Touchless Control of Heavy Equipment using Low-Cost Hand Gesture Recognition

Leyla Khaleghi, Unal Artan, *Student Member, IEEE*, Ali Etemad, *Senior Member, IEEE*, Joshua A. Marshall, *Senior Member, IEEE*

***Abstract*—Human-machine interaction using remote hand gestures is becoming increasingly prevalent across various industries. However, their potential application to heavy construction equipment is often overlooked. This paper presents a robust and inexpensive hand gesture recognition system that was implemented and tested on a robotic 1-tonne wheel loader. The system uses an RGB camera paired with a laptop to process, in real time, hand gestures to control the loader. We first design 4 unique gestures for controlling the loader and then collect 26000 images to train and test a neural network for hand gesture recognition. Our system uses robust landmark detection using an off-the-shelf system prior to gesture recognition. We successfully controlled the loader to excavate in a rock pile by using the proposed hand gesture recognition system.**

***Index Terms*—Hand gesture recognition, Heavy equipment,Internet of Things, Robotics, Deep learning.**

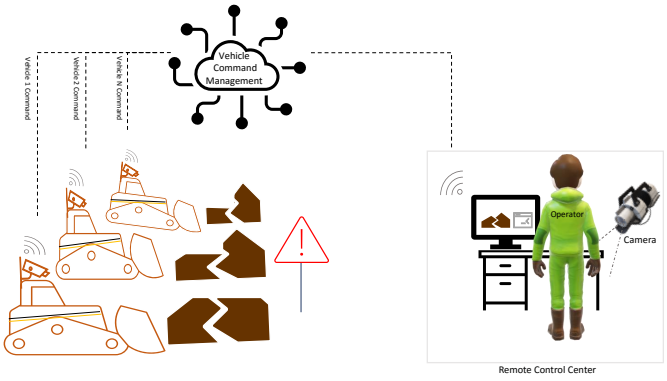

Fig. 1: Overview of a vision-based hand gesture control system for central remote control of a fleet of heavy equipment.

I. Introduction

Human-Machine Interaction (HMI) is experiencing a wave of change and modernization due to recent advances in sensing systems, Internet of Things (IoT) infrastructure, and Artificial Intelligence (AI). An example of novel ways of interacting with machines is the use remote hand gestures for applications ranging from control of common desktop tasks and video games to robotic systems and vehicles [1], [2]. Hand gestures, for example, can be used to control a vehicle's infotainment system which provides a safer driving experience due to the lower cognitive load induced on the driver [3], [4]. The transition from touch-based to touchless interaction by using remote hand gestures has accelerated during the Covid-19 pandemic to improve hygiene and reduce infection risks [5].

Despite the vast potential for the use of touchless HMI for controlling heavy equipment in reducing the risk of injury, fatigue, and the ability to remotely and safely operate such equipment in hazardous environments, this potential has been scarcely explored. Such a platform, which can be created by using advanced IoT infrastructure, can enable remote stations to be built for operators, which would afford them the option of being located in a comfortable setting, away from potential hazards, to maintain reasonable mobility, and able to interact with heavy vehicles hygienically. Furthermore, these stations could be used as a remote control center for an entire fleet of vehicles. In recent times, this approach has been used in many military or civilian applications where complex vehicles could be replaced with multiple simpler ones [6]. An overview of this platform is provided in Fig. 1. With this idea in mind, this article introduces a novel, low-cost, real-time system for control of a wheel loader using hand gestures. The developed system is demonstrated in the field by successfully excavating in a rock pile. A video clip of the system in action can be seen at https://youtu.be/Q5ElpgUa41Y.

The remainder of this article provides details about the developed system, a description of the field experiments, and concluding remarks.

L. Khaleghi is with the Department of Electrical & Computer Engineering, and Ingenuity Labs Research Institute, Queen's University, Kingston, Canada.

U. Artan is with Robert M. Buchan Department of Mining, and Ingenuity Labs Research Institute, Queen's University, Kingston, Canada.

A. Etemad is with the Department of Electrical & Computer Engineering, and Ingenuity Labs Research Institute, Queen's University, Kingston, Canada.

J. A. Marshall is with the Department of Electrical & Computer Engineering, and Ingenuity Labs Research Institute, Queen's University, Kingston, Canada.

II. Proposed System Design

The proposed system is composed of two main components to control the vehicle : A) hand segmentation and landmark estimation; B) gesture recognition. The overall architecture is shown in Fig. 2. In what follows, we describe in detail each of the main components.

A. Hand Detection and Landmark Estimation

Prior works have shown that hand *landmarks* are useful features for vision-based hand gesture recognition systems [7], [8]. Accordingly, our system design incorporates hand landmark detection as the first step. We used the open-source system Mediapipe[1] [9], which has been developed by Google for hand detection and capturing high-quality hand landmarks.

[1]Available online at https://google.github.io/mediapipe/solutions/hands.html.

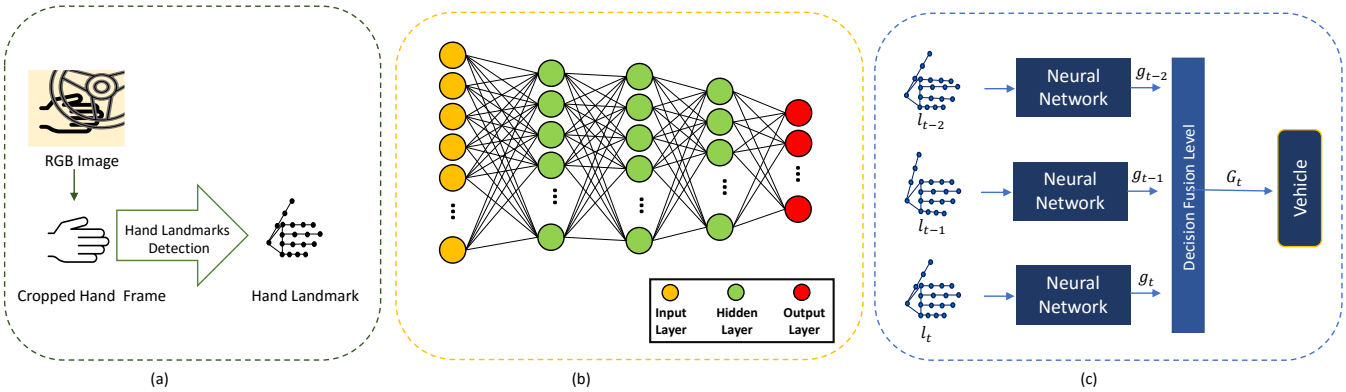

Fig. 2: An illustration of our proposed pipeline: (a) hand detection and landmark estimation; (b) the ANN used in the system; and (c) the gesture recognition module.

Mediapipe is a free and off-the-shelf application for tracking hands and estimating hand landmarks in *real-time* and by using only a CPU with no need for GPUs. Because Mediapipe has been trained on both real-world and synthetic images of hands, it is highly reliable and has been tested in many prior applications [10], [11], [12].

MediaPipe performs multiple tasks for our system, with the first being hand segmentation. The palm region is used to crop the hand from the image and reduce the image size from 1080 920 to 224 224. Next, the image containing only the hand (I 2 $\mathbb{R}^{224\times224\times3}$) is used to estimate 3D landmarks, L. In total, 21 landmarks are estimated from image I by identifying the wrist and fingertip, Distal Interphalangeal Joint (DIP), Proximal Interphalangeal Joint (PIP), and Metacarpophalangeal Joint (MCP) for each finger. The landmarks contain an estimate of x, y, and z, such that L 2 $\mathbb{R}^{21\times3}$. x and y are normalized values using the image width and height, respectively, for normalization and z is the depth relative to the depth of the wrist. The workflow for MediaPipe is shown in Fig. 2(a).

B. Gesture Recognition

The estimated hand landmarks, L, are then used to classify the hand gesture class. An artificial neural network (ANN) Multilayer Perceptron (MLP), shown in Fig. 2(b), is used to estimate the probability of each hand gesture class, p_i and generate $\mathbf{g} = [p_1; p_2; \cdots; p_N]$ where $i = 1; 2; \cdots; N$, and N is the number of classes.

The ANN weights are optimized during training to minimize the cross-entropy loss function according to

$$Loss = \sum_{i=1}^{N} p_i \log(\hat{p}_i); \quad (1)$$

where $\hat{p}_i$ and p_i are the predicted and ground truth probabilities for class i, respectively.

A form of robustness was incorporated into the system by retaining the gesture class probability $\mathbf{g}_{t-2}$, $\mathbf{g}_{t-1}$ obtained from the previous two images I_{t-1} and I_{t-2}. The gesture class with the highest probability, G_t, is calculated using

$$G_t = \underset{i\in\{1;\cdots;N\}}{\operatorname{argmax}} \left(\sum_k p_{i;t-k} \right); \quad (2)$$

where $k = 0; 1; 2$. This component is demonstrated in Fig. 2(c). Due to the static nature of the designed gestures, increasing the window size beyond 3 frames did not improve the performance. Therefore, we use this window size in our system. We trained the neural network using the Adam optimizer for 500 iterations with an initial learning rate of 0:1. To minimize overfitting, during the training process, a dropout of 15% was used.

TABLE I: Comparison of performance and runtime of hand gesture classifiers with two different testing protocols.

Test	Model	Accuracy	F1 score	Runtime (ms)
Cross-Sub.	Naive Bayes	0.764	0.791	**0.19**
	Random Forest	0.942	0.949	6.33
	SVM	0.968	0.974	0.21
	LSTM	0.969	0.974	0.92
	ANN	**0.994**	**0.993**	0.74
Cross-Light	Naive Bayes	0.801	0.837	**0.19**
	Random Forest	0.897	0.889	6.33
	SVM	0.976	0.981	0.21
	LSTM	0.978	0.979	0.92
	ANN	**0.989**	**0.990**	0.74

III. Field Experiments

A. Data Collection and Training

Four hand gestures were used to control the loader. The gesture's loader actions were labelled Stop, Throttle, Forward and Dig, and corresponded to changes in the loader inputs. The estimated hand landmarks using MediaPipe for each hand gesture class is provided in Fig. 3(a).

Images were collected in different lighting (low-light and high-light) conditions. Five operators performed each hand gesture for roughly two minutes. During this time, the hand was lowered and raised, and finger postures were altered to create a robust image set. In total, 26000 images were collected, 5200 for each hand gesture. The captured images were then processed using MediaPipe to create hand landmark estimates.

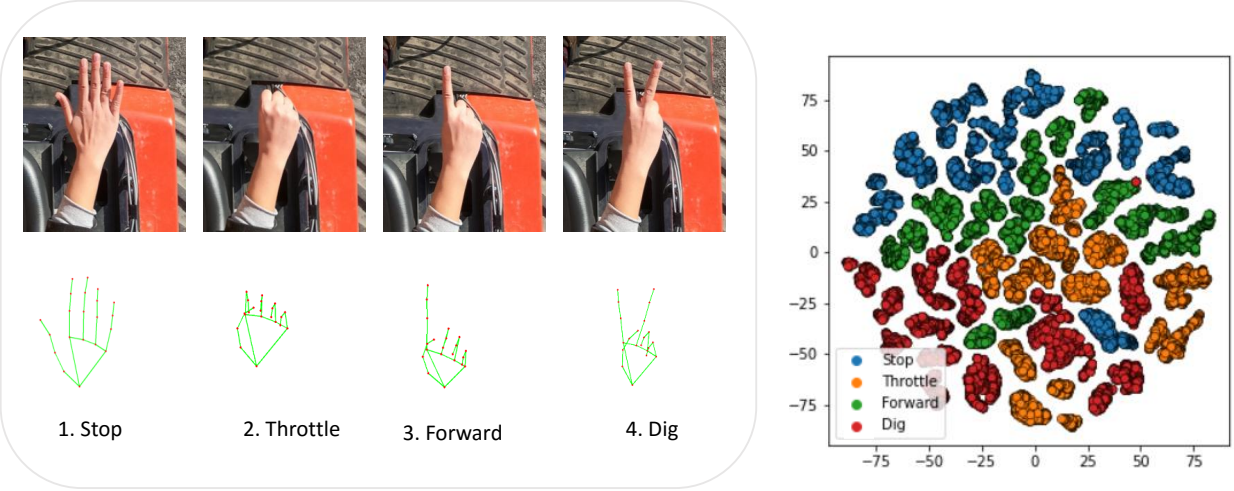

Fig. 3: (a) Example image and estimated hand landmarks for the four hand gestures; (b) visualization of the hand gestures using t-SNE.

The uniqueness of gestures in the collected dataset is visualized using t-Distributed Stochastic Neighbour Embedding (t-SNE) in Fig. 3(b). The high dimensionality of each gesture (63) is reduced to a 2D space, which is easier to visualize. Each colour in the t-SNE plot represents a hand gesture and the different regions.

We used two different test protocols for training and evaluation of our models, Leave-One-Subject-Out (LOSO) (also known as cross-subject) and cross-lighting. In cross-subject, for each trial, one subject was selected for testing, while the remaining subjects were used for training. Also, we performed cross-lighting experiments by testing in high-light data for one subject, while training on remaining subjects' low-light data and vice-versa. In all, five trials for the cross-subject protocol and ten trials for the cross-light protocol were examined. We also compare the performance of the ANN with Long Short-Term Memory (LSTM) network, as well as three classical machine learning algorithms, Naive Bayes, Random Forest, and Support Vector Machine (SVM). We present the mean accuracy and F1 scores for all the classifiers across both test protocols, as well as the runtimes, in Table I. The runtimes capture the time required for each model to classify the gesture in one frame. Based on the table, we observe that the ANN achieves the best performance in comparison to the other classifiers. While models such as Naive Bayes and SVM are expectedly faster than the ANN, our model still performs in realtime given that it can process over 30 frames per second which is the sampling rate of the videos.

B. Vehicle Description

Field experiments were performed using an instrumented Kubota R520s wheel loader shown in Fig. 4. On-board sensors and actuators enable the loader to be remotely operated [13], [14], [15] by the hand gestures, as specified in Table II.

1) Brake: The brake pedal is actuated by a linear servo motor. The position command, $u_B \ 2 \ [0;1]$, where $u_B = 1$ corresponds to a fully depressed brake.

2) Throttle: The throttle pedal is actuated by a linear servo motor. The position command to this motor $u_T \ 2 \ [0;1]$, where $u_T = 1$ corresponds to a fully depressed throttle. Engine RPM is not logged in the current setup, but the tachometer value can be read by the operator by viewing a display on the dashboard.

3) Gear Selection: The selected gear, GS, is one of three states: neutral (NEUT), forward (FWD), and reverse (REV).

4) Proportional Control Valves: An electro-hydraulic proportional valve controls the fluid flow to the dump cylinder. The flow rate is proportional to the command signal $u_D \ 2 \ [\ 1;1]$. Here, negative command values correspond to cylinder retraction and positive values corresponded to extension.

5) Imaging System: An Intel RealSense D435 camera was used to capture the hand images. The camera was connected using USB 3.0 and outputs an RGB image as well as a depth image. The depth information from the camera was not used. The camera was mounted to the roof, as shown in Fig. 4.

6) Control System: An onboard control system was used to send actuator command signals and log sensor data. The control system consists of a main control unit (MCU), seven CAN Peripheral Interface (CPI) sub-control modules, a laptop running the Robot Operating System (ROS) and the hand gesture recognition system. The MCU operated at 10 Hz for sending commands via the CPIs and the developed hand gesture recognition system operated at 30 Hz.

C. Construction Zone and Safety

Field experiments were performed in an open area with a rock pile for excavation experiments, as shown in Fig. 4. The safe practice guidelines from Workplace Safety North Mining Sector (WSN) for work in construction zones were followed [2].

[2]Available online at https://www.workplacesafetynorth.ca/industries/mining

Fig. 4: Kubota R520s wheel loader controlled using hand gestures excavating a rock pile.

TABLE II: Loader input commands for given gestures.

Gesture	Loader Inputs			
	u_B	u_T	GS	u_D
Stop	1	0	NEUT	0
Throttle	1	0.7	NEUT	0
Forward	0	0.7	FWD	0
Dig	0	0.7	FWD	-0.5

Some of the safety items implemented on the loader, such as the tower light that signals the operating mode, an off-on-off handheld safety pendent needed to enable loader operation, and emergency stop buttons located on the rear bumper, are highlighted in Fig. 4.

A scenario may arise where there is a tie in the probability of hand gesture classes. To overcome this scenario, a safety score associated with each class was developed. Each hand gesture class had an associated safety score, with Stop having a score of 4, Throttle a score of 3, Forward a score of 2, and Dig a score of 1. Whenever a tie occured, the class with the highest safety value was used.

IV. Summary and Future Work

We have demonstrated a low-cost, real-time touchless system that uses hand gestures to control heavy equipment. Experiments using a 1-tonne capacity wheel loader with four hand gestures demonstrated successful excavation in a rock pile. The developed system was trained using images captured across different lighting conditions from five subjects. The trained ANN was able to distinguish the four hand gestures with 99% accuracy and in realtime.

For future work, the system could be expanded to centrally control a remote fleet of heavy equipment through a wireless link between the vision system and the fleet, as well as the integration of fleet management software for automatic distribution and efficient operation. With this system, one or a few operators can operate a number of heavy vehicles all at once from a remote command center using hand gestures, allowing a higher degree of mobility as they would no longer be restrained to a single position. To realize such an expanded system, additional gestures may need to be designed to allow the vehicles to 'continue' or 'repeat' a specific action until told otherwise, which would allow the operator to control other vehicles. Lastly, other models such as Graph Neural Networks (GNN) could be explored for hand gesture classification in future work.

References

[1] Biplab Ketan Chakraborty, Debajit Sarma, Manas Kamal Bhuyan, and Karl F MacDorman. Review of constraints on vision-based gesture recognition for human–computer interaction. *IET Computer Vision*, 12(1):3–15, 2018.

[2] Farzin Farhadi-Niaki, S Ali Etemad, and Ali Arya. Design and usability analysis of gesture-based control for common desktop tasks. In *International Conference on Human-Computer Interaction*, pages 215–224. Springer, 2013.

[3] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2368–2377, 2014.

[4] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3D convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015.

[5] Muhammad Bilal Khan, Zhiya Zhang, Lin Li, Wei Zhao, Mohammed Ali Mohammed Al Hababi, Xiaodong Yang, and Qammer H Abbasi. A systematic review of non-contact sensing for developing a platform to contain covid-19. *Micromachines*, 11(10):912, 2020.

[6] Adel Belkadi, Hernan Abaunza, Laurent Ciarletta, Pedro Castillo, and Didier Theilliol. Distributed path planning for controlling a fleet of uavs: application to a team of quadrotors. *IFAC-PapersOnLine*, 50(1):15983–15989, 2017.

[7] Rui Li, Zhenyu Liu, and Jianrong Tan. A survey on 3D hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition*, 93:251–272, 2019.

[8] Weiya Chen, Chenchen Yu, Chenyu Tu, Zehua Lyu, Jing Tang, Shiqi Ou, Yan Fu, and Zhidong Xue. A survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors*, 20(4):1074, 2020.

[9] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

[10] Sriram SK and Nishant Sinha. Gestop: Customizable gesture control of computer systems. In *8th ACM IKDD CODS and 26th COMAD*, pages 405–409. 2021.

[11] Herbert Shin. Web application–utilizing a pose estimation and augmented reality api for hand telerehabilitation. 2021.

[12] Wai Kin Koh, Quang H Nguyen, Youheng Ou Yang, Tianma Xu, Binh P Nguyen, and Matthew Chin Heng Chua. End-to-end hand rehabilitation system with single-shot gesture classification for stroke patients. In *Soft Computing: Biomedical and Related Applications*, pages 59–67. Springer, 2021.

[13] Heshan A Fernando, Joshua A Marshall, Håkan Almqvist, and Johan Larsson. Towards controlling bucket fill factor in robotic excavation by learning admittance control setpoints. In *Field and Service Robotics*, pages 35–48. Springer, 2018.

[14] Johann von Tiesenhausen, Unal Artan, Joshua A Marshall, and Qingguo Li. Hand gesture-based control of a front-end loader. In *IEEE Canadian Conference on Electrical and Computer Engineering*, pages 1–4. IEEE, 2020.

[15] Unal Artan, Heshan Fernando, and Joshua A Marshall. Automatic material classification via proprioceptive sensing and wavelet analysis during excavation. In *IEEE International Conference on Advanced Intelligent Mechatronics*, pages 612–617. IEEE, 2021.