Signal Processing: Image Communication 81 (2020) 115697

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Feature refinement for image-based driver action recognition via multi-scale attention convolutional neural network☆

Yaocong Hu, Mingqi Lu, Xiaobo Lu *

School of Automation, Southeast University, Nanjing, Jiangsu 210096, China
Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China

ARTICLE INFO

Keywords:
Driver action
Attention mechanism
Maximum selection unit
Fine-grained

ABSTRACT

Driver distraction has currently been a global issue causing the dramatic increase of road accidents and casualties. However, recognizing distracted driving action remains a challenging task in the field of computer vision, since inter-class variations between different driver action categories are quite subtle. To overcome this difficulty, in this paper, a novel deep learning based approach is proposed to extract fine-grained feature representation for image-based driver action recognition. Specifically, we improve the existing convolutional neural network from two aspects: (1) we employ multi-scale convolutional block with different receptive fields of kernel sizes to generate hierarchical feature map and adopt maximum selection unit to adaptively combine multi-scale information; (2) we incorporate an attention mechanism to learn pixel saliency and channel saliency between convolutional features so that it can guide the network to intensify local detail information and suppress global background information. For experiment, we evaluate the designed architecture on multiple driver action datasets. The quantitative experiment result shows that the proposed multi-scale attention convolutional neural network (MSA-CNN) obtains the state of the art performance in image-based driver action recognition.

1. Introduction

Along with the high-speed development of the automobile industry, the vehicle retention increases dramatically in the world. As reported by the Chinese Transport Ministry, the number of licensed vehicles will reach 250 million in 2020 [1]. The popularization of automobile provides a comfortable way for travelers, but at the same time, it increases hidden traffic hazard. All over the world, more than 3.5 thousand people die from traffic accidents every day, among which, distracted driving is a major contributor, accounting for more than 80% traffic accidents [2]. In actual driving, distracted driving action (such as leaving the steering wheels, using the cell phone, talking and etc.) is of great danger and reduces the drivers' reaction speed. Therefore, the researches of Advanced Driver Assistance Systems (ADAS) [3–5] with automatic function of driver action recognition have become a very promising topic for traffic security and intelligent transport.

Action recognition aims at dividing still image [6–9] or video clips [10–13] into specific action categories. In recent years, most of the researches about human action recognition are based on video analysis, since video clips include rich motion clues. Moreover, the combination of intra-frame appearance clues and inter-frame motion clues may result in a higher accuracy for action recognition. However, actions can also be recognized based on human's pose and detailed appearance without motion clues. Image-based human action recognition has a practical application requirement and is conductive to video-based recognition, which has attracted general attention in the latest researches.

In earlier researches, image-based action recognition can be carried out in three steps: feature extraction, action representation and action classification. Specifically, Zheng et al. [14] employed Poselet Activation Vector (PAV) to extract pose information of each action, then utilized sparse coding to learn context representation, and lastly classified human actions with Support Vector Machine (SVM). More recently, however, convolutional neural network (CNN) has broken the traditional recognition framework and achieved astonishing progress in related computer vision tasks, such as image classification [15–18], object detection [19–21] and super resolution [22–24]. Qi et al. [9] designed an end-to-end framework lately, in which they employed multi-task CNN to capture pose information, learn feature representation and classify actions in still images. Their implementation outperforms

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2019.115697.

* Corresponding author at: School of Automation, Southeast University, Nanjing, Jiangsu 210096, China.
E-mail addresses: yaoconghu@foxmail.com, 3012221702@qq.com (Y. Hu), xblu2013@126.edu.cn (X. Lu).

https://doi.org/10.1016/j.image.2019.115697
Received 30 January 2019; Received in revised form 2 October 2019; Accepted 8 November 2019
Available online 15 November 2019
0923-5965/© 2019 Published by Elsevier B.V.

traditional action recognition framework and achieves the mean average precision (mAp) of 82.2% in Standard-40 dataset (some example images are illustrated in Fig. 1.).

In this study, driver action recognition is close relevant to above mentioned human action recognition. Therefore, it is reasonable to transfer the state-of-the-art CNN architecture to our specific task. However, driver action recognition is a more challenging task and its difficulties can be summarized as follows:

• Driver action recognition can be regarded as a sub-classification task, and different actions share similar global patterns, but only exist differences in local details.

• Different drivers appear various driving habits (e.g. diverse styles in grasping the steering wheel), so same actions present large intra-class variations in posture.

Therefore, based on these issues, how to learn fine-grained feature representation has become essential to the driver action recognition task. Just around the near before, Hu et al. proposed a multi-stream CNN based method for driver action recognition, which is closely related and coincident with our study. In [25], they trained three-stream shallow CNN framework with receptive fields of different kernel sizes, and then combined multi-scale information at the last convolutional feature maps. Although, the designation of multi-stream CNN improves the network capacity in local representation, there are still some disadvantages which can be concluded into two aspects. Firstly, each stream is separately trained and just combined at the last convolutional layer, that is to say, it fails to fully exploit feature correlation at different scales. In addition, multi stream CNN treats all feature maps equally at pixel-wise and channel-wise, so the saliency of local parts is not explicitly emphasized and redundant global information cannot be diminished.

Aiming at addressing these limitations, in this paper, we design an advanced multi-scale convolutional neural network framework to improve the performance of image-based driver action recognition. In particular, we adopt a maximum selection unit to adaptively fuse multi-scale feature map in each convolutional stage; moreover, in the last convolutional feature map, we apply an attention mechanism to learn weight score at each pixel and each channel. The motivation is that maximum selection unit facilitates competition between multi-scale convolutional feature maps and the use of attention mechanism can guide the network to focus on salient characteristics. Hopefully, the proposed MSA-CNN can learn fine-grained feature representations and further improve the performance of driver action recognition in still images.

In terms of experiment dataset, we create a real driver action dataset (R-DA), which contains 42 816 images, covering 6 different driver actions of normal driving, leaving the wheel, talking on the phone, staring at the phone, driving with smoking, chatting with passengers. As we can see in Fig. 2, all images were filmed in an outdoor vehicle with variant light conditions. Although, this study is not target for recognition under weak illumination, night driver action recognition is a very promising topic and adding these images will bring benefit to future researches. The main creativity and contributions of this study involve four aspects:

• We create a real driver action dataset R-DA, containing 42 816 real driving images of 6 different driver actions for classification.

• Beyond the existing multi-stream CNN, we employ multi-scale convolutional block with maximum selection unit to adaptively learn multi-scale information from different receptive fields of convolutional kernels.

• We incorporate an attention mechanism to learn spatial saliency and channel saliency for feature refinement.

• We validate the effectiveness of our proposed MSA-CNN framework on multiple driver action recognition datasets and report the comparisons with the state of the art.

The rest of our study is organized as follow: we first introduce some recent related works about driver action recognition in Section 2; the proposed MSA-CNN framework and its detailed techniques are elaborated in Section 3; Experiment and quantitative evaluation are discussed in Section 4; lastly, in Section 5, we conclude this paper.

Table 1
Detail information of our self-created dataset and comparisons with other common-used driver action dataset.

Dataset	Scene type	Number of classes	Size
SEU-POSTURE [26]	Real driving scene	4 classes	93 473 images
StateFarm [27]	Real driving scene	10 classes	22 424 training images
S-DA [25]	Simulated driving scene	6 classes	33 162 image
R-DA	Real driving scene	6 classes	42 816 images

2. Related work

In this section, we first introduce corresponding driver action datasets and discuss the related solutions. Then, we give a brief description of the techniques and applications of multi-scale CNN and deep attention mechanism.

2.1. Existing driver action datasets

Many image datasets have been released for driver action recognition. Here, we enumerate detailed information about our self-created R-DA dataset and its comparisons with existing image-based driver action dataset in Table 1.

Zhao et al. in [26] created SEU-POSTURE dataset, containing 93 473 real driving images and covering 4 action categories, that is grasping the wheel, operating the lever, eating, talking on a cell phone. But, unfortunately, their self-created dataset is not open source. StateFarm [27] is a competition driver action dataset public on Kaggle homepage, which involves 22 424 training images and a huge number of unlabeled testing images. Hu et al. in [25] manually annotated 25 000 images of StateFarm for testing. Here, we illustrate all 10 classes of StateFarm dataset in Fig. 2.

In our previous work [25], we designed a simulated driver action dataset (S-DA), where all 33 162 images were filmed by a high resolution camera in a simulated indoor scenario, as shown in Fig. 2. In this study, we create a new dataset (R-DA), which follows the identical 6 categories with S-DA, including normal driving, leaving the wheel, talking on the phone, staring at the phone, driving with smoking, chatting with passengers, except that all 42 816 images were filmed in a real driving scene with variant illumination. It should be noted that the performance of the proposed MSA-CNN framework on these datasets are evaluated in Section 4.

2.2. Existing driver action recognition approaches

Driver action recognition has been extensively researched in recent years. Existing solutions can be roughly categorized into two classes: hand-crafted feature based solutions and deep learning based solutions.

Zhao et al. proposed a series of solutions for driver action recognition, in which they classified drivers' actions on their private SEU-POSTURE dataset. They in [26] utilized skin segmentation and contourlet transformation for feature representation, and then judged driving posture by random forest classifier; in [28] combined homomorphic filter and canny detection for feature extraction and adopted RBF-SVM for classification; in [29] extracted multi-wavelet features and then employed multi-layer perceptron for posture recognition; in [30] utilized Pyramid Histogram of Oriented Gradients (PHOG) to extract multi-scale representation for driver action recognition. Yan et al. [31] employed motion history image (MHI) to capture driver motion information in image sequences.

Some deep learning based driver action recognition solutions have been proposed driver action recognition recently. In [32], Yan et al. first adopted convolutional neural network for driver action recognition, in which they transferred AlexNet with end-to-end learning. Le et al. in [33] designed a multi-scale Faster-RCNN framework to

Fig. 1. Some example images of Stanford-40 dataset.

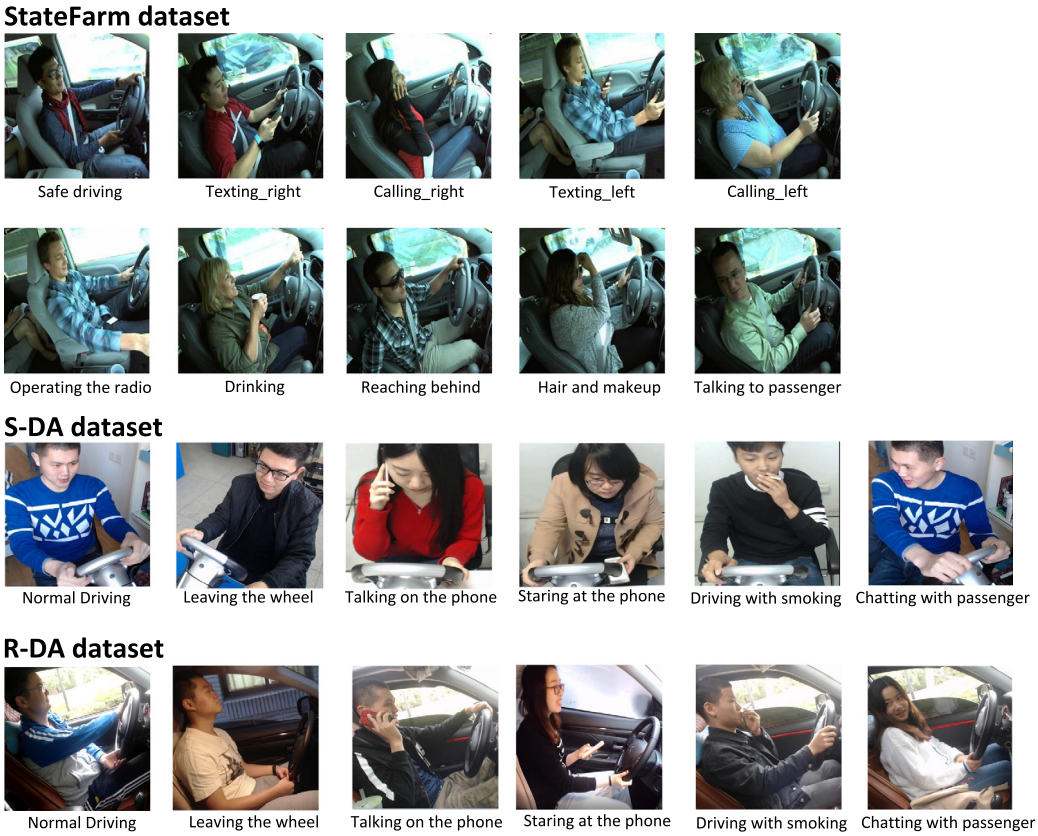


Fig. 2. Example images of existing driver action dataset, including StataFarm, S-DA and R-DA.

detect phoning and leaving the wheel. Koesdwiady et al. in [34] utilized very deep convolutional network (VGGNET) for driver distraction recognition. In the latest study, Hu et al. [25] trained a multi-stream CNN to extract multi-scale feature representation and achieved the state-of-the-art performance on StateFarm dataset and S-DA dataset.

In this work, we adopt multi-scale convolutional block with maximum selection unit and attention mechanism to improve the multi-stream CNN based solution [25]. Our sole aim is refinement of feature representation for more accurate driver action recognition.

2.3. Multi-scale CNN

Different size of convolutional kernels can capture multi-scale feature representations from images, so multi-scale CNN has extensive applications in computer vision. Sermanet et al. in [35] fed both low-level feature maps and high-level feature maps to a classifier for traffic sign recognition. Ciregan et al. [36] designed a multi-column deep neural networks to extract multi-scale feature representations for image classification. Liu et al. [37] employed triplet CNN to capture visual appearance of different scales for person re-identification. On the basis of these studies, in this work, we adopt maximum selection unit at the end of each convolutional stage to adaptively fuse multi-scale feature map.

2.4. Deep attention mechanism

Attention model is a hot topic in recent multimedia research. Specifically, Fu et al. [38] proposed a recurrent attention convolutional

Table 2
Layers and their parameters of the ResNet50.

Layer	Layer parameters
Input	Image size: 3 × 224 × 224.
conv_bc	Size: 64 × 7 × 7, stride: 2, activation: ReLU.
pool_bc	Size: 3 × 3, stride: 2, max pool.
block_1	Size: 64×[1×1, 3×3, 5×5, 7×7], stride: 1, BN, activation: Maxout, ReLU.
block_2	Size: 64×[1×1, 3×3, 5×5, 7×7], stride: 1, BN, activation: Maxout, ReLU.
block_3	Size: 128×[1×1, 3×3, 5×5, 7×7], stride: 2, BN, activation: Maxout, ReLU.
block_4	Size: 128×[1×1, 3×3, 5×5, 7×7], stride: 1, BN, activation: Maxout, ReLU.
block_5	Size: 256×[1×1, 3×3, 5×5, 7×7], stride: 2, BN, activation: Maxout, ReLU.
block_6	Size: 256×[1×1, 3×3, 5×5, 7×7], stride: 1, BN, activation: Maxout, ReLU.
block_7	Size: 512×[1×1, 3×3, 5×5, 7×7], stride: 2, BN, activation: Maxout, ReLU.
block_8	Size: 512×[1×1, 3×3, 5×5, 7×7], stride: 1,BN, activation: Maxout, ReLU.
p_attention	Size: (7 × 7)×(7 × 7), matrix multiply.
c_attention	Size: 512 × 512, matrix multiply.
fc_9	Neuron output: 1000, fully connected.
Output	Neuron output: 6, softmax.

neural network (RA-CNN) to learn discriminative region-based representation for fine-grained image classification. Chen et al. [39] adopted visual attention model to learn multi-scale weight at pixel-wise for semantic segmentation. Zhang et al. [40] in designed a multi-resolution attention convolutional neural network (MRA-CNN) to learn score map for dense crowd counting. Inspired by the above researches, in this study, we are motivated to employ deep attention mechanism to guide the network to focus on local salient regions.

3. Methodology

The overall architecture of our proposed multi-scale attention convolutional neural network (MSA-CNN) for image-based driver action recognition is depicted in Fig. 3. It consists of three main modules: multi-scale convolutional module, attention module and classification module, where multi-scale convolutional module filters images with different sizes of convolutional kernels to extract multi-scale features; attention module weights the importance of learned features at pixel-wise and channel-wise for feature refinement; classification module employs softmax classifier for final classification. Here, we list the relevant layer parameters of the proposed MSA-CNN framework in Table 2.

3.1. Multi-scale convolutional module

The convolution module takes the 224 × 224 × 3 raw images as network input. The first layer is a basic convolutional operation, which filters the input with 64 convolutional kernels by the size of 7 × 7 × 3, and then, max pooling operation follows after the convolutional operation and reduces the convolutional output to a 56 × 56 × 64 feature map with down sampling. Formally, given an input image I and its corresponding class label l, the basic convolutional operation can be represented as follow:

$$x_{bc} = \sigma(I * W + b), \tag{1}$$

$$\mathbf{F}_{bc} = down(x_{bc}), \tag{2}$$

where $*$ represents the convolutional operation; $\theta_{bc} = \{W, b\}$ denotes the parameters of basic convolutional (bc) layer, including weight and bias; $\sigma(\cdot)$ represents the activation function of Rectified linear unit (ReLU); $down(\cdot)$ is the max pooling operation; the learned feature map can be denoted as $\mathbf{F}_{bc}$.

Subsequently, the rest convolution networks are composed of a series of multi-scale convolutional blocks. In Fig. 4, we depict the structure of multi-scale convolutional blocks and its comparisons with the structure of multi-stream CNN [25] and Inception network [16]. In [25], multi-stream CNN contains three streams of convolutional networks. Each stream enjoys filters with different convolutional kernel sizes ($3\times3, 5\times5, 7\times7$) and is combined at the last convolutional layer, as shown in Fig. 4(a). The alternative multi-scale convolutional block employs parallel convolutional kernels with different sizes of 1 × 1, 3 × 3, 5 × 5, 7 × 7, and multi-scale information is fused at the end of each block. The idea of our proposed multi-scale convolutional block is derived from the structure of Inception [16], however, some significant improvements have been made, as shown in Fig. 4(b) and (c). In multi-scale convolutional block, firstly, batch normalization follows after each convolutional operation to solve the problem of internal covariance shift; secondly, maximum selection unit is adopted to fuse multi-scale information, since it can facilitate competition between multi-scale feature map and reduce information redundancy; in addition, shortcut mapping is introduced in each multi-scale convolutional block to alleviate the gradient exploding/vanishing.

Formally, the lth convolutional block convolves the output of the previous block with different scales of convolutional kernels, which can be denoted by the following equation:

$$x^{(l)} = \mathbf{F}^{(l-1)} * W^{(l)} + b^{(l)}, l = \{1, 2, \dots, 8\}, \tag{3}$$

where $\theta^{l}_{mc} = \{W^{(l)}, b^{(l)}\}$ is the parameter of the lth multi-scale convolutional (mc) block; $\mathbf{F}^{(l-1)}$ denotes the feature map of the previous block; $x^{(l)}$ represents the multi-scale feature map of the convolutional operation in the lth block; in particular, the first block takes the basic convolutional feature map as input.

Batch normalization is added after each convolutional operation to improve generalization ability. For a given mini-batch data, the convolutional feature map in the lth block can be denoted as $\{x^{(l)}_1, x^{(l)}_2, \dots, x^{(l)}_K\}$, we calculate the expectations and variances as follow:

$$E(x) = \frac{1}{K}\sum_{i=1}^{K} x^{(l)}_k, \tag{4}$$

$$Var(x) = \frac{1}{K}\sum_{i=1}^{K}(x^{(l)}_k - E(x))^2, \tag{5}$$

where K is the size of a mini-batch; $x^{(l)}_k$ is the feature of the lth block in the kth batch sample. $E(x)$ and $Var(x)$ represent the expectations and variances of the mini-batch. After that the normalized feature is formulated as below:

$$\hat{x}^{(l)}_k = \alpha \cdot \frac{x^{(l)}_k - E(x)}{\sqrt{Var(x) + \epsilon}} + \beta, \tag{6}$$

where ϵ is a small positive constant to ensure the stable in numerical value; α and β denotes the parameters of scale and shift transformation; $\hat{x}^{(l)}_k$ is the normalized feature.

Maximum selection unit is adopted to fuse multi-scale convolutional feature map. The normalized feature value of the lth block can be defined as $x^{(l)}_{scale}(c, i, j)$, in which (c, i, j) refers to the channel and coordinate of the normalized feature, and $scale$ denotes its related convolutional kernel sizes (1 × 1, 3 × 3, 5 × 5, 7 × 7). The output of the maximum selection unit can be expressed by:

$$y^{(l)}(c, i, j) = max(\hat{x}^{(l)}_{1\times1}(c, i, j), \hat{x}^{(l)}_{3\times3}(c, i, j), \hat{x}^{(l)}_{5\times5}(c, i, j), \hat{x}^{(l)}_{7\times7}(c, i, j)), \tag{7}$$

where the unit output $y^{(l)}$ at the position (c, i, i) is the max element-wise value across each scale.

Residual learning is introduced in the proposed multi-scale convolutional block to improve the network capacity. The input of multi-scale

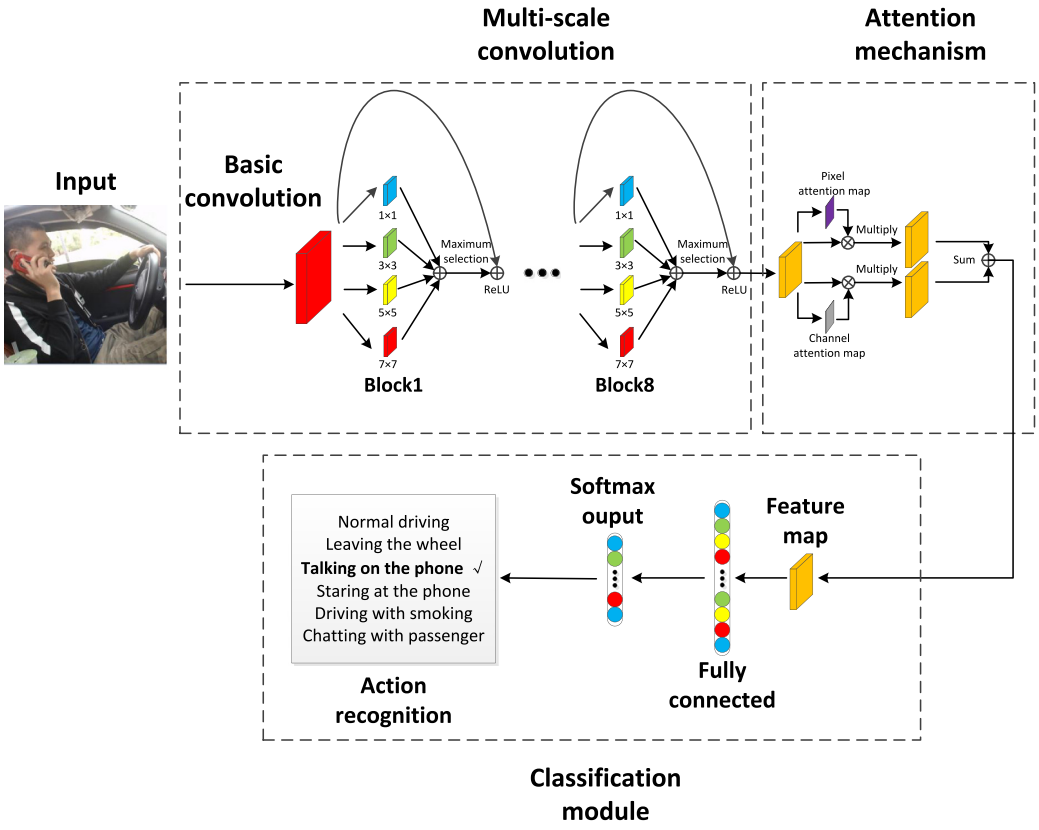
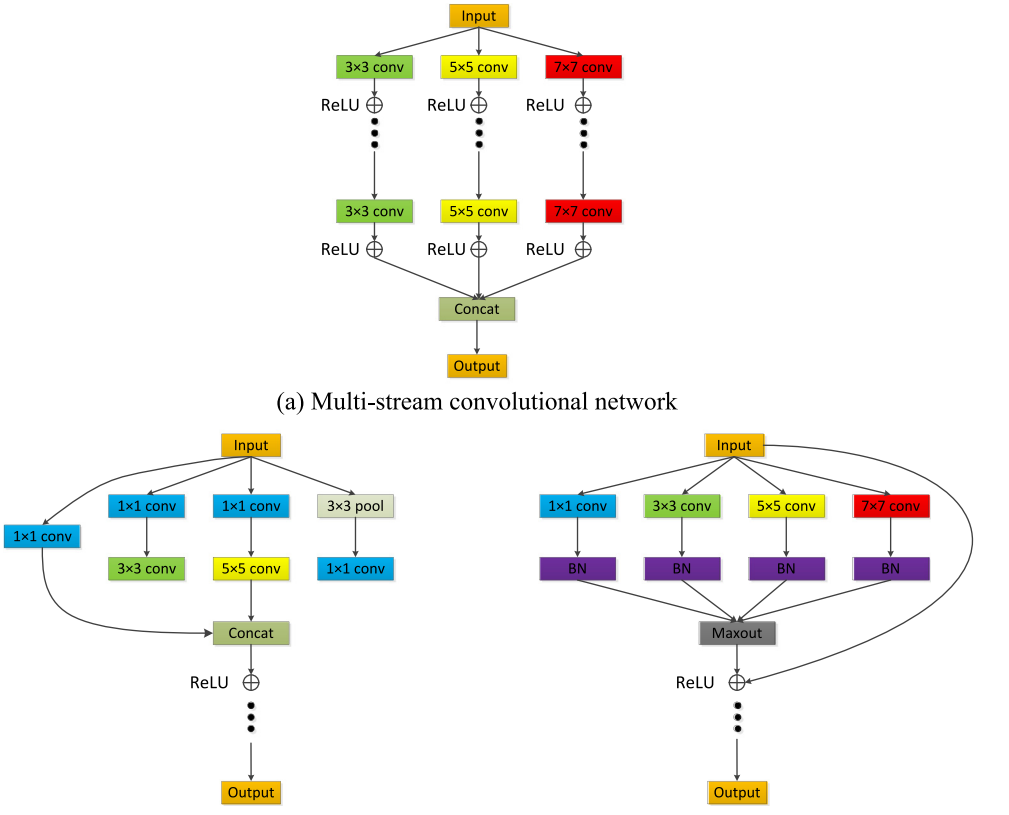


Fig. 3. The proposed MSA-CNN framework for image-based driver action recognition.



(a) Multi-stream convolutional network

(b) Inception convolutional network

(c) The proposed multi-scale convolutional network

Fig. 4. The comparisons of different convolutional structures, where (a) depicts the framework in multi-stream CNN [25], (b) shows the Incpection network [16], (c) is the proposed multi-scale convolutional blocks in our study.

block is connected to its output with shortcut identity and then the block output is activated by the ReLU function. The output feature map of the lth block can be denoted as below:

$$\mathbf{F}^{(l)} = \sigma(\mathbf{F}^{(l-1)} + y^{(l)}), \tag{8}$$

In Eq. (8), it should be noted that the dimension of $\mathbf{F}^{(l)}$ and $\mathbf{F}^{(l-1)}$ must be equal. While the feature map is down sampled in some blocks, the block output can be expressed by Eq. (9):

$$\mathbf{F}^{(l)} = \sigma(\mathbf{F}^{(l-1)} + W_s^{(l)} y^{(l)}), \tag{9}$$

where σ is the ReLU operation, and W_s denotes the projection matrix to be learned.

3.2. Attention module

After eight consecutive multi-scale convolutional blocks, the convolutional module outputs the multi-scale feature representations $\mathbf{F}^{(8)}$ with size of $7 \times 7 \times 512$, which contains rich semantic information. Here, we incorporate attention mechanism to guide the framework concentrate on salient characteristics for feature refinement. To be specific, the designed attention module is able to intensify local detail information and suppress the background information.

Here, in our study, we employ pixel attention and channel attention to learn the saliency of the feature map. Two connection modes is investigated for feature fusion between pixel-wise and channel-wise: series connection and parallel connection.

Pixel attention layer inputs the convolutional feature map and weights the importance of each pixel with a score map. Concretely, the score map can be expressed by the following equation:

$$a_p = tanh(W_{pa} U + b_{pa}) \tag{10}$$

$$S_p = \frac{exp(a_p)}{\sum_{a_{p'} \in P} exp(a_{p'})} \tag{11}$$

where $U \in \mathbb{R}^{49\times512}$ is the reshape of feature map, $\theta_{pa} = \{W_{pa}, b_{pa}\}$ denotes the weight matrix and bias item; $tanh(\cdot)$ is the hyperbolic tangent activation function; furthermore, $S_p \in \mathbb{R}^{49\times49}$ is the computed score map which denotes the saliency of each pixel.

Next the pixel score map is multiplied to the input matrix U, which can be denoted as Eq. (12).

$$\mathbf{F}_p = PA(\mathbf{F}|\theta_{pa}) = S_p \otimes U \tag{12}$$

where $\otimes$ denotes the operation of matrix multiplication and $PA(\cdot|)$ is a mapping from the input feature map to pixel attention feature map; the resulting feature map can be finally reshaped and represented as $\mathbf{F}_p \in \mathbb{R}^{7\times7\times512}$.

Similarly, channel attention mechanism is able to learn the contribution of each channel to its output. For a given feature map $\mathbf{F}$, channel attention layer can automatically learns score map and outputs the channel attention feature map. The corresponding calculation can be denoted as follow:

$$a_c = tanh(W_{ca} V + b_{ca}) \tag{13}$$

$$S_c = \frac{exp(a_c)}{\sum_{a_{c'} \in C} exp(a_{c'})} \tag{14}$$

$$\mathbf{F}_c = CA(\mathbf{F}|\theta_{ca}) = S_c \otimes V \tag{15}$$

where $V \in \mathbb{R}^{512\times49}$ is the reshape of feature map, $\theta_{ca} = \{W_{ca}, b_{ca}\}$ denotes the weight matrix and bias item; $S_c \in \mathbb{R}^{512\times512}$ is the channel score map which quantifies the contribution of each channel; furthermore, $CA(\cdot|)$ can be regarded as a mapping from input feature map to the channel attention feature map and the resulting feature can be finally represented as $\mathbf{F}_c \in \mathbb{R}^{7\times7\times512}$.

We consider two schemes to fuse pixel attention feature map and channel attention feature map, as we can see in Fig. 5. The attention module takes the feature map of last multi-scale convolutional block $\mathbf{F}^{(8)}$ as input. In series connection, the final attention feature map can be represented as:

$$\mathbf{F}_{att} = CA(PA(\mathbf{F}^{(8)})) \tag{16}$$

where the channel attention layer follows after pixel attention layer to weights the feature contribution for feature refinement.

While, in parallel connection scheme, the final attention feature map can be expressed by:

$$\mathbf{F}_{att} = PA(\mathbf{F}^{(8)}) + CA(\mathbf{F}^{(8)}) \tag{17}$$

where the pixel attention and channel attention is parallel in computing, and the output attention map is the sum of feature fusion.

It should be noted that the performance comparisons of two different connection mode in our attention mechanism are reported in next Section 4.

3.3. Classification module

The classification module is composed of a fully connected layer and a softmax layer, which inputs the attention feature map $\mathbf{F}_{att}$ and outputs the probability of different driver actions.

Formally, the fully connected layer reduces the $7 \times 7 \times 512$ attention feature map to 1000-d feature, which can be expressed by the following equation:

$$\mathbf{f} = W_{fc} \mathbf{F}_{att} + b_{fc} \tag{18}$$

where $\theta_{fc} = \{W_{fc}, b_{fc}\}$ denotes the parameter of the fully connected layer and $\mathbf{f}$ represents the learned 1000-d feature vector.

In the softmax layer, the softmax classifier outputs the scores of each driver action categories, which can be expressed by the following equation:

$$score = P(j) = \frac{exp(W_{cls}^{j} \cdot \mathbf{f} + b_j)}{\sum_{j'=1}^{n} exp(W_{cls}^{j'} \cdot \mathbf{f} + b_{j'})} \tag{19}$$

where $P(j)$ denotes the posteriori probability that $\mathbf{f}$ belongs to jth driver action class and $\theta_{cls} = \{W_{cls}, b_{cls}\}$ is the parameter of softmax classifier. $score = \{s_1, s_2, \ldots, s_n\}$ is the output of softmax layer, which predicts the probability distribution of different driver action classes.

3.4. Loss function

Here, we choose the cross entropy function to describe the distance of probability distribution between groundtruth and prediction, which can be expressed by the following equation:

$$\mathcal{L}_{cls} = -\sum_{j=1}^{n} l_j log(P(j)) \tag{20}$$

where l denotes the groundtruth label, $P(j')$ is the output of the softmax layer which represents the probability of the jth class.

For a mini-batch of data, the parameters of whole network can be optimized by adopting softmax loss as supervision, which can be denoted as:

$$\theta_{bc}^*, \theta_{mc}^*, \theta_{pa}^*, \theta_{ca}^*, \theta_{fc}^*, \theta_{cls}^* = \underset{\theta_{bc},\theta_{mc},\theta_{pa},\theta_{ca},\theta_{fc},\theta_{cls}}{\operatorname{argmin}} \left(\sum_{i=1}^{N} \mathcal{L}_{cls} + \|\theta_{bc}\|_2^2 + \|\theta_{mc}\|_2^2 + \|\theta_{pa}\|_2^2 + \|\theta_{ca}\|_2^2 + \|\theta_{fc}\|_2^2 + \|\theta_{cls}\|_2^2 \right), \tag{21}$$

where we add the regularization term $\|\theta\|_2^2$ in our loss function so as to reduce the potential overfitting.

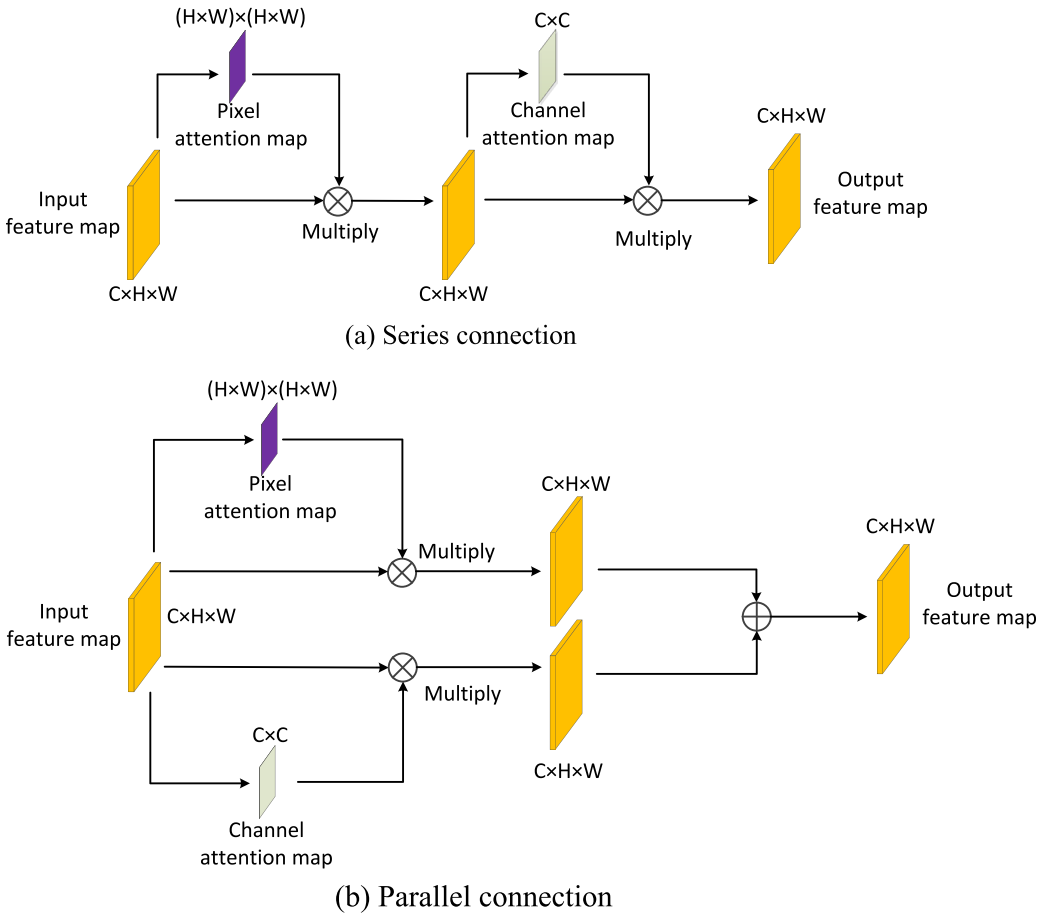


Fig. 5. The illustration of pixel-wise and channel-wise attention mechanisms, where (a) is the series structure and (b) is the parallel structure.

4. Experiment

We use the PyTorch toolbox to build the proposed multi-scale attention convolutional neural network (MSA-CNN) framework. In our implementation, the framework is working on a Intel core I7 serve with NVIDIA GTX-TITAN-X GPU, Ubuntu 18.04 operating system. We adopt Stochastic Gradient Decent (SGD) to update the parameters with the mini-batch size of 80, initial learning rate of 0.001. The training procedure maintains before the softmax loss remains nearly unchanged.

4.1. Experiment setting

We validate the proposed MSA-CNN framework on multiple driver action datasets, including S-DA, R-DA and StateFarm. Here we discuss the data preparation for our comparative experiment.

S-DA dataset: In [25], Hu et al. created a simulated driver action dataset (S-DA). All images in S-DA were filmed in a fixed indoor background, with 6 different driver action categories, marked as C0–C5. These images can be divided into 13 230 training images and 19 932 testing images. The number of images of different classes in S-DA can be shown in Table 3.

C0: Normal driving,
C1: Leaving the wheel,
C2: Talking on the phone,
C3: Staring at the phone,
C4: Driving with smoking,
C5: Chatting with passengers.

Table 3
Number of images on S-DA dataset.

Category	Training set	Testing set
Normal driving	1 764	2 658
Leaving the wheel	1 735	2 612
Talking on the phone	2 823	4 253
Staring at the phone	2 731	4 115
Driving with smoking	2 658	4 003
Chatting with passenger	1 519	2 291
Total	13 230	19 932

StateFarm dataset: StateFarm [27] is a public driver action dataset, containing 22 424 training images and 25 000 testing dataset with manually annotation. All these images were captured in a real driving scene, with 10 different classes, marked as C0–C9. We list the number of images of different classes in Table 4.

R-DA dataset: We create a new driver action dataset in this work. R-DA follows the same six driver action classes with the S-DA dataset, except that all images were filmed in an outdoor vehicle. Here, we classify these images into training set and testing set, and list the number of images of different classes in Table 5.

Data augmentation: We follow the reference of [25] to add augmented data for network training. Two schemes have been employed for data augmentation. One is random crop and another is content-based processing, such as noising, flipping, channel enhancement and etc. We consider that data augmentation increases the robustness of the network and alleviate the latent overfitting.

Table 4
Number of images on StateFarm dataset.

Category	Training set	Testing set
Normal driving	2 489	2 773
Texting_right	2 267	2 528
Calling_right	2 317	2 585
Texting_left	2 346	2 616
Calling_left	2 326	2 594
Operating the radio	2 312	2 577
Drinking	2 325	2 592
Reaching behind	2 002	2 232
Hair and makeup	1 911	2 131
Taking with passengers	2 129	2 372
Total	22 424	25 000

Table 5
Number of images on R-DA dataset.

Category	Training set	Testing set
Normal driving	2 279	3 438
Leaving the wheel	2 246	3 379
Talking on the phone	3 638	5 471
Staring at the phone	3 521	5 305
Driving with smoking	3 429	5 162
Chatting with passenger	1 974	2 974
Total	17 087	25 729

Table 6
Total accuracy rate and accuracy rate of each category of our proposed solution and its comparisons with corresponding methods on R-DA dataset.

Solution	C0	C1	C2	C3	C4	C5	Total
PHOG-MLP [30]	73.3%	70.2%	71.9%	70.8%	67.6%	74.1%	71.0%
PAV-SVM [14]	76.8%	78.2%	75.4%	77.9%	70.3%	73.7%	75.3%
AlexNet [32]	80.5%	82.1%	83.0%	79.5%	78.4%	82.8%	80.9%
VGG19 [34]	86.7%	85.4%	82.9%	85.1%	82.4%	87.0%	84.6%
PAV-Hint CNN [9]	87.9%	88.8%	85.4%	85.9%	84.3%	88.3%	86.4%
Multi-stream CNN [25]	87.3%	88.4%	89.2%	86.3%	87.1%	85.2%	87.4%
MSA-CNN	95.2%	93.6%	94.1%	96.0%	90.9%	94.5%	**94.0%**

4.2. Experiment result on R-DA

We assess the driver action recognition performance of the proposed MSA-CNN framework with both hand-crafted feature based solutions and deep learning based solutions on R-DA dataset. Existing solutions are evaluated for contrast: The solution of Pyramid Histogram Oriented Gradients and Multi-layer Perceptron (PHOG-MLP) [30], the solution of Poselet Activation Vector and Support Vector Machine (PAV-SVM) [14], the solution of CNN framework (AlexNet) [32], the solution of very deep CNN framework (VGGNet) [34], the solution of PAV-Hint CNN [9] and the latest solution of multi-stream CNN [25]. Here, we report the recognition performance of different driver action classes with different solutions in Table 6.

4.2.1. The performance of traditional solution

In [30], Zhao et al. employed Pyramid Histogram of Oriented Gradients (PHOG) to encode the spatial property of local features and classify different driver actions by utilizing multi-layer perceptron. Here, we repeat their PHPG-MLP implementation and achieve the unsatisfactory recognition performance with the total accuracy rate of 71.0% on R-DA dataset. Zheng et al. in [14] incorporated both pose and context information by the combination of Poselet activation vector and sparse coding, and then different actions are recognized by employing support vector machine. We transferred their approach on our specific task and test it on R-DA. The solution of PAV-SVM achieves the total accuracy rate of 75.3%.

Table 7
Total accuracy rate and accuracy rate of each category of our proposed method and its comparisons with corresponding methods on S-DA dataset.

Solution	C0	C1	C2	C3	C4	C5	Total
PHOG-MLP [30]	78.7%	77.2%	75.8%	76.9%	64.9%	78.0%	74.6%
PAV-SVM [14]	79.9%	81.0%	78.3%	77.5%	73.6%	79.4%	77.9%
AlexNet [32]	85.7%	84.2%	85.5%	82.3%	78.9%	85.6%	83.4%
VGG19 [34]	89.4%	89.1%	91.9%	90.3%	87.2%	90.8%	89.8%
PAV-Hint CNN [9]	90.1%	91.7%	90.8%	88.6%	90.4%	91.1%	90.3%
Multi-stream CNN [25]	93.8%	93.4%	94.9%	93.5%	89.8%	94.5%	93.2%
MSA-CNN	98.4%	96.6%	95.7%	98.1%	93.9%	97.8%	**96.7%**

The performance of driver action recognition is largely dependent on the capacity of feature representation. However, traditional features are mainly hand-crafted and cannot be applicable to characterize abstract driver actions. Obviously, these traditional solutions perform weaker than the deep learning based recognition solution, as we can see in Table 6.

4.2.2. The performance of deep learning based solution

In [32], Yan et al. first employed end-to-end deep learning framework to driver action recognition. In their implementation, they built a 8 layers convolutional neural network (AlexNet) to learn feature representation and classify driver actions. Here, we test their network on R-DA and obtains the total accuracy of 80.9%. Deeper neural network contains richer semantic information, so Koesdwiady et al. [34] adopted transfer learning scheme to train a 19 layers convolutional neural network. The VGGNet improves the recognition performance and achieves the total accuracy rate of 84.6%. In [9], Qi et al. designed a PAV-Hint CNN framework for image-based action recognition, in which they employed multi-task learning strategy to jointly learn action class and predict poselet activation vector. We repeat their solution and validate the effectiveness of pose hint learning (86.4% on R-DA). Up to date, the latest solution in the domain of driver action recognition is proposed by Hu et al. [25]. They designed a multi-stream CNN framework of receptive field of different kernel size and then each stream was combined at the last convolutional layers for multi-scale fusion. Their solution surpasses the standard CNN framework and achieves the total accuracy of 87.4%.

Although existing deep learning solution outperforms the hand-crafted based solution and significantly improves recognition accuracy, there are still some limitations: they do not fully exploit the relevance of features at different scales and they fail to learn the saliency of the feature.

4.2.3. The performance of the proposed MSA-CNN framework

In this work, we build a multi-scale attention neural network for driver action recognition. In particular, we design a multi-scale convolutional block with maximum selection unit to adaptively learn multi-scale feature representation; moreover, attention mechanism is adopted to emphasize the local detail and suppress the global background of the input image for feature refinement. The proposed MSA-CNN framework achieves the state-of the art performance with the total accuracy rate of 94.0%. Here, we show the confusion matrix of MSA-CNN solution in Fig. 6. Furthermore, we respectively assess the impact of multi-scale convolutional module and attention module in next Sections 4.4 and 4.5.

4.3. Extensive experiment result on S-DA and StateFarm

In [25], Hu et al. tested their solution on S-DA dataset and StateFarm dataset. Here, in order to illustrate the generalization ability of the proposed MSA-CNN based solution, we perform extensive experiments and report the result and its comparison with corresponding solutions. As shown in Tables 7 and 8, the proposed MSA-CNN also obtains the state of the art performance on S-DA and StateFarm with

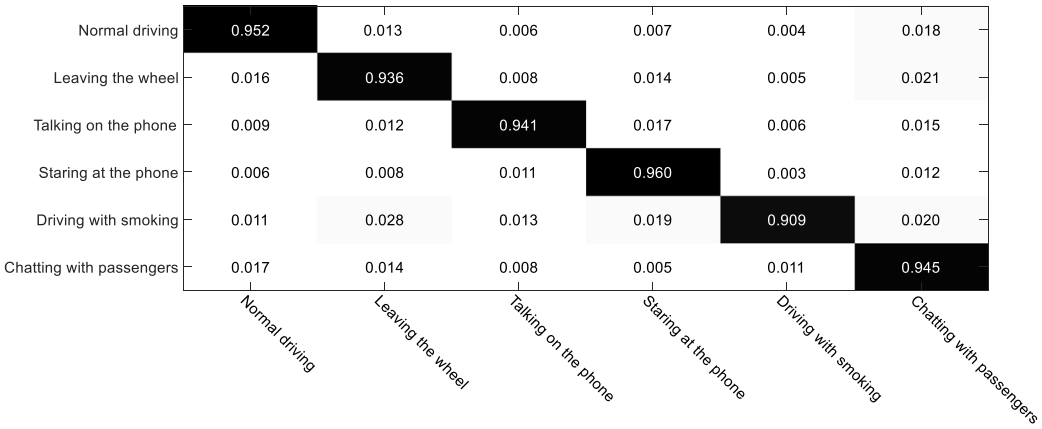


Fig. 6. Confusion matrix of MSA-CNN based solution on R-DA dataset.

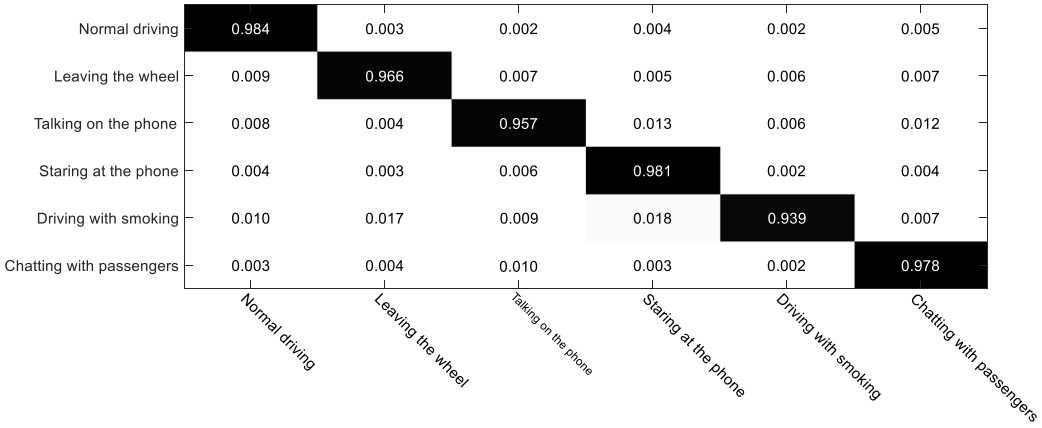


Fig. 7. Confusion matrix of MSA-CNN based solution on S-DA dataset.

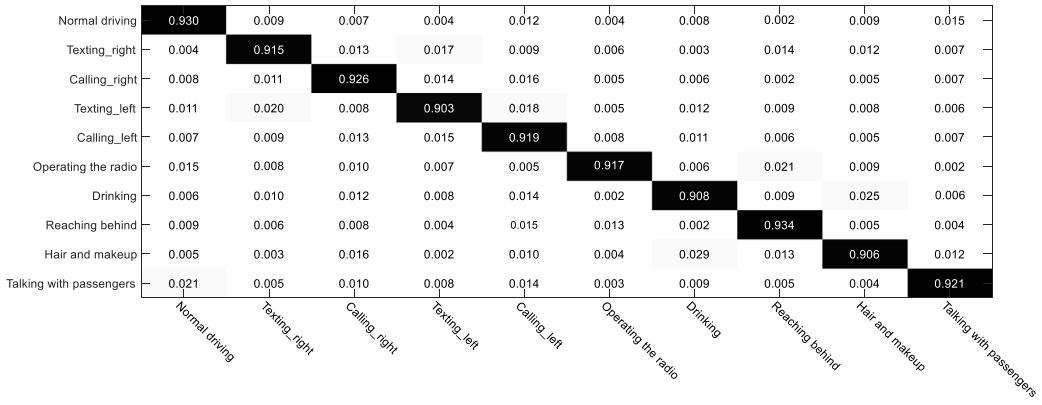


Fig. 8. Confusion matrix of MSA-CNN based solution on StateFarm dataset.

Table 8
Total accuracy rate and accuracy rate of each category of our proposed method and its comparisons with corresponding methods on StateFarm dataset.

Solution	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	Total
PHOG-MLP [30]	63.7%	56.0%	57.4%	59.8%	54.4%	66.1%	68.5%	62.0%	66.9%	64.5%	61.8%
PAV-SVM [14]	69.5%	65.4%	61.6%	62.3%	64.8%	71.3%	64.9%	65.8%	68.1%	66.2%	66.0%
AlexNet [32]	78.2%	66.1%	69.0%	71.3%	65.2%	73.9%	74.8%	77.5%	76.6%	68.5%	72.0%
VGG19 [34]	84.2%	78.8%	83.4%	81.5%	79.8%	80.4%	78.2%	80.0%	79.3%	77.9%	80.4%
PAV-Hint CNN [9]	85.0%	80.2%	81.8%	82.9%	81.5%	80.8%	81.4%	82.3%	80.6%	79.9%	81.7%
Multi-stream CNN [25]	88.1%	84.6%	86.5%	87.8%	86.9%	87.2%	88.9%	85.4%	84.2%	85.1%	86.6%
MSA-CNN	93.0%	91.5%	92.6%	90.3%	91.9%	91.7%	90.8%	93.4%	90.6%	92.1%	**91.8%**

Table 9
The comparisons of total accuracy rate of different convolutional structure in different driver action dataset.

Structure	S-DA	R-DA	StateFarm
Inception [16]	90.1%	85.3%	82.3%
Multi-stream network [25]	93.2%	87.4%	86.6%
Muli-scale block (maxout)	94.8%	91.1%	89.6%
Muli-scale block (concat)	93.9%	90.4%	88.5%

Table 10
The comparisons of total accuracy rate of different attention scheme in different driver action dataset.

Attention scheme	S-DA	R-DA	StateFarm
No attention	94.8%	91.1%	89.6%
Pixel attention	95.7%	91.9%	90.8%
Channel attention	95.3%	92.2%	90.6%
Series connection	96.2%	93.1%	91.5%
Parallel connection	96.7%	94.0%	91.8%

the total accuracy rate of 96.7% and 91.8%. Furthermore, Figs. 7 and 8 show the confusion matrix of the MSA-CNN solution in S-DA dataset and StateFarm dataset.

4.4. Evaluation of performance with multi-scale convolutional block

We consider that the designation of multi-scale convolutional block can capture rich semantic information for feature representation. Moreover, maximum selection unit promotes multi-scale competition and reduces feature redundancy. In order to validate this assumption, we remove the attention module and then compare the performance with multi-stream CNN [25] and Inception network [16]. The quantitative experiment result can be seen in Table 9. Leaving out the attention module, the designed framework still achieves the state of the art performance on existing driver action dataset. For contrast, we adopt feature map concatenation scheme to replace maximum selection unit in our proposed multi-scale convolutional block and report the experiment result on corresponding dataset. As shown in Table 9, the maximum select unit slightly outperforms the concatenation scheme in multi-scale information fusion.

4.5. Evaluation of performance with attention mechanism

Attention mechanism is another important contribution to more accurate driver action recognition. In this study, we adopt pixel-wise and channel-wise attention to weight saliency for feature refinement. Furthermore, we investigate two attention mode to fuse the result of pixel attention and channel attention. Here, we add the attention mechanism after the multi-scale convolutional module and report the quantitative experiment result of separate pixel attention, separate channel attention, series connection and parallel connection. As we can see in Table 10, both pixel attention and channel attention facilitate more accurate driver action recognition. In terms of feature fusion, parallel connection is superior to series connection in corresponding datasets.

4.6. Evaluation of the generalization ability

In order to illustrate the generalization ability of the proposed framework, here, we do an extensive experiment. R-DA dataset and StateFarm dataset contain some similar driver action classes, so we are motivated to employ the R-DA dataset for network training and then test it on a subset of the StateFarm. The new StateFarm-Sub dataset contains 13 874 training images and 15 468 testing images, which retains four driver action classes from the original StateFarm dataset, involving normal driving, calling, texting and talking. Here, we report the recognition performance of the proposed framework on StateFarm-Sub testing dataset with different training data.

Table 11
The comparisons of different training data on StateFarm-Sub testing dataset.

Training data	StateFarm-Sub
Trained from R-DA	94.2%
Trained from StateFarm-Sub	95.7%
Pre-rained from R-DA and fine-tuned from StateFarm-Sub	96.5%

As we can see in Table 11, the performance gap between training on R-DA and training on StateFarm-Sub (94.2% vs. 95.7%) is not very significant, which means that the proposed MSA-CNN framework has a powerful generalization ability. Moreover, the recognition accuracy can reach 96.5%, while pre-training on R-DA dataset and fine-tuning on StateFarm-Sub dataset.

4.7. Visualization of the experiment result

In order to better interpret the trained MSA-CNN model for driver action recognition, here, we employ the Grad-CAM algorithm [41] to visualize the recognition result on multiple databases. As shown in Fig. 9, the proposed MSA-CNN framework can focus on discriminative regions, e.g. it highlights the driver's face for recognizing the action of talking to passenger. These visualization results demonstrate that feature refinement is effective for driver action recognition.

5. Conclusion

In this paper, we proposed a multi-scale attention convolutional neural network (MSA-CNN) to learn fine-grained feature representation for driver action recognition in still images. The designed MSA-CNN architecture is composed of three main modules: multi-scale convolutional module, attention module and classification module. Specifically, the multi-scale convolutional module filters input images with different sizes of convolutional kernels and adaptively fuse multi-scale information by adopting maximum selection unit as activation; attention module learns saliency of the learned feature map at both pixel-wise and channel-wise for feature refinement; classification module employs softmax classifier for final recognition. The effectiveness of proposed MSA-CNN architecture is validated on both self-created driver action dataset and existing dataset. Extensive experiment results shows that the proposed method obtains significant performance improvements compared to the state-of-the-art. For further researches, how to improve the robustness of the algorithm in variant lighting and perspective condition may be a potential research interest.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China (No. 61871123), Key Research and Development Program in Jiangsu Province (No. BE2016739), the State Scholarship Fund from China Scholarship Council (No. 201906090126) and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

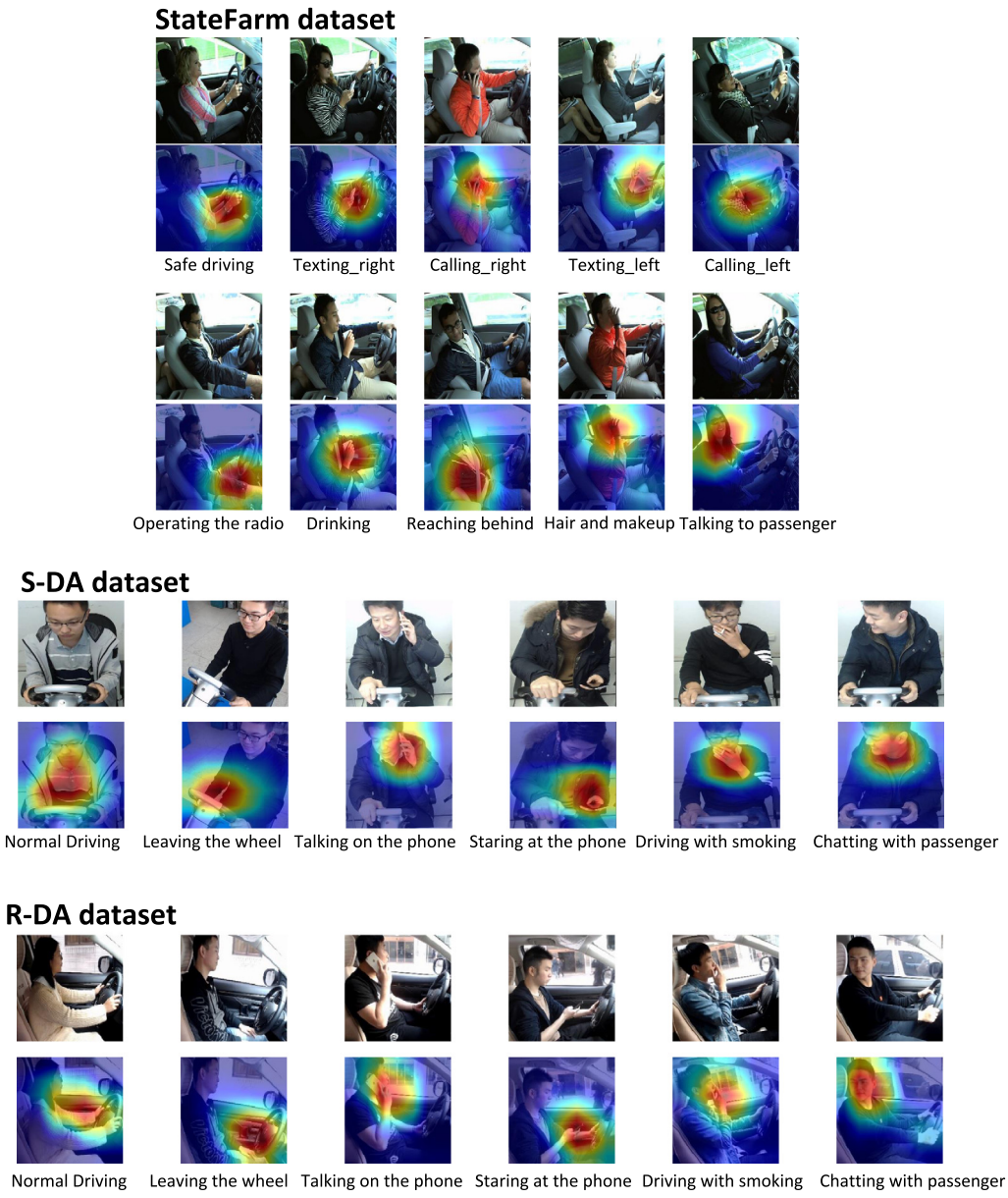


Fig. 9. Visualization of the experimental results.

References

[1] Y. Yanbin, Z. Lijuan, L. Mengjun, S. Ling, Early warning of traffic accident in shanghai based on large data set mining, in: 2016 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), 2016, pp. 18–21, http://dx.doi.org/10.1109/ICITBS.2016.149.

[2] M. Peden, Global collaboration on road traffic injury prevention, Int. J. Inj. Control Saf. Promot. 12 (2) (2005) 85–91.

[3] F. Jimnez, J.E. Naranjo, J.J. Anaya, F. Garca, A. Ponz, J.M. Armingol, Advanced driver assistance system for road environments to improve safety and efficiency, Transp. Res. Procedia 14 (2016) 2245–2254, Transport Research Arena TRA2016.

[4] P. Viswanath, K. Chitnis, P. Swami, M. Mody, S. Shivalingappa, S. Nagori, M. Mathew, K. Desappan, S. Jagannathan, D. Poddar, A. Jain, H. Garud, V. Appia, M. Mangla, S. Dabral, A diverse low cost high performance platform for advanced driver assistance system (adas) applications, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016, pp. 819–827, http://dx.doi.org/10.1109/CVPRW.2016.107.

[5] Y. Ouerhani, A. Alfalou, M. Desthieux, C. Brosseau, Advanced driver assistance system: Road sign identification using viapix system and a correlation technique, Opt. Lasers Eng. 89 (2017) 184–194, 3DIM-DS 2015: Optical Image Processing in the context of 3D Imaging, Metrology, and Data Security.

[6] G. Guo, A. Lai, A survey on still image based human action recognition, Pattern Recognit. 47 (10) (2014) 3343–3361.

[7] S. Yan, J.S. Smith, B. Zhang, Action recognition from still images based on deep vlad spatial pyramids, Signal Process., Image Commun. 54 (2017) 118–129.

[8] Y. Lavinia, H.H. Vo, A. Verma, Fusion based deep cnn for improved large-scale image action recognition, in: 2016 IEEE International Symposium on Multimedia (ISM), 2016, pp. 609–614, http://dx.doi.org/10.1109/ISM.2016.0131.

[9] T. Qi, Y. Xu, Y. Quan, Y. Wang, H. Ling, Image-based action recognition using hint-enhanced deep neural networks, Neurocomputing 267 (2017) 475–488.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, in: CVPR '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 1725–1732, http://dx.doi.org/10.1109/CVPR.2014.223.

[11] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, in: NIPS'14, vol. 1, MIT Press, Cambridge, MA, USA, 2014, pp. 568–576, URL http://dl.acm.org/citation.cfm?id=2968826.2968890.

[12] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1933–1941, http://dx.doi.org/10.1109/CVPR.2016.213.

[13] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, K. Saenko, Long-term recurrent convolutional networks for visual recognition and description, in: 2015 IEEE Conference on Computer Vision and

Pattern Recognition (CVPR), Vol. 00, 2015, pp. 2625–2634, http://dx.doi.org/10.1109/CVPR.2015.7298878, URL doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298878.

[14] Y. Zheng, Y. Zhang, X. Li, B. Liu, Action recognition in still images using a combination of human pose and context information, in: 2012 19th IEEE International Conference on Image Processing, 2012, pp. 785–788, http://dx.doi.org/10.1109/ICIP.2012.6466977.

[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems, in: NIPS'12, vol. 1, Curran Associates Inc., USA, 2012, pp. 1097–1105, URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, http://dx.doi.org/10.1109/CVPR.2015.7298594.

[17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, CoRR abs/1409.1556.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.

[19] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 21–37.

[21] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, http://dx.doi.org/10.1109/CVPR.2016.91.

[22] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.

[23] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 00, 2016, pp. 1646–1654, http://dx.doi.org/10.1109/CVPR.2016.182, URL doi.ieeecomputersociety.org/10.1109/CVPR.2016.182.

[24] B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, Enhanced deep residual networks for single image super-resolution, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vol. 00, 2017, pp. 1132–1140, http://dx.doi.org/10.1109/CVPRW.2017.151, URL doi.ieeecomputersociety.org/10.1109/CVPRW.2017.151.

[25] Y. Hu, M. Lu, X. Lu, Driving behaviour recognition from still images by using multi-stream fusion cnn, Mach. Vis. Appl. 30 (5) (2019) 851–865.

[26] C.H. Zhao, B.L. Zhang, J. He, J. Lian, Recognition of driving postures by contourlet transform and random forests, IET Intell. Transp. Syst. 6 (2) (2012) 161–168.

[27] StateFarm, Kaggle competetion, 2016, URL https://www.kaggle.com/c/state-farm-distracted-driver-detection.

[28] C. Zhao, B. Zhang, J. Lian, J. He, T. Lin, X. Zhang, Classification of driving postures by support vector machines, in: 2011 Sixth International Conference on Image and Graphics, 2011, pp. 926–930, http://dx.doi.org/10.1109/ICIG.2011.184.

[29] C. Zhao, Y. Gao, J. He, J. Lian, Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier, Eng. Appl. Artif. Intell. 25 (8) (2012) 1677–1686.

[30] C.H. Zhao, B.L. Zhang, X.Z. Zhang, S.Q. Zhao, H.X. Li, Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers, Neural Comput. Appl. 22 (1) (2013) 175–184.

[31] C. Yan, F. Coenen, B.L. Zhang, Driving posture recognition by joint application of motion history image and pyramid histogram of oriented gradients, in: Advances in Mechatronics, Automation and Applied Information Technologies, in: Advanced Materials Research, vol. 846, Trans Tech Publications, 2014, pp. 1102–1105, http://dx.doi.org/10.4028/www.scientific.net/AMR.846-847.1102.

[32] C. Yan, B. Zhang, F. Coenen, Driving posture recognition by convolutional neural networks, in: 2015 11th International Conference on Natural Computation (ICNC), 2015, pp. 680–685, http://dx.doi.org/10.1109/ICNC.2015.7378072.

[33] T.H.N. Le, Y. Zheng, C. Zhu, K. Luu, M. Savvides, Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016, pp. 46–53, http://dx.doi.org/10.1109/CVPRW.2016.13.

[34] A. Koesdwiady, S.M. Bedawi, C. Ou, F. Karray, End-to-end deep learning for driver distraction recognition, in: F. Karray, A. Campilho, F. Cheriet (Eds.), Image Analysis and Recognition, Springer International Publishing, Cham, 2017, pp. 11–18.

[35] P. Sermanet, Y. LeCun, Traffic sign recognition with multi-scale convolutional networks, in: The 2011 International Joint Conference on Neural Networks, 2011, pp. 2809–2813, http://dx.doi.org/10.1109/IJCNN.2011.6033589.

[36] D. Ciregan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3642–3649, http://dx.doi.org/10.1109/CVPR.2012.6248110.

[37] J. Liu, Z.-J. Zha, Q. Tian, D. Liu, T. Yao, Q. Ling, T. Mei, Multi-scale triplet cnn for person re-identification, in: Proceedings of the 24th ACM International Conference on Multimedia, in: MM '16, ACM, New York, NY, USA, 2016, pp. 192–196, http://dx.doi.org/10.1145/2964284.2967209, URL http://doi.acm.org/10.1145/2964284.2967209.

[38] J. Fu, H. Zheng, T. Mei, Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4476–4484, http://dx.doi.org/10.1109/CVPR.2017.476.

[39] L. Chen, Y. Yang, J. Wang, W. Xu, A.L. Yuille, Attention to scale: Scale-aware semantic image segmentation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3640–3649, http://dx.doi.org/10.1109/CVPR.2016.396.

[40] Y. Zhang, C. Zhou, F. Chang, A.C. Kot, Multi-resolution attention convolutional neural network for crowd counting, Neurocomputing 329 (2019) 144–152.

[41] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626, http://dx.doi.org/10.1109/ICCV.2017.74.