Hand Gesture Recognition Using an Adapted Convolutional Neural Network with Data Augmentation

Ali A. Alani
Computer Science Department, College of Science,
University of Diyala
Diyala, Iraq
e-mail: alialani@sciences.uodiyala.edu.iq

Georgina Cosma
School of Science and Technology,
Nottingham Trent University
Nottingham, UK
e-mail: georgina.cosma@ntu.ac.uk

Aboozar Taherkhani
School of Science and Technology,
Nottingham Trent University
Nottingham, UK
e-mail: aboozar.taherkhani@ntu.ac.uk

T.M McGinnity
School of Science and Technology,
Nottingham Trent University
Nottingham, UK
e-mail: martin.mcginnity@ntu.ac.uk

***Abstract*—Hand gestures provide a natural way for humans to interact with computers to perform a variety of different applications. However, factors such as the complexity of hand gesture structures, differences in hand size, hand posture, and environmental illumination can influence the performance of hand gesture recognition algorithms. Recent advances in Deep Learning have significantly advanced the performance of image recognition systems. In particular, the Deep Convolutional Neural Network has demonstrated superior performance in image representation and classification, compared to conventional machine learning approaches. This paper proposes an Adapted Deep Convolutional Neural Network (ADCNN) suitable for hand gesture recognition tasks. Data augmentation is initially applied which shifts images both horizontally and vertically to an extent of 20% of the original dimensions randomly, in order to numerically increase the size of the dataset and to add the robustness needed for a deep learning approach. These images are input into the proposed ADCNN model which is empowered by the presence of network initialization (ReLU and Softmax) and L2 Regularization to eliminate the problem of data overfitting. With these modifications, the experimental results using the ADCNN model demonstrate that it is an effective method of increasing the performance of CNN for hand gesture recognition. The model was trained and tested using 3750 static hand gesture images, which incorporate variations in features such as scale, rotation, translation, illumination and noise. The proposed ADCNN was compared to a baseline Convolutional Neural Network and the results show that the proposed ADCNN achieved a classification recognition accuracy of 99.73%, and a 4% improvement over the baseline Convolutional Neural Network model (95.73%).**

Keywords-Convolutional Neural Network; Data Augmentation; Deep Learning; hand gesture recognition.

I. INTRODUCTION

Automatic hand gesture recognition using computer vision is important for various real-world applications such as Sign Language Recognition (SLR). Recently, due to the widespread availability of cost-effective digital cameras, research into sign language recognition has increased dramatically. However, hand gesture recognition still encounters challenges due to factors such as complexity of hand gesture structures, and differences in hand size and hand posture, as well as variation of in light and background, which can influence the performance of hand gesture recognition algorithms [1]. Variation in gestures also exists, because people can sign the same hand gesture differently; indeed the same person may sign the same hand gesture differently. In addition, vision-based hand gesture recognition is also susceptible to environmental factors such as observability of the hand [2], [3]. Previous approaches to hand gesture recognition using classical vision algorithms usually consist of two main steps. The purpose of the first step is to extract features, and the purpose of the second step is to apply a machine learning classifier for performing the classification task. Robust and effective feature representation is a major problem. The preceding handcrafted feature usually requires the user to have some prior knowledge and to implement some preprocessing techniques such as image transformation and segmentation to enhance data quality, extract the relevant parts and prepare the data for the recognition process. The output of the preprocessing stage is clean data that can be used directly and efficiently in the feature extraction process; the quality of pre-processing affects the success of any recognition method. However, traditional hand-designed feature extraction techniques are tedious and time-consuming, and cannot process raw images, in comparison to automatic feature extraction methods by which useful features can be retrieved directly from images.[4], [5].

Deep learning is a subset of machine learning techniques that learn high-level abstractions from data by using multiple levels of representations [4]. The Convolutional Neural Network (denoted as CNN, or ConvNet) is a class of deep feed-forward artificial neural networks that have been applied successfully in image classification and numerous other pattern recognition tasks [6]. In practice, CNN is inspired by a biological visual model, that has shown

978-1-5386-6147-5/18/$31.00 ©2018 IEEE

pioneering results in many different domains such as image classification [6], [7], object and face detection [8] on big image datasets and benchmark datasets such as ImageNet, MNIST, CIFAR-100 and CIFAR10. Currently many traditional artificial intelligence problems such as semantic parsing or visual object recognition have achieved considerable performance using deep learning algorithm techniques such as Convolutional Neural Networks (CNN) [9], [10], [11], [12]. In 2012, Krizhevsky et al. [6] achieved a remarkable result in image classification on the Large-Scale Visual Recognition Challenge (ILSVRC-2012), when the CNN deep learning algorithm was applied to the big ImageNet dataset that contains 1.2 million images with 1000 different object categories. In practice, there are three main reasons why deep learning algorithms have advantages over other neural networks. Firstly, deep learning algorithms have the ability to extract robust and significant features of the input data via several non-linear hidden layers. Secondly, deep learning algorithms have the ability to merge multiple extracted feature vectors efficiently, and finally, the deep learning algorithms have the ability to prevent the overfitting problem using different techniques (such as the dropout technique). With the recent progress of deep learning algorithms in many fields such as in the speech and image recognition domains [13], a major effort is being made to automatically learn hierarchical features from datasets by designing multistage architectures.

This paper presents an adapted version of the CNN deep learning algorithm (ADCNN) applied to the task of hand gesture recognition using the sign language of Peru (LSP) dataset. The Peru (LSP) dataset is widely used in the literature of static gesture recognition. Adopting this dataset allowed for a comparison of the proposed ADCNN against the one which was recently proposed by Flores et al. [12], and the standard baseline CNN. The contribution of this paper is an Adapted Convolutional Neural Network (ADCNN) applied to the challenge of classifying hand gesture images which vary in structure, size, posture, and environmental illumination. Data augmentation was initially applied to the data to shift images both horizontally and vertically to an extent of 20% of the original dimensions randomly; this numerically increases the size of the dataset and adds the robustness needed for a deep learning approach. These images were fed into the ADCNN model which was tuned using network initialization (ReLU and Softmax) and L2 Regularization to eliminate the problem of data overfitting. With these modifications, the ADCNN results demonstrate that it is an effective method of increasing the performance of CNN for hand gesture recognition.

The remainder of the paper is structured as follows. Section II provides a literature review; Section III presents the baseline CNN architecture and the proposed Adapted CNN architecture (ADCNN) for hand gesture recognition. The experimental methodology is described in Section IV, and the results are provided in Section V. Finally, Section VI provides a conclusion and an overview of future work.

II. Literature Review

Recognition of hand gestures plays an important role in nonverbal communication and natural human-computer interaction (HCI). Hand gesture recognition is a very active area of research in computer vision and Machine Learning [14]. Recently, Convolutional Neural Networks (CNNs) have shown substantial performance in different recognition tasks. CNNs are deep learning algorithms that extend the traditional artificial neural network by adding additional constraints to the early layers and increased the depth of the network. Recent work has focused on tuning their architecture to achieve maximum performance on benchmarks datasets such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [6], [15].

Yingxin [10] proposed an approach for hand gesture recognition based on the convolutional neural network, which is utilized to automatically extract the spatial and semantic feature of hand gesture. Their experiments were conducted with both the Cambridge Hand Gesture Dataset (9 classes) and the self-constructed dataset (5 classes), using recognition rate as the evaluation metric. Their results revealed that the CNN approach achieved higher recognition accuracy than the SIFT+SVM predefined feature approach. Flores et al. [12] proposed an approach to recognize static hand gestures whose features varied in scale, rotation, translation, illumination, noise and background. Digital image processing techniques were applied to eliminate or reduce noise, to improve the contrast under variant illumination, to separate the hand from the background of the image and finally, to detect and cut the region containing the hand gesture. Their proposed CNN was applied to perform 24-class hand gesture classification of the alphabet of sign language of Peru (LSP), and achieved 96.20% recognition accuracy. Rocco et al. [16] proposed six CNN architectures to recognize 3 classes of hand gestures: “open”, “closed” and “unknown”. Six architectures were implemented to which their hyper parameters and depth were varied to observe their behavior. The best neural network architecture achieved 73.7% classification accuracy.

John et al. [17] proposed a vision-based hand gesture recognition system for intelligent vehicles based on deep learning, which increased the drivers' comfort without affecting their safety. To extract the representative frames, they proposed tiled image patterns and a tiled binary pattern within a semantic segmentation-based deep learning framework, the deconvolutional neural network. The deep learning framework was employed to extract the representative frames from the video sequence using the deconvolution neural network (*Deconvnetn*) and long-term recurrent convolution network to classify gestures. Their results revealed a classification accuracy of 91% and reported a near real time computational complexity of 110 ms per video sequence.

Kim et al. [11] proposed a dynamic hand gesture recognition by combining a CNN with a weighted fuzzy min-max (WFMM) neural network; each module performs feature extraction and feature analysis, respectively, and their results show that the proposed method can minimize the

influence caused by the spatial and temporal variation of the feature points. Strezoski et al. [18] used the deep learning technique and Doppler radar for hand-based gesture recognition, however, their results of their method are too dependent on the orientation and distance between hand and radar device. Their experiments achieved a classification accuracy of 85.6% for 10-class hand gesture classification, and 93.1% for 7-class hand gesture classification. The authors state that because high classification accuracy is required in practical applications, it would be reasonable to use the seven gestures when the Doppler radar is employed [18].

Molchanov et al. [19] proposed an approach using a 3D Convolutional Neural Networks (3D-CNN) for drivers' hand gesture recognition, using the VIVA hand gesture recognition dataset. VIVA was designed in order to study natural human activity under difficult settings of cluttered background, volatile illumination, and frequent occlusion. The dataset was captured using a Kinect device under real-world driving settings. 3D-CNN was applied on the entire video sequence and the authors applied space-time video augmentation techniques to avoid overfitting. Their results revealed that the proposed 3D-CNN achieved 77.5% correct classification.

III. Baseline and Proposed CNN Architecture for Hand Gesture Recognition

This section provides background to the CNN, a description of the baseline CNN (CNN), and a description of the Adapted CNN (ADCNN) architectures.

A. Convolutional Neural Network

Convolutional Neural Networks have attained success in image classification problems [20], [21]. The CNN contains three types of layers; namely, convolutional layer, sub-sampling or pooling layer, and fully connected layer. Normally, the entire CNN architecture is obtained by stacking several of the above-mentioned layers. The features extracted from a CNN using the three types of layers are hierarchical. The bottom layers in a CNN collect the low-level features and high-level layers collect and learn features with more abstract information; this is useful for classification tasks [22]. A convolution layer investigates the spatially-local correlation of its input and maps it into the next layer, which is called a feature map. Different feature maps are constructed from different kernels. The activity of the i^{th} feature map in the l^{th} layer is calculated by (1)

$$\boldsymbol{y}_i^l = \sum_j f(\boldsymbol{w}_{i,j}^l * \boldsymbol{y}_j^{l-1} + \boldsymbol{b}_i^l) \quad (1)$$

where $\boldsymbol{y}_i^l$ is the i^{th} feature map in l^{th} layer. $\boldsymbol{w}_{i,j}^l$ is the convolutional kernel (weight matrix) for $\boldsymbol{y}_j^{l-1}$, and the weight matrix connects $\boldsymbol{y}_j^{l-1}$ to the feature map $\boldsymbol{y}_i^l$, $\boldsymbol{b}_i^l$ is bias of the i^{th} feature map in the l^{th} layer, and $f(.)$ is a nonlinear activation function, such as the rectified linear units (ReLU) function or sigmoid function. The '*' is denoted as the convolutional operator [22]. The first convolutional layer can be followed by the pooling layer. The function used in the pooling layer is the max function or average function, and the most common function used in this layer is the max pooling function. This max function computes the high-level feature in a local window. The polling layer used to reduce the size of the features and to decrease the required computation time [6].

B. Baseline CNN Architecture: CNN

The baseline deep CNN design is composed of two convolutional layers, two pooling layers and two fully connected layers with ReLU (Rectified Linear Unit). Three dropout performances are in the network (see Table I, Fig. 2). There are two dropout procedures after the two pooling layers, and the third dropout is performed after the first fully connected layer. The baseline structure of the hand gesture recognition model developed in this study is described in Fig.1. As shown Fig. 1, the model consists of two stages: 1) preprocessing in the first stage; and 2) pattern classification using CNN deep learning algorithm with two different architectures in the second stage.

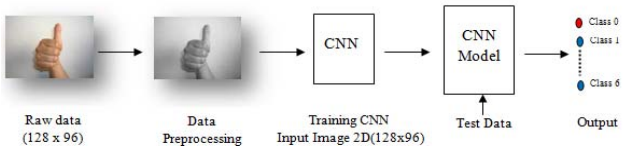

Figure 1. Overview of the proposed hand gesture recognition model.

The first convolutional layer has a kernel size of 5x5 pixels that contains 32 feature maps with a ReLU activation function. This layer takes images with 128×96 pixel values as an input (see Fig 2). The next layer is a Max-Pooling layer that is configured with a pool size of 2×2. The Max-Pooling layer uses the maximum value to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Max-Pooling layer extracts sub regions of 2×2 of the feature map, keeps their maximum value and discards all other values. The next layer is a regularization layer (Dropout) that was configured to randomly exclude 20 percent of neurons to reduce overfitting.

The next hidden layer is a convolutional layer that has a kernel size of 3×3 pixels and also contains 32 feature maps with a ReLU activation function. This layer is followed by another pooling and regularization layer that is the same as the previous layer. Afterwards, a layer called Flatten converts the two-dimensional matrix data to a vector, thereby allowing the final output to be processed by standard fully connected layers to obtain the next layers.

The first fully connected layer with the ReLU activation function contains 128 neurons. This is followed by a dropout layer to exclude 20% of neurons to reduce overfitting. The second fully connected layer containing 64 neurons with the ReLU activation function that receiving the previous 128-dimensional output of the first fully connected layer. The final part of the CNN structure is the output layer which comprises a Softmax activation function, and contains 6 neurons, one for each hand gesture recognition class. The

output is the mapping of the data to the final classes for hand gesture recognition. Fig. 2 represents the CNN method.

TABLE I. BASELINE CNN CONFIGURATION

Layers Operation	Layers Configuration
Convolution	32 filters, 5x5 kernel and ReLU
Max-Pooling	2x2 kernel
Dropout	20%
Convolution	32 filters, 3x3 kernel and ReLU
Max-Pooling	2x2 kernel
Dropout	20%
Flatten layer	800 Neurons
Fully connected	128 Neurons
Dropout	20%
Fully connected	64 Neurons
Output layer	Softmax 6 classes

C. *Proposed Adapted CNN Architecture: ADCNN*

In this section the fine-tuning and adaptation techniques applied to the basic CNN architecture are discussed. This fine-tuned and adapted model, ADCNN, is shown in Fig. 3. The performance of the baseline CNN was improved by tuning the parameters to include network initialization and regularization. The proposed approach also includes data augmentation. The basic CNN and ADCNN architectures are illustrated in Fig. 2 and Fig. 3 respectively.

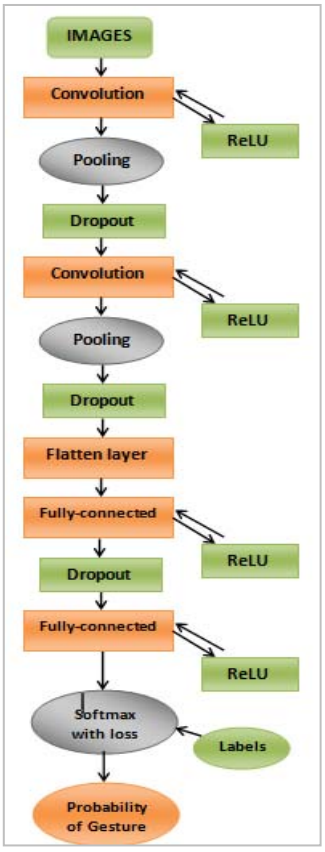

Figure 2. Architecture of the baseline deep CNN (Basic Architecture) for hand gesture recognition (CNN).

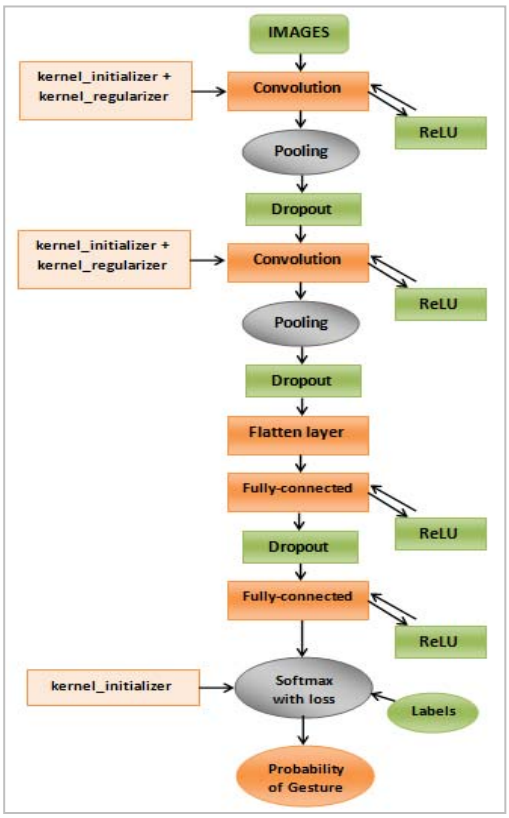

Figure 3. Architecture of the adapted deep CNN (parameters fine-tuning) for hand gesture recognition (ADCNN).

Data augmentation: It is known that the more data a machine learning algorithm has access to, the more effective it can be. Even when the data is of lower quality, algorithms can actually perform better, as long as useful data can be extracted by the model from the original dataset [23]. Data augmentation transforms the base data, and increases the number of training data. The transformed images are usually produced from the original images with very little computation and are generated during training. This means that prior to feeding data to the network for training, a transformation process on the training data are applied. Such transformations allow the network to directly observe the effects of applying them on data and to train the network to behave better on such examples. The goal is not only to reduce over-fitting via augmentation, but also to augment data to improve the classifier.

With regards to ADCNN, random horizontal and vertical shifts were applied to the data to derive new images. In particular, using data augmentation the images are shifted both horizontally and vertically to an extent of 20% of the original dimensions randomly, in order to numerically increase the size of the dataset and to add the robustness needed for a deep learning approach. The transformed images are added to the dataset in order to boost the CNN performance, prevent over-fitting, and also enabled the algorithm to learn better from a variety of images about the same gesture.

Network initialization: initializing the weights in an appropriate way can significantly influence how easily the network learns from the training set (as it effectively preselects the initial position of the model parameters with respect to the loss function to optimize). The ADCNN network architecture uses the uniform He initialization (he_uniform) for all ReLU layers and the uniform Xavier initialization (glorot_uniform) for the output Softmax layer (for effective generalization of the logistic function for multiple inputs).

L2 Regularization: This aims to decrease the complexity of the model while maintaining the same parameter count. L_2 regularization does so by penalizing weights with large magnitudes, by minimizing their L_2 norm. It uses a hyper parameter $\lambda = 0.0001$ to specify the relative importance of minimizing the norm to minimizing the loss on the training set.

IV. Experimental Methodology

This section first provides a description of the dataset which was used to perform the experiments, and the pre-processing method for preparing the images before they are fed into the baseline CNN and proposed ADCNN models. Then, it discusses the measures adopted for evaluating classification accuracy.

A. Dataset

The two architectures are trained and tested on the "Hand Gesture Dataset LSP" dataset which contains gestures for the alphabet of sign language of Peru (LSP), see Table II. The LSP dataset was developed by Flores et al. [12], and a version of that dataset was used for this research. The dataset contains 3750 hand gesture images from 25 people. Each hand gesture belongs to a class of gestures. The dataset comprises images of six unique gestures which are all used in our experiment (see Fig. 4). Generally, the task of hand gesture identification is not complex if images are taken with regular illumination. However, this dataset contains images, taken with varied illumination making the particular hand gesture recognition a challenging task. The dataset consists of a training and a testing set, as shown in Table II. The training set contains 2625 images, and the testing set contains 1125 images. Each image is in JPEG format and has a size of 128x96. Fig.4 shows a sample of the six static hand gestures that have been used for the experiments, taken under different features such as scale, rotation, translation, illumination and noise.

B. Image Pre-Processing

The convolutional layers before the first pooling layers are the bottleneck of the computational efficiency and memory requirements. In order to make a real-time classification, the images are converted to gray scale, as shown in Fig. 5. Thus, applying image pre-processing reduces the number of parameters in the first convolutional layer and reduces computational requirements. Color space conversion was also applied, permitting work on only one color channel instead of processing the three RGB channels.

C. Evaluation Measures

Different measures were adopted to evaluate the performance of the baseline and the adapted CNN algorithms. These were Precision, Recall, F1-score, and Accuracy.

Precision: It also called the Positive Predictive Value. Precision is the fraction of positive predictions divided by the total number of positive class values predicted, and it is calculated by (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

where TP and FP are True Positive and False Positive, respectively.

TABLE II. Dataset Characteristics

Dataset	Classes	Input Dimension	No. of samples	
			Training	*Testing*
LSP	6	(12288, 1)	2625	1125

Figure 4. Examples of LSP hand gestures used for the experiments.

Recall: It is also known as Sensitivity. Recall is the fraction of positive predictions divided by the number of positive class values. Equation (3) is used to calculate Recall.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where FN is False Negative. Table III, shows the notation used by formulas (2) and (3) to calculate Recall and Precision, respectively.

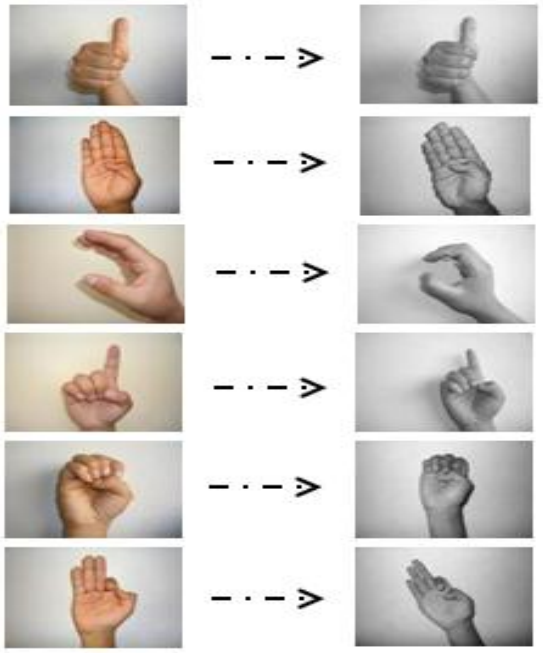

Figure 5. Image Pre-processing

TABLE III. PRECISION AND RECALL

	Relevant	Non-Relevant
Retrieved	True Positives (TP)	False Positives (FP)
Not- Retrieved	False Negatives (FN)	True Negatives (TN)

F1-score: is also called the F-score or F-measure. F1-score conveys the balance between Precision and Recall. The value of F1-score becomes high only if the values of both Precision and Recall are high. F1-scorevalues fall in the interval [0,1], and the highest the value, the better the classification accuracy. F1-scoreis calculate by (4).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Accuracy: is the percentage of correct classifications

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (5)$$

where TN is True Negative.

V. EXPERIMENTAL RESULTS

This section describes the results obtained when using the baseline and adapted CNN architectures. Experiments were performed using Python, mainly taking advantage of the Theano library, Keras, and Scikit-learn libraries. The training set from the LSP dataset (see Table II) was used, to learn features from the training data and to test the models using the testing data, as described in Section IV. Table IV shows the parameters for the CNN and ADCNN architectures. The CNN training process is based on the combination of the back propagation algorithm with the stochastic gradient descent method. The cost function chosen is the categorical *Crossentropy* loss function, and the *Adam* optimizer is used as the optimization function.

Table V presents the results when using the baseline CNN and the proposed ADCNN architectures applied to the hand gesture recognition dataset. The experiment results show that the ADCNN architecture achieved 99.73% accuracy, which was 4% higher than the performance obtained by the baseline CNN architecture (i.e. 95.73%).The confusion matrices of both CNN and ADCNN trained with 10 epochs are shown in Fig.6. The diagonal elements of the confusion matrices represent the number of images which were correctly classified (i.e. number of images for which the predicted label is equal to the true label), while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix the better the classification performance.

TABLE IV. CNN AND ADCNN PARAMETERS

Network	**Baseline CNN (CNN)**	**Proposed CNN (ADCNN)**
Number of training samples	2625	2625
Activation function	ReLU-Softmax	ReLU-Softmax
Learning rate	0.01	0.01
Iterations	10	10
Cost function	categorical crossentropy	categorical crossentropy
Optimization	Adam	Adam
Data augmentation	None	yes
Network initialization	None	ReUL(he_uniform) Softmax(glorot_uniform)
L2 Regularization	None	L2 Regularization

The overall classification performance of the ADCNN network is high, and the confusion matrices show that nearly all of the images which were misclassified by CNN, were correctly classified by ADCNN. Importantly, Fig. 7 and Fig. 8 depict the model accuracy and training loss when adopting CNN and ADCNN, respectively. These figures show that training and testing performance were close together during different epochs, which indicates that the CNN models were not overfitting the data.

TABLE V. CLASSIFICATION RESULTS

Methods	**No. Epoch**	**Precision**	**Recall**	**F1 Score**	**Accuracy (%)**
Baseline CNN	10	0.96	0.96	0.96	95.73
Proposed ADCNN	10	1	1	1	99.73

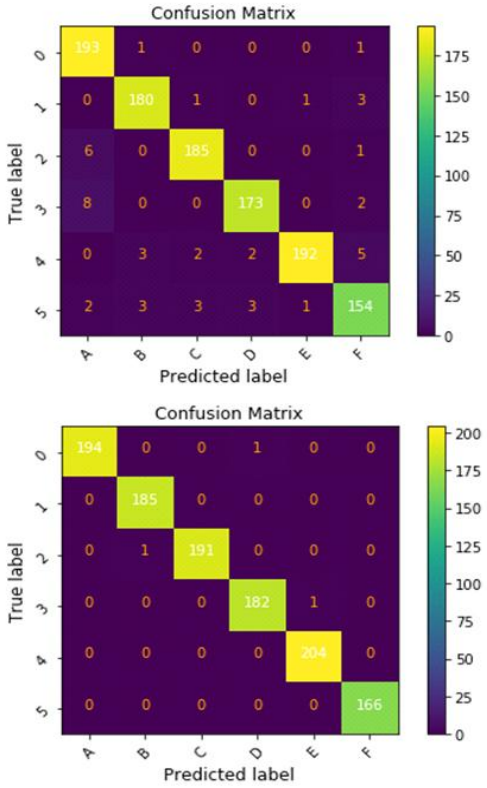

Figure 6. Confusion matrix (a) baseline CNN, (b) proposed ADCNN on the LSP dataset.

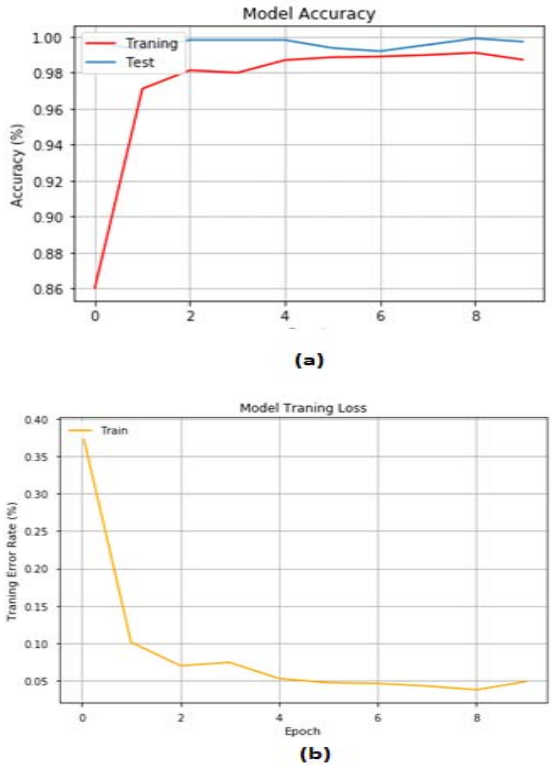

Figure 7. (a) Accuracy for training and testing data, and (b) loss during training process of ADCNN.

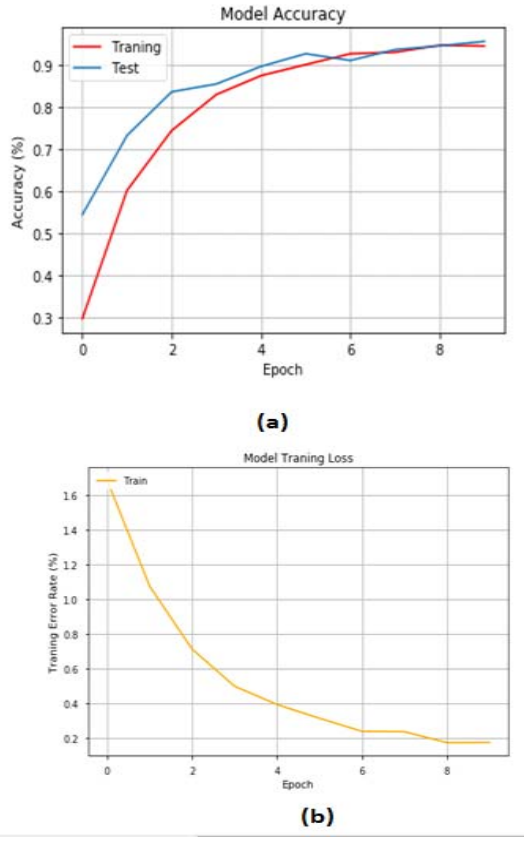

Figure 8. (a) Accuracy for training and testing data, and (b) loss during training process of CNN.

The improvement in CNN performance was attributed to the data augmentation, network initialization and regularization parameters which allowed the ADCNN to achieve much higher classification performance than the baseline CNN without overtraining. Data augmentation creates new transformed images from the base dataset, and the new images are added into the dataset to improve the model's gesture recognition accuracy. It also shifts the training images, making the image decentralized for the kernels to recognize them from various positions.

VI. Conclusion

This paper proposes an Adapted Deep Convolutional Neural Network (ADCNN) suitable for the classification of static hand gesture image datasets which vary in lighting, noise, scale, rotation and translation. Data augmentation was applied to produce transformed images from the original images and this numerically increased the size of the dataset, which in turn increased the robustness needed for a deep learning approach. The augmented images were used to improve the training of the model. The model was trained and tested using 3750 static hand gesture images taken under variations such as scale, rotation, translation, illumination and noise. Moreover, the proposed ADCNN is enhanced by the presence of dropout regularization and L2 Regularization to eliminate the problem of data overfitting. With these

modifications, the ADCNN results (described in Section V) demonstrate that it is an effective method of increasing the performance of CNN for hand gesture recognition.

The proposed ADCNN was compared to a baseline CNN. The results revealed that the proposed approach which includes data augmentation, along with a CNN tuned network using network initialization (ReLU and Softmax), and L2 regularization achieved the highest classification recognition accuracy 99.73% - a 4% improvement over the baseline CNN model which was not tuned using the above mentioned parameters. Flores et al. [12] proposed an approach to recognize static hand gestures whose features varied in scale, rotation, translation, illumination, noise and background. Their approach included applying various digital image processing techniques to eliminate or reduce noise, to improve the contrast under a variant illumination, to separate the hand from the background of the image and finally, and to detect and cut the region containing the hand gesture. Their approach achieved 96.20% recognition accuracy. However, the ADCNN requires significantly less pre-processing than that of the Flores et al. [12] approach, and it has also achieved high classification accuracy of 99.73%. Future work will extend and evaluate ADCNN with other datasets, and on real-time hand gesture classification tasks.

ACKNOWLEDGMENT

G. Cosma, A. Taherkhani and T.M. McGinnity acknowledge the financial support of The Leverhulme Trust (Research Project Grant RPG- 2016-252).

REFERENCES

[1] G. Li, H. Tang, Y. Sun, J. Kong, G. Jiang, D. Jiang, B. Tao, S. Xu, and H. Liu, “Hand gesture recognition based on convolution neural network,” Cluster Computing, 2017.

[2] S. S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: a survey,” Artificial Intelligence Review, vol. 43, no. 1, pp. 1–54, Jun. 2012.

[3] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, “Vision-based hand-gesture applications,” Communications of the ACM, vol. 54, no. 2, p. 60, Jan. 2011.

[4] A. Alani, “Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks,” Information, vol. 8, no. 4, p. 142, Sep. 2017.

[5] S. Lawrence, C. Giles, A. C. Tsoi, and A. Back, “Face recognition: a convolutional neural-network approach,” IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98–113, 1997.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks", In NIPS, 2012..

[7] M. D. Zeiler and R. Fergus,, “Visualizing and Understanding Convolutional Networks,” in European conference on computer vision, Springer, Cham, pp. 818–833, 2014.

[8] A. Toshev, C. Szegedy, D. Erhan, "Deep Neural Networks for Object Detection", Advances in Neural Information Processing Systems, 2013.

[9] H.-S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang, “Hand gesture recognition using combined features of location, angle and velocity,” Pattern Recognition, vol. 34, no. 7, pp. 1491–1501, 2001.

[10] X. Yingxin, L. Jinghua, W. Lichun, and K. Dehui, “A Robust Hand Gesture Recognition Method via Convolutional Neural Network,” 2016 6th International Conference on Digital Home (ICDH), 2016.

[11] H.-J. Kim, J. S. Lee, and J.-H. Park, “Dynamic hand gesture recognition using a CNN model with 3D receptive fields,” 2008 International Conference on Neural Networks and Signal Processing, 2008.

[12] C. J. L. Flores, A. E. G. Cutipa, and R. L. Enciso, “Application of convolutional neural networks for static hand gestures recognition under different invariant features,” 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2017.

[13] G. E. Hinton, “Reducing the Dimensionality of Data with Neural Networks,” Science, vol. 313, no. 5786, pp. 504–507, 2006.

[14] S. Mitra, T. Acharya, "Gesture Recognition: A Survey", IEEE Trans. Systems Man and Cybernetics Part C: Applications and Rev., vol. 37, no. 3, pp. 311-324, May 2007.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[16] I. Rocco, R. Arandjelovic, and J. Sivic, “Convolutional Neural Network Architecture for Geometric Matching,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[17] V. John, A. Boyali, S. Mita, M. Imanishi, and N. Sanma, “Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames,” 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2016.

[18] Strezoski, G., Stojanovski, D., Dimitrovski, I. and Madjarov, G., “Hand Gesture Recognition Using Deep Convolutional Neural Networks,” In International Conference on ICT Innovations, Springer, Cham, pp. 49-58, 2016.

[19] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, “Hand gesture recognition with 3D convolutional neural networks,” 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015.

[20] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, “The application of two-level attention models in deep convolutional neural network for fine-grained image classification,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 842-850 , 2015.

[21] T. X. Luong, B.-K. Kim, and S.-Y. Lee, “Color image processing based on Nonnegative Matrix Factorization with Convolutional Neural Network,” 2014 International Joint Conference on Neural Networks (IJCNN), pp 2130-2135. 2014.

[22] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, “Hyperspectral image classification using two-channel deep convolutional neural network,” 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5079–5082, 2016.

[23] Wang, J., & Perez, L., “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” Convolutional Neural Networks Vis. Recognit., 2017.