IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 16, 2020, accepted November 27, 2020, date of publication December 4, 2020,
date of current version December 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042644

Channel Transformer Network

FUPING ZHANG[1,2], (Student Member, IEEE),
PENGCHENG ZHAO[1,3], (Student Member, IEEE),
AND JIANMING WEI[1], (Member, IEEE)
[1]Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China
[2]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

Corresponding author: Jianming Wei (wjm@sari.ac.cn)

This work was supported in part by the Research and Application of Online-monitoring and Intelligent Emergency Rescue Technology in Hazardous Chemicals Industrial Zone under Grant 19DZ1202200, and in part by the Opening Project of Shanghai Trusted Industrial Control Platform under Grant TICPSH202003004-ZC.

ABSTRACT Current attention or transform modules in Convolutional Neural Networks (CNNs) are designed pursuing lightweight and in-place. Generally, we need to decrease the channel dimension of input feature maps for reducing computation cost firstly. And then we do some transformation for extracting weight maps or converting to other feature space etc. Finally, we increase the channel dimension back for outputting feature maps with the same size as input. When we change the channel dimension, commonly we choose 1×1 convolutional layers or fully connected layers. They are simple and effective, but need learning parameters and consuming more memory with other computation resources. We propose a novel parameter free method named Channel Transformer Network (CTN) to decrease or increase channels for these modules whilst keeping most information with lower computation complexity. We also introduce a Video Co-segment Attentive Network (VCAN) for person re-identification (ReID) to improve pedestrian's noticeable representation across multiple video frames. We embed CTN in Non-local, CBAM, COSAM and VCAN blocks to replace 1×1 convolutional or fully connected layers. Experiments of VCAN and CTN embedding models on Mars dataset for person ReID show significant performance in computation efficiency and accuracy, especially VCAN reaches 90.05% in Rank-1. We believe CTN can also be used in other vision tasks like image classification and object detection etc.

INDEX TERMS Channel transform, person re-identification, pyramid pooling, co-segmentation.

I. INTRODUCTION

CNNs were inspired by biological vision cortex where small areas of neurons are responsive to particular regions of the visual field namely receptive field [1] which is prevalent in modern CNNs. A self-organized neural network model was proposed by Fukushima in [2] for visual pattern recognition named ''neocognitron'', which had a similar hierarchy structure of the visual nervous system introduced by Hubel and Wiesel in [1]. Thanks to this type of hierarchical structure, neocognitron had an important capability to recognize stimulus patterns based on appearance similarity neglecting their position and small distortion, which is just the basis of current popular convolutional and pooling layers in today's dominant CNNs. These works established later research on CNNs, especially after LeCun introducing back-propagation algorithm to it for gradient based learning in LeNet-5 [3] which formed the prototype of contemporary CNNs. With the limitation of dataset scale, network size, hardware etc., early CNNs were very hard to be trained and the recognition accuracy were even worse than traditional algorithms such as SVM, Random Forests etc. Thus, they were not received enough interests till Hinton won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with [4].

An increasing number of works like ResNet [5] and ResNeXt [6] have shown that deeper and wider CNNs can extract rich semantic information. Whilst attention mechanism has been another important factor in deep CNNs since it can improve recognition performance through multiplying feature maps by weight score maps. But there exists a problem that the number of feature maps which is called channel dimension increases dramatically in deep CNNs though feature map spatial size getting smaller. Therefore, it will take too much computation resources if we calculate channel-wise

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

weight score directly. We need to reduce channel dimension and keep spatial size of feature maps.

Normally, we design an attention or transform block which can be inserted into any place in CNNs without affecting the size of input and output feature maps, such as Non-local [7], [8], CBAM [9], COSAM [10], CosegCA [11], etc. In these blocks or modules, 1×1 convolutional layers or fully connected layers are used to reduce channel dimension at the beginning for saving computation complexity. And at the end, they are used again to increase channel dimension for restoring the size of feature maps. It's really convenient and efficient to change channel dimension with 1×1 convolutional layers or fully connected layers, but need learning more network parameters and costing more computation resources.

In this paper, we propose the parameter-free Channel Transformer Network (CTN) to replace 1×1 convolutional layers or fully connected layers for decreasing or increasing channel dimension with less computation and most information in other CNN modules. Specifically, we design CTN with two parts to change channel dimension, one is for decreasing, and the other one is for increasing. We also present the lightweight and efficient Video Co-segment Attentive Network (VCAN) based on CosegCA [11] for person ReID to better study CTN. Then we inject CTN into Non-local [7], [8], CBAM [9], COSAM [10] and VCAN to make them parameter-free when transforming channel dimension. To the best of our knowledge, this is the first work that shows how to change channel dimension in a parameter-free way for saving network parameters and computation resources.

The main contribution of the paper exists in three folds:

(1) We propose a novel method named Channel Transformer Network (CTN) for channel dimension transforming in CNN modules, including decreasing and increasing.
(2) We also introduce a Video Co-segment Attentive Network (VCAN) for person ReID to improve video saliency representation which achieves 90.05% in Rank-1 on Mars dataset outperforming the state-of-the-arts.
(3) Extensive experiments on Mars dataset for video-based person ReID show significant performance of CTN and VCAN in accuracy and computation complexity. CTN embedding models reach more than 89.4% in Rank-1 with lower parameters and computation resources.

The rest of the paper is organized as follows: Recent related works with this paper are reviewed in Section II. Designs of the proposed methods are presented in Section III. Implementation details and experimental results are reported in Section IV. Finally, the conclusion is drawn in Section V.

II. RELATED WORKS

A. PERSON RE-IDENTIFICATION

The task aims to recognize a person through different non-overlapping cameras. It's really a challenging problem since there exist a lot of variations like view point, background clutter, occlusion, misalignment, etc. There have been lots of works focusing on two branches for the problem, one is image-based person ReID [12]–[21], the other one is video-based [8], [10], [22]–[34].

Image-based methods mainly extract local and global spatial information from one image of a person identity such as [12], [13]. Guo *et al.* designed an efficient end-to-end fully convolutional Siamese [35] network and explored multi-level similarity for improving accuracy in [14]. He *et al.* proposed a novel method namely Deep Spatial feature Reconstruction (DSR) to resolve occlusion problem in [15]. Kalayeh *et al.* added a semantic segmentation branch for parsing main human body regions to improve recognition performance in [16]. Ke *et al.* introduced a new ID-adaption network to transform ID-discriminative embedding features to a common discriminative latent space for adapting unseen identities in [17]. Sarfraz *et al.* utilized 14 joint keypoints of human body and different viewpoints to explore fine and coarse pose information for improving pedestrian representation in [18]. Zhang *et al.* utilized attribute information to learn an attribute-semantic and identity-discriminative feature representation for better performance in [19]. Dai *et al.* designed a relearning network with a backbone model pre-trained on a large amount of labeled non-pedestrian images in [20]. It could learn domain-specific features for strong generalization capability of person ReID. Wang *et al.* presented a learning-to-mis-rank formulation and a novel multi-stage network architecture to attack person ReID systems for examining their robustness in [21].

Video-based methods can make full use of spatial and temporal information in multiple frames within a person tracklet. Chen *et al.* introduced OFEI (Optical Flow Energy Image) feature to exploit spatial-temporally stable regions of a pedestrian across frames in [22]. Mclaughlin *et al.* proposed a Siamese [35] network structure in [23]. They used CNN for extracting spatial features of multi-frames and RNN (Recurrent Neural Network) for exploring temporal information from them. Then they utilized temporal pooling layer to fuse the feature maps output from RNN. Yan *et al.* designed a similar structure like [23] in [24]. They used LSTM (Long Short-Term Memory) for aggregating temporal information and fully connected layers for fusing whole representation. Hermans *et al.* showed in [25] that their batch hard triplet loss with soft margin achieved outstanding performance in person ReID compared with both traditional triplet loss and other published variants before. Zhou *et al.* presented a neural network architecture including temporal attention module for metric learning and spatial recurrent module for feature learning in [26]. The temporal attention module could find most discriminative images within the input video sequence. And the spatial recurrent module could ensemble the around information of every point in feature maps to calculate the similarity between two video tracklets. Chen *et al.* divided a long video tracklet into several short subsequences and aggregated top-ranked similarities of them for similarity estimation in [27]. It could minimize the intra-class variation in

appearance whilst keeping other spatial and temporal information. Gao *et al.* revisited and compared temporal modeling approaches for video-based person ReID in [28]. They also proposed a new attention generation network for extracting temporal information. Li *et al.* proposed a spatiotemporal attention model with a diversity regularization term in [29]. It could find distinctive human body parts automatically for better recognition accuracy. Wu *et al.* introduced a stepwise learning method in [30] for utilizing unlabeled pedestrian video sequences to improve the model performance. Zhang *et al.* imposed a reinforcement learning based method [31] to train an agent for discriminating a pair of images one time. Most methods focused on extracting discriminative clip-level features, whereas Isobe *et al.* highlighted the clip-level data augmentation in [32], since inconsistent data augmentation within a video sequence brought additional noise. Temporal information is very important in video sequences, and Li *et al.* exploited the multi-granularity temporal clues in a video clip in [33]. They made use of parallel dilated convolutions with different rates for short-term cues and a temporal self-attention model for long-term dependencies. Wu *et al.* utilized a GNN (Graph Neural Network) to leverage the correlations between the local parts across frames in a tracklet of a person in [34] for better whole pedestrian representation. Long-range dependences in feature maps are very important in recognition performance. Nevertheless, a large number of methods neglected it. Liu *et al.* introduced Non-local [7] to capture it in feature maps within different layers in [8]. They also provided a spatial and temporal efficient method to reduce FLOPs (floating-point operations per second) for saving computation resources. Subramaniam *et al.* formulated a Co-segmentation based Attention Module (COSAM) [10] for video-based person ReID. The module could help to extract a common set of salient feature maps among video frames through a Normalized Cross Correlation (NCC) layer and a summarization layer.

B. ATTENTION MECHANISM

More and more works have proved the excellence of attention mechanism. It has been another important factor in CNN design like depth in ResNet [5], width in Inception [36] and cardinality in ResNeXt [6]. It can help to explore more salient information from feature maps for better recognition accuracy. Wang *et al.* presented a residual style Non-local building block in [7] to capture long-range dependencies in feature maps. Firstly, they used three 1×1 convolutional layers to reduce the channel dimension of the input feature maps by half respectively, and got three scaled feature maps. Secondly, they fused two of them by matrix multiplication followed by Softmax operation to get a weight matrix which was multiplied by the left scaled feature maps later. Finally, they utilized another 1×1 convolutional layer to restore channel dimension as input feature maps, and added it element-wise with the input to get the output self-attention feature maps with the same size as the input ones. Non-local has significant performance for vision recognition tasks, but it's very time and memory consuming. Zhu *et al.* tried to reduce the matrix computation complexity by an asymmetric Non-local architecture in which SPPNet [37] was used to decrease the spatial size of feature maps in [38]. Hu *et al.* proposed a novel channel-wise attention unit named Squeeze-and-Excitation (SE) block [39]. It could be inserted into modern convolutional networks for improving performance like SE-ResNet and SE-Inception. Woo *et al.* introduced a simple but effective attention block termed Convolutional Block Attention Module (CBAM) in [9]. It was inserted as an SE (Squeeze-and-Excitation) module in bottleneck blocks of SE-ResNet [39], and fulfilled channel-wise and spatial-wise attention to improve accuracy.

C. OBJECT CO-SEGMENTATION

Co-segmentation is mainly used to segment objects with the same category from several images even the class is not belonging to the training dataset. It can help to remove background noise and keep salient feature information. Li *et al.* presented a deep learning based Siamese encoder-decoder structure for object co-segmentation in DOCS [40]. They utilized a mutual correlation layer to calculate semantic similarity between a pair of images. Thus, it's hard to co-segment multiple images at a time. They also built a large object co-segmentation dataset with image pairs from PASCAL dataset for training. Chen *et al.* proposed a channel-wise attention based co-segmentation module namely CosegCA in [11]. It's simple yet effective with low computation complexity which was settled in the bottleneck layer of VGG16 [41] for choosing related features semantically. CosegCA provided an efficient instant group co-segmentation method to reduce complexity for co-segmenting several images (more than 2) through group average channel-wise attention in linear time complexity. Hsu *et al.* presented instance co-segmentation aiming to recognize and segment all instances belonging to the same class from two images in [42]. They leveraged co-peak search and instance mask segmentation to outperform the state-of-the-art methods. Zhang *et al.* introduced a spatial and semantic modulated deep network for object co-segmentation in [43]. The spatial modulator could learn a mask with the correlations of image feature descriptors. It focused on the objects of the same class for each image. The semantic modulator was designed for image classification. It could co-segment multi-images at a time without the limitation of paired images. Hung *et al.* proposed a self-supervised learning based part co-segmentation method in [44]. It could segment parts within an image under the help of some loss functions as self-supervised constraints. Lu *et al.* presented a Co-attention Siamese Network (COSNet) [45] to segment foreground objects across video frames through appearance and motion information. COSNet utilized multiple reference frames for useful information which frequently occurred as active foreground objects in segmentation stage.

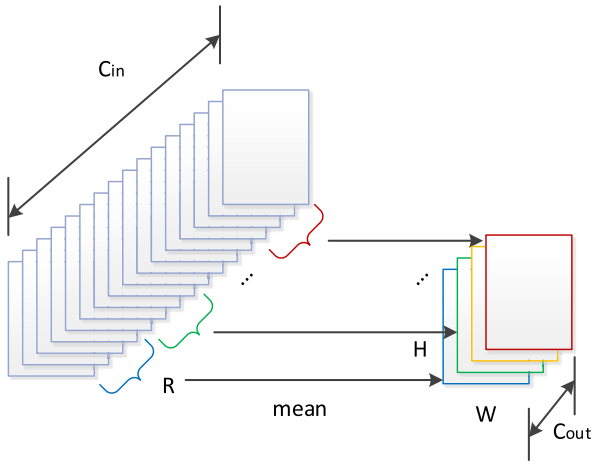


FIGURE 1. Diagram of CDM-SegPool. The input feature maps are segmented into C_{out} parts, and calculated by mean in these segments in channel dimension.

III. METHOD

We propose a novel Channel Transformer Network (CTN) to handle channels transforming problem which includes two parts, one for down-sampling channels named CDM (Channel Down-sample Module) (Fig. 1, Fig. 2, Algorithm 1), and the other one for up-sampling channels called CUM (Channel Up-sample Module) (Fig. 3, Algorithm 2). CTN can be easily embedded into other attention or transform blocks like Non-local [7], [8], CBAM [9], COSAM [10], etc. to replace 1×1 convolutional layers or fully connected layers for reducing parameters and saving computation resources.

We also present a Video Co-segment Attentive Network (VCAN) for person ReID (Fig. 4, Fig. 5) to better study our CTN, in which CosegCA [11] is introduced to video-based person ReID, and adapted for multiple frames (more than 2) to extract pedestrian's noticeable features. And we further leverage CTN instead of fully connected layers in channel attention module (Fig. 5) of VCAN to get model CTN-VCAN with lower network parameters and computational complexity.

A. CHANNEL DOWN-SAMPLE MODULE

Given feature maps with size of $C_{in} \times H \times W$ in which C_{in}, H, W represents input channels, height and width respectively. Our CDM (Channel Down-sample Module) provides two methods named SegPool (Segment Pool) and PymPool (Pyramid Pool) for reducing channels from dimension $C_{in} = 2^n$ to $C_{out} = 2^m$ $(n > m)$ which are both the power of 2. CDM also supplies another method called IrregularPool for decreasing irregular channels, dimension of which is not the power of 2.

1) CDM-SEGPOOL

SegPool separates given feature maps into C_{out} parts according to reduction rate as

$$R = \frac{C_{in}}{C_{out}} = 2^{n-m}, \tag{1}$$

where n and m depend on C_{in} and C_{out}, as shown in Fig. 1, and calculates mean in each part along channel dimension. Then we get down-sampled feature maps with new size of $C_{out} \times H \times W$. We can easily see that, CDM-SegPool is very straight, simple, lightweight and efficient, yet lack of thinking about multi-scale information. It has been witnessed effective in SPPNet [37], FPN (Feature Pyramid Networks) [46] and multi-scale input for multi-stage discriminator [21] etc. that multi-scale information is very important in feature representation. These methods mainly focused on multi-scale spatial information, yet we take care of it in channel dimension from a new insight.

2) CDM-PYMPOOL

Multi-scale information can help to improve recognition accuracy. So, we design CDM-PymPool to capture it along channel dimension in pyramid way like SPPNet [37] but not in spatial field as shown in Fig. 2. CDM-PymPool does the same work as CDM-SegPool for reducing regular channels which are the power of 2. It's based on CDM-SegPool in several rounds with a series of reduction rates. Rounds number is calculated as

$$N_r = log_2^{C_{out}} - 1, \tag{2}$$

where C_{out} is output channels. Reduction rate in each round is calculated as

$$R_k = R_{base} * 2^{k+1}, \quad k \in \{0, 1, \ldots, N_r - 1\}, \tag{3}$$

where R_{base} comes from (1).

In Fig. 2, input feature maps $F_{in} \in \mathbb{R}^{C_{in} \times H \times W}$ are down-sampled by SegPool in several times with different reduction rates got from (3), and then they are concatenated together with global mean and min of input feature maps along channel dimension. At last, we get output feature maps $F_{out} \in \mathbb{R}^{C_{out} \times H \times W}$. The relationship among C_{in}, C_{out} and R_k is given as

$$C_{out} = C_{in} \times \sum_{k=0}^{N_r - 1} \frac{1}{R_k} + 2, \tag{4}$$

where C_{in} and C_{out} are input and output channels respectively, N_r is rounds number got by (2), and R_k is the k^{th} reduction rate got by (3). We can find that we need two another channels got by global mean and min or other methods to get output feature maps with channels number C_{out}.

As introduced above, CDM-SegPool and CDM-PymPool are suitable for input and output channels with the power of 2, not for irregular channels (not the power of 2). So we design another method CDM-IrregularPool for this condition.

3) CDM-IRREGULARPOOL

The method is also based on CDM-SegPool but take responsible for decreasing feature maps on channels with irregular input or output ones, dimension of which is not the power of 2.

As illustrated in algorithm 1, it firstly reduces channels by half in a loop while the latest channel number hc is

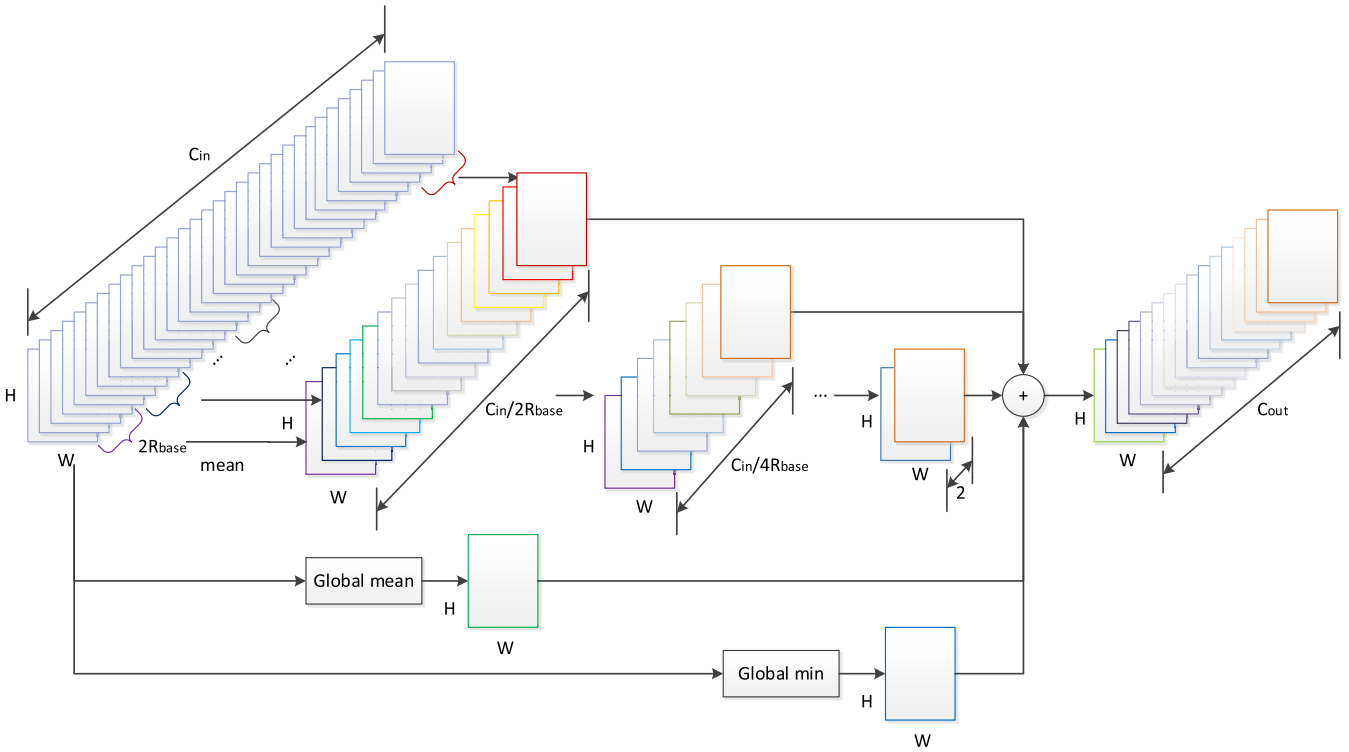


FIGURE 2. Diagram of CDM-PymPool. The channel dimension of input feature maps is reduced in pyramid way based on CDM-SegPool in N_r times, and outputs from all rounds are concatenated with global mean and min or other methods along channel dimension.

Algorithm 1 CDM-IrregularPool

Input: Input feature maps $F_{in} \in \mathbb{R}^{C_{in} \times H \times W}$
Output: Output feature maps $F_{out} \in \mathbb{R}^{C_{out} \times H \times W}$
$hc \leftarrow C_{in}$, $lc \leftarrow C_{out}$
while $hc \geq 2 \times lc$
  if hc can be divided by 2
    do CDM-SegPool with reduction rate 2
    $hc \leftarrow hc/2$
  else
    do CDM-SegPool with reduction rate 2 on $hc - 1$ channels
    concatenate output feature maps with left one
    $hc \leftarrow {}^{hc-1}/_{2} + 1$
end while
if $hc > lc$
  $dc \leftarrow hc - lc$
  choose double dc feature maps and do CDM-SegPool with reduction rate 2
  concatenate output feature maps with left ones
$\boldsymbol{F_{out}} \leftarrow$ output feature maps

bigger than or equal to double output channel number lc. In the loop, if the input channels can be divided by 2, they are decreased by CDM-SegPool with reduction rate 2 in half directly. Else, one channel is neglected and the left are reduced by CDM-SegPool with reduction rate 2. The output feature maps are concatenated with the ignored one along channel dimension. hc is updated with the latest channel number. And then, after the loop, it does CDM-SegPool on double dc channels with reduction rate 2, where dc is the delta channel number between the latest hc and the target lc, and concatenates the output with left ones. At last, we get the final output feature maps with the target channel number C_{out}.

B. CHANNEL UP-SAMPLE MODULE

Our CUM (Channel Up-sample Module) also provides two methods namely Sample (Upsample) and PymPool (Pyramid Pool) for increasing channels from dimension $C_{in} = 2^n$ to $C_{out} = 2^m (n < m)$ which are both the power of 2. CUM presents another method named IrregularPool too for increasing irregular channels, dimension of which is not the power of 2.

1) CUM-SAMPLE

The method just do up-sample nearest for feature maps from $F_{in} \in \mathbb{R}^{C_{in} \times H \times W}$ to $F_{out} \in \mathbb{R}^{C_{out} \times H \times W}$ with expansion rate E as

$$E = \frac{C_{out}}{C_{in}} = 2^{m-n}. \tag{5}$$

We can up-sample the feature maps with a new dimension from $C_{in} \times 1 \times H \times W$ to $C_{in} \times E \times H \times W$, then reshape it to $C_{out} \times H \times W$. It's also very straight and efficient like CDM-SegPool, but do reverse process.

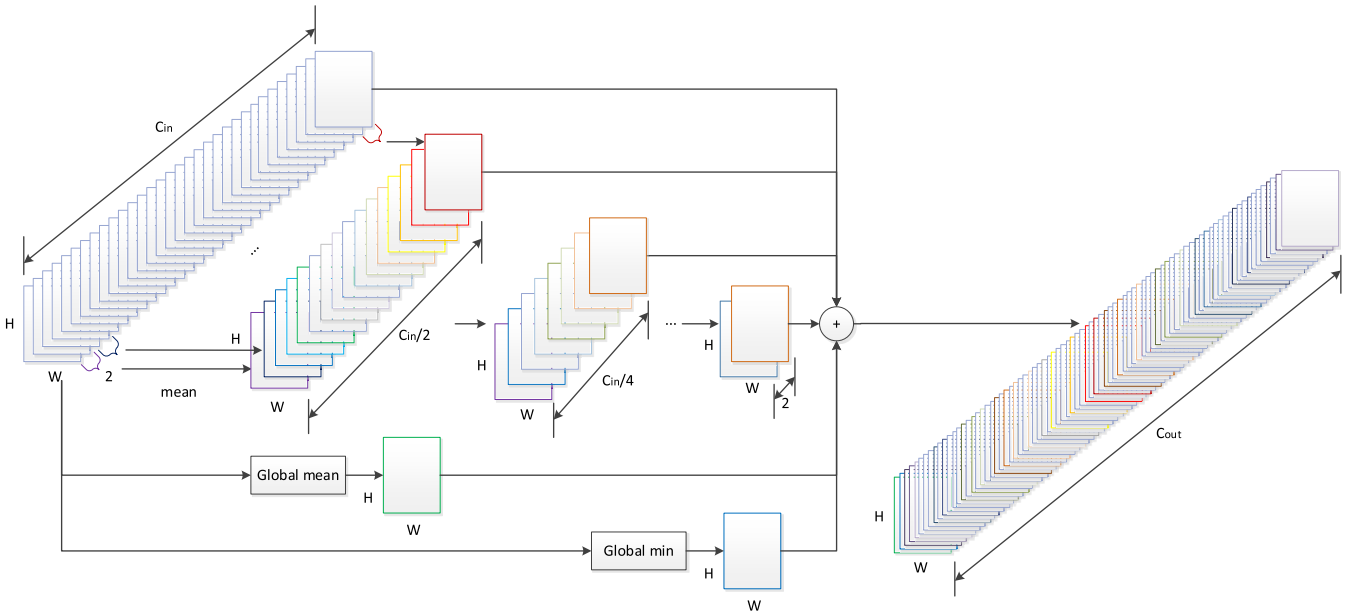


FIGURE 3. Diagram of CUM-PymPool within one expansion procedure. Input feature maps are down-sampled by several CDM-SegPools with reduction rate 2, and all output feature maps are concatenated with original inputs, global mean and min, then outputs feature maps with double C_{in} channels.

2) CUM-PYMPOOL

CUM-PymPool increases input channels mainly based on CDM-SegPool. It expands channels like CDM-PymPool with multi-scale pooling on channels in pyramid way. Expansion rate is as (5), and expansion number is as

$$N_E = m - n. \tag{6}$$

As shown in Fig. 3, in each expansion procedure, we do several rounds CDM-SegPool operation to get multi-scale pooling on channels, rounds number in the k$^{\text{th}}$ procedure is as

$$N_k = log_2^{C_{in}\times 2^k} - 1, \quad k \in \{0, 1, \ldots, N_E - 1\}. \tag{7}$$

We execute N_k times CDM-SegPool with reduction rate 2 in the k$^{\text{th}}$ expansion process. And then we concatenate input feature maps and all these outputs from each CDM-SegPool with global mean and min to get the k$^{\text{th}}$ output feature maps with C_{outk} channels. C_{outk} is as

$$C_{outk} = C_{in} \times 2^k \times \left(1 + \sum_{i=0}^{N_k-1} \frac{1}{2^{i+1}}\right) + 2, \quad k \in \{0, 1, \ldots, N_E - 1\}, \tag{8}$$

$$C_{out} = C_{outk}, \quad (k == N_E - 1). \tag{9}$$

After N_E times expanding, we get output feature maps with C_{out} channels.

3) CUM-IRREGULARPOOL

There also exist the scenes that increasing channels of feature maps with irregular input or output ones, whose channel dimension is not the power of 2. So we design this method to cover it. The module is mainly based on CDM-SegPool to get multi-scale channel pooling information concatenated with input ones as shown in algorithm 2. It increases channels in two loops. In the first loop, channels are increased

Algorithm 2 CUM-IrregularPool

Input: Input feature maps $F_{in} \in \mathbb{R}^{C_{in}\times H\times W}$
Output: Output feature maps $F_{out} \in \mathbb{R}^{C_{out}\times H\times W}$
$lc \leftarrow C_{in}, hc \leftarrow C_{out}$
while $hc \geq 1.5 \times lc$
  if lc can be divided by 2
    do CDM-SegPool with reduction rate 2 on F_{in}
    concatenate F_{in} with output feature maps
    $lc \leftarrow lc + {}^{lc}/_{2}$
  else
    do CDM-SegPool with reduction rate 2 on F_{in} with lc-1 channels
    concatenate F_{in} with output feature maps
    $lc \leftarrow lc + {}^{lc-1}/_{2}$
end while
$F_{new} \leftarrow F_{in}$
while $lc < hc$
  $dc \leftarrow hc - lc, nc \leftarrow$ channels of F_{new}
  while $nc > dc$
    if nc can be divided by 2
      do CDM-SegPool with reduction rate 2 on F_{new}
    else
      do CDM-SegPool with reduction rate 2 on F_{new} with $nc - 1$ channels
  end while
  concatenate F_{in} with F_{new}
  $lc \leftarrow$ channels of the latest F_{in}
end while
$\boldsymbol{F_{out}} \leftarrow$ output feature maps

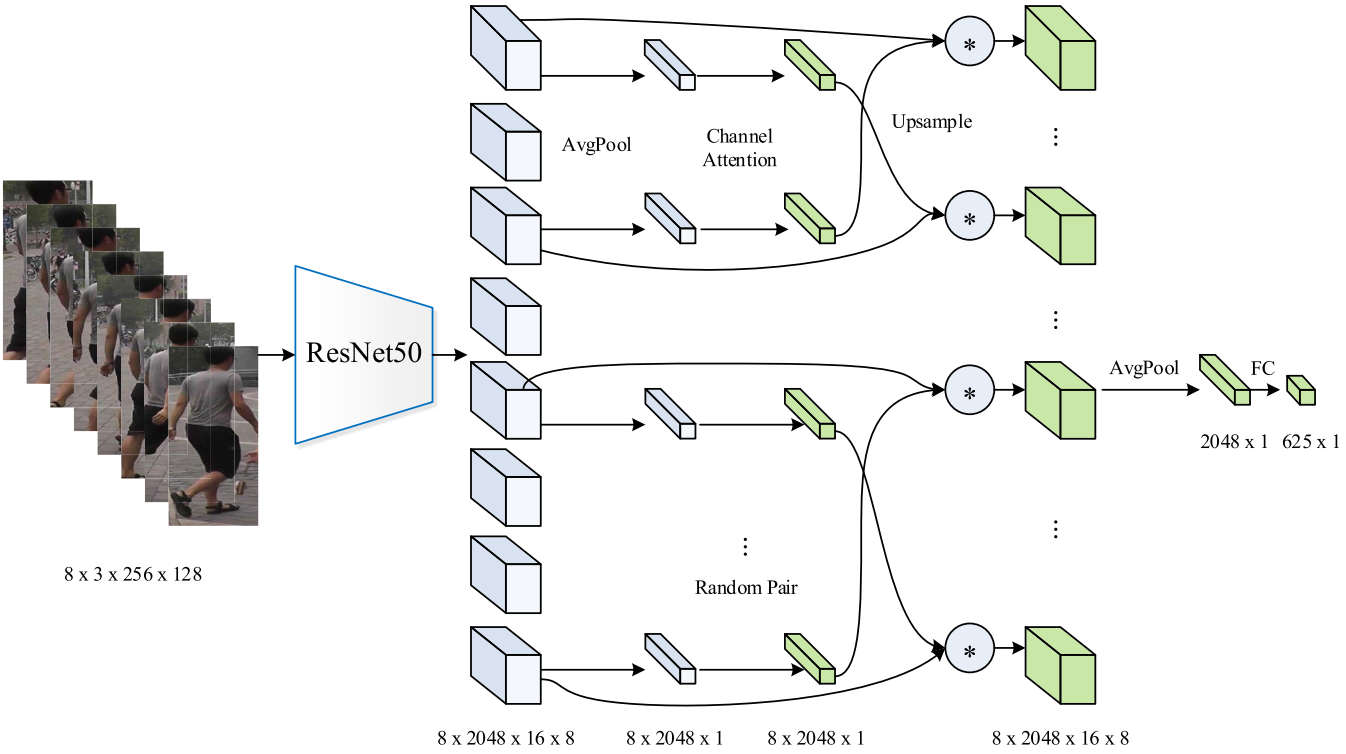


FIGURE 4. Diagram of VCAN. Feature maps of input T video frames are extracted by pre-trained ResNet50, and passed through average pooling and channel attention module to get channel-wise attention scores, then up-sampled with the same size of feature maps to get attention maps. The feature maps accompanied with their attention maps are grouped into $T/2$ pairs randomly, and they are cross multiplied by opposite attention maps in a pair for keeping noticeable information across frames.

gradually while hc is bigger than or equal to 1.5 times the latest lc, where hc equals to the target C_{out} and lc is initialized with input C_{in} but updated in the loop with the latest channel number of F_{in}. If lc can be divided by 2, the channels are decreased by CDM-SegPool with reduction rate 2 in half. Else, one channel is neglected and the left are reduced by CDM-SegPool with reduction rate 2. F_{in} is concatenated with output feature maps along channel dimension, and lc is updated with the latest channel number of F_{in}. In the second loop, it increases channel number like the first loop step by step in which the latest channel number lc is ensured not to exceed hc after increasing until equals to it. Then we get the output feature maps with size of $C_{out} \times H \times W$.

C. VIDEO CO-SEGMENT ATTENTIVE NETWORK

CosegCA [11] is an efficient and lightweight channel-wise attention based co-segmentation module designed for co-segmenting objects with the same category from a pair of images. We believe it can considerably suppress irrelevant objects and background noises from input paired images, and help to extract salient information across video frames like persons and their accessories in video-based person ReID system.

Certainly, it can't be leveraged in our framework directly since multi-frames ($T = 8$) are chosen for learning in which pedestrians and their wearables are what we want. So, we optimized CosegCA for adapting our framework by grouping input feature maps of T frames in pairs randomly to extract salient person and related information. We call it VCAN (Video Co-segment Attentive Network) as shown in Fig. 4.

First of all, we use ResNet50 pre-trained on ImageNet [47] to extract feature maps for given T video frames. Secondly, we do average pooling operation on the feature maps for calculating subsequent attention scores in channel attention module as illustrated in Fig. 5. The first FC layer is used to decrease the size of the input vector with reduction rate 4, and the second FC layer is utilized to increase the size of the transformed vector back. Both of the two FC layers are followed by activation function Tanh and Sigmoid respectively as [11]. And then, we up-sample the attention scores with the same size of the feature maps to get the attention maps. Next, we group the feature maps accompanied with their attention maps into $T/2$ pairs. Finally, we cross multiply feature maps by opposite attention maps in a pair to suppress irrelative objects and background noises while keeping salient pedestrian and related information.

The multiple frames within a person tracklet contain pedestrian and related accessories, while the background and interference information are various. Different channels across the feature maps of one frame represent different semantic information which is irrelevant to object position and scale. Since the attention maps are up-sampled through channel-wise

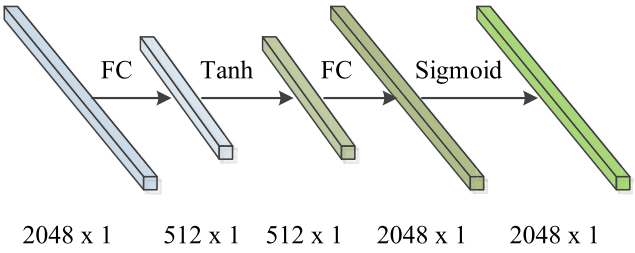


FIGURE 5. Diagram of channel attention module. Two Fully Connected (FC) layers are followed by activation function Tanh and Sigmoid respectively.

attention score vector, the feature maps can help to suppress background and interference information well and retain more information about pedestrian and related wearables when multiplied by the other attention maps. Even if we randomly group feature maps with related attention maps into $T/2$ pairs, and cross-multiply the feature maps by the opposite attention maps in a pair.

IV. EXPERIMENTS

First of all, we inject our CTN(Channel Transformer Network) into Non-local [7], [8], CBAM [9], COSAM [10] and our VCAN (Video Co-segment Attentive Network) to replace 1×1 convolutional layers or fully connected layers. Secondly, we evaluate them respectively in video-based person ReID on MARS [48] dataset. And then, we do some ablation studies to analyze the performance of them. Finally, we compare them with the state-of-the-art methods in video-based person ReID. All experiments in this paper are carried out on two TITAN RTX GPUs (24GB memory).

A. EXPERIMENTAL SETUP

1) DATASET

We select MARS [48] as our training and evaluating dataset, since it's one of the largest datasets in video-based person ReID. It contains 1,261 person identities and about 20,000 tracklets captured by six cameras in Tsinghua University. Each pedestrian occurs at least in two cameras and has around 13.2 tracklets on average.

2) EVALUATION PROTOCOLS

In MARS [48], dataset has been split into train and test sets, and we follow settings in [8] which contains 625 identities in train set and left in test set. We use Rank-1 accuracy of Cumulative Matching Characteristics (CMC) and mean Average Precision (mAP) to evaluate all our experiments.

3) IMPLEMENTATION DETAILS

We realize CBAM [9], COSAM [10], our VCAN and CTN based on Liu's framework [8] with Non-local [7], and choose ResNet50 [5] pre-trained on ImageNet [47] as backbone network. We replace 1×1 convolutional layers or fully connected layers in Non-local, CBAM, COSAM and VCAN with CTN. We follow Liu's settings of hyper-parameters [8] and RRS (Restricted Random Sampling) strategy [29] with $T = 8$ for video frames selection. Selected frames are resized to 256×128 and augmented with random horizontal flip. The last stride of ResNet50 is set to 1 for better performance like [8] in all our experiments.

CBAM [9] is designed to replace SE (Squeeze-and-Excitation) module in bottleneck block of SE-ResNet [39] for improving classification and detection accuracy. We introduce CBAM into video-based person ReID in our framework. Ablation studies show that it's better to insert CBAM between layers in ResNet not as an se-module at the end of each bottleneck block like [9] for video-based person ReID.

COSAM [10] is a co-segmentation based attention module designed for video-based person ReID. It has a normalized cross correlation layer and a summarization layer, which can generate the corresponding spatial attention mask for input feature maps. It uses feature maps of other $T-1$ frames to calculate cross correlations with the feature maps of the selected one frame. So, channel dimension of input feature maps in summarization layer is $(T-1) \times H \times W$, where T is the count of frames sampled from a video tracklet which equals to 8 in our framework, H and W are height and width of input feature maps respectively. We can easily find out that the input channel dimension is irregular (not the power of 2) in the spatial summarization layer, and we need using CDM-IrregularPool module to reduce channel dimension when CTN is embedded into COSAM.

We implement ResNet50-CBAM, ResNet50-COSAM, ResNet50-VCAN and ResNet50-Non-local in the same framework with the same hyper-parameter settings as in [8], and train the four networks for 200 epochs. We follow [8] to choose cross-entropy loss and triplet loss for training and use Adam optimizer with initial learning rate 0.0001 for back-propagation. Learning rate is decayed by 10 every 50 epochs. In each mini-batch, we also select 8 persons with 4 tracklets per-identity and 8 frames per-tracklet as in [8].

B. ABLATION STUDIES

We do ablation studies on Non-local, CBAM, COSAM and VCAN with various reduction rate R which used for reducing input channels for saving computation resources. And we insert them within or after layer1, layer2, layer3 and layer4 of ResNet50 which has 3, 4, 6, and 3 bottleneck blocks respectively.

1) ANALYSIS OF CTN IN NON-LOCAL

We follow [8] to insert 2 Non-local blocks after layer2_3, layer2_4 and 3 Non-local blocks after layer3_4, layer3_5, layer3_6 respectively. And then, we study original Non-local and CTN-Non-local in which 1×1 convolutional layers are replaced by our CTN with various reduction rate R. One thing need to be noted that, there are three 1×1 convolutional layers exist in original Non-local block for reducing channel dimension of input feature maps whilst only one parameter-free CDM (Channel Down-sample Module) is used for it in

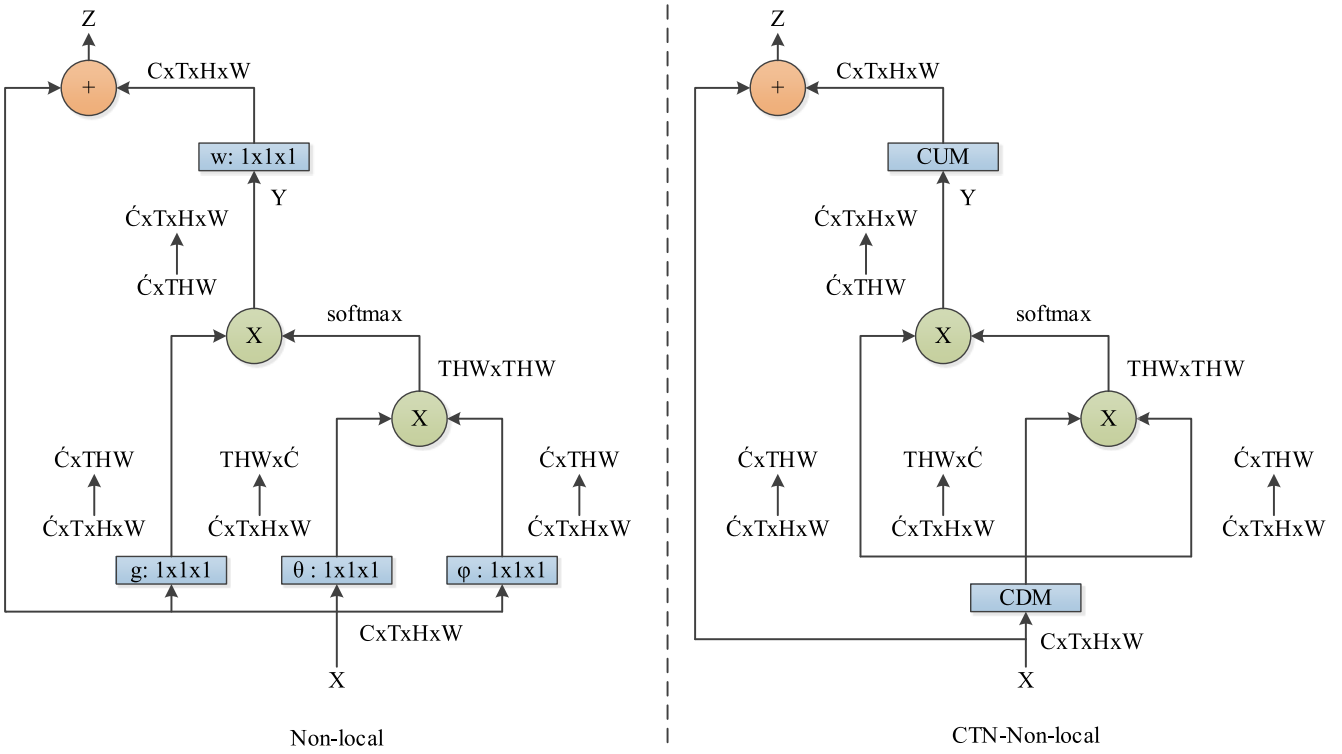


FIGURE 6. Diagram of Non-local and CTN-Non-local. Three 1×1 convolutional layers are used in Non-local for decreasing the channel dimension of input feature maps, yet only one CDM (Channel Down-sample Module) is used for that in CTN-Non-local. The CUM (Channel Up-sample Module) is as the alternative to another 1×1 convolutional layer for increasing channel dimension.

our CTN-Non-local as shown in Fig. 6 for saving computation resources.

We analyze the two additional feature maps F_{add} for PymPool with reduction rate $R = 2$ as shown in Fig. 2, and global mean, min, max and variance of input feature maps are combined for better performance. CDM-PymPool is used to replace the very beginning 1×1 convolutional layers for reducing channels and CUM-Sample is chosen as the alternative to the ending 1×1 convolutional layers for increasing channels in Non-local blocks. The results are illustrated in Table 1 that the pair of mean and min reaches the best result. So we choose global mean and min as the two additional feature maps F_{add} for CDM-PymPool and CUM-PymPool in all our following experiments.

We also check if it's necessary to insert BatchNorm layer after CUM (Channel Up-sample Module) just like original Non-local. The results in Table 2 show that BatchNorm helps to improve Rank-1 accuracy by 0.15% but dropping mAP by 0.09% slightly. We use BatchNorm after CUM like original Non-local block in all our left experiments for Non-local.

We study different combinations of our CDM (Channel Down-sample Module) and CUM (Channel Up-sample Module) in Non-local blocks to check their performance. The results in Fig. 7, 8 and 9 show that the original Non-local model achieves 89.80% in Rank-1 and 81.87% in mAP with reduction rate $R = 2$ (90.0% and 82.8% in Rank-1 and mAP respectively in [8]), higher than CTN-Non-local (CDM-PymPool + CUM-Sample) model which achieves 89.44% and 81.71% in Rank-1 and mAP respectively.

TABLE 1. Comparison of the two additional feature maps F_{add} for PymPool with reduction rate $R = 2$. NL: Non-local, CTN: Channel Transformer Network, CDM: Channel Down-sample Module, CUM: Channel Up-sample Module, DSP: CDM-SegPool, DPP: CDM-PymPool, USA: CUM-Sample, UPP: CUM-PymPool, R1: Rank-1. Top three results are identified in red, blue and green respectively. All following tables and charts use the same settings.

Methods	F_{add}	R1	mAP
NL+DPP+USA	mean+min	89.29	81.80
	mean+max	88.89	81.32
	mean+var	89.29	81.73
	min+max	88.59	80.82
	min+var	89.14	81.67
	max+var	88.38	81.23

TABLE 2. Comparison of CUM with or without BatchNorm layer.

Block	R1	mAP
NL+DPP+USA	89.29	81.80
NL+DPP+USA+BN	89.44	81.71

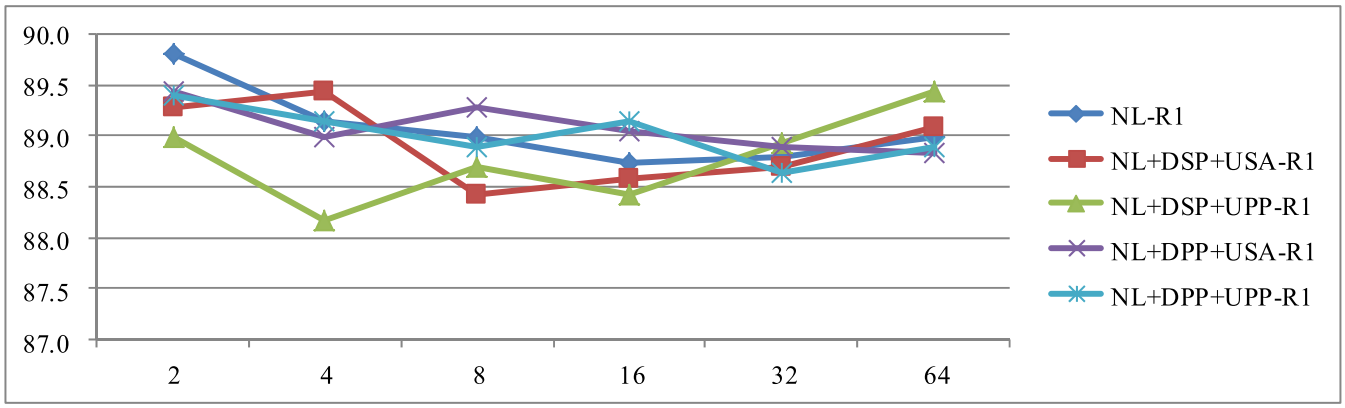


FIGURE 7. Performance chart of Non-local and CTN-Non-local in Rank-1 with various reduction rate *R*. Lateral axis: reduction rate *R*, Vertical axis: Rank-1 in percentage.

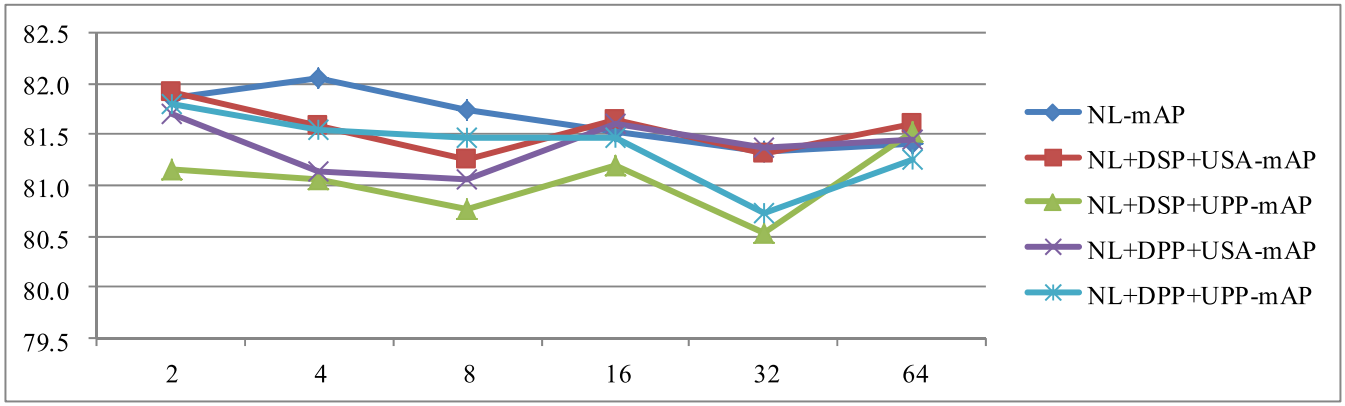


FIGURE 8. Performance chart of Non-local and CTN-Non-local in mAP with various reduction rate *R*. Lateral axis: reduction rate *R*, Vertical axis: mAP in percentage.

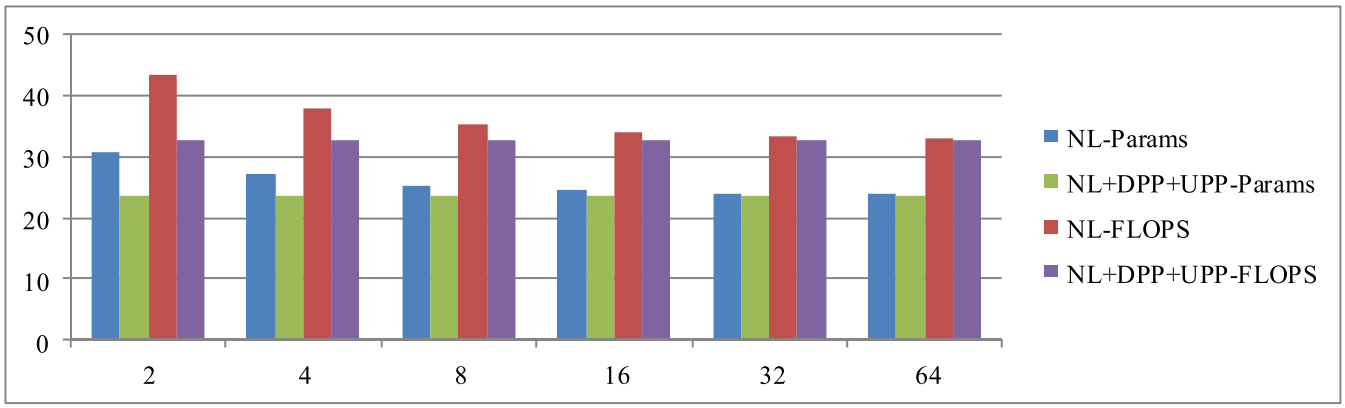


FIGURE 9. Computation complexity chart of Non-local and CTN-Non-local in parameters and FLOPS with various reduction rate *R*. Parameters and FLOPS of DSP, DPP, USA and UPP are very closed respectively, so we select DPP+UPP to represent CTN, following charts also use this setting. Lateral axis: reduction rate *R*, Vertical axis: parameters in M (Million) and FLOPS in G (Giga) calculated with thop.

Yet our CTN-Non-local model reduces parameters, FLOPS (floating-point operations per second) and whole training and evaluating time by 23.81%, 24.82% and 14.15% respectively while sacrificing 0.36% in Rank-1 and 0.16% in mAP with reduction rate 2.

As illustrated in Fig. 7 and 8, the performance of original Non-local model degrades along with bigger and bigger reduction rate *R* since the model's capability decays as shown in Fig. 9, while CTN-Non-local model keeps well or even better accuracy with lower parameters and FLOPS. That

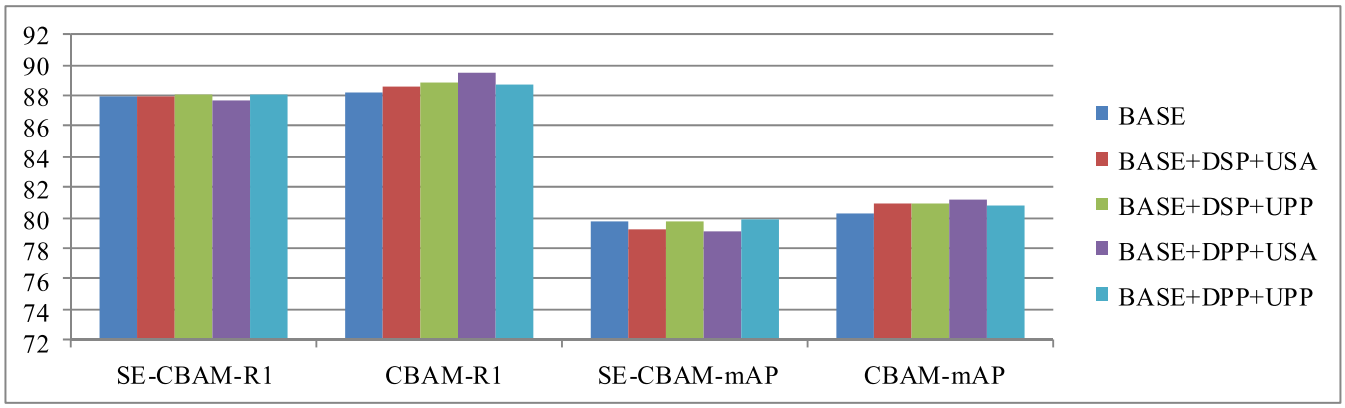


FIGURE 10. Performance chart of SE-CBAM, CTN-SE-CBAM, CBAM and CTN-CBAM in Rank-1 and mAP with reduction rate $R = 16$. BASE: SE-CBAM or CBAM without CTN, Vertical axis: Rank-1 and mAP in percentage.

means our CTN is valid and efficient, and it works very well in Non-local blocks. We also find that PymPool in CDM or CUM can help to improve the accuracy at a very slight cost in FLOPS.

CTN doesn't take any parameters itself, and its computation complexity is low and stable as shown in Fig. 9. It can make Non-local parameter-free if removing the last BatchNorm layer after CUM in CTN-Non-local block, whose reduction rate can be adjusted dynamically according to working condition without re-training.

2) ANALYSIS OF CTN IN CBAM

We follow [9] to treat CBAM as SE (Squeeze-and-Excitation) module [39] named SE-CBAM which is inserted in bottleneck blocks in ResNet50 [5]. And we compare it with CTN-SE-CBAM in which our CTN is used for decreasing and increasing channel dimension in the channel attention module of CBAM. The reduction rate R is set to 16 as [9]. The results are shown in Fig. 10 that the performance of CTN-SE-CBAM is slightly better than SE-CBAM with lower parameters and FLOPS. CTN improves Rank-1 and mAP of SE-CBAM by 0.15% and 0.18% when choosing CDM-PymPool and CUM-PymPool to reduce and increase channel dimension respectively.

We also try to insert original CBAM after layer1, layer2, layer3 and layer4 in ResNet50, not like SE-CBAM as se-module, and compare them with reduction rate 16 in Fig. 10. We find that just insert original CBAM after four layers of ResNet50 can improve Rank-1 and mAP by 0.20% and 0.55% than original SE-CBAM model as [9] on MARS, and save 7.03%, 0.17% and 21.43% in parameters, FLOPS and whole training and evaluating time respectively. Then, we compare CBAM with CTN-CBAM in which our CTN is used to change channel dimension in Fig. 10. The results show that CTN-CBAM (CDM-PymPool + CUM-Sample) can improve Rank-1 and mAP by 1.31% and 0.95% than original CBAM while cutting down 2.9% parameters and 0.03% FLOPS, which proves again our CTN's better performance.

3) ANALYSIS OF CTN IN COSAM

We follow [10] to insert COSAM after the last two layers of ResNet50, but reduce input channel dimension depending on the reduction rate R for unique style in our framework, not like [10] in which channel dimension is reduced to 256 fixedly. We also study different combinations of our CDM (Channel Down-sample Module) and CUM (Channel Up-sample Module) in COSAM blocks with various reduction rate R to check their performance. The results in Fig. 11, 12 and 13 show that the original COSAM model achieves 88.48% in Rank-1 and 80.36% in mAP with reduction rate $R = 2$ (83.7% and 77.2% in Rank-1 and mAP respectively in [10]), lower than CTN-COSAM (CDM-SegPool + CUM-Sample) model which achieves 89.55% and 80.79% in Rank-1 and mAP respectively. And our CTN-COSAM model largely reduces parameters, FLOPS and whole training and evaluating time by 25.35%, 8.04% and 44.86% respectively as well as improving 1.07% in Rank-1 and 0.43% in mAP with reduction rate $R = 2$. The huge reduction in training and evaluating time shows that our CTN has significant efficiency in transforming channel dimension especially for irregular ones (not the power of 2, as described in COSAM part of implementation details) than 1×1 convolutional layers. The results also show that original COSAM model takes more parameters and computation resources as shown in Fig. 13, nevertheless its performance is worse than CTN-COSAM. COSAM model also degrades along with increased reduction rate R, while CTN-COSAM keeps better accuracy with lower parameters and FLOPS as shown in Fig. 11 and 12. It proves our CTN is valid and efficient in COSAM blocks too.

4) ANALYSIS OF CTN IN VCAN

We improve CosegCA [11] for resolving more than two images simultaneously to adapt it in video-based person ReID and name it VCAN (Video Co-segment Attentive Network). Then, we study inserting VCAN after different layers of ResNet50 with reduction rate 2 and compare them in Table 3 to check which layers are most

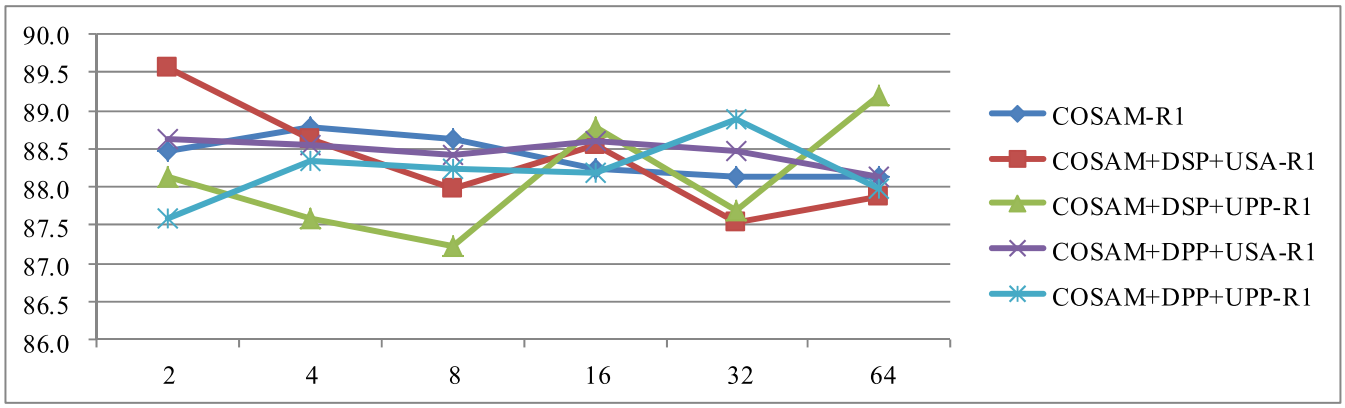


FIGURE 11. Performance chart of COSAM and CTN-COSAM in Rank-1 with various reduction rate *R*. Lateral axis: reduction rate *R*, Vertical axis: Rank-1 in percentage.

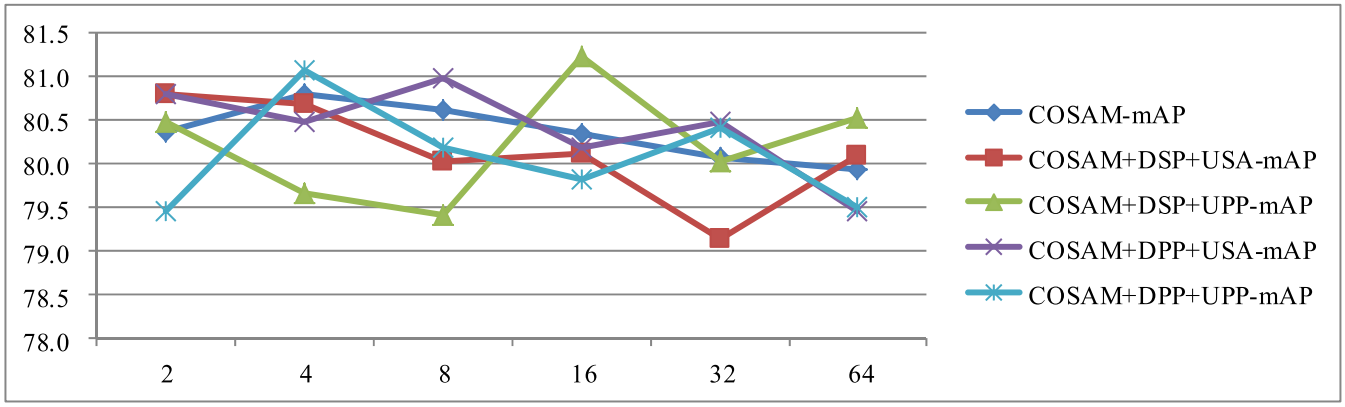


FIGURE 12. Performance chart of COSAM and CTN-COSAM in mAP with various reduction rate *R*. Lateral axis: reduction rate *R*, Vertical axis: mAP in percentage.

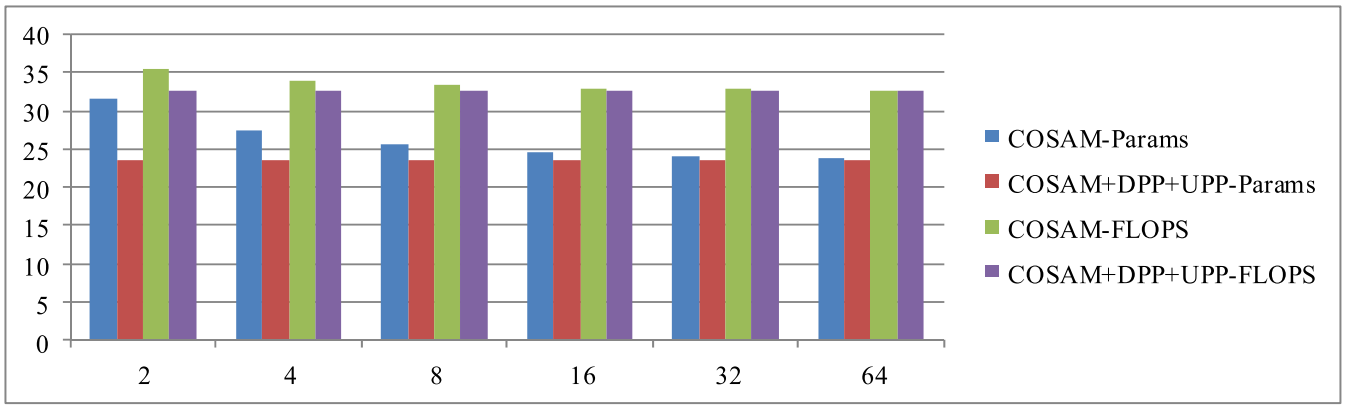


FIGURE 13. Computation complexity chart of COSAM and CTN-COSAM in parameters and FLOPS with various reduction rate *R*. Lateral axis: reduction rate *R*, Vertical axis: parameters in M (Million) and FLOPS in G (Giga) calculated with thop.

suitable for it. And we achieve significant improvement with 90.05% in Rank-1 and 82.16% in mAP when inserting VCAN after the last layer of ResNet50 with lower parameters and FLOPS. It shows that the last layer of ResNet50 is the best place for inserting VCAN in our framework. And we use it in all our left experiments for VCAN.

We also compare VCAN and CTN-VCAN in which channel transforming is replaced by our CTN in channel-wise attention module with various reduction rate *R*. The

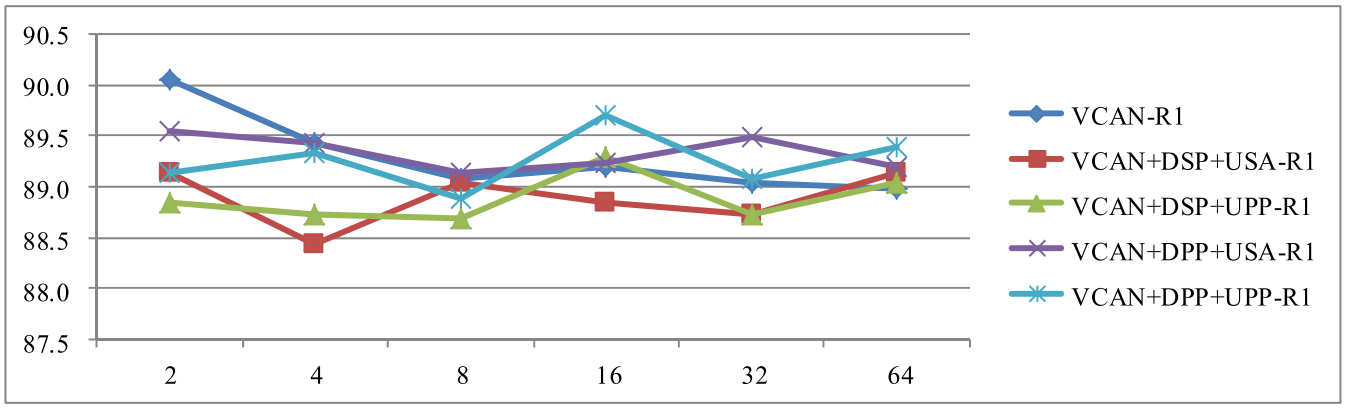


FIGURE 14. Performance chart of VCAN and CTN-VCAN in Rank-1. Lateral axis: reduction rate *R*, Vertical axis: Rank-1 in percentage.

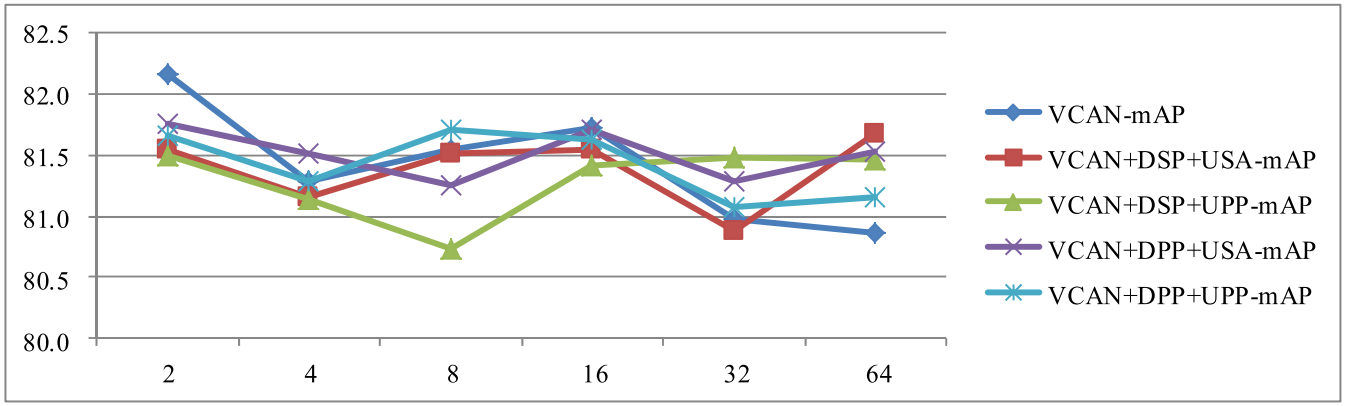


FIGURE 15. Performance chart of VCAN and CTN-VCAN in mAP. Lateral axis: reduction rate *R*, Vertical axis: mAP in percentage.

TABLE 3. Ablation study of VCAN after different layers in ResNet50 with reduction rate 2. Params: Parameters in M (Million), FLOPS: Floating-point operations per second in G (Giga), Elapsed: Whole training and evaluating time in hour.

layers	R1	mAP	Params	FLOPS	Elapsed
1,2,3,4	89.09	81.37	29.088	32.615	3.86
2,3,4	88.94	81.22	29.023	32.610	2.93
3,4	89.09	81.62	28.760	32.606	2.68
4	90.05	82.16	27.710	32.597	2.57

results in Fig. 14, 15 and 16 show that the original VCAN model achieves 90.05% in Rank-1 and 82.16% in mAP with reduction rate $R = 2$, higher than CTN-VCAN (CDM-PymPool + CUM-Sample) model which achieves 89.55% and 81.75% in Rank-1 and mAP respectively. However, our CTN-VCAN model reduces parameters and FLOPS by 15.15%, 0.10% respectively while dropping 0.50% in Rank-1 and 0.41% in mAP with reduction rate 2. It shows that our VCAN itself is a lightweight but efficient model with low parameters and FLOPS. We can also find that the accuracy of VCAN degrades gradually with progressively increased reduction rate *R* just like Non-local and COSAM, whereas CTN-VCAN keeps more stable with well or even better performance not only in accuracy but also in parameters and FLOPS.

5) ANALYSIS OF PYMPOOL IN CTN

Different scale of pooling operations on channel dimension can help the model get channel-wise multi-scale information, since semantic information implicit in channels is affected under pooling operations. We count the comparison experiments of Non-local, CBAM, COSAM, VCAN and their corresponding CTN embedding models respectively in Fig.17. The average percentage of the best models that contain PymPool are 75% and 66.67% according to Rank-1 and mAP respectively. It shows that our PymPool is generally better than direct transformation.

C. COMPARISON WITH STATE-OF-THE-ARTS

We compare our proposed VCAN, CTN-VCAN, CTN-Non-local, CTN-COSAM and CTN-CBAM with the state-of-the-art approaches on MARS dataset in Table 4. Our VCAN and CTN-VCAN achieve significant performance with 90.05% and 89.70% in Rank-1, 82.16% and 81.63% in mAP respectively with lower parameters and FLOPS than

TABLE 4. Comparison with the state-of-the-art video-based person ReID methods on MARS. Elapsed: Whole training and evaluating time in hour. Parameters and FLOPS are calculated with thop in unit M (Million) and G (Giga) respectively.

Methods	Source	R1	mAP	Params	FLOPS	Elapsed
MARS [48]	ECCV16	65.3	47.6	-	-	-
SeeForest [26]	CVPR17	70.6	50.7	-	-	-
TriNet [25]	arXiv17	79.8	67.7	-	-	-
MultiShot [31]	CVPR18	71.2	-	-	-	-
ETAP-Net [30]	CVPR18	80.8	67.4	-	-	-
STAN [29]	CVPR18	82.3	65.8	-	-	-
Revisit [28]	arXiv18	83.3	76.7	-	-	-
Snippet [27]	CVPR18	86.3	76.1	-	-	-
COSAM [10]	ICCV19	84.9	79.9	-	-	-
STICA [32]	arXiv19	86.0	80.8	-	-	-
GLTR [33]	ICCV19	87.02	78.47	-	-	-
STE-NVAN [8]	BMVC19	88.9	81.2	-	-	-
NVAN [8]	BMVC19	**90.0**	**82.8**	30.871M	43.331G	4.06h
AdaptiveGraph [34]	TIP20	89.5	81.9	-	-	-
VCAN (ours)	-	**90.05**	**82.16**	27.710M	32.597G	2.57h
CTN-VCAN (ours)	-	89.70	81.63	**23.512M**	**32.563G**	**2.55h**
Non-local (in our training)	-	**89.80**	**81.87**	30.871M	43.334G	4.06h
CTN-Non-local (ours)	-	89.44	81.53	23.520M	32.576G	3.05h
COSAM (in our training)	-	88.79	80.80	27.566M	34.044G	4.87h
CTN-COSAM (ours)	-	89.55	80.79	23.515M	32.564G	2.79h
CBAM (in our training)	-	88.13	80.24	24.213M	32.582G	2.64h
CTN-CBAM (ours)	-	89.44	81.19	23.512M	32.571G	2.62h

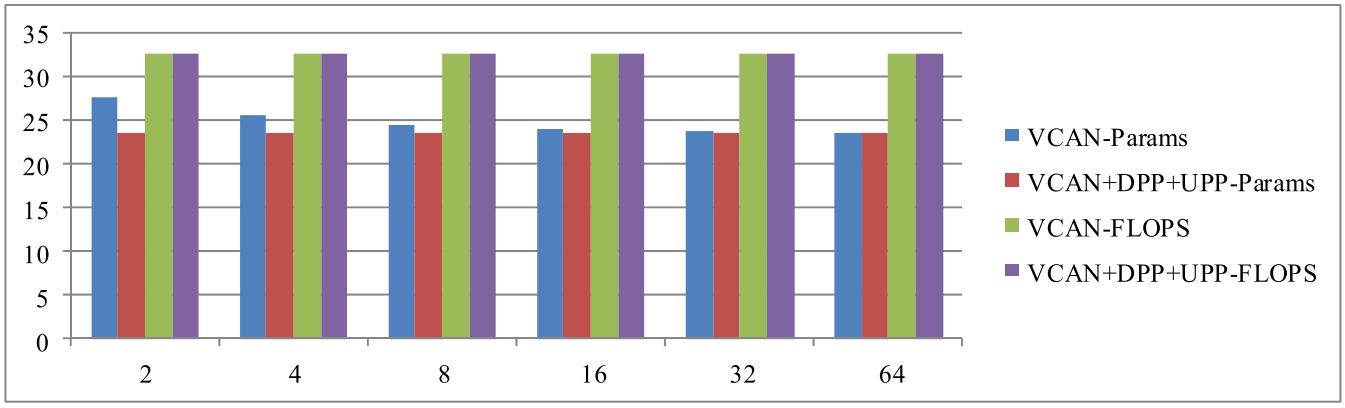


FIGURE 16. Computation complexity chart of VCAN and CTN-VCAN in parameters and FLOPS. Lateral axis: reduction rate *R*, Vertical axis: parameters in M (Million) and FLOPS in G (Giga) calculated with thop.

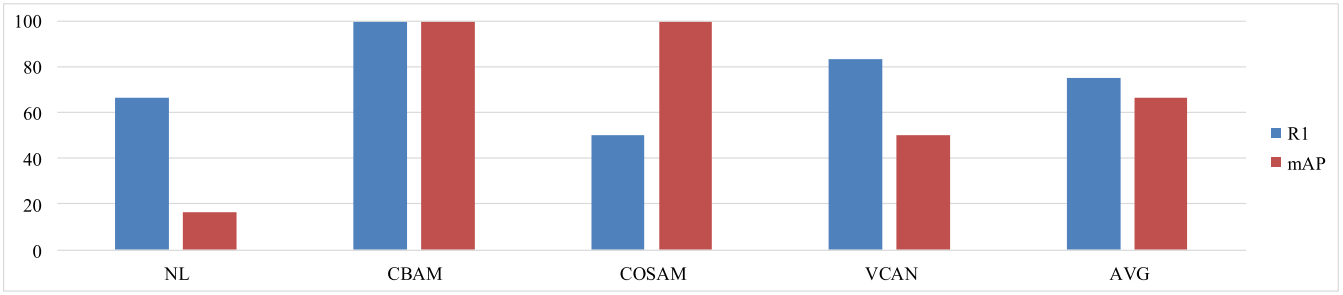


FIGURE 17. The percentage of the best models that contain PymPool according to Rank-1 and mAP via above comparison experiments. Vertical axis: The percentage of the best models that contain PymPool. AVG: Average percentage of Non-local, CBAM, COSAM and VCAN.

NVAN [8] without re-ranking [49]. It's worth to notice that NVAN leverages Non-local [7] blocks to improve recognition accuracy which costs too much computation resources, while our VCAN and CTN-VCAN employ co-segmentation

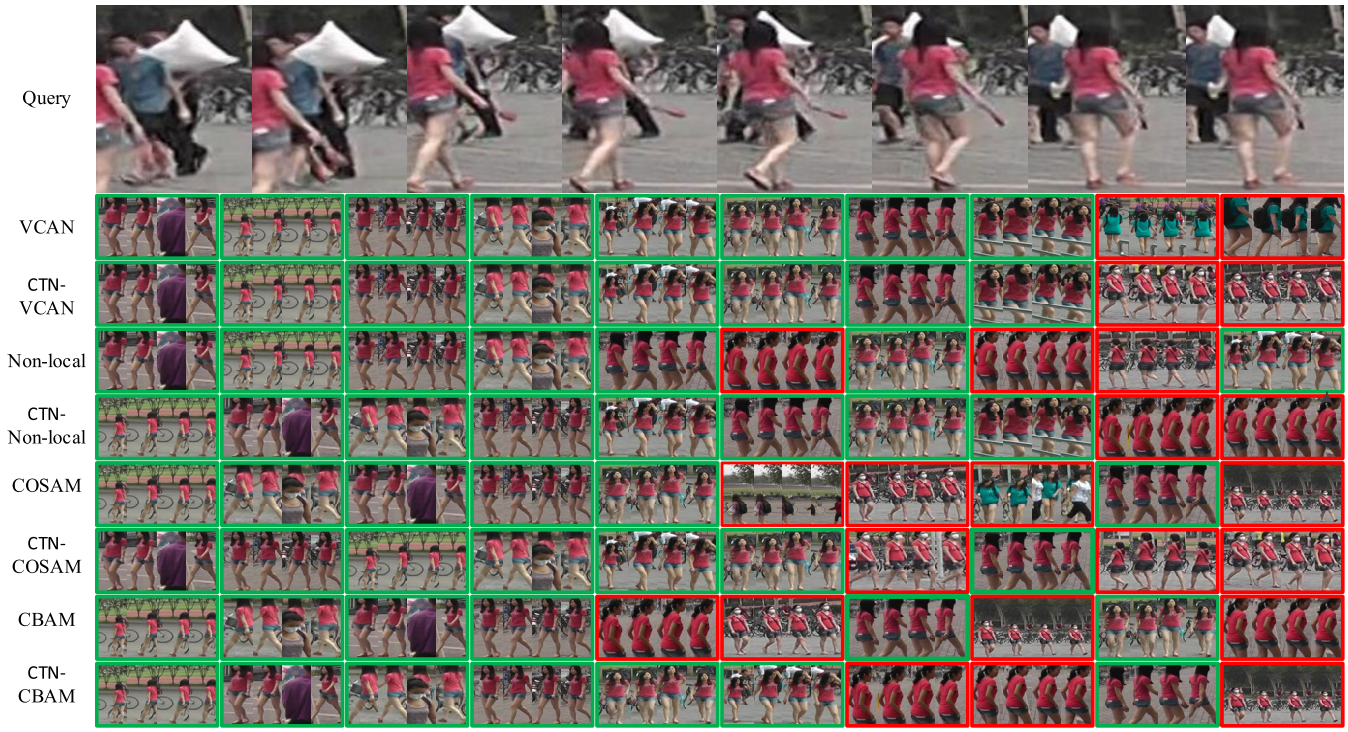


FIGURE 18. Visualization of person ReID on MARS dataset with VCAN, Non-local, COSAM, CBAM and their CTN embedding models. The first row is the query sequence including 8 frames, and following rows are sorted from Rank-1 to Rank-10 (left to right) presented by 4 frames in each one for better exhibition. The match results are indicated by green boxes and red boxes for succeed and failed matches respectively.

block based attention mechanism to get competitive results with lower parameters and FLOPS. Parameters, FLOPS and elapsed time of NVAN [8] are calculated in our framework for fair comparison.

COSAM [10] also uses co-segmentation inspired attention mechanism with a normalized cross correlation layer and a summarization layer to improve the performance of video-based person ReID. It achieves 88.79% and 80.80% in Rank-1 and mAP respectively in our training, higher than published result in [10] with 3.89% and 0.90%, and lower than our VCAN with 1.26% and 1.36% in Rank-1 and mAP respectively. And it costs 89.49% more training and evaluating time than our VCAN. Our CTN-COSAM reaches 89.55% in Rank-1 (higher than COSAM with 0.76%) and 80.79% in mAP (lower than COSAM with 0.01%). The parameters, FLOPS and whole training and evaluating time of our CTN-COSAM are lower than COSAM with 14.70%, 4.35% and 42.71% respectively. The large time reduction shows that our CTN has excellent efficiency in transforming channel dimension especially for irregular channel dimension (not the power of 2) than 1×1 convolutional layers and fully connected layers.

Our CTN-CBAM model also gets high performance with 89.44% and 81.19% in Rank-1 and mAP respectively. It improves Rank-1 and mAP by 1.31% and 0.95% than original CBAM while reducing 2.9% in parameters and 0.03% in FLOPS.

In brief, the results show that our parameter-free Channel Transformer Network (CTN) and Video Co-segment Attentive Network for person ReID (VCAN) achieve outstanding performance in accuracy and computation complexity on MARS dataset.

D. RESULTS VISUALIZATION

We present the match results for a probe sample with various models from our experiments in Fig. 18. It shows that all these models have the capability to resolve occasional occlusion across multi-frames, and our VCAN reaches the best match result. Our CTN embedding models have relative or even better matching performance than original ones as well as lower parameters and computation complexity.

V. CONCLUSION

We propose a novel parameter-free Channel Transformer Network (CTN) to replace 1×1 convolutional layers or fully connected layers for increasing and decreasing channel dimension in a CNN attention or transform block with lower computational complexity. We also introduce a Video Co-segment Attentive Network (VCAN) for person ReID which leverages co-segmentation mechanism to extract salient feature information of common pedestrian and accessories across multiple video frames in a sequence. We then embed CTN into Non-local, CBAM, COSAM and VCAN to replace original 1×1 convolutional layers or fully connected layers in the same video-based person ReID framework. Extensive experiments on MARS dataset show that our CTN and VCAN achieve state-of-the-art performance in accuracy and com-

putation complexity in video-based person ReID. Our CTN has outstanding efficiency in transforming channel dimension especially for irregular one (not the power of 2) than 1×1 convolutional layers and fully connected layers. It's also easy to help other blocks to be parameter-free modules which are convenient to adjust reduction rate dynamically according to computation resources without re-training. We will embed our CTN in other modules and vision tasks like image classification and object detection etc. for further research in future.

REFERENCES

[1] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, Jan. 1962.

[2] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*. Berlin, Germany: Springer, 1982, pp. 267–285.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[6] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1492–1500.

[7] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[8] C.-T. Liu, C.-W. Wu, Y.-C. Frank Wang, and S.-Y. Chien, "Spatially and temporally efficient non-local attention network for video-based person re-identification," 2019, *arXiv:1908.01683*. [Online]. Available: http://arxiv.org/abs/1908.01683

[9] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[10] A. Subramaniam, A. Nambiar, and A. Mittal, "Co-segmentation inspired attention networks for video-based person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 562–572.

[11] H. Chen, Y. Huang, and H. Nakayama, "Semantic aware attention based deep object co-segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 435–450.

[12] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, and Y. Yang, "Improving person re-identification by attribute and identity learning," 2017, *arXiv:1703.07220*. [Online]. Available: http://arxiv.org/abs/1703.07220

[13] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun, "AlignedReID: Surpassing human-level performance in person re-identification," 2017, *arXiv:1711.08184*. [Online]. Available: http://arxiv.org/abs/1711.08184

[14] Y. Guo and N.-M. Cheung, "Efficient and deep person re-identification using multi-level similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2335–2344.

[15] L. He, J. Liang, H. Li, and Z. Sun, "Deep spatial feature reconstruction for partial person re-identification: Alignment-free approach," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7073–7082.

[16] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1062–1071.

[17] Q. Ke, M. Bennamoun, H. Rahmani, S. An, F. Sohel, and F. Boussaid, "Identity adaptation for person re-identification," *IEEE Access*, vol. 6, pp. 48147–48155, 2018.

[18] M. S. Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen, "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 420–429.

[19] S. Zhang, Y. He, J. Wei, S. Mei, S. Wan, and K. Chen, "Person re-identification with joint verification and identification of identity-attribute labels," *IEEE Access*, vol. 7, pp. 126116–126126, 2019.

[20] C. Dai, J. Feng, and R. Zhou, "Learning domain-specific features from general features for person re-identification," *IEEE Access*, vol. 8, pp. 155389–155398, 2020.

[21] H. Wang, G. Wang, Y. Li, D. Zhang, and L. Lin, "Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep mis-ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 342–351.

[22] J. Chen, Y. Wang, and Y. Y. Tang, "Person re-identification by exploiting spatio-temporal cues and multi-view metric learning," *IEEE Signal Process. Lett.*, vol. 23, no. 7, pp. 998–1002, Jul. 2016.

[23] N. McLaughlin, J. M. del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1325–1334.

[24] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang, "Person re-identification via recurrent feature aggregation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 701–716.

[25] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: http://arxiv.org/abs/1703.07737

[26] Z. Zhou, Y. Huang, W. Wang, L. Wang, and T. Tan, "See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4747–4756.

[27] D. Chen, H. Li, T. Xiao, S. Yi, and X. Wang, "Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1169–1178.

[28] J. Gao and R. Nevatia, "Revisiting temporal modeling for video-based person ReID," 2018, *arXiv:1805.02104*. [Online]. Available: http://arxiv.org/abs/1805.02104

[29] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 369–378.

[30] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5177–5186.

[31] J. Zhang, N. Wang, and L. Zhang, "Multi-shot pedestrian re-identification via sequential decision making," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6781–6789.

[32] T. Isobe, J. Han, F. Zhu, Y. Li, and S. Wang, "Intra-clip aggregation for video person re-identification," 2019, *arXiv:1905.01722*. [Online]. Available: http://arxiv.org/abs/1905.01722

[33] J. Li, S. Zhang, J. Wang, W. Gao, and Q. Tian, "Global-local temporal representations for video person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 3958–3967.

[34] Y. Wu, O. E. F. Bourahla, X. Li, F. Wu, Q. Tian, and X. Zhou, "Adaptive graph representation learning for video person re-identification," *IEEE Trans. Image Process.*, vol. 29, pp. 8821–8830, 2020.

[35] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 737–744.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[38] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2019, pp. 593–602.

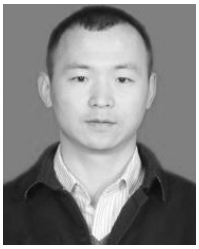
[39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[40] W. Li, O. H. Jafari, and C. Rother, "Deep object co-segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 638–653.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[42] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "DeepCO^3: Deep instance co-segmentation by co-peak search and co-saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8846–8855.

[43] K. Zhang, J. Chen, B. Liu, and Q. Liu, "Deep object co-segmentation via spatial-semantic network modulation," in *Proc. AAAI*, 2020, pp. 12813–12820.
[44] W.-C. Hung, V. Jampani, S. Liu, P. Molchanov, M.-H. Yang, and J. Kautz, "SCOPS: Self-supervised co-part segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 869–878.
[45] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention Siamese networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3623–3632.
[46] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.
[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[48] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 868–884.
[49] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1318–1327.



FUPING ZHANG (Student Member, IEEE) received the B.S. degree in communication engineering and the M.S. degree in electronic and communication engineering from the East China University of Science and Technology, Shanghai, China, in 2003 and 2017, respectively. He is currently pursuing the Ph.D. degree in signal and information processing with the Shanghai Advanced Research Institute, Chinese Academy of Sciences. He joined the Safety and Emergency Laboratory, Shanghai Advanced Research Institute, CAS, in 2014. His research interests include deep learning, computer vision, with a focus on person re-identification, and their applications in embedded systems.

PENGCHENG ZHAO (Student Member, IEEE) received the B.S. degree in electronic information engineering from Southwest Jiaotong University, Chengdu, China, in 2018. He is currently pursuing the joint M.S. degree with the Shanghai Advanced Research Institute, Chinese Academy of Sciences, and Shanghai University. His research interests include deep learning and its application in heterogeneous face recognition.

JIANMING WEI (Member, IEEE) received the Ph.D. degree in communication and information system from Shanghai University, Shanghai, China, in 2005. He is currently a Researcher with the Shanghai Advanced Research Institute, Chinese Academy of Sciences. He is involved in the system design for collaborative processing and information fusion and the application in emergency rescue for public safety. His research interests include the design of wireless sensor networks for signal processing and communication systems.