Natural Non-Invasive Human-Machine Interface Based on Hand Gesture Recognition

Jelena Rodić, Darko Golubović, Nikola Knežević, Kosta Jovanović

***Abstract*— In recent years human-machine interfaces have been identified as an important aspect for enabling safe and efficient human-robot collaboration. In the same period of time, deep learning has made great progress in image classification problems with the evolution of convolutional neural networks This paper presents a hand gesture classification module as a non-invasive natural human-machine interface that exploits deep learning technology. There were various approaches for this task in the past, such as lookup tables, detection of key-point positions of fingers, classic neural networks, etc. This paper implements VGG16 convolutional neural network to solve the task of hand gesture. To capture an image, we use leap motion sensor which is cheap and can work in challenging light conditions, because it uses infra-red emitters to lighten the object. Thus, this approach is useful for factories and production lines. Another contribution of this paper is an extensive database consisting of 20 000 images.**

***Index Terms*— Human-machine interface; Hand gesture; Convolutional Neural Networks; Leap motion sensor; VGG16**

I. Introduction

Enhanced by developments in technology, numerous jobs are being transferred into the domain of robots and sophisticated machines [1, 2, 3]. These production lines perform various tasks. However, because market requirements are changing daily, there is a need for quick adaptation of these lines. Therefore, some manufacturers have developed flexible robotic cells that can serve multiple types of processes [4]. On the other hand, humans can adapt to changes in requirements quickly. This quality makes them valuable considering the dynamic of market needs. Combining human flexibility and robot reliability and consistency could give us a solution to the challenge the market sets without sacrificing the output volume.

The application of such a system is displayed in assembling fiscal cash registers. The process itself requires human involvement because the parts needed for assembly are small and the assembly process is delicate (mounting flat cables, soldering capacitors for PCBs, inserting tape for thermal printers). This process could be significantly less demanding for the operator by using novel technologies.

Firstly, it is possible to show the operator instructions for assembling a specific device using adaptive screens. Secondly, collaborative robots can be used to deliver the right part for the current production phase on time, which would increase the overall speed of production.

Thirdly, it is possible to gain an insight into worker performance (focus, monotony, fatigue) during tasks using EEG headphones. With this approach, the worker could switch assignments occasionally, and stay interested and productive.

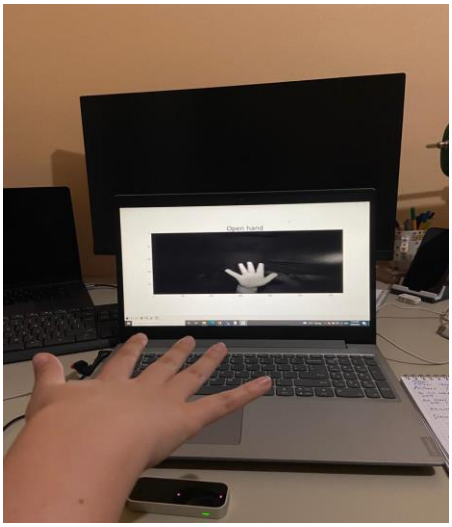

Fig. 1. System setup. Leap Motion sensor connected to PC for classification of hand gestures.

Also, if a person is engaged in dangerous work, this approach can prevent injuries from occurring when their focus is low and fatigue high. In addition to assistive devices, it is necessary to enable intuitive communication between humans and the rest of the system. For this purpose, a Leap Motion sensor is used (Fig. 1).

This sensor can detect hand gestures and provide human-machine interaction. There were a couple of solutions providing the human-machine interface but, some of them were invasive (wearing sensor gloves) or non-intuitive (complex system of buttons and taskbars). Using this type of sensor non-invasive method of human-machine interface is introduced. In this way communication between humans and machines is alleviated and it is more receptive to humans.

In this paper, we present a system for classifying human hand movements. For this purpose, it was necessary to collect the appropriate database, select the neural network architecture that will perform the classification, and finally test and implement the application that works in real-time.

II. Previous work

Paper [5] from the University of Padua implements hand movement recognition. The classification was done over ten basic hand gestures. Also, the paper focuses on static gesticulation using leap motion controllers and kinetic devices. Microsoft's Kinect camera provides 3D images, while leap motion provides only key points in 3D. Here the authors explain that the images from ordinary cameras are 2D variants of the 3D position of the hand, so there is a loss of information.

 ISBN 978-86-7466-930-3

By providing 3D finger positions leap sensor can give more meaningful measurements. With the introduction of Time-of-Flight cameras for widespread use, 3D representations of objects have become more accessible to use. However, recognizing any movements that are not the most basic is still too complex for this type of sensor (leap motion), the paper states. The authors decided to unite these two types of controllers and try to get a more accurate classification. The problem is to calibrate these two devices. They achieved this by combining the fingertips from leap motion and the peaks on the depth camera that represent the fingers. Attributes used are the number and position of the fingers, the center of the palm and the orientation of the hand, angles at which the fingers stand, and the distances between the fingers. The problem occurs when the fingertips touch because sometimes they are detected as one finger. Furthermore, rings, bracelets, etc. can spoil the accuracy too. After PCA analysis, features were input into a multiclass SVM classifier with a Gaussian kernel. The best accuracy authors achieved is 80.9%, using only the leap sensor. After they inserted the data from Kinect, the accuracy jumped to 96%.

Paper [6] deals with the recognition of dynamic hand movements. The attributes used in this paper can be divided into two categories: static and dynamic. Static ones are based on the positions of the fingers and palms, ie. at the relative distances between them. Distances between the tips of adjacent fingers, and between the center of the palm and the tip of the finger are defined as attributes. For example, the Ok symbol is represented by the distance between the thumb and forefinger as a good attribute. Dynamic attributes are defined by finger and palm speeds to detect moving patterns. Those patterns are complicated, but the velocities of the essential points are given by the sensor itself, which makes the job easier. Examples of dynamic attributes used here are the translation of the whole hand (when the palm and fingers move at the same speed along some axis, without rotation), rotation of the hand (when the palm rotates), the precession of the hand (when the palm moves in a circle), swipe index finger, etc. The results of this work show that when the controller detects fingers accurately, the classification methods themselves work accurately. The authors noticed that tracking the fingertips of the middle and little fingers is not very stable. This can cause some problems in inference.

Paper [7] deals with the application of leap controllers to realize a virtual museum for free. Interaction with the virtual world using hand gestures has been suggested. It is necessary to classify the positions of the hand to prevent unwanted moves. First, data based on x, y, and z coordinates of fingertips was obtained, and then attribute extraction was performed. The authors used information on how far the fingertips are above the plane of the palm. They also used information on angles between the fingers and the palm. The classification method used was K nearest neighbors (KNN). They state that the accuracy obtained in other papers is about 80% for the same subjects and about 70% for new ones. The exact accuracy they obtained was 99%, but it was not said how they chose the test set.

Paper [8] was done at the University of Malaysia and deals with the control of drones based on hand movements read from leap sensors. The idea is that leap motion reads the gesture, sends it to the Arduino for recognition, and then sends the command to the drone to control. Three throttle commands are classified - lift, pitch, and roll. The idea was to simplify communication with the drone, which is currently complicated since there are numerous handles, buttons, etc. Therefore, the new commands were up and down, tilt back and forth, and tilt left and right. It is said that the drone worked well when the commands were given clearly. No accuracy or classification methods were stated.

In [9] authors implemented automation of design commands in CAD, Solidworks, and Catia software. It is stated that there is a lack of tools for some very intuitive modifications. The authors wanted to replace the commands given by the mouse and keyboard. For example, the review mode requires translation, rotation, and scaling. The scaling command is the distance between two index fingers. The rotation command is rotating the fist. In the end, a grab, and release command for translation are performed based on the distance between the index fingers. Paper does not use machine learning for classification but only distance mapping. If a command cannot be mapped to the table, then it is omitted. The accuracy they get is 80%.

Paper [10] dealt with a slightly different task. They wanted to recognize handwritten symbols in the air based on the movement of the hand. The key points were sampled at a rate of 100 frames per second. The attributes taken are the positions and velocities of the key points, the angles at which the fingers stand, etc. Finally, the classification was performed using convolutional neural networks. The accuracy they obtained is 92.4%.

Paper [11] deals with the dynamic movements of human hands. Motion information is converted to color images. The conversion of movement into an image is done by tracking the finger position in time. Each finger is assigned to a distinct color, and the intensity of the color changes depending on the time. The authors used the ResNet-50 for the classification. The accuracy they get is about 80%.

III. METHOD

First, we need to define the problem we want to solve. Namely, for the needs of working on the production line in factories, it is convenient to use only commands that do not include pressing the screen and buttons when worker's hands are dirty, or the screen can't detect users' input caused by wearing the safety gloves or other equipment. For our case, it was necessary to select 4 hand commands that could be classified with high precision using the leap motion sensor. Some crucial steps in solving this problem were:

- Obtain or form an adequate dataset that has different commands given by hand movements. We decided to work on images because they are more reliable than the detection of key points from Leap sensors. Also, convolutional neural networks are powerful in image

 ISBN 978-86-7466-930-3

classification, so this method should give promising results.

- Choose 4 distinct commands, which will be able to be easily distinguished from the images returned by the Leap sensor.
- Choose the adequate architecture of the convolutional neural network and tune the hyperparameters. Assess the probability of misclassification on the test set.
- Eventually do a live simulation with a real sensor where after the first detection of the hand in the frame three frames would be captured at 0.33s interval. Then each of these three frames would be passed through a neural network for classification, and finally, a final decision would be made based on a majority vote.

A. Leap Motion Sensor

There are several sensors on the market that handle motion detection. The Leap Motion sensor [12] stands out for its price (it is cheap) and accuracy. Leap Motion Controller is an interactive device specializing in detecting hand movements and finger location based on infrared light emitters and two cameras that receive reflected IR waves. The Leap Motion architecture can be seen in Fig. 2.

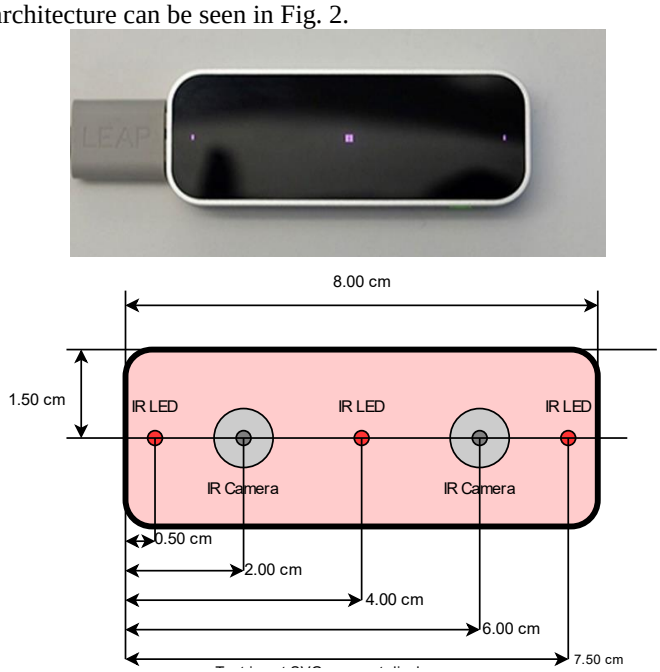

Fig. 2. Leap motion sensor, from [14]

Its field of vision is up to 1m away. There is also a skeletal model of the human hand, which represents five fingers divided into phalanges. We can also obtain a raw image from the sensor, which is the one we used.

B. Database

First approach was to try to train the network on different publicly available datasets. The best result we got was with the database [15]. The problem with this database is that it did not contain various heights or angles on which the hand can be, thus making this network prone to errors once the hand is not perfectly positioned. Considering the application of this paper, we wanted to develop a system that is able to work in real life conditions, thus making this database inadequate for our task. On our versatile test dataset, this network had the accuracy of 91.5%.

After realizing that publicly available databases are not extensive enough to meet our needs, we decided to develop our own dataset using the Leap Motion sensor. Our training set consists of four classes of hand gestures. Each class consists of 5000 images. There is an equal part of men and women in the hand dataset. Also, there are 50% left-hand images in the database. Furthermore, images were obtained at various lighting conditions, angles, and heights. The last two turned out to be important for the network to generalize well. We trained our network on one publicly available data set which had only photos of hands perpendicular to the sensor. Also, all the pictures were formed with a constant distance between the hand and the sensor. This plays a key role because the camera on the leap sensor is in a wide range. That slight change in the distance of the camera results in a tremendous change in the size of the object in the final image. As a result, the same architecture performed significantly worse than when it was trained on our custom dataset. This shows us the importance of a well-rounded dataset, which achieved accuracy of 99.3%. We will compare these results in detail in the section Results.

In the end, we used an unseen person to obtain the test set. Also, the test set contains images taken from different angles and heights. Here are examples from the dataset (Fig. 3):

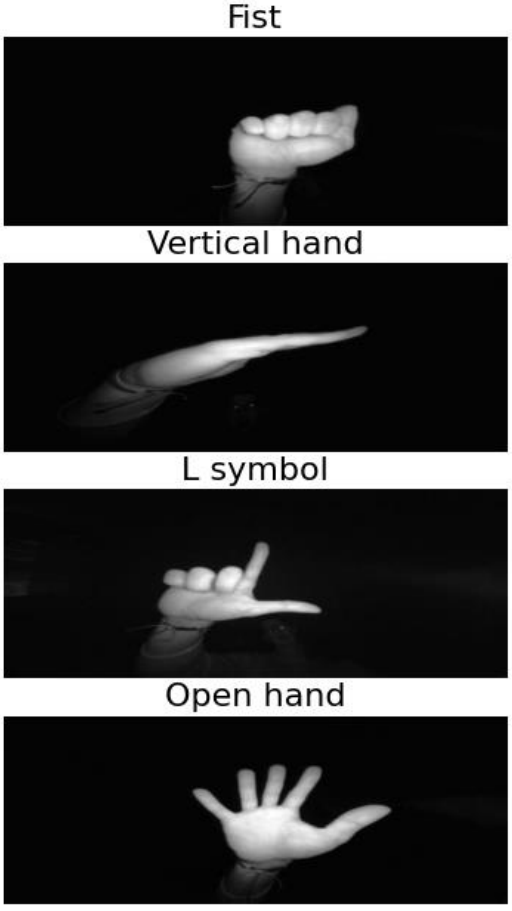

Fig. 3. Example of each hand gesture in our database.

 ISBN 978-86-7466-930-3

C. Model Architecture

VGG16 is a deep convolutional neural network pre-trained on the ImageNet dataset. The VGG16 Architecture was developed and introduced by Karen Simonyan and Andrew Zisserman. The VGG16 model achieved 92.7% accuracy on the classification task, which earned its authors second place in the competition. On the localization task, VGG16 earned first place. This model is widely used both because it is easy to implement and because it is still extremely competitive.

During training, the input to the CNN is a 224 x 224 RGB image. Subtracting the mean RGB value computed on the training set from each pixel is the only pre-processing done here. The image is passed through a stack of convolutional layers, where filters with a small radius are used (Fig. 4). The convolution stride and the spatial padding of convolutional layer input is fixed to 1 pixel. This ensures that the spatial resolution is preserved after the convolution. Five max-pooling layers, which follow some of the convolutional layers, help in spatial pooling. Max-pooling is performed over a 2×2-pixel window, with stride 2. Here is the architecture:

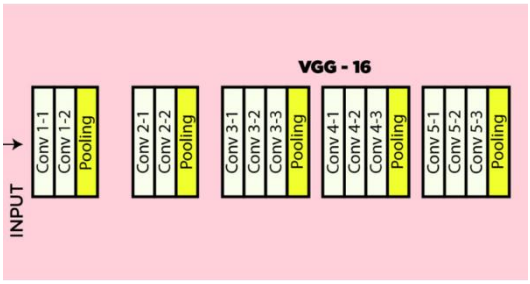

Fig. 4. VGG16 architecture, from [13].

On top of the CNN, we put a classification head that consisted of 2 dense layers. The first one was size 512 with ReLU activation function, and the second one was size 4 with Sigmoid activation function.

D. Hyper-parameters

In this section we will mention some hyper parameters as well as optimizers we used. Firstly, we used an RMS prop optimizer. This is a gradient-based optimizer that deals with vanishing and exploding gradients very well. It uses the moving average of squared gradients to normalize the gradient. This normalization decreases steps for large gradients and increases steps for small gradients. Effectively this means that the learning rate is adjusted based on previous magnitudes of gradients. For the loss function we opted for cross entropy loss.

E. Training

We trained our model during 5 epochs, with batch size 64 and learning rate 0.0001. Train-validation split ratio is 80:20. Another way to stop overfitting of the network was using regularization with L1 and L2 losses combined. Also, we used dropout layer in classification head.

IV. Results

Results of the network which was trained on the small database [11] (the one that did not contain various positions and scales of the hand), has considerably lower accuracy on the versatile test dataset. This can be seen in the confusion matrix below (Fig. 5). The accuracy of this model is 91.5%.

After training our network for five epochs on 20000 images (our database), loss reaches saturation both on validation and training set.

Further, this model has 99.3% accuracy on test data. The confusion matrix is very concentrated around the main diagonal (Fig. 6).

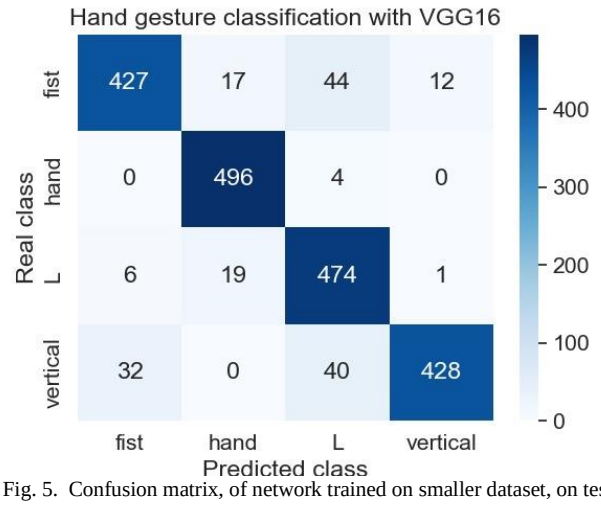

Fig. 5. Confusion matrix, of network trained on smaller dataset, on test data

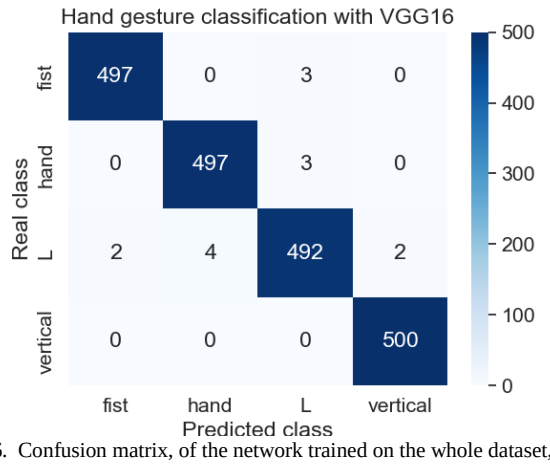

Fig. 6. Confusion matrix, of the network trained on the whole dataset, on test data

We can see that there is no significant difference in these precisions, which suggests that our model is not biased towards any class. This is because our dataset is balanced, and the network can learn the features of each class equally well. However, the L symbol has the highest number of misclassified examples. This could be because it is visually somewhere between the fist and the open hand. Two fingers extended like in the open hand, and three fingers flexed like in the fist

Next, we will show some misclassified instances:

- Fist mistaken for L shape: This could be because of

 ISBN 978-86-7466-930-3

the angle between the forearm and the fist. Therefore, this calls for a more well-rounded database that will contain lots of these examples in the training

- L shape is mistaken for fist: This could be because the index finger is too thin in the image.

In the end, it is important to notice that our system works in real-time. Also, the accuracies are even greater in real-life use because the decision is made using a majority vote from three shots of the subject's hand.

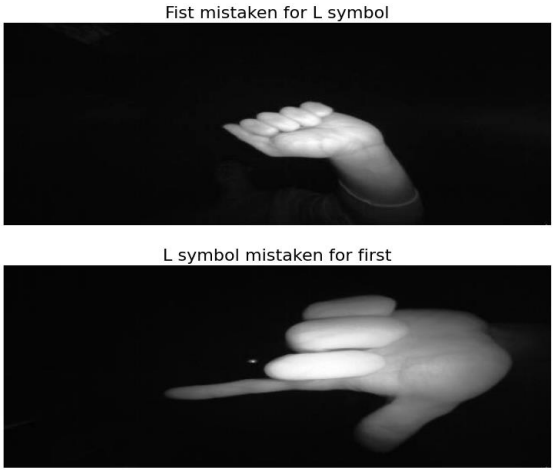

Fig. 7. Examples of failure cases.

V. Conclusion

In this paper, we developed a system for natural and non-invasive human-machine interface using a leap motion sensor. The aim was to classify four hand gesture commands with high precision in real-time. This was done by using deep convolutional network VGG16 and a custom classification head. The inference time per image is 0.23s. To trigger a decision, a hand must be detected by the leap sensor. After that, our system takes three shots at 0.33-second intervals. This setup enables us to use this system in real-time efficiently. The decision is made based on the majority vote of those three shots. This approach gives an accuracy of 99.3% on the test set. Further, our system is robust in terms of distance from the hand to the sensor, and orientation of the hand with respect to the sensor. This is important because in real-life situations workers will approach this device from different angles and heights.

Acknowledgment

This paper was funded by cascading project BrainWatch within Horizon 2020 project Human-Centered Robotics for Connected Factories - SHOP4CF (H2020 grant agreement #873087).

References

[1] Faccio, M., Bottin, M. & Rosati, G. Collaborative and traditional robotic assembly: a comparison model. Int J Adv Manuf Technol 102, 1355–1372 (2019). https://doi.org/10.1007/s00170-018-03247-z

[2] Norberto Pires, J., Godinho, T. and Ferreira, P. (2004), "CAD interface for automatic robot welding programming", Industrial Robot, Vol. 31 No. 1, pp. 71-76. https://doi.org/10.1108/01439910410512028

[3] Bachmann, D., Weichert, F. and Rinkenauer, G. 2015. Evaluation of the leap motion controller as a new contact-free pointing device. Sensors 15, 1, 214

[4] T. Gaspar et al., "Rapid hardware and software reconfiguration in a robotic workcell," 2017 18th International Conference on Advanced Robotics (ICAR), 2017, pp. 229-236, doi: 10.1109/ICAR.2017.8023523.

[5] Marin, G., Dominio, F., and Zanuttigh, P. 2015. Hand gesture recognition with jointlz calibrated leap motion and depth sensor. Multimedia Tools and Applications 1-25.

[6] Shao, L., Hand movement and gesture recognition using Leap Motion Controller.

[7] Sumpeno, S., Dharmayasa, G., Nugroho, S., Purwitasari, D., 2019. Immersive Hand Gesture for Virtual Museum using Leap Motion Sensor Based on K-Nearest Neighbor.

[8] Mutalib, M., Mohd, N., Tomari, M., Sari, S., Ambar, R.,2020. Flying Drone Controller bz Hand Gesture Using Leap Motion.

[9] Xiao, Y., Peng, Q., University of Manitoba, Canada 2017. A hand gesture-based interface for design review using leap motion controller.

[10] McCartney, R., Yuan, J., Bischof, H., Gesture Recognition with Leap Motion Controller.

[11] Lupinetti, K., Ranieri, A., Gianinni, F., Monti, M., 3D dynamic hand gestures recognition using Leap Motion sensor and convolutional neural networks.

[12] https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf , accessed 17.05.2022.

[13] https://www.geeksforgeeks.org/vgg-16-cnn-model/, accessed 17.05.2022.

[14] https://www.researchgate.net/figure/Schematic-view-of-leap-motion-controller-LMC_fig1_266614710, accessed 17.05.2022.

[15] https://www.kaggle.com/gti-upm/leapgestrecog, accessed 17.05.2022.

 ISBN 978-86-7466-930-3