



Hand Gesture Recognition Based on the Fusion of Visual and Touch Sensing Data

F. T. Timbane, S. Du^(✉), and R. Aylward

Department of Electrical Engineering, Tshwane University of Technology,
Gauteng, Pretoria, South Africa
203900783@tut4life.ac.za, dushengzhi@gmail.com,
AylwardRC@tut.ac.za

Abstract. The use of computers has evolved so rapidly that our daily lives revolve around it. With the advancement of computer science and technology, the interaction between humans and computers is not limited to mice and keyboards. The whole-body interaction is the trend supported by the newest techniques. Hand gesture becomes more and more common, however, is challenged by lighting conditions, limited hand movements, and the occlusion of the hand images. The objective of this paper is to reduce those challenges by fusing vision and touch sensing data to accommodate the requirements of advanced human-computer interaction. In the development of this system, vision and touchpad sensing data were used to detect the fingertips using machine learning. The fingertips detection results were fused by a K-nearest neighbor classifier to form the proposed hybrid hand gesture recognition system. The classifier is then trained to classify four hand gestures. The classifier was tested in three different scenarios with static, slow motion, and fast movement of the hand. The overall performance of the system on both static and slow-moving hand are 100% precision for both training and testing sets, and 0% false-positive rate. In the fast-moving hand scenario, the system got a 95.25% accuracy, 94.59% precision, 96% recall, and 5.41% false-positive rate. Finally, using the proposed classifier, a real-time, simple, accurate, reliable, and cost-effective system was realised to control the Windows media player. The outcome of fusing the two input sensors offered better precision and recall performance of the system.

Keywords: Static hand gesture · Hybrid hand gesture recognition · Human-computer interface · Human-machine interface · Touchpad

1 Introduction

Computers have evolved rapidly and are used to connect people around the world. Individuals, organisations, industries, and schools relies on computers to function in terms of communications, to provide services, production, and education. In many ways, it has become an integral part of our lives; in fact, people can undeniably use them to do almost anything they want to do. In the past, the only way one could interact with a computer

was using devices such mouse, keyboard, and joystick [1]. With the development of technology and computer vision, those traditional computer interactions are no longer satisfactory for applications such as interactive games and virtual reality. Much research has been conducted on how humans can interact with computers. Hence, Human-Computer Interaction (HCI) has become a trend of technological evolution towards making life more convenient and easier.

Gesture recognition is one of the approaches employed to focus on accommodating the advanced application of HCI. Also, in circumstances where a person's hands cannot touch, such as in a medical environment, in the industry where there is too much noise, or when giving commands in a military operation, gesture recognition is a solution that can be applied [2, 3]. Gesture recognition became one of the very important fields of research for HCI. It provides the basis for recognising the body, head, and hand movement or posture, and facial expression. This research study mainly focused on hand gesture recognition. Hand gestures offer a means to expressively interact between people using hand postures and dynamic hand movements for communication [4, 5]. Hand gesture recognition is popularly applied to the recognition of sign language [6].

There are various technologies for hand gesture recognition such as vision-based [7] and instrument-based hand gesture recognition [8]. A vision-based hand gesture has been the most common and natural way for people to interact and communicate with one another [9]. Vision-based hand gesture recognition is one of the most challenging tasks in computer vision and pattern recognition. It uses image acquisition, image segmentation, feature extraction, and classification methodology to give the machine the ability to detect and recognise human gestures to convey certain information and to control devices [10]. There are different types of vision-based approaches such as the 3D model, the colour glove, and appearance-based hand gesture recognition [9].

The existing vision-based hand gesture recognition systems have inherent challenges such as lighting conditions, and occlusion, which limit the reliability and accuracy of the systems [11]. The instrument-based approach is the only technology that satisfies the advanced requirements of a hand gesture system [12]. However, this method consists of sensors attached to the glove, which convert the flexion of the fingers into an electrical signal for the determination of the hand gesture. There are several disadvantages that render this technology unpopular. The main concerns include the loss of the naturalness when sensors cables are attached to the hand, rendering it inconvenient, a lot of hardware required, and costly.

The above-mentioned information motivated us to also play a part in solving the challenges faced by the existing vision-based hand gesture recognition. This research aims to design and implement a real-time vision-based hand gesture recognition, by fusing vision and the touch sensing data with the hope of achieving a low cost, accurate and reliable system. The two approaches were developed and tested separately, then combined to realise the proposed system. A machine-learning algorithm was used to train and test the system. Four (Play, Pause, Continue, and Stop) gestures were trained to control a Windows media player for demonstration purposes. The outcome of fusing the two input sensors offered a better precision and recall performance of the hand gesture recognition system.

This system can be very helpful to physically impaired people because they can define gestures and train them according to their needs. The hand gesture recognition system can also improve the HMI to the joystick dependent system to allow people who cannot grab with their hands to be able to operate those systems. Furthermore, the system can also use only the touchpad to recognise gestures during bad lighting conditions or occlusion on the vision side.

2 Related Works

2.1 Hand Gesture Recognition Systems

Parul et al. [13] got 82% of success by using three axial accelerometers to monitor the orientation of the hand. The results are displayed in an application that also converts text to speech to enable those who cannot see or hear. Rikem et al. [14] presented the Micro-Electromechanical System (MEMS) accelerometer-based embedded system for gesture recognition. The overall recognition rate for both modes was 98.5%.

Rohit et al. [15] proposed a data glove-based system to interpret hand gestures for disabled people and to convert those gestures to a meaningful message displayed them on a screen. The approach used IR sensors attached to the glove to translate finger and hand movements to words. Kammari et al. [16] proposed a gesture recognition system using a surface electromyography sensor. The results suggest that the system responds to every gesture on the mobile phone within 300 ms, with an average user-dependent testing accuracy of 95.0% and in user-independent testing 89%. The instrument-based gesture recognition system is one of the best approaches that produce good results on hand gesture recognition. However, the sensors that are used in this approach offer some drawbacks; such as computational expensive, costly, and the wearing of sensors makes people lose their naturalness.

Gotkar et al. [17] designed a dynamic hand gesture recognition for Indian Sign Language (ISL) words to accommodate hearing-impaired people. The results indicated an average accuracy of 89.25%. Laskar et al. [18] proposed the stereo vision-based hand gesture recognition under the 3D environment to recognise the forward and backward movement of the hand as well as the 0-9 Arabic numerals. The results indicated that an average accuracy of 88% of the proposed method for detecting Arabic numerals and forward/backward movements.

2.2 Classifiers for Hand Gesture Recognition

Shroffe et al. [19] presented a hand gesture recognition technique based on electromyography (EMG) signals to classify pre-defined gestures using the artificial neural network (ANN) as the classifier. The average success rate of the overall training was found to be 83.5%. Sharmar et al. [20] developed the handwritten digit recognition based on the neural network to control examination scores in the paper-based test. The proposed method based on the neural network yielded about 90% accuracy of the results. Chethenas et al. [21] presented the design of a static gesture recognition system using a vision-based approach to recognise hand gestures in real-time. The approach followed three stages:

image acquisition, feature extraction, and recognition. The K-curvature algorithm is applied to the proposed system.

Senthamizh et al. [22] presented a real-time human face detection and face recognition. The approach implements the Haar Cascade algorithm for recognition, which was organised by an OpenCV using Python language. The results indicated that 90% to 98% of the recognition rate was achieved. The results varied because of distance, camera resolution, and lighting conditions.

Goel et al. [23] conducted a comparison study of KNN and SVM algorithms. The system recognises 26 letters of the alphabet. It was shown that the KNN classifies data based on the distance metrics, while the SVM needs a proper training stage. The KNN and SVM are used as multi-class classifiers in which binary data belonging to either class is segregated. The comparison results for the two classifiers are shown in Table 1.

Table 1. Comparison between SVM and KNN [22]

Classifier	Training set	Test set	Accuracy rate (%)
SVM	10 000	10 000	98.9
KNN	10 000	10 000	96.47

The results and observations, however, indicate that the SVM is more reliable than the KNN. However, the KNN is less computational than the SVM.

3 Methodology

This section describes the approaches and procedures that are followed to develop and test the prototype. The approach followed by the stages is depicted in Fig. 1.

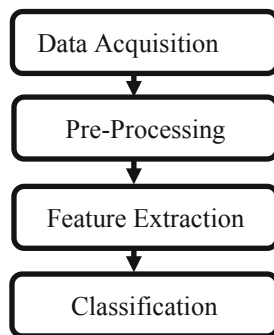


Fig. 1. Flow diagram of hand gesture approach

Data acquisition enables the acquisition of images or video frames from a webcam into MATLAB. In this system, a Logitech webcam is used to capture 2D images and save them in jpg format.

Pre-processing can be defined as the method to eliminate the unnecessary elements of noise in an image. The primary objective of the pre-processing procedure is to transform data into a form that can be more effectively and easily processed by the stages that follows.

The extraction of features is a stage after pre-processing where the success of the gesture recognition system depends on it. The feature extraction stage gives the gesture recognition system the most effective features. An image analysis feature extraction element seeks to identify the inherent features or characteristics of objects found within an image. Upon extracting the correct features from the image and choosing a suitable data set, a machine learning technique can be used to identify the gesture.

Classification is the mechanism of discrimination between different classes. KNN algorithm is used on our proposed system to train and classify different gestures. The KNN method is illustrated in Fig. 2.

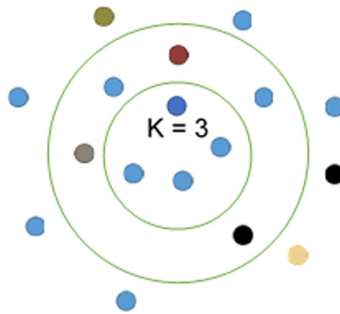


Fig. 2. KNN classification

3.1 Vision-Based Approach

A vision-based hand gesture recognition approach consists of one or more cameras to capture gestures that are then interpreted by a computer using vision techniques. The detailed method of the proposed vision-based system is shown in Fig. 3.

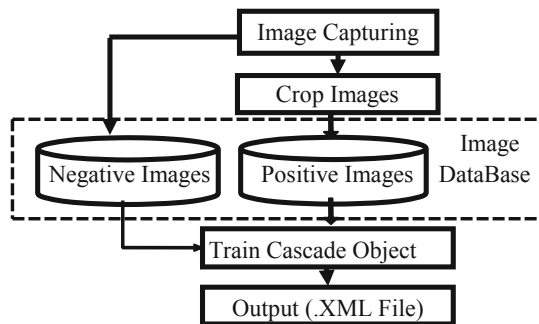


Fig. 3. Flow diagram of vision based approach

The images are captured using an 8 Megapixel Logitech USB webcam. All images containing the ROI (positive images) with different angles and sizes were cropped to form a positive image database. The image database consists of a positive image folder with cropped images and a negative image folder, where 8 000 positive cropped images are stored in the positive image folder and 20 000 negative images are stored in a negative image folder. TrainCascadeObjectDetector is a MATLAB function that is used to train the object detector. An object detection algorithm typically takes a raw image as an input and outputs the most important features in the form of an XML file.

3.2 Touch-Based Approach

This approach follows the mathematical morphology for image segmentation for recognising the touched surface. The analysis of image characteristics and the detection of the touched areas will be done using the morphological method. Figure 4 shows the flow diagram of the touch-based approach.

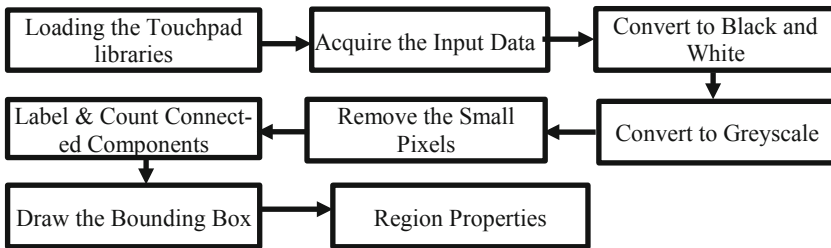


Fig. 4. Flow diagram of touch-based approach

The first step of processing data using the touchpad with MATLAB is to import the touchpad necessary libraries into the MATLAB directory, as well as establish communication between the touchpad and MATLAB. Image acquisition in this situation is the process of applying a touch to the touchpad and output the image in a matrix format. When the image is converted to greyscale, the low-intensity values in an image are displayed as black, whereas the high-intensity values will be displayed as white. The intensity of black and white or binary ranges from 0 to 1. All connected components with pixels below 100 pixels have been removed from the binary image in this study, producing the new binary image with pixels greater than 100 pixels. Connected components are a set of pixels connected or belonging to the same class in a binary image. Connected components labeling is the method of defining the group of pixels connected in a binary image and assigning each one a unique label. The properties that have been considered in this case are the Area and Centroid. The bounding box is an imaginary rectangle that encloses every image and is always parallel to the axes on its sides.

3.3 Fused System

The KNN algorithm was used for classification to combine the two-sensing data. Figure 5 represents the flow diagram of the fused system. The system is divided into phases of

training and testing; Sect. 4 discusses the test phase. During the training phase, the two-sensing data will receive gestures simultaneously at the input. The gesture goes through different stages to output two bounding boxes. The two bounding boxes are then connected, trained using the KNN algorithm, and saved as trained samples.

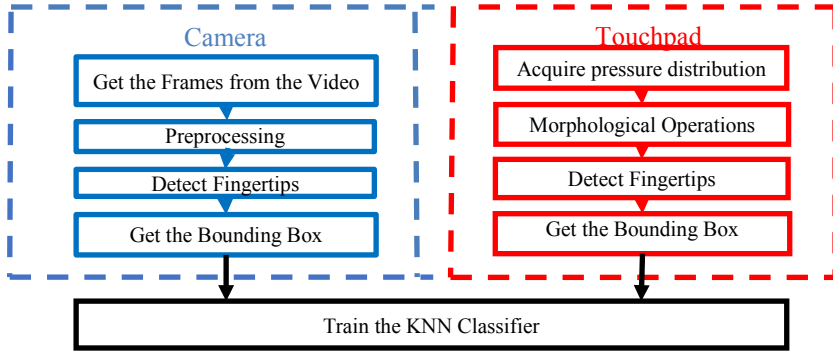


Fig. 5. Flow diagram of the Fused System

The trained samples are saved through a class ID and sample features. Choosing k is one of the most important factors when training the KNN algorithm. The gesture is defined by the number of bounding boxes captured by both sensing data. The class ID is then assigned to the extracted features from each gesture and forms the feature vector. For each gesture, the features are extracted, so that any gesture will have several feature vectors depending on the number of bounding boxes that the gesture has. Therefore, each gesture will have a unique number of feature vectors. In this study, only four gestures were trained as per Fig. 6; therefore, the data set consisted of 400 samples, where each sample had a class of either Play, Pause, Resume, or Stop. Every class had 100 trained samples.

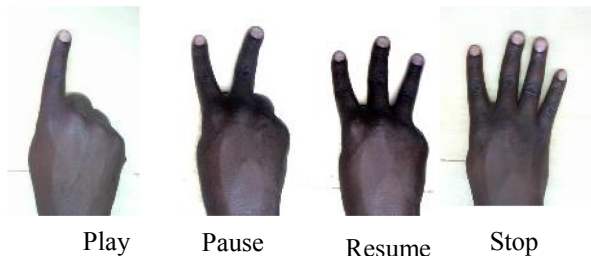


Fig. 6. Controlling gestures

4 Results and Analysis

The system was tested to verify the results, where real-time data from both sensing devices as input were applied. The performance of the system was based on its ability

to recognize the input gesture correctly [24]. The metric that was used to accomplish this is called the recognition rate. Each gesture recognition result was evaluated based on the following equations:

$$\text{Accuracy} = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{1}$$

$$\text{TPR} = \frac{TP}{(TP + FN)} \tag{2}$$

$$\text{FPR} = \frac{FP}{(FP + TP)} \tag{3}$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{4}$$

The system was tested in three different scenarios: static, slow, and fast movement of the hand. The slow and fast-moving hand speed was calculated based on the size of the touchpad (420 × 310 pixels) since the finger was swiped from one end to another end of the touchpad. The calculations were conducted as follows: During the slow-motion, it took 2 s from one end of the touchpad to another end, resulting in a speed of 210 pixels per second. The movement was about 12.7 pixels between two frames. During the fast motion, it took 1 s from one end of the touchpad to another end, which resulted in a speed of 420 pixels per second. The movement was about 19.3 pixels between frames.

The tests were done on both the individual systems (Vision and Touchpad) and fused system, testing three scenarios: static, slow motion, and fast movement of the hand, using the above-mentioned equations. Table 2 shows the comparative performance on static and slow scenarios for the four gestures on vision, touchpad, and fused gesture recognition systems. Due to the size of the touchpad, only two gestures were checked

Table 2. The Performance comparison of vision, touchpad, and fusion methods

Static												
	Accuracy (%)			Precision (%)			TPR recall (%)			FPR (%)		
	Vision	Touch	Proposed	Vision	Touch	Proposed	Vision	Touch	Proposed	Vision	Touch	Proposed
Play	100	100	100	100	100	100	100	100	100	0	0	0
Pause	100	99.94	100	100	100	100	100	99	100	0	0	0
Resume	100	100	100	100	100	100	100	100	100	0	0	0
Stop	100	100	100	100	100	100	100	100	100	0	0	0
Slow motion												
Play	100	100	100	100	100	100	100	100	100	0	0	0
Pause	100	100	100	100	100	100	100	100	100	0	0	0
Resume	100	100	100	100	100	100	100	100	100	0	0	0
Stop	100	100	100	100	100	100	100	100	100	0	0	0
Fast motion												
Play	88.5	95.75	96.5	81.30	95.75	96.04	100	90	97	18.75	4.26	3.96
Pause	78.5	91.58	94	73.95	91.58	93.14	88	87	95	26.05	8.42	6.87

on fast scenarios. The findings in the table indicate that in all scenarios all systems performed very well in terms of the accuracies, precisions, recalls, and false-positive rates. It means that since the precision and recall performed so well and balanced, the systems are reliable.

The bar graphs in Fig. 7 shows the average recognition performance on static and slow-moving gesture scenarios for the four gestures on vision, touchpad, and fused systems. For the four movements in three scenarios, the bar graphs represent the average accuracy, precision, recall, and false-positive rate of recognition. For both static and slow-moving gesture scenarios, the vision-based system achieved 100% performance on precision, accuracy, recall, and 0% false-positive rate.

The vision-based system achieved an 83.5% accuracy, 77.63% precision, 94% recall, and 22.8% false-positive rate for the fast-moving gesture scenario. The decrease of the performance for the fast-moving gesture scenario is believed to be related to the poor image quality, for instance, the impact of blurring when objects move fast.

The touch-based system obtained 100% on the accuracy, precision, recall, and 0% on false-positive rates of slow-moving gesture scenarios. For the fast-moving gesture scenario, the touch-based system obtained 95.25% accuracy, 94.59% precision, 96%, recall, and 5.41% false-positive rate performance.

The proposed fusion system achieved 100% performance in both static and slow-moving gesture scenarios on the accuracy, precision, recall, and 0% false-positive rate. In the fast-moving gesture scenario, the proposed system achieved 95.25% accuracy, 94.59% precision, 96% recall, and 5.41% false-positive rate.

It can be seen that all scenarios, all systems performed very well in terms of accuracy, precisions, recalls, and false-positive rates, however, the fused system performs better than the individual systems on the fast-moving gesture. This shows that the fused system has enhanced the performance of vision and touchpad systems.



Fig. 7. The average results for vision, touchpad and fused systems

5 Conclusion and Future Work

To overcome the challenges faced by a vision-based hand gesture recognition system, this research developed a prototype for a hybrid hand gesture recognition system by fusing heterogeneous sensing data and video. The objective was achieved by implementing the vision-based method and interpretations of the touch data provided by a touchpad. The system used a KNN classifier to trained and classify four hand gestures to control the Windows media player. The outcome of the fused system offered better precision and recall performance. For future work, more attempts could be made to improve the model. Multiple hands for gesture recognition need to be considered. Furthermore, the method developed in this research will be extended to a dynamic hand gesture recognition which is more acceptable to human operators.

References

1. Shah, N., Patel, J.: Gesture recognition technique: a review. *Int. J. Recent Trends Eng. Res.* **3**(4), 550–558 (2017)
2. Rao, S., Rajasekhar, C.H.: Password-based gesture controlled robot. *Int. J. Eng. Res. Appl.* **6**(4), 63–69 (2016)
3. Prabhu, R.R., Sreevidya, R.: Design of robotic arm based on hand gesture control system using wireless sensor networks. *Int. Res. J. Eng. Technol.* **4**(3), 617–621 (2017)
4. Liu, K., Chen, C., Jafari, R., Kehtarnavaz, N.: Fusion of inertia and depth sensor data for robust hand gesture recognition. *IEEE Sens. J.* **14**(6), 1898–1903 (2014)
5. Zhang, Q., Lu, J., Wei, H., Zhang, M., Duan, H.: Dynamic hand gesture segmentation method based on unequal-probabilities background difference and improved fcm algorithm. *Int. J. Innov. Comput. Inform. Control* **11**(5), 1823–1834 (2015)
6. Malik, M., Vishnoi, K.: Gesture recognition technology: a comprehensive review of its application and future prospects. In: 4th International Conference on System Modelling and Advancement in Research Trends College of Computing Science and Information Technology, pp 355 – 361 (2015)
7. Joshi, M., Patil, S.: Vision-based gesture recognition system – a survey. *Int. J. Appl. Innov. Eng. Manage.* **3**(5), 321–324 (2014)
8. Yuvaraju, M., Priyanka, R.: Flex sensor based gesture control wheelchair for stroke and SCI patients. *Int. J. Eng. Sci. Res. Technol.* **6**(5), 543–549 (2017)
9. Itkarkar, R.R., Nandy, A.K.: A study of vision-based hand gesture recognition for human machine interface. *Int. J. Innov. Res. Adv. Eng.* **1**(12), 48–52 (2014)
10. Kumar, S., Balyan, A., Chawla, M.: Object detection and recognition in images. *Int. J. Eng. Dev. Res.* **5**(4), 1029–1034 (2017)
11. Meshram, A.P., Rojatar, D.V.: Gesture recognition technology. *J. Eng. Technol. Innov. Res.* **4**(1), 135–138 (2017)
12. Krishmaraj, N., Kavitha, M.G., Jayasankar, T., Kumar, K.V.: A glove based approach to recognize indian sign language. *Int. J. Recent Technol. Eng.* **7**(6), 1419–1425 (2019)
13. Parul, W., Sanjana, K.K., Sushmitha, M.A., Suraksha, C.: Sign language recognition using a smart hand device with sensor combination. *Int. J. Res. Appl. Sci. Eng. Technol.* **6**(4), 4507–4511 (2018)
14. Riken, M., Ponnammal, P.: MEMS accelerometer based 3D mouse and handwritten recognition system. *Int. J. Innov. Res. Comput. Commun. Eng.* **2**(3), 3333–3339 (2014)

15. Rohit, H.R., Gowthman, S., Sharath, C.A.S.: Hand gesture recognition in real-time using IR sensor. *Int. J. Pure Appl. Math.* **114**(7), 111–121 (2017)
16. Kammari, R., Basha, S.M.: A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *Int. J. Innov. Technol.* **3**(8), 1412–1417 (2015)
17. Ghotkar, A., Vidap, P., Deo, K.: Dynamic hand gesture recognition using hidden markov model by microsoft kinect sensor. *Int. J. Comput. Appl.* **150**(5), 5–9 (2016)
18. Lskar, M.A., Das, A.J., Talukdar, A.K., Kumar, K.: “Stereo vision-based hand gesture recognition under 3D environment”, second international symposium on computer vision and the internet. *Procedia Comput. Sci.* **58**, 194–201 (2015)
19. Shroffe, E.H.D., Manimegalai, P.: Hand gesture recognition based on EMG signal using ANN. *Int. J. Comput. Appl.* **3**(2), 31–39 (2013)
20. Sharma, A., Barole, Y., Kerhalkar, K., Prabhu, K.R.: Neural network-based handwritten digit recognition for managing examination score in paper based test. *Int. J. Adv. Res. Electric. Electron. Instrum. Eng.* **5**(3), 1682–1685 (2016)
21. Chethanas, N.S., Divya, P., Kurian, M.Z.: Static hand gesture recognition system for device control. *Int. J. Electric. Electron. Data Commun.* **3**(4), 27–29 (2015)
22. Senthamizh, S.R., Sivakumar, D., Sandhya, J.S., Ramya, S., Kanaga, S., Rajs, S.: Face recognition using haar – cascade classifier for criminal identification. *Int. J. Recent Technol. Eng.* **7**(6S5), 1871–1876 (2019)
23. Goel, A., Mahajan, S.: Comparison: KNN & SVM algorithm. *Int. J. Res. Appl. Sci. Eng. Technol.* **5**(12), 165–168 (2017)
24. Kaur, J., Kaur, P.: Shape-based object detection in digital images. *Int. J. Res. Appl. Sci. Eng. Technol.* **5**(12), 332–339 (2017)