International Journal of Innovative Computing, Information and Control
Volume **17**, Number **1**, February **2021**

ICIC International ©2021 ISSN 1349-4198
pp. 111–123

A HYBRID CONVOLUTIONAL NEURAL NETWORK AND SUPPORT VECTOR MACHINE FOR DYSARTHRIA SPEECH CLASSIFICATION

Hanifia Dyoniputri[1] and Afiahayati[2,*]

[1]Klikdokter
Special Capital Region of Jakarta 12430, Indonesia

[2]Department of Computer Science and Electronics
Universitas Gadjah Mada
FMIPA UGM Sekip Utara, Bulaksumur, Sleman, Special Region of Yogyakarta 55281, Indonesia
*Corresponding author: afia@ugm.ac.id

Received September 2020; revised December 2020

Abstract. *Dysarthria is a neurological disorder that hinders the sufferers to articulate speech properly. These days, Automatic Speech Recognition (ASR) is being researched and developed to help dysarthria sufferers communicate. One of the basic stages of building an ASR is the speech classification and prediction process. In this study, we introduce a CNN-SVM hybrid model to recognize a 10-digit number pronounced by persons with dysarthria. This hybrid model was built to improve the classification ability of a simple CNN architecture in predicting dysarthric speech. CNN is used to capture the unique spatial features from the audio. The features captured by the CNN are then classified by the SVM, as SVM is known for processing data with large features. We also compared our hybrid model with standard CNN. This study succeeded in proving that the hybrid model was better than CNN with softmax layer, with an average increase in accuracy of 7.5%.*

Keywords: Convolutional neural network, Dysarthria speech recognition, Support vector machine

1. **Introduction.** Dysarthria is one of neurological impairments that occurs because of Amyotrophic Lateral Sclerosis (ALS), Parkinson's Disease (PD), neurological trauma, brain damage and stroke [1,2]. People with dysarthria lose their ability to articulate properly, resulting in vocal sounds that are generally indistinguishable by the listeners [1]. One common approach to establishing better communication between non-dysarthric and dysarthric persons is through Automatic Speech Recognition (ASR). ASR allows computers to analyze dysarthric speech and convert it into a clearer and more understandable form.

Divers machine learning methods have been used to create an ASR, namely Hidden Markov Model (HMM) [3] and Convolutional Neural Network (CNN) [4]. CNN is one method that has been a success in various fields of works, specifically in spatially related problems and has been vastly developed over the world. Generally, CNN learns important features from spatial data by moving and updating kernels through 2 different axes. These features captured by the kernels are then used to distinguish different spatial objects one from another. Several studies [5,6] have used CNN to solve speech recognition problems. For example, [5] proposed CNN architecture for speech recognition and achieved the satisfying performance of CNN, and even more specific, [6] used CNN to recognize dysarthria speech. However, [7] proved that CNN performance could produce better results by using

DOI: 10.24507/ijicic.17.01.111

a Support Vector Machine (SVM) as a substitute for the softmax layer in image classification. [8] researched on dysarthric speech using HMM and SVM and proved that SVM has a better performance in dealing with major consonant deletions, which is similar to the case of dysarthric speech. SVM [9] is one of the machine learning methods that has been used in solving classification tasks. The SVM works by finding the optimal hyperplane that separates one class from another. SVM also provides a smarter way to deal with outliers by finding the optimal margins of each class. Hence, in this study, we aim to provide strong classification results for dysarthric speech using a combination of CNN and SVM.

In this study, we create a novel hybrid architecture of CNN and SVM to recognize 10 digits number pronounced by dysarthric persons. The digits dataset is derived from UA Speech Database from the University of Illinois [10]. Firstly, we transformed the data form from an audio file format into a 4-dimensional array so that it can be processed by the proposed model. Later, we trained the transformed data with CNN and fine-tuned its parameters. Then, we removed the softmax layer from the best performing CNN and fed the data again into the modified CNN model, resulting in the CNN giving feature maps of the data as results. We then trained and fine-tuned the SVM with the feature maps from the previous step. Both prediction results of the CNN and CNN-SVM hybrid model are then compared and analyzed. [10] showed that the speech error rate in the district is differed by age, gender, and the severity of the speaker's disease. We, therefore, considered building different isolated-word recognition focusing on 10 digits of numbers for each speaker. This research is an extended version of the authors' thesis [11].

This research has made contributions as follows.

1) We proposed a novel way on classifying dysarthric speech with CNN-SVM hybrid architecture. CNN is used to capture spatial and to extract features. The SVM is used as a robust high-dimensional classifier, replacing the CNN's softmax layer.
2) The hybrid CNN-SVM architecture performs better in classifying dysarthric speech compared to simple CNN, achieving an accuracy score of 94.29% (7.5% increase over simple CNN).

We arranged the next sections as follows: Section 2 discusses previous researches which are closely related to our research, Section 3 explains speech processing techniques used, Section 4 presents an architectural overview for the hybrid model, Section 5 elaborates about the conducted experiments, also provides the results and discussion, and Section 6 contains the conclusion of the conducted research.

2. **Related Work.** Various machine learning methods have been applied to recognizing dysarthric speech. [12] conducted a study focusing on persons with cerebral palsy, which is categorized as mild dysarthric. [12] used a noisy-channel model to mimic the nature of the distortion in the utterance. [12] observed 3 different speakers and showed that the differential entropy of acoustics and articulation of each speaker may significantly differ one to others. Hence, we train our model in a speaker-dependent manner, after seeing how the entropies of acoustics and articulation differ from one speaker to another.

[13] developed a software named STARDUST based on Hidden Markov Model (HMM) that can improve the likelihood rate of dysarthric speech and normal speech. The study built a high-tolerance speech recognition system that can be used by either both normal or dysarthric speakers. This study proved that isolated word recognition produced high accuracy output compared to continuous recognition. From this research, we decided to conduct our research using the isolated-word-recognition approach. However, we chose CNN-SVM over HMM because we want our model to capture the high-level correlation of the data. According to [8], Support Vector Machine (SVM) is more powerful for

speech recognition with major consonant deletion compared to HMM, which in this case is dysarthric speech. However, it is also stated in [8] that the SVM failed to recognize slow dysarthric speech. Therefore, we used Convolutional Neural Network (CNN) in our study because CNN is able to capture both short and long-range features which in this case is the speed of utterance.

Another approach in recognizing dysarthric speech is by using CNN as conducted by [5,6,14]. [14] proposed a convolutional neural network architecture for Arabic letter speech recognition. [14] stated that the similarity of sound produced between a few letters. CNN managed to recognize similar speech and handled the locality problem. [14] inspired us in terms of the data input, as we also utilized static, delta, and double-delta features as the input. Our work is closely related to [6], which used regular CNN to recognize dysarthric speech, resulting in an average accuracy of 90.43%. [6] also used speaker-dependent and isolated-word approach, which we used in this research. We believe there is still a possibility to improve the performance of the CNN by combining it with SVM, as [7] removed the softmax layer of CNN and replaced it with SVM and achieved a better result in the case of image classification. In another case, [15] combined deep belief network with SVM, which also achieved higher accuracy compared to other machine learning methods. These studies inspired us to combine CNN and SVM. The CNN has been proved on helping the pre-trained process, which will improve the likelihood of data to expected classes. The SVM is then applied for robust classification to high dimensional data.

3. **Speech Preprocessing.**

It has been proven that CNN leads to high accuracy in predicting image because of its effectiveness in extracting features. [5,6,14] showed that there is a possibility to use CNN to recognize audios instead of images. The audio data need to be preprocessed before being processed by CNN. Thus, several steps should be conducted as follows: noise reduction and audio cutting and sampling, framing, and audio feature extraction.

3.1. **Cutting and sampling.**

We used Audacity Open Source software to reduce the noises and cut the audio. We set 2 seconds as the length of cutting value because the speaker with low speech intelligibility has slower utterance. Therefore, other speakers with better utterance needed to be adjusted since the length of data needs to be consistent when using CNN. Afterwards, we applied the sampling step, which is conducted by capturing the amplitude of the audio at a certain time. This step is conducted in order to represent the audio in the form of vector. A sample rate of 16000 Hz was used so that the length of a vector will be 32063 samples, where every sample represents amplitude in a period of time, as illustrated in Figure 1.

3.2. **Framing.**

The framing step divided the audio samples into some frames with certain window size. In this study, a window size of 25 ms was applied along with frameshift value of 10 ms, as suggested by [5]. This step produced 199 frames in total, where each frame contains samples represented by the amplitude.

3.3. **Feature extraction.**

Lattermost, we extracted a few audio features from previous steps, namely Static Coefficient and Dynamic Coefficient. We used the Mel Frequency Spectral Coefficient (MFSC) to obtain Static Coefficient. MFSC is a representation of human hearing, acquired by extracting the energy of Mel frequency in each frame. Therefore, to achieve the Static Coefficient, the original frequency in each frame has to be converted into Mel frequency through several steps as shown by Figure 2. Several steps have to be conducted to produce the Static Coefficient. First, the audio signal that represented time domain should represent the frequency domain by calculating Discrete

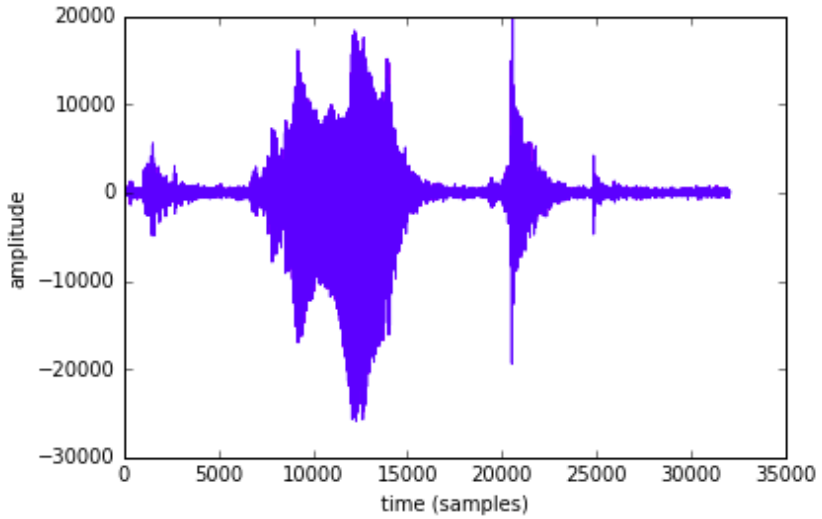

FIGURE 1. Sampled audio signal

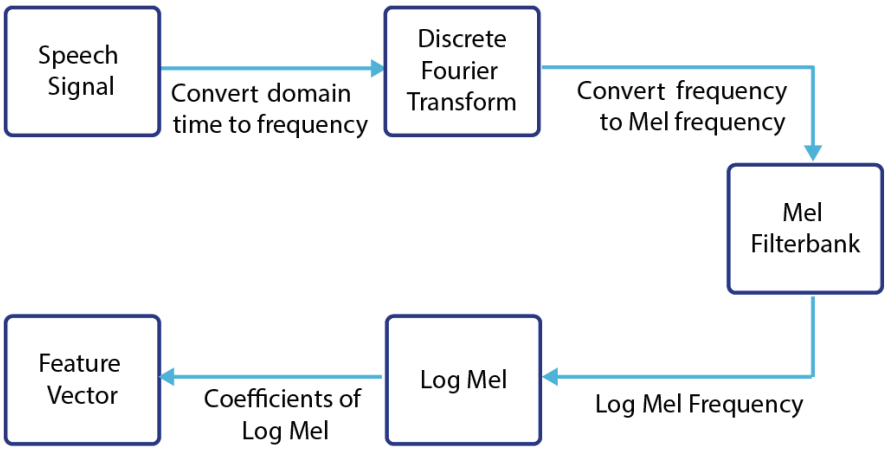

FIGURE 2. MFSC flow diagram

Fourier Transform (DFT) using the Fast Fourier Transform (FFT) algorithm. The result of DFT is a spectrum. Equation (1) describes how to calculate DFT,

$$X_a[k] = \sum_{n=1}^{N-1} x[n] e^{-j2\pi nk/N}, \ 0 \leq k < N \tag{1}$$

where $x[k]$ is samples, a is the index of the frame, k is the index of the coefficient spectrum, N is the number of samples in the period of time, and $e^{-j2\pi nk/N}$ is a periodic function. Afterwards, amount of m of Mel Filterbank is applied to the result of DFT, resulting in spectrum in a frequency of Mel. The number of Filterbank k may vary, but [16] suggested around 25-40 filters as a best practice. Filter m is a triangular filter and described by Equation (2),

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{2}$$

where H_m is the Mel Filterbank, k is the number of filters, and $f(m)$ is the frame in Mel frequency. Finally, taking the logarithm of each Mel Filterbank will result in some coefficient that forms the feature vector as the output of MFSC. In this research, the number of Filterbanks was set to 26, as suggested by [5]. Those filters were applied in each frame, shaping the data into a 2-dimensional array with a size of 199 × 26. 199 represents the number of frames (time-domain) and 26 represents a number of the energy of Mel frequency (frequency domain) as illustrated in Figure 3.

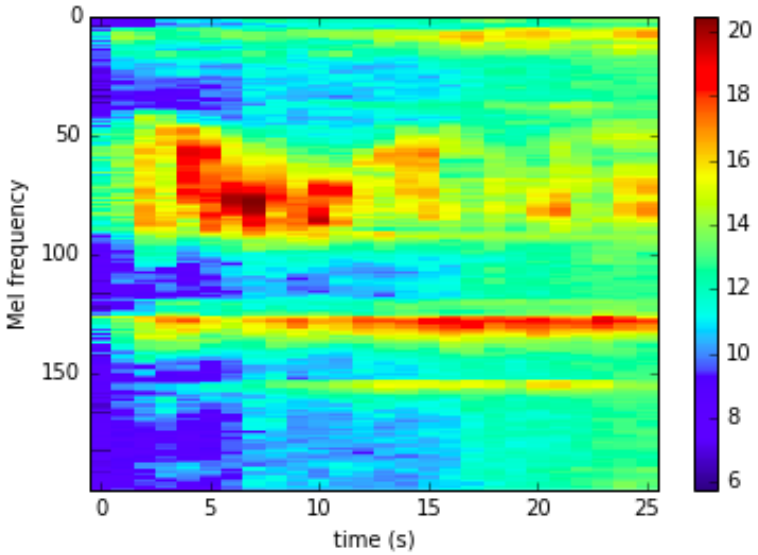

FIGURE 3. Spectogram of MFSC

The speech signal is not constant from frame to frames. The speech signal that is converted into frames will be assumed stationary in a short interval. Although the signal is extracted in frames, there must be correlations among frames. To capture this correlation, Dynamic Coefficient should be applied along with Static Coefficient (MFSC). Dynamic Coefficients can be computed by simply differencing between the feature values for two frames either side of the current frame [17]. A dynamic feature that consists of the first-order time derivatives is called Delta Coefficient, while a dynamic feature that is referred to as the second-order time derivatives is called Acceleration Coefficient or Double Delta Coefficient [18]. Delta Coefficient is used to measure the velocity of speech and earned by taking the derivative of the static data [5] as shown by Equation (3).

$$d(t) = \frac{c(t+1) - c(t-1)}{2} \tag{3}$$

where $d(t)$ is delta value of time t, and $c(t)$ is the cepstrum of time t. Aside, Double Delta Coefficient can be obtained by derivating the formula of Delta Coefficient. In other words, Delta Coefficient is the derivative of Static and Double Delta Coefficient is the derivative of Delta. Thus, each file will possess 3 channels: Static (original coefficient from MFSC), Delta Coefficient, and Double Delta Coefficient. In conclusion, the data were transformed into a 3-dimensional feature vector with a size of 199 × 26 × 3, as illustrated in Figure 4.

4. **Hybrid Architecture.**

4.1. **Convolutional neural network.**

Convolutional Neural Network (CNN) [4] is a neural network that is commonly used to capture patterns or features from spatial data [19]. Aside from capturing patterns, CNN is also good when handling noisy data because

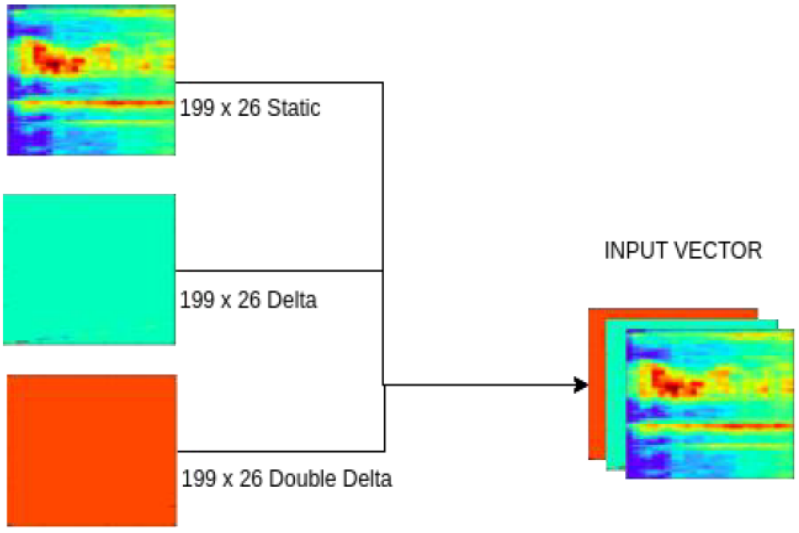

FIGURE 4. Features derived from MFSC with 3 channels: Static, Delta and Double Delta

it has a locality concept, which will detect and tolerate differences [14]. Equation (4) shows how the convolutional layer's filters move through 2-dimensional data [4].

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{4}$$

4.2. **Support vector machine.** Support Vector Machine (SVM) [9] is one of the supervised learning methods that is widely used for classification tasks in various works, and many proved about the goods of its performance. SVM works by finding the optimal separating hyperplane that maximizes the margin between the hyperplane with its support vectors. 3 parameters need to be considered in SVM to determine the best separating hyperplane, which are: kernel type, the value of C parameter, and value of γ parameter. The kernel type determined the type of separating hyperplane used. Several types of kernels are often used, namely Linear kernel, Polynomial kernel, and RBF/Gaussian kernel. The value of the C parameter controls the trade of balance between different classes. The value of γ parameter in RBF kernel (Equation (5)) determines the curvature of the decision boundaries. These three parameters have to be combined and fine-tuned properly to avoid overfitting and to get the optimal output [20].

$$K(X, X_i) = \exp\left(-\gamma ||X - X_i||^2\right) \tag{5}$$

4.3. **Hybrid CNN-SVM.** In this study, we proposed a hybrid architecture of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) to classify dysarthric speech. We used CNN as a feature extractor of each data before it is applied to the classifier (SVM). We trained the CNN using the preprocessed dataset from the previous section. In this study, the size of the input (received data derived from MFSC process) is $199 \times 26 \times 3$ (height, width and depth respectively). The 3 depths were adopted from image recognition, where 3 represents RGB colours representing 3 channels of features (Static, Delta, and Double Delta in our case). Then, every 3D vector of $199 \times 26 \times 3$ was convolved and filtered using 15 filters sized $5 \times 5 \times 3$ with zero paddings and Stride 1, extracting important features in the process. Pooling layer was then applied to capturing significant information while reducing the resolution of the data. In this study, we used max-pooling type of pooling layer with a size of 2×2. Lattermost, a fully-connected layer with softmax activation function was used to classify the data.

We then fine-tuned the CNN, aiming to get the best combination of parameters that result in the best accuracy results. Since CNN has many hyperparameters, whereas the goal of this step is to find the best hyperparameters, tuning all parameters would consume lots of time. Therefore, we decide to keep the value of a few parameters (Table 1). [21] suggested 0.01 initial learning rate with a learning decay of 0.95. This means that the learning rate will be decreased by 5% every mini-batches. The purpose is to optimize model accuracy since the higher learning rate will be faster on updating parameters but may often lead to overfitting. In contrast, the lower learning rate will slow the updates, but the advantage is that it can slowly approach the convergent point without missing it. Convolution stride was set as 1 and the padding was set as 0. It means that the convolutional filters will start at the edge of input size without any padding and move with a shift of 1. [21] recommended these parameters value because larger padding and stride value will cause the convolutional filters to miss a detail of the inputs. [21] also suggested using pooling size and pooling stride with a size of 2×2, as the larger size will make the layer too lossy and leads to a bad result.

TABLE 1. CNN's fixed parameters

Parameter	Value
Learning Rate	0.01
Learning Decay	0.95
Convolution Stride	1
Padding Size	0
Pooling Size	2

In contrast to [5] which used softmax as a classifier, this study intends to replace the softmax layer with SVM as illustrated in Figure 5. In this study, we used the SVM to replace the softmax layer in CNN, as [7] concluded that CNN with SVM classifier is most likely to produce higher accuracy score compared to softmax classifier on CNN. The SVM was then used to classify each class using the feature maps produced by the CNN model. We used a grid-search technique to obtain the best combination of C and γ parameters to produce the model with the highest accuracy. Figure 5 describes the non-hybrid (CNN) architecture and hybrid architecture of CNN and SVM used in this study.

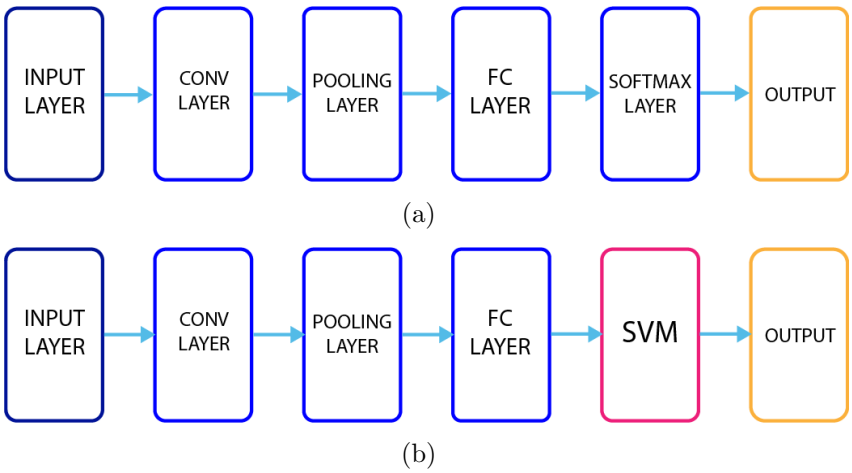

FIGURE 5. Architecture: (a) CNN and (b) hybrid CNN-SVM

Figure 6 illustrated the main hybrid architecture proposed in this study. For example, if kernel size of 12 × 12, feature map of 15, and 128 hidden units are applied into data with a resolution of 199 × 26 × 3, ones will receive an output with a resolution of 188 × 15 × 15. That 3-dimensional array will then be downsampled inside max-pooling layer, resulting in a data with a resolution of 94 × 8 × 15. Afterwards, the output of the max-pooling layer will be flattened and fed as an input for the fully-connected layer. All neurons will be connected to each hidden unit inside the fully-connected layer. Hence, the shape will be 11280 × 1. Finally, the output of fully-connected layer will be used as input of the SVM, which then will classify 10 digit classes.

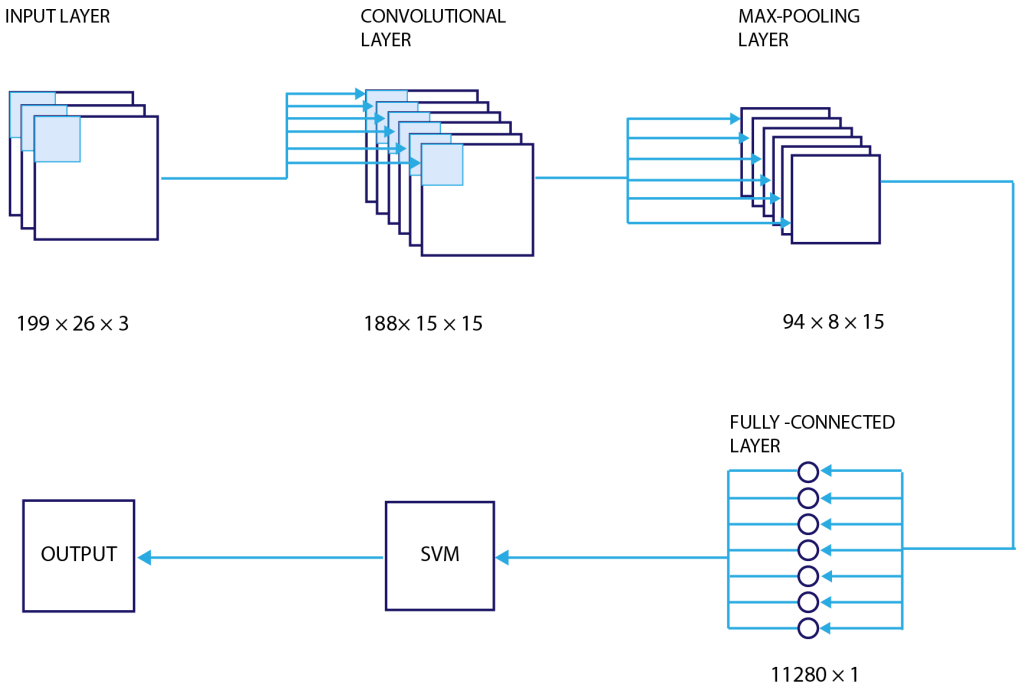

FIGURE 6. Detailed hybrid CNN-SVM architecture

5. **Experimental Results and Discussion.**

5.1. **Dataset.** In this study, we used Dysarthric Speech dataset derived from UA Speech Database from the University of Illinois [10]. This database consists of several isolated words with dysarthric subjects that vary from genders, ages, and dysarthric severity. There are 19 speakers in total, 5 of them are female and the rest 14 are male. The speech materials consist of 765 isolated words including 10 digits words, computer control words, common words and 26 alphabets, with 3 repetitions for each word. This study is limited on 10-digit words from 0 to 9, and those are uttered by 4 chosen speakers, consisting of male and female who had low or middle speech intelligibility as listed in Table 2. These data were recorded 3 times for every digit using 7 different microphones. Therefore, the total of data produced by each speaker would be 210 data (10 digits × 3 repetitions × 7 microphones). The data for each speaker were divided into training and testing data with the proportion of 2:1. Therefore, 140 training data and 70 testing data were derived from each speaker. The 70 data were chosen from the third repetition from 7 different microphones and were kept for a prediction after the models had trained using the other 140 data.

TABLE 2. UA speech dysarthric speakers

Speaker Code	Gender	Age	Speech Intelligibility
F02	Female	30	Low (29%)
F04	Female	18	Middle (62%)
M05	Male	21	Middle (58%)
M07	Male	58	Low (28%)

5.2. **CNN parameters.** We conducted several steps to classify dysarthric speech: finding the best combination of CNN's parameters, finding the best pair of SVM parameters, and performance evaluation of the hybrid model. We examined several experiments to find the best combination of CNN's parameters, namely kernel size experiment, feature maps experiment, hidden units experiment, and epoch experiment. We experimented with various types of kernel sizes such as 8×8, 10×10, 12×12, and 12×8. This kernel experiment was conducted using 15 feature maps, 128 hidden units, and 100 epochs. From Table 3, it can be seen that kernel size 12×8 produced higher average accuracy at 51% compared to kernel size 8×8 at 49%, kernel size 10×10 at 48% and kernel size 12×12 at 50%.

TABLE 3. Average accuracy of kernel experiment

Speaker Code	8×8	10×10	12×12	12×8
F02	59%	53%	58%	62%
F04	38%	42%	40%	38%
M05	44%	39%	42%	48%
M07	53%	58%	60%	55%
Average	**49%**	**48%**	**50%**	**51%**

We then searched for the best number of feature maps to be applied in CNN. The number of feature maps was set to 15, 20, and 25. This feature maps experiment was conducted using 8×8 kernel size, 128 hidden units, and 100 epochs. Table 4 listed the comparison of speaker accuracy by feature maps. By using 25 feature maps, the average accuracy reached 52%, higher than using 15 feature maps (49%) and 20 feature maps (48%).

TABLE 4. Average accuracy of feature maps experiment

Speaker Code	15	20	25
F02	59%	58%	59%
F04	38%	37%	40%
M05	44%	46%	48%
M07	53%	52%	61%
Average	**49%**	**48%**	**52%**

Afterwards, we experimented aimed to obtain the best number of hidden units of the fully-connected layer. The number of hidden units was set to 128, 256, and 512, and the experiment was run using 8×8 kernel size, 15 feature maps, and 100 epoch. As tabulated in Table 5, the highest average accuracy was achieved by using 512 hidden units, reaching 67% of average accuracy. It is scored higher compared to using 128 and 256 hidden units.

TABLE 5. Average accuracy of hidden units experiment

Speaker Code	128	256	512
F02	59%	72%	56%
F04	38%	45%	62%
M05	44%	65%	73%
M07	53%	67%	75%
Average	**49%**	**62%**	**67%**

To find the optimal number of the epoch, we experimented with several epoch values (100, 200, and 300). Again, this experiment was run using 8×8 kernel size, 15 feature maps and 128 hidden units. Table 6 described the average accuracy of epoch experiments by each speaker. It can be seen that 300 epochs produced the highest average of accuracy, achieving 69% compared to 100 epochs at 49% and 200 epochs at 66%.

TABLE 6. Average accuracy of epoch experiment

Speaker Code	100	200	300
F02	59%	74%	72%
F04	38%	50%	62%
M05	44%	65%	65%
M07	53%	75%	76%
Average	**49%**	**66%**	**69%**

Thus, from all experiments we conducted before, we set parameters value for the CNN as follows: kernel size of 12×8, feature maps of 25, hidden units of 512 and epoch of 300. Table 7 tabulated the average accuracy of CNN with softmax trained with these parameters. The model achieved an average of accuracy is 86.79%, with average total words predicted is 60.75 out of 70. This model spent 18 minutes and 33 seconds on average. However, the loss rate is still considered high, with 1.265 on average.

TABLE 7. Softmax CNN's performance measures

Speaker Code	Time (minutes)	Loss Rate	Accuracy	True Positive
F02	18:36	1.73	87.14%	61
F04	18:18	1.07	81.43%	57
M05	18:30	1.2	85.71%	60
M07	18:48	1.06	92.86%	65
Average	**18:33**	**1.265**	**86.79%**	**60.75**

5.3. **SVM parameters.** We then modified the CNN model by removing the softmax layer. We fed the dataset into the modified model, resulting in the model producing feature maps of the data as results instead of orthogonal classes. The feature maps were then trained using SVM. We searched for the best pair of C and γ parameters for the SVM using a grid search method. [20] suggested starting tuning C and γ parameters with a multiplication of it. Hence, in this research, the powers of 10 were chosen to initialize the C and γ parameters. Table 8 shows the values of C and γ parameters used in a grid search. With the method, we obtained the best pair of C and γ parameters for all speakers, which are 10 and 0.001 respectively. Table 9 shows the average accuracy of the best hybrid CNN-SVM model. The model achieved an average accuracy of 94.29%, predicting correct classes 66 out of 70 on average.

TABLE 8. C and γ parameter values for grid search

C	{0.001, 0.01, 1, 10, 100, 1000}
γ	{0.001, 0.01, 0.1, 1}

TABLE 9. Hybrid CNN-SVM's performance measures

Speaker Code	C	γ	Accuracy	True Positive
F02	10	0.001	92.86%	65
F04	10	0.001	98.57%	69
M05	10	0.001	85.71%	60
M07	10	0.001	100%	70
Average			**94.29%**	**66**

5.4. **CNN and hybrid CNN-SVM performance comparison.** Table 7 and Table 9 show that CNN with softmax achieved the classification accuracy of 86.79% with the average of 60.75 out of 70 in predicting correct classes, while CNN with SVM achieved 94.29% of average classification accuracy with the average of predicted words 66 out of 70. Hence, a hybrid of CNN and SVM produced an average accuracy which was 7.5% higher than CNN with softmax.

Looking deeper into classification accuracy for each speaker, speaker F02 achieved 87.14% for accuracy using CNN with softmax. Speaker F02 achieved 92.86% on hybrid CNN-SVM, with an improvement of around 5.72%. Speaker F02 predicted 61 words correctly in CNN with softmax and 65 in hybrid CNN-SVM. Speaker F04 achieved 81.43% of accuracy when using CNN with softmax, with a total of correctly predicted words reached 57 out of 70. However, the hybrid CNN-SVM achieved 98.57% for accuracy by correctly predicted 69 out of 70. The improvement was 17.14%, which is the highest improvement among all speakers. However, Speaker M05 achieved an accuracy score of 85.71% for both CNN and CNN-SVM model, which means that the hybrid CNN-SVM model is not always improving the predicted words.

Speaker M07 achieved 92.86% of accuracy by using CNN with softmax and 100% by using the hybrid model (7.14% of improvement). Speaker M07 (male with low speech intelligibility) achieved the highest classification accuracy in both models. This fact contradicts with [12], which said that speech intelligibility affects classification accuracy. However, our study is trained using a speaker-dependent method, while [12] was not. It can be concluded that using a speaker-dependent training method, speech intelligibility will not affect the classification accuracy. Figure 7 shows a bar chart comparing the CNN and CNN-SVM.

5.5. **Comparison with previous research.** We compared our hybrid CNN-SVM result with [6], as it used the same digit dataset and implemented the isolated-word speaker-dependent approach. [6] achieved an average accuracy of 90.43%, which is 3.64% higher than our simple CNN (86.79%). However, our hybrid CNN-SVM not only managed to increase our simple CNN average accuracy, but also best previous research average accuracy by 3.86%. We believe that our model managed to capture the spatial relationship of the data although it is not as good as [6]. Surprisingly, the average accuracy significantly increased when the feature maps from the CNN were classified by the SVM, outperforming both our simple CNN and previous research CNN [6] (Figure 8). This most likely happens because SVM is very robust in processing high dimensional features better than the softmax layer (in line with [7,15]).

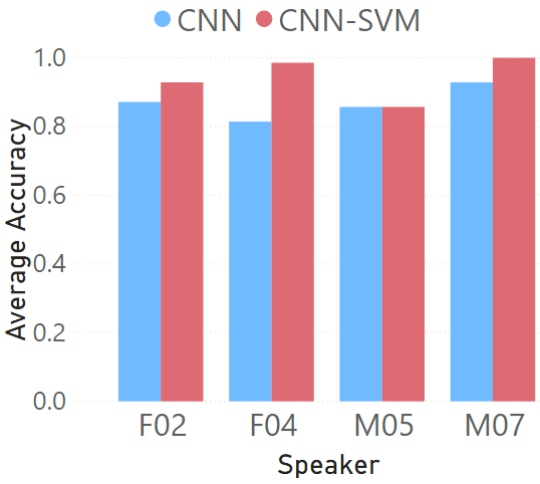

FIGURE 7. Average accuracy comparison by speaker

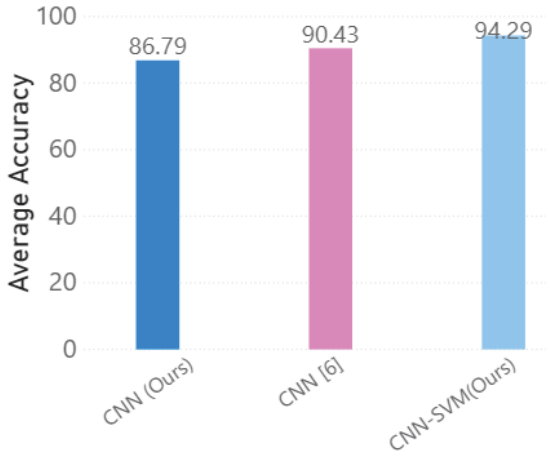

FIGURE 8. Average accuracy comparison to previous research

6. **Conclusions.** In this study, we have introduced a novel way of recognizing digits spoken by dysarthric speakers with CNN-SVM hybrid architecture. We showed that using CNN with SVM as a classifier achieved a better average score of classification accuracy (94.29%), scoring 7.5% higher than simple CNN and 3.64% higher than previous research's CNN. We also showed that the result may vary between each speaker and the respective digit. Thus, we recommend doing the speaker-dependent and isolated word to classify dysarthric speech in the future.

REFERENCES

[1] C. Bhat, B. Vachhani and S. K. Kopparapu, Automatic assessment of dysarthria severity level using audio descriptors, *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.5070-5074, 2017.

[2] M. V. Mujumdar and R. F. Kubichek, Design of a dysarthria classifier using global statistics of speech features, *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.582-585, 2010.

[3] L. Rabiner and B. Juang, An introduction to hidden Markov models, *IEEE ASSP Magazine*, vol.3, no.1, pp.4-16, 1986.

[4] Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, https://www.deeplearning-book.org/, 2016.

[5] O. Abdel-Hamid et al., Convolutional neural networks for speech recognition, *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol.22, no.10, pp.1533-1545, 2014.
[6] M. Dwiastuti and Afiahayati, Speech recognition for people with dysarthria using convolutional neural network, *ICIC Express Letters, Part B: Applications*, vol.10, no.9, pp.849-858, 2019.
[7] Y. Tang, *Deep Learning Using Linear Support Vector Machines*, arXiv:1306.0239 [cs.LG], 2013.
[8] M. Hasegawa-Johnson et al., HMM-based and SVM-based recognition of the speech of talkers with spastic dysarthria, *2006 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings*, vol.3, pp.III-III, 2006.
[9] C. Cortes and V. Vapnik, Support-vector networks, *Mach. Learn.*, vol.20, pp.273-297, 1995.
[10] H. Kim et al., Dysarthric speech database for universal access research, *Proc. of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp.1741-1744, 2008.
[11] H. Dyoniputri, *Dysarthric Speech Classification Using Convolutional Neural Network and Support Vector Machine*, Bachelor Thesis, Universitas Gadjah Mada, Indonesia, 2017.
[12] F. Rudzicz, Towards a noisy-channel model of dysarthria in speech recognition, *SLPAT'10: Proc. of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*, pp.80-88, 2010.
[13] M. Parker et al., Automatic speech recognition and training for severely dysarthric users of assistive technology: The STARDUST project, *Clinical Linguistics & Phonetics*, vol.20, pp.149-156, 2006.
[14] R. A. Rajagede, C. K. Dewa and Afiahayati, Recognizing Arabic letter utterance using convolutional neural network, *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp.181-186, 2017.
[15] Z. Wang, Q. Zheng and M. Lv, Modeling the external truck arrivals in container terminals based on DBN and SVM, *ICIC Express Letters*, vol.12, no.10, pp.1033-1040, 2018.
[16] X. Huang, A. Acero, H. Hon and R. Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall PTR, New Jersey, 2001.
[17] J. Holmes and W. Holmes, *Speech Synthesis and Recognition*, CRC Press, London, 2001.
[18] N. A. Meseguer, *Speech Analysis for Automatic Speech Recognition*, Department of Electronics and Telecommunications, Norwegian University of Science and Technology, 2009.
[19] J. Kurniawan et al., Traffic congestion detection: Learning from CCTV monitoring images using convolutional neural network, *Procedia Computer Science*, vol.144, pp.291-297, 2018.
[20] C. W. Hsu, C. C. Chang and C. J. Lin, *A Practical Guide to Support Vector Classification*, Tech. Rep., Department of Computer Science, National Taiwan University, 2003.
[21] A. Karpathy, Cs231n: Convolutional neural networks for visual recognition, *Neural Networks*, vol.1, no.1, 2016.