Using deep neural networks in classifying electromyography signals for hand gestures

Ahmed M. Al-Khazzar, Zainab Altaweel, Jabbar S. Hussain
Prosthetics and Orthotics Engineering Department, College of Engineering, University of Kerbala, Kerbala, Iraq

Article Info

Article history:

Received Jan 19, 2023
Revised May 8, 2023
Accepted May 23, 2023

Keywords:

Artificial intelligence
Deep neural networks
Electromyography
Neural networks
Smart prostheses

ABSTRACT

Electromyography (EMG) signals are used for various applications, especially in smart prostheses. Recognizing various gestures (hand movements) in EMG systems introduces challenges. These challenges include the noise effect on EMG signals and the difficulty in identifying the exact movement from the collected EMG data amongst others. In this paper, three neural network models are trained using an open EMG dataset to classify and recognize seven different gestures based on the collected EMG data. The three implemented models are: a four-layer deep neural network (DNN), an eight-layer DNN, and a five-layer convolutional neural network (CNN). In addition, five optimizers are tested for each model, namely Adam, Adamax, Nadam, Adagrad, and AdaDelta. It has been found that four layers achieve respectable recognition accuracy of 95% in the proposed model.

Corresponding Author:

Ahmed M. Al-Khazzar
Prosthetics and Orthotics Engineering Department, College of Engineering, University of Kerbala
Kerbala, Iraq
Email: ahmed.m.ahmed@uokerbala.edu.iq

1. INTRODUCTION

Electromyography (EMG) signals can effectively characterize the properties of body neurons and muscles of the human body [1]. They are widely used in prosthetic applications as a method to control the actuators in lower or upper prostheses. Many of these applications use noninvasive surface EMG sensors to measure the muscle activity and control the prosthetic device based on the collected data. Some examples can be found in [2]–[10].

Developing a controller to recognize various hand gestures based on EMG signals is a difficult task. The accuracy of such systems is reported in previous work [11]. The EMG accuracy is affected by various factors such as the location of EMG electrode, the extracted features, and the recognition algorithm [12]. In surface EMG (sEMG) specially, the electrode location greatly affects the signal quality [13].

In addition to the accuracy and signal quality challenges, differentiating between individual finger movements presents further difficulties. The reason is that the EMG signal variances are smaller for finger movements compared to other muscles [14]. Additionally, muscles that control finger gestures are in deep layers of the forearm, making it more difficult to measure the EMG signal [15]. To overcome some of these challenges, an efficient EMG recognition algorithm is required.

The amputee usually is required to undergo training sessions to learn to control the prosthetic device. In the training process, the system needs to collect EMG data related to the amputee and recognize the intent of the amputee and make decisions later to enable the patient to control the prosthetic device. In using EMG based devices, the problem arises that the training should be repeated for each patient. Some patients are reported to drop the EMG based prosthetic device as a result of difficulty in controlling the prosthesis. These difficulties could be originated from the limited number of training sessions and the inherent EMG signal properties. For example, electrical signals from muscles can be affected by interference from power supply, mechanical devices, and activity of other muscles [16].

To overcome some of the difficulties mentioned previously in this section, it is suggested to train neural network models *i.e* deep neural network (DNN) or convolutional neural network (CNN) with different parameters for EMG gesture recognition. In this paper, three models are proposed and trained using an open dataset and the recognition accuracy is reported. The dataset was collected using seven hand gestures. The proposed system should be able to recognize each gesture with acceptable accuracy based on the available EMG data.

Table 1 shows some of the previous literature's recognition accuracies in EMG data classification [10], [12], [14], [17], [18], [19]–[28] using artificial neural networks (ANN) and other classifiers. The number of EMG channels (sensors), classifier types, number of subjects and gestures, and the length of time varies depending on the research. However, the table shows that the recognition accuracy can vary between 64% to 99% depending on the methodology. The results of this paper will be compared with the related research in this table.

Table 1. Comparison of past literature in EMG data classification

Reference	No. of EMG Channels	Classifier	No. of Subjects	Time Length (seconds)	No. of Gestures	Recognition Accuracy
[10]	8	Multiple layer perceptron neural network (MLPNN)	3	7	14	91-94%
[12]	6	Principal component analysis (PCA) algorithm and support vector machine (SVM) classifier	5	5	6 11 17	99.6% 95.6% 95.1%
[14]	6	Artificial neural network (ANN)	12	5	9 17	72.9% 63.8%
[17]	4	Independent component analysis (ICA), Integral root mean square (IRMS), ANN	4	10		90.33%
[18]	4	K-nearest neighbor (KNN), linear discriminant analysis (LDA), SVM	6	0.064	9	91%
[19]	5	Performance measurement index (PNM)	4	1	10	80%
[20]	8	Deep adaptation network (DAN), multiple kernel variant of maximum mean discrepancies (MK-MMD)	23	3-10	22	84.6%
[21]	2	Linear bayesian classifier	4	1	5 11 16	90%-93.5% 83.1%-95.4% 78.8%-90.3%
[22]	4	Canonical correlation analysis (CCA), KNN, LDA	8	5-10	8	82%
[23]	64	Hyperdimensional (HD) computing	5	2	9	78.21%
[24]	2	KNN	30	5	4	94%
[25]	4	1-nearest neighborhood, maximum likelihood estimation (MLE)	8	1-2	8	85.7%
[26]	4	SVM, LDA, and hidden markov model (HMM)	18	1	8	89.3%
[27]	6	SVM	5	4	5	96%
[28]	8	MLPNN	3	5	5	99%

2. METHOD

2.1. The dataset

The dataset used in this paper is generated and made available by [29]. It contains the EMG data from eight channels. The data were collected for seven gestures: 1- index finger only, 2- middle finger only, 3- ring finger only, 4- little finger only, 5- thumb only, 6- rest state, and 7- two finger victory gesture. The data were labeled into seven classes 0 to 6 based on the performed gesture 50 persons repeated each gesture 20 times

The raw data was collected using 200 HZ sampling rate in the Myo armband, then it was preprocessed to remove the unusable data by detecting the abrupt changes, and cropping the signal. Finally, ten features were extracted from each electrode, namely: standard deviation, root mean square, minimum, maximum, zero crossings, average amplitude change, amplitude first burst, mean absolute value, waveform length, Willison amplitude. More than 6,800 set of features are extracted for each of the eight channels. Namely 6,800×80 features are available for the training across the seven gestures. These data were fed to the neural network models as described in the next section.

2.2. Myo armband

The data used in this paper were collected using a Myo armband. A Myo armband is a wearable band that uses eight EMG channels to measure the EMG signals of forearm muscles. It also includes additional sensors such as gyroscope, accelerometer, and magnetometer to aid gesture recognition. The device can send data to various platforms via Bluetooth. Myo armband drivers can recognize five gestures internally. However, raw EMG data can also be read directly from the device. These raw data are used to train the neural network models in this paper.

2.3. Classification with deep neural networks

Recently, neural networks have been widely used in EMG signal classification, as it has shown improvements in classification accuracy [30]. In this paper, two DNN architectures were used, the first one is DNN, which was employed in two models with different numbers of hidden layers. DNN has a simple architecture of fully connected layers, yet it works efficiently in complex classification problems. The second one is CNN, in which, one or more convolutional layer is presented. These layers whether they are fully connected or pooled can extract the important features of the input data. After the models' architecture was selected, the next step is to choose the training parameters as explained in the next section.

2.4. Optimizers

A deep learning system attempts to perform predictions of new data based on a specific algorithm. An optimization algorithm decides the parameter values that in turn can minimize the classification error. These optimization algorithms can affect both the accuracy of the model and the training time. An optimizer modifies the properties of the neural network such as epoch's weights and attempts to minimize the loss function. The expected function of the optimizer, therefore, is to improve the accuracy and reduce the loss [31]. Choosing a suitable optimization algorithm can be challenging since there are a very large number of parameters available. However, it is possible to detect a more favorable optimization algorithm for a specific application. The five optimization algorithms compared in this paper are Adam, Adamax, Nadam, Adagard, AdaDelta. These optimizers are briefly explained in sections 2.4.1. to 2.4.5. [31], [32].

2.4.1. Adagard

Adagard is a gradient based optimization algorithm that adapts the learning rate to the parameters. It uses a different learning curve for each parameter and for every time stop. Adagard advantages are that the learning curve changes per training parameter and it is capable of training sparse data. While its disadvantages are that it can be computationally expensive and it has a decreasing learning curve.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} . g_{t,i} \tag{1}$$

2.4.2. Adadelta

Adadeleta is an update to Adagard that tries to address the decreasing learning rate. It is a powerfull extension of the Adadelta that prevents accumulation of the previous gradients. Adadelta continues the learning regardless of updates. Its advantages are that the learning curve will not decrease compared to Adagard, however, it is still computationally expensive.

$$E[g^2]_t = \gamma . E[g^2]_{t-1} + (1-\gamma) . {g_t}^2 \tag{2}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} . g_t \tag{3}$$

2.4.3. Adam

Adaptive moment estimation (Adam) calculates adaptive learning rates similar to Adagard, and Adadelta. Adam algorithm keeps track of exponentially decaying average of past square gradients v_t and past gradients m_t:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{4}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{5}$$

Bias corrected moments:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{6}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{7}$$

The Adam equation:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} . \hat{m}_t \quad (8)$$

The proposed values for β_1 *is* 0.9 *and for* β_2 is 0.999, and 10^{-8} for ϵ. The experiments show that Adam's performance is better compared to other algorithms. Its advantage is that it is fast and has no decaying learning rate. However, it is still computationally expensive.

2.4.4. Adamax

Adamax is a variant of Adam based on infinity norm. It is a linear gradient based optimization technique which enables adjusting the learning curve based on characteristics of input data. It is suitable for time variant data such as speech.

$$v_t = \beta_2^p v_{t-1} + (1 - \beta_2^p)|g_t|^p \quad (9)$$

$$u_t = \beta_2^\infty v_{t-1} + (1 - \beta_2^\infty)|g_t|^\infty = max(\beta_2 \cdot v_{t-1}, |g_t|) \quad (10)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{u_t} . \hat{m}_t \quad (11)$$

2.4.5. Nadam

Nesterov accelerated adaptive moment estimation (Nadam) combines two algorithms, Adam and Nestrov Momentum. It is similar to Adam in working on momentum, however it replaces RMSprop with Nestrov. Nadam oprimizer's applications are mostly in noisy or high curvature gradients.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} . (\beta_1 \hat{m}_t + \frac{(1-\beta_t) g_t)}{1-\beta_1^t}) \quad (12)$$

3. DEEP NEURAL NETWORK MODELS

3.1. Model 1: Four Layer DNN

Model 1 is a simple DNN that consists of the input layer, two fully connected layers with 256 and 128 neurons, and the output layer as showm in Figure 1. Dropout technique was used to ignore random neurons during the training process. Testing the same model with different optimizers, results in lower accuracy even when increasing the number of epochs, as shown in the Table 2. Using this model, it was found that using different loss functions had no or slight effect on the accuracy, for example, Binary Cross Entropy loss function increased the accuracy to 95.01%. The training parameters were set such that the number of epochs= 100, the batch size= 32, the optimizer is Adam, and the loss function is Categorical cross entropy.

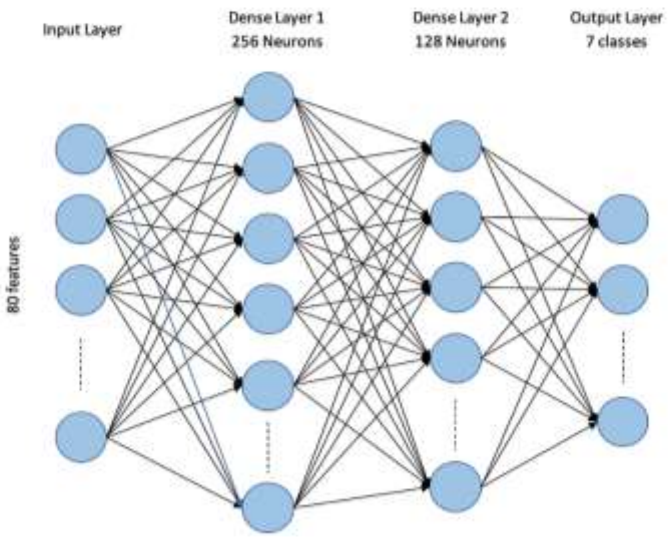


Figure 1. DNN model 1

3.2. Model 2: 8 Layer DNN

In model 2 the number of dense layers was increased. The model consists of the input layer, six fully connected layers with 1,024, 512, 256, 128, 64, and 32 neurons, and the output layer. The training parameters were set similar to model 1. Similar to model 1, testing the same model with different optimizers, resulted in lower accuracy even when increasing the number of epochs, as shown in Table 3.

Table 2. The effect of optimizer on the accuracy of the first model

Optimizer	Accuracy	Loss
Adam	94.7%	5.2%
Adamax	94.7%	5.2%
Nadam	94.4%	5.5%
Adagrad	87.4%	12.5%
AdaDelta	71.04%	28.9%

Table 3. The effect of optimizer on the accuracy of the second model

Optimizer	Accuracy	Loss
Adam	95.1%	4.8%
Adamax	95.01%	4.9%
Nadam	94.02%	5.9%
Adagrad	93.6%	6.3%
AdaDelta	72.3%	27.6%

3.3. Model 3: Five Layer CNN

In this model, two convolutional neural layers with 64 filters each were used with a kernel size of 3, in addition to one dense layer of 100 neurons. The training parameters were the same as the previous models as shown in Figure 2. The same pattern has repeated in this model. Testing the same model with different optimizers results in lower accuracy even when increasing the number of epochs, as shown in Table 4.

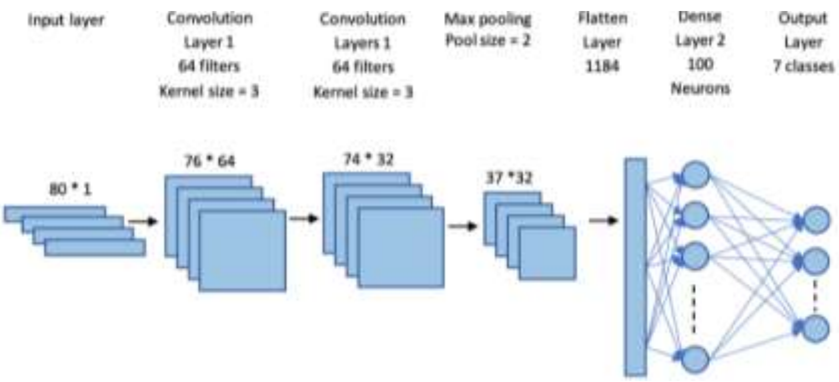


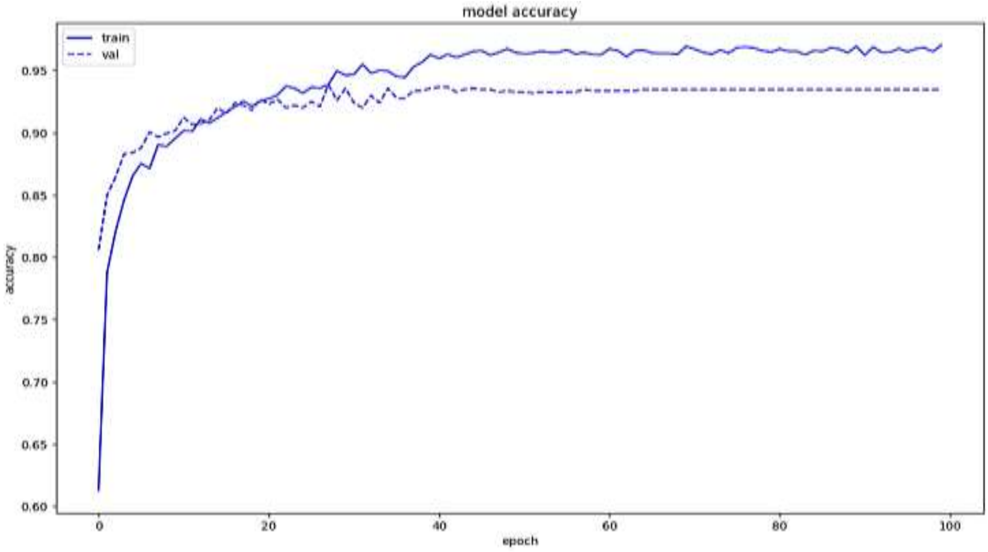
Figure 2. CNN model 3

Table 4. The effect of optimizer on the accuracy of the third model

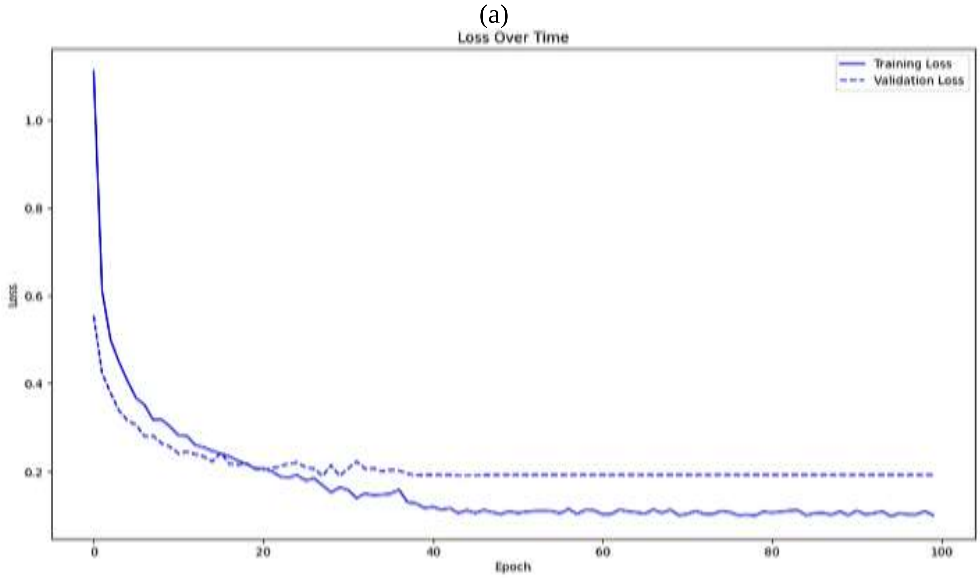
Optimizer	Accuracy	Loss
Adam	95.7%	4.2%
Adamax	95.3%	4.6%
Nadam	95.3%	4.6%
Adagrad	88.6%	11.3%
AdaDelta	71.7%	28.2%

4. RESULTS AND DISCUSSION

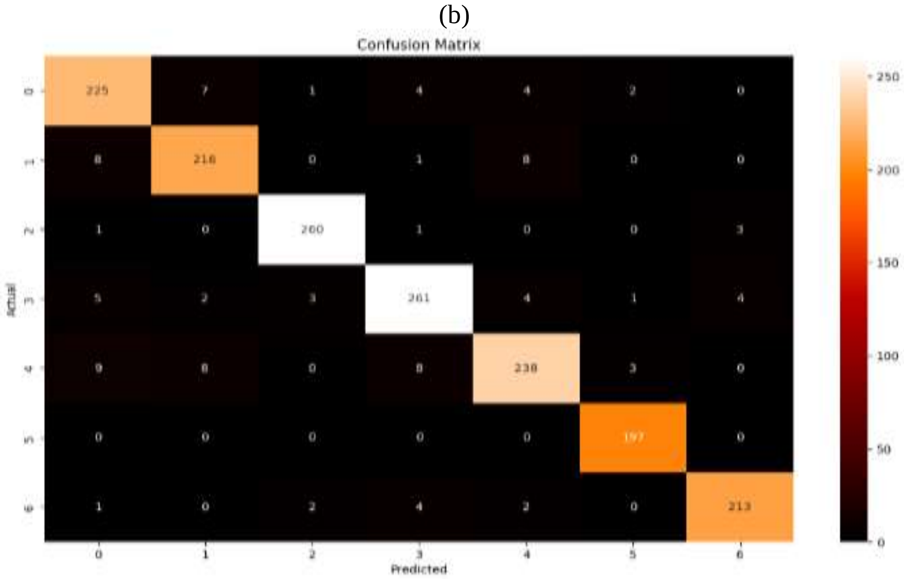
Figures 3(a)-(c), 4(a)-(c), and 5(a)-(c) show the accuracy, loss over time, and confusion matrices for Adam optimizer in models 1, 2 and 3. Table 5 summarizes the confusion matrices of the models. In all three models, class 5 was recognized accurately for up to 100% of samples. Similarly, classes 2 and 6, were recognized with better accuracy and fewer misses than the other classes. The second model had a better performance in recognizing class 6 with 98.2% accuracy. Class 4 had the most misclassifications of all classes with an accuracy between 89.5% and 92.2%. No model shows exceptional accuracy across all classes.



(a)

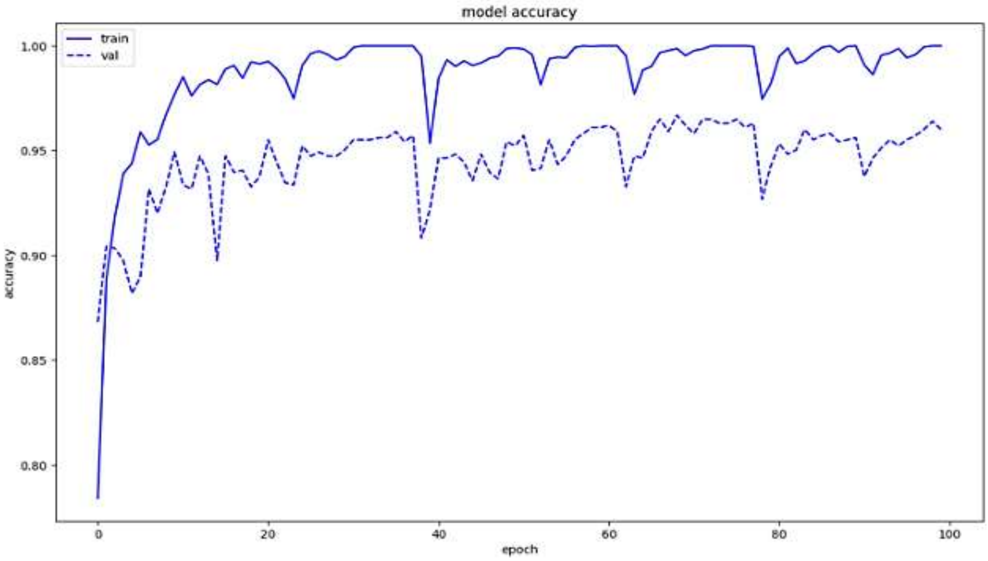


(b)

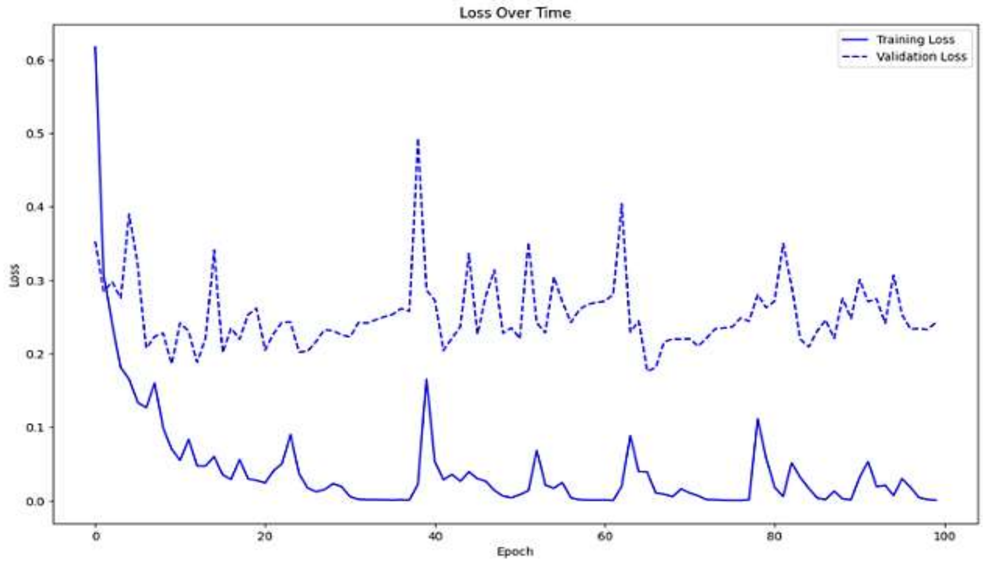


(c)

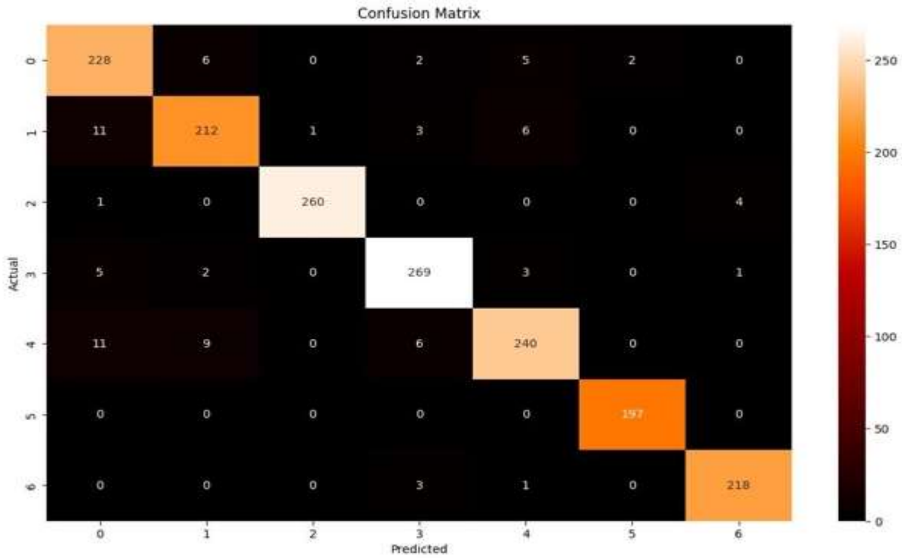
Figure 3. Model 1 accuracy (a) loss over time, (b) confusion matric, and (c) for Adam optimizer



(a)

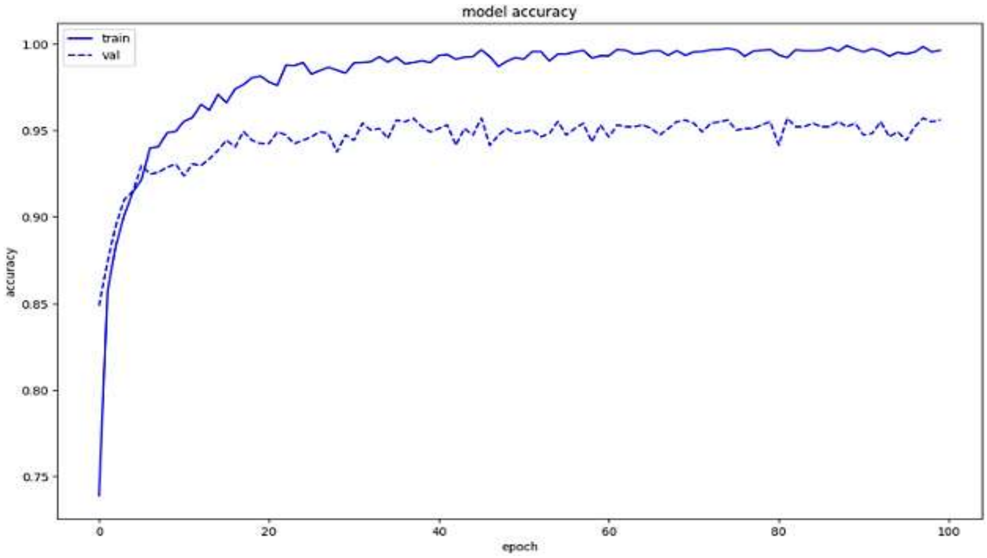


(b)

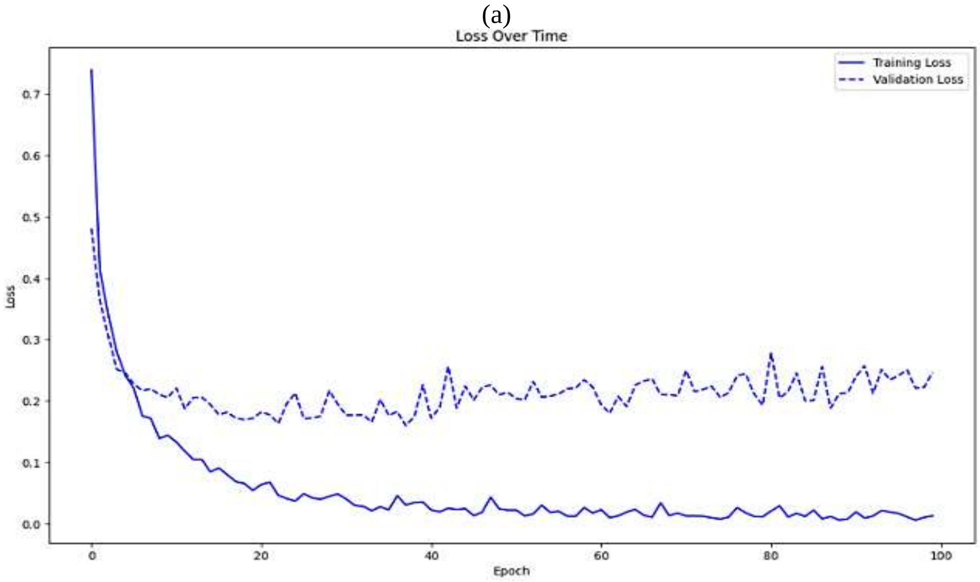


(c)

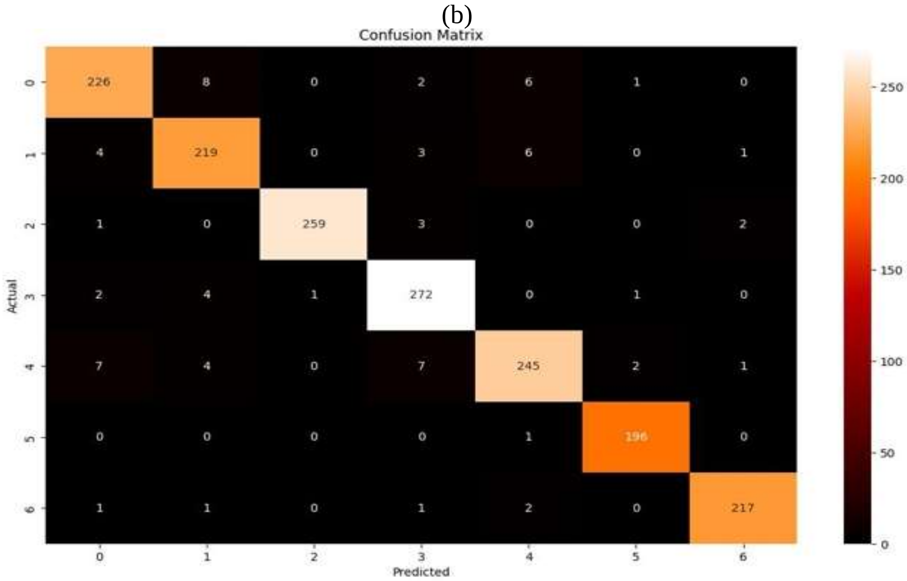
Figure 4. Model 2 accuracy (a) loss over time, (b) confusion matric, and (c) for Adam optimizer



(a)



(b)



(c)

Figure 5. Model 3 accuracy (a) loss over time, (b) confusion matric, and (c) for Adam optimizer

Table 5. Recognition accuracy of different classes in the three models

Model 1 Samples	0	1	2	3	4	5	6
Correctly classified	225	216	260	261	238	197	213
Misclassified	18	17	5	19	28	0	9
Accuracy	92.59%	92.70%	98.11%	93.21%	89.47%	100.00%	95.95%
Model 2 samples	0	1	2	3	4	5	6
Correctly classified	228	212	260	269	240	197	218
Misclassified	15	21	5	11	26	0	4
Accuracy	93.83%	90.99%	98.11%	96.07%	90.23%	100.00%	98.20%
Model 3 samples	0	1	2	3	4	5	6
Correctly classified	226	219	259	272	245	196	217
Misclassified	17	14	6	8	21	1	5
Accuracy	93.00%	93.99%	97.74%	97.14%	92.11%	99.49%	97.75%

Table 6 shows the accuracy across different models. It is clear that the accuracy improvement with increasing number of layers was negligible (about 1% between model 1 and 3). It has been found that 4 layers can achieve an acceptable accuracy rate of 94.7%. Therefore, it is not necessary to increase the number of layers in this particular model. Compared to the results reported in the literature (Table 1), the DNN model used in this paper performed better than most reported results (the datasets used in the literature vary). 95% accuracy could be acceptable for an EMG gesture recognition system.

Table 6. Comparing the recognition accuracy of the three models

Model	Number of Layers	Accuracy	Loss
Model 1	4	94.7%	5.2%
Model 2	8	95.1%	4.8%
Model 3	6 (two convolution layers)	95.7%	4.2%

5. CONCLUSION

Three DNN models were used in this paper to recognize the EMG signals extracted for seven different gestures. The dataset included 6,800 samples. 80 features were extracted for each sample from eight EMG channels. The neural network models used in this paper differed mostly in the number and type of layers (4,6, and 8), and the number of epochs. The first model consisted of the input layer, two fully connected layers with 256 and 128 neurons, and the output layer and resulted in an accuracy of up to 95.01%. The second model consisted of the input layer, six fully connected layers with 1024, 512, 256, 128, 64, and 32 neurons, and the output layer and resulted in an accuracy of up to 94.7%. The third model consisted of two convolutional neural layers with 64 filters with a kernel size of 3, in addition to one dense layer of 100 neurons. This model resulted in a recognition accuracy of upto 95.7%. The results in this paper showed that the accuracy improvement with increasing number of layers was negligible (about 1% between model 1 and 3). Therefore, in recognizing this particular dataset it is not crucial to increase the number of layers. In other words, increasing the number of layers had little to no effect on increasing the recognition accuracy. In addition, various optimizers namely, Adam, Adamax, Nadam, Adagrad, and AdaDelta, were tested across the neural network models. It was found that the Adam optimizer performed the best in recognizing the gestures in this EMG dataset. Using other optimizers resulted in lower accuracy even when increasing the number of epochs. The four-layer DNN model used in this paper archived average recognition accuracy of to 95%. This accuracy can be acceptable for a functional smart prosthesis based on EMG signals. The lower computing requirement and the acceptable accuracy of a four-layer DNN model (compared to even more layers) can be helpful with the limited computing power of the smart prosthesis devices.

REFERENCES

[1] K. F. Liem, G. E. Loeb, and C. Gans, *Electromyography for Experimentalists*, vol. 1987, no. 4. University of Chicago Press, 1987. doi: 10.2307/1445582.
[2] P. J. Kyberd, "Assessment of Functionality of Multifunction Prosthetic Hands," *J. Prosthetics Orthot.*, vol. 29, no. 3, pp. 103–111, Jul. 2017, doi: 10.1097/JPO.0000000000000139.
[3] A. M. Simon, K. L. Turner, L. A. Miller, L. J. Hargrove, and T. A. Kuiken, "Pattern recognition and direct control home use of a multi-articulating hand prosthesis," in *IEEE International Conference on Rehabilitation Robotics*, IEEE, Jun. 2019, pp. 386–391. doi: 10.1109/ICORR.2019.8779539.
[4] S. A. Dalley, T. E. Wiste, T. J. Withrow, and M. Goldfarb, "Design of a multifunctional anthropomorphic prosthetic hand with extrinsic actuation," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 6, pp. 699–706, Dec. 2009, doi: 10.1109/TMECH.2009.2033113.
[5] Y. Losier *et al.*, "An Overview of the UNB Hand System," in *MyoElectric Controls/Powered Prosthetics Symposium Fredericton*, 2011, pp. 2–5.
[6] J. T. Belter and A. M. Dollar, "Novel differential mechanism enabling two DOF from a single actuator: Application to a prosthetic

hand," in *IEEE International Conference on Rehabilitation Robotics*, IEEE, Jun. 2013. doi: 10.1109/ICORR.2013.6650441.
[7] C. Cipriani, M. Controzzi, and M. C. Carrozza, "The SmartHand transradial prosthesis," *J. Neuroeng. Rehabil.*, vol. 8, no. 1, p. 29, 2011, doi: 10.1186/1743-0003-8-29.
[8] H. Liu *et al.*, "Multisensory five-finger dexterous hand: The DLR/HIT hand II," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, IEEE, Sep. 2008, pp. 3692–3697. doi: 10.1109/IROS.2008.4650624.
[9] Y. Kamikawa and T. Maeno, "Underactuated five-finger prosthetic hand inspired by grasping force distribution of humans," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, IEEE, Sep. 2008, pp. 717–722. doi: 10.1109/IROS.2008.4650628.
[10] J. S. Hussain, A. Al-Khazzar, and M. N. Raheema, "Recognition of new gestures using myo armband for myoelectric prosthetic applications," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 6, pp. 5694–5702, Dec. 2020, doi: 10.11591/ijece.v10i6.pp5694-5702.
[11] N. E. Krausz, R. A. L. Rorrer, and R. F. F. Weir, "Design and Fabrication of a Six Degree-of-Freedom Open Source Hand," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 5, pp. 562–572, May 2016, doi: 10.1109/TNSRE.2015.2440177.
[12] Z. Chen, N. Zhang, Z. Wang, Z. Zhou, and D. Hu, "Hand gestures recognition from multi-channel forearm EMG signals," in *Communications in Computer and Information Science*, Springer Singapore, 2017, pp. 119–125. doi: 10.1007/978-981-10-5230-9_13.
[13] L. Mesin, R. Merletti, and A. Rainoldi, "Surface EMG: The issue of electrode location," *J. Electromyogr. Kinesiol.*, vol. 19, no. 5, pp. 719–726, Oct. 2009, doi: 10.1016/j.jelekin.2008.07.006.
[14] T. Jiralerspong, E. Nakanishi, C. Liu, and J. Ishikawa, "Experimental study of real-time classification of 17 voluntary movements for multi-degree myoelectric prosthetic hand," *Appl. Sci.*, vol. 7, no. 11, p. 1163, Nov. 2017, doi: 10.3390/app7111163.
[15] R. Philo, " Gray's Anatomy for Students, 2nd Ed. by Richard L. Drake, A. Wayne Vogl, and Adam W. M. Mitchell ," *Clin. Anat.*, vol. 22, no. 7, pp. 846–847, Oct. 2009, doi: 10.1002/ca.20863.
[16] K. S. Turker, "Electromyography: Some methodological problems and issues," *Phys. Ther.*, vol. 73, no. 10, pp. 698–710, Oct. 1993, doi: 10.1093/ptj/73.10.698.
[17] G. R. Naik, D. K. Kumar, and S. P. Arjunan, "Multi modal gesture identification for HCI using surface EMG," in *MindTrek - 12th International MindTrek Conference: Entertainment and Media in the Ubiquitous Era*, ACM, Oct. 2008, pp. 90–94. doi: 10.1145/1457199.1457219.
[18] X. Chen and Z. J. Wang, "Pattern recognition of number gestures based on a wireless surface EMG system," *Biomed. Signal Process. Control*, vol. 8, no. 2, pp. 184–192, Mar. 2013, doi: 10.1016/j.bspc.2012.08.005.
[19] M. C. Castro, S. P. Arjunan, and D. K. Kumar, "Selection of suitable hand gestures for reliable myoelectric human computer interface," *Biomed. Eng. Online*, vol. 14, no. 1, Apr. 2015, doi: 10.1186/s12938-015-0025-5.
[20] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation," *Sensors (Switzerland)*, vol. 17, no. 3, p. 458, Feb. 2017, doi: 10.3390/s17030458.
[21] X. Chen, X. Zhang, Z. Y. Zhao, J. H. Yang, V. Lantz, and K. Q. Wang, "Hand gesture recognition research based on surface EMG sensors and 2D-accelerometers," in *Proceedings - International Symposium on Wearable Computers, ISWC*, IEEE, Oct. 2007, pp. 11–14. doi: 10.1109/ISWC.2007.4373769.
[22] R. N. Khushaba, "Correlation analysis of electromyogram signals for multiuser myoelectric interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 745–755, Jul. 2014, doi: 10.1109/TNSRE.2014.2304470.
[23] A. Moin, A. Zhou, S. Benatti, A. Rahimi, L. Benini, and J. M. Rabaey, "Analysis of Contraction Effort Level in EMG-Based Gesture Recognition Using Hyperdimensional Computing," in *BioCAS 2019 - Biomedical Circuits and Systems Conference, Proceedings*, IEEE, Oct. 2019. doi: 10.1109/BIOCAS.2019.8919214.
[24] W. T. Shi, Z. J. Lyu, S. T. Tang, T. L. Chia, and C. Y. Yang, "A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study," *Biocybern. Biomed. Eng.*, vol. 38, no. 1, pp. 126–135, 2018, doi: 10.1016/j.bbe.2017.11.001.
[25] K. Rhee and H. C. Shin, "Electromyogram-based hand gesture recognition robust to various arm postures," *Int. J. Distrib. Sens. Networks*, vol. 14, no. 7, p. 155014771879075, Jul. 2018, doi: 10.1177/1550147718790751.
[26] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Sci. Rep.*, vol. 6, no. 1, Nov. 2016, doi: 10.1038/srep36571.
[27] X. Luo *et al.*, "Synergistic myoelectrical activities of forearm muscles improving robust recognition of multi-fingered gestures," *Sensors (Switzerland)*, vol. 19, no. 3, p. 610, Feb. 2019, doi: 10.3390/s19030610.
[28] A. M. Alkhazzar and M. Nama Raheema, "EMG Signal Classification Using Radial Basis Function Neural Network," in *2018 3rd Scientific Conference of Electrical Engineering, SCEE 2018*, IEEE, Dec. 2018, pp. 180–185. doi: 10.1109/SCEE.2018.8684162.
[29] GABRIEL GONZALEZ, "Electromyography(EMG) Dataset." https://www.kaggle.com/code/gabo515/emg-hand-gesture-analysis (accessed Nov. 01, 2022).
[30] G. Jia, H. K. Lam, J. Liao, and R. Wang, "Classification of electromyographic hand gesture signals using machine learning techniques," *Neurocomputing*, vol. 401, pp. 236–248, Aug. 2020, doi: 10.1016/j.neucom.2020.03.009.
[31] M. M. Ata, M. L. Francies, and M. A. Mohamed, "A robust optimized convolutional neural network model for human activity recognition using sensing devices," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 17, May 2022, doi: 10.1002/cpe.6964.
[32] T. Dozat, "Incorporating Nesterov Momentum into Adam," in *ICLR Workshop*, 2016, pp. 2013–2016.

BIOGRAPHIES OF AUTHORS

Ahmed M. Al-Khazzar received Ph.D. degree in Computer Security from University of Portsmouth, UK. He is currently a lecturer in University of Kerbala. His research interests are Biometrics, AI and Smart Prosthesis. He can be contacted at email: ahmed.m.ahmed@uokerbala.edu.iq.

Zainab Altaweel received master's degree in Computer Engineering from University of Bridgeport, CT, USA. She is currently an assistant lecturer in University of Kerbala. Her research interest is in Deep Learning and Human Machine Interaction. She can be contacted at email: zainab.d@uokerbala.edu.iq.

Jabbar Salman Hussain received Ph.D. degree in Electrical Engineering from University of Technology, Iraq. He is currently an assitanet professor in University of Kerbala. His research interests are AI, Smart Prosthesis, Speech Recognitions, Antenna Design and Mobile Radiation Protection Systems. He can be contacted at email: jabbar.salman@uokerbala.edu.iq.