Hand Gesture Recognition Using Leap Motion Controller, Infrared Information, and Deep Learning Framework

Bryan Toalumbo[1](✉) and Rubén Nogales[1,2]

[1] Universidad Técnica de Ambato, Ambato, Ecuador
{btoalumbo7749,re.nogales}@uta.edu.ec
[2] Escuela Politécnica Nacional, Quito, Ecuador
ruben.nogales@epn.edu.ec

Abstract. Hand gesture recognition (HGR) systems are the current topic, attracting interest in many fields. This broad interest is because people use hand movements to communicate and interact with the physical world. HGR systems are overgrowing, and the reason is that they have applications for different fields of study. Fields can be human-computer interaction (HIC), augmented and virtual reality, robotics, medicine, and video games. Recognizing the frames to correspond to the hand gesture from a frames sequence is essential to developing HIC systems. Thus, this paper presents algorithms to detect the images corresponding to a hand gesture from a frame sequence acquired by the Leap Motion Controller. The frames sequence contains non-gestures images because the movement follows a video pattern in which the initial and final images correspond to the transition of the gesture. Therefore, this paper develops an automatic (AID) and manual (MID) images discriminator. Every algorithm returns a dataset with images corresponding to the hand gesture. To validate the algorithms, we present an HGR model with every algorithm. The models take as input the new dataset and feed an architecture based on convolutional neural networks (CNN). Our models recognize five static gestures: open hand, fist, wave in, wave out and pinch. The results show a classification accuracy of 92.31% with MID and 94.70% with AID.

Keywords: Hand gesture recognition · Convolutional neural network · Leap motion controller

1 Introduction

Hand gesture recognition (HGR) systems are active research. This wide interest is because, with hands, people can communicate and interact with the physical world [1, 2]. Likewise, HGR systems are a challenge for researchers because they seek to obtain high accuracy values in classification and recognition using machine learning (ML) models. ML models can fall into overfitting scenarios caused by data sparsity and the high dimensionality of the problem. Moreover, the applications of HGR systems are adaptable to different fields of study. The areas can be human-computer interaction (HCI), robotics,

F. R. Narváez et al. (Eds.): SmartTech-IC 2021, CCIS 1532, pp. 412–426, 2022.
https://doi.org/10.1007/978-3-030-99170-8_30

sign language interpreting, virtual and augmented reality, medicine, and video games [3, 4]. In [5] presents a rehabilitation application for improving upper extremity activity and mobility. Similarly, in [6], an application for the control of electronic devices in operating rooms is presented. In [7] offers an application for the management of a robot in rescue operations. Likewise, in [8], an analysis of human behavior in an instructional and learning scenario in a classroom is presented. The previous applications changed the way people and computers interact with each other due to non-invasive sensor technology.

In [9], the authors classify non-invasive sensors for HGR systems into two categories. The first category includes wearable sensors like Myo Armband and Smart Gloves using inertial sensors, for example: accelerometers, magnetometers, and gyroscopes. These sensors improve the way of interaction. Nevertheless, it presents some limitations in sensitivity measurements, signals noise level, device calibration, discomfort, and sweating due to prolonged use. The second category is non-contact sensors, and it is generally used in 3D depth cameras. Some of them are Microsoft Kinect, Intel RealSense Camera, and Leap Motion Controller (LMC). Sensors to the second category generate more excellent safety and comfort for the user. It also presents problems with sensitivity to lighting conditions, occlusion, complex backgrounds, and especially the interaction in front of the sensor [9, 10]. On the other hand, it provides hand movement in two types of data as spatial position data and images [9].

The spatial position hand gesture is mentioned in papers [9–15]. These papers present HGR models with classifiers as Long-Short-Term Memory (LSTM), Support Vector Machine (SVM), K-Nearest Neighbor (KNN). Whereas, the papers [8, 16–21] use non-contact sensor images to develop HGR models. These HGR models apply pre-processing and feature extraction techniques to the images and use classifiers like Random Forest (RF), Dynamic Time Warping (DTW), SVM, KNN, CNN to classify hand gestures. Similarly, the papers [22, 23] use a CNN architecture to learn the features and classify the image automatically. However, only the papers [24–27] use CNN architecture with LMC's images to develop HGR models.

In [11, 13], the authors mention that the LMC is a low-cost, accurate and dedicated device for capturing hand movements. In addition, LMC can be tracking the hand in a range of 150° vast and 60 cm high, with an accuracy of 0.01 mm [28]. The LMC uses infrared cameras to retrieve images, spatial positions of the hand and fingers. This estimation is about the 3D coordinate axes, whose origin is in the sensor's center. The LMC returns a sequence of grayscale images *f (h, w, 1)*, …, *f (h, w, T),* where the image *f (h, w, t)* contains a snapshot of the hand movement at time t, with t = 1, 2, 3, …, T. The position of the fingertips at time t is represented using the matrix $\boldsymbol{P}_t = [p^{(x)}_{(1,t)}, p^{(y)}_{(1,t)}, p^{(z)}_{(1,t)}; \ldots; p^{(x)}_{(5,t)}, p^{(y)}_{(5,t)}, p^{(z)}_{(5,t)}]^{(leap)}_t$, being $\left[p^{(x)}_{(i,t)}, p^{(y)}_{(i,t)}, p^{(z)}_{(i,t)}\right]$ the vector with the spatial positions of the *i*-th finger concerning the sensor coordinate axes.

This paper aims to recognize five static gestures: open hand, fist, wave in, wave out, and pinch using images captured by the LMC. According to the literature review, these gestures are the most commonly used in HCI applications. For this reason, we use the dataset from [29]. The dataset contains frame sequences that describe the five hand gestures mentioned. In this context, our paper is divided into two parts. The ***first*** part is an automatic (AID) and manual (MID) image discriminator to recognize images containing the hand gesture from frames sequences. Every algorithm returns a dataset with images corresponding to the hand gesture. The ***second*** part is the creation of a CNN architecture to validate AID and MID algorithms. We generate two HGR models. MID dataset trains the first model, and the AID dataset train the second model. Then, we test every model and compare the results. The results are very close, but the second model classifies better than the first model. It is because the AID algorithm discriminates the images in a better way.

AID uses $\boldsymbol{P}_t$ Signals to recognize the block of frames $f(h, w, t_i), \ldots, f(h, w, t_j)$ that contain hand gestures. The t_i is the starts zone, and t_j is the ends zone of a gesture. Then, every element from $f(h, w, t_i), \ldots, f(h, w, t_j)$ is pre-processing to remove the background and noise. Finally, we use the Point Feature Matching (PFM) algorithm to discriminate spullier images from $f(h, w, t_i), \ldots, f(h, w, t_j)$.

MID involves the researchers, and they select and discriminate the images that correspond to a gesture-based on their perception. Then, we remove the background and noise from the image through a pre-process.

2 Related Works

This section proposes review literature about the HGR problem using a Convolutional Neural Network and images acquired by the LMC. We use scientific databases like Science Direct, Springer, ACM digital library, IEEE Xplorer, and a scientific journal, Plos ONE. In the same sense, we used a search string that includes all problem keywords and logic operators. The keywords are Hand Gesture Recognition, Leap Motion Controller, Images, Infrared images, infrared imagery, Convolutional Neural Network. The search string is (hand gesture recognition) AND (leap motion controller OR ("LMC")) and (images OR ("infrared imagery")) AND (convolutional neural network OR ("CNN")). The results obtained filterer by the inclusion and exclusion criteria defined in Table 1.

The inclusion and exclusion criteria showed the works described below.

In [24] proposed an HGR system using images captured by the LMC. The dataset has 800 images of four gestures from five users. The images are segmented using the Gray Threshold technique, and they perform several experiments on the Speeded-Up Robust Features (SURF), Local Binary Pattern (LBP), and Geometric Structure feature extractors. The system uses the Radial Base Function (RBF) neural network as a classifier and reports 99.5% recognition. They mention that LBP performs poorly when the size of the image changes. This paper does not report the amount of data to train, test, and evaluate the neural network.

In [25] proposed a hand gesture recognition system based on infrared images acquired by the LMC. This system characterizes the hand gesture by calculating Depth Spatiograms of Quantized Patterns (DSQP). However, DSQP is an improved modification of LBP, but with too large a feature vector [30]. They use a Compressive Sensing framework to cope with the high dimensionality of the image descriptor by reducing the number of features. They employ an SVM for gesture recognition applying a One-vs-All strategy. The dataset has 2000 images of 10 gestures from 10 users, and it divides into 50/50 for training and testing the system. They report a high accuracy value of 99% to the system.

Table 1. Inclusion and exclusion criteria to filter related works according to our research

Type	Description
Inclusion	Publications from January 2016 to January 2021
	Only work from the databases previously described
	Papers and scientific publications focus on hand gesture recognition models through infrared imaging of the LMC with a CNN architecture
	Papers and scientific publications include the keywords in the abstract, even if they are not in the title
Exclusion	Publications before 2016
	Papers and scientific publications that don't include the use of infrared imaging to recognize hand gestures
	Non-English papers and scientific publishes
	Papers and scientific publishes based on applications but not in the proposed model

A real-time hand gesture recognition system proposes in [26]. The authors build a dataset with 15 gestures, 11 static, and four dynamics from 25 users. The authors annotate the user's distance in front of the LMC and calculate the standard deviation. This calculation is combined with the Otsu algorithm to segment the images. Then, the system extract features from images using Histogram Oriented Gradients (HOG) and LBP. The system uses two layers of SVM classifiers; the first layer is multiclass classifiers using a one-vs-all binary classifier configuration; the second layer implements each previous binary classifier as a bank of binary SVMs. Dataset is divided 80% for training and 20% for testing. They report an average recognition accuracy value of 96.02%. However, in [31], it is mentioned that HOG shows the occurrences of a specific gradient orientation, but the histogram can change considerably due to image rotation or resizing.

In [27], the authors present a system to recognize hand gestures to manipulate 3D objects interactively with images captured from the LMC. The dataset contains 12000 images from 6 gestures. This work does not mention the number of users. Each image is processed by three Feature Extraction Unit (FEU). An FEU has a convolution layer, a ReLU layer, and a max-pooling layer. They report a training and validation accuracy of 98% and 99% for the proposed system. However, the dataset is small and does not guarantee a generalization to recognize hand gestures of different users.

The papers [24–27] use LMC images to recognize hand gestures. However, their datasets are composed of static images perfectly recorded in laboratory environments, with the same light intensity, no noise, and no missing parts. But in fact, the gesture follows a video pattern with images subject to complex background, noise, variable lighting environments, and the interaction zone between the LMC sensor and the user's hand.

3 Methods

The present work uses infrared information from the LMC sensor to recognize the open hand, fist, wave in, wave out, and pinch hand gestures. To develop this work, we use a dataset from [29]. This dataset has frames sequences that represent the hand gestures mentioned above. The frames sequences include non-gesture images because the movement follows a video pattern in which the initial and final images correspond to the transition of the gesture. For this reason, we create AID and MID algorithms to recognize the hand gestures images from a frames sequence. Every algorithm returns a dataset with the hand gesture images. Then, we create a CNN architecture to validate the AID and MID algorithms. The CNN feds with the newly generated datasets and classify the hand gestures.

3.1 Dataset

This paper uses the dataset from [29], which describes a data acquisition protocol. The protocol specifies performing 30 repetitions for every hand gesture during a sampling time of 5 s. The dataset contains nine gestures, five static, and four dynamics from 56 people. Every gesture includes positions spatial sequence $\boldsymbol{P}_t$, and a images sequence $f(h, w, 1), \ldots, f(h, w, T)$, and every element is labeled with $c_t \in \{1, 2, 3, \ldots, 9\}$. The images have a dimension of 320 × 120 pixels.

The dataset presents challenges as different behavior from data per user, hand gesture frame sequences, varying lighting environments, and the different interaction zone between the LMC and the user's hand. Also, the dataset has a variable frame sampling that ranges between 16 to 225 fps. The variation is from computer data processing. These challenges approximate how a user performs the gesture in real life when interacting with a HIC system. But also, these challenges are difficult for the process of classifying and recognizing hand gestures.

3.2 Manual Image Discriminator (MID)

MID constructs a new dataset with the hand gesture images. In this algorithm, the researchers perform the process of recognizing and discriminating the images manually. Frames sequence $f(h, w, 1), \ldots, f(h, w, T)$ contains non-gesture images because the movement follows a video pattern. Figure 1 illustrates the video pattern to the fist gesture. The video starts and ends with the open hand; this shows the gesture is at the i-th time instant of the acquired frames.

The researchers save every image in folders and subfolders structure that identifies the user, the gesture, and repetition, as illustrated in Fig. 2. Images recognized by researchers are different in every repetition, and this causes images distribution in each gesture to be different, resulting in an unbalanced dataset.

An unbalanced dataset causes a classifier to be biased towards a specific class and produces lower efficiency in the classifier. To balance our dataset, we establish a limit n on the number of images selected by the researcher in every repetition. When the number of images that the researchers recognize is less than n, a random image is chosen from the

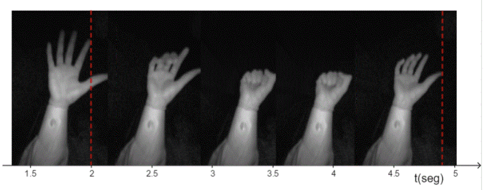
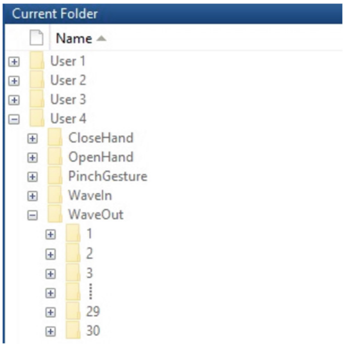


Fig. 1. Example to video pattern in fist gesture


Current Folder

Name

User 1
User 2
User 3
User 4
CloseHand
OpenHand
PinchGesture
WaveIn
WaveOut
1
2
3
⋮
29
30

Fig. 2. Folders and subfolders structures according to the user, gesture, and repetition.

selected images and doubled until complete limit n. To calculate n, we use the Hoeffding inequality formula.

$$1 - \delta = 2e^{-2\varepsilon^2 N} \tag{1}$$

Where $1 - \delta$ is confidence level, ε is margin error, and N is the sample size to test the model.

$$\mathrm{N} = \frac{\log\left(\frac{2}{1-\delta}\right)}{2\varepsilon^2} \tag{2}$$

With $\delta = 0.05$ y $\varepsilon = 0.05$; N is ≈738, N is the minimum sampling number in every gesture to test the model and minimize the overfitting risk. For this reason, we stablish empirically n value in 4. Thus, every gesture train with 5400 images and tests with 1320 images. In this sense, the dataset has 33600 images, close to the amount of data required to avoid falling into an overfitting scenario. Then, the images go through pre-processing, and it consists of applying a Laplacian filter with sigma 0.4 and gamma 0.5 to accentuate the edges. Then each image is segmented by the Gray Threshold level 2 technique to remove the background and eliminate the image's noise.

3.3 Automatic Image Discriminator (AID)

AID algorithm generates a new dataset with images corresponding to the hand gesture automatically. AID algorithm composes by Zone Values, Image Selector, Image Pre-processing, and Point Feature Matching. The Zone Values uses $\boldsymbol{P}_t$ signals to recognize and return the starts (t_i) and ends (t_j) zone values of a gesture. The Image Selector select block of frames$f(h, w, t_i), \ldots, f\left(h, w, t_j\right)$ from frames sequence of a gesture. Image Preprocessing consists in remove the background and noise from$f(h, w, t_i), \ldots, f\left(h, w, t_j\right)$. Finally, Point Feature Matching (PFM) algorithm detects an object based on finding point correspondences between the reference image and the target image. We use PFM to discriminate spullier images from$f(h, w, t_i), \ldots, f\left(h, w, t_j\right)$. Figure 3 shows the AID schema.

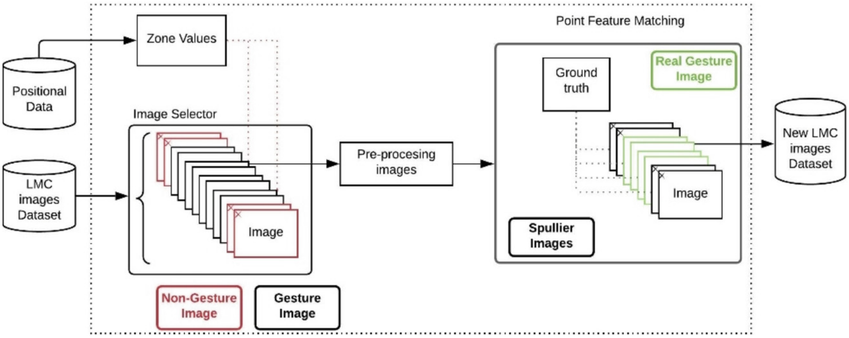


Fig. 3. AID algorithm schema

Zone Values. This algorithm receives the $\boldsymbol{P}_t$ signal as input and returns the i-th time instants where user performs gesture. The i-th time instants of a gesture are represented to t_i and t_j. Where t_i corresponds to a time instant where the user starts the gesture and t_j corresponds to a time instant where the user ends the gesture. To calculate the values for t_i and t_j. We pre-process $\boldsymbol{P}_t$ at k time instants using interpolation and extrapolation techniques. Through experimentation, we empirically defined the value for k in 70. The pre-processed P_t signal is divided into Windows of 18 with a step of 15. In every window, the pre-processed P_t signal is represented by 15 channels. The spatial positions [X, Y, Z] of each finger form the channels. Empirically, we observe that gesture representation occurs at the same time instants in all channels. In this sense, we take only one channel for processing. For every window, we calculate the spectrogram with a Short-time Fourier transform (STFT). STFT returns a matrix where the columns represent the time instants and the rows are frequencies starting at zero. From this matrix, we obtain an average vector, and we calculate the standard deviation. A standard deviations vector is getting at the end of sliding the window over the whole signal. From the standard deviations vector, we take the index of the maximum value. This index defines the corresponding window to the gesture. Figure 4 describes Zone Values process.

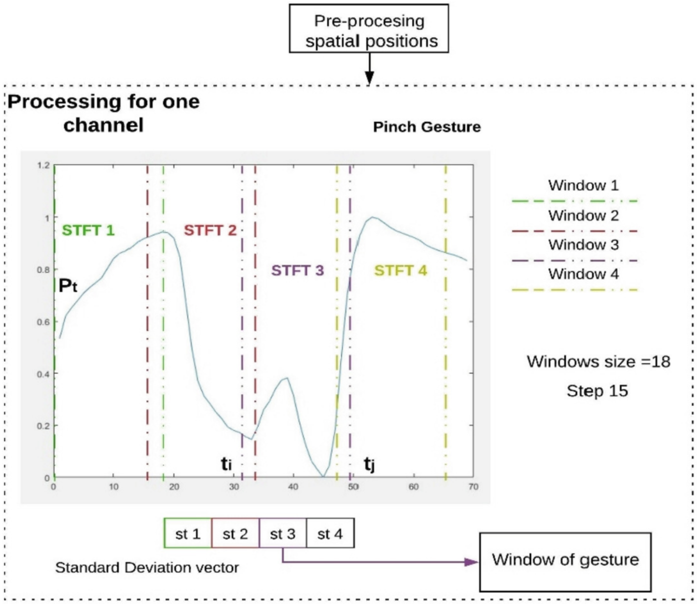


Fig. 4. Zone values process to the pinch gesture.

Image Selector. This algorithm uses the images sequence *f (h, w, 1)*, …, *f (h, w, T)* of a gesture where the image *f (h, w, t)* contains a snapshot of the hand movement at time t, with t = 1, 2, 3, …, T. To extract the block of frames that containing the gesture, we use the t time instants. We normalize t whenever T > k. If the condition is met, T divides into k, and the quotient (Q) is round. Every t element divides into Q whenever Q > 0. The results obtained from this operation are new time instants $f(h, w, t_i), \ldots, f\left(h, w, t_j\right)$. Figure 5 describe the process to standardization in 70 time instants.

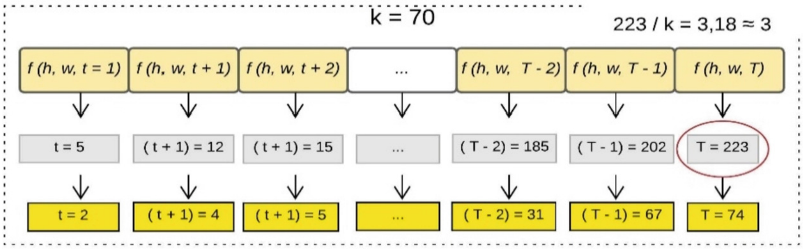


Fig. 5. Example to standardization in k time instants of a gesture.

The new time instants and the values returned by the Zones Value allow obtaining the block of images corresponding to the gesture. For example, in Fig. 4, the third window contains the hand gesture between $t_{i=33}$ and $t_{j=48}$. Then, the images between

$f(h, w, t_{i=33})$ and $f\left(h, w, t_{j=48}\right)$ are taken. Figure 6 illustrates the process of Image Selector.

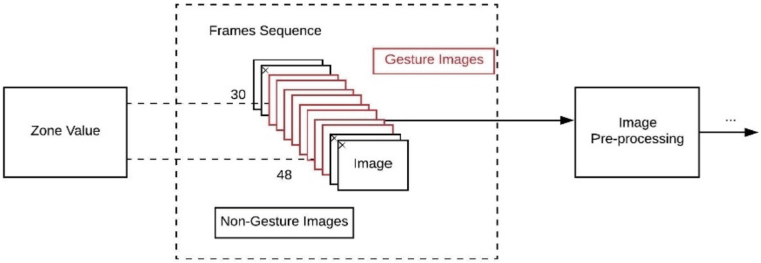


Fig. 6. Example to image selector with $t_{i=33}$ and $t_{j=48}$ in a frames sequence.

Image Pre-processing. The selected images go through pre-processing, and it consists of applying a Laplacian filter with sigma 0.4 and gamma 0.5 to accentuate the edges. Then each image is segmented by the Gray Threshold level 2 technique to remove the background and eliminate the image's noise.

Point Feature Matching. The newly generated dataset contains two problems: spullier images and the unbalanced images distribution in each gesture. To solve these problems, we use object detection using Point Feature Matching (PFM). This algorithm detects an object in an image by extracting the most characteristic points of the object and searches for matching points in the image using SURF. The operation of SURF consists of three parts: feature extraction, feature description, and feature matching. SURF detects objects despite a change of scale or rotation in the plane and is resistant to small amounts of out-of-plane rotation and occlusion [32].

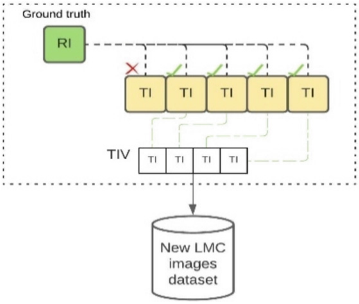


Fig. 7. PFM algorithm to discriminate spullier images

In the PFM, a Ground Truth image of each user's gesture establish as a reference image (RI) and the pre-processed images as target images (TI). From each TI, the strengths are extracted and compared with the IR strengths number. When TI strengths number is equal to the number of RI strengths (TI corresponds to an image containing the gesture), we save TI in a temporary image vector (TIV). When TIV length is less than n, and there are no more ID images, RI is added to TIV until the length of TIV equals n. In this way, the dataset obtains the images corresponding to the gesture, and the classes are balanced. Figure 7 shows the PFM algorithm.

3.4 Convolutional Neural Network

The CNN architecture is defined according to the specific problem that wishes resolved. The CNN is usually composed of two stages: feature learning and classification layers. The feature learning to our CNN architecture is composed of 6 convolutional layers, three pooling layers, six normalization layers, 6 ReLU layers, and two dropout layers. Likewise, the classification layer contains a fully connected layer, a SoftMax layer, and a pixel layer. Figure 8 shows our CNN architecture.

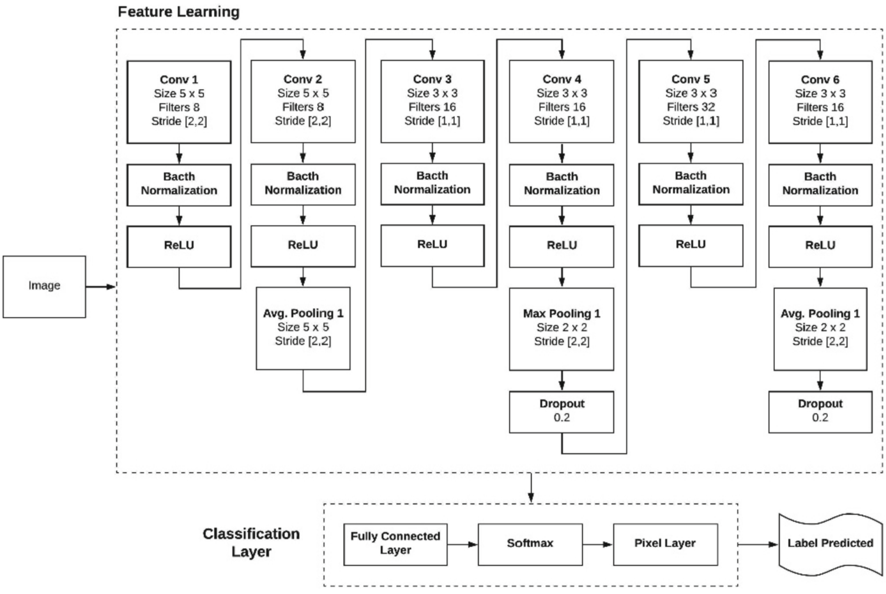

Feature Learning
Image
Conv 1 Size 5 x 5 Filters 8 Stride [2,2]
Bacth Normalization
ReLU
Conv 2 Size 5 x 5 Filters 8 Stride [2,2]
Bacth Normalization
ReLU
Avg. Pooling 1 Size 5 x 5 Stride [2,2]
Conv 3 Size 3 x 3 Filters 16 Stride [1,1]
Bacth Normalization
ReLU
Conv 4 Size 3 x 3 Filters 16 Stride [1,1]
Bacth Normalization
ReLU
Max Pooling 1 Size 2 x 2 Stride [2,2]
Dropout 0.2
Conv 5 Size 3 x 3 Filters 32 Stride [1,1]
Bacth Normalization
ReLU
Conv 6 Size 3 x 3 Filters 16 Stride [1,1]
Bacth Normalization
ReLU
Avg. Pooling 1 Size 2 x 2 Stride [2,2]
Dropout 0.2
Classification Layer
Fully Connected Layer
Softmax
Pixel Layer
Label Predicted

Fig. 8. CNN architecture

Conv 1 and Conv 2 convolutional layers have 8 filters with a size of 5 × 5 and use a stride of 2. The following convolutional layer has a size of 3 × 3 with a stride of 1. Conv 3, Conv 4, and Conv 6 convolutional layer have 16 filters and Conv 5 has 32 filter. All pooling layers have a stride of 2. The first pooling layer has a size of 5 × 5, and the following pooling layers have a size of 2 × 2.

In [33], the authors mentioned there should be multiple convolutional layers before each pooling layer to extract enough features from an image. But, on the other hand, the high dimensionality of the feature vector increases the complexity of the problem and can fall into an overfitting scenario. For this reason, we add the Dropout regularization technique to CNN to avoid falling into this problem.

4 Experimentation and Results

The experiments used an Alienware computer with Windows 10 operating System and Matlab Software version R2019B. The machine has an Intel Core i7-6800K processor and 16 Gb memory ram. To develop the CNN architecture, we use Deep Learning Toolbox with hyperparameters described in Table 2.

Table 2. CNN hyperparameters

Option	Value
Optimizer	Sgdm
Momentum	0.9000
Initial learn rate	0.0100
Learn rate schedule	Piecewise
Learn drop factor	0.2000
Learn rate drop period	5
l2 regularization	1.0000e-4
Gradient threshold method	L2norm
Max epochs	15
Mini-batch size	64
Validation data	1 × 2 cell
Validation frequency	50
Shuffle	Once
Plots	Training-progress

The MID and AID dataset is divided empirically into 80% for train and 20% for testing the model, i.e., the model trains each gesture with 5400 images and tests with 1320 images. The MID and AID models execute three times. In each run, the model obtains different images for training and testing.

Table 3 shows the training and testing accuracy values for each run of the MID model. MID model reports the average test accuracy for MID model is 92.31% with a standard deviation of 0.56%. On other hand, Table 4 shows the training and testing accuracy values for each run of the AID model. AID model reports the average test accuracy is 94.70% with a standard deviation of 2.55%.

The CNN is also used to classify hand gestures from the dataset of [25]. This dataset is available at the following link: https://www.kaggle.com/gti-upm/leapgestrecog, and the images are entered directly into the CNN. Also, the dataset of [25] is divided empirically

Table 3. Accuracy results to MID dataset.

Execution	Train accuracy	Test accuracy	Time
1	97.34%	92.92%	00:34:43
2	98.44%	92.17%	00:33:11
3	98.44%	91.83%	00:35:56

Table 4. Accuracy results to AID dataset.

Execution	Train accuracy	Test accuracy	Time
1	98.72%	97.50%	00:36:40
2	96.20%	92.50%	00:36:59
3	98.80%	94.11%	00:35:40

Table 5. Accuracy results to [25] dataset with our CNN architecture.

Execution	Train accuracy	Test accuracy	Time
1	100%	99.97%	00:16:39
2	100%	99.99%	00:15:59
3	100%	99.95%	00:16:12

into 80% for training and 20% for testing; this model executes three times. Table 5 shows the training and testing accuracy values for each run.

The average test accuracy for the dataset in [25] is high with our CNN architecture. We report to this dataset the average test accuracy is 99.97%, with a standard deviation of 0.20%. This dataset confirms the robustness of our CNN. Accuracy results for this dataset demonstrate the images were recorded in laboratory environments without noise or missing parts.

Our accuracy results obtained for MID and AID are comparable to the accuracy results reported in related works. Table 6 shows a comparative table of the results of our work compared to related works. However, our work is different from related work. Our work takes frames to correspond to the gesture from a frames sequence. The frames sequence has varying lighting environments; the user's hand gesture is at different interaction distances with the LMC. Moreover, related works take static images flawlessly executed and recorded in laboratory environments without noise or missing parts. Because of the above observations, the related works guarantee high results. On the other hand, our work has high classification accuracies, although our dataset presents different challenges for the researchers.

Table 6. Comparative table of test accuracy result of our work compares to related works.

Dataset	Classifier	Test accuracy
[24]	RBF	99.50%
[25]	SVM	99.00%
[26]	SVM	96.02%
[27]	CNN	99.00%
MID	**CNN**	**92.31%**
AID	**CNN**	**94.70%**

5 Conclusions

Recognizing the frames to correspond to the hand gesture from a frames sequence is essential for the development of real-time HIC systems. This paper presents a hand gesture recognition model using infrared information from the LMC with a CNN architecture. Our dataset contains frame sequences that describe five static gestures: open hand, fist, wave in, wave out, and pinch. The frames sequences include non-gesture images because the hand gesture follows a video pattern.

We recognize the hand gesture images from a frames sequence with AID and MID algorithms. These algorithms generate a new dataset with images corresponding to the gesture. We create the MID algorithm to verify the efficiency of the AID results. Our results are high, the AID model has accuracy of 94.70%, and the MID model has accuracy of 92.31%.

The results of MID and AID are similar, but AID has better results; this shows that the AID algorithm recognizes, selects, and discriminates the images that correspond to the hand gesture from frame sequence in a better way. Because the Zone Values and PFM algorithms perform the image recognition and discrimination process. In comparison, the images in the MID dataset are selected and discriminated by the researchers based on their perception.

The results obtained for MID and AID are comparable to the results reported in related works. However, we have challenges with our dataset like different behavior of the data per user, the sequence of frames of a gesture, variable lighting environments, different distances between the LMC and the user's hand. Moreover, related works have static images of the hand gesture, are flawlessly executed, and are recorded in laboratory environments without noise or missing parts. Because of the related works observations, their model has a good class separation and high classification accuracy. It is demonstrated in experiment executed with dataset of [25] in our CNN architecture. The results with our CNN architecture are superior to results of [25].

In this sense, the accuracy of our models tends to decrease because the users have different interaction zone with the LMC sensor in every gesture. Also, the frames sequences have images with a gesture, not recorded perfectly, noise, and complex background.

References

1. Lupinetti, K., Ranieri, A., Franca, G., Monti, M.: 3D dynamic hand gestures recognition using the Leap Motion sensor and convolutional neural networks (2020) [Online]. Available: https://manus-vr.com/. Accessed 4 Jan 2021
2. Yang, Q., Ding, W., Zhou, X., Zhao, D., Yan, S.: Leap motion hand gesture recognition based on deep neural network. In: Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020, pp. 2089–2093 (Aug. 2020). https://doi.org/10.1109/CCDC49329.2020.9164723
3. Hoang, V.T.: HGM-4: a new multi-cameras dataset for hand gesture recognition. Data Br. **30**, 105676 (2020). https://doi.org/10.1016/j.dib.2020.105676
4. Wang, Q., Wang, Y., Liu, F., Zeng, W.: Hand gesture recognition of Arabic numbers using leap motion via deterministic learning. In: Chinese Control Conference, CCC, pp. 10823–10828 (Sept. 2017). https://doi.org/10.23919/ChiCC.2017.8029083
5. Niechwiej-Szwedo, E., Gonzalez, D., Nouredanesh, M., Tung, J.: Evaluation of the leap motion controller during the performance of visually-guided upper limb movements. PLoS ONE **13**(3), 1–25 (2018). https://doi.org/10.1371/journal.pone.0193639
6. Nasr-Esfahani, E., Karimi, N., Soroushmehr, S.M.R.: Hand Gesture Recognition for Contactless Device Control in Operating Rooms (2017). https://doi.org/10.1007/s11548-017-1588-3
7. Shang, W., Cao, X., Ma, H., Zang, H., Wei, P.: Kinect-based vision system of mine rescue robot for low illuminous environment. J. Sens. **2016** (2016). https://doi.org/10.1155/2016/8252015
8. Wang, J., Liu, T., Wang, X.: Human hand gesture recognition with convolutional neural networks for K-12 double-teachers instruction mode classroom. Infrared Phys. Technol. **111**, 103464 (2020). https://doi.org/10.1016/j.infrared.2020.103464
9. Ameur, S., Ben Khalifa, A., Bouhlel, M.S.: Chronological pattern indexing: an efficient feature extraction method for hand gesture recognition with Leap Motion. J. Vis. Commun. Image Represent. **70**, 102842 (2020). https://doi.org/10.1016/j.jvcir.2020.102842
10. Raman, B., Kumar, S., Roy, P.P., Sen, D. (eds.): Proceedings of International Conference on Computer Vision and Image Processing. AISC, vol. 460. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-2107-7
11. Nogales, R., Benalcazar, M.: Real-Time Hand Gesture Recognition Using the Leap Motion Controller and Machine Learning (Nov. 2019). https://doi.org/10.1109/LA-CCI47412.2019.9037037
12. Xue, Y., Gao, S., Sun, H., Qin, W.: A Chinese sign language recognition system using leap motion. In: Proceedings – 2017 International Conference on Virtual Reality and Visualization, ICVRV 2017, pp. 180–185 (Jul. 2017). https://doi.org/10.1109/ICVRV.2017.00044
13. Ameur, S., Ben Khalifa, A., Bouhlel, M.S.: A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion. Entertain. Comput. **35**, 100373 (2020). https://doi.org/10.1016/j.entcom.2020.100373
14. Nogales, R., Benalcázar, M.E.: A Survey on Hand Gesture Recognition Using Machine Learning and Infrared Information. In: Botto-Tobar, M., Zambrano Vizuete, M., Torres-Carrión, P., Montes León, S., Pizarro Vásquez, G., Durakovic, B. (eds.) ICAT 2019. CCIS, vol. 1194, pp. 297–311. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42520-3_24
15. Gopinath, N., Anuja, J., Anusha, S., Monisha, V.: A Survey on Hand Gesture Recognition Using Machine Learning, pp. 3003–3008 (2020)
16. Huang, Y., Yang, J.: A multi-scale descriptor for real-time RGB-D hand gesture recognition. Pattern Recognit. Lett. (2020). https://doi.org/10.1016/j.patrec.2020.11.011

17. Sharma, A., Mittal, A., Singh, S., Awatramani, V.: Hand gesture recognition using image processing and feature extraction techniques. Procedia Comput. Sci. **173**, 181–190 (2020). https://doi.org/10.1016/j.procs.2020.06.022
18. Lazo, C., Sanchez, Z., del Carpio, C.: A Static Hand Gesture Recognition for Peruvian Sign Language Using Digital Image Processing and Deep Learning. In: Iano, Y., Arthur, R., Saotome, O., Vieira Estrela, V., Loschi, H.J. (eds.) BTSym 2018. SIST, vol. 140, pp. 281–290. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16053-1_27
19. Liao, B., Jing, L., Zhaojie, J., Gaoxiang, O.: Hand Gesture Recognition with Generalized Hough Transform and DC-CNN Using RealSense, pp. 84–90 (2018)
20. Pinto, R.F., Borges, C.D.B., Almeida, A.M.A., Paula, I.C.: Static hand gesture recognition based on convolutional neural networks. J. Electr. Comput. Eng., **2019** (2019). https://doi.org/10.1155/2019/4167890
21. Islam, M.R., Mitu, U.K., Bhuiyan, R.A., Shin, J.: Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language. In: Proc. 2018 4th Int. Conf. Front. Signal Process. ICFSP 2018, pp. 115–119 (2018). https://doi.org/10.1109/ICFSP.2018.8552044
22. Li, G., et al.: Hand gesture recognition based on convolution neural network. Clust. Comput. **22**(2), 2719–2729 (2017). https://doi.org/10.1007/s10586-017-1435-x
23. Chang, C.-M., Tseng, D.-C.: Loose Hand Gesture Recognition Using CNN (2019)
24. Zhang, R., Ming, Y., Sun, J.: Hand gesture recognition with SURF-BOF based on Gray threshold segmentation, pp. 118–122 (2016)
25. Blanc-Talon, J., Distante, C., Philips, W., Popescu, D., Scheunders, P. (eds.): ACIVS 2016. LNCS, vol. 10016. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48680-2
26. Mantecón, T., Del Blanco, C.R., Jaureguizar, F., García, N.: A real-time gesture recognition system using near-infrared imagery, pp. 1–17 (2019). https://doi.org/10.1371/journal.pone.0223320
27. Tripathy, S.: Natural gestures to interact with 3d virtual objects using deep learning framework. In: TENCON 2019 – 2019 IEEE Reg. 10 Conf., pp. 1363–1368 (2019). https://doi.org/10.1109/TENCON.2019.8929637
28. Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the accuracy and robustness of the leap motion controller. Sensors (Switzerland) **13**(5), 6380–6393 (2013). https://doi.org/10.3390/s130506380
29. Nogales, R., Benalcazar, M.E., Toalumbo, B., Palate, A., Martinez, R., Vargas, J.: Construction of a Dataset for Static and Dynamic Hand Tracking Using a Non-invasive Environment. In: García, M.V., Fernández-Peña, F., Gordón-Gallegos, C. (eds.) Advances and Applications in Computer Science, Electronics and Industrial Engineering. AISC, vol. 1307, pp. 185–197. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-4565-2_12
30. Mantecón, T., Mantecón, A., Del-Blanco, C.R., Jaureguizar, F., García, N.: Enhanced gesture-based human-computer interaction through a compressive sensing reduction scheme of very large and efficient depth feature descriptors (Oct. 2015). https://doi.org/10.1109/AVSS.2015.7301804
31. Cheon, M.-K., Lee, W.-J., Hyun, C.-H., Park, M.: Rotation invariant histogram of oriented gradients. Int. J. Fuzzy Log. Intell. Syst. **11**(4), 293–298 (2011). https://doi.org/10.5391/ijfis.2011.11.4.293
32. Feature Extraction Using SURF – MATLAB & Simulink – MathWorks América Latina. https://la.mathworks.com/help/gpucoder/ug/feature-extraction-using-surf.html. Accessed 29 Jul. 2021
33. Bao, P., Maqueda, A.I., Del-Blanco, C.R., Garciá, N.: Tiny hand gesture recognition without localization via a deep convolutional network. IEEE Trans. Consum. Electron. **63**(3), 251–257 (2017). https://doi.org/10.1109/TCE.2017.014971