Preva: Protecting Inference Privacy through Policy-based Video-frame Transformation

Rui Lu*‡ Siping Shi*‡, Dan Wang*, Chuang Hu†, Bihai Zhang*
*The Hong Kong Polytechnic University {csrlu, cssshi, csdwang, csbzhang}@comp.polyu.edu.hk
†Wuhan University handc@whu.edu.cn ‡Co-primary Author

***Abstract*—Real-time edge-cloud video analytics systems have been widely used to support such applications as traffic counting, surveillance, autonomous driving, Metaverse, etc. In such a system, the edge and the cloud cooperatively conduct model inference of the video frames captured by the camera of the edge, using a trained DNN model of the video analytics application. The edge conducts initial analytics on the video frames to a split layer of the DNN model; and then sends intermediate results to the cloud for follow-up analytics. In this paper, we show that an attacker can perform reconstruction attacks to the intermediate results; and private information of the raw video frames, e.g., a plate number of a car, can be leaked.**

In this paper, we present Preva, a new Privacy preserving Real-time Edge-cloud Video Analytics system. The core idea of Preva is to conduct image transformation on the video frames, as preprocessing, prior to the video frames starting the edge-cloud video analytics process, so that during edge-cloud video analytics, the intermediate results will not leak private information under attack. We design a policy-based video-frame transformation scheme. Given the resource constraints of the edge, Preva ensures high accuracy in the final video analytics results and minimizes privacy leakage in any split layer. We present a formal privacy analysis and we show that Preva can guarantee privacy leakage under the reconstruction attacks of both outsider attackers and insider attackers. We evaluate Preva through three video analytics applications and we show that Preva outperforms existing systems for 64.4% in analytics accuracy and 59.2% in privacy leakage.

I. Introduction

Recently, video analytics systems have been widely developed to support such applications as traffic counting, surveillance, autonomous driving, Metaverse, etc. According to diverse application requirements, video analytics systems can be classified into edge-side video analytics systems [1], edge-cloud video analytics systems [2], pre-stored video analytics systems [3], etc. In this paper, we study edge-cloud video analytics systems, where the edge and the cloud cooperatively conduct model inference on the video frames collected by the camera of the edge, using the CNN model of the video analytics application, e.g., a YOLO model trained for pedestrian detection. The edge conducts initial analytics on the video frames to a *split layer* of the CNN model and sends the intermediate results of this split layer to the cloud for follow-up analytics. Edge-cloud video analytics systems allow workload offloading from the resource-constrained edge devices to the cloud. They have been widely deployed in the industry. For example, Microsoft Split-brain [4] supports traffic counting using a CNN model MobileNet [5]. The Microsoft Azure Stack Edge conducts analytics of the initial layers of the MobileNet model, and the rest layers are analyzed in the Microsoft Azure Cloud Server.

In an edge-cloud video analytics system, the intermediate results may be hijacked during unreliable edge-cloud communication. In this paper, we show that an attacker can perform reconstruction attack [6], a prominent attack on video analytics currently, on such intermediate results. A reconstruction attack can inverse the intermediate results back to their input status, e.g., through an adversarial neural network decoder [7]. As such, private information in the raw video frames, e.g., a plate of a car, can be leaked.

Establishing trusted execution environment (TEE) [8] or video frame encryption [9] is not viable in resource-constrained edge-cloud systems. There are systems that add noise to the intermediate results [10][11] or develop new privacy-preserving DNN models for video analytics. We will show that the former does not work well if the split layer is in the lower/initial layers of the DNN model and the latter has to give up well-established DNN models and lacks backward compatibility. Another straightforward approach is to directly protect private information, e.g., the plate of a car, human faces. Existing computer vision technologies have been developed to directly protect certain objects or attributes, by replacing [12], blurring [13], etc. However, it requires the set of sensitive objects to be priorly agreed [14]. In this paper, we focus on the privacy preserving of general video analytics applications. That is the whole video frame is considered sensitive. Our privacy is related to attacks, i.e., the video frame should not be leaked under reconstruction attacks.

In this paper, we propose to leverage image transformation technologies [15] to preprocess the video frames in the edge, prior to the edge-cloud video analytics process. Image transformation [9][16] have been developed for image augmentation [17], style transform [18], etc. For example, one can transform the human being in an image into a cartoon figure [19]. In the computer vision community, image transformation has also been used to transform sensitive information. Nevertheless, how to integrate such technologies into resource-constrained edge-cloud video analytics systems, and how to protect the split layer are challenging. Balancing the accuracy of the video analytics results and the privacy leakage in a split layer requires careful design.

In this paper, we develop Preva, a new privacy-preserving edge-cloud video analytics system that can take the comput-

ing and communication resources into consideration and can ensure high-accuracy video analytics as well as minimize the privacy leakage under reconstruction attacks. Preva made two important design choices:

First, we develop a new policy-based transformation scheme for video frames. In a policy-based transformation scheme, an image is transformed by a set of *policy*; a policy can be a change of the color distribution from black to white, or an inversion of the image polarization. We call the policy set in our system the *PrevaPolicy*. Policy-based transformation has much smaller resource requirements and becomes viable for edge-cloud systems. When a video frame is transformed by the PrevaPolicy; the subsequent edge-cloud video analytics can output intermediate results that can sustain reconstruction attacks. To generate PrevaPolicy, we design a new neural network model *PrevaNet*, and a new *adversary training method* to train the PrevaNet model. Finally, we develop a resource-aware algorithm that given PrevaNet, a video frame, and the resource constraints of an edge device, generates PrevaPolicy for this frame that, after the transformation of this frame, the final analytic results of this frame can maintain high accuracy and the privacy leakage at the split layer can be minimized.

Second, generating a PrevaPolicy for each video frame brings about non-trivial computing workloads. We observe that adjacent video frames are similar in the sense that we can generate one PrevaPolicy and reuse it for a number of adjacent frames. To this end, we develop a new algorithm to determine the frames that PrevaPolicy can be reused.

We formally analyze and show that Preva can guarantee privacy leakage under reconstruction attacks. We analyze two types of attackers: outsider attackers which can only access the intermediate results and insider attackers which can pretend to be an edge and thus access both the intermediate results and the PrevaNet model. We show that as the privacy leakage of the transformed video frame is minimized by PrevaPolicy, the privacy leakage of the original video frame is guaranteed.

We evaluate Preva on three representative video analytics applications, namely Face Classification [20], Video Objects classification [21] and Driver Behavior Recognition [22]. We compare three state-of-the-art video analytics systems and show that Preva significantly outperforms existing systems in both privacy leakage and analytics accuracy. Specifically, Preva can increase video analytics accuracy by 25.4%-64.4% and reduce privacy leakage by 10.3%-59.2%.

The contribution of the paper can be summarized as:

- We analyze the reconstruction attacks on edge-cloud video analytics systems and the limitation of existing schemes (Section II). We develop Preva, a new privacy-preserving edge-cloud video system that can provide high-accuracy video analytics and minimize privacy leakage (Section III).
- We develop a formal analysis of the privacy guarantee of Preva under reconstruction attacks of both outsider attackers and insider attackers (Section IV).
- We evaluate Preva with three video analytics applications, and Preva outperforms state-of-the-art systems in both analytics accuracy and privacy protection. (Section V).

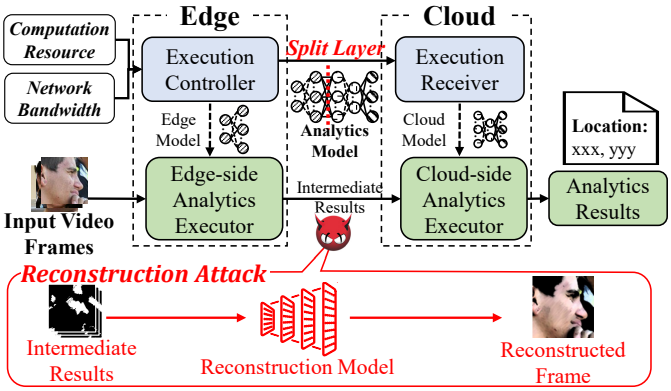


Fig. 1: Real-time edge-cloud video analytics system with reconstruction attack.

II. Background and Motivation

A. Background

A real-time edge-cloud video analytics system receives videos from an edge camera and conducts video analytics through a DNN model inference for real-time feedback, e.g., traffic counting. Edge devices are usually resource-constrained. Thus, an edge-cloud video analytics system offloads the video analytics workloads to the cloud. More specifically, both the edge and the cloud have the DNN model for inference. One such system is Microsoft Split-Brain [4] as shown in Fig. 1. It conducts vehicle counting on a 720p traffic camera in real-time, e.g., typically 30 frames per second. In this system, they first apply a communication-computation tradeoff algorithm to determine the *split layer* of the analytics model, according to the input mobile device computation resource and network bandwidth on the Execution Controller. The Execution Controller partitions the analytics model into edge and cloud models and sends them to Edge-side Analytics Executor and Cloud-side Executor, respectively. In Edge-side Analytics Executor, the input video frames are sent to the edge model for inference and output the intermediate results. The intermediate results are generally the feature maps sent to the cloud for cloud-side analytics.

Reconstruction Attack. With the broad deployment of video analytics applications in the edge-cloud video analytics system, reconstruction attacks emerge. Reconstruction attacks use a reconstruction model to reconstruct the original image. Specifically, a reconstruction attack can take an intermediate result of an NN model during inference as input and reconstructs/recovers the raw image. Fig. 1 shows an example where the image (i.e., a man's face) is recovered from intermediate results. Precisely, there are two operations of a reconstruction attack. The attacker first trains a reconstruction DNN model by minimizing the distance between the reconstructed and the raw image. Then, the attacker hijacks the intermediate result of an analytics model and takes it as input to reconstruct the raw image with the trained reconstruction DNN model. There are many reconstruction models, e.g., Linear-based [6], GAN-based [23], Likelihood Maximization [24], etc. Reconstruction

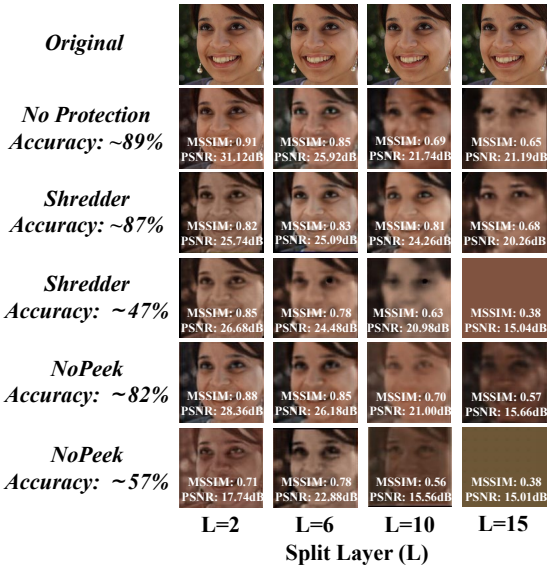


Fig. 2: Reconstructed images from model noise injection methods.

models have different levels of reconstruction capability, which is evaluated as the deviation between the reconstructed image and the original image. The deviation is measured by certain metrics, e.g., MSSIM [25], PSNR [26]. Many works have shown that a reconstruction attack can achieve high attack accuracy where the deviation metrics between the recovered image and the original image is up to 96% [6].

B. Motivation

We conduct a reconstruction attack on edge-cloud systems. The reconstruction model is a commonly used Auto-Encoder as in [6]. We study how likely an attacker can reconstruct a video frame from the intermediate results of the split layer. We study a vanilla edge-cloud system without a protection mechanism and state-of-the-art systems, Shredder [27], NoPeek [28]. Shredder applies a noise injection method. More specifically, Shredder trains a noise distribution for a target split layer; and adds such a noise layer into the CNN model for model inference; so that the CNN model can produce the intermediate results with the noise layer that can defend against reconstruction attacks.

We adopt the edge-cloud video analytics system setup as follows. An AWS DeepLens camera runs on the edge, and the CNN model VGG-19 [29] trained with the Fairface dataset [20] is applied in the system for video classification. Typical multi-scale structural similarity metrics MSSIM [25], PSNR [26] are utilized to measure the privacy leakage degree. We perform a reconstruction attack on the 2nd (lower layers), 6th, 10th, and 15th (higher layers) layer of the VGG-19 model.

The performance of the reconstruction attack is shown in Fig. 2. We see the vanilla edge-cloud system (No protection) fails to protect the privacy of the input image in all attacked layers, as the reconstructed images are at least 65% similar to the original image. We also see the privacy leakage computed by MSSIM of Shredder is 0.82, 0.83, 0.81, and 0.68 at

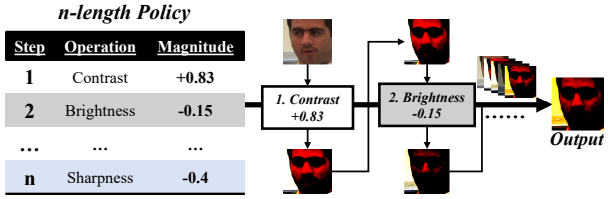


Fig. 3: Policy-based transformation.

the 2nd, 6th, 10th, and 15th layer, respectively, when the classification accuracy is about 87%. To achieve 0.67 average privacy leakage, the classification accuracy of Shredder drops to 47%. Similar results can be observed from NoPeek.

We see that Shredder is less effective in the lower layers of the CNN model. This has been observed and stated in [30][28]. Intrinsically, the information in lower layers is more original to the raw image; e.g., a lower layer contains face characters of the raw image, whereas the information in a higher layer has been embedded into analytics results, e.g., the recognition of gender. To this end, the reconstruction attack becomes much easier in lower layers.

C. Potential Approaches: Policy-based Image Transformation

In this paper, we study whether we can protect privacy in an arbitrary layer. Our idea is to conduct a transformation to the raw image and then conduct edge-cloud video analytics. Such an approach intrinsically transforms the privacy-sensitive features to privacy non-sensitive features, e.g., a human face to a cartoon face, and thus is layer independent.

Image Transformation has been extensively explored in the computer vision community. There are several image transformation approaches, such as pixel-level transformation [9], GAN-based transformation [16], and policy-based transformation [15]. The Pixel-level and GAN-based approaches train a transformation CNN model that will be used in the inference phase to transform a raw image into a transformed image that can protect privacy. The policy-based approach trains a model to generate a policy set that will be used in the inference phase to transform a raw image into a transformed image that can protect privacy.

The transformation CNN models of the pixel-level and GAN-based approaches are used in data centers for raw image transformation. They are complex. For example, the CycleGAN model in [31] is 454.6GFLOPs to process an image with 512x512 resolution.

The policy-based approach can be more practical to edge-cloud video analytics, as a policy set instead of a CNN model is used to transform the raw image into a transformed image in the inference phase.

Policy-based Image Transformation is a two-step transformation solution guided by transformation policies as shown in Fig. 3. Specifically, a transformation policy is a set of basic image operations with a certain length in order. The operations are supported by Pillow Image Library [32], including rotate, contrast, posterize, etc., with a magnitude value. There are two steps in Policy-based transformation. First, it predicts a certain length transformation policy by a neural network model based

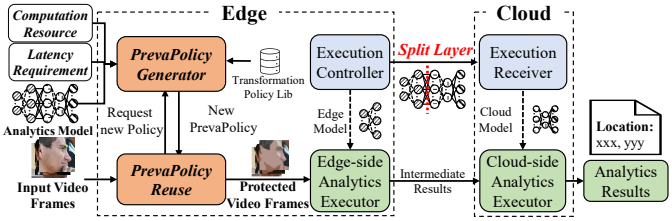


Fig. 4: The workflow of video analytics in Preva.

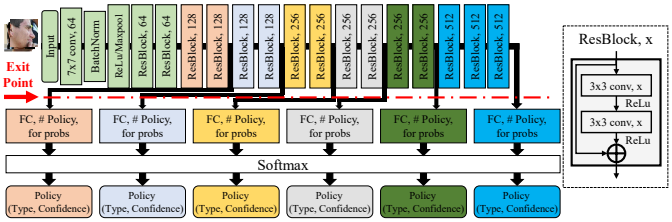


Fig. 5: The structure of PrevaNet.

on extracted features of input images. Second, it performs the image operations of the predicted transformation policy onto the input images step-by-step. Since the prediction of the transformation policy is a classification problem that selects an appropriate policy from finite image operations, it requires a much simpler neural network architecture than the direct transformation methods, i.e., Pixel-based and GAN-based methods. Furthermore, the computation cost of performing policy is extraordinarily light and ignorable.

III. Preva Design

A. Overview

We first present the challenges and a solution overview on integrating the policy-based image transformation into resource-constrained edge-cloud video analytics systems to achieve high analytics accuracy and minimize privacy leakage. There are two challenges:

- **Transformation of a video frame:** As said, we need a policy set, *PrevaPolicy*, to transform a video frame. We need PrevaPolicy to be appropriate in the sense that when it is used in frame transformation, its resource requirements can be supported by edge devices. To generate PrevaPolicy, we develop an adaptive CNN model structure (Section III-B) that the number of layers can be elastically changed when it is used in the PrevaPolicy generation. We call it the *PrevaNet* model. We develop *an adversarial model training method* to train PrevaNet. Finally, we develop a resource-aware algorithm, *PrevaPolicy Generation (PPGen)*, that given PrevaNet, a video frame, and the resource constraints of an edge device, generates PrevaPolicy for this frame that, when performing a transformation on this frame, high accuracy of the final analytic results can be maintained and privacy leakage can be minimized.
- **Transformation of a video stream:** Generating a PrevaPolicy for each frame in a video stream consumes significant computing resources. Since adjacent frames are similar, we can reuse PrevaPolicy. We develop an algorithm, *PrevaPolicy Reuse (PPReuse)* (Section III-C), to determine the frames in a video stream where the PrevaPolicy can be reused.

Video analytics in Preva: As shown in Fig 4, When Preva performs video analytics of a video stream, a frame will first be processed by PPReuse. If a new PrevaPolicy is needed, the frame will then be processed by PPGen to generate a PrevaPolicy for this frame. This frame will then be transformed by the PrevaPolicy. Finally, this frame will start edge-cloud video analytics, e.g., through a YOLOv3 [33] model for traffic counting of this frame. We will show that Preva can guarantee privacy (Section IV), and our evaluation shows that Preva achieves high accuracy in various resource settings (Section V).

Threat Model: We assume that the cloud is curious but honest, and the potential attacker (cloud/third parties) is computationally unbounded. For data privacy, the edge devices locally process the raw video frames with the edge analytics model before sending the intermediate result to the cloud for further processing. Before attacking, the attacker can access the video analytics model in advance, so that he can input its own images into this analytics model to obtain intermediate results, which can then be used to train a reconstruction model. During attacking, the attacker is assumed to be able to hijack the intermediate result of the target victim edge and recover the raw video frames from the intermediate result with the well-trained reconstruction DNN model mentioned above.

B. PrevaPolicy Generation for a Video Frame

There are two phases for PrevaPolicy generation. The first phase is *PrevaPolicy model training*, and a CNN model is designed and offline trained to output appropriate PrevaPolicy, which can not only achieve low privacy leakage and high analytics accuracy but also satisfy the latency requirement of the video analytics task. We call this model the PrevaNet Model. The second stage is *PrevaPolicy model serving*, given the edge-side resource configurations, apply PrevaNet on a video frame to generate a PrevaPolicy specifically for this frame. In the following, we first introduce the structure design of the PrevaNet model, then present its adversarial training process, and finally demonstrate the workflow of the PrevaNet model serving.

1) PrevaNet structure design: The transformation policy generator is intrinsically a multi-class classifier that can choose a proper PrevaPolicy for the input video frame. Since the generator executes in edges with heterogeneous and constrained computation resources, we need to design a classification CNN model that can dynamically adjust the number of executed layers of the generator in the inference phase to achieve the latency requirement.

There are several adaptive DNN models in literature designed for resource-constrained scenarios. For example, the iBranchy model [34], the FlexDNN model [35], and the BranchyNet model [36]. We choose to use the BranchyNet model as our base structure because it enables selective exe-

cution of DNN via proper early-exit control. To dynamically fit the heterogeneous computation resource of different edges, we develop the PrevaNet model by revising the BranchyNet model with more fine-grained exit points.

The architecture of the PrevaNet is shown in Fig. 5. The first three layers of PrevaNet are a convolution layer, a batch norm layer, and a max-pooling layer. Then, a sequence of 15 ResBlocks is followed. Each ResBlock contains two convolution layers and a ReLu activation layer. Six exit points are added to the output of the 4th, 6th, 8th, 10th, 12th, and 15th ResBlock, respectively. Each exit point consists of 2 full-connected layers and outputs the logits of class probabilities with the softmax activation layer. With this architecture, we can achieve elastic PrevaPolicy generation, where edges with fewer computation resources can finish the PrevaPolicy generation at the front exit point, and edges with more computation resources can generate the PrevaPolicy with more layers and complete the generation at the last exit point.

2) PrevaNet adversarial training: We develop an adversarial training method to train PrevaNet in the cloud. The goal of the PrevaNet is to search for the optimal policy from the *transformation policy set* for each input frame. Such a policy should satisfy two requirements: i) to guarantee analytics accuracy that the transformed video frame should maintain similar analytics accuracy as the original video frame, ii) to minimize privacy leakage that the attacker is not able to reconstruct the original video frame from the intermediate result of the transformed video frame.

Analytics accuracy guarantee: We first define *accuracy score* to measure whether the candidate transformation policy can preserve the analytics accuracy. We expect to have an efficient and accurate criterion to judge the accuracy impact of each transformation policy on the analytics model. Inspired by the technique proposed in [15] that can evaluate the correlations between the local linear map and the neural network performance without training, we adopt this technique to calculate the accuracy score of the transformation policy. Specifically, we prepare a mini-batch of data samples $\{x\}_{i=1}^{N}$ and transform it to $\{\hat{x}\}_{i=1}^{N}$ with the candidate transformation policy T. We first calculate the Gradient Jacobian matrix as below:

$$J = \left(\frac{\partial f}{\partial \hat{x}_1}, \quad \frac{\partial f}{\partial \hat{x}_2}, \quad \cdots \quad \frac{\partial f}{\partial \hat{x}_N} \right)^{\top}. \tag{1}$$

Then we compute its correlation matrix:

$$(\Sigma_J)_{i,j} = \frac{(C_J)_{i,j}}{\sqrt{(C_J)_{i,i} \cdot (C_J)_{j,j}}}, \tag{2}$$

where $C_J = (J - \frac{1}{N}\sum_{n=1}^{N} J_{i,n})(J - \frac{1}{N}\sum_{n=1}^{N} J_{i,n})^{\top}$. Let $\sigma_{J,1} \leq \cdots \leq \sigma_{J,N}$ be the N eigenvalues of Σ_J. Then our accuracy score of transformation policy T is given by

$$S_{acc}(T) = \frac{1}{N}\sum_{i=0}^{N-1} \log\left(\sigma_{J,i} + \epsilon\right) + \left(\sigma_{J,i} + \epsilon\right)^{-1}, \tag{3}$$

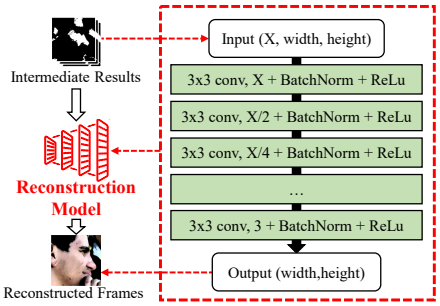


Fig. 6: The architecture of the adversary reconstructor.

where ϵ is set as 10^{-5} for numerical stability. With this accuracy score, we can quickly filter out policies that incur unacceptable performance degradation to the analytics model.

Transformation Policy Set. We consider the transformation policy library adopted by AutoAugment [17] which contains 50 various image transformation functions, including rotation, shift, inversion, contrast, posterization, etc. Certain transformation policies introduce large-scale perturbations to the input video frame, which can impair analytics accuracy. For example, when the analytics task is objection detection, the model can get accurate results when applying a rotation transformation policy on the input image. To satisfy the first requirement of PrevaPolicy, we search the transformation policy library and filter out unacceptable policies which significantly impair the analytics accuracy. Specifically, we calculate the accuracy score of each policy with Equation 3, and the policy is removed from the set if the value is less than the threshold γ; otherwise, the policy remains in the set. Therefore, we can obtain a transformation policy set in which all policies can guarantee the accuracy of the analytics task.

Privacy leakage minimization: Our goal is to choose the PrevaPolicy that can minimize the accuracy of the adversary reconstruction model. The attacker can apply any neural network architecture in the adversary reconstruction model design. We adopt the most powerful reconstructor [6] as the adversary reconstructor, and the architecture is shown in Fig. 6 which is composed of several convolution layers. The adversary reconstruction model is trained to optimize the quality of the reconstructed video frame $\hat{x}$ as close as the original video frame x.We leverage Multi-scale Structural Similarity (MSSIM) [25] to evaluate the performance of the reconstruction model, which can be expressed as:

$$L_{Rec} = 1 - MSSIM(\hat{x}, x). \tag{4}$$

The MSSIM value ranges between 0 and 1. The higher the MSSIM value is, the better quality of the reconstructed video frame. Consequently, the attacker aims to solve the optimization problem below:

$$\theta_{Rec} = \underset{\theta_{Rec}}{argmin}\, L_{Rec}, \tag{5}$$

where θ_{Rec} is the parameter of the adversary reconstruction model. On the contrary, to defend against the reconstruction attack, the PrevaNet should output the PrevaPolicy that can minimize the privacy leakage of the transformed video frame

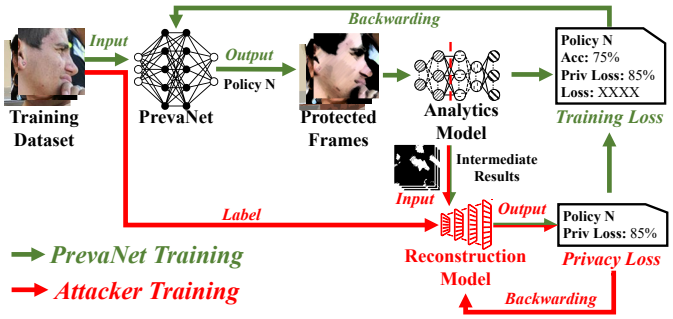


Fig. 7: The adversarial training workflow of PrevaNet on the cloud server.

x^T. The adversary reconstruction model is trained to make each reconstructed frame $\hat{x}$ similar to the transformed frame x^T but different from the original frame x. Therefore, the loss function of PrevaNet can be expressed as below:

$$L_{Gen} = MSSIM(x, \hat{x}) - MSSIM(x^T, \hat{x}). \tag{6}$$

Since the intermediate result of each split layer is different, the attacker needs to train different reconstructors θ^k_{Rec} for each split layer k. Suppose there are M split layers, to minimize the privacy leakage of all the split layers, the PrevaNet can be trained as:

$$\theta_{Gen} = \underset{\theta_{Gen}}{argmin} \sum_{k=1}^{M} (MSSIM(x, \hat{x}^k) - MSSIM(x^T, \hat{x}^k)), \tag{7}$$

where θ_{Gen} is the parameter of the PrevaNet.

The PrevaNet is trained offline in the cloud with the public dataset. As shown in Fig. 7, the adversarial training includes two stages. In the first stage, we optimize the adversary reconstruction model by simulating an attacker, and the parameters of PrevaNet are fixed. In the second stage, we train the PrevaNet to defend against the reconstruction attacker while the parameters of the adversary reconstructor are unchanged. We iteratively performed these two stages until PrevaNet converged.

3) PrevaNet model serving: The trained PrevaNet is distributed to the edge to serve for PrevaPolicy generation. Since the computation resources of each edge are constrained, we need to decide the exit point of PrevaNet to maximize the accuracy of the generated PrevaPolicy under the latency constraint of the edge. We first model the accuracy of the PrevaNet in serving and then introduce the latency constraint of the edge. Finally, we formulate the PrevaNet exit point selection problem and design an algorithm to solve it.

PrevaNet accuracy in serving. The accuracy of PrevaNet in serving is determined by the number of executed layers. In PrevaNet, each exit points correspond to a different number of executed layers, and thus leads to different accuracy of the output generated PrevaPolicy. Suppose the parameters of the trained PrevaNet is θ^*_{Gen}, and it has P exit points. For the exit point p, the accuracy of the output PrevaPolicy is defined as u_p. Let $\mathbf{w} = \{w_1, \ldots, w_P\}$ be the indicator of whether the exit point is selected, and the exit point p is chosen to output the PrevaPolicy when $w_p = 1$, otherwise $w_p = 0$. Since only one exit point can output the PrevaPolicy, we have $\sum_{p=1}^{P} w_p = 1$. Therefore, the PrevaNet accuracy U_{edge} in serving can be expressed as below:

$$U_{edge} = \sum_{p=1}^{P} w_p u_p. \tag{8}$$

PrevaNet latency constraint in edge. When serving on the edge, the execution of PrevaPolicy generation leads to the latency of PrevaNet. The execution time of PrevaPolicy generation is determined by the number of executed layers of the PrevaNet, which corresponds to the exit points of the PrevaNet. Suppose the computation resource of the edge is C GFLOPS, and the required computation resource for exit point p is c_p GFLOP. Thus, the execution time of PrevaNet for exit point p is $t_p = \frac{c_p}{C}$, and the PrevaNet latency T_{Gen} can be expressed as in below:

$$T_{Gen} = \sum_{p=1}^{P} w_p t_p. \tag{9}$$

Since the video analytics task has latency requirements, the PrevaNet latency should satisfy this requirement. Suppose the upper bound of the PrevaNet latency is T^{max}, we have

$$T_{Gen} \leq T^{max}. \tag{10}$$

Problem formulation. Given the PrevaNet model θ^*_{Gen}, video frame $\mathcal{I}$, and edge computation resource C, we need to determine exit point $\mathbf{w}$ to maximize the accuracy U_{edge} of PrevaNet in serving under the latency constraint (10). As such, our **PrevaNet exit point selection problem** is formulated as in below:

$$\begin{aligned} \max_{\mathbf{w}} \quad & U_{edge} \\ s.t. \quad & T_{Gen} \leq T^{max}, \\ & \sum_{p=1}^{P} w_p = 1, \\ & w_p \in \{0, 1\}, \quad \forall p = 1, \ldots, P. \end{aligned} \tag{11}$$

To solve this problem, we design a simple greedy algorithm as shown in Algorithm 1. The main idea is to search all the possible solutions and choose the solution which has the maximum PrevaNet accuracy. Since the searching space is relatively small, we can find the optimal solution in polynomial time. Once the exit point of PrevaNet in the edge is determined, we feed the video frame to the PrevaNet and obtain the PrevaPolicy.

C. Enhancing PrevaPolicy Generation for a Video Stream

PrevaPolicy generation for each frame consumes non-trivial computation resources. For a video stream, we attempt to further reduce the PrevaPolicy generation latency by reusing the generated PrevaPolicy. To achieve this, we develop an algorithm called PPReuse to determine the frames in a video stream where the PrevaPolicy can be reused.

Algorithm 1: Resource-aware PrevaPolicy Generation (PPGen)

Input: θ^*_{Gen}, $\mathcal{I}$, C, T^{max}.
Output: $\mathbf{w}$:output branch indicator of PrevaNet, O_{Preva}: PrevaPolicy of video frame $\mathcal{I}$.

$\mathbf{w} \leftarrow \{1, 0, 0, \ldots, 0\}, U_{edge} \leftarrow U_{edge}(\mathbf{w}, \theta^*_{Gen})$;
for $p = 1, 2, \ldots, P$ **do**
  $w_p = 1$;
  $U \leftarrow U_{edge}(\mathbf{w}, \theta^*_{Gen})$, $T \leftarrow T_{Gen}(\mathbf{w}, C)$;
  if $U_{edge} < U$ *and* $T <= T^{max}$ **then**
    $U_{edge} \leftarrow U, \mathbf{w} \leftarrow \mathbf{w}$;
  else
    $w_p = 0$;
$O_{Preva} \leftarrow$ ***PrevaNet***$(\mathcal{I}, \theta^*_{Gen}, \mathbf{w})$;
Output $\mathbf{w}, O_{Preva}$;

In a video stream, the adjacent video frames are similar. Intuitively, a generated PrevaPolicy can be reused until a large frame difference is detected. To achieve this, we need to detect each frame with a frame difference metric. A representative list of image features has been used by the CV community to detect frame differences, and these features can be grouped into *low-level* features and *high-level* features in terms of the amount of computation required for extraction [2]. Low-level features such as pixel, edge or area differences can be observed directly from raw images. In contrast, high-level features, such as Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF), require multiple computation steps on raw video values for more semantic information extraction. Since the computation resource of the edge is restricted, we adopt low-level features to measure the frame difference. Edge and Area features are two commonly used low-level features in computer vision. Suppose the images in time t and $t-1$ are $\mathcal{I}_t$ and $\mathcal{I}_{t-1}$,respectively. Edge feature captures frame differences $diff_E$ in the contours of objects in a frame and can be calculated as below:

$$\textit{diff}_E(\mathcal{I}_t, \mathcal{I}_{t-1}) = Edge(\mathcal{I}_t) - Edge(\mathcal{I}_{t-1}), \tag{12}$$

where $Edge$ is edge function. Area features capture frame differences $diff_A$ in areas, and can be calculated as below:

$$\textit{diff}_A(\mathcal{I}_t, \mathcal{I}_{t-1}) = Area(\mathcal{I}_t) - Area(\mathcal{I}_{t-1}), \tag{13}$$

where $Area$ is the area function. Both features have their own advantages. The edge feature shows an excellent response to moving objects, and even minor movement of objects can lead to a high difference value. Area feature performs better for detecting new arriving objects, and a new item entering the video frames signifies a new region of motion, resulting in a significant difference value.

We adopt edge and area features in Preva to detect frame changes. Specifically, we combine both features and define the frame changes $Diff$ as follows:

$$\textit{Diff}(\mathcal{I}_t, \mathcal{I}_{t-1}) = \alpha \cdot \textit{diff}_E(\mathcal{I}_t, \mathcal{I}_{t-1}) + \beta \cdot \textit{diff}_A(\mathcal{I}_t, \mathcal{I}_{t-1}), \tag{14}$$

where α and β are two scaling factors. With the defined frame difference metric, each frame first calculates the difference value with the latest frame. The PrevaPolicy is required to generate for the current frame if the difference value is greater than a threshold Γ. The threshold Γ is particularly predefined based on the finetuning adjusting during our experiments.

Algorithm 2: PrevaPolicy Reuse (PPReuse)

Input: Predefined threshold α, β, Γ. At time t, Input Raw Frame $\mathcal{I}_t$.
Output: Protected Frame $\mathcal{I}_o$

for $t = 1, 2, 3, ..., T$ **do**
  $\mathcal{I} \leftarrow downsampling(\mathcal{I}_t)$;
  $\mathcal{I} \leftarrow NTSC(\mathcal{I})$;
  Obtain Edge difference: $\textit{diff}_E = \textit{Edge\_diff}(\mathcal{I}, \mathcal{I}_{saved})$;
  Obtain Area difference: $\textit{diff}_A = \textit{Area\_diff}(\mathcal{I}, \mathcal{I}_{saved})$;
  if $\alpha \cdot \textit{diff}_E + \beta \cdot \textit{diff}_A > \Gamma$ **then**
    Request new PrevaPolicy: $P_t \leftarrow$***PrevaNet***$(\mathcal{I}_t)$;
    for *all operation* $p \in P_t$ **do**
      Apply operation $\mathcal{I}_t = p(\mathcal{I}_t)$;
    Update $\mathcal{I}_{saved} = \mathcal{I}, P_{saved} = P_t$;
  else
    for *all operation* $p \in P_{saved}$ **do**
      Apply operation $\mathcal{I}_t = p(\mathcal{I}_t)$;
  Output $\mathcal{I}_o = \mathcal{I}_t$;

The PPReuse algorithm is summarized in Algorithm 2. We first downsample the shape of the frame to 128 × 128 to reduce the execution time and transfer the RGB-color image into grayscale with the NTSC formula [37]. Then, we compute the frame difference with the saved frame. If the difference value exceeds the threshold, a new PrevaPolicy is required to generate and apply to the current frame. The current frame and the new PrevaPolicy are saved for processing the next frame. Otherwise, the current frame will be transformed with the saved PrevaPolicy.

IV. Privacy Analysis

We now analyze the privacy leakage of the Preva system. We first formally model the system, privacy leakage, and policy-based transformation. We then analyze two types of adversaries: (1) one that can only hijack intermediate results without knowledge of whether the edge device has conducted protection mechanisms, e.g., our transformation scheme; we call it an outsider attacker; and (2) one that not only hijacks intermediate results but also accesses PrevaNet. It is possible since the adversary can pretend to be an edge device and participate in the Preva system; thus, it can obtain the PrevaNet while trying to hijack the targeted victim edge device, and we call it an insider attacker.

A. Preva Modeling

***k*-split Video Analytics Neural Network Model**: Consider a neural network-based video analytics model $f(\theta^*)$ with trained parameters θ^* consists of m layers and $\theta^* = (\theta^{*(1)}, \theta^{*(2)}, \ldots, \theta^{*(m)})$. Suppose $f(\cdot)$ is a composition of m functions, i.e. $f = f_1 \circ f_2 \circ \ldots \circ f_m$, where, f_i represents the i-th layer and the corresponding parameter is $\theta^{*(i)}$. To reach the latency requirement of an edge-cloud video analytics system, the analytics model is split into two parts from k-th layer. We define the k-split video analytics model as $f = f^k_L \circ f^k_R$, where $f^k_L = f_1 \circ \ldots \circ f_k$ executes in the edge and $f^k_R = f_{k+1} \circ \ldots \circ f_m$ executes in the cloud, respectively, and the output of f^k_L is sent from the edge to the cloud for the following execution.

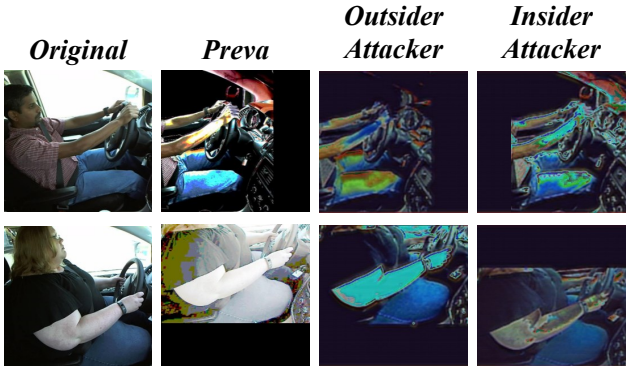


Fig. 8: Visual comparison of transformed and reconstructed frame examples by Preva, outsider and insider attackers.

Privacy Leakage: The privacy information of the edge input can be inferred through being reconstruction attacked. Given the output of f_L^k as $z = f_L^k(x)$, the attacker attempts to discover an input $\hat{x}$, such that the corresponding output of f_L^k is very close to z. This can be formulated as the following optimization problem:

$$\hat{x} = \underset{\hat{x}}{argmin} \, \|f_L^k(\hat{x}) - z\|, \tag{15}$$

where $\|\cdot\|$ is a norm for measuring the distance between the two outputs. We define the privacy leakage P_k of a k-split neural network as the similarity between the reconstructed input $\hat{x}$ and the original input x, that is,

$$P_k = MSSIM(\hat{x}, x) \tag{16}$$

where $MSSIM(\cdot)$ is the Multi-scale Structural Similarity metric for images. The higher the similarity, the higher the privacy leakage.

Policy-based Transformation: A policy T is composed by a set of n transformation functions $T = t_1 \circ t_2 \circ \ldots \circ t_n$. Each input x can be transformed to $x^T = T(x)$ with a transformation policy T. We intend to transform the input before feeding it into the analytics model, and the output of f_L^k is changed from $z = f_L^k(x)$ to $z^T = f_L^k(T(x))$. This leads to not only a change to the privacy leakage as $\hat{P}_k = SSIM(\hat{x}^T, x)$, where $\hat{x}^T$ is reconstructed based on z^T, but also a change to the accuracy of the analytics result as $\hat{L} = \mathcal{L}(f(\hat{x}^T))$, where $\mathcal{L}$ is the loss function of the analytics model.

B. Privacy Analysis

The privacy leakage mainly depends on the similarity between the reconstructed frame and the original frame, and the reconstructed frame is determined by the attacker model. We consider two types of attackers: the outsider attacker and the insider attacker. Specifically, we assume that both attackers have unlimited computing resources and can fully access the intermediate results of the target edge device. Moreover, the insider attacker can buy (or collude with) one edge and then own a trained PrevaNet, which can be leveraged in its attacks. We analyze the privacy guarantee of Preva with these two attackers in the following.

1) Outsider Attacker: The attacker reconstructs the input of the target edge with the intermediate results of this edge.

Analysis. In this attack, the attacker ignores the existence of the transformation policy generator and attempts to train a reconstruction model with the intermediate results of the target edge directly. Let θ_a be the reconstruction model of the attacker, and it can be derived by the following optimization problem:

$$\theta_a = \underset{\theta_a}{argmin} \, \|h(z^T, \theta_a) - x'\|, \tag{17}$$

where $x' \in D_{train}$ is the training sample, and $h(\cdot, \theta)$ is the reconstruction model. When the attacker is strong enough(i.e., with a strong neural architecture and enough training data), the reconstructed input $h(z^T, \theta_a)$ can approximate the transformed input $\hat{x}$ with high confidence, and the upper bound on the privacy leakage in this attack is in below:

$$\hat{P}_k \leq MSSIM(x^T, x). \tag{18}$$

This implies that the maximum privacy leakage is determined by the difference between the transformed and original frames. Since the transformation policy T is generated to minimize the privacy leakage of the original input, the attacker cannot infer the sensitive information from the transformed input and the privacy of the target edge is guaranteed.

2) Insider Attack: The attacker reconstructs the input of the target edge with the intermediate results of this edge and the specific PrevaNet of other colluded edges.

Analysis. In this attack, the attacker can leverage the specific PrevaNet model $g(\cdot, \theta_c)$ of the concluded edge and train a reconstruction model θ_a by minimizing the distance between the reconstructed input and original input with the following optimization problem:

$$\theta_a = \underset{\theta_a}{argmin} \, \|g^{-1}(h(z^T, \theta_a), \theta_c) - x'\|, \tag{19}$$

where $g^{-1}(\cdot, \theta_c)$ is the reverse of the PrevaNet in the concluded edge. Similarly, with enough computing resources and strong neural architecture, the attacker can reconstruct the original input transformed by $g(\cdot, \theta_c)$ with high confidence, and the upper bound on the privacy leakage in this attack is in below:

$$\hat{P}_k \leq MSSIM(g^{-1}(x, \theta_c), x). \tag{20}$$

This implies that the maximum privacy leakage is determined by the difference between the PrevaNet of the concluded edge and the target edge. Since the PrevaNet in serving varies as the heterogeneous resources of different edges, the attacker cannot infer the sensitive information of the input in the target victim edge with a different PrevaNet model and the privacy of the target edge is also guaranteed.

3) Experiments: We implement these two attackers to verify our analysis. The video analytics model is a MobileNetv3 [5] for the driver behavior recognition, and the test video frames are from the Statefarm [22], a Driver Behavior Recognition Video dataset. The outsider attacker is trained with a small number of input and intermediate result pairs obtained through the edge analytics model. The insider attacker is

TABLE I: The Application Specifications.

Application	*Facial Classification*	*Video Object Classification*	*Driver Behavior Recognition*
Analytics Model	ResNet50 [38]	VGG-19 [29]	MobileNetv3 [5]
Frames Resolution	224p	720p	1080p
Frame Size	4.33KB	28.5KB	42.5KB
Delay Requirement	100ms	40ms	33.33ms
Frame Rate	10FPS	25FPS	30FPS
Dataset	Fairface [20]	ILSVRC2015 [21]	StateFarm [22]

trained with the PrevaPolicy generated by the PrevaNet and the training pairs of the outsider attacker. Both attackers recover the raw input frames from the intermediate results of the 4th layer of MobileNetv3, and the results are shown in Fig. 8. The original column is the raw video frame, and the Preva column shows the transformed video frame based on the PrevaPolicy. Outsider attacker and insider attacker columns present the reconstructed video frames. We can see both attackers cannot successfully reconstruct the raw video frames, and the privacy of the raw video frames is well protected.

V. Evaluation

A. Experimental Setup

We evaluate Preva on an edge-cloud simulation environment we build. For edge devices, we use Amazon AWS DeepLens, a widely used smart camera alongside an Intel Atom CPU with 8GB memory running Ubuntu OS-16.04 LTS. For the cloud, we use a workstation server with an Nvidia RTX 3090 powerful GPU and an AMD Ryzen 9 5900X CPU running on Ubuntu 18.04 LTS. The edge devices and the cloud communicate through a wired network cable, and the bandwidth is controllable by a python script.

Applications and datasets. We use three representative video analytics applications to evaluate Preva: 1) *Facial Classification* (FC) which predicts the gender of humans from images of Fairface [20] dataset, 2) *Video Object Classification* (VOC) which predicts the categories of objects in ILSVRC2015 VID video dataset [21], and 3) *Driver Behavior Recognition* (DBR) which recognizes the behavior of driver from Statefarm [22] Driver Behavior recognition video dataset. Detailed specifications of each video analytics application are shown in Table I.

Baselines. We compare Preva with three state-of-the-art edge-cloud analytics systems, which are *open-source* and widely used as benchmarks. In addition, we also implement a no-protection setup named Vanilla to serve as the lower bound of privacy protection performance.

- ***Vanilla (No-Protection)***: It is a non-privacy-preserving edge-cloud video analytics system developed on Microsoft Rocket System with a greedy algorithm to select the split layer based on the bandwidth restriction.
- ***Deep Private-Feature Extraction (DPFE)*** [11] : The privacy protection mechanism of DPFE is to train a private-feature extractor by modifying the analytics model topology and re-training all the model parameters.
- ***Arden*** [10]: A lightweight privacy-preserving mechanism consisting of arbitrary data nullification and random noise addition is introduced in Arden to achieve differential privacy of edge-cloud video analytics system.
- ***Shredder*** [27]: It learns additive noise distributions that significantly reduce the information content of communicated data while maintaining the inference accuracy to protect the privacy of the edge-cloud video analytics system.

Evaluation Metrics. We evaluate the performance of Preva and baselines from three aspects: privacy leakage, analytics utility, and latency. To assist the measurement, we choose the following three metrics:

- ***Privacy Leakage Metric:*** Our system takes a certain reconstruction model as an input. Therefore, in our evaluation, we take the state-of-the-art GAN-based model GMI as our system input, as presented in [23]. It inverts the intermediate results to the input frame for each application. Multi-scale Structural Similarity (MSSIM) [25] is adopted to quantify the privacy leakage between the reconstructed frame and raw input frame. Specifically, it conducts multiple scales through multiple steps of sub-sampling. It has been shown to outperform Structural Similarity (SSIM) on the different subjective frames, especially video datasets [39]. The MSSIM between frame x and y is computed as follows.

$$MSSIM(x,y) = [L_m(x,y)]^{\alpha M} \cdot \prod_{j=1}^{M} [C_j(x,y)]^{\beta_j} \cdot [S_j(x,y)]^{\gamma_j} \quad (21)$$

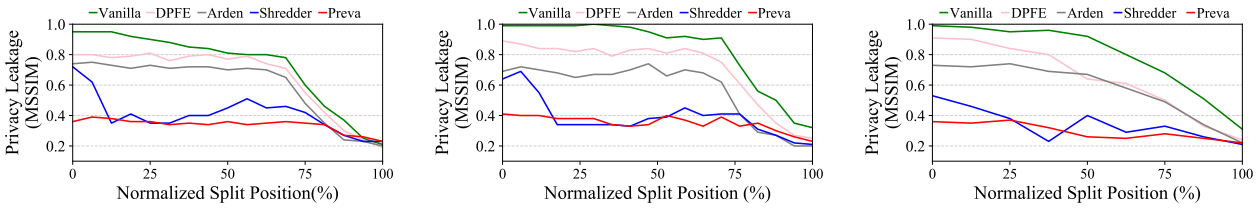
where M is the time of down-sampling resolution, i.e., $j = 1$ represents the original inputs. $L(x,y), C(x,y), S(x,y)$ are the Luminance, Contrast and Structure similarity between frame x and y, respectively.

We also adopt another common metric for deviation between frames, peak signal-to-noise ratio (PSNR) [26], to quantify the privacy leakage between the reconstructed frame and raw input frame. The PSNR between frame x and y is computed as follows.

$$PSNR(x,y) = 20 \cdot \log_{10}(MAX) - 10 \cdot \log_{10}(MSE) \quad (22)$$

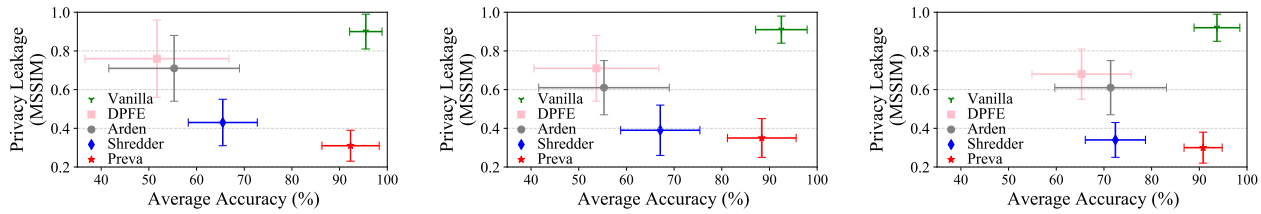
where MAX is the maximum possible pixel value of the frame, i.e., 255, and MSE, mean square error, measures the average numerical difference between pixels from x and y. The reconstructors can achieve an average MSSIM of about 0.94 to attack the vanilla analytics model.

- ***Analytics Utility Metric:*** We utilize different metrics to measure the utility of different analytics applications. For Application FC, the utility metric is the average accuracy of gender predictions from humans faces frames, computed as $\frac{g}{t}\%$, where s is the number of correct prediction gender and t is the total number of validation datasets. For Application VOC, we choose average top-5 accuracy as the utility metric, which means any of the analytics model's top-5 highest probability prediction match with the expected answer is considered as a correct prediction. We compute it as $\frac{l}{t}\%$, where s is the number of correct predictions and t is the total number of validation datasets. For Application DBR, the utility metric is considered as the proportion of correctly predicted categories among all possible drivers' behaviors, computed as $\frac{b}{t}\%$, where s is the number of correct predictions and t is the total number of drivers behaviors.
- ***Latency Miss Rate:*** Since our system takes bandwidth as an input factor, it is deployed on top of a network with bandwidth,

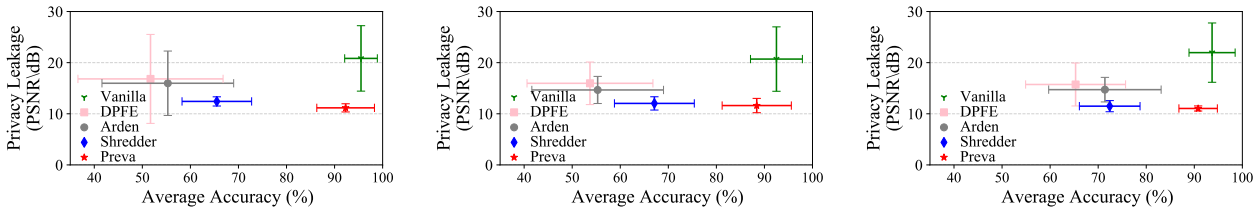


(a) Application FC (b) Application VOC (c) Application DBR

Fig. 9: Comparison of the privacy leakage in different privacy protection systems when the video analytics model of each application is split at different layers.



(a) Application FC measured by MSSIM (b) Application VOC measured by MSSIM (c) Application DBR measured by MSSIM



(d) Application FC measured by PSNR (e) Application VOC measured by PSNR (f) Application DBR measured by PSNR

Fig. 10: Comparison of the tradeoff between privacy and utility in different privacy protection systems.

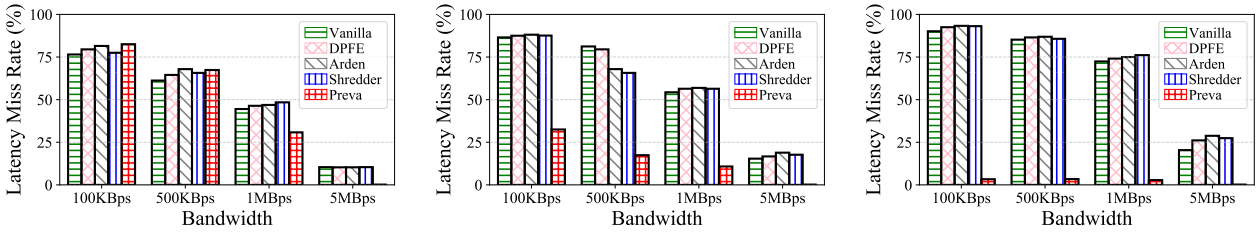
i.e., {100K,500K,1M,5M}Bps. In practice, however, a network has no hard guarantee on its bandwidth. The bandwidth can occasionally be smaller. Thus, latency miss rates (LMR) can occur. It is typical to use LMR as a metric to evaluate how significant the designed systems are. For example, ACCM-PEG [40], AWStream [41] using real-world bandwidth dataset with bandwidth dynamics. We quantify the latency miss rate by the percentage of the video analytics tasks that do not meet the latency requirement of a video analytics application with respect to the total video analytics tasks.

PreveNet Training specifications. We use Adam optimizer with an initial learning rate of 0.001 and exponential decay for PrevaNet training. The training-validation division rates in datasets are 85%-15%, 85%-15%, and 80%-20% for FC, VOC and DBR, respectively, and the training batch size is 512.

B. Overall Performance

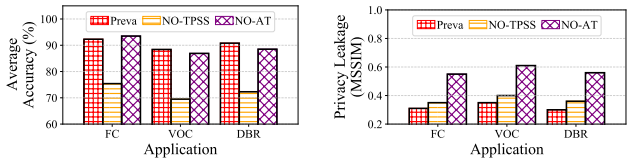
1) Improvement of Privacy Protection: For Preva and baselines, we evaluate the privacy leakage from different split positions of a video analytics model in each application. The experimental results are shown in Fig. 9. Since the number of layers in different analytics models is various, we normalize split positions to the relative position of an analytics model, i.e., split position 50% is the position of the middle layer of a model. For Vanilla, the privacy leakage is very high when split in the front 75% layers and naturally decreased when split in the last 25% layers. Therefore, we focus on the privacy leakage of different baselines when split in the front 75% layers of the analytics model. For Application FC, as shown in Fig. 9(a), we observe that Preva outperforms baselines, especially when split in the front 12.5% layers, and the privacy leakage is reduced by about 60.3%, 40.3%, 29.4%, and 26.2% compared to Vanilla, DPFE, Arden, and Shredder, respectively. Similar results can be found in the other two applications, as shown in Fig. 9(b) and Fig. 9(c). Remarkably, when split at the front 12.5% layers, Preva outperforms baselines, with the privacy leakage reduced up to 11.03% in application VOC and up to 32% in application DBR. Therefore, Preva improves the privacy protection performance against the reconstruction attack in all split layers. The underlying reason is that Preva transforms the input frames before performing the video analytics task, making it difficult for the attacker to reverse the intermediate result of the transformed frames to the original raw frames.

2) Improvement of Privacy and Utility Tradeoff: The overall analytics accuracy and privacy leakage in different applications under different protection systems are shown in Fig. 10. The privacy leakage is counted on average when the analytics model is split at all layers. For application FC as shown in Fig. 10(a), the average accuracy of Vanilla is about 95.5% with 0.9 MSSIM privacy leakage. DPFE and Arden achieve 51.7% and 55.3% accuracy with 0.76 and 0.71 MSSIM, respectively. Shredder achieves a lower privacy leakage with about 0.43



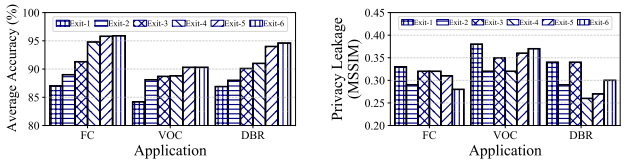
(a) Application FC (b) Application VOC (c) Application DBR

Fig. 11: Comparison of the latency miss rate in different privacy protection systems under various bandwidth conditions.



(a) Accuracy (b) Privacy

Fig. 12: The component analysis for Adversarial Training (AT) and Transformation Policy Set Selection (TPPS).



(a) Accuracy (b) Privacy

Fig. 13: The component analysis for PrevaNet Exit Point Selection.

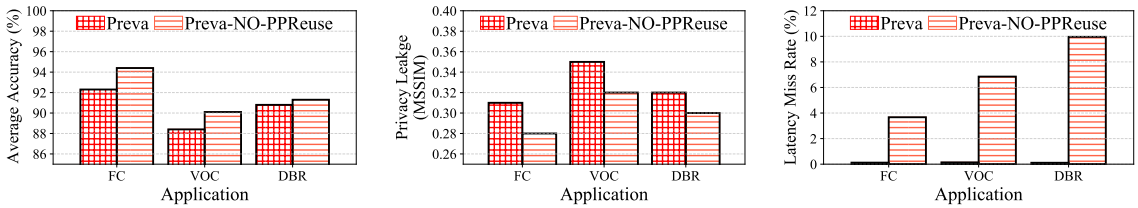
MSSIM, but it suffers a high degrading accuracy of about 65.5%. Preva outperforms DPFE, Arden, and Shredder, with the average analytics accuracy improved 43.9%, 40% and 29.0%, respectively, and the privacy leakage reduced 59.2%, 56.3%, and 27.9%, respectively. Compared to Vanilla, Preva sacrifices 3.4% of average accuracy but reduces 65.6% of privacy leakage. Similar results also shown in Fig. 10(b) and Fig. 10(c) for application VOC and DBR. Specifically, Preva outperforms baselines with an improvement of average analytics accuracy up to 64.4% in VOC and 39.1% in DBR, and with a reduction of privacy leakage up to 50.7% in VOC and 55.9% in DBR. We also measure the privacy leakage by PSNR, and observe similar privacy leakage reduction and analytics accuracy improvement to that by MSSIM as shown in Fig. 10(d)-(f). Specifically, Preva outperforms baselines with a reduction of privacy leakage up to 46.42% in FC, 43.9% in VOC and 49.7% in DBR, respectively. All the previous results prove that Preva not only has better privacy protection performance but also guarantees analytics accuracy in an acceptable range. The main reason is that Preva transforms the input frame with PrevaPolicy, which minimizes privacy leakage in all split layers and guarantees analytics accuracy.

3) Improvement of Latency Miss Rate: Since latency is an essential requirement for edge-cloud video analytics systems, we measure the latency miss rate of a video stream in Preva and baseline privacy protection systems. This evaluation is performed under 4 different network bandwidths, ranging from 100KBps to 5MBps, and the results are shown in Fig. 11. When the maximum bandwidth is 100KBps, for application FC, all the privacy protection systems, including Preva, have highly similarly latency miss rates at about 76.5%-82.5%. However, Preva outperforms the baseline systems in application VOC and DBR, and the latency miss rate is reduced by about 63.1% and 96.6%, respectively. A similar result is observed when the bandwidth is 500KBps bandwidth. The underlying reason is that the dataset of application FC is an image dataset, and the similarity of adjacent frames is low; thus, Preva can not reuse the generated PrevaPolicy to improve the efficiency, while application VOC and DBR analyze a video stream dataset and the adjacent frames can reuse the generated PrevaPolicy to reduce the latency miss rate. When the bandwidth is increased more than 1MBps, Preva outperforms all baselines in these three applications, with a reduction of latency miss rate in about 36.4%, 36.4%, and 81.1% for application FC, VOC, and DBR, respectively. Particularly, when the bandwidth reached 5MBps, Preva outperforms all systems with a latency miss rate reduction up to 95% in all applications. These results mainly benefit from the PrevaNet design that the privacy protection of all split layers is guaranteed, and Preva can flexibly choose the split layer to reduce latency based on the edge resources.

C. Component Analysis Study

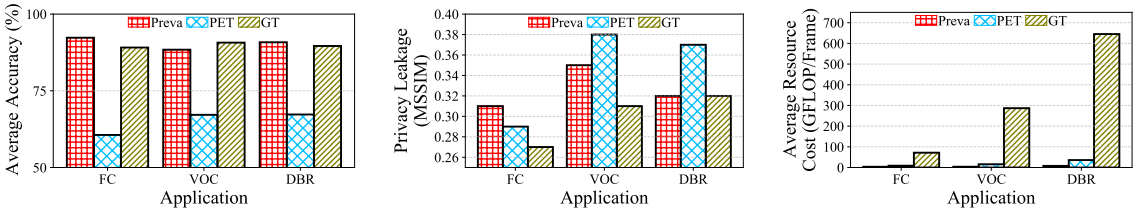
In this section, we explore the impact of the internal components of Preva for a better understanding of their contributions.

1) Impact of PrevaNet Training: There are two components to obtain the PrevaNet: Transformation Policy Set Selection (TPPS) and Adversarial Training design (AT). To take a closer look at the contribution of each component, we first implemented two breakdown versions of Preva: 1)**NO-TPPS** is a sub-version of Preva without selecting the transformation policies by accuracy score defined in Equation (3), and it randomly selects several policies from the transformation policy library [17]. 2)**NO-AT** is a sub-version of Preva without adversarial training. The adversary reconstruction model is static and trained only based on the intermediate results pairs obtained from Vanilla. Fig. 12 shows the results of average analytics accuracy and privacy leakage with different protection systems. For the NO-TPPS system, the analytics accuracy decreases by about 21.3%, and the privacy leakage increases by at least 12.9% compared to Preva. It leads to the fact that TPPS can significantly improve the analytics accuracy of



(a) Accuracy (b) Privacy (c) Latency Miss Rate

Fig. 14: The component analysis for PPReuse.



(a) Accuracy (b) Privacy (c) Resource Consumption

Fig. 15: The component analysis for different image transformation approaches.

Preva while reducing privacy leakage. The underlying reason is that the TPPS component filters out the policies that can vastly reduce the analytics accuracy. For NO-AT system, it presents a similar analytics accuracy and significantly higher privacy leakage compared to Preva. Specifically, in application FC, both NO-AT and Preva achieve about 91.7% analytics accuracy, while the privacy leakage of NO-AT increased by at least 66.7% compared to Preva. Therefore, Adversarial Training plays a vital role in the privacy protection of Preva.

2) Impact of PrevaNet Exit Point Selection: We design an experiment to investigate the influence of exiting PrevaNet at different exit points. For example, as shown in Fig. 5, there are 6 available exit points. We compare the average analytics accuracy and privacy leakage MSSIM on applications FC, VOC and DBR. It can be seen from Fig. 13(a) that, in FC, the accuracy is slightly improving from 85.4%-95.9% with the increasing execution layers of analytics models. The privacy leakage is floating in a small range as the findings in Fig. 13(b) from 0.28-0.33. We also observe similar results in VOC and DBR. These observations reveal that the reduction of execution layers will degrade the performance of the PrevaNet, but the influence is very small and can be accepted.

3) Impact of PrevaPolicy Reusing: We implement Preva sub-versions without the PPReuse algorithm (**Preva-NO-PPReuse**) to see how the PPReuse algorithm improves the performance of Preva in a video stream. The evaluation is deployed in the aforementioned three applications, and the results are shown in Fig. 14. For application FC, Preva-NO-PPReuse achieves similar analytics accuracy and privacy leakage compared to Preva. However, it significantly increases the latency miss rate by nearly 37 times. Similar results can be found in VOC and DBR. Particularly, in application DBR, the latency miss rate of Preva-NO-PPReuse is about 90 times more than Preva. These results indicate that the PPReuse algorithm can vastly reduce latency miss rate at the cost of acceptable privacy leakage, especially in the scenario that a sequence of adjacent frames is similar in a video stream.

4) Impact of Image Transformation Approaches: To better illustrate the choices of image transformation approaches, we introduce an experiment to prove that the policy-based transformation is the most appropriate for privacy-preserving in Preva. We implemented two state-of-the-art transformation approaches, including 1) Pixel-based Transformation (**PET**) [9], and 2) GAN-based Transformation (**GT**) [16] based on CycleGAN [31]. Fig. 15(a-c) shows the average analytics accuracy, privacy leakage and average resource cost of different image transformation approaches in Application FC, VOC and DBR, where the average resource cost of each frame is calculated as the GFLOPs/Frame. Specifically, for analytics accuracy, Preva outperforms PET by up to 0.52 times and achieves a similar value to GT. For privacy leakage, GT performs the best, and Preva is a little more than GT, with about 0.13 times. However, the resource cost of GT is significantly more than Preva, which increased by 19 $\sim$ 81 times, and it indicates that GT is not applicable in resource-constraint edges. Taking all this into consideration, the policy-based transformation approach is the most appropriate choice, which can provide low privacy leakage and guarantee analytics accuracy when performed in resource-stringent edge devices.

VI. Related Work

Our study falls into the category of privacy protection in edge-cloud video analytics systems using computer vision technologies. We present related work in edge-cloud video analytics systems, computer vision mechanisms, and recent privacy-protection video analytics systems.

Edge-cloud video analytic systems are one type of video analytics system. The edge is the video source, and the edge will conduct (part of) video analytics for real-time response or for privacy protection. With stringent resource constraints, the studies on edge-cloud video analytics systems emphasized the schemes to accelerate system performance while maintaining

high accuracy. There are hardware acceleration schemes, by boosting edge-side GPU [42], using FPGA [43], applying new wireless network controller [44], etc. There are algorithm studies on splitting the DNN model inference between the edge and the cloud [45][46]. There are systems developed to adapt to dynamics in network throughput [47], video contents [48][49], etc. Edge-cloud systems have been developed in industry, integrating research algorithms and technologies, e.g., Microsoft Split-brain [4], Amazon Kinesis [50], Huawei [51], etc.

Preva takes the computing and network resource constraints into the video frame transformation development. Preva emphasizes protecting the reconstruction attacks at the split layer, the key vulnerability of the edge-cloud video analytics system. Preva can integrate advanced performance acceleration methods developed in the edge-cloud video analytic systems.

Computer vision mechanisms in image privacy protection have been developed with attempts to modify or remove the sensitive information in an image. Existing mechanisms can be classified according to the way in which an image is modified [52]: (1) filtering, e.g., blurring or pixelating [53], (2) encryption, e.g., Advanced Encryption Standard (AES) has been used [54][55], (3) face de-identification, e.g., alter a face region through replacing real faces with synthetic ones [56], (4) object removal, e.g., by removing the interest objects and reconstructing missing parts to create a seamless image, and (5) object replacement, e.g., a stick figure or a silhouette, to replace the protected object in an image. Many technologies have been developed to support these mechanisms, such as image transformation, blurring, inpainting [57], etc.

This study emphasizes how to enhance edge-cloud video analytics systems for privacy protection. We leverage computer vision technologies, yet we ensure viable systems where the technologies can be integrated under system constraints.

Privacy-preserving video analytics systems has attracted increasing attention. According to where the protection mechanisms are performed, we can classify existing works into privacy-preserving cloud video analytics systems and edge-related video analytics systems. Cloud video analytics systems answer queries on analytics results. The attacks are membership inference attack [58], attribute attack [59], etc., which are less likely reconstruction attacks for raw video frames. Cloud system has abundant computing resources and thus can afford computation-intensive solution approaches. For example, Visor [8] designs a secure framework with hybrid TEE. PECAM [39] applies generative adversarial networks (cycleGAN) to hide sensitive information.

Our study falls into the category of privacy-preserving edge-cloud video analytics systems. Here, the edge is the data owner, and the cloud only assists video analytics related computation. Reconstruct attacks are the common attacks in this scenario. One approaches is to inject noise [10][11]to the intermediate results, e.g., Shredder [27] and NoPeek [28]. As discussed, noise injection methods may be less effective in the lower layers of the DNN model. Another approach is to redevelop/retrain the video analytics DNN model into a new privacy-preserving DNN model, e.g., DeepObfuscator [60]. These methods require the edge-cloud video analytics systems to be updated with new models, which may bring about backward compatibility problems. Preva complements these approaches with privacy-preserving image transformation prior to edge-cloud video analytics.

VII. Conclusion

This paper presented Preva, a new privacy-preserving edge-cloud video analytics system. Preva can protect the privacy of the model inference of edge-cloud video analytics systems from reconstruction attacks, a common attack that hijacks the communications of the intermediate results between the edge and the cloud and reconstructs sensitive raw video frames. Existing privacy-preserving methods are limited in the scenarios where the lower layers of the DNN model are difficult to protect (e.g., adding noise methods), or they require the edge-cloud video analytics systems to update their DNN models, which brings about backward compatibility issues (e.g., methods redeveloping new privacy-preserving DNN models). Intrinsically, Preva adds carefully designed video frame transformation to transform frames prior to edge-cloud video analytics. Preva can thus easily work with existing DNN models, and Preva can protect any intermediate results of split layers of the DNN model from both outsider and insider attackers. Preva is designed to be resource-efficient for resource-constrained edge devices, and it maintains high analytics accuracy with minimized privacy leakage.

Acknowledgement

Dan Wang's work is supported by GRF 15210119, 15209220, 15200321, 15201322, ITF-ITSP ITS/070/19FP, CRF C5018-20G.

References

[1] X. Zeng, B. Fang, H. Shen, and M. Zhang, "Distream: scaling live video analytics with workload-adaptive distributed edge intelligence," in *Proc. of ACM SenSys'20*, Virtual Event, Nov. 2020.

[2] Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. H. Xu, and R. Netravali, "Reducto: On-camera filtering for resource-efficient real-time video analytics," in *Proc. of ACM SIGCOMM'20*, Virtual Event, Aug. 2020.

[3] T. Xu, L. M. Botelho, and F. X. Lin, "Vstore: A data store for analytics on large videos," in *Proc. of EuroSys'19*, Dresden, Germany, Mar. 2019.

[4] J. Emmons, S. Fouladi, G. Ananthanarayanan, S. Venkataraman, S. Savarese, and K. Winstein, "Cracking open the dnn black-box: Video analytics with dnns across the camera-cloud boundary," in *Proc. of HotEdgeVideo'19 Workshop*, 2019.

[5] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proc. of IEEE/CVF ICCV'19*, Seol, Korea, Oct. 2019.

[6] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proc. of IEEE/CVF CVPR'16*, Las Vegas, NV, USA, June 2016.

[7] Y. Güçlütürk, U. Güçlü, K. Seeliger *et al.*, "Reconstructing perceived faces from brain activations with deep adversarial neural decoding," in *Proc. of NeurIPS'17*, Long Beach, CA, USA, Dec. 2017.

[8] R. Poddar, G. Ananthanarayanan, S. Setty, S. Volos, and R. A. Popa, "Visor: Privacy-preserving video analytics as a cloud service," in *Proc. of USENIX Security'20*, Virtual Event, Aug. 2020.

[9] W. Sirichotedumrong, T. Maekawa, Y. Kinoshita, and H. Kiya, "Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain," in *Proc. of IEEE ICIP'19*, Taipei, Taiwan, Mar. 2019.

[10] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not just privacy: Improving performance of private deep learning in mobile cloud," in *Proc. of ACM SIGKDD'18*, London, UK, Aug. 2018.

[11] S. A. Osia, A. Taheri, A. S. Shamsabadi, K. Katevas, H. Haddadi, and H. R. Rabiee, "Deep private-feature extraction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 54–66, 2018.

[12] Q. Sun, A. Tewari, W. Xu, M. Fritz, C. Theobalt, and B. Schiele, "A hybrid model for identity obfuscation by face replacement," in *Proc. of CVF'18*, Munich, Germany, Sept. 2018.

[13] N. Vishwamitra, B. Knijnenburg, H. Hu, Y. P. Kelly Caine *et al.*, "Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images," in *Proc. of IEEE/CFV CVPR'17 workshops*, Honolulu, HI, USA, July 2017.

[14] T. Orekondy, M. Fritz, and B. Schiele, "Connecting pixels to privacy and utility: Automatic redaction of private information in images," in *Proc. of IEEE/CFV CVPR'18*, Salt Lake City,UT,USA, June 2018.

[15] W. Gao, S. Guo, T. Zhang, H. Qiu, Y. Wen, and Y. Liu, "Privacy-preserving collaborative learning with automatic transformation search," in *Proc. of IEEE/CVF CVPR'21*, Virtual Event, June 2021.

[16] W. Sirichotedumrong and H. Kiya, "A gan-based image transformation scheme for privacy-preserving deep neural networks," in *Proc. of EUSIPCO'21*, Virtual Event, May 2021.

[17] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proc. of IEEE/CVF CVPR'19*, Long Beach, CA, June 2019.

[18] D. Kotovenko, A. Sanakoyeu, P. Ma, S. Lang, and B. Ommer, "A content transformation block for image style transfer," in *Proc. of IEEE/CVF CVPR'19*, Long Beach, CA, USA, June 2019.

[19] R. Hasan, P. Shaffer, D. Crandall, E. T. Apu Kapadia *et al.*, "Cartooning for enhanced privacy in lifelogging and streaming videos," in *Proc. of IEEE/CFV CVPR'17 workshop*, Honolulu, HI, USA, July. 2017.

[20] K. Kärkkäinen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age," *arXiv:1908.04913*, 2019.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[22] Kaggle, "StateFarm Distracted Driver Detection," 2019. [Online]. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection-video.

[23] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proc. of IEEE/CFV CVPR'20*, Virtual Event, June 2020.

[24] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City,UT,USA, June 2018.

[25] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. of IEEE ACSSC'03*, Pacific Grove, CA, USA, Nov 2003.

[26] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *Proc. of IEEE ICPR'10*, Istanbul, Turkey, Aug. 2010.

[27] F. Mireshghallah, M. Taram, P. Ramrakhyani *et al.*, "Shredder: Learning noise distributions to protect inference privacy," in *Proc. of ACM ASPLOS'20*, Lausanne, Switzerland, Mar. 2020.

[28] P. Vepakomma, A. Singh, E. Zhang, O. Gupta, and R. Raskar, "Nopeek-infer: Preventing face reconstruction attacks in distributed inference after on-premise training," in *Proc. of IEEE FG'21*, Virtual Event, Dec. 2021.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[30] C. Shi, L. Chen, C. Shen, L. Song, and J. Xu, "Privacy-aware edge computing based on adaptive dnn partitioning," in *Proc. of IEEE GLOBECOM'19*, Waikoloa, HI, USA, Dec. 2019.

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of IEEE ICCV'17*, Venice, Italy, Oct. 2017.

[32] "Pillow Image Module." [Online]. Available: https://pillow.readthedocs.io/en/stable/reference/Image.html

[33] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[34] S. K. Nukavarapu, M. Ayyat, and T. Nadeem, "ibranchy: An accelerated edge inference platform for lot devices," in *Proc. of ACM/IEEE SEC'21*, San Jose, CA,, Dec. 2021.

[35] B. Fang, X. Zeng, F. Zhang, H. Xu, and M. Zhang, "Flexdnn: Input-adaptive on-device deep learning for efficient mobile vision," in *Proc. of ACM/IEEE SEC'20*, Virtual Event, Nov. 2020.

[36] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *Proc. of IEEE ICPR'16*, Cancun,Mexico, Dec. 2016.

[37] J. Lu and K. N. Plataniotis, "On conversion from color to gray-scale images for face detection," in *Proc. of IEEE/CVF CVPR'09 Workshops*, Miami, FL, June 2009.

[38] K. He, X. Zhang *et al.*, "Deep residual learning for image recognition," in *Proc. of IEEE/CVF CVPR'16*, Las Vegas, NV, USA, June 2016.

[39] H. Wu, X. Tian, M. Li, Y. Liu *et al.*, "Pecam: privacy-enhanced video streaming and analytics via securely-reversible transformation," in *Proc. of ACM MobiCom'21*, New Orleans, US, Mar. 2021.

[40] K. Du, Q. Zhang, A. Arapin, H. Wang, Z. Xia, and J. Jiang, "Accmpeg: Optimizing video encoding for accurate video analytics," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 450–466, 2022.

[41] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. A. Lee, "Awstream: Adaptive wide-area streaming analytics," in *Proc. of ACM SIGCOMM'18*, Budapest Hungary, Aug. 2018.

[42] S. Cass, "Taking ai to the edge: Google's tpu now comes in a maker-friendly package," *IEEE Spectrum*, vol. 56, no. 5, pp. 16–17, 2019.

[43] S. Wang, C. Zhang, Y. Shu, and Y. Liu, "Live video analytics with fpga-based smart cameras," in *Proc. of HotEdgeVideo'19 Workshop*, 2019.

[44] Z. Li, Y. Shu, G. Ananthanarayanan, L. Shangguan, K. Jamieson, and P. Bahl, "Spider: A multi-hop millimeter-wave network for live video analytics," in *Proc. of ACM/IEEE SEC'21*, San Jose, CA,, Dec. 2021.

[45] Z. Xiao, Z. Xia, H. Zheng, B. Y. Zhao, and J. Jiang, "Towards performance clarity of edge video analytics," in *Proc. of ACM/IEEE SEC'21*, San Jose, CA,, Dec. 2021.

[46] C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu *et al.*, "Videoedge: Processing camera streams using hierarchical clusters," in *Proc. of ACM/IEEE SEC'18*, Bellevue, WA, Oct. 2018.

[47] S. Paul, U. Drolia, Y. C. Hu, and S. T. Chakradhar, "Aqua: Analytical quality assessment for optimizing video analytics systems," in *Proc. of ACM/IEEE SEC'21*, San Jose, CA,, Dec. 2021.

[48] B. Luo, S. Tan, Z. Yu, and W. Shi, "Edgebox: Live edge video analytics for near real-time event detection," in *Proc. of ACM/IEEE SEC'18*, Bellevue, WA, Oct. 2018.

[49] S. Jain, X. Zhang, Y. Zhou, G. Ananthanarayanan, J. Jiang *et al.*, "Spatula: Efficient cross-camera video analytics on large camera networks," in *Proc. of ACM/IEEE SEC'20*, Virtual Event, Nov. 2020.

[50] A. Roy, Ben and M. Nehal, "Video analytics in the cloud and at the edge with AWS DeepLens and Kinesis Video Streams," June 2018. [Online]. Available: aws.amazon.com/blogs/machine-learning/video-analytics-in-the-cloud-and-at-the-edge-with-aws-deeplens-and-kinesis-video-streams

[51] "Video Surveillance HUAWEI CLOUD." [Online]. Available: huaweicloud.com/intl/en-us/solution/sc_technology_videosurveillance

[52] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "Visual privacy protection methods: A survey," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4177–4195, 2015.

[53] N. Raval, A. Machanavajjhala, and L. P. Cox, "Protecting visual secrets using adversarial nets," in *Proc. of IEEE/CFV CVPR'17 workshop*, Honolulu, HI, USA, July. 2017.

[54] K. Tajik, A. Gunasekaran, R. Dutta, B. Ellis, R. B. Bobba, M. Rosulek, C. V. Wright, and W.-c. Feng, "Balancing image privacy and usability with thumbnail-preserving encryption." *IACR Cryptol*, p. 295, 2019.

[55] M.-R. Ra, R. Govindan *et al.*, "P3: Toward privacy-preserving photo sharing," in *Proc. of USENIX NSDI'13*, Lombard, IL, USA, Apr. 2013.

[56] T. Li and L. Lin, "Anonymousnet: Natural face de-identification with measurable privacy," in *Proc. of IEEE/CVF CVPR'19 workshops*, Long Beach, CA, USA, June 2019.

[57] H. Yu, J. Lim *et al.*, "Pinto: enabling video privacy for commodity iot cameras," in *Proc. of ACM SIGSAC'18*, Toronto, Canada, Oct. 2018.

[58] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. of IEEE SP'17*, San Jose, CA, USA, May 2017.

[59] P. C. Roy and V. N. Boddeti, "Mitigating information leakage in image representations: A maximum entropy approach," in *Proc. of IEEE/CVF CVPR'19*, Long Beach, CA, June 2019.

[60] A. Li, J. Guo, H. Yang *et al.*, "Deepobfuscator: Obfuscating intermediate representations with privacy-preserving adversarial learning on smartphones," in *Proc. of ACM IoTDI'21*, Virtual Event, May 2021.