Privacy-Preserving Video Classification with Convolutional Neural Networks

Sikha Pentyala, Rafael Dowsley and Martine De Cock

Abstract—Many video classification applications require access to personal data, thereby posing an invasive security risk to the users' privacy. We propose a privacy-preserving implementation of single-frame method based video classification with convolutional neural networks that allows a party to infer a label from a video without necessitating the video owner to disclose their video to other entities in an unencrypted manner. Similarly, our approach removes the requirement of the classifier owner from revealing their model parameters to outside entities in plaintext. To this end, we combine existing Secure Multi-Party Computation (MPC) protocols for private image classification with our novel MPC protocols for oblivious single-frame selection and secure label aggregation across frames. The result is an end-to-end privacy-preserving video classification pipeline. We evaluate our proposed solution in an application for private human emotion recognition. Our results across a variety of security settings, spanning honest and dishonest majority configurations of the computing parties, and for both passive and active adversaries, demonstrate that videos can be classified with state-of-the-art accuracy, and without leaking sensitive user information.

Index Terms—Privacy preserving, Secure multi-party computation

I. INTRODUCTION

Deep learning based video classification is extensively used in a growing variety of applications, such as facial recognition, activity recognition, gesture analysis, behavioral analysis, eye gaze estimation, and emotion recognition in empathy-based AI systems [3], [47], [42], [46], [49], [51], [58], [65], [72]. Many existing and envisioned applications of video classification rely on personal data, rendering these applications invasive of privacy. This applies among other tasks to video surveillance and home monitoring systems. Similarly, empathy-based AI systems expose personal emotions, which are most private to a user, to the service provider. Video classification systems deployed in commercial applications commonly require user videos to be shared with the service provider or sent to the cloud. These videos may remain publicly available on the Internet. Users have no control over the deletion of the videos, and the data may be available for scraping, as done for instance by Clearview AI [30]. The need to protect the privacy of individuals is widely acknowledged [56]. Concerns regarding privacy of user data are giving rise to new laws and regulations such as the European GDPR and the California Consumer Privacy Act (CCPA), as well as a perceived tension between

the desire to protect data privacy on one hand, and to promote an economy based on free-flowing data on the other hand [37]. The E.U. is for instance considering a three-to-five-year moratorium on face recognition in public places, given its significant potential for misuse [17].

A seemingly straightforward technique to keep user videos private is to deploy the deep learning models of the service providers at the user-end instead of transferring user data to the cloud. This is not a viable solution for several reasons. First, owners of proprietary models are concerned about shielding their model, especially when it constitutes a competitive advantage. Second, in security applications such as facial recognition, or deepfake detection, revealing model details helps adversaries develop evasion strategies. Furthermore, powerful deep learning models that memorize their training examples are well known; one would not want to expose those by revealing the model. Finally, deployment of large deep learning models at the user end may be technically difficult or impossible due to limited computational resources. For these reasons, ML tasks such as video classification are commonly outsourced to a set of efficient cloud servers in a Machine-Learning-as-a-Service (MLaaS) architecture. Protecting the privacy of both the users' and the service provider's data while performing outsourced ML computations is an important challenge.

Privacy-preserving machine learning (PPML) has been hailed, even by politicians [20], [73], as a potential solution when handling sensitive information. Substantial technological progress has been made during the last decade in the area of Secure Multi-Party Computation (MPC) [21], an umbrella term for cryptographic approaches that allow two or more parties to jointly compute a specified output from their private information in a distributed fashion, without revealing the private information to each other. Initial applications of MPC based privacy-preserving inference with deep learning models have been proposed for image [2], [24], [36], [41], [52], [62], [61], [63] and audio classification [7]. We build on this existing work to create the first end-to-end MPC protocol for private video classification. In our solution, videos are classified according to the well-known *single-frame* method, i.e. by aggregating predictions across single frames/images. Our main novel contributions are:

- A protocol for selecting frames in an oblivious manner.
- A protocol for secure frame label aggregation.
- An evaluation of our secure video classification pipeline in an application for human emotion detection from video on the RAVDESS dataset, demonstrating that MPC based video classification is feasible today, with state-of-the-art classification accuracies, and without leaking sensitive user information.

Sikha Pentyala is with the School of Engineering and Technology, University of Washington, Tacoma, WA, USA. Email: sikha@uw.edu

Rafael Dowsley is with the Faculty of Information Technology, Monash University, Clayton, Australia. Email: rafael.dowsley@monash.edu

Martine De Cock is with the School of Engineering and Technology, University of Washington, Tacoma, WA, USA and Ghent University, Ghent, Belgium. Email: mdecock@uw.edu
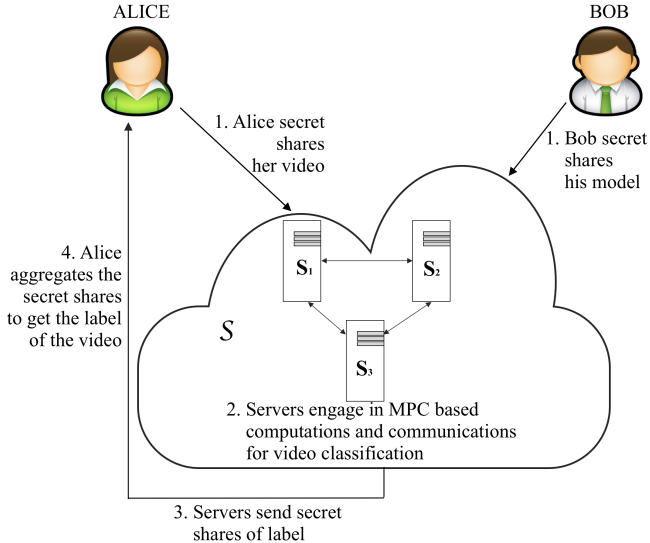
Fig. 1. Privacy-preserving video classification as an outsourced computation problem, illustrated for the 3-party computation setting (3PC) with 3 servers S_1, S_2, and S_3

Fig. 1 illustrates the flow of our proposed solution at a high level. The video of end user *Alice* should be classified with *Bob*'s model in such a way that no one other than Alice sees the video, and no one other than Bob sees the model parameters. Below we refer to both Alice's video and Bob's model parameters as "data". In Step 1 of Fig. 1, Alice and Bob each send secret shares of their data to a set S of untrusted servers ("parties"). While the secret shared data can be trivially revealed by combining all shares, nothing about the data is revealed to any subset of the servers that can be corrupted by the adversary. This means, in particular, that none of the servers by themselves learns anything about the actual values of the data. Next, in Step 2, the servers execute MPC protocols for oblivious frame selection, image classification, and frame label aggregation. Throughout this process, none of the servers learns the values of the data nor the assigned label, as all computations are done over secret shares. Finally, in Step 3, the servers can reveal their shares of the computed class label to Alice, who combines them in Step 4 to learn the output of the video classification.

Steps 1 and 3-4 are trivial as they follow directly from the choice of the underlying MPC scheme (see Sec. III). The focus of this paper is on Step 2, in which the servers (parties) execute protocols to perform computations over the secret shared data (see Sec. IV). MPC is concerned with the protocol execution coming under attack by an adversary which may corrupt parties to learn private information or cause the result of the computation to be incorrect. MPC protocols are designed to prevent such attacks being successful. There exist a variety of MPC schemes, designed for different numbers of parties and offering various levels of security that correspond to different threat models, and coming with different computational costs. Regarding threat models, we consider settings with *semi-honest* as well as with *malicious* adversaries. While parties corrupted by semi-honest adversaries follow the protocol instructions correctly but try to obtain additional information, parties corrupted by malicious adversaries can deviate from the protocol instructions. Regarding the *number* of parties (servers), some of the most efficient MPC schemes have been developed for 3 parties, out of which at most one is corrupted. We evaluate the runtime of our protocols in this honest-majority 3-party computing setting (3PC), which is growing in popularity in the PPML literature, e.g. [24], [41], [62], [67], [57], and we demonstrate how even better runtimes can be obtained with a recently proposed MPC scheme for 4PC with one corruption [23]. Our protocols are generic and can be used in a 2PC, dishonest-majority setting as well, i.e. where each party can only trust itself. Note that in the 2PC setting, the computation can be performed directly by Alice and Bob if they are not very limited in terms of computational resources. As known from the literature, and apparent from our results, the higher level of security offered by the 2PC setting comes with a substantial increase in runtime.

After discussing related work in Sec. II and recalling preliminaries about MPC in Sec. III, we present our protocols for privacy-preserving video classification in Sec. IV. The MPC protocols we present in Sec. IV enable the servers to perform all these computations without accessing the video $\mathcal{V}$ or the convolutional neural network (ConvNet) model $\mathcal{M}$ in plaintext. In Sec. V we present an experimental evaluation of our method when applied to emotion recognition from videos of the RAVDESS dataset. Our ConvNet based secure video classification approach achieves accuracies at par with those in the literature for this dataset, while not requiring leakage of sensitive information. Our prototype classifies videos that are 3-5 sec in length in under 14 sec on Azure F32 machines, demonstrating that private video classification based on MPC is feasible today.

II. RELATED WORK

Privacy-preserving video classification. Given the invasive nature of video classification applications, it is not surprising that efforts have been made to protect the privacy of individuals. Non-cryptography based techniques such as anonymizing faces in videos [60], pixel randomization to hide the user's identity [32], compressing video frames to achieve visual shielding effect [45], lowering resolution of videos [64], using autoencoders to maintain privacy of the user's data [27], and changes in ways the videos are captured [70] do not provide any formal privacy guarantees and affect the accuracy of the inference made. Solutions based on *Differential Privacy (DP)* [69] introduce noise, or replace the original data at the user end by newly generated data, to limit the amount of information leaked, at the cost of lowering accuracy. The recently proposed "Visor" system requires secure hardware (trusted execution environments) for privacy-preserving video analytics [59].

In contrast to the approaches above, in this paper we pursue the goal of having *no leakage* of information during the inference phase, *without* requiring special secure hardware. To the best of our knowledge, our approach is the first in the open literature to achieve this goal for private video classification. To this end, we leverage prior work on cryptography based private image classification, as described below, and augment it with novel cryptographic protocols for private video frame selection and label aggregation across frames.

Cryptography based image classification. There are 2 main approaches within cryptography that enable computations over encrypted data, namely *Homomorphic Encryption* (HE) and *Secure Multiparty Computation* (MPC). Both have been applied to secure inference with trained neural networks, including for image classification with ConvNets [10], [29], [40], [41], [57], [16], [61], [62], [67], [68]. Neither have been applied to video classification before. While HE has a lower communication burden than MPC, it has much higher computational costs, making HE less appealing at present for use in applications where response time matters. E.g., in state-of-the-art work on private image classification with HE, Chillotti et al. ([18]) report a classification time of ~ 9 sec for a 28×28 MNIST image on 96vCPU AWS instances with a neural network smaller in size (number of parameters) than the one we use in this paper. As demonstrated in Sec. V, the MPC based techniques for image classification based on Dalskov et al. ([24]) that we use, can label images (video frames) an order of magnitude faster, even when run on less powerful 32vCPU Azure instances (~ 0.5 sec for passive 3PC; ~ 1 sec for active 4PC). We acknowledge that this superior performance stems from the flexibility of MPC to accommodate honest-majority 3PC/4PC scenarios. HE based private image classification is by design limited to the dishonest-majority 2PC setting, in which our MPC approach is too slow for video classification in (near) real-time as well.

Emotion recognition. A wide variety of applications have prompted research in emotion recognition, using various modalities and features [6], [34], [35], [71], including videos [75], [31], [53], [54], [26]. Emotion recognition from videos in the RAVDESS benchmark dataset, as we do in the use case in Sec. V, has been studied by other authors in-the-clear, i.e. without regards for privacy protection, using a variety of deep learning architectures, with reported accuracies in the 57%-82% range, depending on the number of emotion classes included in the study (6 to 8) [5], [50], [9], [1]. The ConvNet model that we trained for our experimental results in Sec. V is at par with these state-of-the-art accuracies. Jaiswal and Provost [33] have studied privacy metrics and leakages when inferring emotions from data. To the best of our knowledge, there is no existing work on privacy-preserving emotion detection from videos using MPC, as we do in Sec. V.

III. PRELIMINARIES

Protocols for Secure Multi-Party Computation (MPC) enable a set of parties to jointly compute the output of a function over the private inputs of each party, without requiring any of the parties to disclose their own private inputs. MPC is concerned with the protocol execution coming under attack by an adversary $\mathcal{A}$ which may corrupt one or more of the parties to learn private information or cause the result of the computation to be incorrect. MPC protocols are designed to prevent such attacks being successful, and can be mathematically proven to guarantee privacy and correctness. We follow the standard definition of the Universal Composability (UC) framework [11], in which the security of protocols is analyzed by comparing a real world with an ideal world. For details, see Evans et al. ([28]).

		MPC scheme	Reference
passive	2PC	OTSemi2k	semi-honest adaptation of [22]
	3PC	Replicated2k	[4]
active	2PC	SPDZ2k	[22], [25]
	3PC	SPDZ-wise Replicated2k	[23]
	4PC	Rep4-2k	[23]

TABLE I
MPC SCHEMES USED IN THE EXPERIMENTAL EVALUATION FOR 2PC (DISHONEST MAJORITY) AND 3PC/4PC (HONEST MAJORITY)

An adversary $\mathcal{A}$ can corrupt any number of parties. In a *dishonest-majority* setting at least half of the parties are corrupt, while in an *honest-majority* setting, more than half of the parties are honest (not corrupted). Furthermore, $\mathcal{A}$ can have different levels of adversarial power. In the *semi-honest* model, even corrupted parties follow the instructions of the protocol, but the adversary attempts to learn private information from the internal state of the corrupted parties and the messages that they receive. MPC protocols that are secure against semi-honest or *"passive"* adversaries prevent such leakage of information. In the *malicious* adversarial model, the corrupted parties can arbitrarily deviate from the protocol specification. Providing security in the presence of malicious or *"active"* adversaries, i.e. ensuring that no such adversarial attack can succeed, comes at a higher computational cost than in the passive case.

The protocols in Sec. IV are sufficiently generic to be used in dishonest-majority as well as honest-majority settings, with passive or active adversaries. This is achieved by changing the underlying MPC scheme to align with the desired security setting. Table I contains an overview of the MPC schemes used in Sec. V. In these MPC schemes, all computations are done on integers modulo q, i.e., in a ring $\mathbb{Z}_q = \{0, 1, \ldots, q - 1\}$, with q a power of 2. The pixel values in *Alice*'s video and the model parameters in *Bob*'s classifier are natively real numbers represented in a floating point format. As is common in MPC, they are converted to integers using a fixed-point representation [15]. When working with fixed-point representations with a fractional bits, every multiplication generates an extra a bits of unwanted fractional representation. To securely "chop off" the extra fractional bits generated by multiplication, we use the deterministic truncation protocol by Dalskov et al. ([24], [23]) for computations over $\mathbb{Z}_{2^k}$. Below we give a high level description of the 3PC schemes from Table I. For more details and a description of the other MPC schemes, we refer to the papers in Table I.

Replicated sharing (3PC). After *Alice* and *Bob* have converted all their data to integers modulo q, they send secret shares of these integers to the servers in S (see Fig. 1). In a replicated secret sharing scheme with 3 servers (3PC), a value x in $\mathbb{Z}_q$ is secret shared among servers (parties) $S_1, S_2,$ and S_3 by picking uniformly random shares $x_1, x_2, x_3 \in \mathbb{Z}_q$ such that $x_1 + x_2 + x_3 = x \mod q$, and distributing (x_1, x_2) to S_1, (x_2, x_3) to S_2, and (x_3, x_1) to S_3. Note that no single server can obtain any information about x given its shares. We use $[\![x]\!]$ as a shorthand for a secret sharing of x. The servers subsequently classify Alice's video with Bob's model by computing over the secret sharings.

Passive security (3PC). The 3 servers can perform the following operations through carrying out local computations on their own shares: addition of a constant, addition of secret shared values, and multiplication by a constant. For multiplying secret shared values $[\![x]\!]$ and $[\![y]\!]$, we have that $x \cdot y = (x_1 + x_2 + x_3)(y_1 + y_2 + y_3)$, and so S_1 computes $z_1 = x_1 \cdot y_1 + x_1 \cdot y_2 + x_2 \cdot y_1$, S_2 computes $z_2 = x_2 \cdot y_2 + x_2 \cdot y_3 + x_3 \cdot y_2$ and S_3 computes $z_3 = x_3 \cdot y_3 + x_3 \cdot y_1 + x_1 \cdot y_3$. Next, the servers obtain an additive secret sharing of 0 by picking uniformly random u_1, u_2, u_3 such that $u_1 + u_2 + u_3 = 0$, which can be locally done with computational security by using pseudorandom functions, and S_i locally computes $v_i = z_i + u_i$. Finally, S_1 sends v_1 to S_3, S_2 sends v_2 to S_1, and S_3 sends v_3 to S_2, enabling the servers S_1, S_2 and S_3 to get the replicated secret shares (v_1, v_2), (v_2, v_3), and (v_3, v_1), respectively, of the value $v = x \cdot y$. This protocol only requires each server to send a single ring element to one other server, and no expensive public-key encryption operations (such as homomorphic encryption or oblivious transfer) are required. This MPC scheme was introduced by Araki et al. ([4]).

Active security (3PC). In the case of malicious adversaries, the servers are prevented from deviating from the protocol and gain knowledge from another party through the use of information-theoretic message authentication codes (MACs). For every secret share, an authentication message is also sent to authenticate that each share has not been tampered in each communication between parties. In addition to computations over secret shares of the data, the servers also need to update the MACs appropriately, and the operations are more involved than in the passive security setting. For each multiplication of secret shared values, the total amount of communication between the parties is greater than in the passive case. We use the MPC scheme *SPDZ-wiseReplicated2k* recently proposed by Dalskov et al. ([23]), with the option with preprocessing for generation of the multiplication triples that is available in MP-SPDZ [39].

a) MPC primitives.: The MPC schemes listed above provide a mechanism for the servers to perform cryptographic primitives through the use of secret shares, namely addition of a constant, multiplication by a constant, and addition and multiplication of secret shared values. Building on these cryptographic primitives, MPC protocols for other operations have been developed in the literature. We use:

- Secure matrix multiplication π_{DMM}: at the start of this protocol, the parties have secret sharings $[\![A]\!]$ and $[\![B]\!]$ of matrices A and B; at the end of the protocol, the parties have a secret sharing $[\![C]\!]$ of the product of the matrices, $C = A \times B$. π_{DMM} is a direct extension of the secure multiplication protocol for two integers explained above, which we will denote as π_{DM} in the remainder.
- Secure comparison protocol π_{LT} [13]: at the start of this protocol, the parties have secret sharings $[\![x]\!]$ and $[\![y]\!]$ of integers x and y; at the end of the protocol they have a secret sharing of 1 if $x < y$, and a secret sharing of 0 otherwise.
- Secure argmax π_{ARGMAX}: this protocol accepts secret sharings of a vector of integers and returns a secret sharing of the index at which the vector has the maximum value. π_{ARGMAX} is straightforwardly constructed using the above mentioned secure comparison protocol.

- Secure RELU π_{RELU} [24]: at the start of this protocol, the parties have a secret sharing of z; at the end of the protocol, the parties have a secret sharing of the value $\max(0, z)$. π_{RELU} is constructed from π_{LT}, followed by a secure multiplication to either keep the original value z or replace it by zero in an oblivious way.
- Secure division π_{DIV}: for secure division, the parties use an iterative algorithm that is well known in the MPC literature [14].

IV. METHODOLOGY

The servers perform video classification based on the single-frame method, i.e. by (1) selecting frames from the video $\mathcal{V}$ (Sec. IV-A); (2) labeling the selected frames with a ConvNet model $\mathcal{M}$ (Sec. IV-B); and (3) aggregating the labels inferred for the selected frames into a final label for the video (Sec. IV-C). The video $\mathcal{V}$ is owned by *Alice* and the model $\mathcal{M}$ is owned by *Bob*. Neither party is willing or able to reveal their video/model to other parties in an unencrypted manner.

A. Oblivious frame selection

We assume that *Alice* has prepared her video $\mathcal{V}$ as a 4D array (tensor) A of size $N \times h \times w \times c$ where N is the number of frames, h is the height and w is the width of the frame, and c represents the number of color channels of the frame. As explained in Sec. III, Alice has converted the pixel values into integers using a fixed-point representation. The values of the dimensions N, h, w, c are known to *Bob* and the set of servers S. All other properties of the video are kept private, including the video length, the frames per second (fps), and video capture details such as the type of camera used. Moreover, *Bob* and the servers S do not learn the values of the pixels, i.e. the actual contents of the frames remain hidden from *Bob* and S (and anyone else, for that matter). For an illustration of *Alice*'s input, we refer to the top of Fig. 2, where $N = 4$, $h = 2$, $w = 2$, and $c = 1$.
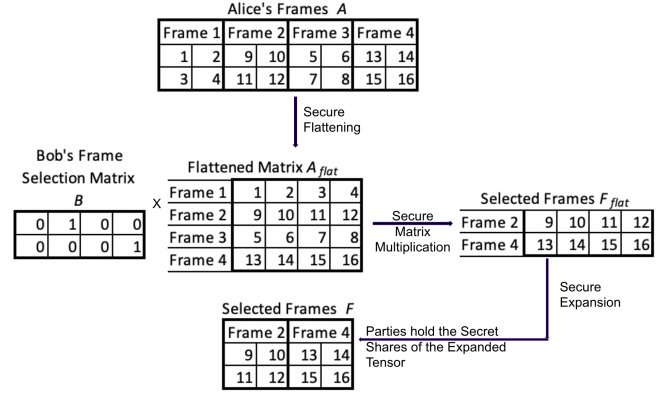

Fig. 2. Illustration of oblivious frame selection. The assumption is made that *Alice* has 4 frames in total, each of size $2 \times 2 \times 1$, and *Bob* needs to select 2 frames, namely Frames 2 and 4. *Alice* has a tensor A of size $4 \times 2 \times 2 \times 1$ and *Bob* has a 2D-matrix B of size 2×4. A is flattened securely to form A_{flat} of size 4×4. A secure matrix multiplication $B \times A_{\mathsf{flat}}$ is performed resulting in F_{flat}, a 2×4 matrix holding the 2 selected frames. This matrix is then expanded to matrix F of size $2 \times 2 \times 2 \times 1$.

Bob samples a fixed number of frames from *Alice*'s video, without revealing to *Alice* the frames he is selecting, as such knowledge might allow *Alice* to insert malicious frames in the video in the exact positions that *Bob* is sampling. We assume that *Bob* has a vector b of length n, with the indices of the n frames he wishes to select. These indices can for instance be $1, 1+d, 1+2d, \ldots$ for a fixed window size d that is known to *Bob*. In the example in Fig. 2, $n = 2$, both 2nd and 4th frames are selected.

The idea behind protocol π_{FSELECT} for oblivious frame selection, as illustrated in Fig. 2, is to flatten A into a matrix that contains one row per frame, use a matrix B with one-hot-encodings of the selected frames, multiply B with A, and finally expand the product. In more detail: *Bob* converts each entry i of list b (which is an index of a frame to be selected) into a vector of length N that is a one-hot-encoding of i, and inserts it as a row in matrix B of size $n \times N$. *Alice* and *Bob* then send secret shares of their respective inputs A and B to the servers S, using a secret sharing scheme as mentioned in Sec. III. None of the servers can reconstruct the values of A or B by using only its own secret shares.

Next the parties in S jointly execute protocol π_{FSELECT} for oblivious frame selection (see Protocol 1). On line 1, the parties reorganize the shares of tensor A of size $N \times h \times w \times c$ into a flattened matrix A_{flat} of size $N \times (h \cdot w \cdot c)$. On line 2, the parties multiply $[\![B]\!]$ and $[\![A_{\mathsf{flat}}]\!]$, using protocol π_{DMM} for secure matrix multiplication, to select the desired rows from A_{flat}. On line 3, these selected rows are expanded again into a secret-shared tensor F of size $n \times h \times w \times c$ that holds the selected frames. $F[1], F[2], \ldots, F[n]$ are used in the remainder to denote the individual frames contained in F. Throughout this process, the servers do not learn the pixel values from A, nor which frames were selected.

Protocol 1 Protocol π_{FSELECT} for oblivious frame selection

Input: A secret shared 4D-array $[\![A]\!]$ of size $N \times h \times w \times c$ with the frames of a video; a secret shared frame selection matrix $[\![B]\!]$ of size $n \times N$. The values N, h, w, c, n are known to all parties.
Output: A secret shared 4D-array F of size $n \times h \times w \times c$ holding the selected frames

1: $[\![A_{\mathsf{flat}}]\!] \leftarrow$
   $\mathsf{RESHAPE}([\![A]\!], N \times h \times w \times c, N \times (h \times w \times c))$
2: $[\![F_{\mathsf{flat}}]\!] \leftarrow \pi_{\mathsf{DMM}}([\![B]\!], [\![A_{\mathsf{flat}}]\!])$
3: $[\![F]\!] \leftarrow \mathsf{RESHAPE}([\![F_{\mathsf{flat}}]\!], n \times (h \times w \times c), n \times h \times w \times c)$
4: **return** $[\![F]\!]$

B. Private frame classification

We assume that *Bob* has trained an "MPC-friendly" 2D-ConvNet $\mathcal{M}$ for classifying individual video frames (images), and that *Bob* secret shares the values of the model parameters with the servers S, who already have secret shares of the selected frames from *Alice*'s video after running Protocol π_{FSELECT}. By "MPC-friendly" we mean that the operations to be performed when doing inference with the trained ConvNet are chosen purposefully among operations for which efficient MPC protocols exist or can be constructed. Recall that a standard ConvNet contains one or more blocks that each have a convolution layer, followed by an activation layer, typically with RELU, and an optional pooling layer. These blocks are then followed by fully connected layers which commonly have

RELU as activation function, except for the last fully connected layer which typically has a Softmax activation for multi-class classification. The operations needed for all layers, except for the output layer, boil down to comparisons, multiplicatons, and summations. All of these cryptographic primitives can be efficiently performed with state-of-the-art MPC schemes, as explained in Sec. III. Efficient protocols for convolutional, RELU activation, average pooling layers, and dense layers are known in the MPC literature [24]. We do not repeat them in this paper for conciseness. All these operations are performed by the servers S using the secret shares of *Bob*'s model parameters and of the selected frame from *Alice*'s video, as obtained using π_{FSELECT}.

As previously mentioned, Softmax is generally used as the activation function in the last layer of ConvNets that are trained to perform classification. Softmax normalizes the logits passed into it from the previous layer to a probability distribution over the class labels. Softmax is an expensive operation to implement using MPC protocols, as this involves division and exponentiation. Previously proposed workarounds include disclosing the logits and computing Softmax in an unencrypted manner [44], which leaks information, or replacing Softmax by Argmax [7], [24]. The latter works when one is only interested in retrieving the class label with the highest probability, as the Softmax operation does not change the ordering among the logits. In our context of video classification based on the single-frame method however, the probabilities of all class labels for each frame are required, to allow probabilities across the different frames to be aggregated to define a final label (see Sec. IV-C).

We therefore adopt the solution proposed by Mohassel and Zhang [55] and replace the Softmax operation by

$$f(u_i) = \begin{cases} \dfrac{\mathrm{RELU}(u_i)}{\sum\limits_{j=1}^{C} \mathrm{RELU}(u_j)}, & \text{if } \sum\limits_{j=1}^{C} \mathrm{RELU}(u_j) > 0 \\[4ex] 1/C, & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, C$, where $(u_1, u_2, \ldots, u_C)$ denote the logits for each of the C class labels, and $(f(u_1), f(u_2), \ldots, f(u_C))$ is the computed probability distribution over the class labels. Pseudocode for the corresponding MPC protocol is presented in Protocol 2. At the start of Protocol π_{SOFT}, the servers have secret shares of a list of logits, on which they apply the secure RELU protocol in Line 1. Lines 2-5 serve to compute the sum of the RELU values, while on Line 6 the parties run a secure comparison protocol to determine if this sum is greater than 0. If Sum_{relu} is greater than 0, then after Line 6, $[\![cn]\!]$ contains a secret sharing of 1; otherwise it contains a secret sharing of 0. Note that if $cn = 1$ then the numerator of the ith probability $f(u_i)$ should be $X_{\mathsf{relu}}[i]$ while the denominator should be Sum_{relu}. Likewise, if $cn = 0$ then the numerator should be 1 and the denominator C. As is common in MPC protocols, we use multiplication instead of control flow logic for such conditional assignments. To this end, a conditional based branch operation as "**if** p **then** $q \leftarrow r$ **else** $q \leftarrow s$" is rephrased as "$q \leftarrow p \cdot r + (1-p) \cdot s$". In this way, the number and the kind of operations executed by the parties does not depend on the actual values of the inputs, so it does not leak information that could be exploited by side-channel attacks. Such conditional

	SM$_{approx}$ for Frames						
Labels →	1	2	3	4	5	6	7
Frame 1	0	0	0	0	0.28	0	0.72
Frame 2	0	0	0	0	0.55	0.45	0
Frame 3	0	0	0	0	0.83	0.17	0
Frame 4	0	0.21	0	0	0.48	0.31	0
prob$_{sum}$	0	0.21	0	0	2.14	0.93	0.72

Output Label L is 5

Fig. 3. Illustration of label aggregation. Let us assume that $n = 4$ frames were selected for secure inference, and that there are $C = 7$ classes. SM_{approx} holds the inferred probability distribution over the class labels for each frame. Class label 5 is selected as the final label because it has the highest sum of probabilities across all classified frames.

assignments occur in Line 7 and 10 of Protocol π_{SOFT}, to assign the correct value of the numerator and the denominator.

Protocol 2 Protocol π_{SOFT} for approximate Softmax

Input: A secret shared list $[\![logits]\!]$ of logits of size C, where C is total number of class labels
Output: A secret shared list $[\![SM_{approx}]\!]$ of size C of probabilities for the class labels

1: $[\![X_{relu}]\!] \leftarrow \pi_{RELU}([\![logits]\!])$
2: $[\![Sum_{relu}]\!] \leftarrow 0$
3: **for** $j = 1$ **to** C **do**
4: $\quad [\![Sum_{relu}]\!] \leftarrow [\![Sum_{relu}]\!] + [\![X_{relu}[i]]\!]$
5: **end for**
6: $[\![cn]\!] \leftarrow \pi_{LT}(0, [\![Sum_{relu}]\!])$
7: $[\![denom]\!] \leftarrow \pi_{DM}([\![cn]\!], [\![Sum_{relu}]\!]) + \pi_{DM}((1 - [\![cn]\!]), C)$
8: $[\![denom\_inv]\!] \leftarrow \pi_{DIV}(1, [\![denom]\!])$
9: **for** $i = 1$ **to** C **do**
10: $\quad [\![numer]\!] \leftarrow \pi_{DM}([\![cn]\!], [\![X_{relu}[i]]\!]) + (1 - [\![cn]\!])$
    $\quad [\![SM_{approx}[i]]\!] \leftarrow \pi_{DM}([\![numer]\!], [\![denom\_inv]\!])$
11: **end for**
12: **return** $[\![SM_{approx}]\!]$

A protocol π_{FINFER} for performing secure inference with *Bob*'s model $\mathcal{M}$ (which is secret shared among the servers) over a secret shared frame f from *Alice*'s video can be straightforwardly obtained by: (1) using the cryptographic primitives defined in Sec. III to securely compute all layers except the output layer; (2) using Protocol π_{SOFT} to compute the approximation of the Softmax for the last layer. The execution of this protocol results in the servers obtaining secret shares of the inferred probability distribution over the class labels for frame f.

C. Secure label aggregation

As illustrated in Fig. 3, we aggregate the predictions across the single frames by selecting the class label with the highest sum of inferred probabilities across the frames. We implement this securely as Protocol 3. To classify a video $\mathcal{V}$, the servers: (1) obliviously select the desired frames as shown in Line 2; (2) securely infer the probability distribution SM_{approx} of all classes labels generated by the model $\mathcal{M}$ on a specific selected frame, as shown in Line 4; (3) add these probabilities, index-wise, to the sum of the probabilities corresponding to each class that is obtained throughout the selected frames (Line 5-6); (4) securely find the index L with maximum value in the aggregated list (Line 8). L represents the class label for

the video. At the end of Protocol 3, the servers hold a secret sharing $[\![L]\!]$ of the video label. Each of the servers sends its secret shares to *Alice*, who uses them to construct the class label L for the video.

Protocol 3 Protocol $\pi_{LABELVIDEO}$ for classifying a video securely based on the single-frame method

Input: A video $\mathcal{V}$ secret shared as a 4D-array $[\![A]\!]$, a frame selection matrix secret shared as $[\![B]\!]$, the parameters of the ConvNet model $\mathcal{M}$ secret shared as $[\![M]\!]$
Output: A secret share $[\![L]\!]$ of the video label

1: Let $[\![prob_{sum}]\!]$ be a list of length C that is initialized with zeros in all indices.
2: $[\![F]\!] \leftarrow \pi_{FSELECT}([\![A]\!], [\![B]\!])$
3: **for all** $[\![F[j]]\!]$ **do**
4: $\quad [\![SM_{approx}]\!] \leftarrow \pi_{FINFER}([\![M]\!], [\![F[j]]\!])$
5: $\quad$ **for** $i = 1$ **to** C **do**
6: $\qquad [\![prob_{sum}[i]]\!] \leftarrow [\![prob_{sum}[i]]\!] + [\![SM_{approx}[i]]\!]$
7: $\quad$ **end for**
8: **end for**
9: $[\![L]\!] \leftarrow \pi_{ARGMAX}([\![prob_{sum}]\!])$
10: **return** $[\![L]\!]$

V. RESULTS

A. Dataset and model architecture

We demonstrate the feasibility of our privacy-preserving video classification approach for the task of emotion detection using the database [1] [48]. We use 1,248 video-only files with speech modality from this dataset, corresponding to 7 different emotions, namely *neutral* (96), *happy* (192), *sad* (192), *angry* (192), *fearful* (192), *disgust* (192), and *surprised* (192). The videos portray 24 actors who each read two different statements twice, with different emotions, for a total of 52 video files per actor. For all emotions except for neutral, the statements are read with alternating normal and strong intensities; this accounts for the fact that there are less "neutral" instances in the dataset than for the other emotion categories. As in [9], we leave out the *calm* instances, reducing the original 8 emotion categories from the RADVESS dataset to the 7 categories that are available in the FER2013 dataset [12], which we use for pre-training. The videos in the RAVDESS dataset have a duration of 3-5 seconds with 30 frames per second, hence the total number of frames per video is in the range of 120-150. We split the data into 1,116 videos for training and 132 videos for testing. To this end, we moved all the video recordings of the actors 8, 15 (selected randomly) and an additional randomly selected 28 video recordings to the test set, while keeping the remaining video recordings in the train set.

We used OpenCV [8] to read the videos into frames. Faces are detected with a confidence greater than 98% using MTCNN [74], aligned, cropped, and converted to gray-scale. Each processed frame is resized to 48×48, reshaped to a 4D-array, and normalized by dividing each pixel value by 255.

For *Bob*'s image classification model, we trained a Conv-Net with ~ 1.48 million parameters with an architecture of [(CONV-RELU)-POOL]-[(CONV-RELU)*2-POOL]*2-[FC-RELU]*2-[FC-SOFTMAX]. We pre-trained[2] the feature layers

---

[1] https://zenodo.org/record/1188976
[2] With early stopping using a batch size of 256 and Adam optimizer with default parameters in Keras [19].

		Avg. Time (sec)	Avg. Comm (GB)
Passive	2PC	511.64	669.35
	3PC	**13.12**	2.49
Active	2PC	8423.81	7782.74
	3PC	48.20	10.98
	4PC	**18.40**	4.60

TABLE II

AVERAGE TIME TO PRIVATELY DETECT EMOTION IN A RAVDESS VIDEO OF DURATION 3-5 SECONDS. THE AVERAGE TIME IS COMPUTED OVER A SET OF 10 VIDEOS WITH A NUMBER OF FRAMES IN THE 7-10 RANGE, AND WITH N_THREADS=32 IN MP-SPDZ. COMMUNICATION IS MEASURED PER PARTY.

on the FER 2013 data to learn to extract facial features for emotion recognition, and fine-tuned[3] the model on the RAVDESS training data. Our video classifier samples every 15th frame, classifies it with the above ConvNet, and assigns as the final class label the label that has the highest average probability across all frames in the video. The video classification accuracy on the test set is 56%. For inference with the MPC protocols, after training, we replace the Softmax function on the last layer by the approximate function discussed in Section IV-B. After this replacement, the accuracy of the video classifier is 56.8%. This is in line with state-of-the-art results in the literature on emotion recognition from RAVDESS videos, namely 57.5% with Synchronous Graph Neural Networks (8 emotions) [50]; 61% with ConvNet-LSTM (8 emotions) [1]; 59% with an RNN (7 emotions) [9], and 82.4% with stacked autoencoders (6 emotions) [5].

B. Runtime experiments

We implemented the protocols from Sec. IV in the MPC framework MP-SPDZ [39], and ran experiments on co-located F32s V2 Azure virtual machines. Each of the parties (servers) ran on separate VM instances (connected with a Gigabit Ethernet network), which means that the results in Table II cover communication time in addition to computation time. A F32s V2 virtual machine contains 32 cores, 64 GiB of memory, and network bandwidth of upto 14 Gb/s. For the ring $\mathbb{Z}_{2^k}$, we used value $k = 64$.

Table II presents the average time needed to privately classify a video. The MPC schemes for 4PC (with one corrupted party) are faster than 3PC (with one corrupted party), which are in turn substantially faster than 2PC. Furthermore, as expected, there is a substantial difference in runtime between the semi-honest (passive security) and malicious (active security) settings. These findings are in line with known results from the MPC literature [23], [24]. In the fastest setting, namely a 3PC setting with a semi-honest adversary that can only corrupt one party, videos from the RAVDESS dataset are classified on average in 13.12 sec, which corresponds to approximately 0.5-0.6 sec per frame, demonstrating that privacy-preserving video classification with state-of-the-art accuracy is feasible in practice. While the presented runtime results are still too slow for video classification in real-time, there is a clear path to substantial optimization that would enable deployment of our proposed MPC solution in practical real-time applications. Indeed, MPC schemes are normally divided in two phases:

the offline and online phases. The runtime results in Table II represent the time needed for both. The offline phase only performs computations that are independent from the specific inputs of the parties to the protocol (*Alice*'s video and *Bob*'s trained model parameters), and therefore can be executed long before the inputs are known. By executing the offline phase of the MPC scheme in advance, it is possible to improve the responsiveness of the final solution.

VI. CONCLUSION AND FUTURE WORK

We presented the first end-to-end solution for private video classification based on Secure Multi-Party Computation (MPC). To achieve state-of-the-art accuracy while keeping our architecture lean, we used the single-frame method for video classification with a ConvNet. To keep the videos and the model parameters hidden, we proposed novel MPC protocols for oblivious frame selection and secure label aggregation across frames. We used these in combination with existing MPC protocols for secure ConvNet based image classification, and evaluated them for the task of emotion recognition from videos in the RAVDESS dataset.

Our work provides a baseline for private video classification based on cryptography. It can be improved and adapted further to align with state-of-the-art techniques in video classification in-the-clear, including the use of machine learning for intelligent frame selection. While our approach considers only spatial information in the videos, the model architecture in Sec. IV-B can be replaced by different architectures such as CONV3D, efficient temporal modeling in video [43], single and two stream ConvNets [38], [66] to fuse temporal information. Many such approaches use popular ImageNet models for which efficient MPC protocols are available in the literature [24], [41], opening up interesting directions for further research.

REFERENCES

[1] M. Abdullah, M. Ahmad, and D. Han. Facial expression recognition in videos: An CNN-LSTM based model for video classification. In *International Conference on Electronics, Information, and Communication*, pages 1–3, 2020. 3, 7

[2] N. Agrawal, A. Shahin Shamsabadi, M.J. Kusner, and A. Gascón. QUOTIENT: two-party secure neural network training and prediction, 2019. 1

[3] M.R. Ali, J. Hernandez, E.R. Dorsey, E. Hoque, and D. McDuff. Spatio-temporal attention and magnification for classification of Parkinson's disease from videos collected via the Internet. In *15th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 53–60, 2020. 1

[4] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara. High-throughput semi-honest secure three-party computation with an honest majority. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 805–817, 2016. 3, 4

[5] E. Bagheri, A. Bagheri, P.G. Esteban, and B. Vanderborgth. A novel model for emotion detection from facial muscles activity. In *Iberian Robotics conference*, pages 237–249. Springer, 2019. 3, 7

[6] U. Bhattacharya, T. Mittal, R. Chandra, T. Randhavane, A. Bera, and D. Manocha. STEP: Spatial temporal graph convolutional networks for emotion perception from gaits. In *34th AAAI Conference on Artificial Intelligence*, pages 1342–1350, 2020. 3

[7] Kyle Bittner, Martine De Cock, and Rafael Dowsley. Private speech classification with secure multiparty computation. *Available at https://arxiv.org/pdf/2007.00253.pdf*, 2020. 1, 5

[8] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008. 6

[9] S. Bursic, G. Boccignone, A. Ferrara, A. DAmelio, and R. Lanzarotti. Improving the accuracy of automatic facial expression recognition in speaking subjects with deep learning. *Applied Sciences*, 10(11):4002, 2020. 3, 6, 7

[3]With early-stopping using a batch size of 64 and SGD optimizer with a learning rate 0.001, decay as 10^{-6}, and momentum as 0.9.

[10] M. Byali, H. Chaudhari, A. Patra, and A. Suresh. Flash: fast and robust framework for privacy-preserving machine learning. *Proceedings on Privacy Enhancing Technologies*, 2020(2):459–480, 2020. 3

[11] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000. 3

[12] P.L. Carrier, A. Courville, I.J. Goodfellow, M. Mirza, and Y. Bengio. FER-2013 face database, Université de Montréal. https://datarepository.wolframcloud.com/resources/FER-2013, 2013. 6

[13] O. Catrina and S. De Hoogh. Improved primitives for secure multiparty integer computation. In *International Conference on Security and Cryptography for Networks*, pages 182–199. Springer, 2010. 4

[14] O. Catrina and S. De Hoogh. Secure multiparty linear programming using fixed-point arithmetic. In *European Symposium on Research in Computer Security*, pages 134–150. Springer, 2010. 4

[15] O. Catrina and A. Saxena. Secure computation with fixed-point numbers. In *14th International Conference on Financial Cryptography and Data Security*, volume 6052 of *Lecture Notes in Computer Science*, pages 35–50. Springer, 2010. 3

[16] H. Chaudhari, R. Rachuri, and A. Suresh. Trident: Efficient 4pc framework for privacy preserving machine learning. In *27th Annual Network and Distributed System Security Symposium, NDSS*, pages 23–26, 2020. 3

[17] F.Y. Chee. EU mulls five-year ban on facial recognition tech in public areas. *Reuters, Innovation and Technology*, Jan 16, 2020. 1

[18] Ilaria Chillotti, Marc Joye, and Pascal Paillier. Programmable bootstrapping enables efficient homomorphic inference of deep neural networks. Cryptology ePrint Archive, Report 2021/091, 2021. https://eprint.iacr.org/2021/091. 3

[19] François Chollet et al. Keras. https://keras.io, 2015. 6

[20] Commission of Evidence-Based Policymaking. The Promise of Evidence-Based Policymaking. https://www.cep.gov/content/dam/cep/report/cep-final-report.pdf, 2017. 1

[21] R. Cramer, I. Damgård, and J.B. Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015. 1

[22] Ronald Cramer, Ivan Damgård, Daniel Escudero, Peter Scholl, and Chaoping Xing. SPDZ$_{2^k}$: Efficient MPC mod 2^k for dishonest majority. In *Annual International Cryptology Conference*, pages 769–798. Springer, 2018. 3

[23] A. Dalskov, D. Escudero, and M. Keller. Fantastic four: Honest-majority four-party secure computation with malicious security. Cryptology ePrint Archive, Report 2020/1330, 2020. 2, 3, 4, 7

[24] A. Dalskov, D. Escudero, and M. Keller. Secure evaluation of quantized neural networks. *Proceedings on Privacy Enhancing Technologies*, 2020(4):355–375, 2020. 1, 2, 3, 4, 5, 7

[25] I. Damgård, D. Escudero, T. Frederiksen, M. Keller, P. Scholl, and N. Volgushev. New primitives for actively-secure MPC over rings with applications to private machine learning. In *IEEE Symposium on Security and Privacy (SP)*, pages 1102–1120, 2019. 3

[26] D. Deng, Z. Chen, Y. Zhou, and B. Shi. MIMAMO Net: Integrating micro-and macro-motion for video emotion recognition. *arXiv preprint arXiv:1911.09784*, 2019. 3

[27] M. D'Souza, J.F. Dorn, A. Dorier, C.P. Kamm, S. Steinheimer, F. Dahlke, C. EP Van Munster, B. MJ Uitdehaag, L. Kappos, and M. Johnson. Autoencoder as a new method for maintaining data privacy while analyzing videos of patients with motor dysfunction: Proof-of-concept study. *Journal of Medical Internet Research*, 22(5):e16669, 2020. 2

[28] D. Evans, V. Kolesnikov, and M. Rosulek. A pragmatic introduction to secure multi-party computation. *Foundations and Trends in Privacy and Security*, 2(2-3):70–246, 2018. 3

[29] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016. 3

[30] K. Hill. The secretive company that might end privacy as we know it. *The New York Times*, Jan 18, 2020. 1

[31] M. Hu, H. Wang, X. Wang, J. Yang, and R. Wang. Video facial emotion recognition based on local enhanced motion history image and CNN-CTSLSTM networks. *Journal of Visual Communication and Image Representation*, 59:176–185, 2019. 3

[32] J. Imran, B. Raman, and A.S. Rajput. Robust, efficient and privacy-preserving violent activity recognition in videos. In *35th Annual ACM Symposium on Applied Computing*, page 20812088, 2020. 2

[33] M. Jaiswal and E. Provost. Privacy enhanced multimodal neural representations for emotion recognition. In *34th AAAI Conference on Artificial Intelligence*, pages 7985–7993, 2020. 3

[34] X. Jia, X. Zheng, W. Li, C. Zhang, and Z. Li. Facial emotion distribution learning by exploiting low-rank label correlations locally. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9841–9850, 2019. 3

[35] W. Jiao, M.R. Lyu, and I. King. Real-time emotion recognition via attention gated hierarchical memory network. *arXiv preprint arXiv:1911.09075*, 2019. 3

[36] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasana. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium*, pages 1651–1669, 2018. 1

[37] L. Kalman. New European data privacy and cyber security laws – one year later. *Communications of the ACM*, 62:38–39, 2019. 1

[38] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014. 7

[39] Marcel Keller. MP-SPDZ: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1575–1590, 2020. 4, 7

[40] N. Koti, M. Pancholi, A. Patra, and A. Suresh. Swift: Super-fast and robust privacy-preserving machine learning. *arXiv preprint arXiv:2005.10296*, 2020. 3

[41] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma. CrypTFlow: Secure TensorFlow inference. In *41st IEEE Symposium on Security and Privacy*, 2020. 1, 2, 3, 7

[42] D. Li, C. Rodriguez, X. Yu, and H. Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1459–1469, 2020. 1

[43] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 7

[44] J. Liu, M. Juuti, Y. Lu, and N. Asokan. Oblivious neural network predictions via MiniONN transformations. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 619–631, 2017. 5

[45] J. Liu, Y. Xia, and Z. Tang. Privacy-preserving video fall detection using visual shielding information. *The Visual Computer*, pages 1–12, 2020. 2

[46] K. Liu, M. Zhu, H. Fu, H. Ma, and T. Chua. Enhancing anomaly detection in surveillance videos with transfer learning from action recognition. In *28th ACM International Conference on Multimedia*, page 46644668, 2020. 1

[47] X. Liu, W. Liu, M. Zhang, J. Chen, L. Gao, C. Yan, and T. Mei. Social relation recognition from videos via multi-scale spatial-temporal reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3566–3574, 2019. 1

[48] S.R. Livingstone and F.A. Russo. The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PloS One*, 13(5), 2018. 6

[49] S. Manna, S. Ghildiyal, and K. Bhimani. Face recognition from video using deep learning. In *5th International Conference on Communication and Electronics Systems (ICCES)*, pages 1101–1106, 2020. 1

[50] E. Mansouri-Benssassi and J. Ye. Synch-graph: multisensory emotion recognition through neural synchrony via graph convolutional networks. In *34th AAAI Conference on Artificial Intelligence*, 2020. 3, 7

[51] . Meinich-Bache, S. L. Austnes, K. Engan, I. Austvoll, T. Eftestl, H. Myklebust, S. Kusulla, H. Kidanto, and H. Ersdal. Activity recognition from newborn resuscitation videos. *IEEE Journal of Biomedical and Health Informatics*, 24(11):3258–3267, 2020. 1

[52] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R.A. Popa. Delphi: A cryptographic inference service for neural networks. In *29th USENIX Security Symposium*, pages 2505–2522, 2020. 1

[53] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha. M3ER: Multiplicative multimodal emotion recognition using facial, textual, and speech cues. In *34th AAAI Conference on Artificial Intelligence*, pages 1359–1367, 2020. 3

[54] T. Mittal, P. Guhan, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha. Emoticon: Context-aware multimodal emotion recognition using frege's principle. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14222–14231, 2020. 3

[55] P. Mohassel and Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38, 2017. 5

[56] National Science and Technology Council. National Privacy Research Strategy. https://www.nitrd.gov/PUBS/NationalPrivacyResearchStrategy.pdf, 2016. 1

[57] A. Patra and A. Suresh. BLAZE: Blazing fast privacy-preserving machine learning. *arXiv preprint arXiv:2005.09042*, 2020. 2, 3

[58] S. Peng, L. Chen, C. Gao, and R.J. Tong. Predicting students' attention level with interpretable facial and head dynamic features in an online tutoring system. In *34th AAAI Conference on Artificial Intelligence*, pages 13895–13896, 2020. 1

[59] R. Poddar, G. Ananthanarayanan, S. Setty, S. Volos, and R.A. Popa. Visor: Privacy-preserving video analytics as a cloud service. In *29th USENIX Security Symposium*, pages 1039–1056, 2020. 2

[60] Z. Ren, Y. Jae Lee, and M.S. Ryoo. Learning to anonymize faces for privacy preserving action detection. In *European Conference on Computer Vision (ECCV)*, pages 620–636, 2018. 2

[61] M.S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar. XONN: Xnor-based oblivious deep neural network inference. In *28th USENIX Security Symposium*, pages 1501–1518, 2019. 1, 3

[62] M.S. Riazi, C. Weinert, O. Tkachenko, E.M. Songhori, T. Schneider, and F. Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. In *Asia Conference on Computer and Communications Security*, pages 707–721, 2018. 1, 2, 3

[63] B.D. Rouhani, M.S. Riazi, and F. Koushanfar. DeepSecure: Scalable provably-secure deep learning. In *55th Annual Design Automation Conference (DAC)*, 2018. 1

[64] M.S. Ryoo, B. Rothrock, C. Fleming, and H.J. Yang. Privacy-preserving human activity recognition from extreme low resolution. In *31st AAAI Conference on Artificial Intelligence*, pages 4255–4262, 2017. 2

[65] A.P. Shah, V. Vaibhav, V. Sharma, M.A. Ismail, J. Girard, and L.P. Morency. Multimodal behavioral markers exploring suicidal intent in social media videos. In *International Conference on Multimodal Interaction*, page 409413. ACM, 2019. 1

[66] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27*, pages 568–576. 2014. 7

[67] S. Wagh, D. Gupta, and N. Chandran. SecureNN: 3-party secure computation for neural network training. *Proceedings on Privacy Enhancing Technologies*, 1:24, 2019. 2, 3

[68] S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin. Falcon: Honest-majority maliciously secure framework for private deep learning. *Proceedings on Privacy Enhancing Technologies*, 2021(1):188–208, 2021. 3

[69] H. Wang, Z. Wu, Z. Wang, Z. Wang, and H. Jin. Privacy-preserving deep visual recognition: An adversarial learning framework and a new dataset. *arXiv preprint arXiv:1906.05675*, 2019. 2

[70] Z.W. Wang, V. Vineet, F. Pittaluga, S.N. Sinha, O. Cossairt, and S. Bing Kang. Privacy-preserving action recognition using coded aperture videos. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2

[71] Z. Wei, J. Zhang, Z. Lin, J.Y. Lee, N. Balasubramanian, M. Hoai, and D. Samaras. Learning visual emotion representations from web data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13103–13112, 2020. 3

[72] J. Wu, L. Wang, Li. Wang, J. Guo, and G. Wu. Learning actor relation graphs for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9964–9974, 2019. 1

[73] R. Wyden. Wyden pushes for stronger security in collection of personal information. https://www.wyden.senate.gov/download/20170515-wyden-mpc-letter-to-cep, 2017. 1

[74] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 6

[75] S. Zhao, Y. Ma, Y. Gu, J. Yang, T. Xing, P. Xu, R. Hu, H. Chai, and K. Keutzer. An end-to-end visual-audio attention network for emotion recognition in user-generated videos. In *34th AAAI Conference on Artificial Intelligence*, pages 303–311, 2020. 3