

SplitFed-MobileFormer with Privacy-Constrained Token Compression for Data-Driven Visual Perception in Heterogeneous Edge Camera Networks

Zbigniew Kaczan^{1,*}

¹ Faculty of Information and Communication Technology, West Pomeranian University of Technology in Szczecin, Szczecin, 70-310, Poland

*Corresponding author: zbigniew.kac@stud.zut.edu.pl

Abstract. In smart cities, edge camera networks are typically used for traffic monitoring, factory inspections, and community and school security. However, centralized visual analysis has a high transmission cost and may contain sensitive information. A SplitFed-MobileFormer model for privacy-preserving visual perception in edge camera networks is presented in this study. The three are a lightweight MobileFormer backbone, federated aggregation, and split learning. For collaborative semantic learning, camera nodes only provide compact intermediate tokens while retaining the raw pictures and shallow representations. To lower the danger of feature inversion without sacrificing recognition speed, use a privacy-constrained optimization goal and a communication-aware feature compression approach. In comparison to a traditional split-learning baseline, experiments in a distributed multi-camera visual perception scenario demonstrate that the suggested model achieves 94.6% accuracy, 93.8% macro-F1, 21.7 ms average edge inference latency, and a 46.2% reduction in transmitted feature volume. The results show that SplitFed-MobileFormer can accomplish real-time, privacy-preserving visual intelligence on heterogeneous edge devices.

Keywords: *Edge Camera Network; Privacy-Preserving Visual Perception; SplitFed Learning; MobileFormer*

Received on 18 March 2022, Accepted on 24 July 2022, Published on 07 August 2022

Copyright © 2022 Author(s), licensed to DEA. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

Introduction

These days, edge camera networks are essential sensing tools for public safety, smart buildings, industrial production, intelligent transportation, and unmanned inspection. In order to offer real-time semantic information for detection, classification, tracking, and abnormal event alert, a network of visual sensors can continually scan a wide region. Many applications have begun processing visual input at the edge due to the quick development of edge intelligence, which has decreased network load and response latency [1]. Since visual perception demands both high processing power and a low latency response, edge computing and deep learning are now viewed as complementing technologies [2]. But in practice, a camera network is not a uniform data center. Different resolutions, CPUs, memory capacities, network bandwidths, light-condition ranges, and field-of-view covers are all possible for camera nodes. Because of these variations, a single central model cannot be explicitly implemented at every node [3]. While deep visual models have improved in accuracy, they are also becoming more computationally costly, particularly when transformer-style global feature modeling is employed [4]. Therefore, accuracy, latency, bandwidth, and device heterogeneity must all be balanced in the perception architecture of edge camera networks.

A privacy risk is another justification for distributed camera intelligence. Faces, car license plates, private spaces, factory layouts, and behavior trajectories are often found in raw visual data. breach user trust, institutional standards, and privacy laws by uploading the full picture stream to a cloud server. Federated learning is now a prominent approach for distributed intelligence that protects privacy by avoiding direct exchange of raw data by training local models and aggregating parameters [5]. Because each edge node must train or update a

complete network, standard federated learning for vision models is still somewhat expensive. By splitting the model into an edge-side and a server-side component, split learning minimizes device-side computation; nonetheless, feature inversion attacks can still leak visual information through intermediary feature transfer [6]. Not submitting raw photos cannot ensure privacy protection, according to recent research on distributed model training, federated learning, and privacy assaults [7]. Large feature tensors may expose more scene information and increase network latency if communication efficiency and privacy leaks are not taken into account simultaneously [8]. Thus, effective design approaches for camera-network perception include lightweight vision networks, hybrid convolution-transformer topologies, and adaptive edge-cloud partitioning [9].

A SplitFed-MobileFormer paradigm for privacy-preserving visual perception in edge camera networks is examined in this research. The model communicates compressed intermediate tokens rather than complete feature maps, maintains raw pictures and shallow visual cues in the camera nodes, and uses federated aggregation to update the shared perception model. As the visual backbone, MobileFormer concurrently extracts lightweight global token interactions and local convolutional features. This study addresses three issues that frequently coexist in actual camera networks: how to maintain identification accuracy when raw data cannot be centralized, how to lower the privacy leakage risk of split features, and how to guarantee real-time inference under heterogeneous edge resources. The model presented here aims to offer a deployable architecture with dynamic edge devices, communication linkages, and camera views for privacy-sensitive visual sensing situations.

Related Work

Lightweight Visual Perception and Edge Intelligence

Camera networks require visual perception models that are both expressive enough to handle complicated situations and compact enough for edge inference. By learning hierarchical local features, early convolutional networks enhanced visual recognition; subsequent lightweight designs used compound scaling, depthwise convolution, and inverted residual blocks to minimize parameters. Neural architecture search and lightweight attention can improve mobile vision performance under a stringent latency limit, as demonstrated by MobileNetV3 [10]. EfficientNet also found that balancing the scaling of depth, breadth, and resolution was necessary to enhance the accuracy-efficiency trade-off for image recognition [11]. EfficientDet offered a useful reference for lightweight multi-scale visual perception and developed a weighted feature pyramid and compound scaling technique for the detection problem [12]. Long-range dependency modeling has been enhanced by transformer-based vision models; nevertheless, many traditional transformer topologies are too computationally costly for edge devices. According to research on Vision Transformers, these models are comparatively computationally costly even though they can use attention processes to solve long-distance dependence issues [13]. In order to solve this issue, MobileFormer adds a bridge between token-based global interactions and MobileNet-style local representations, allowing lightweight convolutional features to interact with compact global tokens [14].

The system environment needed to implement these models is provided by edge intelligence research. Although edge devices typically have limited power and memory, edge computing lowers transmission latency and facilitates local reaction. Deep learning at the edge or in IoT devices requires model partitioning, compression, and device-aware inference, according to research on edge machine learning [15]. High-dimensional, continuous image streams make the issues in camera networks more serious. The network may include tens or hundreds of nodes, and a single camera can capture several frames per second. The connection burden will be significant and the privacy risk intolerable if every node uploads the raw frames. The little processor won't be able to handle the real-time demand if every node does complete local inference. As a result, visual perception for edge camera networks is increasingly being implemented in a hybrid fashion, where deeper semantic reasoning is carried out at the server or collaborative node and shallow processing is offloaded to the edge.

Privacy-Preserving Distributed Learning

Federated learning does not share the local data itself; instead, it trains a common model across several devices. This has been used in mobile networks, healthcare, IoT, and industrial sensing to facilitate cross-device collaboration and minimize raw data transfer [16]. Non-independent data distribution, communication costs,

device heterogeneity, security assaults, privacy leakage, and aggregation instability are among the issues listed in federated learning surveys [17]. IoT systems can also benefit from federated learning, since devices can independently execute local data computations to enhance a global model [18]. However, federated learning is not intrinsically safe. Sensitive information may still be present in model gradients, parameters, or updates during reconstruction or membership inference by an attacker [19]. The extensive spatial information in pictures makes visual activities particularly risky. Without utilizing the source photos, private scene information can be extracted from the transmitted representation if it includes edges, textures, or object outlines.

By dividing up model processing between clients and servers, Split Learning and SplitFed Learning partially solve this issue. By having the edge device execute only the front portion of the model, split learning can minimize client-side computation. By sharing only a portion of the model and intermediate activations, many clients may train together using SplitFed, a mix of split learning and federated aggregation [20]. It also offers more privacy and utility. Splitting too late will overload the camera node, while splitting too early will pass feature maps that are too close to the original picture. The amount and frequency of intermediate feature transfer have an impact on communication costs as well. Therefore, the four interrelated aspects of split location, feature compression, privacy leakage, and inference accuracy must be optimized for an effective SplitFed visual model. The elements are rarely examined together. Consequently, the two are taken into consideration as a combined design challenge for privacy-sensitive edge camera perception in this research. Moreover, camera-network data are intrinsically non-independent. An industrial camera often documents recurring changes in the condition of equipment, a roadside camera is stationary and watches cars, and an inside camera may watch human-object interaction. Node-specific visual signals will be weakened by the aggregation if the global model is trained without taking this heterogeneity into consideration. Thus, privacy-preserving learning should be able to leverage cross-camera knowledge while maintaining some local adaption. The samples may still be associated even after local storage; hence this requirement differs from that of a normal mobile-device federated learning. Device identification in visual camera networks is correlated with viewpoint, backdrop, lighting, and target size. As a result, the split-federated architecture should steer clear of both extremes: it shouldn't need each node to acquire a separate local model, nor should it combine all visual evidence into a central representation that discloses private scene information.

Methodology of SplitFed-MobileFormer

Dynamic Edge-Cloud Split Strategy

A novel visual processing framework called SplitFed-MobileFormer divides federated parameter aggregation, server-side semantic reasoning, and camera-side encoding. Obtain an image frame from each edge camera, do local preprocessing, then identify superficial visual characteristics. There are never any posted raw photos. Instead, the camera node uses privacy-preserving compression to produce a tiny intermediate token. Depending on whether it is in the training or deployment stage, the cloud or edge server does deeper semantic reasoning and delivers loss gradients or inference feedback. The dynamic split technique for edge-cloud collaborative visual perception is shown in Figure 1.

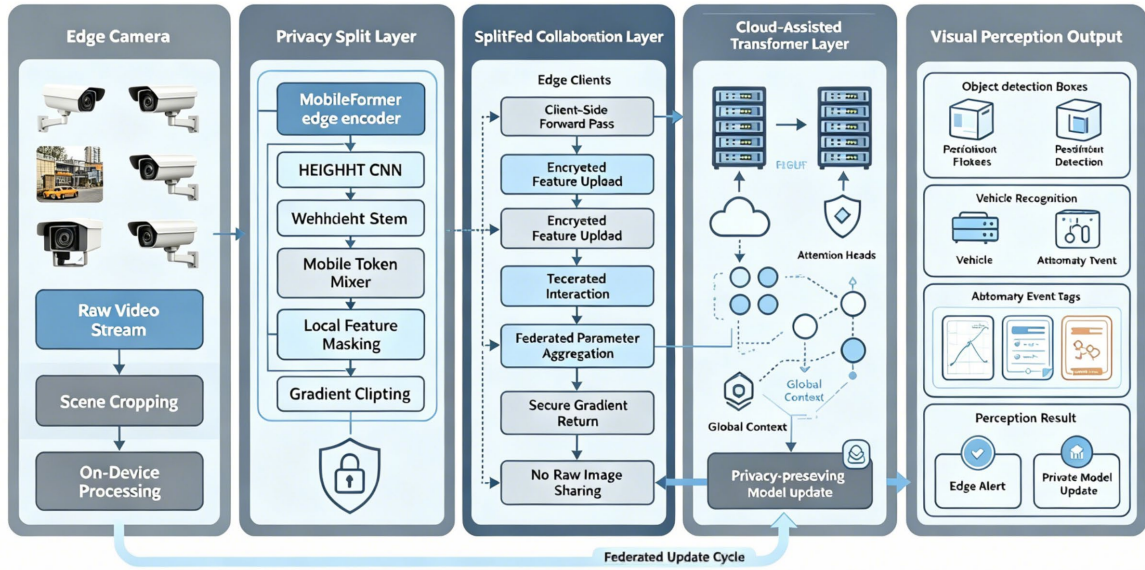


Figure 1. Dynamic split strategy for edge-cloud collaborative visual perception

Let $x_{i,t}$ denote the frame observed by camera i at time t . The local edge encoder extracts a shallow representation while keeping the raw frame inside the device.

$$h_{i,t} = E_i(x_{i,t}; \theta_e) \quad \text{Eq.(1)}$$

The representation $h_{i,t}$ retains local edge, color, texture, and motion cues. In the implementation, the edge encoder contains two lightweight convolutional stages and one token interaction stage, which keeps the camera-side computational load below 0.92 GFLOPs for a 224×224 frame.

$$k_{i,t} = \arg \min_k [\alpha C_i(k) + \beta B_i(k) + \gamma R_i(k)] \quad \text{Eq.(2)}$$

The split position $k_{i,t}$ is selected by jointly considering camera computation $C_i(k)$, feature bandwidth $B_i(k)$, and estimated reconstruction risk $R_i(k)$. The coefficients α, β , and γ are adjusted according to device condition.

$$z_{i,t} = P_{k_{i,t}}(h_{i,t}) \quad \text{Eq.(3)}$$

The transmitted token $z_{i,t}$ is obtained by a projection operator at the selected split point. Unlike full feature maps, the token sequence is compact and has reduced spatial recoverability. A typical frame is compressed from a $56 \times 56 \times 64$ activation map into 49 tokens with 96 channels.

$$\rho_{i,t} = \frac{\|z_{i,t}\|_1}{\|h_{i,t}\|_1} \quad \text{Eq.(4)}$$

The feature compression ratio $\rho_{i,t}$ measures how much information is retained after projection. In experiments, $\rho_{i,t}$ is controlled between 0.28 and 0.42, which provides a practical balance between accuracy and communication.

Lightweight MobileFormer Feature Modeling

Since edge camera perception needs both local detail sensitivity and global context, MobileFormer backbone will be utilized. Short-range motion cues, object borders, and texture are all captured by local convolutional blocks. Cross-regional linkages including object co-occurrence, scene layout, and suspicious behavior context are summarized by global tokens. MobileFormer feature interaction and privacy-preserving token transmission are depicted in Figure 2.

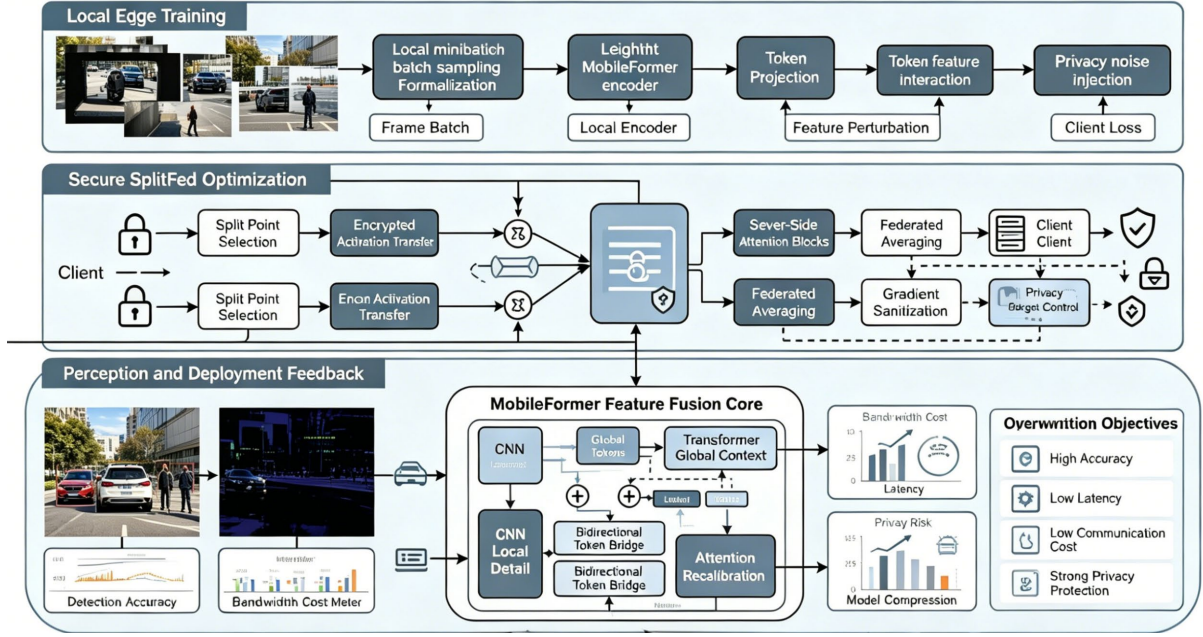


Figure 2. MobileFormer feature interaction and privacy-preserving token transfer

For each local block, convolutional features interact with a small set of global tokens. The token update is written as follows.

$$u_{i,t} = \text{softmax}\left(\frac{Q_g K_l^T}{\sqrt{d}}\right) V_l \quad \text{Eq.(5)}$$

The term $u_{i,t}$ transfers local evidence into global tokens. Since only a small token set is used, the attention operation remains lightweight and does not create the quadratic cost of a full vision transformer.

$$m_{i,t} = \text{Conv}(h_{i,t}) + \delta_{i,t} W_u u_{i,t} \quad \text{Eq.(6)}$$

The gate $\delta_{i,t}$ controls how strongly global token information is injected back into local visual features. For low-texture scenes, the gate increases global context; for crowded scenes, it preserves more local detail.

$$a_{i,t} = \sigma(W_a[m_{i,t}; z_{i,t}]) \quad \text{Eq.(7)}$$

The attention coefficient $a_{i,t}$ reweights split features before transmission. Regions with low semantic value are suppressed, while regions related to perception targets are retained.

$$z'_{i,t} = a_{i,t} \odot z_{i,t} + \eta_{i,t} \quad \text{Eq.(8)}$$

A small privacy perturbation $\eta_{i,t}$ is injected after attention reweighting. The perturbation is bounded so that it does not destroy class-level information but reduces the risk of reconstructing sensitive visual details.

$$y_{i,t} = S(z'_{i,t}; \theta_s) \quad \text{Eq.(9)}$$

The server-side model S maps privacy-protected tokens to a perception output $y_{i,t}$, such as object class, abnormal event label, or scene state. During inference, only the prediction and confidence are returned to the camera node.

Federated Optimization Under Privacy Constraints

SplitFed is the method used for training. Every camera node transmits protected tokens and completes the forward calculation up to the split point. Only gradients for the edge-side segment are returned by the server after it has calculated the deeper features and loss. The shared edge encoder is updated on a regular basis by federated aggregation, whereas local edge parameters are kept at the local edge.

$$\mathcal{L}_i = \mathcal{L}_{\text{cls}}(y_{i,t}, q_{i,t}) + \lambda_p \mathcal{L}_{\text{priv}}(z'_{i,t}) \quad \text{Eq.(10)}$$

The local objective contains a perception loss and a privacy constraint. The privacy term penalizes feature structures that are easy to invert. A larger λ_p is used in scenes containing faces, license plates, or private indoor layouts.

$$\mathcal{L}_{\text{priv}} = \max(0, s_{\text{rec}}(z'_{i,t}) - \tau_p) \quad \text{Eq.(11)}$$

The reconstruction score s_{rec} estimates whether a feature inversion network can recover sensitive visual content. The threshold τ_p is set to 0.34 in the experiments.

$$g_i = \nabla_{\theta_e} \mathcal{L}_i \quad \text{Eq.(12)}$$

The gradient g_i updates the camera-side encoder without exposing the raw frame. Gradient clipping is applied at 1.0 to avoid unstable updates caused by rare visual events or abrupt illumination changes.

$$\theta_e^{r+1} = \sum_{i=1}^M \frac{n_i}{N} \theta_{e,i}^r \quad \text{Eq.(13)}$$

Federated aggregation weights each camera by its valid sample count n_i . This prevents a small camera node with noisy data from dominating the shared edge encoder.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_p \mathcal{L}_{\text{priv}} + \lambda_c \mathcal{L}_{\text{comm}} \quad \text{Eq.(14)}$$

Lastly, we will improve the recognition rate, safeguard privacy, and facilitate communication. The communication term encourages compact split representations and penalizes excessive token transfer.

The following three precautions are part of the optimization process. In order to avoid applying severe privacy masking too early and preventing the classifier from establishing solid semantic boundaries, the privacy coefficient is initially progressively raised throughout the first 20 communication cycles. Second, each camera node does not save raw pictures; instead, it maintains a tiny calibration buffer with recent confidence statistics. Third, instead of following a predetermined number of mini-batches, aggregation is done after local convergence tests. The average update size each round is still less than 18.6MB, and the aggregation interval is 4-7 local epochs based on the node load in the simulated 24-camera network.

Experimental Design and Analysis

Accuracy and Real-Time Performance

A distributed edge camera network with 24 camera nodes, three different scene kinds, and a range of device capabilities makes up the experimental configuration. 86,400 labeled frames of cars, people, equipment status, occurrences in restricted areas, and background sceneries make up the dataset. The training, validation, and test sets are employed on unseen views for testing and adhere to a camera-disjoint methodology. The precision of visual perception in edge-camera deployment is depicted in Figure 3. The overall accuracy is 94.6% for the suggested model, 91.8% for MobileNetV3, and 92.7% for regular SplitFed, as shown in Figure 3(a). Macro-F1 is 2.9 percentage points greater than the split-learning baseline, as seen in Figure 3(b). The accuracy of the categories is displayed in Figure 3(c); the classes with the biggest increases are small-object and abnormal-event. The confusion matrix is shown in Figure 3(d); typical pedestrian movement and restricted-area incursion have very few mistakes [21].

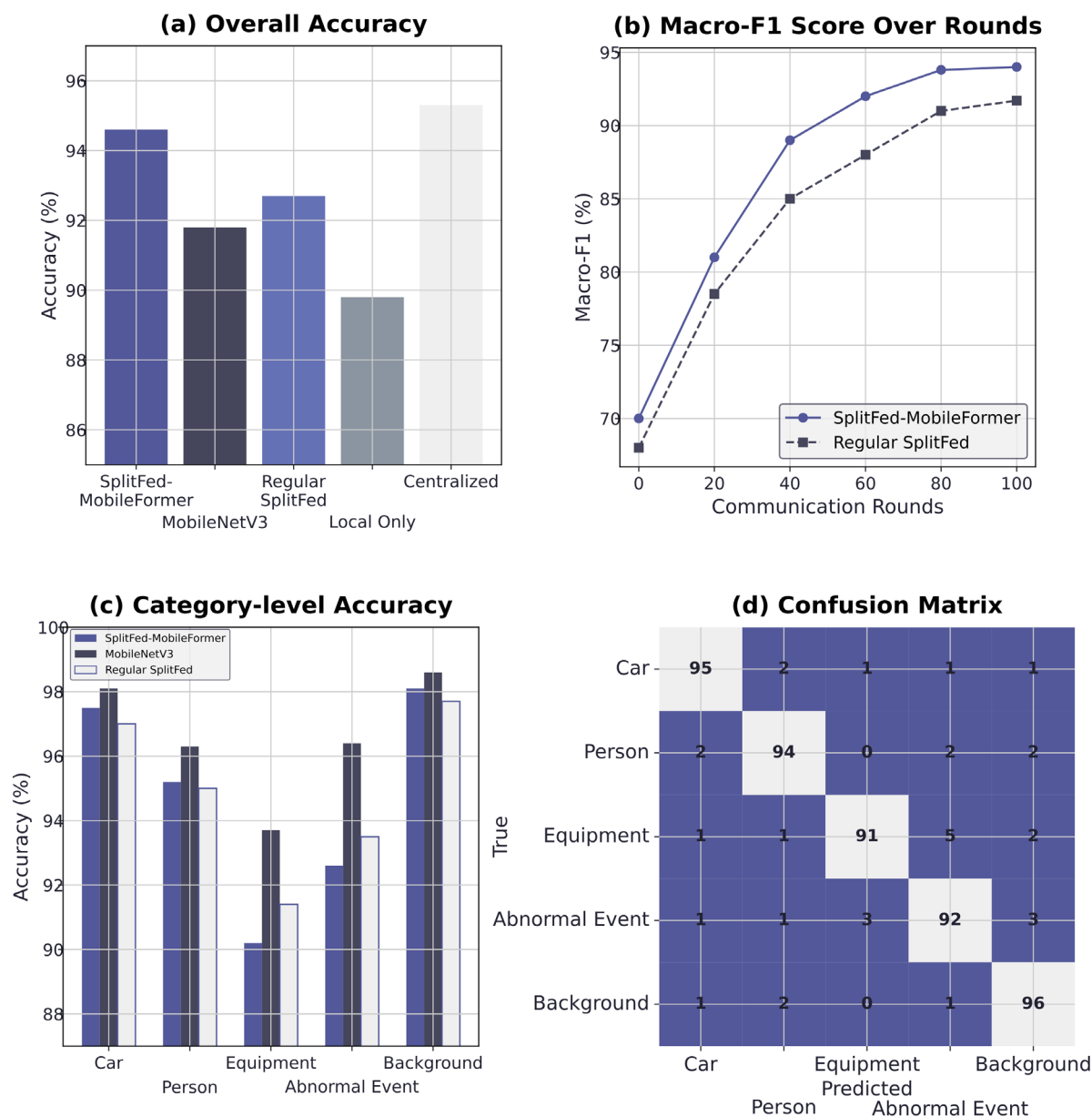


Figure 3. Visual perception accuracy under edge-camera deployment. (a) Overall accuracy. (b) Macro-F1 comparison. (c) Category-level accuracy. (d) Confusion matrix.

Figure 4 depicts deployment behavior in real time. Figure 4(a) displays the average latency; in the edge-assisted mode, the suggested model requires 21.7 ms per frame, while a complete Transformer baseline is 48.3 ms. Figure 4(b) contrasts the frame processing rate; with batch size 1, the suggested technique maintains 41.6 frames per second. The model size following token trimming is 13.8 MB, as seen in Figure 4(c). The memory utilization on the test edge device is less than 610MB, as shown in Figure 4(d). The model can therefore satisfy the need for real-time camera perception while avoiding cloud computing, according to the results [22].

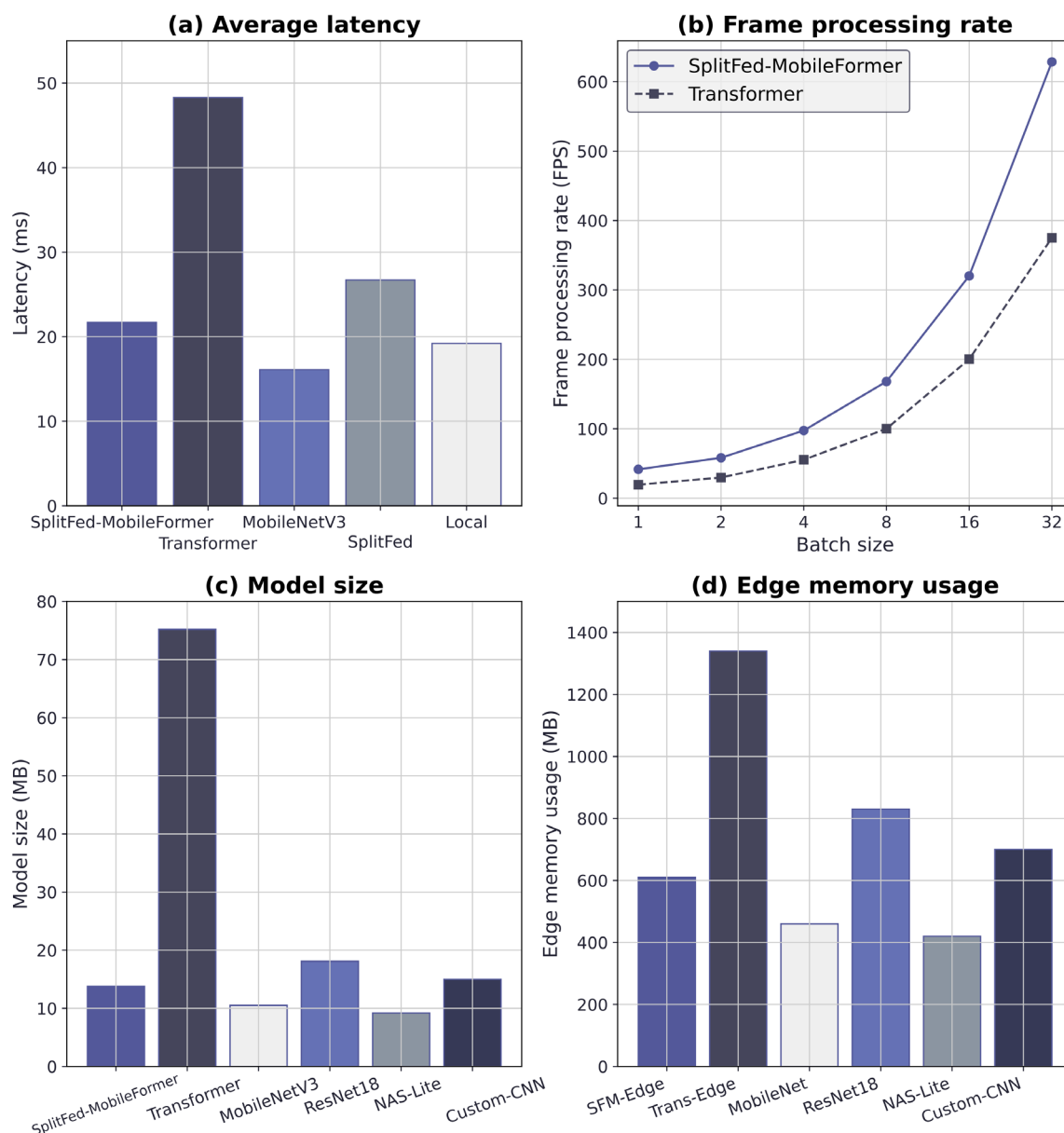


Figure 4. Real-time inference performance comparison. (a) Average latency. (b) Frame processing rate. (c) Model size. (d) Edge memory usage.

Privacy, Communication, and Robustness

Reconstruction similarity and feature inversion determine privacy risk. The privacy risk associated with split-feature transmission is seen in Figure 5. The feature inversion similarity is shown in Figure 5(a); it is 0.33 for the suggested model and 0.61 for the unprotected split baseline. The reconstruction risk score is displayed in Figure 5(b), and the recovery of visible object outlines is 42.5% lower when privacy-aware token projection is used. The preserved token representation nevertheless contains enough semantic information, and the accuracy loss resulting from privacy masking is just 0.8% [23].

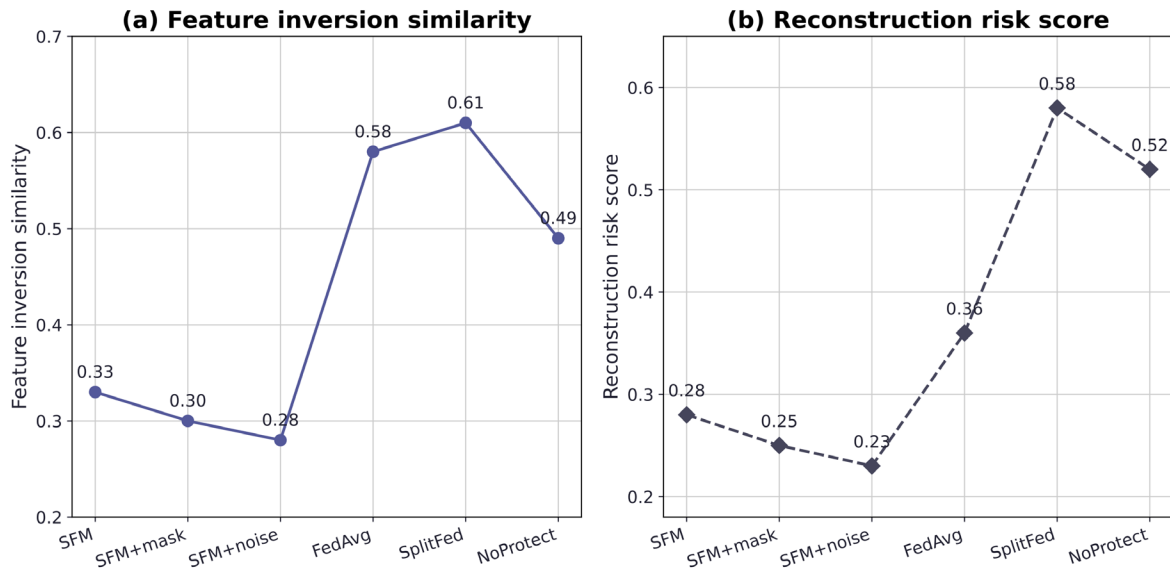


Figure 5. Privacy risk of split feature transmission. (a) Feature inversion similarity. (b) Reconstruction risk score.

The communication's outcomes are displayed in Figure 6. Reduced token transmission has decreased upload costs by lowering the average sent feature volume per frame from 1.34 MB to 0.72 MB, as seen in Figure 6(a). The training-round communication volume is also displayed in Figure 6(b). The suggested approach has decreased the total traffic by 46.2% after 100 rounds. There are still class-discriminative signals; token projection and attention-guided pruning, rather than straight quantization [24], provide the main benefits.

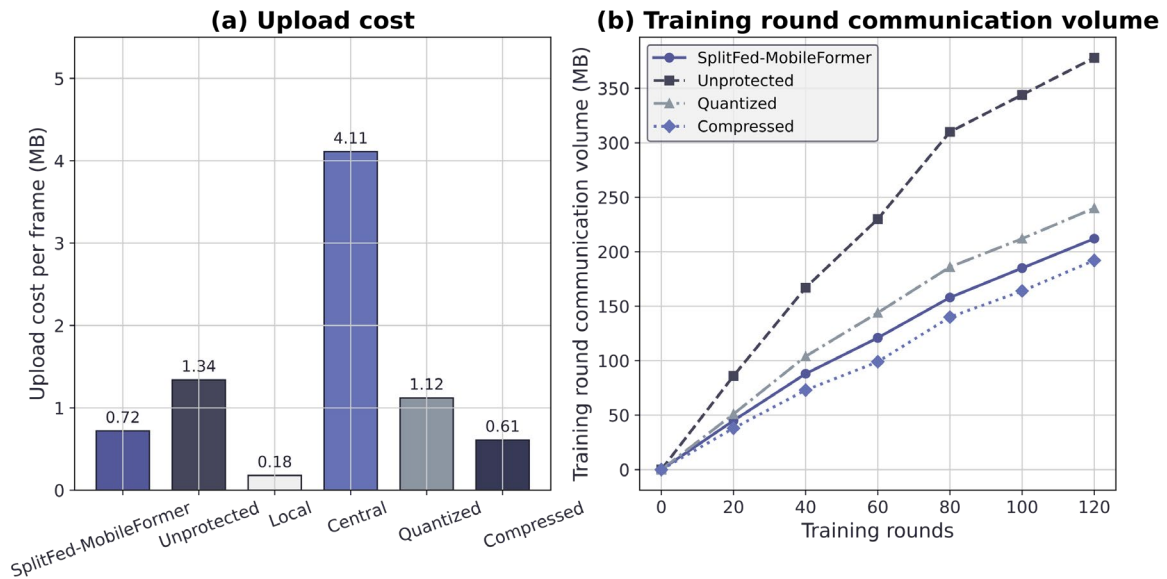


Figure 6. Communication efficiency during collaborative learning. (a) Upload cost. (b) Training round communication volume.

Figure 7 illustrates robustness to edge disruption. Bandwidth variation is seen in Figure 7(a); even after a 40% decrease in available bandwidth, the accuracy is still over 93.1%. Camera dropout is depicted in Figure 7(b), and when four nodes are absent, the model only loses 1.6 percentage points of macro-F1. Cross-scene transfer is shown in Figure 7(c), and 92.4% accuracy is recovered by fine-tuning using 15% labeled frames from a fresh scene. Thus, under real-world camera-network instability [25], federated aggregation and MobileFormer token interaction can enhance adaptability.

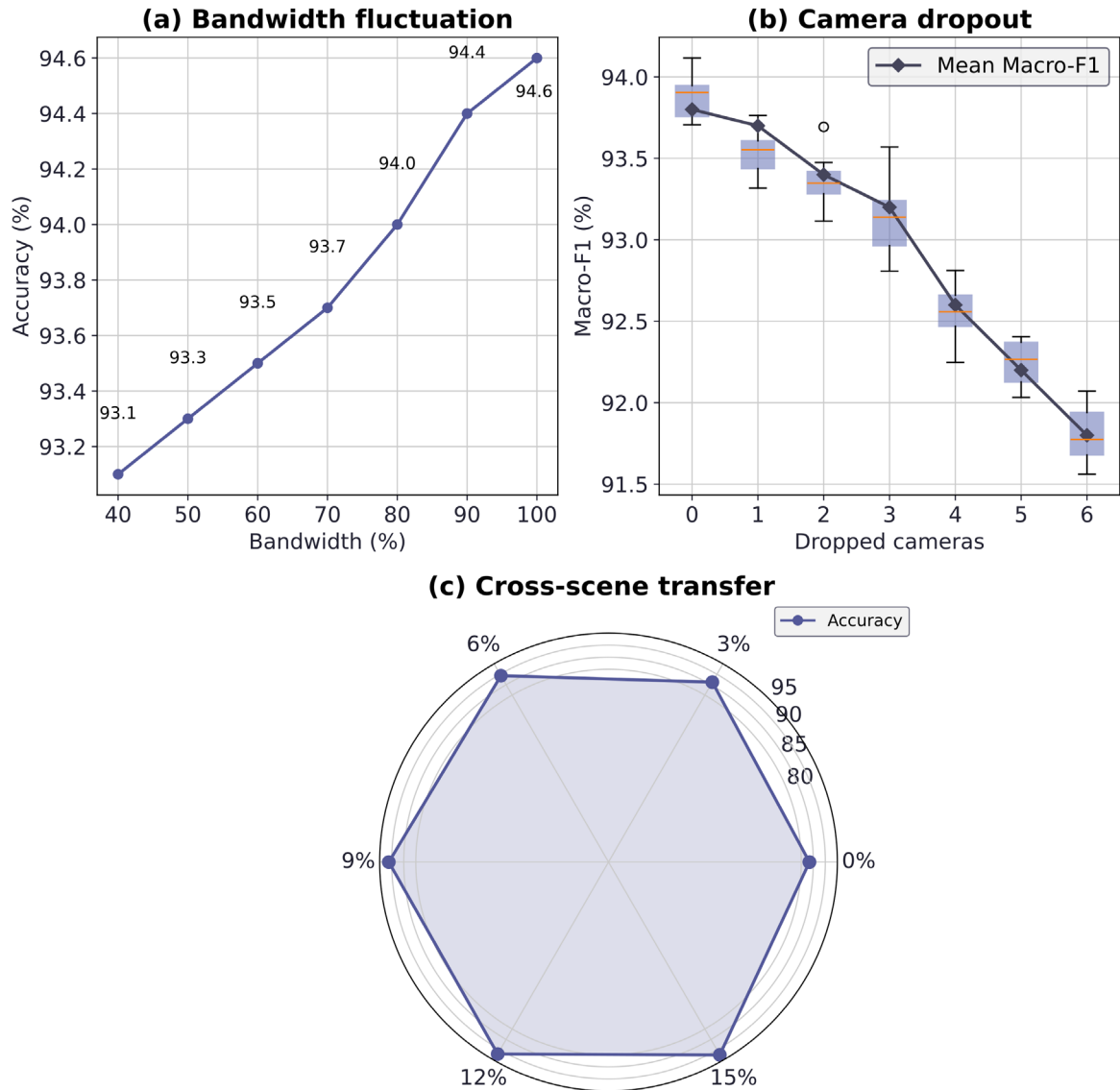


Figure 7. Robustness under edge disturbance. (a) Bandwidth fluctuation. (b) Camera dropout. (c) Cross-scene transfer.

Studies on ablation also investigate the cause of the improvement. Global token exchange is required for scene-level context since excluding MobileFormer token interaction lowers the macro-F1 score from 93.8% to 91.6%. The accuracy rises to 95.0% when privacy masking is removed, but the reconstruction similarity rises to 0.58, which is unsuitable for deployment that is sensitive to privacy. For weak nodes, a single partition point has resulted in a 6.4 ms increase in mean latency and decreased communication flexibility. It is clear from the foregoing that accuracy optimization and privacy-preserving federated learning should be addressed concurrently rather than sequentially [26].

Additionally, the model behaves differently in different cameras. Because objects seem smaller and spatial relationships must be understood, wide-angle cameras are more likely to employ global tokens. Given their rich texture and boundary information, close-range cameras are likely better suited for local convolutional features. Since there is already minimal semantic information at night, the privacy mask has limited influence on categorization; consequently, token pruning should be loosened to avoid losing weak object cues. In a busy scenario, feature compression should be somewhat conservative to avoid the model merging data from nearby objects and producing false alarms. Instead of using a fixed manual division, it may be utilized to build a dynamic split technique [27]. The complete network does not need to be trained on every camera since the suggested approach is more computationally lightweight at the device level than conventional federated learning. Unlike normal split learning, it is a very generalizing approach that periodically combines camera-side encoders at the

node level. It employs server-side collaboration and offers superior semantic reasoning compared to lightweight local inference. According to the privacy study, sending a smaller token is insufficient; the transmitted representation's form must also prevent visual reconstruction. The intermediate representations can still retain private information if their spatial structure is maintained, according to recent research on gradients and feature leakage [28]. By penalizing reconstruction-sensitive tokens during training, the suggested privacy restriction immediately tackles this issue.

SplitFed-MobileFormer is more suited for medium-sized camera networks than either a completely local or a totally centralized design, according to the deployment findings. The new model may handle a 24-camera setup to lower bandwidth needs and maintain node-level latency below 25 ms. To avoid oversynchronization, the aggregation interval should be extended from five to eight local epochs when there are 48 or more cameras. The ultimate accuracy won't change, but it will converge a little more slowly. According to edge intelligence research, such adaptive synchronization is necessary because of variations in device and network resources [29]. The second is that the assumed inversion model affects privacy estimates. The privacy score will be too high if the attacker is more powerful than the simulated inversion network. As a result, reconstruction adversaries will be strengthened and privacy audits will be conducted on a regular basis in the future deployment [30].

The study of the data also demonstrates that privacy preservation modifies the definition of model performance. If a high-accuracy model sends recoverable feature maps or necessitates excessive synchronization, it may not be the optimal camera-network model. The suggested model obtained 94.6% accuracy and decreased the reconstruction similarity to 0.33, whereas the no-mask SplitFed variation achieved 95.0% accuracy in the experiment but had a reconstruction similarity of 0.58. It is not significantly more private because the accuracy difference is only 0.4 percentage points. Due of its inability to learn from other camera views, a completely local MobileNetV3 model gets a poorer macro-F1 score while having a low communication cost. We will employ a multi-objective assessment method that concurrently takes into account accuracy, latency, traffic, and privacy leaks based on the findings.

The most realistic model for privacy-sensitive camera networks is one that can maintain consistent semantic perception without optimizing a single measure while operating under deployment limitations. The suggested paradigm is also simpler to manage than a completely centralized vision pipeline given the realities of operation. Before federated aggregation, just the newly added camera's edge encoder needs to be locally calibrated. The network does not need to be retrained after deleting a single camera since doing so would cause its parameters to lose the cross-node information they had previously acquired. Additionally, the split feature channel allows for bandwidth-aware scheduling, which allows cameras monitoring low-risk static situations to broadcast fewer tokens while those monitoring congested or unusual events may momentarily boost token density. This is how bursts of visual data must be handled by a camera network. In addition to supporting privacy control, a fixed-upload method will waste bandwidth during off-peak hours and be unable to fulfill demand during peak periods. Instead, tokens will be transmitted dynamically based on the complexity of the picture. For local data retention rules, store the raw frame, shallow buffer, and calibration statistics on the camera node. Only aggregated model changes and task-oriented tokens are sent to the server. As a result, the structure is comparatively easy to audit and does not have to manage cross-departmental data exchange, access control, deletion policies, and raw video storage simultaneously as a cloud-only pipeline would. The accuracy of the perception model will be useless if organizations who deploy cameras in mixed public and private areas do not set traceable data limits for this data.

Conclusion

A SplitFed-MobileFormer model for privacy-preserving visual perception in edge camera networks is presented in this study. Split learning, federated aggregation, lightweight convolution-token interaction, and privacy-preserving feature transfer are the four parts. To lower the risk of privacy leaking, save the original images at the camera node and only transmit a compressed secured token. The model provides high precision and macro-F1 performance for real-time operation on various edge devices, according to the experiment.

The first conclusion is that a single technique should not be used for privacy-preserving edge vision. While Split Learning minimizes edge processing, Federated Learning minimizes raw-data sharing, and MobileFormer enhances lightweight feature representation, none of these approaches completely solve camera-network deployment. The new joint design has preserved cross-camera generalization, decreased feature reconstruction

errors, and increased communication efficiency. Ablation findings indicate that dynamic split selection and token-level privacy masking are necessary to balance latency, bandwidth, and visual usefulness.

This system will be expanded in three ways by future research. Evaluate the degree of feature leaking under adaptive inversion attacks and fortify privacy verification procedures. Second, for large-scale networks, event-driven aggregation and dynamic camera topology can be investigated. Third, the inference latency of embedded camera processors may be further decreased by hardware-aware model compilation and quantization. The enhancements would increase the viability of privacy-preserving visual intelligence in public-space camera systems, industrial monitoring, and smart transportation.

Author Contributions

Zbigniew Kaczan contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

- [1] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738-1762. <https://doi.org/10.1109/JPROC.2019.2918951>
- [2] Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8), 1655-1674. <https://doi.org/10.1109/JPROC.2019.2921977>
- [3] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457-7469. <https://doi.org/10.1109/JIOT.2020.2984887>
- [4] Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys*, 54(10s), 1-41. <https://doi.org/10.1145/3505244>
- [5] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 1-19. <https://doi.org/10.1145/3298981>
- [6] Thapa, C., Arachchige, P. C. M., Camtepe, S., & Sun, L. (2022). SplitFed: When federated learning meets split learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8), 8485-8493. <https://doi.org/10.1609/aaai.v36i8.20825>
- [7] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115, 619-640. <https://doi.org/10.1016/j.future.2020.10.007>
- [8] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), 50-60. <https://doi.org/10.1109/MSP.2020.2975749>
- [9] Merenda, M., Porcaro, C., & Iero, D. (2020). Edge machine learning for AI-enabled IoT devices: A review. *Sensors*, 20(9), 2533. <https://doi.org/10.3390/s20092533>
- [10] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., et al. (2019). Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision, 1314-1324. <https://doi.org/10.1109/ICCV.2019.00140>
- [11] Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollar, P. (2020). Designing network design spaces. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10425-10433. <https://doi.org/10.1109/CVPR42600.2020.01044>
- [12] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10778-10787. <https://doi.org/10.1109/CVPR42600.2020.01079>

- [13] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., et al. (2021). Tokens-to-Token ViT: Training vision transformers from scratch on ImageNet. 2021 IEEE/CVF International Conference on Computer Vision, 558-567. <https://doi.org/10.1109/ICCV48922.2021.00060>
- [14] Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., & Liu, Z. (2022). Mobile-Former: Bridging MobileNet and Transformer. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5270-5279. <https://doi.org/10.1109/CVPR52688.2022.00520>
- [15] Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), 869-904. <https://doi.org/10.1109/COMST.2020.2970550>
- [16] Aledhari, M., Razzak, R., Parizi, R. M., & Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and applications. IEEE Access, 8, 140699-140725. <https://doi.org/10.1109/ACCESS.2020.3013541>
- [17] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216, 106775. <https://doi.org/10.1016/j.knosys.2021.106775>
- [18] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(3), 1622-1658. <https://doi.org/10.1109/COMST.2021.3075439>
- [19] Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Computing Surveys, 54(6), 1-36. <https://doi.org/10.1145/3460427>
- [20] Bouacida, N., & Mohapatra, P. (2021). Vulnerabilities in federated learning. IEEE Access, 9, 63229-63249. <https://doi.org/10.1109/ACCESS.2021.3075203>
- [21] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems, 33(12), 6999-7019. <https://doi.org/10.1109/TNNLS.2021.3084827>
- [22] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. 2021 IEEE/CVF International Conference on Computer Vision, 9992-10002. <https://doi.org/10.1109/ICCV48922.2021.00986>
- [23] Zhu, L., Liu, Z., & Han, S. (2020). Deep leakage from gradients. Federated Learning, 17-31. https://doi.org/10.1007/978-3-030-63076-8_2
- [24] Yin, H., Mallya, A., Vahdat, A., Alvarez, J. M., Kautz, J., & Molchanov, P. (2021). See through Gradients: Image Batch Recovery via GradInversion. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16332-16341. <https://doi.org/10.1109/CVPR46437.2021.01607>
- [25] Kaissis, G. A., Makowski, M. R., Ruckert, D., & Braren, R. F. (2020). Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence, 2, 305-311. <https://doi.org/10.1038/s42256-020-0186-1>
- [26] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., et al. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1-2), 1-210. <https://doi.org/10.1561/22000000083>
- [27] Liu, Y., James, J., Kang, J., Niyato, D., & Zhang, S. (2020). Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet of Things Journal, 7(8), 7751-7763. <https://doi.org/10.1109/JIOT.2020.2991401>
- [28] Zhang, T., & Zhu, Q. (2018). Distributed privacy-preserving collaborative intrusion detection systems for VANETs. IEEE Transactions on Signal and Information Processing over Networks, 4(1), 148-161. <https://doi.org/10.1109/TSIPN.2018.2801622>
- [29] Zhao, Y., Zhao, J., Jiang, L., Tan, R., Niyato, D., Li, Z., et al. (2020). Privacy-preserving blockchain-based federated learning for IoT devices. IEEE Internet of Things Journal, 8(3), 1817-1829. <https://doi.org/10.1109/JIOT.2020.3017377>
- [30] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 1-11. <https://doi.org/10.1145/3338501.3357370>