Recognizing Camera Wearer from Hand Gestures in Egocentric Videos

https://egocentricbiometric.github.io/

Daksh Thapar
Indian Institute of Technology Mandi
Mandi, India

Aditya Nigam
Indian Institute of Technology Mandi
Mandi, India

Chetan Arora
Indian Institute of Technology Delhi
New Delhi, India

Figure 1: The above figure shows cutting (first two rows) and washing (last two rows) activities performed by two different subjects, as seen from the head-mounted egocentric cameras. The thesis of this paper is that egocentric cameras are able to capture wearer identifying hand gesture signatures, merely by looking at various activities being performed by the wearers. While it may be difficult for the reader to visually, our deep neural network-based model correctly identifies that activities in rows 1 and 3 have been performed by the same wearer, whereas the other subject has performed activities in rows 2, 4.

ABSTRACT

Wearable egocentric cameras are typically harnessed to a wearer's head, giving them the unique advantage of capturing their points of view. This characteristic has led to the concerns about egocentric cameras leaking wearer's privacy. Hoshen and Peleg [9] have shown that egocentric cameras indirectly capture the wearer's gait, which can be used to identify a wearer based on their egocentric videos. The authors have shown a wearer recognition accuracy of up to 77% over 32 subjects. However, an important limitation of their work is that such gait features can be extracted only from walking sequences of a wearer. In this work, we take the privacy threat a notch higher and show that even the wearer's hand gestures, as seen through an egocentric video, leak wearer's identity. We have designed a model to extract and match hand gesture signatures from egocentric videos. We demonstrate the threat on the EPIC kitchen dataset containing 55 hours of the egocentric videos acquired from 32 subjects doing various activities. We show that: (1) Our model can recognize a wearer with an accuracy of up to 73% based on the same activity, i.e., the model has seen 'cut' activity by a wearer in the train set, and recognizes the wearer based on another 'cut' activity by him/her while testing. (2) The hand gesture signatures transfer across activities, i.e., even if our model does not see 'cut' activity of a wearer at the train time, but sees other activities such as 'wash', 'mix' etc., the model can still recognize a wearer with an accuracy of up to 60%, by matching hand gesture signatures of 'cut' at test time with train time signatures of 'wash' or 'mix'. (3) The hand gesture features even transfer across subjects, i.e., even if the model has not seen any activity by some subject, one can still verify a wearer (open-set), and predict that the same wearer has performed both activities with an Equal Error Rate of 15.21%. The code, trained models are available at https://egocentricbiometric.github.io/

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413654

CCS CONCEPTS

• **Computing methodologies** → **Biometrics**; *Scene understanding*; Matching.

KEYWORDS

Egocentric videos, privacy, camera wearer recognition, hand gestures

ACM Reference Format:
Daksh Thapar, Aditya Nigam, and Chetan Arora. 2020. Recognizing Camera Wearer from Hand Gestures in Egocentric Videos: https://egocentricbiometric.github.io/. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413654

1 INTRODUCTION

Unlike point and shoot, hand held cameras that capture only when a user explicitly gives a command, egocentric cameras, due to their always-on nature, can potentially capture wearer's intimate private details as well. Hence, the video contributors and the research community have been careful in sharing egocentric videos captured in sterile environments such as kitchen, vacation tours, outdoor activities only. However, there is a massive oversight regarding other ways in which such videos can allow the wearer's privacy breach.

It has been shown by Hoshen and Peleg[9], that it is possible to recognize a wearer from his/her egocentric videos. The result is significant since the wearer is usually never seen in his/her egocentric videos. However, a first person camera, by virtue of being tied to a wearer's head, also captures his/her gait profile, which is a well known biometric signature of a person. For finding the gait style Hoshen and Peleg have computed the optical flow from an egocentric video and simply trained a Convolutional Neural Network classifier to identify the wearer. They have demonstrated their attack by showing an accuracy of 77% for recognizing a wearer in a dataset containing 32 subjects.

The attack proposed by Hoshen and Peleg has one significant restrictive assumption. It requires an attacker to capture many egocentric videos of the wearer while walking. Hence, a cursory look at the finding may suggest that there is no implication on the EPIC kitchens dataset. There are no walking activities in the dataset, and the creators have made sure that there are no reflective surfaces present in the scene, which may help others to identify the wearers. However, a more in-depth look reveals that even in this case, the egocentric videos still capture the wearer's hands and the way the person handles objects, or how he/she executes a particular action.

The focus of this paper is on exploring, if hand gestures visible in an egocentric video may also reveal any wearer identifying information. Our work aims to answer the following specific questions: (1) Given egocentric videos of various people, performing a specific activity, can we identify ($1:N$ matching) the camera wearer using only their hand gestures? (2) Given two anonymous videos picked from the public video-sharing dataset performing a specific activity, can we verify ($1:1$ matching) the claim that both the videos belong to the same camera wearer or not? Further, we would like to understand both of the above objectives in two sub-settings, (a) when we have the same activity as the query activity, but maybe in a different context or background, performed by the wearer available in the gallery, or (b) when we do not have the same but some other activity performed by the wearer in the gallery.

We propose the following approach to mount a hand gesture based wearer recognition attack in this paper. We compute the dense optical flow from the egocentric videos and use it for the wearer recognition. Optical flow choice is important so that the proposed model does not over-fit any appearance-based similarity from the background, handled objects, or the wearer's hands. For the first objective of wearer recognition, we formulate the problem as a multi-class classification problem. The training data available in the publicly available egocentric datasets are small and leads to severe over-fitting when training a deep neural network model from scratch. Hence, we propose a 2-stream model using 3D-CNN models (C3D[26] or I3D [2]) pre-trained over huge sports-1M dataset [12] and LSTM [8] or BiLSTM [20], and fine-tune the model using optical flow computed from egocentric videos. We also utilize the hand masks extracted from input video frames. These act as attention regularization for learning the behavioral pattern from optical flows.

For the second objective of wearer verification, we formulate the problem as learning a distance metric using triplet loss. This maps a video to a point in the embedding space as defined by the network. The training process ensures that the embedding distance between the videos from the same subject is closer as compared to a video from a different wearer. The training utilizes online semi-hard negative mining with a dynamic adaptive margin. To avoid over-fitting, we first fine-tune the proposed network for the classification as described above, and then train the network for distance metric in an end-to-end fashion.

We have used EPIC kitchens dataset [3] for training as well as demonstrating the proposed attack. Figure 1, shows two activities performed by two different wearers from the dataset. It can be observed that there is a vast amount of background variation both in inter as well as intra-subject activities. However, the proposed network successfully recovers the wearer specific behavioral characteristics and correctly identifies the actual wearers.

Contributions: The key findings and the specific contributions of this paper are as follows:

(1) To the best of our knowledge, ours is the first work demonstrating that hand gestures based behavioral features can be extracted from the optical flow in an egocentric video. We believe that the results of our study hold important implications for the safe public sharing of egocentric videos.
(2) For the closed-set settings, when the camera wearers are known at the training time, we report an accuracy of 70% in recognizing a wearer from a set of 28 subjects in the EPIC kitchens dataset.
(3) We show that hand gestures are unique across the activities. We train the proposed model leaving a particular activity, and then try to recognize a wearer using the left activity at the test time. We achieve an equal error rate of up to 18.72% in this scenario.
(4) Taking the threat level a step higher, we demonstrate that it is possible to recognize an uncooperating or anonymous wearer. In the open-set settings, when we do not have any video of a wearer available at the train time, our model can still verify that the two videos are coming from the same wearer with an equal error rate of 15.28%. We demonstrate this by using videos from 14 subjects in EPIC kitchens dataset for training and then using the remaining 14 subjects for the verification test.

2 RELATED WORK

First Person Recognition from Third Person Camera: There have been techniques that assume the presence of another third-person camera (wearable or static) present simultaneously to the egocentric camera and aims to identify the camera wearer in the third person view. In [4], the authors exploit multiple wearable cameras sharing fields-of-view to measure visual similarity and identify the target subject. Whereas, in [1], the common scene observed by the wearer and a surveillance camera has been used

to identify the wearer. Other works compute the location of the wearer directly [7, 15] or indirectly (using gaze, social interactions, etc.) [16, 17], which is then used to identify the wearer. Poleg et al. [18] identifies the wearer based upon the similarity in head motion established from first person optical flow and tracking the subject's head in the third person video. They have observed that the head motion gets embedded in the scene captured from the egocentric camera. They suggested sharing the averaged out optical flow of the egocentric videos to be used as a wearer's signature. They assume that the averaged out optical flow signatures does not reveal any identifying information about the camera wearer. Yonetani et al. [28] used the similar signatures as Peleg et al. to identify the wearer based upon the motion correlation over supervoxel hierarchies in head motion established from first person optical flow and tracking a subject's head in the third person video. Yagi et al. [27] has used the egocentric videos for predicting the future location of target people visible in the egocentric videos based on pose and scale of the person.

First Person Recognition from Wearable Sensors: Tao et al. [24] have shown that the gait features could also be captured from wearable sensors like accelerometer and gyroscope. As these sensors capture the movement of the body, which is caused by the movement of the legs, one indirectly captures the gait signature. Our work hypothesizes that similar to gait pattern while walking, hand gestures of a person while performing a particular activity also follow a behavioral pattern like gait. Since these gestures are visible in egocentric videos, they can be used to identify the camera wearer.

First Person Recognition from Egocentric Videos: There have been relatively fewer works on recognizing the camera wearer or his/her attributes from egocentric videos. Hoshen and Peleg [9] have shown that the identity of the camera viewer can be extracted from his/her egocentric video. They computed block-wise optical flows from the given egocentric video and trained a small CNN for camera wearer classification. Finocchiaro et al. [6] estimated the height of the camera from the ground using only the egocentric video without any intrinsic camera information. They have used a 2-stream CNN based regression model, which regresses the height of the camera wearer from the given input RGB video and its derived optical flows. The authors have achieved a mean average error of 14.04 cms over a range of 103 cms of data.

Other Related First Person Video Analysis Tasks: In other related works, which do not target wearer recognition, Jian and Graumann [10], have proposed to infer the wearer's pose from the egocentric camera. They have given a dynamic programming and learning-based approach that gives the full body 3D joint positions of the wearer in each frame. The technique uses both the optical flow as well as static scene structures to reveal the viewpoint (e.g., sitting vs. standing). Moreover, many works have proposed the use of hand gestures captured in egocentric videos for various computer vision tasks. Singh et al. [22] extracted trajectory aligned features for extracting salient hand gestures features for activity recognition in egocentric videos. Sun et al. [23] has used the hand gestures and object detection for predicting the actions that the camera wearer might perform in the next few frames. Kapidis et al. [11] have used the hand gestures and gaze estimation for multi-task learning of the action recognition task.

3 PROPOSED METHODOLOGY

As described in the introduction, we use optical flow, as observed in the egocentric videos, to train our proposed wearer recognition model. We have trained our model for the two tasks: wearer classification, and open set wearer recognition. Below we describe the architecture of our model as well as the details of the proposed training routine.

3.1 Pre-processing

The first step in the proposed methodology is to pre-process the videos. The videos have been sub-sampled into segments of 1 second each. Each video segment is converted to a frame rate of 15 frames per second. We also resize each frame to a size of $128 \times 128 \times 3$.

Optical Flow: We compute dense optical flow between each consecutive frame using Gunner Farneback's algorithm [5]. Hence, for each frame, we get $128 \times 128 \times 3$ dimensional optical flow matrix, where the first two channels depict the flow at each pixel in x and y directions. We pre-compute the magnitude of flow at each pixel and store it as the third channel in the optical flow matrix.

Hand Mask: Since the egocentric videos contain various challenges like rapid changes in illuminations, significant camera motion, and complex hand-object manipulations, there is a need to assist the proposed model in looking at the right areas and features. We compute hand masks from the egocentric videos and use them to regularize the network learning by using them as an attention mask. Li and Kitani [13] have proposed a hand mask extraction module using a collection of regressors indexed by a global color histogram. We follow their approach and use a bank of 48 Gabor filters (eight orientations, three scales, both real and imaginary components) to capture local textures. The posterior distribution of a pixel x given local appearance feature l, and global appearance feature g, is computed by marginalizing over different scenes c. The posterior distribution and can be approximated as:

$$\mathbb{P}(x \mid l, g) = \sum_{c} \mathbb{P}(x \mid l, c)\, \mathbb{P}(c \mid g), \tag{1}$$

where $\mathbb{P}(x \mid l, c)$ is the output of discriminative global appearance-specific regressor and $\mathbb{P}(c \mid g)$ is the conditional distribution of a scene c given global appearance features g. We use the pre-trained discriminative global appearance-specific regressor [21] for computing $\mathbb{P}(x \mid l, c)$ and the conditional $\mathbb{P}(c \mid g)$ is approximated using a uniform distribution over the five nearest global features as computed in [21]. The distribution $\mathbb{P}(x \mid l, g)$ provides the probability of each pixel in a given image, whether it belongs to hand. Since the input frame is of the size $128 \times 128 \times 3$, we get a probability matrix of 128×128. We quantize the output probabilities to $\{0, 1\}$ to create a hand mask. To make it compatible with the pre-trained networks, we have replicated each image's mask three times in the channels. Hence for each frame, we get a $128 \times 128 \times 3$ mask input.

Since each video segment consists of 15 frames, hence, for each video segment we get a optical flow input of size $15 \times 128 \times 128 \times 3$ and mask input of size $15 \times 128 \times 128 \times 3$.

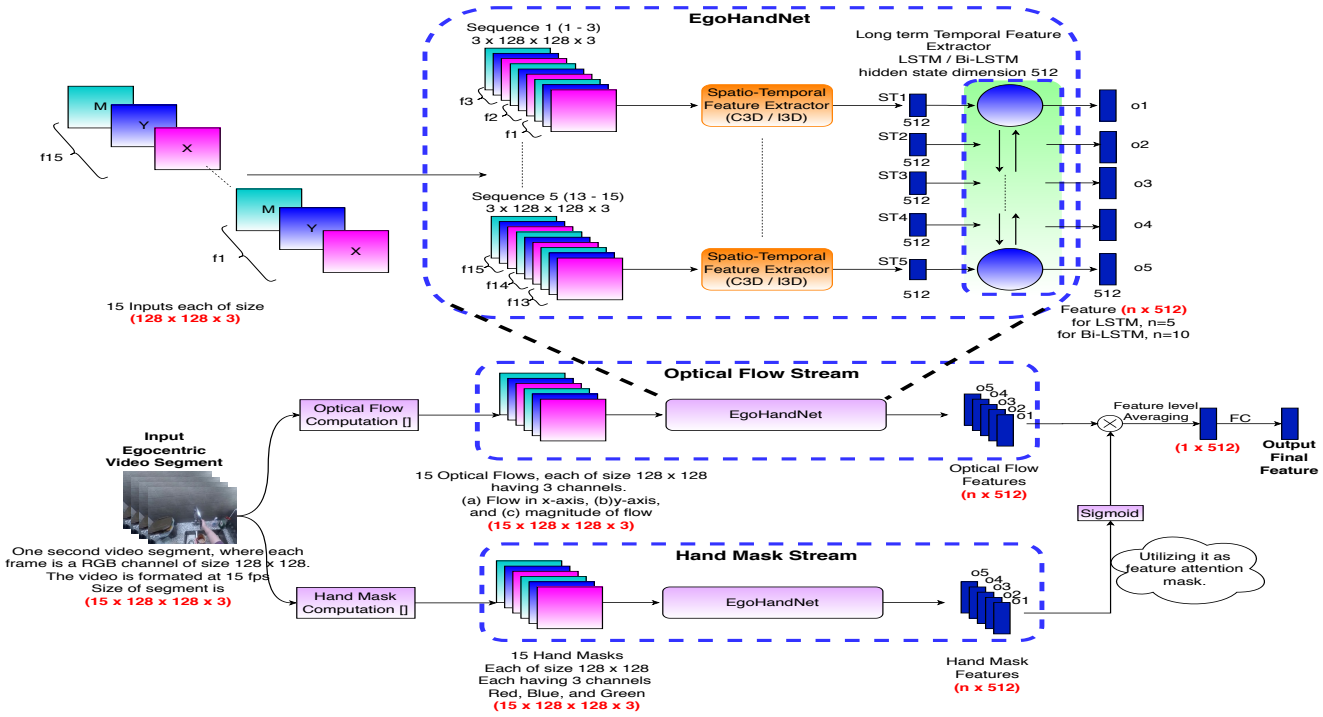

Figure 2: Proposed network architecture. Each 3D convolutional and fully connected layer is followed by a ReLU (except the final layer). The final layer has softmax activation for classification task, and L_2 normalization for the metric learning.

3.2 Network Architecture

As discussed above, the proposed model consists of 2 streams: optical flow and hand masks. Each stream gets a $15 \times 128 \times 128 \times 3$ input, and produces an output feature vector of $n \times 512$, where n is 5 or 10 depending upon the two experimented configurations described later in the section. Both the streams have identical architecture, called EgoHandNet hereon, as shown in Figure 2. Note that though the architecture is identical, the weights in the two streams are not tied.

The EgoHandNet itself consists of two modules, spatio-temporal feature extractor, and long term temporal feature extractor. The 15 frames given as input to the EgoHandNet module are further divided into five sub-sequences of 3 frames. Thus the size of each sequence is $3 \times 128 \times 128 \times 3$. We apply a 3D CNN based spatio-temporal feature extractor in order to extract a 512 dimensional feature for each of the sub-sequence. We have experimented with pre-trained C3D [26] or I3D [2] architectures for extracting theses spatio-temporal features (denoted as ST in Figure 2). Later, a long-term temporal feature extractor (LTT) is applied to learn long-term temporal dependencies between the previously extracted and five 512 dimensional *ST* features. For this purpose, we have experimented with LSTM and Bi-LSTM network architectures. Both the configurations have 512-dimensional hidden state representations. The, long term temporal feature extractor gives us an output of $n \times 512$ where $n = 5$ for LSTM, and 10 for Bi-LSTM configuration.

We apply EgoHandNet in both the optical flow and hand mask streams giving us a feature vector of $n \times 512$ from each stream. We use a Sigmoid layer on the hand mask features, to realize it as attention that can regularize the network learning. Later, it is element-wise multiplied with optical flow features. This enables the hand mask stream to learn suitable attention over the optical flow stream features, to extract the best possible masked optical flow features. We average the masked features, channel-wise, and temporally to give a 512 dimensional feature vector over which the final fully connected layer is applied.

3.3 Training Routine

For training the proposed network, we first pre-train each stream individually using the following step-wise training procedure:

(1) Fine-tune only the pre-trained spatio-temporal feature extractor. We experiment with C3D and I3D backbones.
(2) Freeze all the weights of the spatio-temporal feature extractor and only train the LSTM/Bi-LSTM for the task (classification or open-set verification).
(3) Fine-tune each stream individually in an end-to-end fashion for the task.
(4) Finally, fine-tune both the streams jointly, and the full network is trained for the task.

3.4 Loss Function for the Classification Task

To train our network for the classification task, we have used a 28 class (corresponding to the number of subjects in the Epic Kitchens dataset) classification layer as the final layer and have trained the network using softmax cross-entropy loss function. This loss function can learn inter-class variability but lacks to enforce intra-class

compactness. Moreover, it requires a lot of training data for each wearer. These issues can be solved by training the network with a triplet loss function, as suggested in [19] also. Besides handling inter-class variability and intra-class compactness, the loss helps us work with lesser data as well. Given a data having n data-points, when training with triplet loss, we can form $\binom{n}{3}$ triplet data points, thus bypassing the data scarcity problem in the egocentric videos.

3.5 Loss Function for Open Set Verification Task

To justify our claim of privacy breach more realistically, we propose an "open-set recognition setting instead". In the open set formulation, a suitable distance metric needs to be learned (trained only over the gallery data) that can differentiate between the signatures obtained from the video sequences of same wearers (genuine) and different wearer imposter (imposter). During testing, the learned metric is used to find the nearest gallery video sequence to any query sequence from our database.

For the open set verification formulation, our model's final layer is changed to a fully connected layer with 1024 neurons. This implies that, given a one-second clip of 15 frames, the network produces a feature embedding of 1024 dimensions. The embedding is then normalized onto a unit hypersphere centered at the origin.

Triplet Loss: For training the distance metric model, we have used the triplet loss as described in [19]. For a given video V, the network produces an embedding $\theta(V)$, such that given two videos V_x and V_y, if V_x and V_y belong to the same subject, then $\mathcal{E}(\theta(V_x), \theta(V_y)) = 0$, otherwise, $\mathcal{E}(\theta(V_x), \theta(V_y)) \geq \beta$, where β is the margin, $\mathcal{E}(X, Y)$ is the Euclidean distance between embeddings X and Y.

To make the embeddings of the videos from same subject similar, we minimize the following loss:

$$\mathcal{L}_p(x, y) = \frac{1}{N} \sum_{i=1}^{N} \left(\theta_i(V_x) - \theta_i(V_y)\right)^2, \tag{2}$$

where N is the size of an embedding. On the other hand, to make the embeddings of different class videos as far as possible, we minimize the hinge loss between the embeddings as shown below:

$$\mathcal{L}_n(x, y) = \frac{1}{N} \sum_{i=1}^{N} \max\left(0, \beta - \left(\theta_i(V_x) - \theta_i(V_y)\right)^2\right). \tag{3}$$

Since both the tasks are needed to be accomplished simultaneously, we combine both the losses to form a single triplet loss. It is defined over a set of three embeddings, $\theta(V_a)$: embedding of a anchor video, $\theta(V_p)$: embedding of a video from the same subject as anchor, and $\theta(V_n)$: embedding of a video from some subject, not same as the one in anchor. The loss is expressed as:

$$\mathcal{L}(a, p, n) = \frac{1}{N} \sum_{i=1}^{N} \max\Big(0, \\ (\theta_i(V_a) - \theta_i(V_p))^2 - (\theta_i(V_a) - \theta_i(V_n))^2 + \beta\Big) \tag{4}$$

To efficiently train the network, we have used hard negative mining [19] and dynamic adaptive margin [25], as described.

Hard Negative Mining: While forming a triplet for training, choosing a suitable negative pair is a demanding job. Given the enormous amount of negative pairs, choosing them randomly will lead to easy triplets. The network will easily learn these triplets ignoring the challenging ones, degrading the network's performance. To avoid that, we need to choose the hard triplets and back-propagate losses only pertaining to them. A triplet is said to be "hard" when the distance between the embeddings of anchor and negative (d_n), and anchor and positive (d_p) is lesser than the margin (β). To compute such triplets, we have to calculate the embeddings of each video in the dataset before making every batch, which can be a cumbersome task. Hence, while creating a batch, we randomly choose 1000 triplets, compute d_n and d_p for each triplet and only choose those whose $d_n - d_p \leq \beta$ for batch making.

Dynamic Adaptive Margin: Another training challenge is the diminishing number of hard triplets as "hardness" is defined w.r.t. the margin β. As the training process progresses, the number of hard triplets reduces for a fixed margin. This behavior can be accounted for during the selection process of triplets. As the model trains, the embeddings of the anchor, positive and negative differ with every epoch such that d_p decreases and d_n increases. A triplet that was considered "hard" in the initial epochs may not be considered the same in the later epochs as the distance between their embeddings varies continuously. To overcome this problem, we need to vary the margin adaptively, starting from some low value and increasing it once the "hard" triplet becomes less hard. We have increased our margin by a step of 0.05 when the number of "hard" triplets mined is less than a threshold of 50 for 3 consecutive batches.

↓Test	Genuine Matching					Imposter Matching				
Train→	cut	mix	put	take	wash	cut	mix	put	take	wash
cut	3.3	2.1	5.1	2.2	2.9	0.52	0.24	1.07	0.45	0.75
mix	1.2	1.6	1.7	0.8	1.2	0.27	0.12	0.56	0.24	0.39
put	5.5	3.2	12.1	4.4	7.1	1.23	0.57	2.47	1.06	1.73
take	2.9	1.2	3.8	3.1	3.1	0.48	0.23	0.99	0.41	0.68
wash	5.1	1.4	8.4	4.3	6.9	1.08	0.52	2.21	0.94	1.53

Table 1: Number of genuine (at the scale of 10^5) and imposter matching (at the scale of 10^7) considered for EPIC kitchens dataset. One second clip a data input. [The table serves to indicate the complexity of the proposed recognition task]

4 EXPERIMENTS AND RESULTS

In order to justify our hypothesis, we have validated our proposed approach by performing rigorous experimental analysis. The initial task is to discover the optimal network architecture systematically. Hence, we have performed an extensive ablation study over the architecture of the optical flow stream. Recall that the same architecture has been used for both streams of the proposed model.

4.1 Dataset Specifications

In this work, we have used the EPIC kitchens dataset [3]. It consists of 55 hours of egocentric video comprising of 32 subjects. The dataset has labeled 125 different activities performed by the subjects. Considering the scope of this work, we have chosen activities that

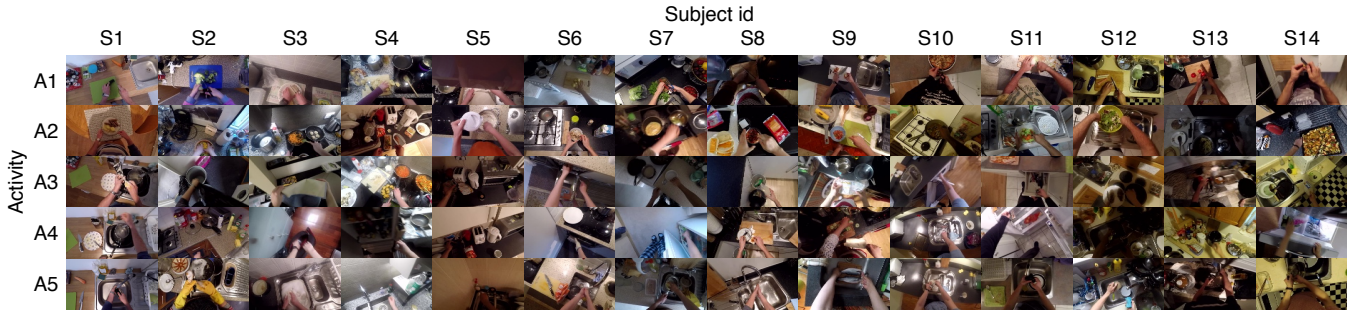

Figure 3: Shown 14 subjects performing 5 different activities. Row corresponds to one activity and column represents a subject.

↓Tested on Trained on→	[C3D - Pipeline] M1 = C3D, M2 = C3D-LSTM, M3 = C3D-BLSTM															
	cut			mix			put			take			wash			
	(in%)	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3
cut	EER	14.73	13.37	13.11	19.19	18.46	17.79	18.42	17.32	16.56	16.54	15.72	15.61	16.87	15.75	15.14
	CRR	59.07	62.50	64.75	48.99	52.42	54.26	56.33	59.69	59.29	49.17	53.81	56.68	56.26	56.35	56.84
mix	EER	14.04	12.33	11.50	10.43	10.70	10.79	27.29	24.89	24.97	8.07	7.26	6.99	15.71	14.77	14.82
	CRR	64.15	65.45	68.72	67.80	69.76	73.87	64.04	64.73	66.01	63.98	64.42	62.95	65.68	67.73	71.24
put	EER	14.78	14.30	13.39	18.84	17.38	16.78	18.69	18.32	17.87	14.09	12.86	13.44	13.18	13.16	12.82
	CRR	50.78	55.34	59.95	35.33	35.39	39.31	59.90	59.99	62.24	57.75	60.33	63.51	60.47	63.85	64.52
take	EER	21.57	20.54	19.94	24.15	22.58	22.31	18.73	18.30	17.39	22.51	21.48	21.13	21.56	20.42	20.27
	CRR	51.41	53.13	55.68	59.61	60.50	62.35	62.57	62.86	63.80	50.27	51.09	51.28	56.98	58.37	59.24
wash	EER	15.02	13.15	12.93	20.33	18.47	17.05	19.93	18.90	18.13	12.34	11.71	10.50	22.50	21.69	21.16
	CRR	52.32	54.61	56.76	52.07	53.68	55.41	54.50	56.39	57.36	57.69	58.72	59.88	55.57	57.42	58.67
	[I3D - Pipeline] M4 = I3D, M5 = I3D-LSTM, M6 = I3D-BLSTM															
		M4	**M5**	**M6**	**M4**	**M5**	**M6**	**M4**	**M5**	**M6**	**M4**	**M5**	**M6**	**M4**	**M5**	**M6**
cut	EER	14.02	12.81	**12.05**	**18.16**	18.31	18.40	18.23	16.69	**16.15**	**15.20**	15.73	15.70	15.84	14.72	**13.47**
	CRR	60.44	63.71	**64.95**	49.45	**54.89**	54.39	54.09	59.56	**61.76**	52.43	**54.84**	53.41	55.71	59.50	**59.89**
mix	EER	14.24	**11.42**	11.51	11.76	11.01	**10.01**	28.27	**24.55**	24.59	9.11	7.13	**6.67**	14.67	14.60	**14.04**
	CRR	66.39	67.98	**68.85**	67.92	70.46	**72.50**	61.73	64.20	**68.41**	62.87	62.59	**63.43**	65.54	70.32	**70.66**
put	EER	14.01	**13.74**	13.92	18.77	16.44	**16.24**	19.60	18.21	**17.53**	14.96	12.62	**12.12**	13.45	**12.87**	12.91
	CRR	51.96	58.46	**60.31**	36.17	39.38	**39.47**	60.53	62.94	**63.82**	56.16	**62.27**	61.57	60.27	64.91	**65.17**
take	EER	20.82	19.48	**19.23**	23.89	22.02	**21.31**	18.54	17.72	**16.84**	22.19	21.01	**20.71**	20.83	19.76	**19.69**
	CRR	52.81	55.95	**56.14**	59.95	62.08	**62.47**	62.67	62.94	**63.93**	50.84	50.91	**51.87**	57.06	60.07	**60.21**
wash	EER	14.73	12.68	**12.67**	19.01	16.56	**16.52**	19.42	18.51	**17.69**	12.23	10.22	**10.09**	21.74	20.69	**20.22**
	CRR	53.55	56.37	**57.23**	52.91	55.65	**55.72**	55.23	56.82	**57.40**	58.28	60.03	**60.34**	56.41	58.61	**58.99**

Table 2: Ablation study for exploring optimal network architecture selection. Only optical flow stream is trained as Siamese. For each experimentation, the best results achieved are highlighted in bold font.

follow the following two criteria: (1) Duration of performing the activity should be greater than a second. This ensures that we have enough amount of data to extract the behavioral pattern of a person. (2) Since the activities could be of variable length, at least 60% or 500 instances of the activity must be more than a second. This ensures that we have enough data points. Based on the above criteria, we have chosen five activities viz *cut, mix, put, take*, and *wash*.

4.2 Training and Testing Protocol

While training the classifier for the closed-set verification and activity level open-set verification, the first 50% videos for each subject are selected for training (gallery samples) while the remaining half are used for testing (probe samples). Similarly, while training for subject-level open-set verification, all the data of the first half subjects for each activity have been taken for training. The remaining samples belonging to unseen subjects have been taken for testing the system. Figure 3 shows examples of frames corresponding to 14 subjects performing the above mentioned five activities, viz *cut, mix, put, take*, and *wash*.

Table 1 shows the data specifications for each of the different activities. One can observe that the dataset contains labeled activities only for 28 subjects. Hence we have used only those subjects. For the activity *mix*, out of 28 subjects, data belonging to 12 subjects is

entirely missing. Hence, the number of subjects for *mix* activity is only 16.

4.3 Performance Evaluation Metrics

In order to validate the proposed network, we have performed two types of analysis, (a) classification followed by (b) verification in closed and open-set scenarios. The standard performance metrics used for both are described below:

Classification: We report the percentage of correctly classified data-points (probe samples), i.e., accuracy (%) for each activity.

Verification: The standard metrics viz, Equal Error Rate, and Correct Recognition Rate have been computed. (1) The Equal Error Rate (EER) is defined as the percentage error achieved at the matching threshold, where False Reject Rate and False Acceptance Rate becomes equal. Lower EER signifies a better performance. (2) Correct Recognition Rate (CRR) is defined as the percentage of correctly identified data-points (probe samples) with respect to the total number of data-points, which is rank-1 accuracy.

4.4 Ablation Study for Model Selection

We have performed a rigorous ablation study to choose the appropriate architecture for our model. We have conducted the study only on the optical flow stream. Further, verification training has been done only under a close-set scenario. We have made six different configurations, namely: (1) $M1$: Model having only C3D [26] network. (2) $M2$: Model having C3D and LSTM [8], (3) $M3$: Model having C3D and Bi-LSTM [20], (4) $M4$: Model having only I3D [2] network, (5) $M5$: Model having I3D and LSTM, and (6) $M6$: Model having I3D and Bi-LSTM. In order to use the hand masks cues in the ablation study, instead of fusing the hand-mask feature at the end, as is done for the final model, we mask the optical flow at the input itself using the hand mask. This has been done only for the ablation study.

Table 2 presents the results. One can observe that C3D being the simplest of 3D CNN architectures, forms the baseline for this task. Whereas I3D extracts multi-scale spatio-temporal optical flow features, it can boost the performance by 1-2% over C3D. This is clearly observed from the performance gain in M4 over M1. Moreover, using the LSTM layer over C3D or I3D features gives a major advantage as neither C3D nor I3D can efficiently capture long term temporal dependencies essentially required for behavioral pattern extraction. Bi-LSTM marginally improves performance by capturing long term temporal dependencies in both forward and backward directions. Finally, model $M6$ emerges as the best performing model utilizing I3D as the spatio-temporal feature extractor and Bi-LSTM as a temporal feature extractor. We use Dual-Stream-$M6$ ($DS - M6$) model for all further analysis.

4.5 Performance Analysis

Classification: We have analyzed our final $DS - M6$ as a classification model for wearer recognition. The classification accuracy is computed over various activities individually. The comparative analysis of various activities is given below:

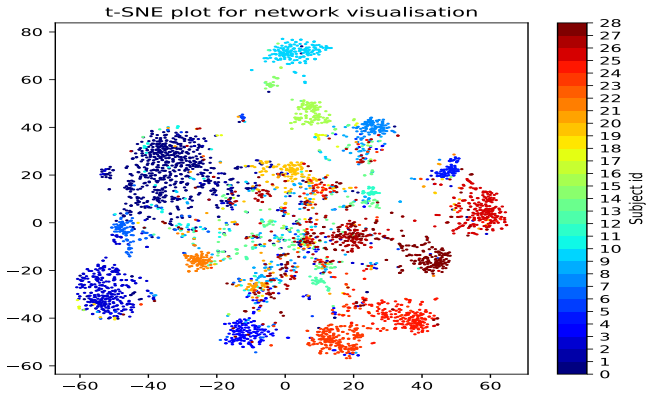

Figure 4: The t-SNE [14] plots for all 28 subjects of the feature maps extracted from last layer of proposed $DS - M6$ model. Same colour denotes feature corresponding to same subject.

Activity	cut	mix	put	take	wash
Accuracy (%)	50.61	60.09	52.43	48.21	53.80

It is important to note that since we have 28 subjects, the chance level accuracy is only 3.57%. One can observe from the table that except for *mix* activity, the classification performance for the other four activities lie in the range of 48% to 53%. Comparing these with chance level performance, it is evident that the proposed model can classify the egocentric camera wearer using hand gestures. The performance for *mix* activity comes out to be better than others, due to its cyclic periodicity supporting learning better discrimination.

Close-set Verification: To analyze the proposed methodology for the verification task, all the subjects were used for training and testing. We have trained our $DS - M6$ model (as a Siamese) for verification over each of our five activities simultaneously. While testing, subject verification has been done by matching wearers videos performing each activity with every other activity, e.g., finding out the subject id of the nearest sample of *wash* activity (in the gallery) to the given probe sample of *mix* activity. This generates 25 combinations, as shown in Table 3. We report the equal error rate (EER) and correct recognition rate (CRR) for each such combination. One can observe that the verification error of matching activity with itself (diagonal elements in the table) is low, validating that our model can verify a subject for a known activity easily. Moreover, the CRR for self-activity matching is significantly better than the classification accuracy of the network. This shows that triplet loss function performs better training when data has a high amount of intra-class variation and when the data is scarce.

Performance gain ($DS - M6$): One can observe from tables 2 and 3, that training and testing the single-stream $M6$ model directly over optical flows provides an average CRR of 55.86% (overall the subjects performing all activities). It involves a total of $108,639$ testing data points over which the recognition has been done. After training and testing, the same $M6$ model over masked optical flows achieves an average CRR of 61.06% (with a gain of 5%). Finally, training and testing the end-to-end Dual Stream ($DS - M6$) model supersedes both of them by attaining an average CRR of 62.9% (gain

↓Tested on Trained on→	(in%)	cut	mix	put	take	wash
	EER	11.35	17.69	16.04	14.93	13.09
cut	CRR	65.20	57.58	62.50	53.51	60.76
	EER	10.72	9.84	24.19	6.25	13.06
mix	CRR	69.42	73.35	69.26	64.29	71.38
	EER	18.26	20.52	16.39	19.93	19.47
put	CRR	57.97	62.70	65.32	52.11	61.18
	EER	11.92	16.08	16.74	9.74	19.77
take	CRR	59.31	56.16	57.80	61.59	60.03
	EER	13.15	16.07	17.25	11.42	12.88
wash	CRR	62.28	41.09	64.69	62.33	65.71

Table 3: Close-set verification accuracy. The $DS - M6$ model is trained (as Siamese) over each activity simultaneously and tested by matching it with all other activities.

↓Tested on Trained on→	(in%)	cut	mix	put	take	wash
	EER	-	25.62	24.75	23.53	23.03
cut	CRR	-	46.40	51.27	45.09	49.60
	EER	19.54	-	31.59	18.72	21.75
mix	CRR	58.61	-	58.84	56.21	61.54
	EER	25.57	29.23	-	27.53	28.35
put	CRR	49.28	52.84	-	45.60	50.98
	EER	19.17	24.13	25.25	-	31.66
take	CRR	51.47	48.40	49.64	-	52.37
	EER	21.01	24.93	25.04	19.20	-
wash	CRR	51.07	32.26	53.75	50.31	-

Table 4: Open-set verification: The $DS - M6$ model is trained by leaving one activity at a time, but matching that activity with each of the other training activities (leave one activity out).

of 7% over direct optical flow and 2% over masked optical flow with 7, 605 and 2, 173 more correct classifications respectively).

Activity Level Open-set Verification: This analysis is done to validate if our model recognizes a person performing some new activity at the test time. We train our $DS - M6$ model in a leave-one-out manner at the activity level. That is, the activity on which the network has to be tested was left out during the training. During testing, the left-out activity has been matched with all the training activities to recognize the wearer. The results are shown in Table 4, reporting the equal error rate (EER), and correct recognition rate (CRR) for this task. It can be seen that while testing on unseen activities, the performance drops as compared to close-set verification by around 10%. This reinforces that the proposed network is not over-fitting over any activity. Moreover, it strengthens the claim of a unique personalized behavioral hand gesture pattern.

Subject Level Open-set Verification: We perform this experiment to understand if our model can recognize that the two activities seen at the test time have been performed by the same subject, even when none of the activities by this subject were seen at train time. For this experiment, we have trained our $DS - M6$ model on first half of the subjects and tested it on the rest half of the subjects. It is trained (as Siamese model) for verification on each activity and then tested by matching each activity sample with itself for wearer recognition. The results of subject-level open-set analysis are:

Activity	cut	mix	put	take	wash
EER (%)	19.25	16.59	21.17	15.21	20.77
CRR (%)	59.89	66.32	58.10	52.49	57.25

Comparing these with closed set results of Table 3, it is evident that there is only a minor decrease in performance, proving its generalization over unseen subjects. This further reinforces the claim of "Privacy Threat" established using hand gestures in egocentric videos can be scaled to uncooperative and anonymous wearers.

Feature Visualization: As a qualitative analysis of the proposed model, we have generated the t-SNE [14] plot of the feature vectors belonging to 28 different subjects performing all five activities and extracted from the last layer of $DS - M6$ model. Figure 4 shows the result. The same color denotes features corresponding to the same subject. One can observe that despite having vast intra-class variability for a particular subject, the network achieves intra-class compactness. Further, even though there is some overlap between the classes, the inter-class variability is still evident from the plot.

5 CONCLUSION

In this paper, we show that one can identify an egocentric camera wearer (with an accuracy as high as 70%) using only the hand gestures. This enables us to recognize a camera wearer even when the wearer is neither visible nor walking and can be considered as a privacy breach in the publicly available benchmark egocentric datasets such as EPIC kitchens, FPSI, etc. Our experiments validate that one can extract behavioral features from the hand gestures as visible in the egocentric videos. The proposed 2-stream model ($DS - M6$) can learn behavioral patterns using only input optical flow, whereas second streams provide an attention-based regularization from hand masks extracted from input video frames. We have demonstrated that such discriminative features can be used successfully to match a person when performing some specific activity using their hands. To justify the generalization of these features, we perform open-set verification at the subject and activity level. The obtained results clearly establish the feature robustness by identifying camera wearer for an unseen activity and unseen subject (up-to-a accuracy of 66%).

ACKNOWLEDGMENTS

This work was supported in part by the DST, Government of India, under project id T-138.

REFERENCES

[1] Shervin Ardeshir and Ali Borji. 2016. Ego2top: Matching viewers in egocentric and top-view videos. In *European Conference on Computer Vision*. Springer, 253–268.

[2] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *European Conference on Computer Vision (ECCV)*.

[4] Chenyou Fan, Jangwon Lee, Mingze Xu, Krishna Kumar Singh, Yong Jae Lee, David J Crandall, and Michael S Ryoo. 2017. Identifying first-person camera wearers in third-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5125–5133.

[5] Gunnar Farnebäck. 2003. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*. Springer, 363–370.

[6] Jessica Finocchiaro, Aisha Urooj Khan, and Ali Borji. 2017. Egocentric height estimation. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1142–1150.

[7] Joel A Hesch and Stergios I Roumeliotis. 2012. Consistency analysis and improvement for single-camera localization. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 15–22.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] Yedid Hoshen and Shmuel Peleg. 2016. An egocentric look at video photographer identity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4284–4292.

[10] Hao Jiang and Kristen Grauman. 2017. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 3501–3509.

[11] Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp. 2019. Multitask Learning to Improve Egocentric Action Recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[13] Cheng Li and Kris M Kitani. 2013. Pixel-level hand detection in ego-centric videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3570–3577.

[14] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[15] Ana Cristina Murillo, Daniel Gutiérrez-Gómez, Alejandro Rituerto, Luis Puig, and Josechu J Guerrero. 2012. Wearable omnidirectional vision system for personal localization and guidance. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 8–14.

[16] Hyun S Park, Eakta Jain, and Yaser Sheikh. 2012. 3d social saliency from head-mounted cameras. In *Advances in Neural Information Processing Systems*. 422–430.

[17] Hyun Soo Park, Eakta Jain, and Yaser Sheikh. 2013. Predicting primary gaze behavior using social saliency fields. In *2013 IEEE International Conference on Computer Vision*. IEEE, 3503–3510.

[18] Yair Poleg, Chetan Arora, and Shmuel Peleg. 2014. Head motion signatures from egocentric videos. In *Asian Conference on Computer Vision*. Springer, 315–329.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[20] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[21] Suriya Singh, Chetan Arora, and CV Jawahar. 2016. First person action recognition using deep learned descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2620–2628.

[22] Suriya Singh, Chetan Arora, and CV Jawahar. 2017. Trajectory aligned features for first person action recognition. *Pattern Recognition* 62 (2017), 45–55.

[23] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Rahul Sukthankar, Kevin Murphy, and Cordelia Schmid. 2019. Relational action forecasting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 273–283.

[24] Weijun Tao, Tao Liu, Rencheng Zheng, and Hutian Feng. 2012. Gait analysis using wearable sensors. *Sensors* 12, 2 (2012), 2255–2283.

[25] Daksh Thapar, Gaurav Jaswal, Aditya Nigam, and Vivek Kanhangad. 2018. PVSNet: Palm Vein Authentication Siamese Network Trained using Triplet Loss and Adaptive Hard Mining by Learning Enforced Domain Specific Features. *arXiv preprint arXiv:1812.06271* (2018).

[26] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[27] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. 2018. Future person localization in first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7593–7602.

[28] Ryo Yonetani, Kris M Kitani, and Yoichi Sato. 2015. Ego-surfing first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5445–5454.