An ECAPA-TDNN Based Network for Hand Gesture Recognition on Skeletal Data

Yirui Yin *

Shanghai Starriver Bilingual School, Shanghai, 201108, China

* Corresponding Author Email: yirui_yin2024@163.com

Abstract. Due to the high variety of sign languages, it is essential to present a model that could recognize the hand gesture recognition. The state-of-art model is mainly driven by convolution neural networks (known as CNN), and researches are on optimizing CNN architectures. The CNN networks are too large and require long time to train. To address these challenges, we developed a more accurate and robust ECAPA-TDNN structure for recognition. The ECAPA-TDNN is a structure of multiple one- dimensional neural networks with one-dimensional convolution, activation layers, and batch normalization. On the challenging SHREC 2017 3D Shape Retrieval Contest dataset, the ECAPA-TDNN achieved an accuracy of 92.9%, which is 2% higher than the state-of-the-art accuracy achieved by CNNs.

Keywords: Terms—ECAPA-TDNN; hand gesture recognition; Human-computer interaction.

1. Introduction

The application of sign gesture recognition is essential in scenarios including phone applications, industrial applications, and other applications that strongly tie to daily life. All circumstances require a high accuracy model that state-of-art is yet not achieved.

The hand gesture recognition could trace back to 2007, where the study is about image browsing [1] by Y. Fang et al. On the study of June 2014 [2], Raviteja Vemulapalli et al classified the hand gesture by transforming the joints 3D data to vector space, and use a combination of dynamic time warping, Fourier temporal pyramid representation and linear SVM. However, this combination of traditional algorithm is less accurate and time costly. In 2012, Lu Xia et al have conducted algorithm through histograms of 3D joints [3], using the protocol of Li Wanqing et al in 2010 [4]. However, comparing to state-of-art algorithm, this research's result is lower in accuracy.

In 2016, Zhi Liu et al research in three-dimensional Convolutional Neural Network ($3D^2CNN$), building the algorithm with support vector machine (SVM) [5]. It has become the state-of-art solution since then. Gongfa Li has developed over the idea of support vector machine and CNN in 2019 [6]. In addition, the team used error back propagation algorithm to increase the validity and robustness of the whole model. However, the accuracy of $3D^2CNN$ is not promising as pro- posed. In reality, the $3D^2CNN$ is 82% accurate, comparing to a 96% accuracy they proposed [5].

Indeed, most of the recent ideas are based on convolutional neural network, LSTM, and other forms of neural networks on hands gesture recognition [7] [8]. However, most neural networks are large and lower in accuracy. Thus, we use ECAPA-TDNN to solve the time costs and accuracy. We change the multi-channel data into multiple one channel data. We are motivated by the recent progress in speaker verification and propose building a hand gesture recognition model based on ECAPA-TDNN. It has been proven to be accurate and robust [9].

2. Methods

2.1. Feature Description

In hand gesture recognition, the movement of the hand is described by the changes in the vectors that represent the joints. At each joint, the position is identified as a three- dimensional vector. If we name a joint as a multivariate function p, we can express the form as:

$$p_i = (x_i, y_i, z_i) \quad (1)$$

For example, data from 22 joints, each with a vector pi, can be devoted as a 66-dimensional vector:

$$s_i = (s_1, s_2, s_3, \cdots, s_{66}) \quad (2)$$

From an image, we can extract the position of each joint, as well as the depth of the joint, which can be represented as a combination of a 66-dimensional vector. Assuming we have m images, the vector will be a $66 \times m$matrix.

2.2. ECAPA-TDNN for hand gesture recognition

Figure 1 shows the network structure of ECAPA-TDNN. It is a methodology developed from the Time Delay Neural Network (TDNN) architecture, which is very efficient for speaker verification. ECAPA stands for Emphasized Channel Attention, Propagation, and Aggregation, which is about the characteristic of the model: attention based [9].

The model first has an input of one-dimensional variables. Then, it uses a one-dimensional convolution layer as the first layer for the input. This is then modified with Rectified Linear Units (ReLU, non-linearities). The ReLU layer is empirically the performing active layer. It is further re-centered and re- scaled by a batch normalization (also known as batch norm, BN) layer with$k = 5$, kernel size equals 5. The dilation spacing, written as d in the graph, increases by layers.

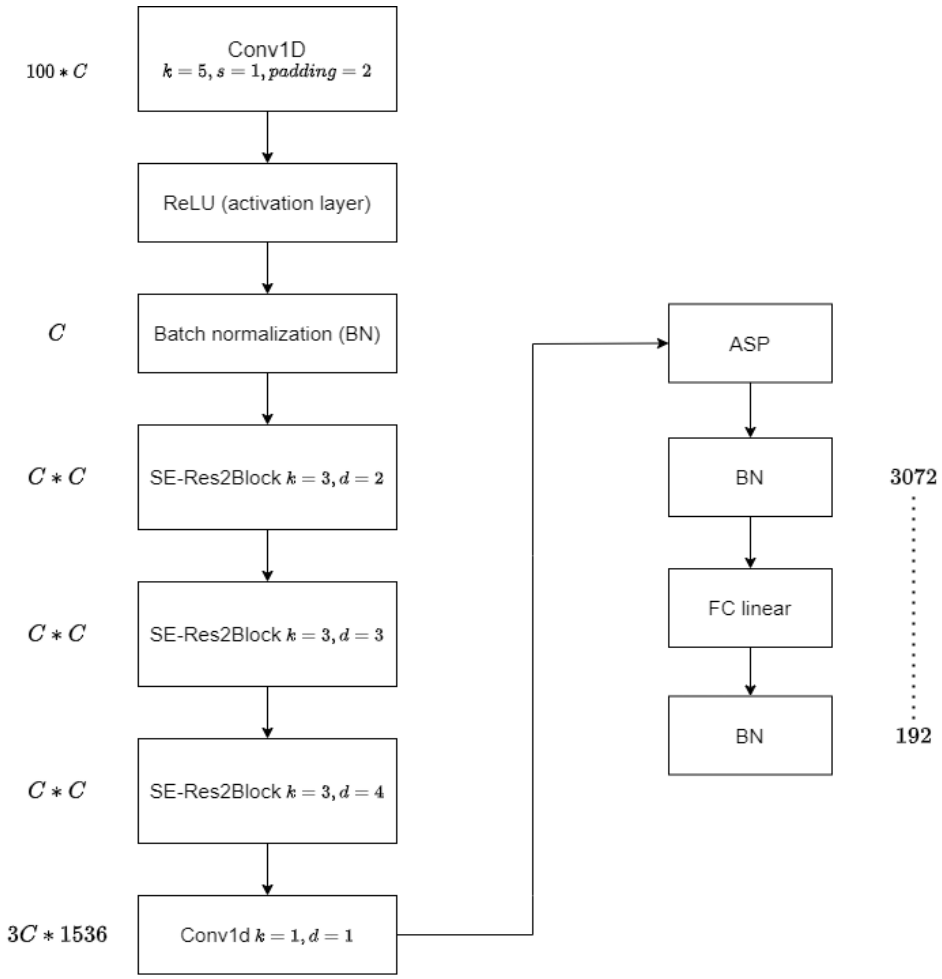

Fig 1. The Network Structure of ECAPA-TDNN. C stands for the channels and sizes. k is representing the kernel size. s is stride of the Conv1D layer.

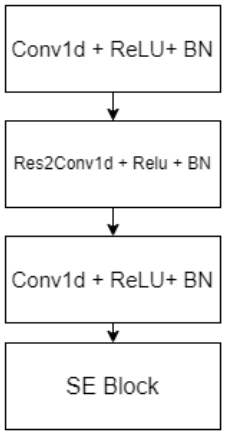

Fig 2. The network structure of SE-Res2Block, which has a kernel size of.

There are three SE-Res2Blocks in the architecture. Each of the SE-Res2Blocks contributes to the final convolution layer and ReLU layer. In other words, the signals are still kept just like in a time delay neural network, where the first layer contributes to the final classification layer.

In each SE-Res2Block, shown in Figure 2, the network will have four blocks. Within the first three blocks, there is a combination of a Conv1D layer or a Res2Conv1D layer and ReLU activation layer, and batch normalization layer. The last block is a SE Block that enhances the performance of the result. The C in the graph means the number of channels, which is 66 since there are twenty-two joints with two- dimensional coordinates and depth information. Each channel is a small neural network that generates the results.

In the RE-Res2Block, each is built by a similar structure but with Res2 Dilated central Res2Net [10]. The FC block in Figure 1 and the SE-Block denote attentive statistics pooling and Squeeze-and-Excitation block [11]. In some situations, researchers use other attentive statistics pooling layers as well. By the feature structure indicated in Figure 3, it will generate a result based on a softmax layer with 66 channels.

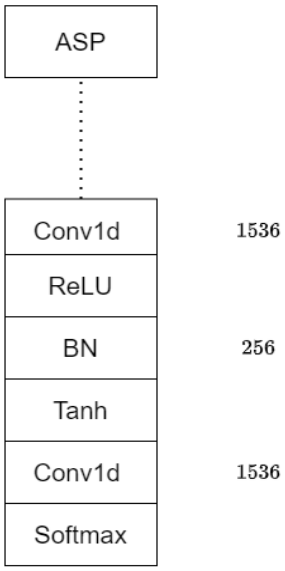

Fig 3. ASP network structure that elaborates on the block in Figure 1.

In the ASP structure that Figure 3 further explains on the architecture in Figure 1, the ASP model changes the $3C * 1536$ vector to the one-dimensional vector. The ASP model demonstrates the a six-layer process that begins with introducing a Conv1D with 1536 in size. It is used as a similar structure comparing to Re-Res2Block. In ASP, Tanh layer could be used to increase accuracy and stability. The key idea of ASP is to decrease the size of the vector first, then turn the size back, which could increase the robustness.

3. Experiment

3.1. Dataset introduction

In the essay, the methodologies are used on SHREC 2017 [12], an online open data set. The data set has two characteristics.

All models are in result of a 3D image with 22 joints. Beyond the two-dimensional picture, the data set has an axis of depth. It provides a more complex circumstances, and add accuracy for the test result by the model.

The data set contains 28 participants for testing. It brings more uncertainty and close up to real life applications. All the participants could have different joints data which ends up different in the neural networks. Thus, the data set can prove the stability of the networks.

3.2. Convolutional Neural Networks

The Convolutional Neural Networks (CNN) have proven to be good at image recognition, including gesture recognition and body recognition as well [8]. The CNN also has great performance in famous Yolo algorithm. However, the state-of- art is taking long time and calculation in convolution layers and pooling layers. In another word, the multi-dimensional could meet difficulties of time costs in effective dataset.

In each Convolution Neural Network, the current neuron is the value of last the last neuron n_{last} times the weight of the neuron w_{last}. The last level is the sum of all related neurons,

$$f = \sum_i n_i * w_i + b \tag{3}$$

In which b is standing for bias from the previous calculation. Then, the output of equation (3) will go through the activation function, which could be ReLU, Sigmoid or other linear, non-linear activation function. The result will be a number within 0 to 1 that represents probability of different categories. Some researchers demonstrate a possible resolution for using a “high and low” strategy to optimize the CNN network [8]. The training methods divided the part into three unique branches.

In the higher branch, each convolution layer has a kernel size of 7, comparing to the lower branch which has a kernel size of only 3. The other branch is used as a bias branch that add in the full connect layer in concatenation.

During their experience, the σ, which is activation function, is at best accuracy using ReLU function,

$$Re\,L\,U\big(f(x)\big) = max\big(0, f(x)\big) \tag{4}$$

The use of maximize, in another word max pooling method, is empirically the best performing algorithm in both one-dimensional matrix and multi dimension matrix. By the calculation, it should be 0.88% more accurate than the algorithm of using average pooling method.

The successful use of one dimension data, present a possible resolution that used in signal processing. The “high and lower” branch structure and all the optimization has improved the challenging data set to a state-of-art highest accuracy rate. However, the algorithm is still limited to CNN structures.

3.3. ECAPA-TDNN

The ECAPA-TDNN is presented with a strong ability of recognizing the hand gesture during the experiment, with a test result of 92.97% in 200 epochs of training. The time is significantly less than the training time of CNN epochs. On the Google Colab platform with T4 GPU for calculation, the training last for 12 minutes and 37 seconds. It demonstrates a strong possibility for large model training.

In this model of 66 channels with 14 classes, we use a learning rate of 0.001 at first. Then, with each 20 epochs, we decreased to 55% of the latest learning rate. The losing rate is, thus, a linear

increasing rate that could better control the learning accuracy of the model, preventing it from over-fitting and decrease the accuracy results.

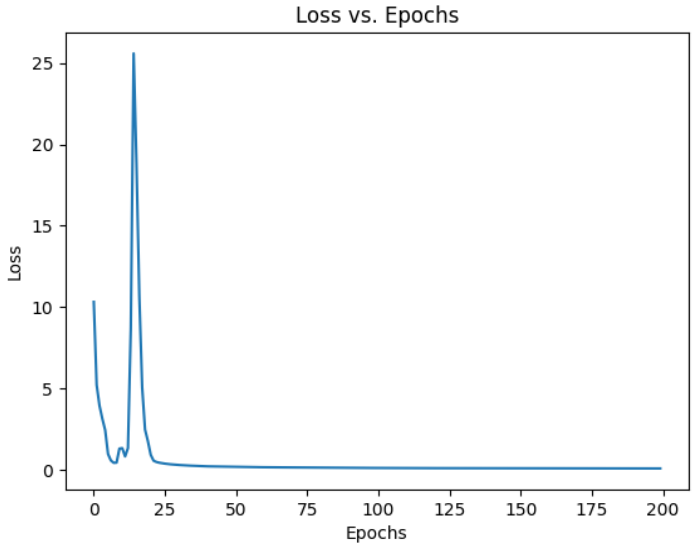

Fig 4. The losing of the model given the range of epoch from 1 to 200.

In the Figure 4, the loss of each epoch is decreasing as the epoch increase. It is due to the learning rate decrease as the epoch increases. However, it helps to prevent the model from over-fitting. There is also a sharp increase around epoch 14 to epoch 15, which is fitting to the sudden decrease in training accuracy and test accuracy in the following.

We also see a significant slower grow of accuracy rate after 50 epochs. However, the changes still exist which is harder to see on the graph. We divided the dataset as 70% training data, and 30% of testing data. We also find the stability in the changing of loss for each epoch. We think it helps the increase of accuracy. Moreover, it demonstrates the robustness of the model.

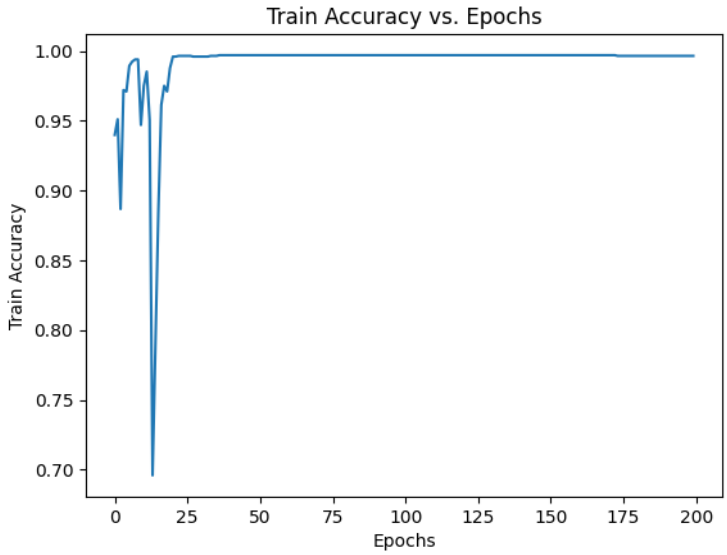

Fig 5. The training accuracy of the model given the range of 1 to 200.

In the Figure 5, the training accuracy comparing to the epochs ranging from 1 to 200. The training of the ECAPA-TDNN, due the multiple layers in ECAPA-TDNN, has a high accuracy at an early stage. It is 87.02% during the first epoch, while there is a significant increase during the first few epochs. However, there is a sharp decrease in epoch 14 and epoch 15 that is 63.81% and 74.64% accurate. Then, there is a bounce back to 88.45%. Further, it is not happened in other epochs, which proves a larger training epoch number is necessary for ECAPA-TDNN despite a well performance in the beginning.

In the end, there is a drop from 93.095% accuracy to 92.976%. We take this 0.119% changes as a reasonable flow. However, it is possible direction to make ECAPA-TDNN a better performance architecture.

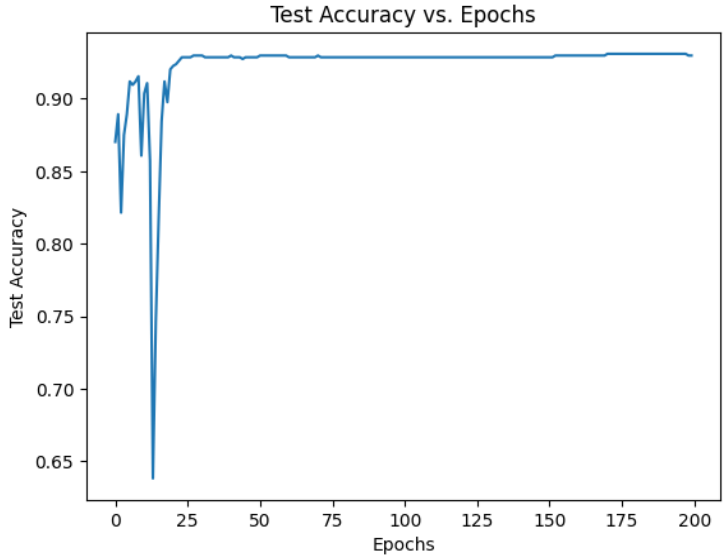

Fig 6. The testing accuracy of the model given the range of 1 to 200.

The Figure 5 shows a similar change with Figure 6. However, the training accuracy of the model is still increasing comparing to the stationary performance of test accuracy in the later 30 epochs. It also demonstrates a possibility of increase with a different dataset or parameters.

We also test with multiple learning rates and changes for the learning rate. Empirically, a high learning rate in the process demonstrate a poor performance. Especially constant learning rate can be over-fitting after around 30 epochs. In practice, the decreasing learning rate can significantly help on increase accuracy. The accuracy increases from 83% to over 90% when we imply the application of decreasing learning rate.

Comparing to the set of 55% of the latest learning rate for every 20 epochs, 60% per every 20 epochs also demonstrate a high accuracy of 91%. However, we find the accuracy struggle to increase when we imply 80% of the latest learning rate for every 20%. During the experiments, we find the range better be 50% to 75% depending on other parameters of the neural network.

3.4. Comparing the results

In the classification, the ratio of the precision is defined as the equation

$$precision = \frac{TP}{TP+FP} \tag{5}$$

In the equation (5), the TP is the number of true positive and $FP$is the number of the false positive. The recall rate is expressed as the equation

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

In the equation (6), the TP is the number of true positive and $FN$is the number of the false negative. The accuracy rate we used in comparison is the $F1$rate, which is expressed as,

$$F1 = \frac{2*precision*recall}{precision+recall} \tag{7}$$

The $F1$ perceive recall rate as important as precision rate. It is also meaningful in the comparison between different algorithms. By the equation (7), the accuracy rate lays within 0 to 1.

Table 1. Comparison of accuracy.

approaches	accuracy
Histograms of 3D joints [13]	78.97%
k-nearest neighbors & Riemannian geometry [15]	79.61%
recurrent neural network (RNN) [16]	84.68%
random forest [14]	90.00%
high and low structure CNN [8]	91.28%
ECAPA-TDNN (ours)	92.97%

In the experiments, multiple trials are used in terms of methodology introduced above. In the research, traditional convolution neural network could achieve an accuracy of 78.97% [13]. The k-nearest neighbors combining with Riemannian geometry receives a performance of 79.61% [15]. In some cases, random forest could achieve a result of 90.00% accurate [14]. Comparing to other neural network architecture, including RNN and CNN which is 84.68% and 91.28% accurate respectively [15] [8], our work performed a 92.97% accuracy in testing. Under the complexity and camera restriction of SHREC 2017 with 14 kinds of gesture to recognize, few can achieve our accuracy and robustness.

4. Conclusion

In our research, we prove the possibility that ECAPA-TDNN, a model in sound recognition that perform well in one dimensional signal, could work in picture and complicated multi-dimensional situations. We divided the information of joints in hand gestures to multiple one-dimension signals, providing an accurate and robust model using ECAPA-TDNN. The model is much quicker in train and more reliable under complicated dataset and increasing number of gestures. We think the future development of hand gesture recognition could developed on the model of ECAPA-TDNN. The accuracy could increase. In addition, the datasets could be more complicated to test the model.

References

[1] Y. Fang, K. Wang, J. Cheng, and H. Lu, “A real-time hand gesture recog- nition method,” in *2007 IEEE International Conference on* Multimedia *and Expo*, 2007, pp. 995–998.

[2] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588–595, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID: 1732632

[3] L. Xia, C.-C. Chen, and J. K. Aggarwal, “View invariant human action recognition using histograms of 3d joints,” in *2012 IEEE Computer Soci- ety Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 20–27.

[4] Z. Z. L. Z. Li, W., “Action recognition based on a bag of 3d points,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885616300592

[5] Z. Liu, C. Zhang, and Y. Tian, “3d-based deep convolutional neural network for action recognition with depth sequences,” *Image and Vision Computing*, vol. 55, pp. 93–100, 2016, handcrafted vs. Learned Representations for Human Action Recognition.

[6] G. Li, H. Tang, Y. Sun, J. Kong, G. Jiang, D. Jiang, B. Tao, S. Xu, and H. Liu, “Hand gesture recognition based on convolution neural network,” pp. 2719–2729, 2019.

[7] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, “An end-to- end spatio-temporal attention model for human action recognition from skeleton data,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/ view/11212

[8] G. Devineau, F. Moutarde, W. Xi, and J. Yang, “Deep learning for hand gesture recognition on skeletal data,” in *2018 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018)*, 2018, pp. 106–113.

[9] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Em- phasized channel attention, propagation and aggregation in tdnn based speaker verification," 2020.

[10] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," 04 2019.

[11] J. Heo, H.-s. Shin, J.-H. Kim, C.-y. Lim, and H.-J. Yu, "Convolution channel separation and frequency sub-bands aggregation for music genre classification," 11 2022.

[12] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, "Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset," in *3DOR-10th Eurographics Workshop on 3D Object Retrieval*, 2017, pp. 1–6.

[13] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," in *Sixth IEEE International Conference on* Automatic *Face and Gesture Recognition, 2004. Proceedings.*, 2004, pp. 529–534.

[14] Z. Jinqing, Z. Feng, C. Xu, H. Jing, and W. Ge, "Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition," *Multimedia Tools and Applications*, vol. 76, pp. 1–20, 10 2017.

[15] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3d human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015.

[16] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2017. [Online]. Available: https://doi.org/10.1109%2Ficip.2017.8296809.