SecFePAS: Secure Facial-expression-based Pain Assessment with Deep Learning at the Edge

Kanwal Batool
University of Amsterdam
The Netherlands

Saleem Anwar
Rotterdam University of Applied Sciences
The Netherlands

Zoltán Ádám Mann
University of Halle-Wittenberg
Germany

***Abstract*—Patient monitoring in hospitals, nursing centers, and home care can be largely automated using cameras and machine-learning-based video analytics, thus considerably increasing the efficiency of patient care. In particular, Facial-expression-based Pain Assessment Systems (FePAS) can automatically detect pain and notify medical personnel. However, current FePAS solutions using cloud-based video analytics offer very limited security and privacy protection. This is problematic, as video feeds of patients constitute highly sensitive information.**

To address this problem, we introduce SecFePAS, the first FePAS solution with strong security and privacy guarantees. SecFePAS uses advanced cryptographic protocols to perform neural network inference in a privacy-preserving way. To counteract the significant overhead of the used cryptographic protocols, SecFePAS uses multiple optimizations. First, instead of a cloud-based setup, we use edge computing with a 5G connection to benefit from lower network latency. Second, we use a combination of transfer learning and quantization to devise neural networks with high accuracy and optimized inference time. Third, SecFePAS quickly filters out unessential frames of the video to focus the in-depth analysis on key frames. We tested SecFePAS with the SqueezeNet and ResNet50 neural networks on a real pain estimation benchmark. SecFePAS outperforms state-of-the-art FePAS systems in accuracy and optimizes secure processing time.

I. Introduction

Facial-expression-based pain assessment (FePAS) has versatile applications in healthcare, such as real-time monitoring of patients in their homes or in nursing centers [1], [2]. FePAS offers significant benefits for patient care. It facilitates accurately gauging pain levels, particularly in patients who struggle with verbal communication, such as infants [3], the elderly [4], and ICU (Intensive Care Units) patients [5]. Upon pain detection, nursing personnel can be alerted to provide appropriate treatment. Recent advances in smart cameras, edge networks, and computer vision based on deep learning have made it possible to perform FePAS automatically and with good accuracy [5].

However, video streams of patient faces are highly sensitive data, requiring strong protection for privacy reasons. Only authorized healthcare providers should be able to access patient data to prevent privacy breaches and misuse. Thus, robust privacy-preserving techniques are essential in FePAS. Using cloud-based video analytics services in the context of FePAS constitutes a privacy risk, as the FePAS video streams must remain inaccessible to cloud providers and other cloud tenants [6].

The standard approach to guarantee privacy in video processing is to obscure personally identifiable regions – such as faces – in the video frames, for example by blurring them or by covering them with an opaque overlay [7], [8]. Such approaches, however, are inappropriate for FePAS, since they make it impossible to analyze facial expressions.

To address privacy concerns while allowing effective analytics, recent research [9]–[17] introduced cryptographic frameworks for secure neural network inference (SNNI). These approaches address the problem where a server possesses a valuable pre-trained neural network model F. The server offers inference with F as a service, without disclosing F itself. A client can use the service to perform predictions on her private data x while keeping x secret from the server. The client can learn the inference result $F(x)$ without gaining any additional information beyond what can be inferred from $F(x)$, while the server learns nothing about x or $F(x)$. SNNI protocols solve this problem, satisfying the needs of both parties. One application could be privacy-preserving FePAS, where the server evaluates pain levels from facial images without learning the image contents. A challenge in using SNNI protocols is their significant computation and communication overhead [18], [19]. Thus, a major goal of recent research in SNNI has been to reduce this overhead, while upholding high accuracy [20].

Benefiting from this trend, this paper presents SecFePAS[1], the first SNNI-based FePAS solution. Building on SNNI for analyzing selected frames of the video feed has the major advantage of offering strong privacy guarantees, which is essential in FePAS. Specifically, we use Cheetah [17], a state-of-the-art SNNI approach that works well with Convolutional Neural Networks (CNNs), the type of neural network often used for image classification tasks.

Beside the used SNNI approach, also the neural network used in the analytics has large impact on performance, in terms of both speed and accuracy. In this research, we experiment with different CNN architectures, resulting in

[1]The source code of SecFePAS is publicly available at https://github.com/KanwalBat00l/SecFePAS.

different trade-offs between inference speed and accuracy. For efficiently training the neural network, SecFePAS uses a transfer learning approach: the neural network is first pre-trained on a general image dataset, followed by fine-tuning on a dataset specific to FePAS. SecFePAS also uses quantization to reduce the bitlength of model parameters. This leads to a reduction in model size and to a speed improvement during the SNNI process, both of which are very advantageous, especially in a resource-constrained edge setup. Through careful combination of transfer learning and quantization, SecFePAS achieves a significant speedup while only incurring a negligible accuracy drop.

SecFePAS is based on an edge computing architecture, carefully partitioning functionality between an end device (e.g., a smart camera) and an edge server (e.g., a nearby server of the nursing center). The selection of representative frames from the video feed and their preprocessing (cropping, scaling) are performed by the end device. The edge server stores the CNN model. End device and edge server perform secure inference using an SNNI protocol. Thus, sensitive patient data never leaves the end device in the patient's room. Since the edge server is just one network hop away from the end device, network latency is much lower than for existing cloud-based video analytics solutions. Using a 5G network connection between end device and edge server results in a high bandwidth, which helps further decrease communication time.

The main contribution of this paper is the design and prototypical implementation of SecFePAS. SecFePAS is the first approach to provide FePAS with strong privacy guarantees and high accuracy, by combining carefully trained and quantized CNNs, advanced cryptographic protocols for SNNI, and edge computing with 5G networks. SecFePAS stands out from existing edge-based video analytics by implementing privacy-preserving neural network inference for pain assessment. It addresses unique challenges in model security and data protection throughout the inference process. We evaluated the SecFePAS prototype on the UNBC-McMaster Shoulder Pain Expression archive [21] using the SqueezeNet [22] and ResNet50 [23] CNNs. The results show that SecFePAS outperforms the state-of-the-art in terms of accuracy and speed.

II. Related Work

Secure image or video classification utilizes advanced cryptographic techniques such as Homomorphic Encryption (HE) [24], Secure Multi-Party Computation (MPC), and Trusted Execution Environments (TEE) to protect privacy. Solutions such as Visor [25] leverage cloud infrastructure and TEEs to protect sensitive information during video analytics. Visor executes video pipelines within a hybrid TEE that extends across both the CPU and GPU on the trusted client and server. PPVC [26] employs the ConvNet model to incorporate MPC protocols for secure single-frame video classification in the cloud. The model was first trained on the FER 2013 dataset, consisting of 30,000 images of 48x48 pixels. Then, the model was fine-tuned on the RAVDESS dataset, consisting of video clips of duration 3 to 5 s at 30 frames per second (fps), sampling every 15th frame, leading to 2 fps video. Only spatial information was taken into account. Innovations like Crypto3D [27] and CryptoMask [28] enhance privacy through 3D feature extraction and face recognition, respectively. However, these methods face challenges related to scalability, computational overhead, and accuracy. These issues are particularly critical in sensitive applications, such as facial-expression-based pain assessment, which demands efficient processing, high accuracy, and strict data confidentiality.

To our knowledge, SecFePAS is the first approach to achieve secure automated facial-expression-based pain assessment. We built on existing research in cryptographic methods for secure image classification and enhanced it with new optimisation techniques tailored to secure video frame analysis at the edge.

III. Proposed Solution: SecFePAS

SecFePAS integrates the functionalities of FePAS and SNNI to provide a secure system for pain assessment using facial expressions in video streams. It leverages CNNs for facial analysis in video frames, edge computing for efficient processing close to the camera source, and 5G networks for fast and reliable data transmission. SecFePAS incorporates an SNNI solution as a black box, which uses advanced cryptographic techniques, such as Homomorphic Encryption (HE) and Secure Multi-Party Computation (MPC) to keep all input data encrypted and inaccessible throughout the entire process. As output, it generates encrypted pain scores that can only be decrypted by authorized devices. This configuration supports seamless integration into healthcare systems and makes it ideal for real-world FePAS applications.

Fig. 1 gives an overview of the SecFePAS workflow. This workflow consists of two main phases: a preparatory offline phase (the upper two thirds of the figure) and an online phase (the lower third of the figure). In the following, we first describe the overall setup assumed by SecFePAS, then the online phase, and finally the preparatory offline phase.

A. Setup

Stakeholders. SecFePAS involves three key stakeholders, and an optional fourth one:

- *Patient*: natural person, about whom a video feed is captured.
- *Healthcare service provider* (e.g., nursing station): organization providing healthcare services to patients, which include monitoring patients' condition.
- *Machine learning model provider*: company specializing in training neural networks, furnishing the healthcare service provider with a trained model for facial-expression-based pain detection.

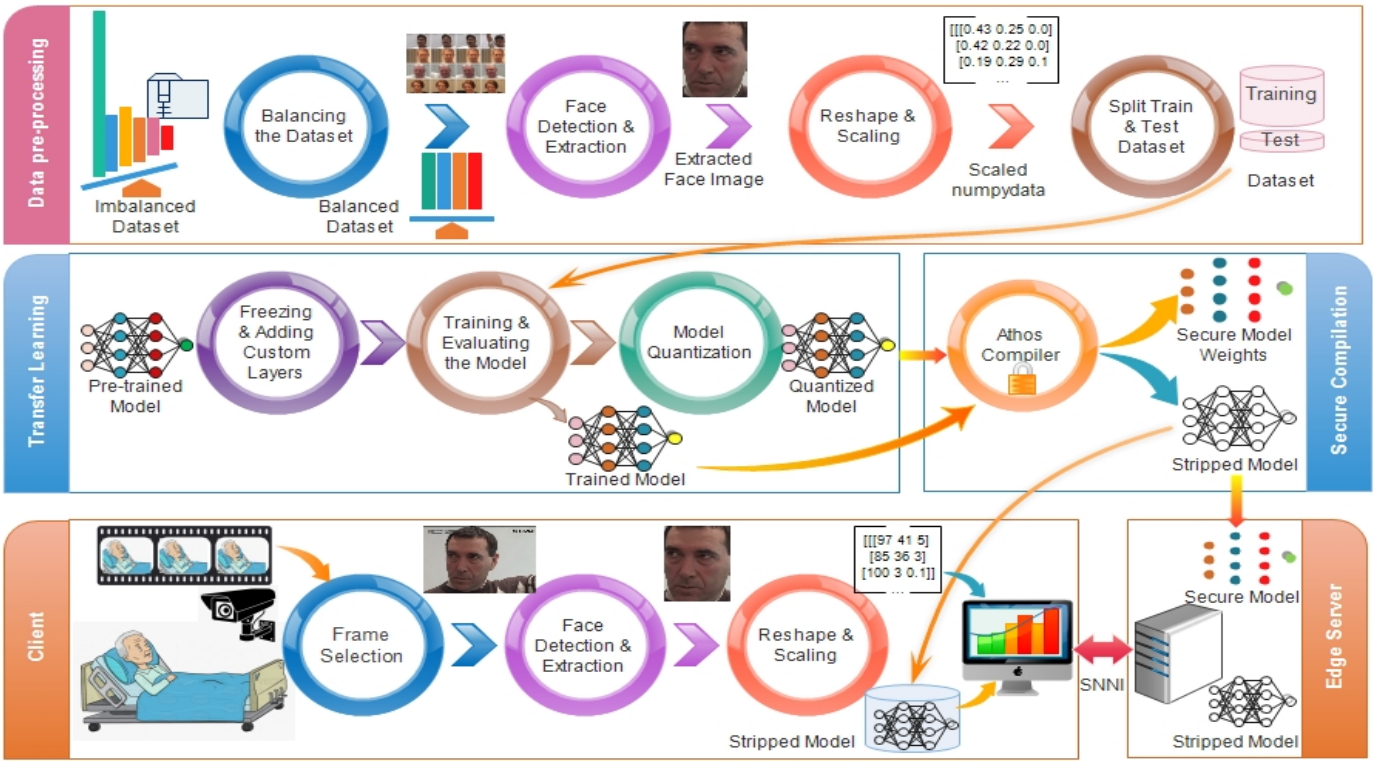


Fig. 1: Workflow of a SecFePAS system

- Optionally, the edge server may be provided by a dedicated *edge infrastructure provider* (e.g., a telecommunication operator). Alternatively, this role may be filled by the healthcare service provider or the machine learning model provider.

Confidentiality requirements. The healthcare service provider needs to have access to the patient's sensitive medical data to be able to provide the healthcare service. However, for the machine learning model provider and – if it is a separate entity – for the edge infrastructure provider, there is no need to have access to sensitive patient data, and thus, such access should be prohibited.

The neural network model specially tuned for facial-expression-based pain detection is the intellectual property of the machine learning model provider. The machine learning model provider offers inference with this neural network as a service, but wants to keep the weights and other parameters of the model secret from other parties.

Infrastructure. The neural network model for facial-expression-based pain detection is trained on the backend infrastructure of the machine learning model provider. After training, the model is deployed to an *edge server*. A *smart camera*, owned by the healthcare service provider, is stationed in the patient's room and captures a video stream of the patient. The smart camera has computational resources, allowing it to perform some limited processing locally. Processing that requires more resources can be performed on the edge server [29].

B. Online phase

The smart camera captures a real-time video stream of the patient. To be able to apply the neural network model for facial-expression-based pain assessment, some pre-processing activities are needed first.

Pre-processing. All pre-processing takes place in the smart camera. First, the video is decoded into individual frames. In our proof-of-concept implementation, we use OpenCV [30] (Open Computer Vision Library) for this.

In principle, pain detection could be performed on every frame. However, this is not feasible, given the large amount of computation and communication necessary for privacy-preserving analytics on images of realistic size. Also, this is not necessary, since subsequent frames typically only differ marginally from each other. Therefore, the most important pre-processing step is *frame selection*, i.e., selecting the frames for in-depth analysis. Our aim is to select the minimum number of frames necessary for ensuring that no pain signals are missed.

To achieve this, we perform frame selection in two steps. Facial expressions associated with significant pain sensations typically last between 5 to 15 seconds [31]. To ensure that no pain expression is missed, frames are selected periodically every 5 seconds in the first step. In many cases, there is no significant change even between frames that are 5 seconds apart. Thus, in the second step, we apply Temporal Redundancy Reduction [32] to discard redundant frames from the ones selected in the first step.

TABLE I: Specifications of the used CNN models vs Baseline (∗ marks the baseline model). Layer types: Conv = convolutional, BN = batch normalization, FC = fully connected, ReLU = rectified linear unit, Trunc = truncation

Benchmark	Specifications				Layers				
	Input size	Parameter memory	Feature (Activation) memory	FLOPS	Conv	BN	FC	ReLU	Trunc
*ConvNet	48 x 48	5.92 MB	879.31 MB	5.38 GFLOPS	5	0	3	7	0
SqueezeNet	224 x 224	5 MB	30 MB	837 MFLOPS	26	0	0	26	26
ResNet50	224 x 224	98 MB	103 MB	4 GFLOPS	53	49	1	49	98

This way, we select only frames in which there is a change, justifying the computational cost of in-depth analysis.

From the selected frames, SecFePAS extracts the part that contains the patient's face. For this purpose, MTCNN (Multi-Task Cascaded Convolutional Neural Network [33]) is applied, which can quickly and accurately detect faces. Each processed frame is adjusted, cropped to the facial part, resized to a standard size (224 x 224 in our implementation), reshaped to a 4D-array, and normalized by dividing each pixel value by 255.

All pre-processing steps are performed locally in the smart camera. No advanced domain-specific analytics is needed for these steps. No sensitive data is sent, and thus, no SNNI is needed at this stage. The pre-processing steps are thus much cheaper than the subsequent SNNI stage.

SNNI. The pre-processed video frame is used as input to the specialized neural network model, located on the edge server, for facial-expression-based pain assessment. To protect the confidentiality of video frames from the machine learning model provider and potentially from the edge infrastructure provider, as well as to protect the confidentiality of the neural network model from the healthcare service provider, the inference is done using SNNI. That is, the smart camera and the edge server engage in an SNNI protocol to do the inference securely.

As a result, the SecFePAS application running on the smart camera learns the inference output, which indicates the pain level. A high pain level triggers an alarm for the experts of the healthcare service provider, so that they can provide timely medical assistance to the patient.

According to the SNNI security guarantees, the healthcare service provider learns nothing about the weights and other parameters of the neural network (beyond what the output reveals). Sensitive patient data remains secret from the process running on the edge server. Neither the machine learning model provider nor the edge infrastructure provider learns anything about the patient data.

TABLE II: Balanced dataset, created from the UNBC-McMaster Shoulder Pain expression archive [21]

Class	Label	PSPI Code	Frame Samples
1	No Pain	0	3092
2	Little Pain	1	2909
3	Moderate Pain	2	2351
4	Extreme Pain	3+	3109

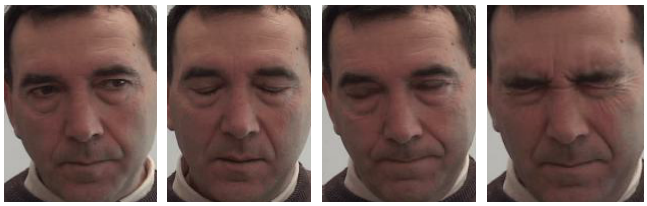
C. Preparatory offline phase

The aim of this phase is to prepare a neural network model for facial-expression-based pain assessment in images, such that secure inference with the model is possible with high accuracy and optimized execution time. The activities of this phase are carried out by the machine learning model provider, prior to the online phase.

Dataset preparation. Researchers from the University of Northern British Columbia and McMaster University collected a dataset of 200 video sequences, comprising 48,398 color frames with a resolution of 320 × 240 pixels, of 25 adults. The majority class, "no pain," comprises 82.71% of the dataset (40,029 frames). Images showing different pain levels account for only 17.29% (8,369 frames). Thus, the dataset is highly imbalanced. To address the potential classification bias, we created a subset by randomly selecting frames and categorizing them into different pain levels using the Prkachin and Solomon Pain Intensity (PSPI) scale [31], as shown in Table II. Fig. 2 shows facial expressions of a patient experiencing different levels of pain. Images are cropped, reshaped, and scaled in the same way as above in the online phase. For training, 80% of the set is used; the remaining 20% is reserved for validation.

Creating the model. The neural network architecture has to be chosen carefully, so that it allows accurate and fast inference. For our proof-of-concept implementation, we experimented with two CNNs widely used in image classification: ResNet50 and SqueezeNet. In addition, we used the ConvNet model as a baseline, as it has been used for secure video analytics previously [26]. Table I details the architecture of the used models.

These models were first pre-trained on a general image dateset, namely the ImageNet-1k dataset, comprising images with a resolution of 224x224. We then fine-tuned the models using the FePAS-specific dataset described



No Pain Little Pain Moderate Pain Extreme Pain

Fig. 2: Sample images from the UNBC-McMaster Shoulder Pain Expression archive [21], ©Jeffrey Cohn.

TABLE III: Comparison of SecFePAS variants with the baseline (marked with $*$). The best result per column is in boldface. End-to-end time is the time of the online phase, encompassing client-side preprocessing and SNNI.

Benchmark	Memory Use	Evaluation Accuracy	Framework	End2End Time		Communication
				Client	SNNI	
*ConvNet [26] (RAVDESS)	885.23 MB	56%	MP-SPDZ [34]	-	511.64s GPU Runtime	669.35GB
ResNet-50 (PAIN)	96MB	**84.78%**	SCI_{HE}	1.09s	297.5s	30.94GB
			Cheetah	1.09s	78.6s	2.25GB
ResNet-50-FP16 (PAIN)	44.8MB	84.74% Loss=0.04%	SCI_{HE}	1.09s	238s	24.75GB
			Cheetah	1.09s	63.6s	1.8GB
SqueezeNet (PAIN)	5MB	55.6%	SCI_{HE}	1.09s	55.3s	9.01GB
			Cheetah	1.09s	22.12s	0.67GB
SqueezeNet-FP16 (PAIN)	**3.3MB**	55.55% Loss=0.05%	SCI_{HE}	1.09s	40.5s	6.6GB
			Cheetah	1.09s	**19.21s**	**0.58GB**

above. The process of pre-training the model using a general dataset and then fine-tuning it using a task-specific dataset, also known as transfer learning, has multiple advantages. Not only does it reduce the time needed for the task-specific training, but it also alleviates the need for a large amount of task-specific training data. With transfer learning, we can achieve high accuracy more quickly and using only a relatively small task-specific dataset.

The next step in SecFePAS is model quantization. By reducing the bitlength of the weights in the model, we reduce the memory requirements as well as the execution time of the SNNI process; both of these are important goals, especially in an edge computing setup. Specifically, we convert the weights from 32-bit floating-point (FP32) to FP16. This conversion can decrease the model size by up to 50%, since each weight takes up half the space compared to float32. As we will see later, the accuracy loss stemming from this quantization step is negligible.

Preparing for SNNI. SecFePAS uses Cheetah, a state-of-the-art SNNI approach [17] for performing secure inference. Like several other SNNI solutions [15], [35], Cheetah follows a compilation-based approach. That is, the TensorFlow code defining the neural network model is compiled, using the Athos[2] compiler [36], into a distributed C++ program that incorporates the cryptographic protocols for performing SNNI with the given model. The "stripped model" created by Athos specifies the model architecture, but not the weights. This stripped model is shared with the client (the smart camera). The weights of the model are kept secret on the server side.

IV. Experiments

This section presents the results of preliminary experiments with the SecFePAS prototype.

Experimental setup. We experimented with two SNNI frameworks: Cheetah [17] and SCI_{HE} [15], which in turn use the SEAL [37] library, enhanced with HEXL [38] acceleration, and the EMP [39] toolkit. We optimized the implementation by converting to FP16 and fine-tuned parameter settings to support FP16 conversion. We used a precision of $f = 12$ for fixed-point values. SecFePAS is implemented in C++ and compiled with gcc version 11.4 on Ubuntu 22.04.4 LTS. The programs used 4 threads.

All experiments were conducted on a WebGPU-Space equipped with an Intel Xeon Silver 4314 CPU @ 2.40GHz (32 cores) and 250GB of RAM. To evaluate SecFePAS in a resource-constrained environment, the server side of the SNNI process was run on a virtual machine with restricted memory. The client side was run on the host machine, which also had limited memory. Memory limitations for both setups were imposed using the setrlimit tool[3].

Results. We evaluated the models using frames with dimensions of 224x224x3, which are significantly larger than the 48x48x3 frames used in previous studies with the baseline model. Table III shows the empirical results. It compares the baseline approach (using the ConvNet model trained on the RAVDESS dataset)[4] with four variants of SecFePAS, trained on the PAIN dataset. These four variants differ in the used neural network model (ResNet50 vs. SqueezeNet) and in whether the model was applied unchanged or it was quantized to FP16. Each model is tested with both the SCI_{HE} and Cheetah SNNI frameworks.

As can be seen from the table, SqueezeNet achieves similar accuracy as the baseline model, while ResNet50 achieves much better accuracy. In terms of all resource consumption metrics (memory use, end-to-end time, communication), we can observe the same pattern: ResNet50 leads to a significant reduction compared to the baseline, while SqueezeNet leads to even further reduction. In addition, Cheetah is much more efficient than SCI_{HE}. Quantization to FP16 results in significant savings in resource consumption in all cases and for all relevant metrics, while its impact on accuracy is negligible.

[2]https://github.com/mpc-msri/EzPC/tree/master/Athos

[3]https://linux.die.net/man/2/setrlimit

[4]To our knowledge, no prior research addressed privacy-preserving facial pain assessment. Therefore, we used secure facial emotion detection [26] from videos employing MPC as our baseline model for comparison purposes, using the numbers reported in that paper.

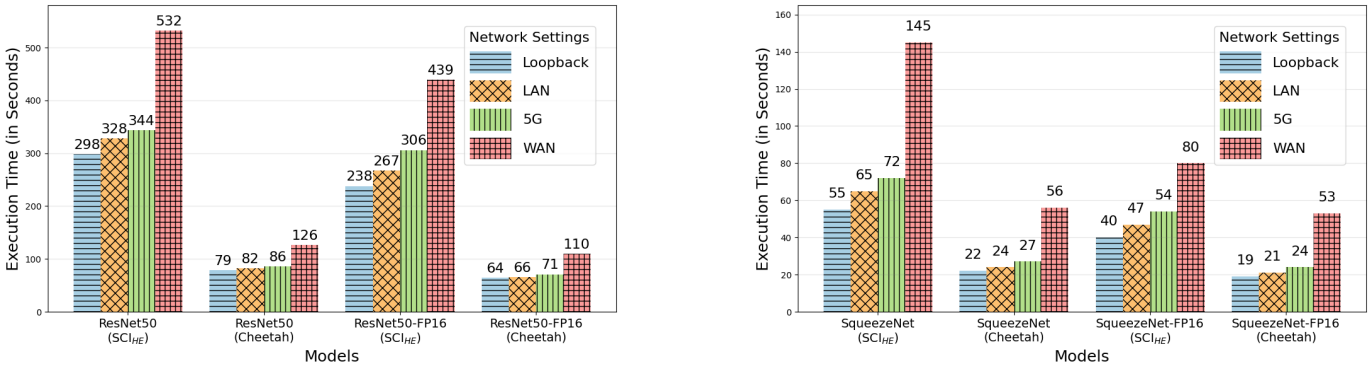


Fig. 3: Impact of different network technologies on SNNI execution time. Left: ResNet50, right: SqueezeNet

Thus, we can conclude that SecFePAS significantly outperforms the baseline. Using the SqueezeNet model and the Cheetah SNNI framework, SecFePAS achieves roughly the same accuracy as the baseline, but about two orders of magnitude more efficiently. Using the ResNet50 model and the Cheetah SNNI framework, SecFePAS achives much better accuracy than the baseline, about one order of magnitude more efficiently.

In a second experiment, we investigated the impact of the network connection between client and server. The network connection does not influence accuracy or memory use, but it does influence SNNI time. Fig. 3 compares different network technologies: Loopback (7 GBps bandwidth, 0.048 ms round-trip time (RTT)), LAN (1 GBps bandwidth, 1 ms RTT), 5G (1 Gbps bandwidth, 5 ms RTT), and WAN (400 MBps bandwidth, 40 ms RTT). As the figure shows, using 5G networks offers a significant performance boost over WAN in all cases. This demonstrates the viability of an edge-based approach using 5G compared to a cloud-based approach using a WAN connection.

Overall, SecFePAS achieves high accuracy and is much faster than the state-of-the-art. However, further reducing the run time should be a key focus for future work, to better satisfy the performance needs of FePAS applications.

V. Discussion

The empirical results of the previous section demonstrate the potential of SecFePAS in achieving accurate, efficient, and secure facial-expression-based pain assessment. The results of SecFePAS are promising and already significantly outperform the state of the art.

However, two main challenges remain. First, while the accuracy of almost 85% achieved with ResNet50 is quite high, it might not be high enough for a real deployment. Also, factors like individual differences in pain expression, head poses, illumination conditions and occlusions etc. complicate accurate effective pain assessment [40], [41], thus potentially lowering accuracy in real scenarios. We expect that better sensor technology (e.g., higher resolution and luminous sensitivity), more advanced image preprocessing (e.g., fusion from multiple cameras) and more capable machine learning models (e.g., vision transformers) will help improve accuracy in the future. Future research should also investigate how various methods of selecting the validation subset affect model accuracy.

The second challenge is speed. Unlike traditional FePAS systems, SecFePAS guarantees the protection of input and output data, and of the model. However, the cryptographic techniques that are needed for this introduce significant computational overhead and require extensive communication. As we have seen, combining a powerful SNNI framework, quantization, and 5G technology increases speed significantly, but again, this may not be enough in practical applications. Our future work will mainly concentrate on further improving speed. For this purpose, we will investigate the use of more advanced cryptographic protocols (e.g., function secret sharing [42]) as well as the use of model compression techniques beyond quantization (e.g., neural architecture search, knowledge distillation [43]). Also parallelization, GPU usage, and hardware acceleration are potential topics for future research.

VI. Conclusion

This research introduced SecFePAS, the first solution for secure facial-expression-based pain assessment. We integrated an optimized SNNI framework with advanced cryptographic techniques to ensure the protection of sensitive video data. Our evaluation confirmed that SecFePAS operates efficiently as an edge-only solution while providing high accuracy. However, the used cryptographic methods can be computationally demanding and require significant communication. Further work is needed, aiming at improving accuracy together with run time reduction for SecFePAS.

Acknowledgment. This work received funding from the European Union's Horizon Europe research and innovation programme under Grant No. 101168311 (LICORICE). This work includes sample images provided by UNBC-McMaster Shoulder Pain Expression archive benchmark [21], ©Jeffrey Cohn; permission to use these images was granted in the agreement.

References

[1] S. Kaltwang, O. Rudovic, and M. Pantic, "Continuous pain intensity estimation from facial expressions," in *International Symposium on Visual Computing*. Springer, 2012, pp. 368–377.

[2] G. D. De Sario, C. R. Haider, K. C. Maita, R. A. Torres-Guzman, O. S. Emam, F. R. Avila, J. P. Garcia, S. Borna, C. J. McLeod, C. J. Bruce *et al.*, "Using AI to detect pain through facial expressions: A review," *Bioengineering*, vol. 10, no. 5, p. 548, 2023.

[3] M. S. Salekin, G. Zamzmi, D. Goldgof, R. Kasturi, T. Ho, and Y. Sun, "Multimodal spatio-temporal deep learning approach for neonatal postoperative pain assessment," *Computers in biology and medicine*, vol. 129, p. 104150, 2021.

[4] M. Kim, X. Jiang, K. Lauter, E. Ismayilzada, and S. Shams, "Secure human action recognition by encrypted neural network inference," *Nature communications*, vol. 13, no. 1, p. 4799, 2022.

[5] E. Othman, P. Werner, F. Saxen, A. Al-Hamadi, S. Gruss, and S. Walter, "Automatic vs. human recognition of pain intensity from facial expression on the X-ITE pain database," *Sensors*, vol. 21, no. 9, 2021.

[6] A. Palm, Z. Á. Mann, and A. Metzger, "Modeling data protection vulnerabilities of cloud systems using risk patterns," in *Proceedings of the 10th System Analysis and Modeling Conference (SAM)*. Springer, 2018, pp. 1–19.

[7] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "Enabling live video analytics with a scalable and privacy-aware framework," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 3s, Art. 64, 2018.

[8] C. Lachner, Z. Á. Mann, and S. Dustdar, "Towards understanding the adaptation space of AI-assisted data protection for video analytics at the edge," in *IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2021, pp. 7–12.

[9] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*. PMLR, 2016, pp. 201–210.

[10] D. Demmler, T. Schneider, and M. Zohner, "ABY-A Framework for Efficient Mixed-Protocol Secure Two-Party Computation," in *NDSS*, 2015.

[11] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 619–631.

[12] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1651–1669.

[13] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar, "XONN: XNOR-based oblivious deep neural network inference," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1501–1518.

[14] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, "Delphi: A cryptographic inference system for neural networks," in *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, 2020, pp. 27–30.

[15] D. Rathee, M. Rathee, N. Kumar, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, "CrypTFlow2: Practical 2-Party secure inference," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 325–342.

[16] A. Patra, T. Schneider, A. Suresh, and H. Yalame, "ABY2.0: Improved Mixed-Protocols Secure Two-Party Computation," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2165–2182.

[17] Z. Huang, W.-j. Lu, C. Hong, and J. Ding, "Cheetah: Lean and fast secure two-party deep neural network inference," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 809–826.

[18] R. de Vries and Z. Á. Mann, "Secure neural network inference as a service with resource-constrained clients," in *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, 2023, pp. 1–10.

[19] J. Prins and Z. Á. Mann, "Secure neural network inference for edge intelligence: Implications of bandwidth and energy constraints," in *IoT Edge Intelligence*. Springer, 2024, pp. 265–288.

[20] Z. Á. Mann, C. Weinert, D. Chabal, and J. W. Bos, "Towards practical secure neural network inference: the journey so far and the road ahead," *ACM Computing Surveys*, vol. 56, no. 5, 2023.

[21] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, "Painful data: The UNBC-McMaster shoulder pain expression archive database," in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE, 2011, pp. 57–64.

[22] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size," arXiv preprint, arXiv:1602.07360, 2016.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[24] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.

[25] R. Poddar, G. Ananthanarayanan, S. Setty, S. Volos, and R. A. Popa, "Visor: Privacy-preserving video analytics as a cloud service," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1039–1056.

[26] S. Pentyala, R. Dowsley, and M. De Cock, "Privacy-preserving video classification with convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8487–8499.

[27] B. Liu, R. Wang, Z. Ba, S. Zhou, C. Ding, and Y. Hong, "Poster: Cryptographic inferences for video deep neural networks," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 3395–3397.

[28] J. Bai, X. Zhang, X. Song, H. Shao, Q. Wang, S. Cui, and G. Russello, "CryptoMask: Privacy-preserving face recognition," in *International Conference on Information and Communications Security*, 2023, pp. 333–350.

[29] Z. Á. Mann, A. Metzger, J. Prade, and R. Seidl, "Optimized application deployment in the fog," in *Proceedings of the 17th International Conference on Service-Oriented Computing (ICSOC)*. Springer, 2019, pp. 283–298.

[30] G. Bradski, "The OpenCV library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[31] K. M. Prkachin and P. E. Solomon, "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain," *Pain*, vol. 139, no. 2, pp. 267–274, 2008.

[32] B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 533–544, 1995.

[33] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using Multi-task Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[34] M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1575–1590.

[35] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, "CrypTFlow: Secure TensorFlow inference," in *IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 336–353.

[36] N. Chandran, D. Gupta, A. Rastogi, R. Sharma, and S. Tripathi, "EzPC: Programmable and efficient secure Two-Party computation for machine learning," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 496–511.

[37] "Microsoft SEAL (release 3.6)," https://github.com/Microsoft/SEAL, November 2020.

[38] F. Boemer, S. Kim, G. Seifu, F. DM de Souza, and V. Gopal, "Intel HEXL: accelerating homomorphic encryption with intel avx512-ifma52," in *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2021, pp. 57–62.
[39] X. Wang, A. J. Malozemoff, and J. Katz, "EMP-toolkit: Efficient Multi-Party computation toolkit," https://github.com/emp-toolkit, 2016.
[40] A. Semwal and N. D. Londhe, "Computer aided pain detection and intensity estimation using compact CNN based fusion network," *Applied Soft Computing*, vol. 112, p. 107780, 2021.
[41] O. Rudovic, V. Pavlovic, and M. Pantic, "Automatic pain intensity estimation with heteroscedastic conditional ordinal random fields," in *International Symposium on Visual Computing*. Springer, 2013, pp. 234–243.
[42] K. Gupta, D. Kumaraswamy, N. Chandran, and D. Gupta, "LLAMA: A low latency math library for secure inference," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 4, pp. 274–294, 2022.
[43] Z. Lyu, T. Yu, F. Pan, Y. Zhang, J. Luo, D. Zhang, Y. Chen, B. Zhang, and G. Li, "A survey of model compression strategies for object detection," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 48 165–48 236, 2024.