Smart Attendance Management System Based on Face Recognition Using CNN

Syam Kakarla†, Priyaranjan Gangula‡, M.Sai Rahul§, C. Sai Charan Singh¶ and T. Hitendra Sarma*

Srinivasa Ramanujan Institute of Technology
Anantapur, Andhra Pradesh, 515 701, India
Email: †Syamkakarla98@gmail.com, ‡priyaranjanreddy.g@gmail.com,
§msairahul98@gmail.com, ¶chandrapalsaicharansingh@gmail.com, *hitendrasarma@ieee.org

***Abstract*—Convolutional Neural Networks has been playing a significant role in many applications including surveillance, object detection, object tracking, etc. Extensive research is recorded for face recognition using CNNs, which is a key aspect of surveillance applications. In most recent times, the Face Recognition technique is widely used in University automation systems, Smart Entry management systems, etc. In this paper, a novel CNN architecture for face recognition system is proposed including the process of collecting face data of students. Experimentally it is shown that the proposed CNN architecture provides 99% accuracy. Further, the proposed CNN framework is used to develop a "Smart Attendance Management System (SAMS)", which is a web-based application, to provide attendance of students using face recognition, in realtime. The proposed application is easy to deploy and maintain.**

***Keywords*—Face recognition, Convolutional Neural Networks(CNNs), Deep Learning, Data Augmentation, Smart Attendance Management System (SAMS).**

I. Introduction

Face Recognition is one of the popular physiological bio-metric techniques, where the odds of falsifying the user information is less and more credible. It is more secure when compared to other physiological bio-metric techniques such as Fingerprint Recognition, Iris Recognition and behavioral bio-metric techniques such as Signature Recognition, Voice Recognition and Keystroke Recognition.

Surveillance [1] is a method of monitoring and it also assists in keeping track of the activities by recording the footage of the region where it is installed. Due to the advances in technology of the surveillance cameras, it can play a major role in identification, authentication, criminal detection and so on.

The rise of deep learning has laid the path in solving lot of problems which have been considered as hard tasks for the traditional machine learning algorithms. Especially, CNNs have revolutionized the way of approach in solving problems related to Computer Vision and Image Processing there by providing efficient results. CNN based *face recognition* system has gained popularity due to its efficiency in providing authenticity by identifying the faces automatically [2], [3].

Convolutional Neural Network(CNN) is a category of neural network architectures [4], which extracts the features implicitly without any prior intuition, in contrast to conventional machine learning algorithms where feature selection or feature extraction is necessary. The CNN has gained lot of attention due to its generalisable nature in providing feasible solutions in computer vision related problems such as face recognition [5], object detection [6], image segmentation etc.Over the years, the CNN architectures have evolved to address the problems in computer vision. Few of such popular architectures are LeNet [7], AlexNet [8], VGG16 [9], Inception, etc.

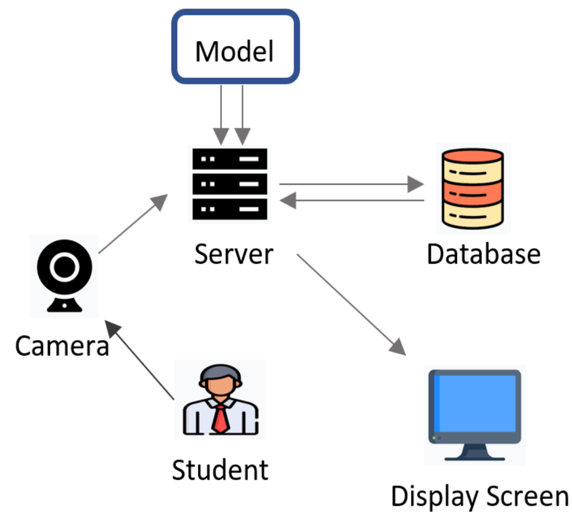


Fig. 1: Overview of the proposed application - SAMS

There are many efficient architectures [10]–[12] proposed in the literature for smart attendance using face recognition. However, the generalisability of these models *i.e.,* to handle a real world scenario, is poor. So it is necessary to develop custom models corresponding to the applications.

This paper proposes a novel CNN architecture motivated by the popular architectures that are published in recent times and also develops a web application for online attendance posting [13], called Smart Attendance Management System (SAMS).

The major contributions of the paper are

- Novel CNN Architecture for end-to-end face recognition system.
- Collecting face data of students though an automated

system.

- Smart Attendance Management System (SAMS): A web-based application.

This paper is organized as follows. Section II presents an overview of Smart Attendance Management System (SAMS) along with the Data Collection and Augmentation. The proposed CNN architecture is explained in Section III. Experimental study is presented in Section IV. Section V provides the Conclusion.

II. Smart Attendance Management System (SAMS)

This section describes the proposed web-based application for Smart Attendance Management System(SAMS) using face recognition by CNNs. This section also presents the challenges of acquiring huge train and test datasets for real time applications.

A. *SAMS - An Overview*

The developed web based application for attendance posting [14] based on face recognition is called "Smart Attendance Management System (SAMS)[1]. It is developed using *flask framework* and *python*. An overview of the proposed application is presented in Fig-1.

The main components of the application are:

1) Front-End Interface (Camera)
2) Server
3) CNN Model
4) Database
5) End Result (Display Screen)

Front-End Interface is a web page which takes the video frame as input through the camera using *open computer vision (OpenCV)* library. **Server** is the back-end of the application which maintains the connection with front-end interface, model, database and display screen. The server receives video frame as input from the front-end interface and the face is detected, which is given as an input to the **model** and the model predicts the label. The server checks if the attendance is posted previously. If not, the attendance is posted into the **database** including the timestamp. The details of the corresponding student are shown using the **display screen**, which is web page. The Fig-2 represents the images of front-end interface and end result of the application.

B. *Data Collection*

The face data of the students is collected using an automated system, which accesses the system camera to take the video frames and transforms as a dataset by performing the following sequence of operations:

- Identify the location of face in the video frame.
- Extract the face image and convert into gray scale image.

[1]The code for the proposed web application and CNN model including training and deployment is open-sourced in GitHub (https://github.com/syamkakarla98/Face_Recognition_Using_Convolutional_Neural_Networks)

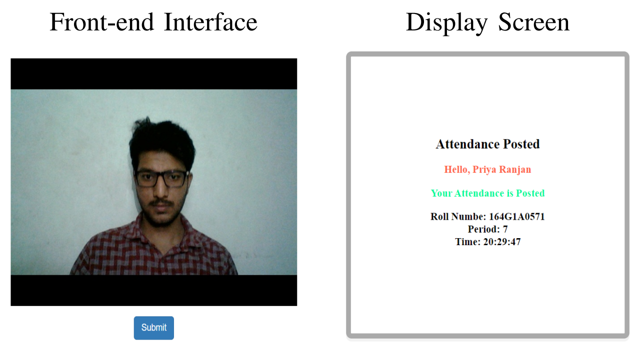


Fig. 2: A Sample input image and the corresponding output

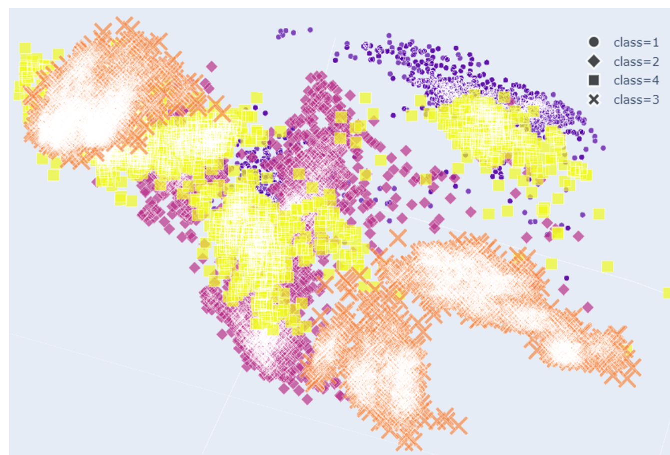


Fig. 3: Three Dimensional View of the Face Dataset

- Attach the label w.r.t to the class of the image and write onto a *csv* file.

The sample images from the collected data with respect to each class is shown in Fig-4. *Principal Component Analysis (PCA)* is used to reduce the dimensions of the data into three dimensions and corresponding three dimensional scatter plot is constructed for understanding the patterns in the data. It is shown in Fig-3.

C. *Data Augmentation*

In real time scenario, collecting huge amount of data is a difficult and tedious task. In order to achieve high accuracy, large volume of training data is required. Hence, *data*

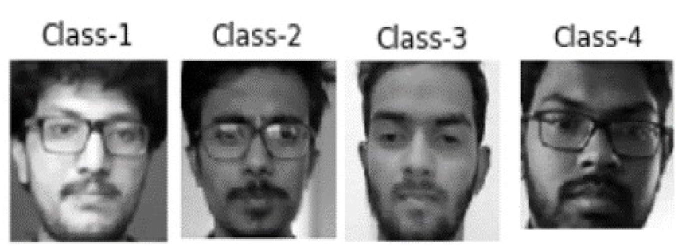


Fig. 4: Sample Images of the Dataset with their Class Labels

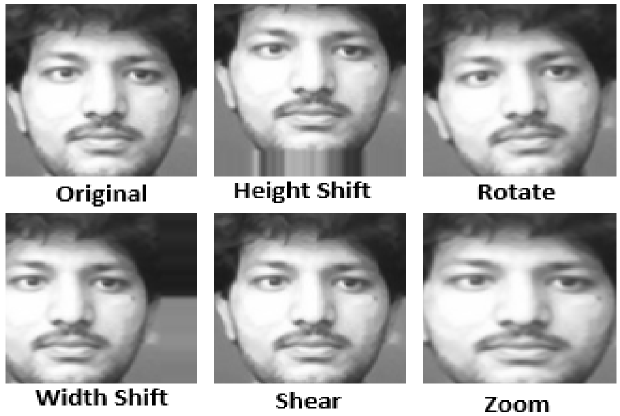


Fig. 5: Sample Images after Data Augmentation

augmentation [15] technique is used to generate new samples by manipulating the existing data. More recently, Masi *et.al.,* [16] presented a novel Face-specific data augmentation for unconstrained face recognition. In the current work, synthetic images are being generated randomly, by applying the following operations

- Zoom
- Shear
- Height shift
- Rotation
- Width shift

The synthetic images generated after data augmentation are shown in Fig-5.

D. Train and Test data

The initial samples collected to form the dataset are 10,029 face images which contains four classes. Each class consists of 2,500 face images except class-3 which contains 2,529 face images. The dataset is normalized during training to boost up the processing and the train data is augmented during the training process of CNN. The details of train and test data are shown in Table-I

TABLE I: Details of Train and Test Data

	Percentage	#Samples
Train	85	8524
Test	15	1505

III. Proposed CNN for SAMS

This section presents the detailed explanation of the proposed CNN architecture for this particular application, SAMS. It also covers the explanation for different layers used to build the CNN model.

The proposed CNN architecture consists of 20 layers which includes:

- Two Dimensional Convolutional Layer (Conv2D)
- Batch Normalization Layer
- Max Pooling Layer
- Dense Layer

The two dimensional convolutional layer is used to extract the features from the previous input, the batch normalization layer is used in order to normalize the input and also overcome the problem of vanishing gradient [17] and exploding gradient. The max pooling layer is used to reduce the dimensionality of the input and the dropout layer is used to avoid the over fitting problem.

The architecture takes a gray scale image with shape (100, 100, 1) and results the class label of the image as prediction. The total number of parameters of the CNN are 7,658,629 of which 7,656,197 are trainable and 2,432 are non trainable. The detailed description of the architecture is shown in Table-II

TABLE II: Architecture of the proposed CNN

Layer	Output Shape	#parameters
Conv2D	(None, 98, 98, 64)	640
Batch_Normalization	(None, 98, 98, 64)	256
Conv2D_1	(None, 96, 96, 64)	36928
Batch_Normalization_1	(None, 96, 96, 64)	256
Conv2D_2	(None, 96, 96, 64)	102464
Batch_Normalization_2	(None, 96, 96, 64)	256
Max_Pooling2D	(None, 48, 48, 64)	0
Dropout	(None, 48, 48, 64)	0
Conv2D_3	(None, 46, 46, 128)	73856
Batch_Normalization_3	(None, 46, 46, 128)	512
Conv2D_4	(None, 44, 44, 128)	147584
Batch_Normalization_4	(None, 44, 44, 128)	512
Conv2D_5	(None, 44, 44, 128)	409728
Batch_Normalization_5	(None, 44, 44, 128)	512
Max_Pooling2D_1	(None, 22, 22, 128)	0
Dropout_1	(None, 22, 22, 128)	0
Conv2D_6	(None, 20, 20, 256)	295168
Batch_Normalization_6	(None, 20, 20, 256)	1024
Max_Pooling2D_2	(None, 10, 10, 256)	0
Dropout_2	(None, 10, 10, 256)	0
Flatten	(None, 25600)	0
Dense	(None, 256)	6553856
Batch_Normalization_7	(None, 256)	1024
Dense_1	(None, 128)	32896
Batch_Normalization_8	(None, 128)	512
Dense_2	(None, 5)	645

The intermediate layer visualizations of the convolutional layers Conv2D_1,Conv2D_2, Conv2D_3, Conv2D_4, Conv2D_5 of the CNN model after training process with respect to an input image is shown in Fig-6

IV. Experimental Results

The proposed Convolution Neural Network(CNN) is developed using an open source python library, Tensorflow [18]. The experiments are performed on the system with following configuration.

- **Graphical Processing Unit (GPU)** used is 1X Tesla K80 with 2496 CUDA cores, 12GB GDDR5 VRAM
- **Central Processing Unit (CPU)** used if 1X single core hyper threaded Xeon Processors, 45MB Cache with 12.6 GB RAM and 320GB disk.

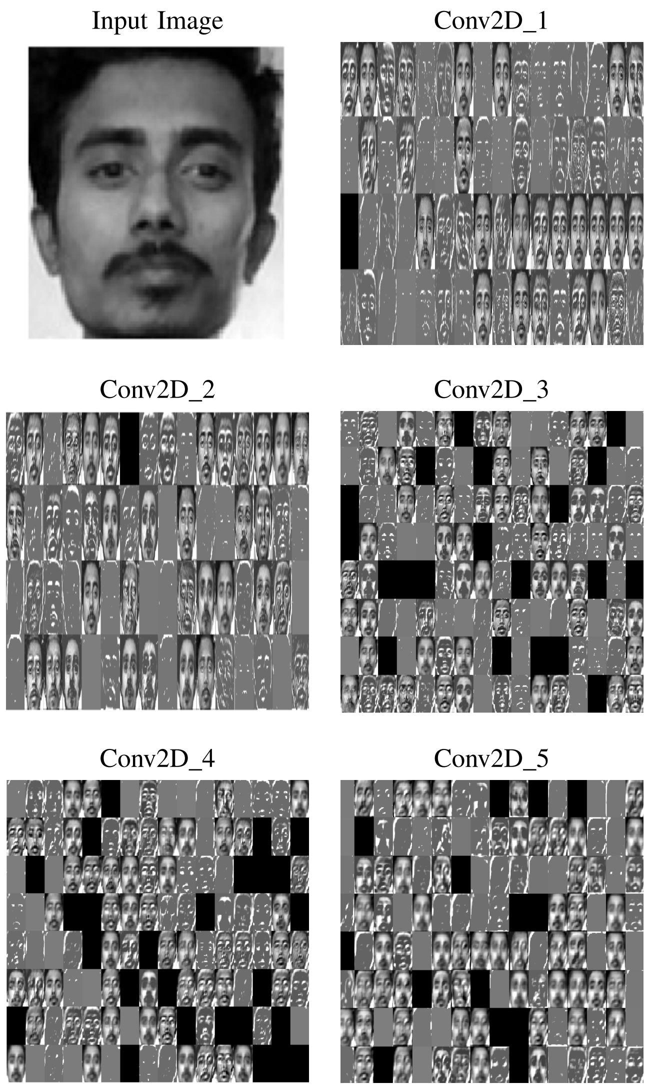
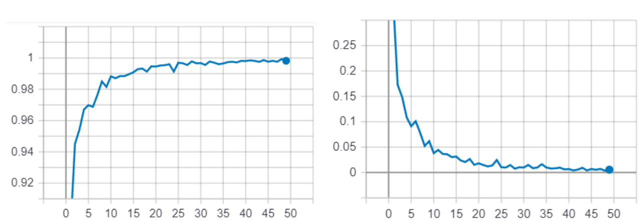


Fig. 6: Conv2D Layer Visualisation



(a) (b)

Fig. 7: (a) Increase of accuracy w.r.t #epochs (b) Decrease of loss w.r.t #epochs

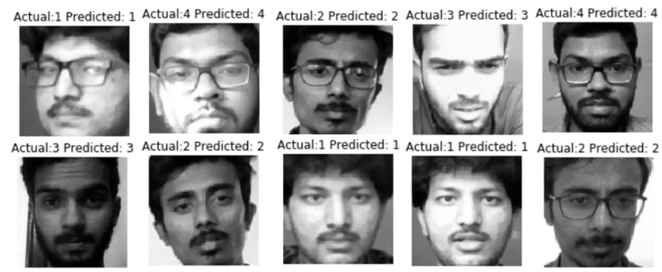


Fig. 8: Predicted Labels of Test Images

A. *Training*

The proposed Convolutional Neural Network (CNN) is trained using RMSprop optimizer with learning rate 0.001, categorical cross entropy as loss metric, 256 as batch size over 50 epochs.

The ReduceLROnPlateau, EarlyStopping, TensorBoard and ModelCheckpoint callbacks are used during the training of CNN for reducing the learning rate if there is no improvement in the learning process of CNN, for stopping the training process if there is no improvement in accuracy, visualizing model graphs and checkpointing the model weights for future use respectively.

The accuracy and the loss of the CNN during the training with respect to the number of epochs are represented in Fig-7-a and Fig-7-b respectively. It is evident that with increase in number of epochs the accuracy is increasing and loss is decreasing and at a certain number of epochs, both the loss and accuracy do not change further.

B. *Analysis of Classification Accuracy*

The performance measures for classification are *Kappa* statistic [19] which is used to find out the inter-rater reliability, *Precision* which represents the ratio of correctly predicted positive observations to total predicted positive observations, *Recall* which is the ratio of correctly predicted positive observations to the all observations in actual class.

F1 score is the weighted average of both *Precision* and *Recall*. The class-wise accuracy and the above mentioned performance measures are depicted in the Table-III and Table-IV, respectively.

The predictions of the proposed CNN model with test images delivers imperial result, which is shown in Fig-8 and the confusion matrix is shown in Fig-9

TABLE III: Class-wise Accuracy

Class Label	Accuracy
1	100.0%
2	99.49%
3	100.0%
4	100.0%

Finally, the **accuracy** and **loss** of the CNN model are recorded as **99.86%** and **0.0057** respectively.

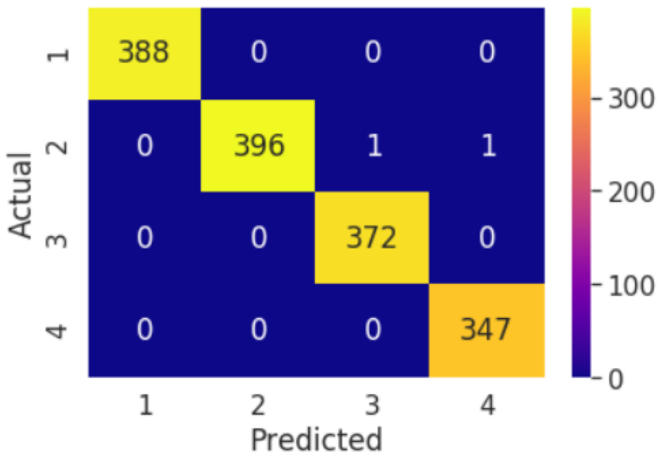


Fig. 9: Confusion Matrix

TABLE IV: Summary of Classification Accuracy

Overall Accuracy	99.8671%
Average Accuracy	99.874%
Kappa Score	99.8226%
F1-Score	99.8674%
Precision	99.8611%
Recall	99.8743%

V. Conclusion

This paper formally introduces the role of CNN in Face Recognition and adaption of CNN in attendance posting. The workflow of web application, Smart Attendance Management System (SAMS) is explained in detail. The data collection and data augmentation for developing the CNN model is discussed. This paper also proposes a novel CNN model for face recognition which is further used in developing SAMS. The experimental results shows efficiency of the proposed CNN model and the web application SAMS. SAMS is easy to deploy and maintain. The future scope of the paper is to build a robust application for smart attendance management, for more number of students, in real time.

References

[1] K. Sun, Q. Zhao, J. Zou, and X. Ma, "Attendance and security system based on building video surveillance," in *International Conference on Smart City and Intelligent Building*. Springer, 2018, pp. 153–162.

[2] M. Coşkun, A. Uçar, Ö. Yildirim, and Y. Demir, "Face recognition based on convolutional neural network," in *2017 International Conference on Modern Electrical and Energy Systems (MEES)*. IEEE, 2017, pp. 376–379.

[3] M. He, J. Zhang, S. Shan, M. Kan, and X. Chen, "Deformable face net for pose invariant face recognition," *Pattern Recognition*, vol. 100, p. 107113, 2020.

[4] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 142–150.

[5] M. Swapna, Y. K. Sharma, and B. Prasad, "A survey on face recognition using convolutional neural network," in *Data Engineering and Communication Technology*. Springer, 2020, pp. 649–661.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[7] A. El-Sawy, E.-B. Hazem, and M. Loey, "Cnn for handwritten arabic digits recognition based on lenet-5," in *International conference on advanced intelligent systems and informatics*. Springer, 2016, pp. 566–575.

[8] Z.-W. Yuan and J. Zhang, "Feature extraction and image retrieval based on alexnet," in *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, vol. 10033. International Society for Optics and Photonics, 2016, p. 100330E.

[9] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018, pp. 169–175.

[10] S. Khan, M. H. Javed, E. Ahmed, S. A. Shah, and S. U. Ali, "Facial recognition using convolutional neural networks and implementation on smart glasses," in *2019 International Conference on Information Science and Communication Technology (ICISCT)*. IEEE, 2019, pp. 1–6.

[11] S. Arya, H. Mesariya, and V. Parekh, "Smart attendance system usign cnn," *arXiv preprint arXiv:2004.14289*, 2020.

[12] E.-J. Cheng, K.-P. Chou, S. Rajora, B.-H. Jin, M. Tanveer, C.-T. Lin, K.-Y. Young, W.-C. Lin, and M. Prasad, "Deep sparse representation classifier for facial recognition and detection system," *Pattern Recognition Letters*, vol. 125, pp. 71–77, 2019.

[13] R. Fu, D. Wang, D. Li, and Z. Luo, "University classroom attendance based on deep learning," in *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*. IEEE, 2017, pp. 128–131.

[14] Z. Zhiyao and B. M. Di, "Multiplet selection: a practical study of multi-faces recognition for student attendance system," in *Proceedings of the 2nd International Conference on Image and Graphics Processing*, 2019, pp. 42–46.

[15] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, 2018, pp. 117–122.

[16] I. Masi, A. T. Tran, T. Hassner, G. Sahin, and G. Medioni, "Face-specific data augmentation for unconstrained face recognition," *International Journal of Computer Vision*, vol. 127, no. 6-7, pp. 642–667, 2019.

[17] J. Kolbusz, P. Rozycki, and B. M. Wilamowski, "The study of architecture mlp with linear neurons in order to eliminate the "vanishing gradient" problem," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 97–106.

[18] L. Yuan, Z. Qu, Y. Zhao, H. Zhang, and Q. Nian, "A convolutional neural network based on tensorflow for face recognition," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE, 2017, pp. 525–529.

[19] B. D. Eugenio and M. Glass, "The kappa statistic: A second look," *Computational linguistics*, vol. 30, no. 1, pp. 95–101, 2004.