A Lightweight Action Recognition Method for Unmanned-Aerial-Vehicle video

Meng Ding, Ning Li[1], Ziang Song, Ruixing Zhang, Xiaxia Zhang
College of Electronic and Information Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing, P. R. China
{dm, lnee}@nuaa.edu.cn, {350767234, 374169703, 2693676740}@qq.com

Huiyu Zhou
School of Informatics
University of Leicester
Leicester, UK, LE1 7RH
h.zhou@ecit.qub.ac.uk

***Abstract*—In recent year, due to motility and wide coverage, unmanned aerial vehicle (UAV) has been widely applied in surveillance system. Human action recognition in UAV video is essential for surveillance video understanding. However, existing action recognition methods suffer from heavy computing, which makes it hard to deploy in real applications. In this paper, a lightweight action recognition method for UAV video(LARMUV) is proposed. This method is based on TSN and adopt MobileNetV3 as backbone, which greatly reduces amount of computing and parameters. Self-attention mechanism is adopted to capture temporal structure among different frames. For loss function, Focal Loss is used to putting more focus on hard, misclassified examples. Last but not least, knowledge distillation is employed to enhance the performance of our model, which transfer knowledge from a larger teacher model to student model. Experimental results on HMDB51, UCF101 and UAV dataset show that our method can achieve competitive performance compared to baseline methods while run in real-time mode.**

Keywords-action recognition; UAV; MobileNetV3; self-attention

I. INTRODUCTION

In recent year, unmanned aerial vehicle (UAV) has been widely applied in both military and civilian fields due to its agility, economy, and versatility. Compared with traditional fixed camera, camera on UAV shoots video from the air, which has broader horizon. In addition, UAV has great range of movement, which can cover greater range of surveillance. Given its advantages, UAV has been playing more and more important role in surveillance system. Human action recognition is a key technique for surveillance video understanding. Therefore, human action recognition in UAV video is of great importance.

Traditional action recognition methods [1], [2]can be divided into two stages. It first extracts pre-defined spatial-temporal features from videos, and then the extracted spatio-temporal features are classified using classifiers such as support vector machines (SVM). Dense trajectory [1] is one of the most classic action recognition methods. It first extracts dense trajectories by tracking dense points, and then extracts trajectory shape, Histogram of Oriented Gradients feature (HOG), Histogram of Optical Flow (HOF) as well as Motion Boundary Histogram (MBH) descriptors along the trajectory, and finally performs action classification using a SVM classifier.

[1]Ning Li, corresponding author, lnee@nuaa.edu.cn

Traditional methods rely on manually designed features, and the process of feature extraction is time-consuming. Deep learning has developed rapidly in the field of computer vision, and has also achieved great success in action recognition [3]–[8]. Deep learning methods do not need to extract features manually, and can achieve end-to-end learning. [4] proposed a architecture which incorporates spatial (RGB images) and temporal (optical flow) networks separately, which are then combined by late fusion. TSN [5] introduced a sparse temporal sampling strategy, which removes the redundant information between consecutive frames. The disadvantage of 2D convolution is that it cannot capture temporal structure among frames, so methods based on 2D CNN needs optical flow as temporal features. However, extracting optical flow is really time-consuming. [6] proposed 3D convolution for spatio-temporal feature learning, which can model appearance and motion simultaneously. As the expansion of convolutional kernel from 2D to 3D, the amount of network parameters and calculations increase significantly. I3D [7] extended the classic 2D convolutional network such as Inception to 3D ConvNet. Recently, Facebook proposed slowfast [8], which combines a fast frame rate pathway and a slow frame rate pathway, and achieved state-of-the-art accuracy on action recognition benchmarks.

Existing action recognition methods are computational expensive and memory intensive, so it is difficult to meet the time requirements in practical applications, especially for embedded devices with limited computing resource such as UAV. UAV usually take shots from high altitudes, with different angles, making it difficult to distinguish human action on the ground. In view of the limited memory and computing resource of UAV platform, as well as the complexity of video background information from the UAV's perspective, this paper proposes a lightweight action recognition framework based on the TSN architecture, which can achieve both high speed and high-quality performance.

The contributions of our work can be summarized as follows:

- To increase the operating speed, this paper chooses the lightweight network MobileNetV3 as the feature extraction network, which reduces the amount of parameters and calculations significantly;

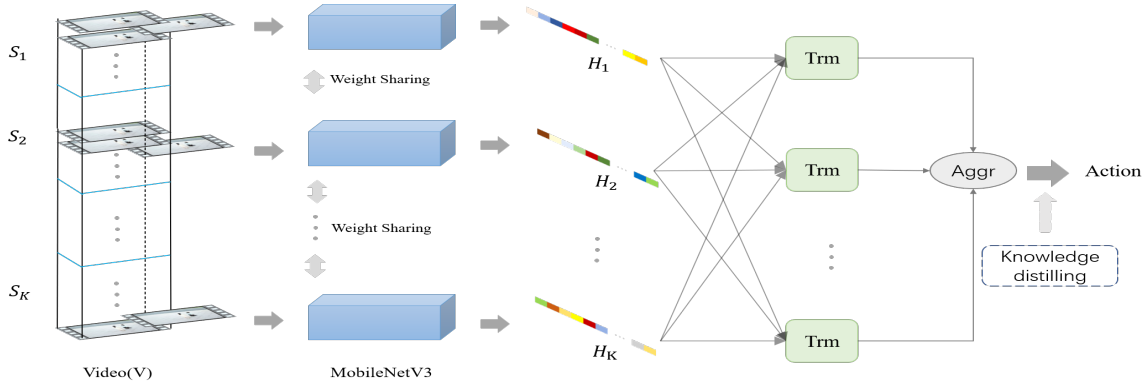

Fig. 1. Overview of A lightweight Action Recognition Framework for UAV Video (LARMUV)

- Vanilla TSN use aggregation function such as mean or max pooling, which failed to capture the temporal structure among different frames. This paper uses the Transformer, that is the self-attention mechanism, to model the dependence among frames.
- To pay more attention to misclassified, hard examples during training, Focal Loss is adopted to down-weights the loss of well-classified samples and focus on hard samples during optimizations.
- Finally, this paper introduces knowledge distillation to enhance the performance of our model. By transferring knowledge from a large teacher model to student model, the performance can be improved without increasing the amount of parameters and calculations.

II. Methodology

This paper introduces a lightweight action recognition method for UAV video(LARMUV), as is shown in Figure 1. LARMUV can be considered as a single stream architecture, based on the TSN architecture. Given a video V, we first divide it into K segments $\{S_1, S_2, ...S_K\}$, and then sample K frames $\{T_1, T_2, ...T_K\}$ from these segments. We feed the K frames into feature extraction network MobileNetV3 to get hidden vector $\{H_1, H_2, ...H_K\}$. Considering that the extraction of optical flow is time-consuming, our method discards the optical flow stream and only preserve RGB stream. Then Transformer, that is the self-attention mechanism, is used to model the temporal structure between different frames. Here we get the output vectors $O = \{O_1, O_2, ...O_K\}$ after Transformer layer. Finally we aggregate the output vectors and get the final classification probability p through the fully connected layer.

Considering that MobileNetV3 has few parameters, so its feature extraction ability is limited. Here knowledge distillation is adopted to further improve the performance of our model. The first step is to train a teacher network with a large amount of parameters. Here the I3D network pre-trained on the large-scale Kinetics dataset is selected as the teacher network. Then in the process of student network training, by matching the output of student network and teacher network, we can transfer the learned knowledge from teacher network to student network. Thereby, performance of student network is improved without increasing the amount of parameters and calculations.

A. MobileNetV3

As neural network becomes deeper and deeper, it achieves significant accuracy improvements in many visual recognition tasks. However, expensive computation and intensive memory requirements hinders their deployment in device with low memory and real-time applications [9]. Therefore, it is critical to reduce deep neural network's storage and computational cost. MobileNetV3 [10] is an efficient network for mobile and embedded device, which aims to reduce the amount of network parameters and calculations. For unmanned aerial vehicle, due to its limited computational power and memory resource, MobileNetV3 is adopted as the feature extraction network. MobileNetV3's network structure is optimized by the hardware-aware network architecture search (NAS) algorithm complemented by the NetAdapt algorithm. The main improvements of MobileNetV3 architecture design can be summarized as follows:

- Combining the depthwise separable convolutions of MobileNetV1 [11] and the inverted residual structure with Linear Bottlenecks of MobileNetV2 [12], greatly reducing the amount of parameters and calculations.
- Lightweight attention modules based on the squeeze and excitation (SE) is introduced, which adaptively recalibrates channel-wise feature responses.
- The swish activation function is replaced by h-swish function.

$$\text{swish}[x] = x \cdot \sigma(x), \tag{1}$$

in which $\sigma(\cdot)$ function is computationally expensive, so h-swish approximates $\sigma(\cdot)$ function with its piece-wise linear hard analog to reduce the amount of memory and calculations:

$$\text{h-swish}[x] = x \cdot \frac{ReLU6(x+3)}{6} \tag{2}$$

B. Self-attention

Through the MobileNetV3 network, we can obtain feature vectors $\{H_1, H_2, ...H_K\}$ of K frames. For vanilla TSN, a consensus function is used to aggregate K vectors, such as mean or max pooling. However, directly using mean or max pooling ignores the temporal structure of the video. [13] proposed to use Long short-term memory network (LSTM) to model the temporal structure of video. The main drawback is that LSTM cannot be computed in parallel, so as K increases, the running speed of model will inevitably slow down. Transformer [14] is adopted to model the temporal structure among different frames. Transformer is widely used in sequence modeling tasks, which has achieved great success in natural language processing. The Transformer layer is a combination of a self-attention layer and a point-wise feed-forward layer. First, attention mechanism can be formalized as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{(QK^T)}{\sqrt{d}}\right)V \tag{3}$$

When Q, K and V are the same matrix, it is called the self-attention mechanism. Instead of performing a single attention function, the multi-head attention linearly projects Q, K and V h times, and perform attention function in parallel. Then we concatentate results of h heads to get the final output, namely

$$\begin{gathered}\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right) W^o, \\ \text{where head}_i = \text{Attention}\left(QW_i^q, KW_i^k, VW_i^v\right).\end{gathered} \tag{4}$$

Compared to single-head attention, the multi-head attention introduces more parameters and enhances the expressive ability of the model.

For point-wise feed-forward layer, it consists of two linear layers with a $ReLU$ activation function:

$$\text{FFN}(x) = max(0, xW_1 + b_1)W_2 + b_2. \tag{5}$$

Here we take the feature vector of the video frames as input, namely $Q = K = V = \{H_1, H_2, ..., H_K\}$, $H \in R^{K\times d}$, where d is the dimension of the feature vector and K is the number of frames of the video, and get the output vector $O \in R^{K\times d}$ finally.

C. Focal Loss

Focal Loss [15] reshapes the standard cross entropy loss to down-weight the loss assigned to well-classified examples, and pay more attention to hard negative examples. Take binary classification problem as an example, we define p as the model's estimated probability for class with label $y = 1$, in which $p \in (0, 1)$. p_t is defined as

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \tag{6}$$

The binary cross entropy loss can be formulated as:

$$\text{CE}\left(p_t, y\right) = -\log\left(p_t\right) \tag{7}$$

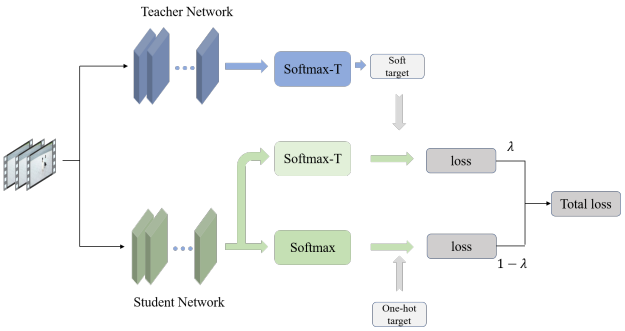

Fig. 2. Illustration of knowledge distillation.

Focal Loss adds a modulating factor $(1-p_t)^\gamma$ to cross entropy criterion, which is defined as:

$$\text{Focal\_loss}\ \left(p_t, y\right) = -\left(1 - p_t\right)^\gamma \log\left(p_t\right), \tag{8}$$

in which $\gamma = 2$ works well in our experiments. The modulating factor is used to reduce weights of well-classified examples. For example, when label $y = 1$ and $p = 0.9$, the modulating factor $(1 - p_t)^\gamma = 0.01$, which means the weight is 1% of cross entropy criterion.

D. Knowledge Distillation

Knowledge distillation [16] refers to the process of transferring the learned knowledge of the teacher network to the student network. Generally speaking, teacher networks refers to large cumbersome model, even ensemble of several models, while student network is smaller. Teacher model can learn better knowledge representation of videos, however, it can hardly deploy in real applications because their expensive computation cost. The output probability encodes the learned knowledge representation of teacher network: when model correctly predicts a class, it also assigns a smaller probability to a similar class. For example, from the output probability it can be inferred that the similarity between 'squat' and 'bend' is higher than 'squat' and 'walk'. While the true label is one-hot encoded, the correlations among classes is ignored. Therefore, to transfer knowledge from teacher network to student network, we train the student network by letting it match both the soft output of teacher network and the ground truth label.

The process of knowledge distillation is shown in Figure 2. Teacher network with larger parameters is trained at first. Here I3D model is selected as the teacher network, which is pre-trained on the Kinetics dataset, and then trained on the action recognition dataset. To further improve the performance of the teacher network, the teacher network has two streams: spatial and temporal stream, which are trained separately and then fused. Then student network is trained to fit both the soft output of teacher network and the ground truth label. Let p and q denote the output of the student network and teacher network respectively, y denotes the one-hot encoding of the true label, and the loss function is:

$$L = (1 - \lambda)\,\text{loss}(y, p) + \lambda\,\text{loss}(q, p). \tag{9}$$

TABLE I
EXPERIMENTAL RESULTS ON 3 DATASETS, USING ONLY RGB STREAM.

	Pre-training	HMDB51	UCF101	UAV
Vgg16	ImageNet	42.3	75.41	62.14
ResNet18	ImageNet	43.82	77.75	58.96
ResNet34	ImageNet	45.94	80.5	70.42
ResNet50	ImageNet	47.54	81.33	76.17
Inception v1	ImageNet	45.94	83.23	76.67
Inception v3	ImageNet	46.36	80.24	77.96
C3D	Scratch	7.10	40.26	36.74
C3D	Kinetics	41.60	80.16	78.86
I3D	ImageNet	43.06	74.25	74.58
I3D	Kinetics	66.03	90.83	92.13
LARMUV(Our method)	ImageNet	55.14	86.41	87.81

TABLE II
COMPARISON OF NETWORK PARAMETERS, COMPUTATIONAL COST AND RUNNING SPEED.

	Params(M)	FLOPs(M)	VPS	UAV Accuracy
Vgg16	134.5	15466.4	67	62.14
ResNet18	11.2	1816.1	212	58.96
ResNet34	21.4	3667.0	167	70.42
ResNet50	23.8	4098.5	118	76.17
Inception v1	11.3	2018.1	117	76.67
Inception v3	23.8	5743.1	79	77.96
C3D	78.4	8660.9	59	78.86
I3D	53.8	6445.2	8	92.13
LARMUV(Our Method)	10.6	226.8	157	87.81

Here λ is a hyper-parameter, which controls the weight of the loss corresponding to the soft target q and the real label y.

However, directly matching the output of teacher network q is not a good choice. As the probability of correct class is too high, and the probabilities of other classes are too small. As a result, it is difficult to distill from the teacher network. softmax-T is introduced here to "soften" the original output probability. The formula is as follows:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \tag{10}$$

in which z is the logit before softmax function. If $T = 1$, softmax-T is equivalent to standard softmax function. Using a higher value for T produces a softer probablity distribution over classes. Meanwhile, the entropy of q would increase, therefore provides more information to be distilled. We set $T = 2$ in our experiments.

III. EXPERIMENTS

A. Dataset

We evaluate the our method on both benchmarks, including HMDB51, UCF101 datasets and UAV dataset. HMDB51 dataset contains 5100 video clips divided into 51 categories; UCF101 dataset contains 13320 video clips divided into 101 categories. To further evaluate our method in video under UAV, we construct a human action recognition dataset under UAV, which is shooted in an altitude of 20 to 30 meters by DJI Mavic Pro. It contains 964 video clips divided into 11 categories: bend, handshake, clap, hug, kick, push, rid, run, squat, walk, walk with multiple persons. We use accuracy as evaluation metric in our experiments.

B. Implementation Details

We use MobileNetV3-Large as the backbone network, and use ImageNet pre-trained weights as initialization. Multiple data augmentation strategies are employed to alleviate overfitting, including:

- Random cropping: we choose the cropping position randomly from the middle, upper left, lower left, upper right, and lower right of the frame. The input frame size is fixed as 256 $\times$ 340 and the cropping width and height are randomly sampled from $\{256, 224, 192, 168\}$. Then we scale the image to a size of 224x224 as the input to the backbone network;
- Horizontal flipping: Flip the video frame horizontally with a probability of 0.5;
- Color jittering: Randomly change the brightness, contrast and saturation of the frame.

We set number of segments K as 3 in our experiments. Considering computation cost, we use 1 layer Transformer and set number of heads as 2. During training, We use Adam for optimization,and set learning rate as to 1e-3, weight decay as 0.0001. The experiments were carried out on the Nvidia 1080Ti GPU.

C. Baselines

We compare our method with a variety of action recognition methods, mainly in two categories: methods based on 2D and 3D convolution. The method based on 2D convolution is based on the vanilla TSN architecture with different feature extraction networks, including Vgg16, ResNet18, ResNet34, ResNet50, Inception v1, Inception v3. For methods based on 3D convolution, C3D and I3D with different pre-training strategy are included.

D. Results

Table I shows experimental results of our method and baselines on HMDB51, UCF101 and UAV datasets. Our method achieves 55.14%, 86.41%, 87.81% accuracy on HMDB51, UCF101 and UAV datasets respectively. Our method outperforms all the other baselines except for I3D model pre-trained on Kinetics dataset.

E. Comparison of Network Parameters, Computational Cost and Running Speed.

In this section, we compare the network parameters, computational cost and running speed of our method and baseline methods. The computational cost is measured by floating-point operations (FLOPs). The running speed is measured by videos per seconds (VPS) processed at inference time. The experiment was carried out on a single Nvidia 1080Ti GPU.

TABLE III
COMPARISON OF MODELS

	MobileNetV3	Self-attention	Focal Loss	Knowledge Distillation	UCF101	HMDB51	UAV
Model 1	√				48.2	81.97	78.85
Model 2	√	√			50.00	82.47	81.10
Model 3	√		√		51.99	82.56	81.77
Model 4	√			√	51.43	83.91	84.25
Model 5	√	√	√	√	55.14	86.41	87.81

As shown in Table II, the parameters and FLOPs of our method are significantly lower than other baseline models. As a result, our method achieves 157 videos per seconds at inference time while achieves competitive performance. Though the accuracy of our model is slightly lower than I3D, LARMUV is an order of magnitude faster than I3D. For embedded devices with limited computing resource such as UAV, our method has great advantage in practical applications.

F. Ablation Study

We set backbone network as MobileNetV3, and compare the effectiveness of several improvements of our method, including the self-attention mechanism, Focal Loss, and knowledge distillation. As shown in Table III, Model 2, Model 3 and Model 4 show better performance than Model 1, which verifies the effectiveness of the our proposed improvements. Model 5 combines these improvements and achieves the highest accuracy on all datasets.

IV. CONCLUSION

In this paper, a lightweight action recognition framework is proposed for video under UAV. To reduce computational cost and memory requirement, the lightweight network MobileNetv3 is adopted as the backbone network. To model the temporal structure among different frames, self-attention mechanism is employed here. Focal Loss is adopted to pay more attention to misclassified, hard examples during training. Finally, knowledge distillation from teacher network further improve the performance of our proposed LARMUV without additional cost. The experimental results on benchmarks and UAV dataset show that our method has achieved competitive performance while greatly reducing the amount of calculation and parameter. Through running speed analysis, our method can achieve real-time recognition, which make it advantageous in actual deployment. For future work, although optical flow is time-consuming, there is no doubt that it is effective for action recognition. Therefore, it is worth thinking about how to use the network to learn features related to optical flow.

ACKNOWLEDGMENT

This work received support from Science and Technology on Electro-optic Control Laboratory and Aviation Science Foundation Project (ASFC-20175152036) and Key Project on Artificial intelligence(1004-56XZA19008). The authors are also grateful for the support of their colleagues at the Key Laboratory of Radar Imaging and Microwave Photonics, Ministry of Education.

REFERENCES

[1] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR 2011*. IEEE, 2011, pp. 3169–3176.

[2] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[3] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 695–712.

[4] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[5] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[7] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[8] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6202–6211.

[9] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[10] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1314–1324.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[13] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

Position can be chosen from: Prof. / Assoc. Prof. / Asst. Prof. / Lect. / Dr. / Ph. D Candidate/ Postgraduate / Ms.				
Paper ID	Position, Full Name, Working unit & nation	Email address	Research Interests	Personal website (if any)
E3020	Ms., Meng Ding, Nanjing University of Aeronautics and Astronautics, China	dm@ nuaa.edu.cn	video processing, computer vision	
	Assoc. Prof., Ning Li, Nanjing University of Aeronautics and Astronautics, China	lnee@ nuaa.edu.cn	computer vision	http://faculty.nuaa.edu.cn /ln2/zh_CN/index.htm
	Ms., Ziang Song, Nanjing University of Aeronautics and Astronautics, China	350767234 @qq.com	object detection	
	Ms., Ruixing Zhang, Nanjing University of Aeronautics and Astronautics, China	374169703 @qq.com	video processing, computer vision	
	Ms., Xiaxia Zhang, Nanjing University of Aeronautics and Astronautics, China	2693676740 @qq.com	object detection	
	Prof., Huiyu Zhou, University of Leicester, UK	h.zhou@ ecit.qub.ac.uk	machine learning, computer vision	https://www2.le.ac.uk/departments /informatics/people/huiyu-zhou