Real-Time Identification of Medical Equipment Using Deep CNN and Computer Vision

Jaya Rubi[1](✉), R. J. Hemalatha[1], and Bethanney Janney[2]

[1] Department of Biomedical Engineering, Vels Institute of Science, Technology and Advanced Studies, Pallavaram, Chennai, India
jayarubiap@gmail.com

[2] Department of Biomedical Engineering, Sathyabama Institute of Science and Technology, Chennai, India

Abstract. Sign language is a way of communication in which hand gestures and symbols are used to connect with each other. Communication provides interaction among people to exchange feelings and ideas. Similarly, when it comes to the handling of medical equipment using a robot, sign language should not be a barrier to carrying out such applications. The purpose of this work is to provide a real-time system that can convert Sign Language (ISL) to text format. Most of the work is based on the handcrafted feature. This paper concentrates on introducing a deep learning approach that can classify the signs using the convolutional neural network. First, we make a classifier model using the signs, then using Kera's implementation of convolutional neural network using python we analyze those signs and identify the surgical tools. Then we process another real-time system that uses skin segmentation to find the Region of Interest in the frame. The segmented region is fed to the classifier model to predict the sign. The predicted sign would gradually identify the surgical tool and convert the sign into text.

Keywords: Deep CNN · surgical equipment · computer vision · kera's implementation · Gesture recognition · Image processing

1 Introduction

Sign language is an important real-time system used amongst several groups of people, especially deaf and dumb people. Sign language or gesture recognition will allow the verbal exchange barrier among exceptional languages like American sign language, Chinese language signal language, Indian signal language, and so forth. The proposed system aims to build a system that can transform these signs with the use of the computer vision technique, further it can be transformed into any language. There are different kinds of studies that focus on building accurate systems and most works are primarily based on pattern recognition. The machines that use simple features and functions aren't always enough in maximum cases and that is the reason this Hybrid approach is delivered to clear up this trouble [8]. However, for a real-time machine, we need quicker strategies

S. Nandan Mohanty et al. (Eds.): ICISML 2022, LNICST 471, pp. 311–319, 2023.
https://doi.org/10.1007/978-3-031-35081-8_25

to solve our problems. Nowadays our computer systems are stepping forward with the speed of processing and the usage of a parallel implementation. Using the Graphics Processing Unit (GPU) system helps solve several issues that may be solved via parallel computing [19, 20]. In the proposed method, we use an area of interest to determine the sign that has been presented in front of the camera, and the analysis and segmentation of this sign are correlated with the medical equipment it is associated with. The proposed proposal has an accuracy of up to 87% and further by increasing the epoch values and training the model the overall output can be improvised and the process of identification can be made much easier.

2 Methodology

Sign language is a complicated system with a critical problem known as background noise. Technique to avoid such issues is to run a convolutional neural network at the images. This method is potentially important to increase the efficiency of the classification and apply it to real-world applications [1, 9]. With the proposed methodology the efficiency of the system was increased to 87% (Fig. 1).

Fig. 1. Image analysis

The basic procedure is followed to give the image as input. The camera easily recognizes a handheld object as the initializing process starts. Features were extracted with the specified program designed in python using Kera's implementation [8]. The next process is the classification and prediction of the object or the hand gesture.

Several studies indicated that most object detection problems used image data sets and bounding box maps to train the models. One of the hard troubles in identification and prediction is marking the bounding box map for each image, and this method is pretty high-priced. Moreover, a region-of-interest predictor is proposed using skin segmentation. The Images were cropped from the segmented bounding regions and sent to a classifier for further prediction [10].

The processing of the image is executed in a completely calculative and step-by-step way to acquire clear results. The capturing of the image by the camera is a vital step. It was noted that the lighting conditions and the readability of the camera are major factors that could affect the result. in addition, the complete processing of the picture has been defined in elements to understand the conversions, classifications, and predictions [20] (Fig. 2).

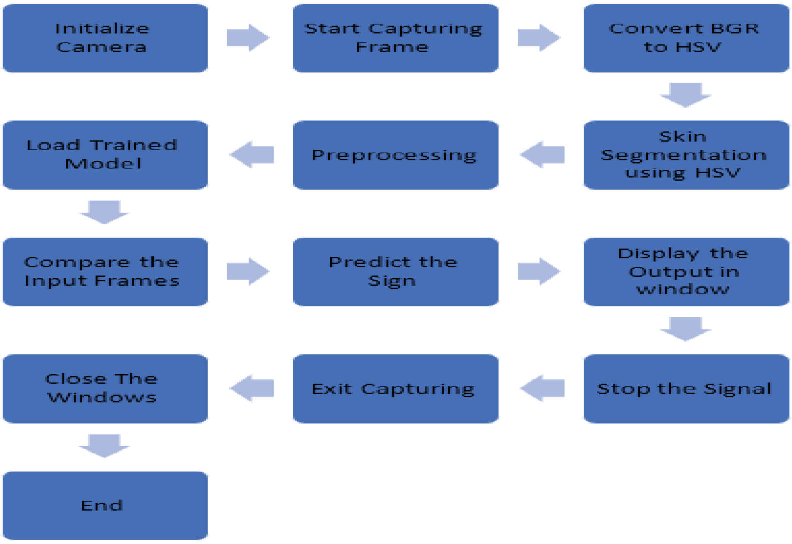

Fig. 2. Gesture recognition process using segmentation and boundary detection

The gesture recognition process has three basic steps that could be broadly classified as image capturing, image segmentation and image processing. The initialization happens by starting the camera. As the camera is initiated within a fraction of a second the camera starts to capture the frame. The data collected initialized the camera to capture frames. The images were converted from hue, saturation, and value (HSV) to BGR. This format helps us to display the image in a clear manner [11]. The next process is the skin segmentation process which further helps in the preprocessing stage. The trained model is loaded into the system which helps us to compare the input frames and prediction of the gestures. Once the prediction is complete the signal is stopped and capturing is closed. The below-mentioned Fig. 3 depicts the flow chart for capturing the image and processing it [19].

2.1 Data Collection

The first step is data collection. The data collection file must run successfully. As the webcam is initialized a frame would be visible in the upper right corner. The gesture is then made visible in front of the camera. Once the gesture is visible on the screen, it can be captured with the click of the ‘c’ key to take a picture. Then the data is automatically saved according to the commands used in the program [12].

2.2 Training Data

The next important step is to train the data. To avoid overfitting, the model, an epoch of 200 is used. The images should be taken from various angles and positions. This would allow the system to work efficiently. The accuracy of the proposed work is displayed

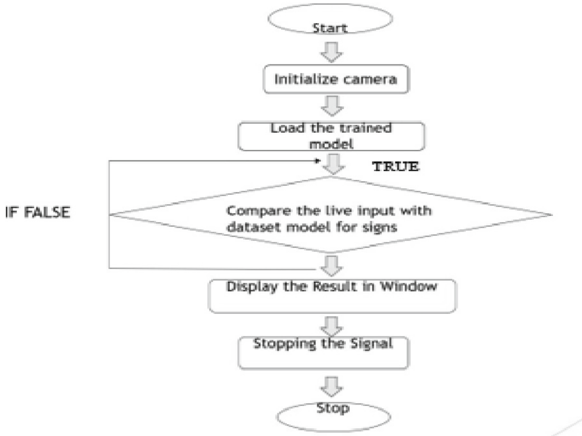

Fig. 3. Flowchart for processing the image

according to the comparison of training and test data present on the map. The recommended use is 350 frames per word for the training data and 250 frames per word for the test data folder [17].

It is also important to build a convolutional neural network on Kera's implementation technique. The comparison between deep learning and machine learning is not very widespread. Deep learning has emerged as a totally popular subset of machine learning due to its high performance on unique sorts of facts. One of the potential methods to categorise deep learning is to classify images to construct a convolutional neural network (CNN) [18]. Python's Kera's library makes it very easy to create a convolutional neural network (CNN). The images are fed to the computer systems, and they perceive these images as pixels. Pixels are mostly associated. Certain bunch or groups of pixels may constitute the edge or boundary region. It might also represent certain patterns in an image [13].

The created dataset is handy to be used in Kera's library. This makes it easy for us to load the dataset. The total variety of images in both training and test files furnished within the dataset offers the entire wide variety of images for training and testing respectively [16].

2.3 Data Preprocessing

The next process is to transform the given datasets into a particular fomat that would be acceptable to the model during the training process. As the format for the training is achieved the model has to be built for processing. The model type that has been selected to use is Sequential. Sequential models are easy to create in Kera [2]. The model can be built layer by layer which helps in analyzing the system with every step.

Each layer can be added to the model using the add() function. These are convolutional layers that process an input image viewed as a two- dimensional matrix. The number of nodes given is 64 for the initial layer and 32 for the next layer in each layer. This wide variety may be set higher or lower depending on the scale of your data set.

The kernel size is the size of the convolution's filter matrix [3]. The activation function used for the first two layers is ReLU or Rectified Linear Activation. This model will then make its prediction based totally on whichever alternative has the best chance [15] (Figs. 4 and 5).

Fig. 4. Training Data Collection gesture 1

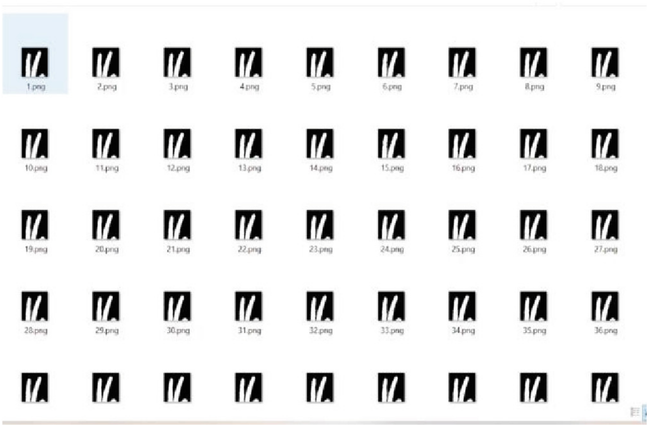

Fig. 5. Training Data Collection gesture 2

2.4 Compiling CNN

The following step is to bring together the complete version by compiling the CNN. Compiling a version requires three parameters: losing, optimizing, and standardizing.

Fig. 6. Training Data Collection gesture 3

The usage of an optimizer takes authority over the rate of learning. SGD is often a correct optimizer in most instances. The SGD optimizer adjusts the rate of learning at some point in the training system. To teach the model, use the fit() characteristic on the model with parameters for the training of data (Train X), goal data (Train y), validation statistics, and several epochs [4]. For validation information, we have used the test set provided within the data set. The range of epochs shows how frequently the model iterates over the records. Over epochs, the version improves to a positive factor. After this factor, the model stops improving in the course of each epoch. For this model, we specify the number of epochs to 20.

The complete process of training is done in Kera which is an open-source software library. This library presents a Python interface for artificial neural networks. Kera also serves as a medium for the TensorFlow data library which helps to train any model for the machine learning process. TensorFlow is a free-of-cost, open-source machine learning software library and it potentially helps the machine to understand the gesture [5]. It can be used for a variety of tasks, with a specific focus on deep neural network training and inference [14].

3 Results and Discussion

The image data were collected and placed in two different folders. One set of data was placed in the test folder and the other in the training folders. Around 350 images for each gesture were collected in the training folder. Around 250 images were collected per gesture for testing purposes. Each shield has folders for both training and testing modes. These directories are created while the code is running. As far as the program is concerned, the first thing to do is record capture, as shown in Fig. 6.

Here the image is captured and saved in a separate file. As the amount of testing and training data changes, the encoding needs to be changed. This process helps to keep the training model intact (Fig. 9).

Here the image is captured and saved in a separate file. As the amount of testing and training data changes, the encoding needs to be changed. This process helps to keep

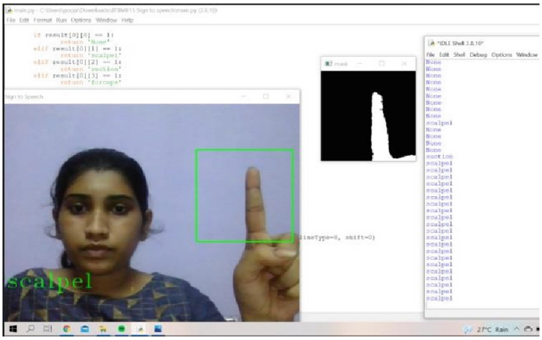

Fig. 7. Gesture 1 depicting “SCALPEL”

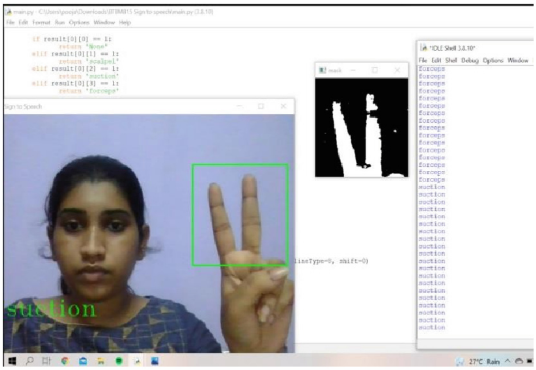

Fig. 8. Gesture 2 depicting “SUCTION”

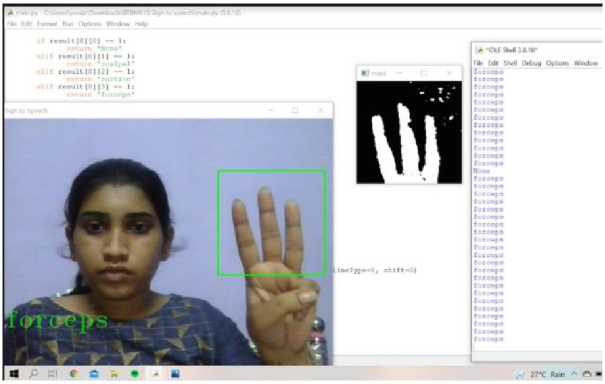

Fig. 9. Gesture 3 depicting “FORCEPS”

the training model intact. The loss function used is the categorical cross-entropy for configurations of three or more classes. As the above figures depict the association of each gesture with a surgical tool, it is very clear that the model has been trained well using a deep convolutional neural network. Figure 6. Depicts the gesture 1 and represents the corresponding surgical tool associated with it as “scalpel”, similarly Fig. 7 depicts gesture number 2 which will represent an equipment called “suction “device. Lastly Fig. 8 depicts gesture 3 and it is represented as the third tool called the “forceps”. Thus, as we train the model and kept testing the epoch values, the efficiency of the model as well as the accuracy increased [6]. The representation of surgical tools along with gestures has numerous applications. Robots can be trained using the model to identify surgical tools and assist doctors in surgeries [7, 8].

4 Conclusion

This paper provides valuable information on why sign language interpreting should be included in machine learning models to benefit the future of robotic surgery. A complete model involves collecting, training, and testing data that was used to make final predictions. The most important requirements were convolutional neural networks and TensorFlow. The result is received as a text for the input which is an icon image/gesture. So, in a very straightforward and simple way, a sign language interpreter for three gestures was designed and programmed. Future research areas may involve creating user interfaces with robotic arms using sign language interpreters for manipulation, double-hand gestures, etc. [11, 13]. This model can be used for practicing and mastering robotic arms for robotic manipulation.

References

1. Zhao, K., Zhang, K., Zhai, Yu., Wang, D., Su, J.: Real-time sign language recognition based on video stream. In: The National Natural Science Foundation (NNSF) of China, 10 September 2020 UTC from IEEE Xplore
2. Panda, A.K., Chakravarty, R., Moulik, S.: Hand gesture recognition using flex sensor and machine learning algorithms. In: 2020 IEEE- EMBS Conference on Biomedical Engineering and Science(2020)
3. Hamza, K.H., Zhang, X., Mu, X., Odekhe, R., Alhassan, A.B.: In: Engineering 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) (2018)
4. Jin, J., Chung, W.: School of Mechanical Engineering, Korea University, Seoul 02841 Korea. Sensors **19**(2), 289 (2019). https://doi.org/10.3390/s19020289
5. Zhao, D., Yang, J., Koye, M.O., Wang, S.: A novel non-contact abnormal gait recognition approach based on extended set membership filter. IEEE Access **7**, 76471–76753 (2019)
6. Young, K.Y., Cheng, S.L., Ko, C.H., Tsou, H.W.: Development of a comfort-based motion guidance system for a robot walking helper. J. Intell. Robot. Syst. **100**(2), 379–388 (2020)
7. Ahmed, S.F. et al.: Mobility assistance robot for disabled persons using electromyography (EMG) sensor. In: 2018 IEEEInternational Conference on Innovative Research and Development (ICIRD), pp. 1–5 (2018). https://doi.org/10.1109/ICIRD.2018.8376304
8. Roizenblatt, M., Edwards, T.L., Gehlbach, P.L.: Robot-assisted vitreoretinal surgery: current perspectives. Dove Press J. (2018)

9. Meello, R., Jimenez, M., Souza, F., Ribeiro, M.R., Frizero-Neto, A.: Towards a new generation of smart devices for mobility assistance: cloud walker, a cloud-enabled cyber-physical system. In: 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob) (2018)
10. Miseikis, J., et al.: Lio-a personal robot assistant for human-robot interaction and care application. Published on PMC (2020)
11. Sahoo, A.K., Brahma, B., Pattanaik, A.: Design & development of robotic arm for medical surgery. In: 2nd International Conference on Power and Embedded Drive Control (ICPEDC) (2019)
12. Zhao, X., et al.: A smart robotic walker with intelligent close-proximity interaction capabilities for elderly mobility safety. Front. Neurobot. **14**, 575889 (2020)
13. Ishak, M.K., Kit, N.M.: Design and implementation of robot-assisted surgery based on internet of things (IoT). In: International Conference on Advanced Computing and Applications (ACOMP) (2017)
14. Sagitov, A., Gavrilova, L., Soy, T.T., Li, H.: Design of simple one arm- surgical robot for minimally invasive surgery. In: 12th International Conference on Development in E-System Engineering (DeSE) (2019)
15. Miao, Y., Jiang, Y., Muhammad, G.: Telesurgery robot based on 5G tactile internet. Mob. Netw. Appl. **23**, 1645–1654 (2018)
16. de Smet, M.D., Gerrit, J.L, Faridpooya, K., Mura, M.: Robotic-assisted surgery in ophthalmology. Curr. Opin. Ophthalmol. **29**(3), 248-253 (2018)
17. Jason, D., Wright, M.D.: Robotic-assisted surgery (balancing evidence and implementation). Jama, **318**(16), 1545-1547
18. Jin, J., Chung, W.: Obstacle avoidance of two-wheel differential robots considering the uncertainty of robot motion on the basis of encodes odometry information. Sensors, **19**(2), 289 (2019)
19. Hamza, K.K., Zhang, X., Mu, X., Odekhe, R., Alhassan, A.B.: Modeling and stimulation of transporting elderly posture of multifunctional elderly- assistance and walking-assistant robot. In: 8th International Conference on CYBER Technology in Automation Control and Intelligent Systems (CYBER) (2018)
20. Ahmed, S.F., et al.: Mobility assistance robot for disabled persons using electromyography sensor. In: IEEE International Conference on Innovative Research and Development (ICIRD) (2018)