Air Traffic Controller’s Behavior Recognition from Still Images by Transfer Learning

Zheng Xiang *, Deyang He 1, Weinan Deng 1

School of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China

Deyang He 1: 978227625@qq.com

Weinan Deng 1: 302920643@qq.com

* Corresponding Author: Zheng Xiang; xz3_83@163.com; 18281001387

Abstract: The fatigue problem of air traffic controllers is one of the important reasons that endanger the safety of civil aviation. In this paper, we introduce behavior recognition into the field of fatigue monitoring of civil air traffic controllers. We investigate the method of automatic recognition of controller's behaviors by using transfer learning and model fusion. The task of behavior recognition for controllers can be regarded as a multi-classification problem (Such as putting heads on desks for a nap, lay with face upward, nod to sleep, talking with others, directing planes normally, Hold up chin on left hand, writing notes on strips radio check, Resting hand on forehead ). Our study solve this problem from three aspects: (1) Set up a data set of controller behavior, including 18247 pictures of 10 actions; (2) Employ two-stream CNN(Resnet50,Inception V3) to extract different features; (3) We investigate different fusion strategy to improve the accuracy of action recognition of controllers. Experiments demonstrate that our method achieves a recognition accuracy of 94.2% and 93.7% on the CAFUC-Controller and Kaggle-Driving datasets.

1. Introduction
Air traffic controllers are one kind of aircrew. Their responsibility is to monitor the flight dynamics of each aircraft in front of the radar screen closely. They issue various instructions to the pilots through radio communication equipment. Due to the environmental pressure of air traffic control industry and its 24-hour shift system, it is likely to cause fatigue of controllers in work. The Controllers’ fatigue effect on Civil Aviation safety severely. Therefore, effective identification of fatigue status of controllers and scientific classification of fatigue levels has always been a hot issue in civil aviation safety research [1].

The method of image detection has friendly human-computer interaction which has little influence on the controller. It has enormous application prospects. At present, in the field of fatigue monitoring, image-based fatigue detection methods are mainly aimed at face recognition [2-3]. In this paper, we take the initiative of proposing the application of behavior recognition in the field of controller’s fatigue monitoring. We think that besides face, body movements also contain various fatigue information elements. If we can recognize the relevant behavior of controllers effectively, we can know the fatigue-related information. Due to the fixed working seats and closed working scenes of the controllers, the behavior of the controllers is relatively single. We consider 10 possible behaviors (including nodding to sleep, putting heads on desks for a nap, holding up one's chin, directing planes normally,
talking with others, resting hands on forehead, writing notes on strips and checking radio) that may occur when controllers working and fatiguing.

Therefore, we set up a dataset of controller’s behaviors, including 18247 pictures of 10 categories. In order to identify the controller’s behavior, we design a two-stream neural network structure composed of different models (ResNet50 and InceptionV3), then design and compare different fusion strategies.

2. Related work

Behavior recognition has a broad range of application areas. In recent years. After VGG model became the mainstream. Computational efficiency and low parameters of the model become the trend. Therefore, He K et al. [15] designed the residual networks(Res Nets), which resolved the problem of the lower accuracy of the deeper neural network. Szegedy C et al. [16] released designed Inception v3 networks. In [7], Li et al. proposed a driver abnormal behavior detection method, employing Fast-RCNN [17] method with VGG16 model, including using mobile phones and hands off the wheel. In [8], Gkioxari et al. employed RCNN network to recognize 10 actions in still images. Lavinia Yet al. [10] proposed a fusion method that concatenates two and three deep convolutional neural networks(CNN) for motion recognition. Qi et al. [11] Classified human behavior on VGG19 model by using the method of transfer learning, and achieved 82.2% effect on Stanford-40 dataset [12]. On the other hand, recent studies use different neural network stream to extract different features, and then combine these features to achieve better robustness [5-6-18]. This procedure is also called feature fusion. The types of data fusion can be early fusion (feature level) or late fusion (model level). Therefore, this paper designs a two-stream neural network structure composed of different models, and designs and compares different fusion strategies.

3. Methodology

In this section, we design and explain the two-stream neural network model. At first, we introduce the structure of each stream. Additionally, we introduce the details of two different fusion strategies. Consequently, we introduce the procedure of training neural network. The relevant mathematical symbols and the meanings they represent are listed in Table 1.

### Table 1 Related mathematical symbols and their representation.

| Symbols | Representation |
|---------|----------------|
| $V^i$   | The feature vector of the $i$th stream |
| $F_i$   | The $i$th stream's feature map |
| $F_{st}$| The early fusion model's stacked feature map |
| $S_i$   | The $i$th stream's classification score |
| $S_{late}$ | The late fusion model’s final score |
| $k$     | Behaviors of air traffic controllers(10 class) |
| $\delta_{soft}$ | Softmax layer parameters |

3.1. Network architecture

Owing to our dataset is small. We select the method of transfer-learning to construct the model of each stream and remove the fully-connected layer at the top of the InceptionV3 and ResNet50 networks and connected the global average pooling layer directly to 10 softmax layer.

Each stream convolutional neural network is trained and calculated the classification results independently. All illustrations of each stream architectures are shown in Figure 1.

The first stream convolutional neural network is InceptionV3. The input layer is RGB images with size of 360×480×3. The second stream convolutional neural network is ResNet50. The input layer is RGB images with size of 224×224×3.
In the first stream, through 311 hidden layers of InceptionV3, $8 \times 13 \times 2048$ feature maps are extracted. Then the global average pooling layer reduces the last feature maps to 2048-d. In the second stream, through 173 hidden layers of ResNet50, $7 \times 7 \times 2048$ feature maps are extracted. Then the global average pooling layer reduces the last feature maps to 2048-d.

### 3.2. Fusion strategy

We design two fusion models, early fusion model and late fusion model. The illustration of early fusion model is shown in Figure 2. In the early fusion model, the multi-scale feature maps are combined from each stream which is the last feature map. Then the global average pooling layer reduces the last feature maps to 2048-d. Its output is connected to 10 softmax classifiers.

The stacked feature map is illustrated as Eq.(1). Let $V^i$ stand for 2048 feature vectors from the global average pooling layer in the $i$th stream. Conditional probability $P(k | V^i)$ can be computed by ith softmax classifier.

$$F_{st}(x, y, 2d-1) = F_1(x, y, d)$$
$$F_{st}(x, y, 2d) = F_2(x, y, d)$$

(1)

Where $F_1 \in R^{8 \times 13 \times 2048}$, $F_2 \in R^{7 \times 7 \times 2048}$, $F_{st} \in R^{(8 \times 13 \times 2048, 7 \times 7 \times 2048)}$ and $F_{st}(x, y, d)$ represents the value of the pixel matrix which has two dimensions $x$ and $y$ in the $d$th feature map.

In the late fusion model, Convolutional neural network of each stream is trained separately and output a classification score. Traditionally, the final score of late fusion is computed by a weighted average of all streams’ score. However, if the classifiers are independent of each other. Tang, P et al. [13] demonstrate that Bayesian model may have better results. Therefore, the later fusion strategy is mainly based on the Bayesian model. Figure 3 shows all the illustration of late fusion strategy.
Where $P(k|V)$ is the conditional probability of $V$ belonging to the category $k$. According to the Bayesian principle, we can compute the joint probability $P(k|V^1, V^2)$ by the following formula.

$$S^k_{\text{late}} = \frac{P(k|V^1, V^2)}{\sum_{k'} P(k'|V^1, V^2) / P(k')} = \frac{\prod_{i=1}^{2} P(k|V^i) P(V^i|P(k)) P(k)}{\sum_{k'} \prod_{i=1}^{2} P(k'|V^i) P(V^i|P(k')) P(k')}$$

$$= \frac{S^1_i S^2_j / P(k)}{\sum_{k'} (S^1_i S^2_j / P(k'))}$$

Where $S^k_i$ stand for the conditional probability computed in Equation (2). $P(k)$ stand for the priori probability of $k$th category. We assume that the behaviors of air traffic controllers according to the same distribution in the training set and the testing set. As a consequence, the final fusion score can be computed by Equation (3).

### 3.3 Network Training

We employ parameters of InceptionV3 model and ResNet50 model on ImageNet and finetune on our CAFUC-Controller dataset. We lock in the weight of some layers of ImageNet, and retrain the weight of other layers. After several experiment, we found that the best training effect in InceptionV3 model is to retrain at layer 172, locking the weight of the layer 0-171. Similarly, the best training effect in ResNet50 model is to retrain at layer 152, locking the weight of the layer 0-151. The optimizer we choose is Adam-optimizer.

### 4. Experiment

We use Keras to construct our neural network model. Our experimental platform: OS: Ubuntu 16.04, CPU: InterCore I7-3820, GPU: NVIDIA GTX 1080Ti.

#### 4.1 Experiment Setup

We create a dataset of controller’s behavior called CAFUC-Controller dataset. It contains 18247 images with ten categories. Our platform for collecting experimental data is the digital radar simulator. Digital Radar Simulator, a professional simulation training system, is developed for training...
g the air traffic controllers. We recruit 10 volunteers and ask them to do the above-mentioned related actions on the simulator. Experimental equipment and dataset samples as shown in Figure 4. CAFUC-Controller dataset contains training set (15,506 images) and testing set (2,741 images). The number and type of controller behavior in training and testing data sets are shown in Table 2. C0-C9 represent different categories of controller’s behaviors.

| Category | Training set | Testing set | Behavior |
|----------|--------------|-------------|----------|
| C0       | 1707         | 192         | putting heads on desks for a nap |
| C1       | 2113         | 158         | lay with face upward |
| C2       | 868          | 247         | nod to sleep |
| C3       | 1691         | 342         | talking with others |
| C4       | 1805         | 388         | directing planes normally |
| C5       | 1976         | 364         | Hold up chin on left hand |
| C6       | 1728         | 328         | Hold up chin on right hand |
| C7       | 1300         | 298         | writing notes on strips |
| C8       | 722          | 108         | radio check |
| C9       | 1596         | 316         | Resting hand on forehead |

Figure 4. Simulator and some dataset samples.

4.2 Experiment Results & Discussion
We use VGG16 as the base model, and then compare the results of each stream's model and fusion model. We list the accuracy of each model and their accuracy for each category. The Table 7 represents the results on the CAFUC-Controller dataset. We abbreviate our two-stream model to TCNN. In order to gain a better understanding of each model, we introduce Class Activation Map to analyze the model [14]. Some body parts play an important role in the controller's behavior. Different colors represent different weights. The red part has the highest weight. From the class activation map as shown in Figure 5, we can see that the weight of these body parts in VGG16 model is not obvious. The weight of these body parts in ResNet50 model and InceptionV3 model is rising.
Table 3. Comparison with different models on CAFUC-Controller dataset

| Model   | Resnet | InceptionV3 | VGG16 | TCNN(early) | TCNN(late) |
|---------|--------|-------------|-------|-------------|------------|
| C0(%)   | 85.3   | 88.1        | 76.8  | **84.1**    | 86.7       |
| C1(%)   | 93.7   | 91.0        | 73.4  | **96.7**    | 95.3       |
| C2(%)   | 89.5   | 81.3        | 75.7  | **91.8**    | 87.8       |
| C3(%)   | 94.3   | 90.7        | 74.1  | **92.5**    | 95.9       |
| C4(%)   | 88.1   | 83.5        | 73.3  | **94.1**    | 86.9       |
| C5(%)   | 89.1   | 87.6        | 74.5  | **92.0**    | 87.1       |
| C6(%)   | 93.2   | 91.1        | 77.4  | **97.6**    | 94.8       |
| C7(%)   | 92.1   | 94.3        | 76.8  | **96.1**    | 95.4       |
| C8(%)   | 91.9   | 95.1        | 81.2  | **97.1**    | 93.7       |
| C9(%)   | 89.8   | 90.3        | 76.3  | **97.5**    | 94.3       |
| Total(%)| 90.7   | 89.3        | 75.9  | **94.1**    | 91.8       |

Table 4. Comparison with different model on Kaggle-Driving dataset

| Model       | Training Accuracy | Testing Accuracy | Training loss | Testing loss |
|-------------|-------------------|------------------|---------------|--------------|
| ResNet50    | 0.9987            | 0.8596           | 0.0038        | 0.7260       |
| InceptionV3 | 0.9995            | 0.9345           | 0.0016        | 0.3046       |
| TCNN(late)  | 0.9973            | 0.9036           | 0.0095        | 0.4296       |
| TCNN(early) | 0.9995            | **0.9376**       | 0.0317        | **0.2900**   |

In order to prove the portability of our model, we also experiment on the open source dataset Kaggle-Driving [4]. It contains 10 types of driver's behaviors. This dataset is similar with our self-built dataset. They are all behavior recognition in fixed position. Table 4 shows the experiment results.

From Table 3 and 4, we can see that our fusion models are better than single model. For the late fusion model, we find that the total accuracy of the late fusion model is slightly higher than the ResNet50 model. But in some classes, the accuracy is lower than that of the single model. The late fusion model cannot obtain the feature correlation between different models. It just combines the scores of individual models. For the early fusion model, we found that the accuracy of the early fusion model has reached 94.1% which is two percentage points higher than ResNet50 model. In the early fusion model, the feature map before the global average pooling layer of each stream is stacked, and the correlation between the features of different model is considered. Therefore, the early fusion model outperforms state-of-the-art.
5. Conclusions
In this paper, we design a two-stream convolution neural network to recognize the behavior of air traffic controllers. Our experiment is based on self-built dataset and open source dataset Kaggle-Driving. The experiments show that our method outperforms the single model can effectively recognize the controller's behavior. However, our model is based on still images and not efficient in video-based recognition. In the future work, we will prepare to optimize the model and implement in the video. We believe that our method can be improved by reducing the size of data input. For example, the most important feature points such as skeleton data can be obtained by eliminating the insignificant information in the image [9].

Acknowledgments
This work was supported by “Research on Flight Planning Centralized Processing and Operational Standards Project” (H2018-14) and a grant from College students' entrepreneurship innovation training program of Civil Aviation Flight University of China (S201910624130).

References
[1] Nealley M A, Gawron V J. The effect of fatigue on air traffic controllers[J]. The International Journal of Aviation Psychology, 2015, 25(1): 14-47.
[2] Mandal B, Li L, Wang G S, et al. Towards detection of bus driver fatigue based on robust visual analysis of eye state[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(3): 545-557.
[3] Gu W H, Zhu Y, Chen X D, et al. Hierarchical CNN-based real-time fatigue detection system by visual-based technologies using MSP model[J]. IET Image Processing, 2018, 12(12): 2319-2329.
[4] https://www.kaggle.com/c/state-farm-distracted-driver-detection
[5] Ergun H, Sert M. Fusing deep convolutional networks for largescale visual concept classification[C]//2016 IEEE Second International Conference on Multimedia Big Data (BigMM). IEEE,2016: 210-213.
[6] Guo J, Gould S. Deep CNN ensemble with data augmentation for object detection[J]. arXiv pre-print arXiv:1506.07224, 2015.
[7] Hoang Nga Le T, Zheng Y, Zhu C, et al. Multiple scale faster-rccn approach to driver's cell-phone usage and hands on steering wheel detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016: 46-53.
[8] Gkioxari G, Girshick R, Malik J. Contextual action recognition with rccn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1080-1088.
[9] Pang B, Zha K, Lu C. Human Action Adverb Recognition: ADHA Dataset and A Three-Stream Hybrid Model[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018: 2325-2334.
[10] Lavinia Y, Vo H H, Verma A. Fusion based deep CNN for improved large-scale image action recognition[C]//2016 IEEE International Symposium on Multimedia (ISM). IEEE, 2016: 609-614.
[11] Qi T, Xu Y, Quan Y, et al. Image-based action recognition using hint-enhanced deep neural networks[J]. Neurocomputing, 2017, 267: 475-488.
[12] Yao B, Fei-Fei L. Modeling mutual context of object and human pose in human-object interaction activities[C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010: 17-24.
[13] Tang P, Wang H, Kwong S. G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition[J]. Neurocomputing, 2017, 225: 188-197.
[14] Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2921-2929.
[15] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]/Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[16] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]/Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.

[17] Ren S, He K, Girshick R, et al. Towards real-time object detection with region proposal networks, 2016[J]. arXiv preprint arXiv:1506.01497.

[18] Zhang Y, Zhou D, Chen S, et al. Single-image crowd counting via multi-column convolutional neural network[C]/Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 589-597