Video-based Violence Detection by Human Action Analysis with Neural Network

Yunqing Zhao[a], Wilton W. T. Fok*[a], C.W. Chan[a]
[a]Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong SAR

ABSTRACT

In recent years, human action analysis is a focal point in video processing, especially on action recognition and safety surveillance. It always performs as an auxiliary tool to minimize the manpower-resource on special tasks. This paper explores the human action analysis in a specified situation, based on the human posture extraction by pose-estimation algorithm. Deep neural network(DNN) methods was used, composed of residual learning blocks for feature extraction and recurrent neural network for time-series data learning. All these modules can be applied on real-time videos, classifying different security levels of actions between two people, with 91.8% accuracy on test set. Meanwhile, some other classical network structures were compared as baselines. After forward inference process of the neural network model, a logic enhancement algorithm was raised and applied in this paper, due to the prediction error between two classes. Experiments were conducted on real-time videos, achieving satisfying performance.

Keywords: Video processing, Pose estimation, Human action analysis, Violence detection, Neural network, Residual learning, Long-Short-Term-Memory

1. INTRODUCTION

With rapid growth on machine learning algorithms in recent years, computer vision has been applied to a wide range of scenes like swimming pool monitoring, license plate recognition(LPR) or face recognition. In some special places, like passages of an apartment or hotel, people always care about the security problems in prohibition of crime or violence. Such events detection or action recognition tasks have been explored by different methods, such as two-stream methods[1], or combination of traditional machine learning algorithms and Convolutional Neural Network[2,18](CNN).

In this research, we proposed a deep-learning-based algorithm, combined with some revised ways to deal with this security level classification problems based on human action analysis. The proposed algorithm can learn from the human posture data extracted by Pose-Estimation Algorithm[3]. Deep residual blocks[4] were used to study abstract residual information and followed by a recurrent neural network[8] to learn from the processed time-distributed data, analyzing dynamic video frames and making predictions of security level. This system can be employed as an auxiliary tool for real-time surveillance video processing for violence detection and precaution.

An increasing number of researchers focus on the action recognition based on human posture related calculation, such as using angles of limbs as input data[5] to detect violence actions. However, few people use time-series information from continuous video frames to make classification, for example, coordinates of multiple person joints in consecutive actions.

In this paper, we calculate the Euclidian distance of two people of corresponding joints on each frame, then we send the distance matrix captured from 20 frames as inputs to neural network models to learn fluctuation regulations and make classification on security level. Due to the random coordinates loss, a logic enhancement algorithm was used to keep the classification result reliable by setting conditions of changing output flag of system. Comparisons were made among different network models, and the experiment results showed that our proposed structure was optimal either on training convergence speed or test property.

*Corresponding author, e-mail: wilton@hku.hk

2. BACKGROUND INFORMATION

2.1 Human Posture Estimation

Firstly, we introduce some related tools. The OpenPose algorithm[3] is a real-time Human Posture Estimation system detecting human joints coordinates by RGB cameras, which means we do not need expensive devices. It can gain 18 key points on human joints and this 18-point-connection forms the human-skeleton used as a vector to localize one people.

2.2 Residual Learning

Convolutional neural network has been a popular method for image classification[6] , feature extraction[14], etc. However, with growing number of network layers, learning capability degradation will appear. A new method called deep residual learning was proposed by *Dr. Kaiming He* in 2016[4], which solved this problem by learning residual information instead, prohibiting gradient vanishment or degradation when executing backpropagation[7]. Figure 1 shows the basic structure of the residual block which contains several convolutional layers in the main path and a shortcut.

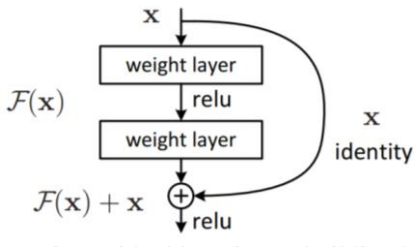


Figure 1. Residual Learning Block

For each residual block:

$$y_l = H(x_l) + F(x_l)$$

$$x_{l+1} = y_l$$

where x_l is the input, $H(x_l) = x_l$ means the identical mapping, $F(x_l)$ represents the residual function and y_l is the output. *f* is the activation function between each residual block. Recursively, we have:

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i)$$

which means, the output of $\boldsymbol{L^{th}}$ residual block can be viewed as the sum of one original input and the inner complex mapping. Moreover, supposing the loss function is ***J***, the calculation in backpropagation process can be written as:

$$\frac{\partial J}{\partial x_l} = \frac{\partial J}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial J}{\partial x_L}(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i))$$

The constant "***1***" ensures the stable gradient when iteratively updating the connection weights in network layers. Thus, we may consider introducing a relatively deeper neural network by this way.

2.3 Recurrent Neural Network

Recurrent neural network[8] (RNN) can handle the time-series data learning problems and perform regression or classification tasks.

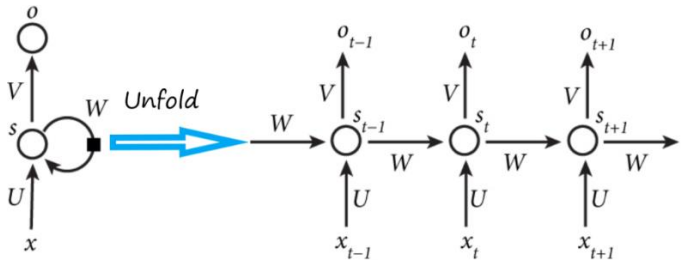


Figure 2. Unfold RNN unit on timeline

In Figure 2, x_t means the input at the time t, W, V, U are connection weights, s_t represents the hidden state in each unit. The output of one unit can be input of the same unit but in the next time step. For example, the input x_t can be one word in a sentence, to let the neural network learn the context information for machine translation[11, 12, 19].

However, the gradient vanishment or explosion will also occur in this normal recurrent neural network format. Fortunately, we can deal with this by employing Long-Short-Term-Memory(LSTM)[9], an improved version of the general RNN, which introduces some logical gate units[11, 12].

Motivated by the safety needs, with combination of these useful tools, we may obtain the exact human location information with their joints, use the data record to analyze the fluctuation regulation between two people with neural networks and make classification of their actions. Followed by the previous work like Amarjot Singh[5] and Aloysius[20], we proposed this real time detection system and make it available for auxiliary monitoring in some special scenes.

3. PROBLEM FORMULATION

3.1 Establishment of Dataset

To complete this violence detection task, we need video clips containing the humans' interaction. Since there is no such public dataset that can express the human joints with accurate coordinates in any situation, we developed this new dataset used in this paper and will make it available to public very soon.

Assuming that the security level can be reflected by fluctuation intensity of interaction between people, for example, the distance of two people's joints may change drastically in a fighting process or keep steady relatively in peace, these videos in the dataset are divided into three categories, representing different security levels named by "***No Warning***", "***Caution***" and "***Fighting***".

3.2 Data Preprocessing

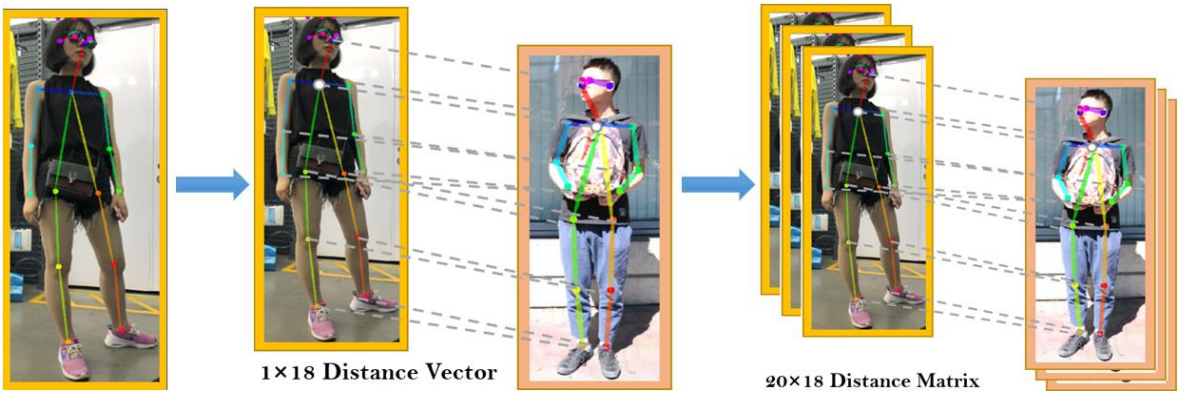


Figure 3. Input distance matrix from captured frames. We extract 1*18 vector between two people(left) and pile the vector on 20 consecutive frames to form a 20*18 matrix as an input training sample(right).

In this project, firstly we use OpenPose algorithm, a real-time multi-person detection system to gain 18 key points on one human with coordinates of each joint(represented by normalized ***x*** and ***y*** real number) [3, 13]. Then, we calculate the Euclidian distance between **two** people, with regard to each corresponding joint, as left part of Figure 3 shows.

In each frame of the video clips, since this algorithm can extract the 18*2 joints coordinates(because of ***x*** and ***y*** coordinates) per person, 18 corresponding distance values can be obtained. These calculation results will be passed as *Numpy* vectors with dimension of 1*18 on every single frame.

Moreover, because the human actions are continuously performed, we cannot recognize the security level by just looking at one frame of the video clip. As the right part of Figure 3 shows, we captured 20 consecutive frames and repeatedly perform the calculation process as the left part of Figure 3. By this way we can form one input data sample with the dimension being 20*18, which will be trained and learnt in neural network.

The final processed dataset we built contains **47666** samples: **35749** training samples and **2383** validation samples and **9534** testing samples respectively, three classes in total(mentioned in chapter **3.1**). Each input sample is of 20*18 matrix format obtained from every 20 consecutive frames.

3.3 Training methods

In this paper, we developed a module-based neural network which can learn the fluctuation intensity regulation from the processed data and classify the security level between two people in real-time. The original training videos are divided into clips with 20 frames of each without overlap, then transformed into 20*18 matrix using preprocessing methods as mentioned in chapter **3.2**. Each input sample(with 20*18 format) will go through training each time in the neural network, the obtained model will make prediction by every 20 consecutive frames on the testing video clips.

4. EXPERIMENTS AND COMPARISON

4.1 Construction of the neural network

In this paper, the proposed module-based network is called "***Module-net***". Figure 4 describes the structure of ***Module-net***. This neural network model firstly employs residual blocks for feature extraction, composed of 1-dimension convolutional layers and the shortcuts[4], followed by a couple of pairs of dense layer[15] and dropout layer[10] which can prohibit the overfitting by randomly de-activating the connection weights. Finally, an LSTM network layer[9] is built for time-distributed data learning, because each input data is extracted from 20 consecutive frames and they are time-related. We built this model such that it can learn from a series of consecutive actions performed between two people and recognize whether they are fighting or not, or just in a caution level.

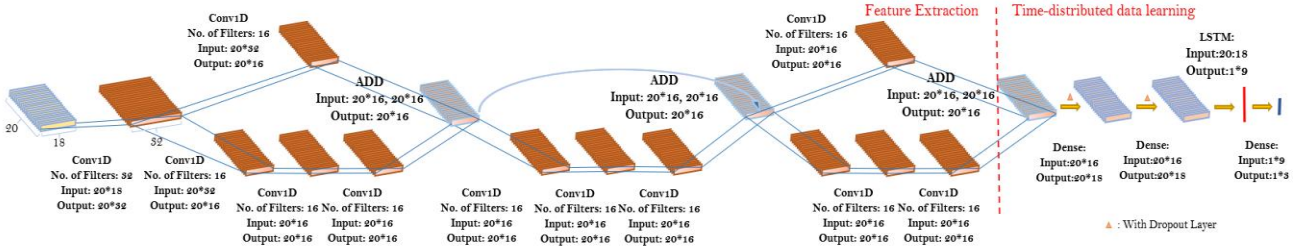


Figure 4. The structure of ***Module-net***. The left-hand side of the dotted line is the Feature Extraction part, by residual learning[4], it can learn more abstract information than that of other classical structures, with nearly no degradation of the gradient. The right part LSTM network[9] can learn from the time-distributed data.

The size of each filter in 1D convolutional layer is set to ***1*n***, where ***n*** denotes the length of input data in each layer. By learning the residual information of the input signal, it can prohibit gradient vanishment and degradation.
The ***L1*** regularization[16] was used for feature selection in the learning process, since not all joints are equal-important in

the distance vector extracted by pose-estimation algorithm, implicitly. Finally, we apply *Cross-Entropy*[17] as the loss function in the learning process in this classification task.

4.2 Training Process and Testing Results

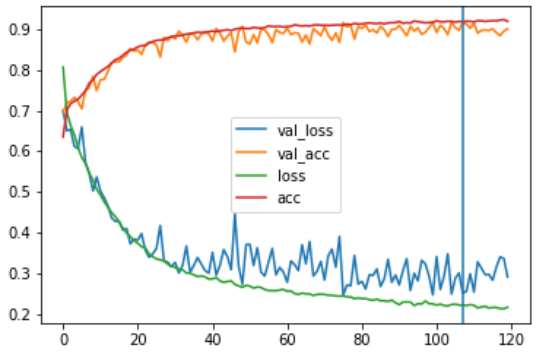


Figure 5. Training process of *Module-net*

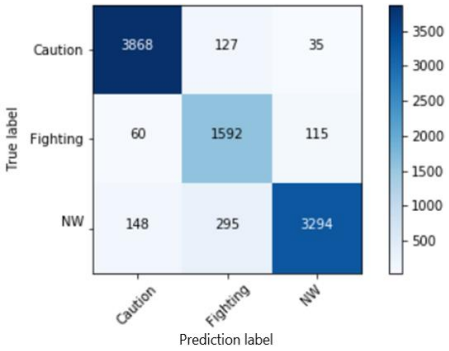


Figure 6. Confusion matrix table of classification result on test set

As has been mentioned before, **35749** samples were used for training and **2383** samples for validation in the training process. **9534** samples for testing to check the learning outcome of the model. In Figure 5, the training process of ***Module-net*** shows the fast convergence at nearly 50 epochs(the abscissa), with relatively high accuracy(the ordinate). The classification results on the test set are summarized as a **3*3** confusion matrix table(the dataset have **3** categories). Each digit in the grid represents the number of test samples with true label on the corresponding ordinate and prediction label on the abscissa. By this confusion matrix table, we can clearly see that the accuracy on test set can achieve 91.8%.

Besides, we also built two classical models as baseline for comparison. Baseline 1 employs some 1-dimension convolutional neural network layers just without special structure like shortcut of the residual block in ***Module-net***(the left part of Figure 4), and then followed by one LSTM layer. Baseline 2 introduces the normal Dense-dropout pair with one LSTM layer. Both the baseline networks have the same settings on hyperparameters such as the number of filters or the filter size, compared with ***Module-net***. We also make sure that these two baseline networks have similar parameters scale with ***Module-net***.

However, they gained poor performance either on convergence speed or accuracy on test set, and other properties as well(Figure 7.1-Figure 9). The proposed ***Module-net*** performs more stable in the training process and the two baseline networks show strong instability, which can be seen from the bluelines(representing the loss value in training process on validation set)as below. Besides, due to serious oscillation of training process of baseline 1, we applied 25% rather than 20% of the dataset as testing data to verify its property.

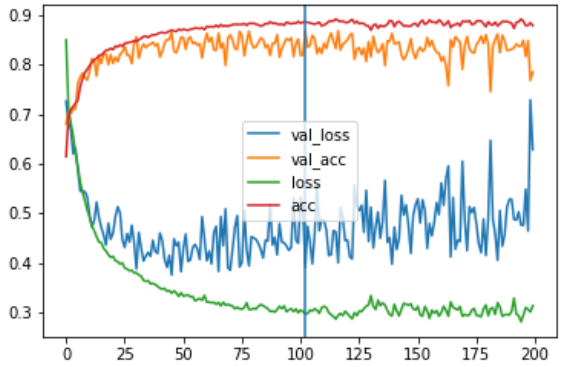


Figure 7.1. Training process of Baseline 1

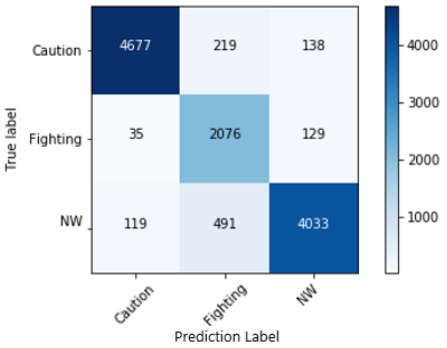


Figure 7.2. Confusion matrix table of classification result of Baseline 1

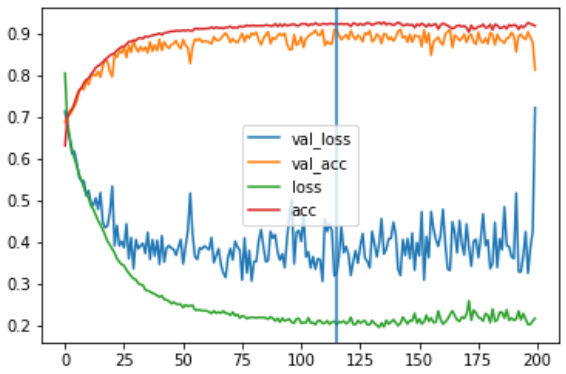


Figure 8.1. Training process of Baseline 2

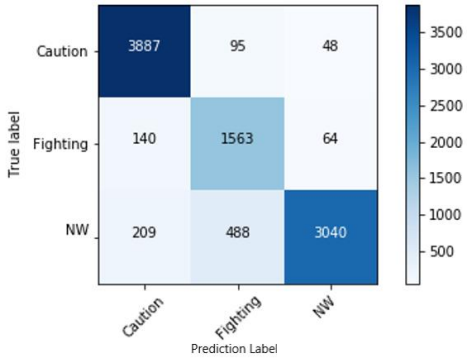
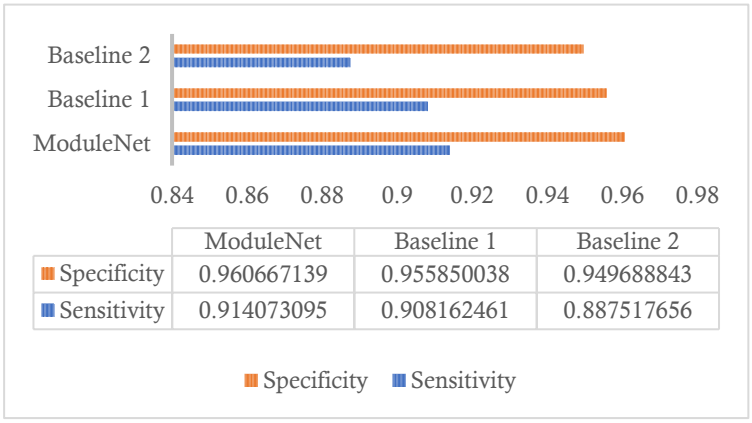


Figure 8.2. Confusion matrix table of classification result of Baseline 2



	ModuleNet	Baseline 1	Baseline 2
Specificity	0.960667139	0.955850038	0.949688843
Sensitivity	0.914073095	0.908162461	0.887517656

Figure 9. Comparison of Specificity & Sensitivity on three models

These three models achieved testing accuracy 91.8(Module-net), 90.5%(Baseline 1) and 89%(Baseline 2) respectively. Figure 9 demonstrates the Specificity & Sensitivity of them. It should be noticed that most of the errors happen when true label of the video clip is ***No Warning***, but the network predicts ***Fighting***. This rendered ambiguous error on these two classes can be attributed to the random human posture estimation data loss affected by noise signal(which will make the value in distance matrix jump sharply, representing the format like ***Fighting***). On the other hand, the short video clip prediction (in order to acquire real-time results) narrows the margins for this kind of errors.

As a solution of this issue, after getting the prediction results from the well-trained model on every 20 frames, we specify an auxiliary *Logical Enhancement Algorithm* to keep the output value from neural network more stable and reliable. Considered that it is hard to recognize that whether people are in the real fighting status or not in a very short time (or affected by noise as mentioned above) even though we made classification using neural network model, we revise this case by setting a hidden state and a current state, aiming at ensuring the reliability of prediction value from the neural network and make it closer to human feelings perceptually.

The hidden status points to the last prediction result, and the current status represents the latest prediction value. If and only if the continuous two prediction results(i.e. prediction results from the consecutive 40 frames) are the same, then the output flag can be switched, except the change between "***Caution***" and "***No Warning***", which means, the final system output label can be switched within these two states totally depends on the real-time prediction result, because we assume that there is no essential difference between them.

This algorithm logic is shown as the following example:

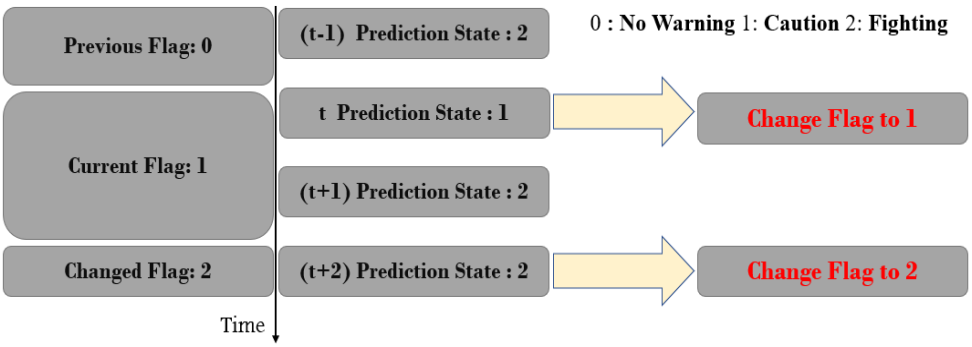


Figure 10. The condition of output flag changing. Only by continuously predicting "***Fighting***" twice will switch the flag of the final output. Considered that the frame per second(FPS) is about 25, we can ignore the discrepancy, and this can improve the robustness of the whole system.

The process expressed by Figure 10 is the final step of this detection system. The whole algorithm logic in this paper can be summarized as follows:

1. *Capture 20 frames from test video to get 20*18 distance matrix as input by using Pose Estimation Algorithm.*
2. *Each input sample will go through the* ***Module-net*** *and gain the prediction label(one of the* ***three*** *classes).*
3. *The prediction value will be verified by the proposed Logic Enhancement Algorithm.*
4. *Output the verified flag as the final label of the primary 20 frames.*

Figure 11. Three kinds of output flag after Neural Network Prediction and Logical Enhancement Algorithm verification

Figure 11 shows three frames captured from testing videos. After prediction by the network model, the Logical Enhancement Algorithm will verify prediction value by setting conditions of flag changing of the system, which can enhance the reliability of the true label. Due to the efficient and light structure of this system, the testing process can be performed in real-time(about 25 FPS) on a normal computer with GPU: NVIDIA GeForce GTX 1050 Ti, which means we do not need extremely high computational capability to carry the safety monitoring task in this situation.

5. CONCLUSIONS

Encouraged by the various violence detection methods proposed by others[5, 20], we developed this module-based system, to classify the security level between two people with input being the Euclidian distance matrix, gained from Human Posture Estimation Algorithm. The proposed "***Module-net***" achieved better performance on test set when compared with other baseline models. We also stabilized our algorithms by a logic algorithm for revising the value-loss-caused prediction error and make the system output closer to the real status perceptually. Furthermore, due to relatively low computational power requirement, this project can be achieved on lightweight monitoring equipment for auxiliary violence detection. Smart unmanned surveillance is a very important issue nowadays, and we hope that our work can be further upgraded in the future. As for potential improvements, since our work depends on the human posture data, if

non-accurate estimation of human joints, non-accuracy output value will be shown(e.g. value-loss prediction error). We may consider discarding this dependence by applying other structures like 3D convolution, but it requires huge computing power, which cannot be implemented in real time on mobile terminals, so it is necessary to explore a balance.

REFERENCES

[1] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." Advances in neural information processing systems (2014).

[2] Bouindour, Samir, et al. "Abnormal event detection using convolutional neural networks and 1-class SVM classifier." 1-6 (2017).

[3] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291-7299 (2017).

[4] He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778 (2016).

[5] Singh A, Patil D, Omkar SN. Eye in the Sky: Real-time Drone Surveillance System (DSS) for Violent Individuals Identification using ScatterNet Hybrid Deep Learning Network. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1629-1637 (2012).

[6] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems, pp. 1097-1105 (2012).

[7] Hecht-Nielsen R. Theory of the backpropagation neural network. In Neural networks for perception, pp. 65-93 (1992). Academic Press.

[8] Elman JL. Finding structure in time. Cognitive science, 14(2):179-211 (1990).

[9] Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." 9th International Conference on Artificial Neural Networks: ICANN '99, 850-855 (1999).

[10] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 1;15(1):1929-58 (2014).

[11] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 1;18(5-6):602-10 (2005).

[12] Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. In Thirteenth annual conference of the international speech communication association (2012).

[13] Toshev A, Szegedy C. Deeppose: Human pose estimation via deep neural networks. InProceedings of the IEEE conference on computer vision and pattern recognition, pp. 1653-1660 (2014).

[14] Chen Y, Jiang H, Li C, Jia X, Ghamisi P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing, 54(10):6232-51(2016).

[15] Yao, Xin. "Evolving artificial neural networks." Proceedings of the IEEE 87.9: 1423-1447(1999).

[16] Ng, Andrew Y. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance." Proceedings of the twenty-first international conference on Machine learning. ACM (2004).

[17] Shore, John E., and Robert M. Gray. "Minimum cross-entropy pattern classification and cluster analysis." IEEE Transactions on Pattern Analysis and Machine Intelligence 1: 11-17 (1982).

[18] Ji, Shuiwang, et al. "3D convolutional neural networks for human action recognition." IEEE transactions on pattern analysis and machine intelligence 35.1: 221-231 (2012).

[19] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[20] Aloysius, Lim Hock Wyi, et al. "Human posture recognition in video sequence using pseudo 2-d hidden markov models." ICARCV 2004 8th Control, Automation, Robotics and Vision Conference. Vol. 1. IEEE (2004).