SemantJournals American Journal of Technology Advancement 2997-9382, Vol 03, No. 03, 2026

Article

Real Time Hand Gesture-Based Light Control System Using Computer Vision and Machine Learning

Taha Abdulwahid Mahmood
Department of Electrical Engineering, College of Engineering, University of Kirkuk, Kirkuk, Iraq
Email: taha-abdulwahid@uokirkuk.edu.iq

Abstract: In this paper, we aim to develop a full-stack system to control LEDs based on hand gestures using computer vision and machine learning methods. The system recognizes finger movements using the MediaPipe library, classifies the number of raised fingers based on the KNN (K-Nearest Neighbors) and transmits the result information to the Arduino board for the use of the information. A database of finger poses taken in different light and filming angles was collected, this data set was randomly split into 80% train and 20% test. Different values of the number of neighbors (k) have been tested, and the accuracy obtained is 92% for k = 5. The operation had more efficiency in larger hands with the mean response time of 0.15 seconds, which was acceptable for real-time interaction.

Keywords: KNN, Arduino, MediaPipe, computer vision, machine learning, and human-computer interaction.

1. Introduction

Computer interaction (HCI) is a core discipline whose needs evolve over time, but where recently developments in computer vision and machine learning are increasingly informing better interactions and access to technology [1]. With the growing pervasiveness of smartphones in our everyday lives, more intuitive and interactive user interfaces are in demand [2]. In this work, the paper describes a complete system for managing lighting based on gesture control which breaks into new areas within home automation and assistive technologies. The system works based on finger movement tracking, for which the MediaPipe library is used to obtain high-level accuracy in tracking hands. Raised finger number is recognized by using K-Nearest Neighbors (KNN) algorithm that is trained to output which category that a given input data belongs to from training data. An all-inclusive dataset was captured that includes different finger poses in different lighting conditions and under various shooting angles, improving the reliability of the system in real-life operation environments.

According to the experimental results, (k=5) was the optimal value of neighbor count, that according to which the classification accuracy reached up to 92%. The performance of the system with respect to different conditions, including the size of hand, the shooting angles and lighting effects was investigated and it was found that it works better with a bigger hand. The response time was in the order of 0.15 s thus allowing real-time interaction. In addition, this study is not limited to a lighting control system and it is a step towards an implementation to a wide application field. These applications are relevant to home automation (users can be able to control domestic appliances without touching them), assistive technology (which help disabled individuals in day-to-day life) and wireless control systems increasing productivity in industrial settings. Precise matching between computer vision and machine learning technologies and their end applications in improving quality of life and coming up with novel solutions for everyday problems will be suggested through our study. We also anticipate the results from this study will motivate continued research and progress in the area, which will ultimately lead to a more natural and efficient human experience with technology.

Literature Review

Numerous studies have addressed gesture recognition technologies, utilizing a range of algorithms such as artificial neural networks, tree-based algorithms, and KNN. Research by Waqas et al., demonstrated the effectiveness of KNN in classifying gestures with high accuracy. However, most previous studies did not integrate real-time image processing with physical device interaction, highlighting a gap in the existing research [3].

J. S. Vimali et al., The article proposes creating a less expensive version of a hand gesture controller using an Arduino board. A pair of ultrasonic sensors are used to provide input. The sensors are utilized to calculate changes in the distance of our hand movements when we make the gesture. The Arduino will calculate the change in distance and recognize the user's activity. The Arduino will request that the system CPU do the action specified by the user. The proposed paradigm reduces the time required to conduct fundamental actions such as open, minimize, maximum, play, and save [4].

Marthed Wameed and Ahmed M Alkamachi, in this study, the proposed system has two parts: The first step is to recognize and classify hand motions in real time using computer vision technology; this is accomplished by machine learning, specifically the Media Pipe algorithm. The MediaPipe algorithm is divided into three sections: the first detects the palm of the hand, the second identifies 21 3D points on the palm, and the third classifies hand motions by comparing the dimensions of those points. The second portion, which is dependent on the first, states that after detecting and classifying hand signals, the system controls the robot using hand gestures, as each hand gesture corresponds to a certain action that the robot does. The experimental results demonstrated that, depending on environmental factors such as light intensity, distance, and tilt angle (between hand gesture and camera), the suggested method may effectively control the robot's movement via hand gestures [5].

Nandhini Priya R, in this project, we implemented a simple Arduous-based hand gesture control that allows you to control a few web browser functions such as switching between tabs, scrolling up and down in web pages, shifting between tasks (applications), playing or pausing a video, and increasing or decreasing the volume (in VLC Player) using hand gestures. In conclusion, everyone requires this hand Gesture tool to operate their desktop/laptop because it is simple and comfortable to use [6].

Dr. Sunil Chavan et al., this study describes the design, operation, and successful testing of a rover operated wirelessly with hand gestures. A vision-based hand gesture-based solution is proposed for giving real-time control of an Arduino-based robot. Combining all of these ideas into a single gadget capable of recognizing hand motions and wirelessly transmitting data to other devices for surveillance and other applications [7].

Divyanshu Jeena et al., in our gesture control project, a two-wheel robot is controlled using a variety of hand gestures. Gesture recognition, which recognizes picture signals, is achieved through image processing using several methods. Robots can assume any form, but some have the look of a living human. These robots interact with humans directly, creating a user-friendly interface. The early systems were primarily employed for robotic navigation and control in the absence of a natural medium. To progress toward a feasible answer to this requirement, we have created a short project in which we will give commands to the robot by hand gestures. With this image processing technology, we can operate the robot with our fingertips. We employed image processing to capture these commands. In this approach, the robot should go in the correct direction [8].

RAHUL MEENA, The goal of this work was to apply machine learning algorithms, specifically K-nearest neighbor (KNN) and fully connected feed-forward neural network (Multilayer perceptron), to data collected from 5 different gestures using an IMU MPU 6050 sensor mounted on the Robotic Arm via BLE HM-10. Subsequently, the above-mentioned methods were compared, and it was decided that the feed-forward neural network approach performed better than the K-nearest neighbour algorithm. We reached the accuracy of 65-72% with the K-nearest neighbour technique and with the feed-forward neural network model [9].

Zehra KARAPINAR SENTURK and Melahat Sevgul BAKAY, for this study, hand gesture data from the UCI2019 EMG dataset obtained from the myo Thalmic armband were identified using six different machine learning methods. We compared the performance of (ANN), (SVM), (k-NN), (NB), (DT), and (RF) methods using metrics such as accuracy, precision, sensitivity, specificity, classification error, kappa, root (RMSE), and correlation. The data corresponds to seven hand gestures. The experiments included 700 samples from seven classes (100 samples per group). The classification ratio was 0.8- 0.2, 80% for (training) and 20% for testing the classifier. Among these methods, previous research has demonstrated that NB is the best method based on its accuracy. The classification accuracy for each gesture varies from 97.52% to 100%. According to the performance values, it can be said that this study provides successful results for detecting and recognizing seven hand gestures compared to the literature. The proposed technology can be used for HMI and smart devices control (e.g., prosthetics, wheelchairs, entertainment systems). [10].

Yuang Xiong, this paper offers an overview on the development of gesture recognition, from classical methods toward the state-of-the arts deep learning methods, and highlights challenges and technical difficulties that need to be further tackled in future. It compares a number of popular classification methods including Naive Bayes, K-Nearest Neigbors (KNN), Random Forests, XGBoost, Support Vector Classifiers (SVCs) and Convolutional Neural Networks. In addition, this paper tests the algorithms against static and dynamic gestures and discusses the tradeoff of efficiency and effectiveness in the given contexts. Results indicated that properly selecting and optimizing a machine learning algorithm can greatly enhance the accuracy and robustness of the gestural recognition systems and thus can serve as an effective practice in gesture recognition research and applications. [11].

Muhammad Imran Saleem et al. This introduce an advanced full-duplex communication system for deaf and mute people (D-M) using machine learning. These people, who use sign language to communicate, are a vital part of our society and can offer much needed contributions. Their one problem is that they can't communicate with those who don't know sign language. This issue is addressed in this paper, which introduces a communicating system which allows ND-M to communicate with D-M without having to learn sign language. The system is inexpensive, dependable, and simple to use, and it is built around a commercial off-the-shelf (COTS) Leap Motion Device. Hand gesture data from D-M persons is collected using an LMD device and processed using a Convolutional Neural Network (CNN) method. The suggested system was validated through a number of trials, with hand gesture detection accuracy above 95% [12].

Material Requirements

Below is a list of materials needed to implement a finger recognition system using Arduino. These materials are essential to ensure that the system works efficiently and smoothly as shown in figure 1:

1. Arduino Board: The Arduino board is the central device in this project, as it processes signals from the computer to control the LEDs. The Arduino Uno was used in the project, which acts as a link between the computer and the LEDs, receives data from the program and executes the appropriate commands.
2. LEDs: 5 LEDs were used, the LEDs can be of different colors or all of them are the same color. Each LED will represent a certain number of raised fingers. The LEDs light up based on the

number of raised fingers, providing direct visual interaction.

3. Resistors: Five resistors (220 ohms) were used to protect the LEDs from overcurrent, which helps preserve the life of the LEDs.
4. Connecting wires, as many as needed. They are used to connect the LEDs and other components to the Arduino board, making it easier to create the electrical circuit.
5. Webcam Used to track the movement of the user's hand and fingers, allowing the system to interact with the user dynamically.

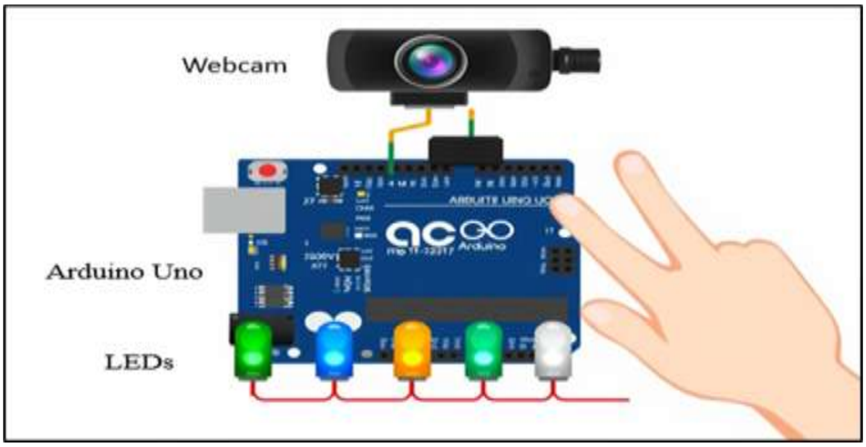

Figure 1. Material Requirements for the Arduino

Based Finger Recognition System

Software Requirements

It is important to identify the software requirements needed to ensure the project is implemented. These requirements include installing the appropriate software and necessary libraries. Below is a list of the basic requirements:

1. Python: Python programming language must be installed on your computer. Python is a powerful and easy-to-learn programming language, making it ideal for our project to implement hand tracking code and control LEDs in real time.
2. Required Libraries

The necessary libraries must be installed via the pip package manager. These libraries include:
OpenCV: For image and video processing.
MediaPipe: For hand tracking and finger feature extraction.
NumPy: For numerical data processing.
scikit-learn: For implementing the KNN algorithm to classify the number of fingers.
PySerial: For communicating with the Arduino board via serial port.

3. Arduino IDE: A program that provides an integrated development environment for programming and uploading the code for controlling LEDs and uploading it to the Arduino board.

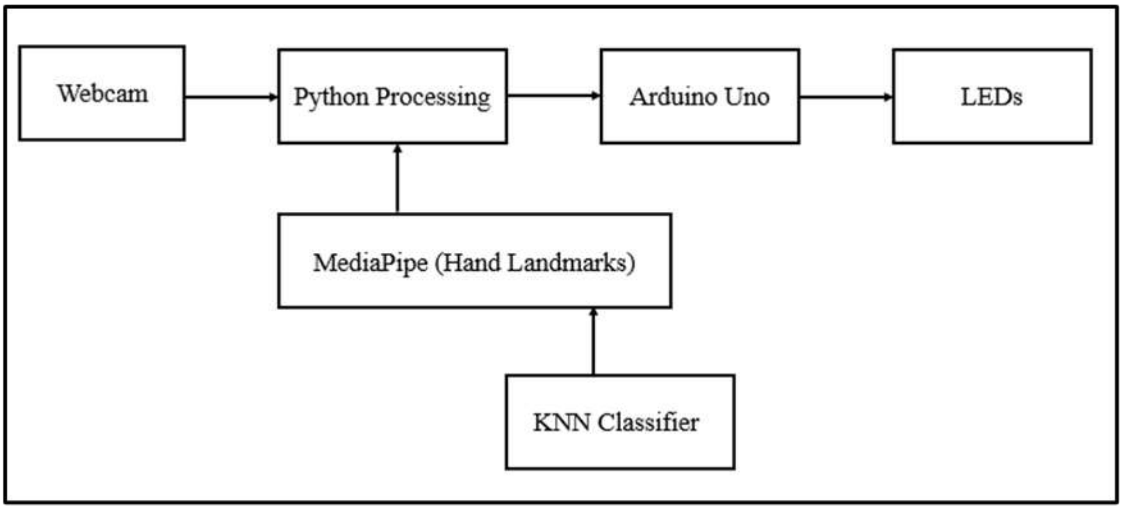

Figure 2. Existing system of hand gestures

2. Materials and Methods

KNN (K-Nearest Neighbors) Classification

KNN is a machine learning algorithm used for classification or estimation. It is based on the simple principle that similar objects are close to each other in space. Input: KNN requires a dataset containing features and targets [13]. Neighborhood checking: The algorithm searches for the K nearest neighbors of a new data point. Determine the number of neighbors (K), you must determine the number of neighbors to be considered. K can be an odd number to avoid tie in classification [14]. Calculate distance; Distance measures are used such as; Euclidean distance, most common, calculated using the formula:

$$d = \sqrt{\sum_{i=1}^{n} (xi - yi)^2}$$

Determine the nearest neighbors; after calculating the distances, the points are ranked based on their distance from the new point, and K nearest neighbors are chosen. Classification or estimation: Classification: The class of the new point is determined by the votes of the neighbors. Each neighbor gives a vote to its class, and the class with the most votes is considered the class of the new point. Estimation: In the case of estimation, the average of the target values of the nearest neighbors is calculated [15].

A dataset is used to train a KNN model to classify the number of raised fingers. Two datasets are used; Features and Targets:

Features (X)

The dataset X consists of a 2D array representing a set of angles for each finger. Each row in this array represents a different state of the number of fingers raised. The array contains 5 features (columns) representing each finger:

Column 1: represents the thumb
Column 2: represents the index finger
Column 3: represents the middle finger
Column 4: represents the pinky finger
Column 5: represents the ring finger
Each feature can be defined as a binary variable:
Xi: Where i represents the finger (from 1 to 5).

If the finger is raised: xi=1.
If it is not raised: xi=0.
Thus, the general situation can be represented as follows:

$$X = [x_1\ x_2\ x_3\ x_4\ x_5]$$

The values entered in X represent whether the finger is raised or not (1 if raised, 0 if not raised). For example:

means no finger is raised.[0 ,0 ,0 ,0 ,0]
means only the thumb is raised.[0 ,0 ,0 ,0 ,1]
[1, 1, 0, 0, 0] means both the thumb and index finger are raised.
.1.2 Objectives (y)5

The data set y represents the categories or targets corresponding to each instance of X. Each value in y indicates the number of fingers raised.

Data Representation
0: No finger raised.
1: One finger raised.
2: Two fingers raised.
3: Three fingers raised.
4: Four fingers raised.
5: All fingers raised.
The objective y can be defined as follows:

$$y = \sum_{i=1}^{5} xi$$

KNN Model Training and Performance Analysis

A comprehensive training dataset was created, where images of different finger poses were captured under various lighting conditions and shooting angles. We used an 80/20 data split, where 80% of the data was allocated for training and 20% for testing, ensuring the reliability of the results.

The best k value During our training process, several different number of neighbors (k), called hyperparameter, were explored to investigate the impact on the model's accuracy. The initial model using k=3 resulted in an accuracy of 89%, however the model was highly sensitive to noise, which impacted the classification performance under non-ideal conditions. In the computation for a k influenc of 5 we found accuracy to be at 92% which is the one that provide the best trade- off between accuracy and stability for a practical usage. When k=7 was used, the model reached the accuracy of 90%, however, it slowed the computational response time as more neighbours had to be calculated, and therefor processed, which, in turn also affected its response time. In order to objectively judge performance of the system, the performance is systematically tested under different conditions to make sure the model can be used in real world condition. The experiments covered a range of critical elements, starting with lighting. The system performed accurately and robustly under bright light conditions. We also tested the influence of noise on classification accuracy in low light conditions and we observed that the obtained performance dropped noticeably, showing the importance of good lighting. Furthermore, natural usage environments were simulated as a test of natural lighting, i.e., how well the model could tolerate natural changes in light. In the case of shooting direction, the system performed well when tested on the hand directed towards the camera. We further conducted experiments from different angle groups: 30°, 60°, 90° in order to study the effect of hand deviation on classification accuracy, and we observed a less classification accuracy with increasing deviation angles. The accuracy of the model was also tested at the distance of the hand from the camera, finding that the distance from the fingers led to a drop in the ability to detect fingers with higher accuracy.

As for hand size, the system was experimentally applied to different hand specimens (from child hand to adult hand) to explore the influence of size on the performance of the model. The model was able to generalize to different hand sizes, but worked better on adult hands. Lastly, the response time was captured as the time it took to raise the hand and turn on the light, resulting in an averaged processing time of 0.15 seconds. This rate of speed makes the system ideal for a real-time interaction, which will improve both the user experience and the play experience.

3. Results and Discussion

Experiments showed over 90% accuracy in recognizing the number of raised fingers. KNN algorithm test results showed that k=5 had the highest accuracy of 92%. The model's robustness to diverse lighting situations made it suitable for practical applications. The system was integrated with a genuine device to operate LEDs based on detected finger count. Table 1 shows the system's accuracy for various values of (k):

Table 1. Presents the accuracy of system at various values

Number of neighbors (k)	Classification accuracy (%)
3	89%
5	92%
7	90%
10	88%

The results indicate that the KNN algorithm is an effective choice for gesture recognition applications, offering high accuracy and fast performance. However, challenges related to data noise and lighting variations must be addressed. The development of more complex models, such as neural networks, could improve performance.

4. Conclusion

The results of this study verify the high accuracy and efficiency of the proposed finger tracking system, which can achieve a recognition accuracy of 92% of hand gestures based on KNN algorithm with number of neighbors k=5. High accuracy, with response time 0.15s or less, makes the system suitable for interactive applications that have real-time signal processing needs. Despite these encouraging results, the study demonstrated that the performance of the system is influenced by several environmental conditions, including lighting and image quality. Performance that outperforms was seen in strong lighting and against large hands and clear numbers. This finding suggests the necessity of additional improvements in order to achieve higher system efficiency especially in non-ideal conditions, such as low illumination and non-normal angles of incidence. A significant step in creating user-friendly human-machine interfaces is represented by this work. The technology paves the way for cutting-edge applications in a variety of domains, such as virtual and augmented reality technologies, smart home systems, and smart device control systems. The system's usefulness and added value in the realm of human-computer interaction are highlighted by these encouraging applications.

REFERENCES

[1] S. Ozbay and M. F. Ozkan, “Enhancing human–computer interaction (HCI) through machine learning application,” in Proc. 8th Int. Symp. Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2024.

[2] T. A. Mahmood, “Unified grounding systems for future projects to ensure public safety and improve performance,” Int. J. Innovative Sci. Res. Technol., vol. 9, pp. 2439–2450, Aug. 2024, doi: 10.38124/ijisrt/IJISRT24AUG1412.

[3] M. Waqas, M. A. Khan, A. Rehman, T. Saba, and Z. Mehmood, “A gesture-based control system using K-nearest neighbor and support vector machine,” IEEE Access, vol. 9, pp. 78660–78670, 2021.

[4] J. S. Vimali, A. V. Lakshmi, and R. Praveena, “Low-cost hand gesture recognition system using Arduino

and ultrasonic sensors," Int. J. Sci. Technol. Res., vol. 9, no. 3, pp. 4562–4566, 2020.
[5] M. Wameed and A. M. Alkamachi, "Hand gestures robotic control based on computer vision," Int. J. Intelligent Systems and Applications in Engineering, vol. 11, no. 2s, pp. 1013–1021, 2023.
[6] R. N. Priya, "Arduino based hand gesture control of computer," Int. Res. J. Modernization in Eng. Technol. Sci., 2023.
[7] S. Chavan, R. Jadhav, J. Mankar, S. Thakur, S. Ansari, and V. Panchal, "Real-time hand gesture recognition based control of Arduino robot," Int. J. Trend Sci. Res. Develop., 2022.
[8] D. Jeena, E. Pal, H. Garg, and S. S. Deswal, "Gesture controlled robot using machine learning," Int. Res. J. Modernization in Eng. Technol. Sci., 2021.
[9] R. Meena, "Machine learning (ML) for gesture recognition," Ph.D. dissertation, Indian Institute of Technology Madras, 2021.
[10] Z. K. Senturk and M. S. Bakay, "Machine learning-based hand gesture recognition via EMG data," Advances in Distributed Computing and Artificial Intelligence Journal, 2021.
[11] Y. Xiong, "Research on gesture recognition technology based on machine learning," Applied and Computational Engineering, vol. 104, no. 1, pp. 116–121, 2024.
[12] M. I. Saleem, A. Siddiqui, S. Noor, M.-A. Luque-Nieto, and E. Nava-Baro, "A machine learning based full duplex system supporting multiple sign languages for the deaf and mute," Sci, vol. 13, p. 3114, 2023.
[13] E. J. Dhulkefl, A. W. Abdulsattar, Z. Mahmood, and T. A. M. Khudhur, "Design of a hybrid intelligent traffic signal control system using nearest neighbor algorithm and deep reinforcement learning with SUMO simulator," J. Res. Eng. Comput. Sci., vol. 3, no. 3, pp. 31–40, 2025.
[14] C. Cui, M. S. Sunar, and G. Eg Su, "Deep vision-based real-time hand gesture recognition: A review," PeerJ Computer Science, vol. 11, e2921, 2025, doi: 10.7717/peerj-cs.2921.
[15] M. C. Ergene, A. Durdu, and H. Cetin, "Imitation and learning of human hand gesture tasks of the 3D printed robotic hand by using artificial neural networks," in Proc. 8th Int. Conf. Electronics, Computers and Artificial Intelligence (ECAI), Jun. 2016, pp. 1–6.