ICIC Express Letters
Volume **16**, Number **7**, July **2022**

ICIC International ©2022 ISSN 1881-803X
pp. 779–786

FACE LIVENESS CLASSIFICATION USING MOBILENET AND SUPPORT VECTOR MACHINES

Badia Tahi Felix and Suharjito

Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
badia.felix@binus.ac.id; suharjito@binus.edu

Received June 2021; accepted September 2021

Abstract. *The facial recognition system is a biometric technology that has been widely applied in various applications. However, there are still many security weaknesses in facial recognition systems. The user's face can be manipulated using an object in the form of a face photo print or face photo displayed on a smartphone. There are several studies that have conducted research on this topic which aims to create a model that can classify real faces and spoof faces with the best level of accuracy. Various techniques have also been developed in this field to make a better classification. One popular technique that is often used today is deep learning with convolutional neural networks which has been widely used to be able to make predictions with a good level of accuracy. The objective of this study is to explore a classification model that utilizes a fine-tuning of the pre-trained MobileNet V2 as a feature extractor and support vector machines as binary classifier. This proposed model uses the NUAA spoofing database that gets better results than most state-of-the-art liveness classification with an accuracy of 99.72% and equal error rate of 0.0028.*

Keywords: Face liveness, Face anti-spoofing, MobileNet, Support vector machine

1. **Introduction.** The human body is created with different characteristics from one another. From this uniqueness, a technology that can recognize and analyze physical characteristics and human behavior is developed, called biometrics [1]. Face detection is one of the technologies in the biometrics field to identify a person's identity. Along with the times, the use of facial recognition technology especially for authentication security has been very widely used in various ways because with the use of these technologies it no longer needs the help of other properties such as cards or passwords that must be remembered so that it is easier and more practical [2].

Although there have been many advances in the last few decades in face recognition technology, there are still many weaknesses and they are vulnerable to attack [3]. Vulnerabilities that often occur are attacks carried out by disguising someone's identity by means of using two-dimensional face photos and also photos of faces that are displayed from the smartphone screen. Various types of attacks have succeeded in making deception so that facial recognition technology is not able to recognize between real faces and imitation faces.

In previous studies, there have been many significant developments in terms of detecting facial liveness using various methods such as texture based [4-7,19,20], and motion based [8-10] methods all of which aim to be able to recognize facial characteristics. One of the most popular ways to find features in facial images is the use of deep learning with a Convolutional Neural Network (CNN) [9-14]. In this study, MobileNet V2 is combined with the SVM classifier to perform liveness classification. The reason for using MobileNet

DOI: 10.24507/icicel.16.07.779

V2 in this study is because MobileNet V2 is a CNN architecture that provides high accuracy results while keeping the parameters and mathematical operations as low as possible so that it is suitable for use on small computers or mobile devices. The applications of SVM classifier on CNN are because SVM has a very good ability to perform binary classification by minimizing error on unseen data, especially with non-noisy data.

2. **Related Works.**

2.1. **Related research.** Disguising faces in order to trick people or not to be recognized by others is a practice that has been known for hundreds of years and even into modern times. Research to find out the characteristics of real faces and fake faces has also been carried out in this field with various theories and techniques. Various kinds of research on face liveness classification have been conducted and various techniques have been developed to go beyond the state of the art. In general, face liveness classification techniques are divided into several fields such as texture-based, motion-based and CNN-based methods.

Texture-based methods are techniques used to extract information contained in facial images using certain algorithms such as Local Binnary Pattern (LBP) [4,5,19,20], specular reflection ratio and channel distribution [6]. Chan et al. [4] used a method of taking face images with the use of a flash directed towards the subject's face. Every time an image is taken, there will be two images taken, one without flash and one with flash. To get the appropriate illumination values, classification is done using the Support Vector Machine (SVM) from the sample data obtained. To analyze the spectrum in face images, Local Binary Pattern (LBP) is also implemented. Luan et al. [6] conducted a liveness detection study by utilizing feature extractions such as specular reflection ratio, hue channel distribution and blurriness to distinguish genuine face images and spoof face images. Specular reflection ratio is used to determine the geometrical shape of the face object which in this study found that the original face image has a rough specular reflection. Hue channel distribution is used to find out the combination of colors and bluriness is used to find deeper feature information on facial images. To find patterns from facial image samples, Support Vector Machine (SVM) was applied through LibSVM Library.

For motion-based methods, it will usually detect unique parts of the face to distinguish between a real face and a fake face. This technique will usually look for differences by comparing movements such as head movements, blinking or mouth movements. There are several studies that focus on this method [7-10]. Siddiqui et al. [7] combined traces of image features such as texture, movement and also the background around the image. For extracting feature information use LBP and for detecting motion use a histogram. This combination results in resistance to different attacks. Singh and Arora [8] detected facial liveness using morphological operations techniques to analyze the specific movements of the face, the movements of the eyes and mouth. The data used to detect movement is video data that contains the movements of the face of an object where the object must make an opening and then close the mouth and then followed by opening the eyes and then closing the eyes.

There are several liveness classification studies that use CNN as a technique for classifying real and fake faces [9-14,21,23]. Alotaibi and Mahmood [9] classified spoof faces and real faces using CNN with a non-linear diffusion-based method. Li et al. [10] conducted a research on liveness by developing ideas by extracting deep partial features from CNN-VGGFace and using the principal component analysis method to reduce feature dimensions so that over-fitting conditions can be avoided, and then the final step is to use SVM. Seo and Chung [11] proposed Thermal Face-CNN which can distinguish between real faces and fake faces based on thermal differences obtained by the infrared camera, and then use CNN for the classification. Ge et al. [12] proposed a combination of the CNN-LSTM network by extracting discriminative features from video frames using the

CNN network, and then the results of feature extraction were used by LSTM to study the dynamic frames across the video clips. Larbi et al. [13] proposed a face-spoofing method based on CNN's multi color architecture called DeepColorFASD by looking for RGB, HSV and YCbCr coloring effects on the CNN network and also proposed a fusion based voting method. Zhu et al. [14] proposed Contour Enhanced Mask R-CNN (CEM-RCNN) frame by inserting object contour measurements into the R-CNN framework where the RPN and R-CNN heads are trained separately. Then Song et al. [21] proposed combination of SPMT + TFBD for face PAD and a decision-level cascade strategy. There are also studies that use CNN as a feature extractor and use SVM as a classifier [23].

2.2. **MobileNet V2.** Deep learning became famous during the ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) competition, namely the computer vision Olympiad held in 2012 at which time the architecture succeeded in classifying 1.2 million high-resolution images using GPU [15]. MobileNet is one of CNN's popular architectural models for image classification. What makes MobileNet special is the CNN architecture which can provide good accuracy while keeping the parameters and mathematical operations as low as possible [16]. This makes this architecture very suitable for application on mobile devices, embedded systems or computers with low specifications. MobileNet V2 is lighter, faster and more efficient than MobileNet V1 [22]. MobileNet V2 builds on the idea of MobileNet V1 which uses depth-wise separable convolutions with the addition of linear bottlenecks between the layers and a shortcut connection between the bottlenecks. In MobileNet V2, the parameters are reduced by up to 30 percent, resulting in a smaller computational cost but with excellent accuracy. The full MobileNet V2 architecture is shown in Table 1.

TABLE 1. MobileNet V2 architecture [22]

Input	Operator	t	c	n	s
224 × 224 × 3	Convolution 2D	–	32	1	2
112 × 112 × 32	bottleneck	1	16	1	1
112 × 112 × 16	bottleneck	6	24	2	2
56 × 56 × 24	bottleneck	6	32	3	2
28 × 28 × 32	bottleneck	6	64	4	2
14 × 14 × 64	bottleneck	6	96	3	1
14 × 14 × 96	bottleneck	6	160	3	2
7 × 7 × 160	bottleneck	6	320	1	1
7 × 7 × 320	Convolution 2D 1 × 1	–	1280	1	1
7 × 7 × 1280	Avg Pooling 7 × 7	–	–	1	–
1 × 1 × 1280	Convolution 2D 1 × 1	–	k	–	–

2.3. **SVM classifier.** Support Vector Machine (SVM) is a model that is intended to solve data classification problems [17]. SVM is good enough to solve both linear and non-linear classification problems. The way SVM works is to create hyperplane lines that can separate the distribution of data into different classes. Then the SVM algorithm will find the distance between the closest point of the two classes to the hyperplane line which is called a margin. After the maximum margin is obtained, an optimal hyperplane will be formed. If the data is a linear class, it will separate the data into two different classes.

3. Proposed Method.

3.1. **Dataset.** The dataset used is a public dataset, namely NUAA photograph imposter database [18]. The NUAA dataset consists of 12,614 face images extracted from real face videos and fake face videos. The video recorded using a webcam was captured over three different sessions under different lighting conditions. To take a fake face is to use a printed photo which is then recorded again via a webcam. The NUAA dataset is divided into two parts, namely the training set and the test set where for the training set there are 1732 real face images and 1748 fake face images and then for the test set there are 3362 real face images and also 5761 fake face images.

In Figure 1, it can be seen that real face photos are a mixture of photos of male and female faces taken using different lighting and there are faces that wear glasses and those that are not. For imposter face photos, it is a face photo that is printed on photo paper to be recorded again via a webcam as can be seen in Figure 2. The experiment followed the standard protocol from the dataset by measuring the accuracy of the facial image classification and also measuring the Equal Error Rate (EER). Each face image used in the dataset will be pre-processed, where each image size will be resized to the standard input size of the MobileNet architecture, which is 224 × 224.

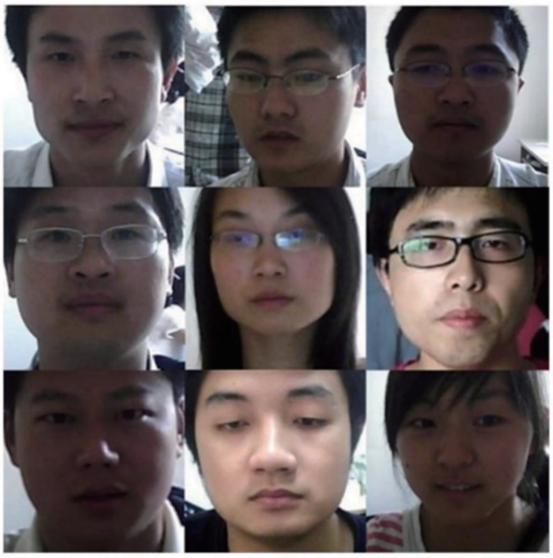


Figure 1. Sample images from real face

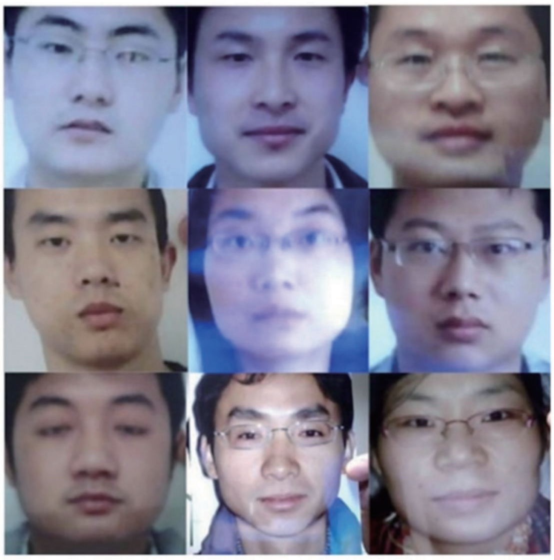


Figure 2. Sample images from fake face

3.2. **Deep learning architecture.** The method we propose is face liveness classification based on CNN MobileNet V2 and SVM linear classifier at the top layer. The MobileNet V2 model has the advantage of an efficient architecture with good results and SVMs are excellent linear classifiers for solving classification problems especially with non-noisy data. The overall architecture in this study can be seen in Figure 3.

The input is an RGB image of a face photo in the form of a real face photo or a spoof face photo. Each image input must be preprocessed in size to match the MobileNet input (224 × 224 × 3) and image normalization is also carried out so that the value is between 0 and 1. MobileNet V2 will carry out the feature extraction process and after it becomes fully connected layer then in the last layer the SVM classifier will do its job to classify each input image to be categorized into a spoof face class or real face class. This process will continue to repeat during the training process depending on how many repetitions are given.

In Figure 4, there are some changes made to the MobileNet V2 architecture to function as a feature extractor by replacing the existing top layer. After the top layer is removed, it is replaced with a new layer such as adding global_average_pooling2d then dropout layer to prevent a model from overfitting and dense layer 1 unit which is set to SVM classifiers. SVM classifier will replace the fully connected layer for performing binary

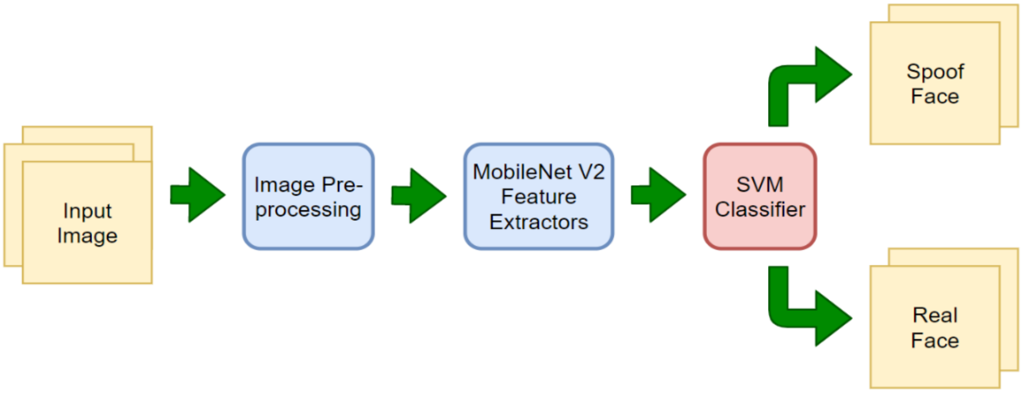


Figure 3. Architecture of MobileNet V2 + SVM classifier

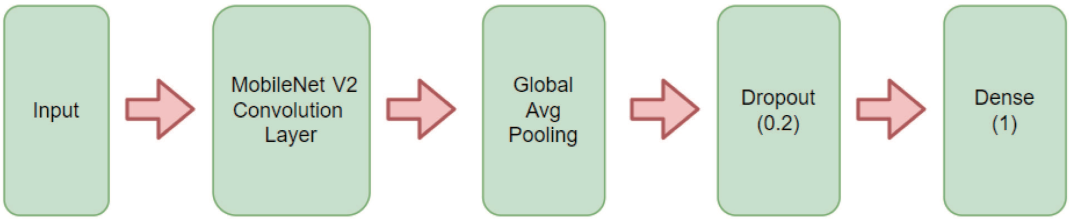


Figure 4. Changes to the top layer of MobileNet V2

classification. To activate the SVM classifier, there are two things that must be done, namely by implementing L2 regularizers with linear activation on the fully connected layer binary classification and also implementing the hinge lost function in the compile section. By making changes to these two things, it will activate the SVM classifier feature in the architectural model.

3.3. **Parameters settings.** There are several parameters that are used to achieve maximum training by setting the regularizer kernel, namely L2 regularizers with a value of 0.1 then for the optimizer using RMSprop with learning rates of 0.01, 0.001 and 0.0001. To apply SVM as a classifier, the hinge lost function is used. In this experiment, the experiment was carried out with 40 epoch. Then the checkpoint model is also applied so that in the end the model that is stored is the best model based on maximum accuracy monitoring.

3.4. **Evaluation metric.** To evaluate the performance of the MobileNet V2 and SVM classifier there are several parameters used. The parameters used are Accuracy, FAR, FRR and EER.

1) Accuracy

To be able to calculate accuracy, what must be done is to create a confusion matrix from the results of testing data. From the matrix, we will get True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Then we can find the accuracy by using Formula (1).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

2) False Acceptance Rate (FAR)

FAR is a measure of how likely it is that a system will receive access from unauthorized users. The formula for calculating FAR can be seen as Formula (2).

$$FAR = \frac{FP}{FP + TN} \tag{2}$$

3) False Rejection Rate (FRR)

FRR is a false recognition rate measurement that looks at how much a system erred in denying access from authorized users. The formula for calculating FRR can be seen as Formula (3).

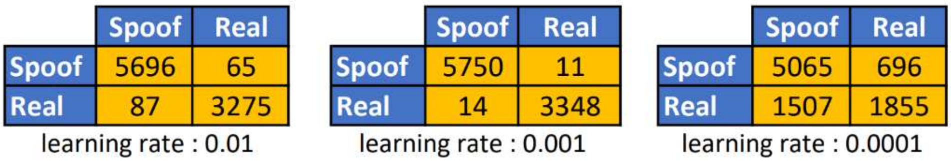
$$FRR = \frac{FN}{FN + TP} \tag{3}$$

4) Equal Error Rate (EER)

EER is a method used to define the threshold value when the error rate of FAR and FRR is the same or it can also be the meeting point of FAR and FRR. The best approaching EER with the smallest precision error can be found using Formula (4).

$$EER = \frac{FAR\mathcal{T} + FRR\mathcal{T}}{2} \tag{4}$$

4. **Experimental Results.** In this section, we will present the results obtained from the liveness classification experiment using the MobileNet V2 and SVM classifier on the NUAA dataset. Experiments were carried out using the tensorflow library and Keras with the Python programming language. The hardware used for training is a GPU card which has 4 GB of memory. Each training uses an epoch of 40 with a batch size of 16. Trainable layers are set true so the weight of each layer will be updated every epoch during training. From all the experimental results, it can be seen that some of the experiments got quite good results. From each experiment, the model is tested using test dataset and the test results are made into a confusion matrix as shown in Figure 5.



	Spoof	Real
Spoof	5696	65
Real	87	3275

learning rate : 0.01

	Spoof	Real
Spoof	5750	11
Real	14	3348

learning rate : 0.001

	Spoof	Real
Spoof	5065	696
Real	1507	1855

learning rate : 0.0001

Figure 5. Confusion matrix results

The results of each experiment are shown via confusion matrix in Figure 5. From the confusion matrix image, we can know the TP, FP, FN and TN values. From these values, we can calculate the performance parameters to find out how good the model of this experiment is.

Table 2 shows the results of the calculation of the performance model parameters from the three experiments conducted. After obtaining the TP, FP, FN and TN results, calculations are carried out to get Accuracy, FAR, FRR and EER according to the formulae previously described. FAR and FRR values are used to find the EER value of each learning rate. From the calculation results, it can be seen that a model with a learning rate of 0.001 gets the best results with an accuracy value of 99.72% and EER value of 0.0028.

Table 2. Experimental results

Learning rate	Accuracy	FAR	FRR	EER
0.01	98.33%	0.0194	0.0150	0.0172
0.001	**99.72%**	**0.0032**	**0.0024**	**0.0028**
0.0001	75.85%	0.2728	0.2293	0.2510

The results of this method are compared with the state of the art that has used the NUAA photograph imposter database, where the accuracy and EER values are compared. From the table, it can be seen that the proposed method with a learning rate of 0.001 produces good accuracy and also produces the best performance on the EER results which can be seen in Table 3.

Table 3. Performance comparison in NUAA dataset

Propose method	Accuracy	EER
Kernel Fusion [20]	–	1.8
n-LBPnet [19]	98.2%	0.018
SPMT + SSD [21]	99.16%	0.89
MobileNet V2 + SVM	**99.72%**	**0.0028**

5. Conclusions.

The purpose of this paper is to explore the liveness classification method by combining feature extraction techniques from the MobileNet V2 architecture and the SVM classifier, which produces better results compared to the state of the art that has used the NUAA photograph imposter database. The accuracy results obtained were 99.72% and the EER results were 0.0028. For future work, we intend to do tests with other databases to measure how accurate this proposed method is when applied to different datasets and also try MobileNet V3 as a feature extractor.

REFERENCES

[1] S. Chakraborty and D. Das, An overview of face liveness detection, *International Journal on Information Theory (IJIT)*, vol.3, p.2, 2014.

[2] L. Li, P. L. Correia and A. Hadid, Face recognition under spoofing attacks: Countermeasures and research directions, *IET Biometrics*, vol.7, no.1, pp.3-14, 2017.

[3] Y. A. U. Rehman, L. M. Po and M. Liu, LiveNet: Improving features generalization for face liveness detection using convolution neural networks, *Expert Systems with Applications*, vol.108, pp.159-169, 2018.

[4] P. P. Chan, W. Liu, D. Chen, D. S. Yeung, F. Zhang, X. Wang and C. C. Hsu, Face liveness detection using a flash against 2D spoofing attack, *IEEE Trans. Information Forensics and Security*, vol.13, no.2, pp.521-534, 2017.

[5] K. Grover and R. Mehra, Face spoofing detection using enhanced local binary pattern, *International Journal of Engineering and Advanced Technology (IJEAT)*, vol.9, no.2, 2019.

[6] X. Luan, H. Wang, W. Ou and L. Liu, Face liveness detection with recaptured feature extraction, *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp.429-432, 2017.

[7] T. A. Siddiqui, S. Bharadwaj, T. I. Dhamecha, A. Agarwal, M. Vatsa, R. Singh and N. Ratha, Face anti-spoofing with multifeature videolet aggregation, *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp.1035-1040, 2016.

[8] M. Singh and A. S. Arora, A robust anti-spoofing technique for face liveness detection with morphological operations, *Optik*, vol.139, pp.347-354, 2017.

[9] A. Alotaibi and A. Mahmood, Deep face liveness detection based on nonlinear diffusion using convolution neural network, *Signal Image and Video Processing*, vol.11, no.4, pp.713-720, 2017.

[10] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li and A. Hadid, An original face anti-spoofing approach using partial convolutional neural network, *2016 6th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp.1-6, 2016.

[11] J. Seo and I.-J. Chung, Face liveness detection using Thermal Face-CNN with external knowledge, *Symmetry*, vol.11, no.3, p.360, 2019.

[12] H. Ge, X. Tu, W. Ai, Y. Luo, Z. Ma and M. Xie, Face anti-spoofing by the enhancement of temporal motion, *2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, pp.106-111, 2020.

[13] K. Larbi, W. Ouarda, H. Drira, B. B. Amor and C. B. Amar, DeepColorFASD: Face anti spoofing solution using a multi channeled color spaces CNN, *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.4011-4016, 2018.

[14] X. Zhu, S. Li, X. Zhang, H. Li and A. C. Kot, Detection of spoofing medium contours for face anti-spoofing, *IEEE Trans. Circuits and Systems for Video Technology*, 2019.

[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, vol.25, pp.1097-1105, 2012.

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

[17] C. Cortes and V. Vapnik, Support-vector networks, *Machine Learning*, vol.20, no.3, pp.273-297, 1995.

[18] X. Tan, Y. Li, J. Liu and L. Jiang, Face liveness detection from a single image with sparse low rank bilinear discriminative model, *European Conference on Computer Vision*, 2010.

[19] G. B. De Souza, D. F. da Silva Santos, R. G. Pires, A. N. Marana and J. P. Papa, Deep texture features for robust face spoofing detection, *IEEE Trans. Circuits and Systems II: Express Briefs*, vol.64, no.12, pp.1397-1401, 2017.

[20] S. R. Arashloo, J. Kittler and W. Christmas, Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features, *IEEE Trans. Information Forensics and Security*, vol.10, no.11, pp.2396-2407, 2015.

[21] X. Song, X. Zhao, L. Fang and T. Lin, Discriminative representation combinations for accurate face spoofing detection, *Pattern Recognition*, vol.85, pp.220-231, 2019.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.4510-4520, 2018.

[23] H. Dyoniputri and Afiahayati, A hybrid convolutional neural network and support vector machine for dysarthria speech classification, *International Journal of Innovative Computing, Information and Control*, vol.17, no.1, pp.111-123, 2021.