International Journal of Gender, Science and Technology
..ISSN: 2040-0748
UGC Care Group I Journal
Vol-13 Issue-01 Dec 2024

Hand Gesture Recognition using Convolution neural networks

Dr.Vaka Murraali Mohan[1]., Nilesh Kumar Methri[2].,Nishitha Aluri[3].,Nithin Gaddam[4].,Pranay Kunta[5].

[1] Principal& Professor of CSE, [2,3,4,5]Students B.Tech-CSE(CS),
Malla Reddy Institute of Technology and Science.,Maisammaguda.,Medchal.,Ts,India

[1]vakamuralimohan@gmail.com, [2]nileshmethri@gmail.com, [3]nishithaaluri29@gmail.com, [4]nithingoudgaddam1122@gmail.com, [5] pranaykunta12@gmail.com,

Abstract—

Computers are employed in many different sectors and are an integral part of our daily lives. Conventional input devices, such as a mouse and keyboard, enable human-computer interaction. Hand gestures may facilitate and be a helpful medium for human-computer interaction. Individual differences exist in the direction and form of gestures. Thus, there is non-linearity in this issue. The superiority of Convolutional Neural Networks (CNNs) for picture categorization and representation has been shown by recent research. A static hand gesture detection approach utilizing CNN was developed in this study because CNN can learn complicated and non-linear correlations among pictures. The dataset underwent many forms of data augmentation, including rescaling, zooming, shearing, rotation, and width and height shifting. 1600 photos total, split into 10 classes, were used for testing after the model had been trained on 8000 photographs. The accuracy of the enhanced data model was 97.12%, about 4% higher than that of the unaugmented model (92.87%).

Index Terms: Data augmentation, Convolutional Neural Network, Static hand gesture recognition.

I. INTRODUCTION

A definition of gesture has been provided by Bobick and Wilson. They describe gesture as the intentional movement of the body meant to convey information to other actors [1]. A successful gesture requires the same kind of information to be shared by the sender and the recipient. There are two categories for gestures: dynamic and static. A static gesture strives to stay almost constant over time, while a dynamic gesture aims to change over time. The emphasis of this experiment is on static gesture recognition. Applications for automatic hand gesture detection might be found in a number of fields, including virtual reality, robotics, design, and—most importantly—sign language.

How to teach a computer to recognize hand movements is the main challenge. The form and position of the fingers in different hand motions varies. Therefore, one of the aspects of hand gestures that has to be addressed is non-linearity. The content information and metadata that the photos carry may be used for this purpose. Hand gesture photos' meta information may be used to identify the motions. The procedure combines the two goals of classification and feature extraction. The characteristics of a picture must be retrieved before any gesture can be recognized. Any classification technique need to be used once those characteristics have been extracted. Therefore, how to extract and infer those characteristics for classification is the primary challenge. Large features are necessary for both recognition and categorization. Natural data in its raw form cannot be processed by conventional models for pattern recognition [2]. As a result, extracting features from raw data requires a great deal of work and is not automated. CNNs are a kind of deep learning neural network that can dynamically extract information [3], and fully linked layers have the potential to be used for categorization. In order to improve efficiency and lower computational complexity and memory constraints, CNN combines these two processes. It is also capable of comprehending the intricate and non-linear connections between the pictures.

Consequently, a CNN-based strategy will be used to address the issue. As a result, the focus of this research is on the detection of static hand movements. To do this, a CNN model that can

evaluate vast amounts of picture data and identify static hand motions was built. The other goals are to examine current gesture detection techniques and assess how data augmentation affects deep learning.

II. RELATED WORKS

The identification of hand gestures has been the subject of several studies, some of which are noteworthy. An artificial neural network (ANN) based on form fitting approach has been used to construct a hand gesture recognition system [4]. In this system, hand detection was achieved by filtering and then using a color segmentation algorithm on the YbrCr color space. The hand morphology then approached the form of the hand. An ANN was given the characteristics of hand form and finger orientation after they were retrieved. They used this strategy and were able to obtain 94.05% accuracy. Additionally, an ANN-based gesture detection system has been created [5]. Skin tones were used in this approach to segment photos. Pixel changes via cross sections, boundaries, and scalar descriptions like aspect and edge ratios were the characteristics used for the artificial neural network. These feature vectors were established and then supplied to the ANN for training. There was around 98% accuracy. It was suggested to use a statistical technique based on haar-like traits to identify motions [6]. The model in this system was learned using the AdaBoost technique. There were two tiers to the whole project. At a higher level, gestures were identified using a stochastic context-free grammar. Postures at the lower level were identified. For every input, a terminal string was produced in accordance with the grammar. Each rule's probability was computed, and the rule with the greatest probability for the provided string was chosen as the one. The input's gesture was returned together with the gesture linked to this rule.
However, there are several disadvantages to feature extraction using manual means. It takes a while to extract every characteristic, and not all of them will be. Furthermore, human bias might develop in the extraction. Next is automated feature engineering, which is neither laborious or prejudiced by humans. Additionally, automated feature engineering may be used to collect almost all of the features. CNN can extract useful characteristics from structured data. As a result, automated feature engineering was adopted, and CNN, or deep learning, began to take shape. A 3D CNN was suggested to be used in an algorithm to identify hand motions [7]. This technique relied on testing the pictures' depth and intensity for identification. On the VIVA challenge dataset, they also used the data augmentation approach, and their accuracy was about 77.5% [7]. A different CNN-based gesture detection method that is resilient to the five invariants of scale, rotation, translation, illumination, noise, and backdrop was put forward [8]. Peruvian Sign Language (LSP) was the dataset utilized. Their accuracy on the LSP Dataset was 96.20%.

III. EXPERIMENTAL METHODOLOGY

The description of the CNN setup and dataset that were used is given in this section. Figure 1 displays the technique flowchart. The method consists of gathering data, pre-processing, setting up CNN, and creating the model.

A. Training and Input Data

A camera was used to gather the images required for the model's validation and training. Ten different people made the motions.

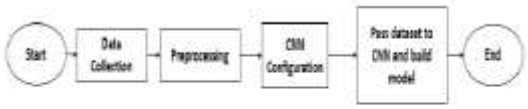


Fig. 1. System Framework

people facing the webcam. It is believed that the input photographs only include one hand, that motions were performed with the right hand, and that the hand and palm are approximately vertical and facing the camera. If the hand's contrast is strong and the backdrop is less complicated, the identification process will be quicker and more effective. Therefore, it is hypothesized that the photos' backgrounds were more homogeneous and less complicated.

B. Pre-Processing To improve efficiency and lower computational complexity, a basic pre-processing was conducted to the dataset. First, Z. ZivKovic's background removal technique [9] [10] was used to eliminate the photographs' backgrounds. The K-gaussian distribution, which chooses the proper gaussian distribution for each pixel and offers improved flexibility on changing sceneries owing to

light variations, is the primary basis for background reduction.

Once the backdrop is removed, the hand picture is all that is left. The pictures were then turned into grayscale versions. CNN will learn more quickly since grayscale pictures only have one color channel [11]. The next step was to apply morphological erosion [12]. A median filter was then used to cut down on the noise. Reducing noise is often desired in signal processing [13]. The pre-processing stages are shown in Figure 2. After then, the pictures were downsized to 50 by 50 so CNN could use them.

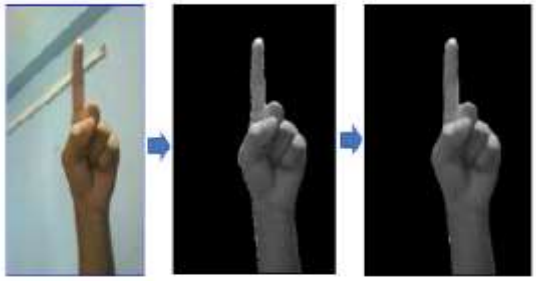


Fig. 2. Steps of Preprocessing

This experiment also made use of another dataset called the "Hand Gesture Recognition Database" [14], in addition to the one we had created ourselves. Other things from these photographs were eliminated by choosing the biggest object, in this example, the hand.

The result of choosing the biggest item is shown in Figure 3.

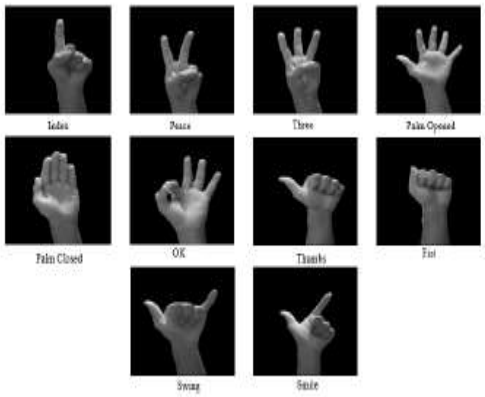
C. Information Base

Index, Peace, Three, Palm Opened, Palm Closed, OK, Thumbs, Fist, Swing, and Smile were the ten static gestures that we chose to identify. There are 160 photos for assessment and 800 images for training in each class. Thus, there are 8000 pictures in all.

(a)

(b)

Figure 3. The images are (a) the original and (b) the image after choosing the largest object (1600) for testing and training. A sample of the completed dataset may be seen in Figure 4.



The "Hand Gesture Recognition Database" [14] has ten classes with 2000 photos each, including Palm, I, Fist, Fist Moved, Thumb, Index, OK, Palm Moved, C, and Down. In Figure 5. D, a database snapshot is shown. CNN Setup

The CNN used in this study to identify hand gestures is made up of two output layers, two max pooling layers, two convolution layers, and two fully linked layers.

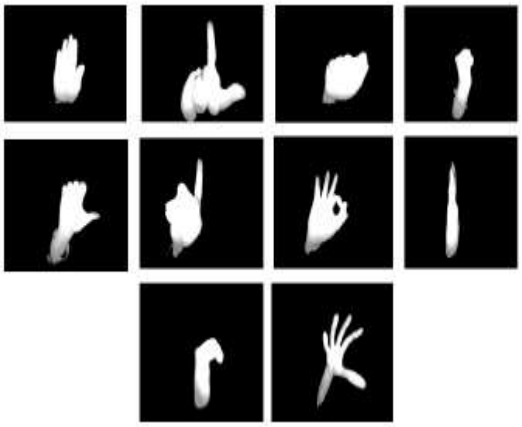


Figure 5: Sample Images from the Database Layer for Hand Gesture Recognition. To avoid over-fitting, the network has three dropout performances [15].

With a 3x3 kernel, the first convolution layer has 64 distinct filters. Rectified Linear Unit is the activation function that is used in this layer (ReLU). ReLU was used to add non-linearity [16], and it has been shown to outperform other activation functions like sigmoid or tanh. Since it is an input layer, the input size must be specified. The default stride is used. Given that the input form is 50x50x1, this network needs a grayscale picture that is 50x50 in size. The feature maps are created by this layer and are then sent to the next layer. Subsequently, a max pooling layer with a 2x2 pool size on CNN extracts the maximum value from a 2x2 window. Gradually, the representation's spatial dimension decreases as the pooling layer retains just the highest value and discards the remainder. This layer chooses just the most significant elements, which aids the network in comprehending the pictures better.

The next layer is an additional convolution layer with 64 distinct filters, a 3x3 kernel size, and the default stride.

In this layer, the activation function was once again ReLU. Another max pooling layer with a pooling size of 2x2 comes after this layer. To keep the model from over-fitting, the first dropout was included into this layer, which randomly removes 25% of the total number of neurons. The flatten layer receives the output from this layer.

Table I: Config of CNN

Model Content	Details
First Convolution Layer	64 filters of size 3x3, ReLU, input size 50x50
First Max Pooling Layer	Pooling Size 2x2
Second Convolution Layer	64 filters of size 3x3, ReLU
Second Max Pooling layer	Pooling size 2x2
Dropout Layer	Excludes 25% neurons randomly
First Fully connected Layer	256 nodes, ReLU
Dropout Layer	Excludes 25% neurons randomly
Second Fully Connected Layer	256 nodes, ReLU
Dropout Layer	Excludes 25% neurons randomly
Output Layer	10 nodes for 10 classes, SoftMax
Optimization Function	Stochastic Gradient Descent (SGD)
Learning Rate	0.001
Metrics	Loss, Accuracy

IV. EXPERIMENTAL RESULT

The findings of the experiment employing the CNN setup as shown in Table I are described in this section. According to the experimental results, the model that benefited from temporary data augmentation attained an accuracy of 97.12%, which is around 4% better than the model that did not get any augmentation data.

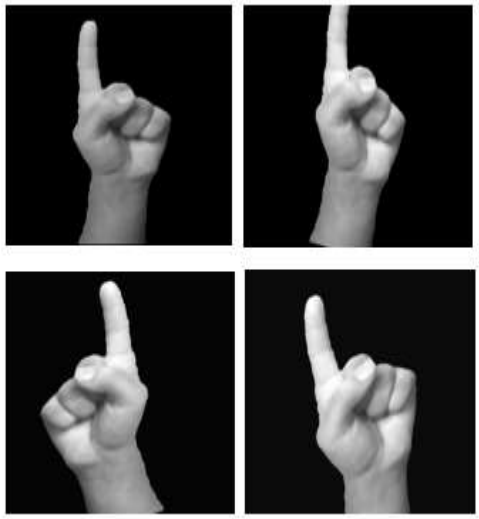
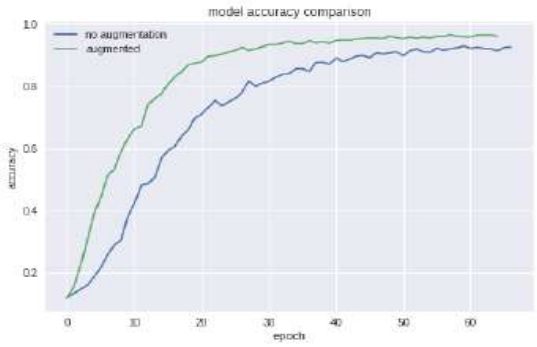
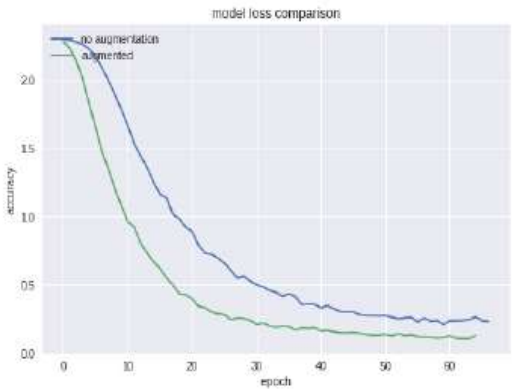


Fig. 6. Effects of Data Augmentation



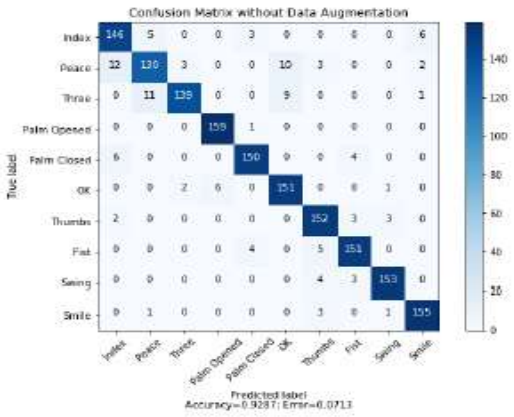
(a)



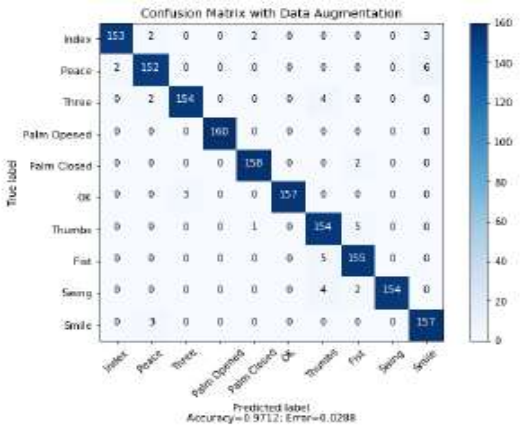
(b)

Figure 7 compares the models' (a) accuracy and (b) loss.

Figure 7 displays the accuracy and loss comparison graphs. The graphs demonstrate that after 65 epochs, the non-enhanced model was early halted, and after 67 epochs, the augmented model was early stopped. At every epoch, the enhanced model's accuracy was greater than the non-augmented model's. Additionally, it demonstrates that the enhanced model's accuracy progressed more quickly than the non-augmented model. The enhanced model's loss was also lower than that of the non-augmented model. Figure 8 provides the confusion matrices for both models. The number of tuples that the models successfully categorized is shown by the diagonal numbers, whereas the number of tuples that were incorrectly classified is indicated by the off-diagonal values. The performance improves with a greater diagonal value. The augmented model outperformed the non-augmented model on the three classes (Index, Peace, and Three) according to these matrices, indicating that the extra quantity of data supplied to the model contributed to its improved performance. That explains why the upgraded model performed better.



(a)



(b)

Figure 8: Confusion Matrix for the Following Models: (a) Non-augmented and (b) Augmented

This assertion is supported by further performance metrics shown in Table II. Figure 9 shows that the testing and training accuracy were almost same across each epoch, proving that data augmentation did not cause the model to become over-fit. A second run of the experiment was conducted with train-test split 65-35. The accuracy for the 65-35 split was found to be 96.57%, indicating that the model may be adjusted to fit a bigger test set.

Classification result for Table II

Performance Measure	CNN without augmentation	CNN with augmentation
Precision	0.9291	0.9718
Recall	0.9287	0.9713
F-Measure	0.9289	0.9715
Accuracy	92.87%	97.12%

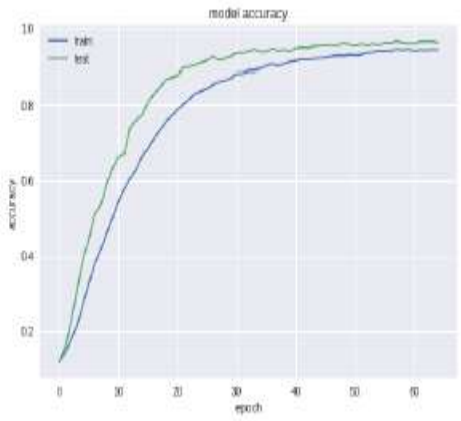


Figure 9. Training vs. Testing the Augmented Model's Accuracy

K Nearest Neighbors (KNN) and Support Vector Machine (SVM) models received the same dataset as input.

The accuracy of these models was 72% and 75%, respectively. A contributing factor to the subpar results obtained by SVM and CNN is the non-linear dataset's lack of adaptation. Because the datset was given in an unprocessed format, their accuracy was inferior to that of CNN. The accuracy of 98.95% was obtained by using CNN on the "Hand Gesture Recognition Database" [14] with a splitting ratio of 70:30, demonstrating the flexibility of the suggested approach for various datasets. Figure 10 displays the confusion matrix derived from the dataset [14].

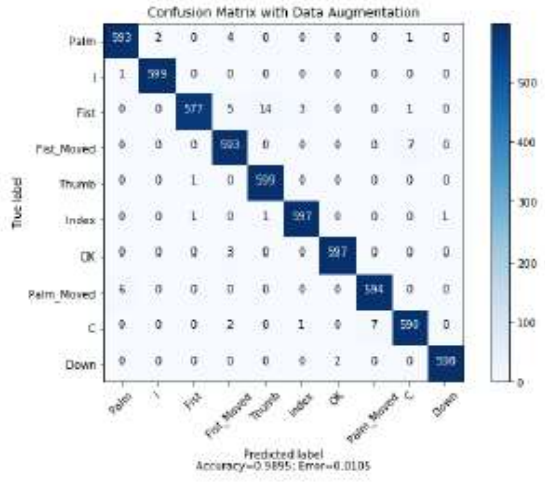


Fig. 10. Confusion Matrix of Hand Gesture Recognition Database

V. CONCLUSION AND FUTURE WORKS

This study investigates the benefits and difficulties associated with hand gesture recognition. Additionally, it examines how data augmentation affects deep learning. After doing this study, we can conclude that CNN is a data-driven approach and that deep learning greatly benefits from data augmentation. Even with the system's effective gesture recognition, there is still room for additional expansion. For instance, by using knowledge-driven methodologies like Belief Rule Base (BRB), which is

often used in situations where ambiguity arises [20] [21] [22] [23] [24]. As a result, gesture recognition may be done more precisely. The list of gestures that can be recognized may be expanded. The backdrop was supposed to be less complicated. Consequently, another extension may be the ability to recognize motions against complicated backgrounds. This system is unable to recognize motions performed with both hands. Therefore, the ability to recognize movements done with both hands may be a future project.

REFERENCES

[1] A. D. Wilson and A. F. Bobick, "Learning visual behavior for gestur analysis," in Proceedings of International Symposium on Computer Vision-ISCV. IEEE, 1995, pp. 229–234.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, p. 436, 2015.

[3] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in Shape, contour and grouping in computer vision. Springer, 1999, pp. 319–345.

[4] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," Engineering Applications of Artificial Intelligence, vol. 22, no. 8, pp. 1141–1158, 2009.

[5] T.-N. Nguyen, H.-H. Huynh, and J. Meunier, "Static hand gesture recognition using artificial neural network," Journal of Image and Graphics, vol. 1, no. 1, pp. 34–38, 2013.

[6] Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using haar-like features and a stochastic context-free grammar," IEEE transactions on instrumentation and measurement, vol. 57, no. 8, pp. 1562–1571, 2008.

[7] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 1–7.

[8] C. J. L. Flores, A. G. Cutipa, and R. L. Enciso, "Application of convolutional neural networks for static hand gestures recognition under different invariant features," in 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON). IEEE, 2017, pp. 1–4.

[9] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in null. IEEE, 2004, pp. 28–31.

[10] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," Pattern recognition letters, vol. 27, no. 7, pp. 773–780, 2006.

[11] M. Grundland and N. A. Dodgson, "Decolorize: Fast, contrast enhancing, color to grayscale conversion," Pattern Recognition, vol. 40, no. 11, pp. 2891–2896, 2007.

[12] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," IEEE transactions on pattern analysis and machine intelligence, no. 4, pp. 532–550, 1987.