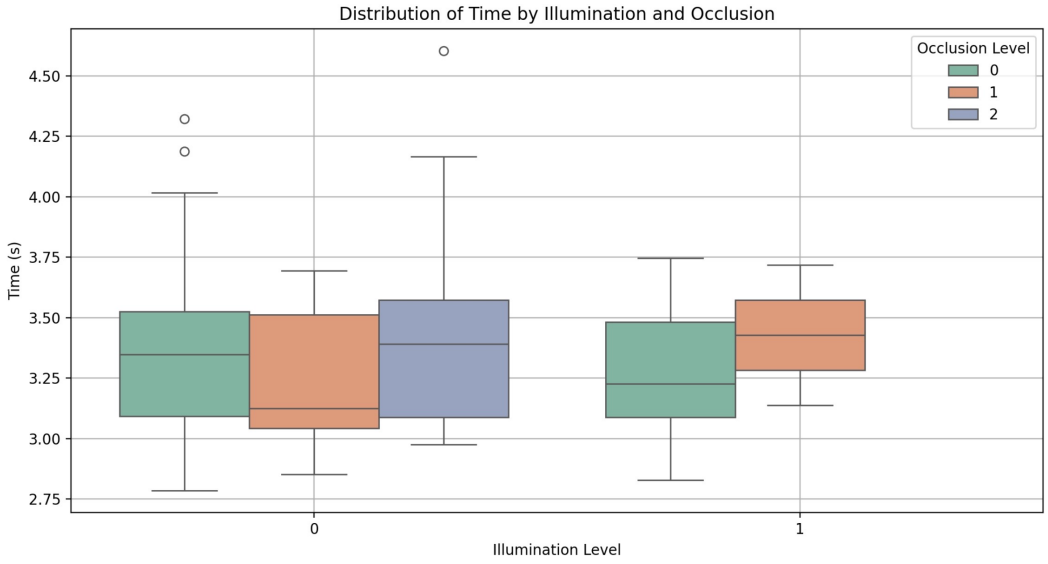


Figure 6: Impact of illumination and occlusion on time efficiency of face segmentation. There is little to no effect on time efficiency of EfficientViT-SAM under these varying conditions.

and finetuned **MTCNN** face detection model with **EfficientViT-SAM** to efficiently and accurately replace faces in any image with a blurred representation.

While *EfficientViTGuard* is an effective proof-of-concept, our full-stack system also paves the way for future extensions to handling continuous video streams. Some future work that would help enable this include optimizing optical flow models, such as RAFTTeed and Deng [2020], to track blurred pixels between frames, and deploying our system on real camera hardware. All in all, *EfficientViTSAM* leverages recent advancements in TinyML to address our world's pressing need for more privacy-preserving solutions.

## Acknowledgements

The authors would like to thank Professor Song Han for his excellent teaching of many of the concepts used in this work through the TinyML course, along with his feedback on the project presentation. The authors would also like to thank the TinyML course TAs Han Cai and Ji Lin for their valuable feedback on the project propsal and presentation.

## References

H. Cai, C. Gan, and S. Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 2022.

J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019.

T. Jaichuen, N. Ren, P. Wongapinya, and S. Fugkeaw. Blur & track: Real-time face detection with immediate blurring and efficient tracking. In *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 167–172. IEEE, 2023.

A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5325–5334, 2015.

S. Minaee, P. Luo, Z. Lin, and K. Bowyer. Going deeper into face detection: A survey. *arXiv preprint arXiv:2103.14983*, 2021.

H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.

Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020.

S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.