Postprint: Action Recognition Algorithm Based on DRN and Faster R-CNN Fusion Model

Authors: Yang Nan, Yang Shen, Thünen

Date: 2018-09-12T00:00:00+00:00

Abstract

This study addresses the issue that traditional single-person action recognition algorithms are susceptible to influences from variations in pedestrian morphology, background, and illumination. Based on the classification effectiveness of the Dilated Convolutional Residual Network (DRN) and the accuracy of the object detection network Faster R-CNN in target tracking, a fused network model combining DRN and Faster R-CNN is proposed. This model incorporates DRN' s dilated convolutional residual blocks into Faster R-CNN to replace the original general convolutional layers. Two improvements are made to the fused model: adding a batch normalization layer before each layer, and replacing some two-layer residual blocks with three-layer dilated convolutional residual blocks. Experimental results demonstrate that the three fused network recognition algorithms achieve higher mAP on the Olympic Sports Dataset compared to other action recognition algorithms. Among them, the fused model containing three-layer dilated convolutional residual blocks exhibits the best recognition performance, with mAP reaching 78.9%.

Full Text

Preamble

Title: Behavior Recognition Algorithm Based on DRN and Faster R-CNN Fusion Model

Authors: Yang Nan, Yang Shen, Du Neng
Affiliation: School of Information Science & Engineering, Wuhan University of Science and Technology, Wuhan 430081, China

Abstract: Traditional single-person behavior recognition algorithms are susceptible to variations in pedestrian pose, background clutter, and lighting conditions. Leveraging the classification accuracy of dilated convolutional residual networks (DRN) and the target tracking precision of Faster R-CNN, this paper

proposes a novel fusion network model that integrates DRN' s dilated convolutional residual blocks into Faster R-CNN to replace its standard convolutional layers. Two key improvements are introduced: (1) batch normalization layers are added before each network layer, and (2) three-layer dilated convolutional residual blocks replace select two-layer residual blocks. Experimental results on the Olympic Sports Dataset demonstrate that the three proposed fusion network variants achieve higher mean average precision (mAP) than existing behavior recognition methods. The three-layer dilated convolutional residual block fusion model exhibits the best recognition performance, attaining an mAP of 78.9%.

Keywords: behavior recognition; DRN; Faster R-CNN

0 Introduction

Human behavior recognition has attracted significant attention in applications such as intelligent video surveillance, video retrieval, and human-computer interaction [1]. Despite extensive research efforts worldwide, achieving practical accuracy remains challenging due to complex backgrounds, illumination variations, appearance differences, and diverse motion patterns. Conventional behavior recognition approaches based on machine learning typically consist of feature extraction and behavior classification stages. Common feature extraction methods include LBP (Local Binary Patterns), HOG (Histogram of Oriented Gradients) [3], and SIFT (Scale-Invariant Feature Transform) [4], while popular classifiers include decision trees and Support Vector Machines (SVM) [5,6]. These traditional methods suffer from incomplete feature representation and high manual effort requirements.

In recent years, convolutional neural networks (CNNs) have demonstrated remarkable success in image classification and object recognition tasks. At the 2012 ImageNet ILSVRC competition, Krizhevsky et al.' s AlexNet [7] achieved a top-5 error rate of 16%, drawing widespread attention to CNNs in computer vision. Subsequent competitions saw the emergence of various CNN architectures, with He et al.' s ResNet [8] winning ILSVRC 2015. ResNet' s skip connections effectively mitigate the "degradation" problem in very deep networks. Building upon ResNet, the Dilated Residual Network (DRN) [9] incorporates dilated convolutions to enlarge the receptive field without pooling layers, thereby preserving spatial resolution and retaining maximal detail from input images. This approach enhances classification performance compared to standard ResNet.

Object recognition can be decomposed into target tracking and behavior classification. Traditional machine learning-based behavior recognition methods often perform direct classification without explicit target tracking, such as temporal and spatial modeling [10,11], manual feature extraction followed by classification [12], or HOG+CNN feature extraction with temporal ordering and SVM classification [13]. In contrast, algorithms that jointly perform target tracking and classification have achieved unprecedented results on various benchmarks

[14-17]. Girshick et al.' s R-CNN [15] improved mean average precision (mAP) from 34.3% to 66%, though training was complex and time-consuming. Subsequent work introduced Fast R-CNN [16] and Faster R-CNN [17], which employ CNNs for both target tracking and classification, significantly reducing detection time while improving mAP. This paper combines DRN' s classification accuracy with Faster R-CNN' s tracking precision to create a unified network for behavior recognition.

---

1 Deep Convolutional Neural Network Models

CNNs extend standard neural networks with convolutional layers for feature extraction and pooling layers for downsampling. In convolutional layers, each neuron connects only to a local region of the previous layer. Each convolutional layer contains multiple filters (feature maps), with each filter comprising n×n neurons ($n \geq 1$). These filters share weights across the input, where the shared weights constitute the convolution kernel. This section introduces the DRN and Faster R-CNN architectures.

1.1 DRN Network

DRN is a variant of ResNet, a residual network with skip connections developed by He et al. As network depth increases, model performance plateaus and further depth can cause "vanishing gradients," "exploding gradients," or "degradation" —where training and test accuracy decline despite increased capacity, unrelated to overfitting. To address degradation, DRN proposes a residual structure (Fig. 1), where a two-layer residual learning module transforms the mapping function from H(x) to F(x) + x, with x as input and ReLU as the activation function.

DRN extends ResNet by incorporating dilated convolutions, which maintain consistent receptive fields and output dimensions without pooling. As illustrated in Fig. 2, (a) shows 1-dilated convolution (equivalent to standard convolution), while (b) and (c) demonstrate 2-dilated and 4-dilated convolutions, respectively. By replacing pooling layers, dilated convolutions preserve spatial resolution while expanding receptive fields, maximizing retention of input image details and improving classification performance over ResNet.

1.2 Faster R-CNN

To address the computational bottleneck of selective search in R-CNN and Fast R-CNN, Faster R-CNN introduces a Region Proposal Network (RPN) [17] to generate target proposals efficiently. The RPN structure is shown in Fig. 3. Before the first convolution, anchors of three scales and three aspect ratios are generated at each spatial location, yielding nine target boxes per location. These boxes scan the entire image with a stride of 16, producing 20,000–40,000

candidate boxes. After removing boundary-crossing boxes, 6,000–10,000 remain for RPN training.

The RPN network begins with a 3$×$3 convolutional layer using 512 filters, stride 1, and 'SAME' padding to preserve dimensions, followed by ReLU activation. Two fully connected layers, "rpn_{{{cls}}*{{score}}}" *and* "*rpn*{{{bbox}}*{{pred}}},*" *output foreground scores and bounding box regression information. Dimension transformation layers with "reshape" adjust tensor dimensions as needed.* "*rpn*{{{cls}}_{{prob}}}" applies softmax, while "proposal" generates final proposals by removing boundary boxes, applying Non-Maximum Suppression (NMS) [19] based on foreground scores, and using regression adjustments. The final output consists of 256 proposals.

The loss functions for RPN are defined as follows. Let i be the anchor index, N be the batch size (N = 10), p_i be the predicted probability that anchor i contains a foreground object, and p_i* be the ground-truth label (1 for foreground, 0 for background). Let t_i be the predicted bounding box coordinates (top-left and bottom-right) and t_i* be the ground-truth coordinates.

The total loss combines classification and regression losses:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

where λ is a weighting parameter. The classification loss L_{cls} is log loss:

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1-p_i^*)(1-p_i)]$$

The regression loss L_{reg} uses the robust smooth L1 loss from [13]:

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$$

where

$$R(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

2 Data Preprocessing and Fusion Network Structure

2.1 Data Preprocessing

We employ the Olympic Sports Dataset, released by Stanford University in 2010, comprising 16 sport categories (basketball, weightlifting, long-distance running, etc.) with 50 videos per category (Fig. 4). From the color dataset, we select 5,000 images, annotate them with ground-truth bounding boxes, and

generate 5,000 mirrored versions, creating a training set of 10,000 images. An additional 2,000 images serve as a validation set for hyperparameter tuning, and 5,000 images constitute the test set. Since validation and testing focus solely on behavior recognition accuracy without requiring bounding box evaluation, these sets do not require manual ground-truth annotations.

2.2 Fusion Network Architecture

The fusion network structure is depicted in Fig. 5. The orange components correspond to the RPN from Faster R-CNN (Section 1.2). The golden "Roi_{data}" layer stores region proposals from the RPN, while "Roi_{pool5}" is the Region of Interest (RoI) pooling layer that standardizes all features to 7$×$7 for the fully connected layers.

The core fusion model (dashed box) replaces VGG16 [20] in the original Faster R-CNN with DRN. The first layer is a 7$×7convolutionwithpadding2andstride2, halvingtheinputsize.Thisisfoll
poolinglayers("pool1", "pool2", "pool3")with2×$2 kernels, stride 2, and 'SAME' padding, each reducing spatial dimensions by half. After "pool3" , the network exclusively uses dilated convolutional residual blocks (DR1–DR12) without additional pooling. Blocks DR1–DR6 use 1-dilated convolutions, while DR7–DR12 employ 2-dilated or 4-dilated convolutions to increase receptive fields without further downsampling.

Each dilated residual block contains two 3$×$3 convolutional layers with stride 1 and 'SAME' padding. The number of filters varies per block: 16 for DR1, 32 for DR2, 64 for DR3–DR4, 128 for DR5–DR6, 256 for DR7–DR8, and 512 for DR9–DR12. The dilation rate is indicated after each block name (e.g., "DR1 16 1" denotes 16 filters and dilation rate 1).

Following the residual blocks, two fully connected layers with 4,096 neurons each include dropout layers to mitigate overfitting. The "cls_{score}" layer has 17 neurons (16 behavior classes + background), with "cls_{prob}" applying softmax to output class probabilities. The "bbox_{pred}" layer has 68 neurons providing bounding box regression offsets for all 17 classes. Other connections remain consistent with the original Faster R-CNN.

---

3 Improved Fusion Models

This section addresses potential vanishing/exploding gradient issues through two enhancements.

3.1 Fusion Model with Batch Normalization

Deep networks can suffer from vanishing or exploding gradients, causing distribution shifts in later layers that slow training as the network continuously adapts to new data distributions. Batch Normalization (BN) [28] prevents this

by adding learnable normalization layers before each network layer. For a batch input {x_1, ···, x_m}, BN computes:

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta$$

where _B and σ_B^2 are the batch mean and variance, is a small constant (0.0001), γ and β are learnable parameters, and y_i is the BN layer output. During testing, BN uses population statistics:

$$E[x] \leftarrow E_B[\mu_B], \quad Var[x] \leftarrow \frac{m}{m-1}E_B[\sigma_B^2]$$

In the improved fusion model (Fig. 5), BN layers are added before every layer in the dashed box, including before both convolutional layers in each DR1–DR12 residual block and before all fully connected layers. BN layers are also added to the RPN components.

3.2 Fusion Model with Three-layer Residual Blocks

The original fusion model uses two-layer residual blocks. Inspired by [9], we replace DR1–DR6 with three-layer dilated residual blocks (Fig. 6) while keeping post-pool3 blocks unchanged. In this design, the first and third layers are $1\$\times 1 convolutions with dilation 1, while the middle layer uses $3\times\$3$ convolutions with variable dilation rates. All layers have stride 1 and 'SAME' padding. This modification increases network depth, requiring BN layers before each layer as described in Section 3.1. Three-layer residual blocks demonstrate superior performance in deep networks compared to their two-layer counterparts.

---

4 Experiments and Results

4.1 Fusion Network Training

Training employs alternating optimization on a GPU-based TensorFlow implementation, requiring approximately three days. After training, test images are processed to generate multiple bounding boxes with

class scores. NMS with threshold 0.7 removes redundant boxes, retaining only those with class probabilities exceeding 0.8. Fig. 7 shows a $600\$\times450 basketball image where two players are correctly detected with behavior labels and confidence scores. Fig.$ hammer throw image where the action is detected, though other individuals fail to exceed the 0.8 threshold and remain unmarked.

The trained model achieves 77.2% mAP on the test set. Comparative results against other methods on the Olympic Sports Dataset are shown in Table 1. The proposed fusion model outperforms existing behavior recognition algorithms, the original Faster R-CNN, and YOLO [26] and SSD [27] implementations on this dataset.

Table 1: mAP comparison of proposed and existing algorithms

Algorithm	mAP
Proposed fusion model	77.2%
Reference [21]	69.2%
Reference [22]	76.4%
Reference [23]	73.7%
Reference [24]	72.3%
Reference [25]	75.1%
Reference [10]	72.1%
Original Faster R-CNN	67.3%
YOLO [26]	76.4%
SSD [27]	76.8%

4.2 Improved Fusion Models

Both improved models are trained identically to the original fusion model, with only the enhanced components modified. Table 2 compares their performance.

Table 2: mAP comparison of improved fusion models

Model	mAP
Original fusion model	77.2%
BN-added fusion model	78.5%
Three-layer residual block fusion model	78.9%

The BN-enhanced model achieves 78.5% mAP, confirming that the original model experienced minor gradient issues that BN successfully mitigated. The three-layer residual block model performs best at 78.9% mAP, demonstrating that deeper residual blocks are more effective for this classification task.

5 Conclusion

This paper proposes a fusion network that integrates DRN' s dilated convolutional residual blocks into Faster R-CNN' s shared convolutional layers, leveraging DRN' s classification strengths and Faster R-CNN' s tracking precision. Two enhancements—adding BN layers and employing three-layer residual blocks —yield further improvements. All three fusion models surpass existing behavior recognition methods and the original Faster R-CNN on the Olympic Sports Dataset, with the three-layer residual block model achieving the highest mAP of 78.9%. However, the detection speed remains limited at approximately 5 frames per second, significantly slower than YOLO (~45 fps) and SSD (~58 fps). Future work will focus on accelerating detection while maintaining recognition accuracy.

References

[1] Mei Yang, Wang Yongxiong, Qing Qi, et al. A method of human behavior recognition based on key frame [J]. Optical Technology, 2017, 43(4): 323-328.

[2] Wang Zhongmin, Zhang Zong, Heng Xia, et al. Research on new human behavior recognition method combining CNN with decision tree [J]. Application Research of Computers, 2017, 34(12): 3569-3572.

[3] Xiao Yuling. Human behavior recognition combined with HOG//HOF cascade features and multilayer classifier [J]. Computer Engineering and Design, 2017, 38(9): 2567-2572.

[4] Quy N H, Quoc N H, Anh N T L, et al. 3D human face recognition using sift descriptors of face' s feature regions [C]// Proc of the 1st IEEE International Conference on Computer Communication and the Internet. Cham: Springer, 2015: 117-126.

[5] Ayumi V, Fanany M I. A comparison of SVM and RVM for human action recognition [C]// Proc of International Conference on Industrial Internet of Things. 2015.

[6] Prasad S, Ramkumar B. Passive copy-move forgery detection using SIFT, HOG and SURF features [C]// Proc of IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology. Cham: Springer, 2017: 706-710.

[7] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [C]// Proc of International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012: 1097-1105.

[8] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C]// Computer Vision and Pattern Recognition. 2016.

[9] Yu Fisher, Koltun V, Funkhouser T. Dilated residual networks [J]. Computer Science, 2017: 636-644.

[10] Niebles J C, Chen C W, Li Feifei. Modeling temporal structure of decomposable motion segments for activity classification [C]// Proc of European Conference on Computer Vision. Springer-Verlag, 2010: 392-405.

[11] Koller D, Tang K, Li FeiFei. Learning latent temporal structure for complex event detection [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2012: 1250-1257.

[12] Liu Jingen, Kuipers B, Savarese S. Recognizing human actions by attributes [C]// Computer Vision and Pattern Recognition. IEEE, 2011: 3337-3344.

[13] Liu Fang, Xu Xiangmin, Qing Chunmei. Temporal order information for complex action recognition [C]// Proc of IEEE International Conference on Consumer Electronics-China. IEEE, 2017.

[14] Razavian A S, Azizpour H, Sullivan J, et al. CNN features off-the-shelf: an astounding baseline for recognition [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition Workshops. Washington DC: IEEE Computer Society, 2014: 512-519.

[15] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2014: 580-587.

[16] Girshick R. Fast R-CNN [J]. Computer Science, 2015.

[17] Ren Shaoqing, He Kaiming, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.

[18] Zhang Yongbing, Sun Lulu, Wang Xingzhen. ReLU convolutional neural network-based image denoising method: CN 106204468 A [P]. 2016.

[19] Chen Jinghui, Ye Xining. Improvement of non-maximum suppression algorithm in pedestrian detection [J]. Journal of East China University of Science and Technology: Natural Science Edition, 2015, 41(3): 371-378.

[20] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. Computer Science, 2014.

[21] Wang L, Yu Qiao, Tang X. Latent hierarchical model of temporal structure for complex activity classification [J]. IEEE Trans on Image Processing, 2014, 23(2): 810-822.

[22] Yan Shiyang, Smith J S, Lu Wenjin, et al. CHAM: action recognition using convolutional hierarchical attention model [J]. Computer Science. 2017.

[23] Sharma S, Kiros R, Salakhutdinov R. Action recognition using visual attention [J]. Computer Science, 2015.

[24] Liu Fang, Xu Xiangmin, Qiu Shuoyang, et al. Simple to complex transfer learning for action recognition [J]. IEEE Trans on Image Processing, 2015, 25(2): 949.

[25] Chen Xu, Hero A, Savarese S. Shrinkage optimized directed information using pictorial structures for action recognition [J]. Computer Science, 2014.

[26] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection [C]// Computer Vision and Pattern Recognition. 2016.

[27] Liu Wei, Anguelov D, Erhan D, et al. SSD: single shot multibox detector [J]. Computer Science. 2015: 21-37.

[28] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [J]. Computer Science. 2015: 448-456.

Note: Figure translations are in progress. See original paper for figures.