CatNet: Class Incremental 3D ConvNets for Lifelong Egocentric Gesture Recognition

Zhengwei Wang[1]*, Qi She[2]†, Tejo Chalasani[1], and Aljosa Smolic[1]

[1]V-SENSE, Trinity College Dublin
{zhengwei.wang,CHALASAT,SMOLICA}@tcd.ie

[2]Intel Labs China
qi.she@intel.com

Abstract

*Egocentric gestures are the most natural form of communication for humans to interact with wearable devices such as VR/AR helmets and glasses. A major issue in such scenarios for real-world applications is that may easily become necessary to add new gestures to the system e.g., a proper VR system should allow users to customize gestures incrementally. Traditional deep learning methods require storing all previous class samples in the system and training the model again from scratch by incorporating previous samples and new samples, which costs humongous memory and significantly increases computation over time. In this work, we demonstrate a lifelong 3D convolutional framework – c(**C**)la(**a**)ss increment(**t**)al net(**Net**)works (CatNet), which considers temporal information in videos and enables lifelong learning for egocentric gesture video recognition by learning the feature representation of an exemplar set selected from previous class samples. Importantly, we propose a two-stream CatNet, which deploys RGB and depth modalities to train two separate networks. We evaluate CatNets on a publicly available dataset – EgoGesture dataset, and show that CatNets can learn many classes incrementally over a long period of time. Results also demonstrate that the two-stream architecture achieves the best performance on both joint training and class incremental training compared to 3 other one-stream architectures. The codes and pre-trained models used in this work are provided at* `https://github.com/villawang/CatNet`.

1. Introduction

With development and popularity of VR/AR devices recently, there is an increasing demand to work with these devices intuitively. Gestures are the most natural form for humans to interact with such type of devices, in which hand gestures can be conveniently captured by cameras integrated in the devices in first person view. This motivates accurate recognition of meaningful gestures from such egocentric gesture videos.

Video recognition systems for such VR/AR applications in the real world should ideally be designed in a way to support incremental update and customization of gestures. Different communicative gestures should be customized for different VR games [45]. Traditional machine learning/deep learning approaches require training data of all classes accessed at the same time, which is hardly achievable in such real-world situations. For instance, when a new gesture should be added to a system, the model needs to be retrained by incorporating the gesture video samples of previous and new classes, which requires significant memory for storing all previous class videos and increasing computational cost over time. A system with capability of lifelong learning would therefore be very beneficial for such scenarios, in which incremental learning makes use of memory efficiently, enables fast learning for new class samples and does not forget the previous class samples. In this work, we demonstrate a c(**C**)la(**a**)ss increment(**t**)al net(**Net**)works (CatNet) for an open-set problem rather than a close-set problem, which learns new classes i.e., the class variants larger than instance variants.

Hand-crafted features are commonly adopted in traditional video gesture recognition [23, 26, 39]. With more large-scale datasets being released and development of deep neural networks (DNNs), DNNs are playing a more and more important role in this field [21, 5]. Different from image recognition, temporal information along each frame needs to be considered for video understanding. The 3D convolutional network (ConvNet) becomes a popular architecture for learning spatiotemporal features from video clips. Benefiting from large-scale video datasets being released [19, 4, 2], deep 3D ConvNets have achieved striking results in video action recognition tasks [16, 6, 9]. Com-

*This work is finacially supported by Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776.

†Corresponding author

pared to current popular video action datasets e.g., UCF-101 [36], Kinetics [19], egocentric gesture video is in first person view, in which two modalities RGB and depth can be captured at the same time. This indicates more information can be used to train the models for the egocentric gesture video recognition. Two-stream 3D ConvNets [35] is proposed for video action recognition by using optical flow [10] in addition to RGB frames but optical flow is difficult to compute and to use for large-scale datasets [9]. We evaluate our models on a recently released large-scale egocentric gesture video dataset named EgoGesture [46], in which RGB and depth modalities are provided. Benefiting from RGB-D video, we propose a two-stream architecture that deploys RGB and depth as two streams for egocentric gesture video recognition in this work, which deals the inconsistent quality of RGB and depth frames (see Figure 1) across different scenes (6 different scenes are included in the dataset) during the recording. Figure 1 shows two gesture examples in two scenes respectively i.e., in a walking state with a dynamic background on the left and in a stationary state facing a window with drastically changing sunlight on the right. It can be noticed that the quality of the RGB input and the depth input are not consistent i.e., walking and outdoor capture can result in poor depth data, while illumination changes from changing sunlight can affect distribution of RGB pixels. Fusing features produced by a two-stream architecture can mitigate this issue, which results in a better overall performance. Previous work has shown that the frame-based approaches (e.g., VGG-16) are ineffective for the EgoGesture video recognition [46] because such methods do not take account into temporal information. Video-based approaches are required for accurate recognition in this scenario.

Figure 1: Visualization of gestures in different scenes. Left: The participant in a walking state with a dynamic background. Right: The participant in a stationary state facing a window with drastically changing sunlight.

Significant advances have been made recently in computer vision and deep learning tasks including object recognition, detection, segmentation, etc. However, most of the models can only be trained in a batch setting, in which training data of all object classes is required for training the model in a roll. Lifelong learning [27] is a strategy to enable training the model continuously. To overcome the issue addressed earlier in the context of egocentric video recognition, in which the system should be able to learn new gestures incrementally, we introduce a lifelong learning framework – c(**C**)la(**a**)ss increment(**t**)al net(**Net**)works (CatNet), which is specifically designed for lifelong egocentric gesture video recognition based on 3D convolutional networks (ConvNets). Importantly, we propose a two-stream CatNet using RGB and depth input as separate streams, which achieves the best performance in the class incremental learning task.

To summarize, our contribution are three-fold:

- To the best of our knowledge, we are the first to address the class incremental issues in the area of egocentric gesture video recognition and introduce the lifelong learning approaches to this area.
- We propose a two-stream CatNet for egocentric gesture video recognition, which treats RGB and depth as two separate streams and this type of CatNet is shown to perform best in the class incremental task.
- Our results show that CatNets can learn many classes incrementally over a long period of time i.e., the highest mean accuracy of presented CatNet has achieved 0.884.

2. Related Work

We introduce some recent literature with respect to video action recognition, EgoGesture video recognition and lifelong learning in this section.

2.1. Video Action Recognition

The success of convolutional networks (ConvNets) in object detection [30], object recognition [22], panoptic segmentation [24] tasks etc. has attracted growing interest for deploying them to other areas of computer vision. Video understanding has became a very popular research area recently, which is driven by several released large-scale datasets such as Kinetics [19], YouTube-8M [2], ActivityNet [4] and Sports-1M [18]. Unlike image tasks, video tasks require not only spatial information for each frame but also temporal information for neighboring frames, which poses a challenge for traditional methods performing on image tasks. Video understanding for untrimmed video datasets e.g., ActivityNet is still very challenging today because it requires to consider the possibility of accomplishing additional tasks such as untrimmed action classification and detection. Work discussed in this paper only considers the trimmed video scenario.

Many methods have been proposed for video action recognition by introducing temporal information to the model. 3D convolution has been firstly introduced in [17], which enables 3D convolutional networks (3D ConvNets) to extract features from both spatial and temporal dimensions. With the success in learning spatiotemporal information from consecutive frames by using 3D convolutional

modules, several 3D types of architectures have been proposed in this field e.g., I3D [6], P3D [28], T3D [9] and R3D [16]. The work in [16] addresses that it is important to use a pretrained model that is trained on a large-scale video dataset for a specific video task, which is able to avoid issues such as overfitting, difficult to converge and long time for training. The authors also demonstrate the efficacy of using R3D (use ResNet block as backbone for 3D convolution) for video action recognition, providing good performance and flexible architectures.

By using more than one modality for video action recognition, multimodal representation has achieved remarkable results [40, 35, 6, 11, 41]. A typical architecture is the two-stream ConvNet [35, 6], which uses RGB frames and optical flow [10] for training two separate networks. However, the computation of optical flow is very expensive, which limits its deployment in practice [9]. There are lots of depth cameras available on the market with acceptable price e.g., RealSense Camera SR300, which makes RGB frames and depth maps conveniently accessible for the egocentric-like datasets e.g., EgoGesture. In this work, we apply a two-stream 3D ConvNet by using RGB frames and depth frames, where the R3D is used as the backbone for our 3D ConvNet.

2.2. Egocentric Gesture Video Recognition

Datasets Like EgoGesture [5, 46], GreenScreen [7] pave the wave for end-to-end learnable DNN architectures to address large-scale egocentric gesture recognition problems. Cao *et al*. [5] propose a neural network architecture by using a 3D ConvNet in tandem with spatiotemporal transformer modules and a LSTM for recognizing egocentric gestures from trimmed egocentric videos. In their network design, conceptually 3D ConvNets calculate the motion features and STTMs compensate for the ego motion. Shi *et al*. [31] improve on this approach by replacing spatiotemporal transformer modules with spatiotemporal deformable modules to overcome the issue of non-availability of local geometric transformations.

Chalasani and Smolic [8] propose a different network architecture that extracts embeddings specific to ego hands which are calculated as output from their encoder and decoder based architecture, which simultaneously computes hand segmentation. The embeddings thus generated for each trimmed video are then used in LSTMs to discern the gesture present in the video.

In a different approach, Abavisani *et al*. [1] propose a training strategy to use knowledge from multi-modal data to get better performance on unimodal 3D ConvNets. Unlike Cao *et al*. [5], they train a separate network for each available modality and use a new spatiotemporal semantic alignment loss function, which they propose to share the knowledge among all the trained networks.

The scope for application of recognizing gestures from trimmed videos is limited. To address this issue, Köpüklü *et al*. [21] introduce a network architecture that could enable offline working CNN based networks to work online using a sliding window approach.

However, the idea of lifelong learning for egogesture recognition has not been explored in any of the mentioned papers. Given a new gesture, the entire network has to be trained with all the gestures starting the training process from the beginning, which becomes cumbersome as the number of gestures increases incrementally.

2.3. Lifelong Learning

Current state-of-the-art DNNs have achieved impressive performance on a variety of individual tasks. However, it still remains a substantial challenge for deep learning, which is learning multiple tasks continuously. When training DNNs on a new task, a standard DNN forgets most of the information related to previously learned tasks. This phenomenon is known as "catastrophic forgetting" [25].

There are three scenarios in the area of lifelong learning [38]: (1) Task incremental learning, where the task ID is provided during testing; (2) Domain incremental learning, where the task ID is not provided during testing and the model does not have to infer the task ID; and (3) Class incremental learning, the task ID is not provided during testing and the model has to infer the task ID. The first scenario is the easiest one and the model is always informed about which task is going to be performed. In this case, the model can be trained with task-specific components. A typical network for such a scenario can have a "multi-headed" output layer for each task and the rest of the model can be shared across tasks [38]. A typical example for the second scenario is that the environment is changing e.g., image background changes but the objects remain the same for an object recognition task. The model has to solve the task but does not infer how the environment changes [12]. The last scenario is the most challenging one which requires the model to infer each task. For example, the model has to learn new classes of objects incrementally in an object recognition task. In this work, we focus on the most challenging scenario – class incremental learning, where we address the importance for learning gesture classes incrementally regarding the egocentric gesture video recognition.

Catastrophic forgetting appears when the new instance is significantly different from previous observed examples. Current strategies such as replay of old samples [15, 29] and regularization [3, 14] can be deployed to mitigate this problem. FearNet was proposed in [20], where a generative neural network [27, 44, 43, 42] is used to create pseudo-samples that are intermixed with recently observed examples stored in its hippocampal network. PathNet [13] was proposed as an ensemble method, where a generic algorithm is used to

find the optimal path through a neural network of fixed size for replication and mutation. Ideally, the lifelong learning should be triggered by the availability of short videos of single objects and performed online on the hardware with fine-grained updates, while the mainstream of methods we study are limited with much lower temporal precision as our previous sequential learning models [32, 33]. In [29], iCaRL was proposed to cache the most representative samples from previous classes by using representation learning, which demonstrates good performance on class incremental learning. It is also easy to be extended to any type of network architectures. Benefiting from these advantages, we incorporate iCaRL into our CatNet to realize a lifelong learning system for egocentric gesture video recognition.

3. Methodology

In this section, we first elaborate on the type of 3D ConvNets investigated in this work, which is known as R3D. Then we present a two-stream 3D ConvNet for egocentric gesture video recognition (EgoGesture dataset is used in this work). Finally we introduce a CatNet, which incorporates the class incremental learning strategy with 3D ConvNets. Two evaluation metrics are presented at the end of this section.

3.1. Architectures

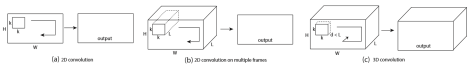
Two types of R3D architectures are investigated in this work, which use ResNet and ResNeXt respectively as the block unit. The difference between ResNet and ResNeXt is refered in [16]. Three models are studied, which are ResNet-50 using 16 frames as an input (ResNet-50-16f), ResNeXt-101 using 16 frames as an input (ResNeXt-101-16f), and ResNeXt-101 using 32 frames as an input (ResNeXt-101-32f) [16, 21].

As mentioned earlier in the paper, temporal information is important for video understanding. 3D convolution has became a popular operation to preserve the temporal properties of a video. Figure 2(a) illustrates the difference between the 3D convolutional operation and the 2D convolutional operation. 2D ConvNets lose track of temporal information of the input after every convolutional operation while 3D ConvNets are able to output a video clip by feeding a video clip, which preserves the temporal information. Figure 2(b) illustrates previous popular architectures for video action recognition. Previous two-stream architectures learn the temporal information by using 3D convolution and optical flow [35, 11, 6, 10]. Optical flow represents the motion over time can be calculated from every two neighboring frames. Traditional two-stream architectures all use RGB frames and optical flow as two streams. However, the computation of optical flow is very complex i.e., computing over each individual frame, which is difficult in real-world applications [9]. We propose the use of depth frames as another stream with RGB frames for the EgoGesture dataset. It should be noticed that, differing from optical flow, the objective of the depth stream is not to extract temporal information. It aims to provide the depth level, in which different backgrounds e.g., brightness, indoor and outdoor in EgoGesture may drive different effects on RGB frames. The temporal information can be preserved by using 3D convolution. Figure 2(c) shows the two-stream architecture deployed in this work. Two 3D ConvNets are trained independently by using RGB and depth videos (see the training flow in Figure 2(c)) and the second last layer features of two networks are concatenated with each other (see the testing flow in Figure 2(c)), which is used for clustering during testing (we will explain this in the next section).

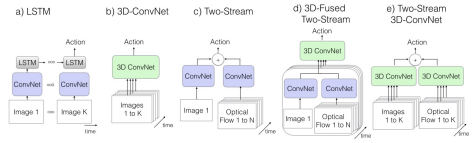
3.2. CatNet

We incorporate iCaRL [29] with 3D ConvNets for class incremental EgoGesture video recognition in this work and we call this framework CatNet. The whole training process for a CatNet is summarized in Algorithm 1 and Algorithm 2 and is visualized in Figure 3. The core part of CatNet is to cache some previous class samples that are the most representative of the old class i.e., see the green block in Figure 3. The memory caches the selected video samples and their corresponding predictions for previous class samples, which is achieved by learning the **feature representation** (Algorithm 2). The feature representation is computed by the mean value of features (i.e., the second last layer output of the 3D ConvNet) corresponding to one class (see the feature mean matrix in Figure 3). We then cache the first k samples in which features of those samples are the closest to the representation (feature mean). The cached samples play two roles during the class incremental learning. First, the cached samples are used to compute the representation for each class, which is used for inference (see the nearest mean classifier in Figure 3). Second, the prediction is used to compute the distillation loss during the training in Algorithm 1. The inference procedure is summarized in Algorithm 3 and the yellow block Figure 3. The feature is extracted from a testing video, which is to be compared with the cached feature mean matrix. The class minimizing the L_2 distance is assigned as the predicted class. *All features mentioned in this work are L_2-normalized.*

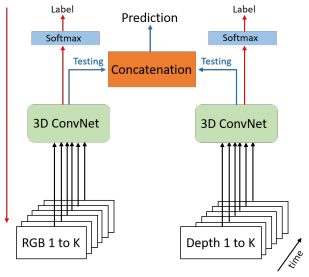
Two evaluation metrics are used to validate the performance for each model during class incremental learning, which are mean accuracy and backward transfer (BWT) [31]. Table 1 shows an accuracy matrix R, which is able to observe the performance of a trained model changing over time. The row represents the model $\mathcal{M}_i$ trained on task i. The column represents the testing data from task i. The gray part is the BWT score, which measures the accuracy over previously encountered tasks (average of gray



(a) Comparison between 2D and 3D convolutional operations. Left: 2D convolution on an image, in which the output is an image. Middle: 2D convolution on a video clip, in which the output is an image. Right: 3D convolution on a video clip, in which the output is also a video clip. Figure from [37].



(b) Previous two-stream video architectures in the literature. K stands for the total number of frames in a video while N stands for a subset of neighboring frames of the video. Figure from [6].



(c) Two-stream 3D ConvNet used in this work.

Figure 2: Illustration of 3D ConvNets and the two-stream 3D ConvNet used in this work.

Algorithm 1 Training a CatNet for EgoGesture

Input:

- m ▷ *Number of added new classes*
- $\mathbf{X} \in \mathbb{R}^{N\times C\times L\times H\times W}$, $\mathbf{Y} \in \mathbb{R}^{N\times m}$ ▷ *New-class video clips and labels, N number of clips, C number of channels, L clip length, H frame height, W frame width*
- K cached samples for each previous class ▷ $\mathbf{X}_{cached} \in \mathbb{R}^{P\times C\times L\times H\times W}$ *where $P = K \times n$, n is number of learned classes*

Require:

- Current model $\mathcal{M}$ and weights $\mathbf{\Theta}$ ▷ *We denote the first layer weight until the last layer weight as* $\Theta_1, \Theta_2, \ldots, \Theta_t$

Training starts:

$q = \mathcal{M}(\mathbf{X}_{cached}, \Theta_{1\sim t})$ ▷ *Softmax prediction for previous samples*

Optimizing (e.g., BackProp) with loss function below:

$\mathcal{L} = -\sum_{x_i\in X, y_i\in Y}\sum_{j=1}^{m} y_{i,j}\log(\mathcal{M}(x_i,\Theta_{1\sim t}))-\sum_{x_i\in X_{cached}, q_i\in q}\sum_{j=1}^{P} q_{i,j}\log(\mathcal{M}(x_i,\Theta_{1\sim t}))$ ▷ *This contains the new-class cross entropy loss and the old-class distillation loss.*

Training finishes

Algorithm 2 Learning the Feature Representation

Input:

- $\mathbf{X} \in \mathbb{R}^{N\times C\times L\times H\times W}$ ▷ *New-class video clips*

Repeat for m classes:

$\mathbf{X}_i \in \mathbf{X}$ ▷ *Samples of one new class*

$\mathcal{F} = \mathcal{M}(\mathbf{X}_i, \Theta_{1\sim t-1})$ ▷ *Extract the second last layer feature for one new-class samples*

$\mu \leftarrow \frac{1}{|\mathcal{F}|}\sum_{\mathcal{F}_i\in\mathcal{F}} \mathcal{F}_i$

for $k = 1 : K$ **do**

$p_k \leftarrow \text{argmin}_{x\in X_i} \|\mu - \frac{1}{k}(\mathcal{M}(x,\Theta_{1\sim t-1}) + \sum_{j=1}^{k-1}\mathcal{M}(p_j,\Theta_{1\sim t-1})\|$

end for

$\mathbf{X}_{cached} \leftarrow (p_1, p_2, \ldots, p_K)$

Output:

$\mathbf{X}_{cached}$

Algorithm 3 Inference

Input:

- $x \in \mathbb{R}^{C\times L\times H\times W}$ ▷ *Testing video clips*

Require:

Trained model $\mathcal{M}$ and weights $\Theta_{1\sim t-1}$

$P = K \times n$ cached image set for all n classes $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_n \in \mathbb{R}^{K\times C\times L\times H\times W}$ ▷ *Cached exemplar set*

Compute exemplar feature means:

1: **for** $k = 1 : n$ **do**

2: $\mu_k \leftarrow \frac{1}{K}\sum_{x_i\in\mathbf{x}_k}\mathcal{M}(x_i,\Theta_{1\sim t-1})$

3: **end for**

Output:

$y \leftarrow \text{argmin}_{k=1,\ldots,n}\|\mathcal{M}(x,\Theta_{1\sim t-1}) - \mu_k\|$

elements in Table 1). BWT indicates the performance related to the memorization capability. The mean accuracy, average of last row elements in Table 1, demonstrates the overall performance on each task for the final model.

4. Experiments

In this section, we describe experimental evaluation in this work. All models are tested on a public egocentric gesture dataset – EgoGesture. Details of network settings are provided in the Appendix.

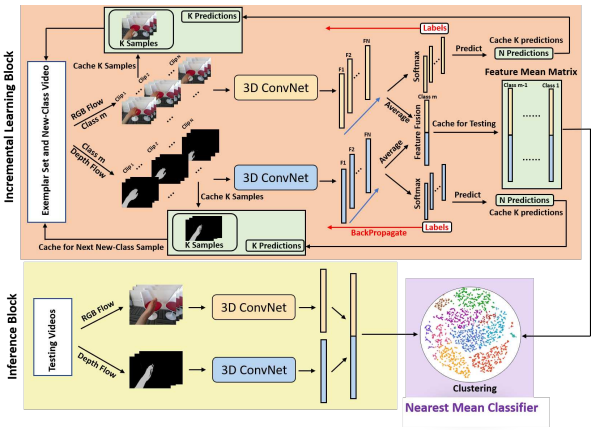


Figure 3: Schematic of a two-stream CatNet for EgoGesture video recognition.

Table 1: Accuracy matrix R during lifelong training, where $\mathcal{M}_i$ is the model trained using training data Tr_i in task T_i, Te_i is the testing data in task T_i, and R_{ij} = classification accuracy of the model $\mathcal{M}_i$ training on Tr_i and testing on Te_j. The number of tasks is N. Gray color represents the BWT score.

R	Te_1	Te_2	$\cdots$	Te_N
$\mathcal{M}_1$	R_{11}	R_{12}	$\cdots$	R_{1N}
$\mathcal{M}_2$	R_{21}	R_{22}	$\cdots$	R_{2N}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\mathcal{M}_N$	R_{N1}	R_{N2}	$\cdots$	R_{NN}

4.1. EgoGesture Dataset

EgoGesture is a recent multimodal large-scale video dataset for egocentric hand gesture recognition [46]. There are 83 classes of static and dynamic gestures collected from 6 diverse indoor and outdoor scenes. There are 24,161 video gesture samples and 2,953,224 frames, which are collected in RGB and depth modalities from 50 distinct participants. We follow the previous work [46, 21] to process the data, in which data was split by participants into training (60%), validation (20%) and testing (20%). Participant IDs 2, 9, 11, 14, 18, 19, 28, 31, 41, 47 were used for testing and 1, 7, 12, 13, 24, 29, 33, 34, 35, 37 were used for validation. The rest of data was used for training. Similar to [21], we also included validation data during training.

4.2. Class Incremental Learning

We focus on one of the lifelong learning scenarios in this study – class incremental learning. Every time we extended the model, we added 5 new classes. In order to get good generalization for our model on class incremental learning, we firstly trained our model using the first 40 classes (we will refer this initial task as task 0 in the later part of this paper). We then trained our model using 5 new classes (41 – 45) as task 1. We repeated this procedure until task 9, in which data of classes 81 – 85 was used. As a result, we have 10 tasks (including the initial training on the first 40 classes) over the class incremental learning process.

4.3. Implementation Details

Three models were investigated in this work, which are ResNeXt-101-32f, ResNeXt-101-16f, ResNet-50-16f. Each model was tested by using 4 feature representations, which are depth input, RGB input, RGB and depth input (RGB-D) and two-stream. All models were first pretrained on Kinetics [19, 16].

Following previous work [21], we used the following methods to pre-process the data during training: (1) Each frame was firstly spatially resized to 112×112 pixels; (2) Each frame was scaled randomly with one of $\{1, \frac{1}{2^{1/4}}, \frac{1}{2^{3/4}}, \frac{1}{2}\}$ scales and then randomly cropped to size 112×112; (3) Spatial elastic displacement [34] with $\alpha = 1$ and $\sigma = 2$ was applied to cropped and scaled frames; (4) A fixed length clip (16-frame and 32-frame used in this work) was generated around the selected temporal position. If the video is shorter than the fixed length, we looped it as many times as possible; (5) We performed mean subtraction for each input channel, where mean values of ActivityNet [4] were used. Finally we get the following types of inputs to our model: (1) Depth input, which has the size of 1 channel $\times$ 16/32 frames $\times$ 112 pixels $\times$ 112 pixels; (2) RGB input, which has the size of 3 channels $\times$ 16/32 frames $\times$ 112 pixels $\times$ 112 pixels; (3) RGB-D input, which has the size of 4 channels $\times$ 16/32 frames $\times$ 112 pixels $\times$ 112 pixels. Stochastic gradient descent was carried out to optimize the model when using backpropagation, which has a weight decay of 0.001 and 0.9 for momentum. For task 0 training (first 40 classes), the learning rate was started from 0.001 and divided by 10 at the 25th epoch. Training was completed after 50 epochs. For class incremental learning, the learning rate was started from 0.001 and divided by 10 at the 6th epoch. Training was completed after 12 epochs. Batch size was set to 64 in the experiment.

During the testing session, testing frames were first scaled to the size of 112×112 and then cropped around a central position at scale 1. A testing video clip (with length 16 or 32) was generated at the central temporal position of a whole video. If the testing video clip was shorter than the required length, we looped it as many times as necessary. All testing frames were mean centered the same way those used for training.

5. Results

The presentation of results is divided into two parts. The first part compares the performance of different feature representations i.e., depth, RGB, RGB-D and two-stream. The second part compares the performance of dif-

ferent 3D ConvNets i.e., ResNeXt-101-32f, ResNeXt-101-16f and ResNet-50-16f. We use the joint training model as an upper bound comparison, which is trained by using the data of all classes. Mean accuracy and memorization capability are utilized to measure the performance.

5.1. Comparison of Feature Representations

5.1.1 Final Model Accuracy for All Tasks

Table 2 shows the mean accuracy across different tasks using different feature representations. It can be noticed that the two-stream approach achieves the highest accuracy for both joint training and class incremental training for all three different architectures, which indicates that two independent feature extractors for depth and RGB inputs should be beneficial for both joint training and lifelong learning. Previous work [1, 5, 46] has demonstrated that using RGB-D can outperform those only using one modality input in terms of joint training. However, it seems that the RGB-D feature representation is not beneficial to lifelong learning as it can be noticed that the depth feature representation performs better than the RGB-D feature representation for ResNext-101-16f and ResNet-50-16f during training the CatNet.

Table 2: Mean accuracy for different feature representations. Bold text indicates the highest accuracy

Method		Joint training	CatNet
ResNeXt-101-32f	Depth	0.909	0.845
	RGB	0.905	0.859
	RGB-D	0.922	0.861
	Two-stream	**0.932**	**0.884**
ResNeXt-101-16f	Depth	0.883	0.840
	RGB	0.850	0.826
	RGB-D	0.891	0.834
	Two-stream	**0.907**	**0.865**
ResNet-50-16f	Depth	0.870	0.843
	RGB	0.865	0.792
	RGB-D	0.867	0.830
	Two-stream	**0.900**	**0.854**

5.1.2 Memorization Capability

BWT is carried out in this work for measuring the memorization capability. Table 3 summarizes that BWT in Figure 4 (left). Compared to other feature representation approaches, the two-stream feature representation shows that the model produces lighter color in the matrix over time, which indicates a better memorization capability. Similar to the mean accuracy, the RGB-D feature representation performs worse than the depth feature representation for ResNeXt-101-16f and ResNet-50-16f. These results indicate that the one-stream CatNet is not able to fully make use of RGB-D information when only concatenating RGB and depth as an input to the model with respect to EgoGesture video recognition. Thus we provide such a two-stream strategy which shows good performance for both joint training and lifelong learning.

5.2. Comparison of Architectures

5.2.1 Final Model Accuracy for All Tasks

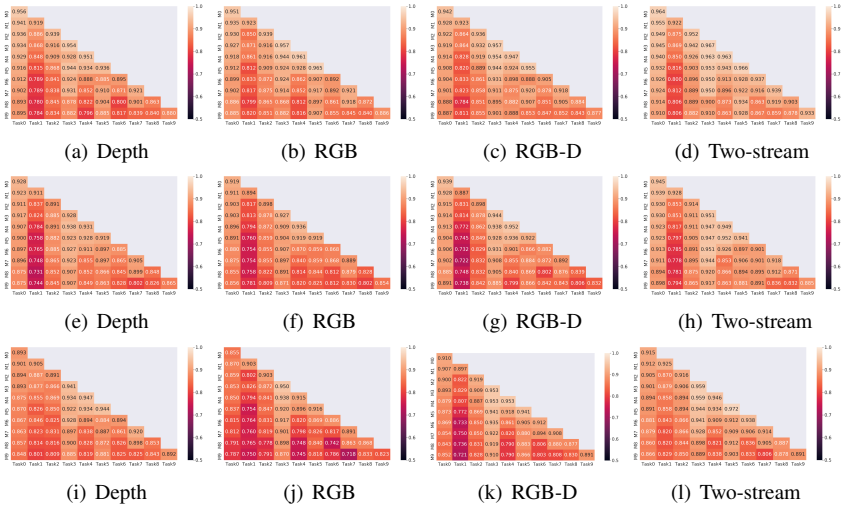
Table 4 shows the mean accuracy of joint training and class incremental training across different architectures. It can be noticed that ResNeXt-101-32f achieves the best performance for both joint training and lifelong learning across different feature representations. ResNeXt-101-32f has the same depth as ResNeXt-101-16f, but a longer temporal frame length is used for the input clip when using ResNeXt-101-32f, which is able to preserve more temporal information from videos.

Table 4: Comparison between architectures using mean accuracy. Bold text indicates the best performance.

Method		Joint training	CatNet
Depth	ResNeXt-101-32f	**0.909**	**0.845**
	ResNeXt-101-16f	0.883	0.840
	ResNet-50-16f	0.870	0.843
RGB	ResNeXt-101-32f	**0.905**	**0.859**
	ResNeXt-101-16f	0.850	0.826
	ResNet-50-16f	0.865	0.792
RGB-D	ResNeXt-101-32f	**0.922**	**0.861**
	ResNeXt-101-16f	0.891	0.834
	ResNet-50-16f	0.867	0.830
Two-stream	ResNeXt-101-32f	**0.932**	**0.884**
	ResNeXt-101-16f	0.907	0.865
	ResNet-50-16f	0.900	0.854

5.2.2 Memorization Capability

Table 5 shows BWT across three different architectures for each feature representation. The rank of BWT for each feature representation is the same, which is ResNeXt-101-32f, ResNeXt-101-16f and ResNet-50-16f from high to low. Because BWT can also be affected by the initial performance of the model i.e., initial classification performance for the first 40 classes in our case, we also test the initial classification accuracy for the first 40 classes for each model as seen in Table 5. The initial accuracy has exactly the same order as BWT in terms of ranking the three architectures for each representation i.e., ResNeXt-101-32f, ResNeXt-101-16f and ResNet-50-16f. This indicates that a deeper model is able to improve the initial performance on the initial task



(a) Depth (b) RGB (c) RGB-D (d) Two-stream

(e) Depth (f) RGB (g) RGB-D (h) Two-stream

(i) Depth (j) RGB (k) RGB-D (l) Two-stream

Table 3: Memorization capability across different feature representations. Bold text indicates the highest accuracy.

Model		BWT
ResNeXt-101-32f	Depth	0.873
	RGB	0.880
	RGB-D	0.882
	Two-stream	**0.900**
ResNeXt-101-16f	Depth	0.865
	RGB	0.849
	RGB-D	0.856
	Two-stream	**0.887**
ResNet-50-16f	Depth	0.863
	RGB	0.823
	RGB-D	0.853
	Two-stream	**0.880**

Figure 4: Left: Classification accuracy matrix R for three architectures i.e., ResNext-101-32f (top), ResNext-101-16f (middle) and ResNet-50-16f (bottom). The vertical axis is the model $\mathcal{M}_i$ trained on the task T_i. The horizontal axis is the task T_i data. Lighter color indicates better performance. Right: Table summary of the figure on the left.

(task 0) but can not benefit the memorization capability on a lifelong learning task.

Table 5: Comparison between different architectures for memorization capability. Bold text indicates the best performance.

Model		BWT	Initial accuracy
Depth	ResNeXt-101-32f	**0.873**	**0.956**
	ResNeXt-101-16f	0.865	0.928
	ResNet-50-16f	0.863	0.894
RGB	ResNeXt-101-32f	**0.880**	**0.951**
	ResNeXt-101-16f	0.849	0.919
	ResNet-50-16f	0.823	0.878
RGB-D	ResNeXt-101-32f	**0.882**	**0.942**
	ResNeXt-101-16f	0.856	0.939
	ResNet-50-16f	0.853	0.910
Two-stream	ResNeXt-101-32f	**0.900**	**0.964**
	ResNeXt-101-16f	0.887	0.945
	ResNet-50-16f	0.880	0.915

6. Discussion

Figure 5 shows the features produced by a feature mean matrix according to the depth input and the RGB input respectively for all 83 classes (horizontal represents different classes). Given a feature mean matrix $\mathcal{S} \in \mathbb{R}^{m \times c}$, where m is the number of features and c is the number of classes, it is not easy to visualize because of the large feature number. We average $\mathcal{S}$ over the feature dimension, which is derived as $\frac{1}{m}\sum_i^m \mathcal{S}_i$, for visualization. It can be seen that the depth feature representation and the RGB feature representation are quite different from each other. Because our model uses the mean exemplar set as a reference for classification, the two-stream approach, which fuses depth features and RGB features from the second last layer, can be beneficial for this case.

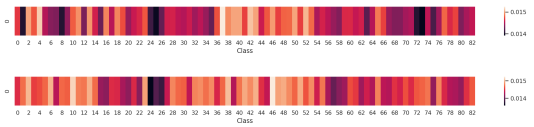


Figure 5: Visualization of features extracted by the two-stream CatNet.

7. Conclusion

In this paper, we investigate class incremental learning in the context of egocentric gesture video recognition, in which we address the issue in such scenarios for real-wold applications is that may easily become necessary to add new gestures to the system. A 3D convolution based framework named CatNet is introduced and we demonstrate the efficacy of CatNets on the EgoGesture dataset, in which the performance on the class incremental task does not drop significantly compared to joint training. Importantly, we propose the use of a two-stream architecture for the CatNet, in which two 3D ConvNets are trained independently by feeding RGB and depth inputs. Results demonstrate that the two-stream CatNet performs better than 3 other one-stream CatNets both on the mean accuracy and the memorization capability. Results also demonstrate that CatNet exhibits some forgetting of knowledge, which can be further investigated in the future.

References

[1] Mahdi Abavisani, Hamid Reza Vaezi Joze, and Vishal M Patel. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1165–1174, 2019.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[3] Marcus K Benna and Stefano Fusi. Computational principles of synaptic memory consolidation. *Nature neuroscience*, 19(12):1697, 2016.

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–970, 2015.

[5] Congqi Cao, Yifan Zhang, Yi Wu, Hanqing Lu, and Jian Cheng. Egocentric gesture recognition using recurrent 3D convolutional neural networks with spatiotemporal transformer modules. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3763–3771, 2017.

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

[7] Tejo Chalasani, Jan Ondrej, and Aljosa Smolic. Egocentric gesture recognition for head mounted ar devices. *Adjunct Proceedings of the IEEE and ACM International Symposium for Mixed and Augmented Reality*, 2018.

[8] Tejo Chalasani and Aljosa Smolic. Simultaneous segmentation and recognition: Towards more accurate ego gesture recognition. 2019.

[9] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Amir Hossein Karami, Mohammad Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3D convnets: New architecture and transfer learning for video classification. *arXiv preprint arXiv:1711.08200*, 2017.

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[12] Fan Feng, Rosa HM Chan, Xuesong Shi, Yimin Zhang, and Qi She. Challenges in task incremental learning for assistive robotics. *IEEE Access*, 2019.

[13] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[14] Stefano Fusi, Patrick J Drew, and Larry F Abbott. Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611, 2005.

[15] Alexander Gepperth and Cem Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016.

[16] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[17] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

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

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[20] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[21] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, and Gerhard Rigoll. Real-time hand gesture detection and classification using convolutional neural networks. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] Li Liu and Ling Shao. Learning discriminative representations from RGB-D video data. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[25] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[26] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[27] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

[28] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.

[29] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] Qi She, Fan Feng, Xinyue Hao, Qihan Yang, Chuanlin Lan, Vincenzo Lomonaco, Xuesong Shi, Zhengwei Wang, Yao Guo, Yimin Zhang, et al. OpenLORIS-Object: A dataset and benchmark towards lifelong object recognition. *arXiv preprint arXiv:1911.06487*, 2019.

[32] Qi She, Yuan Gao, Kai Xu, and Rosa HM Chan. Reduced-rank linear dynamical systems. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[33] Qi She and Anqi Wu. Neural dynamics discovery via gaussian process recurrent neural networks. *arXiv preprint arXiv:1907.00650*, 2019.

[34] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

[35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

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

[38] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[39] Jun Wan, Yibing Zhao, Shuai Zhou, Isabelle Guyon, Sergio Escalera, and Stan Z Li. Chalearn looking at people RGB-D isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–64, 2016.

[40] Cheng Wang, Haojin Yang, and Christoph Meinel. Exploring multimodal video representation for action recognition. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1924–1931. IEEE, 2016.

[41] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[42] Zhengwei Wang, Graham Healy, Alan F Smeaton, and Tomas E Ward. Use of neural signals to evaluate the quality of generative adversarial network performance in facial image generation. *Cognitive Computation*, 12(1):13–24, 2020.

[43] Zhengwei Wang, Qi She, Alan F Smeaton, Tomas E Ward, and Graham Healy. Neuroscore: A brain-inspired evaluation metric for generative adversarial networks. *arXiv preprint arXiv:1905.04243*, 2019.

[44] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks: A survey and taxonomy. *arXiv preprint arXiv:1906.01529*, 2019.

[45] LI Yang, Jin HUANG, TIAN Feng, WANG Hong-An, and DAI Guo-Zhong. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019.

[46] Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu. EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia*, 20(5):1038–1050, 2018.