https://doi.org/10.22581/muet1982.2204.08

2022, 41(4) 74-86

Interactive restaurant menu ordering system using projection mapping and machine learning

Sara Qadeer, Khuhed Memon, Ghulam Hyder Palli

Department of Electronic Engineering, Mehran University of Engineering and Technology, Jamshoro Sindh Pakistan

[*] Corresponding author: Sara Qadeer, Email: sara.rajput@faculty.muet.edu.pk

Received: 27 June 2021, Accepted: 06 June 2022, Published: 01 October 2022

KEYWORDS

Projection Mapping

Machine Learning

Restaurant

Interactive Table

Menu Ordering System

ABSTRACT

Integration of automation and technology in every aspect of life is the global trend in the present age. It encompasses every facet of society including education, medicine, entertainment, defense, clothing, and restaurants, among others. Restaurants have also embraced technologies to facilitate their customers. Technologies like tablets for customer order, augmented reality, projection mapping for menu projection, touchscreen panels for order placement are among few worth mentioning. This paper proposes an entertaining way for the customers to place their orders without a waiter, using Machine Learning & Projection Mapping. The table serves as an interactive medium to browse through the menu and communicate with the kitchen with the touch of a finger. We have developed the product using TensorFlow and OpenCV using Python, along with a webcam or IP camera. The Graphical User Interface (GUI) has been designed in QT designer using pyqt5. This system is user friendly and can be installed in every environment. It can prove to be a low cost, high accuracy, real-time alternative to the existing technologies and can work in the cluttered environment with different illuminations with 96% accuracy.

1. Introduction

Machine Learning and Artificial Intelligence is the new electricity of this era. It has revolutionized the world and every industry in it. The availability of big data originating from a vast number of smart machines available these days has played a significant role in bringing about this change. The humungous increase in the processing capabilities of our computers (CPUs & GPUs) has empowered us to perform millions of computations in fractions of a second, thereby putting this massive data to great use in no time. In addition to this tremendous amount of data and computational capabilities, the development of extremely efficient Machine Learning algorithms and open-source software libraries have played a key role in this rise of Artificial Intelligence. This research is an attempt to design and develop an interactive, fast, and accurate restaurant menu ordering system in which the dining table acts as a touch panel to place the order directly to the kitchen without having to wait for someone to do it manually or having to invest in expensive touch gadgets prone to physical harm by the customers or staff of the restaurant. This has been achieved by using a multimedia projector and an IP camera placed directly above the dining table. The projector projects the graphical user interface, developed in QT designer using pyqt5, on the table whereas the camera continuously acquires the user's hand activities on the table using TensorFlow object (hand/nail) detection. This is followed by the User

Datagram Protocol (UDP) transmission of the selected menu to the kitchen.

The organization of the paper is such that the work related to food ordering systems and technologies for dining experiences is discussed in section 2. Section 3 provides information on the equipment, algorithm, experimental setup, and implementation methodology of a real-time interactive restaurant table. Section 4 evaluates the performance of the developed system with respect to speed and accuracy and presents the results. Finally, the paper concludes in section 5 and acknowledgements are given in section 6.

2. Literature Review

Dixon et al. [1] present the results of a national survey on customers' perceptions of 11 restaurant technologies. The survey compares various technologies like pagers for table management, kiosk-based food ordering, waiting in line, Internet-based ordering, virtual menus available tableside, among others and it suggests that 100% of the restaurant customers consider virtual tableside menus to be the most valuable technology among all others. This builds a strong motivation for our research since the primary market of this project is the restaurants' diners, making their choice and comfort of foremost importance. Spence and Fiszman [2] discuss various technologies at the dining table, emphasizing the efficiency of ordering systems that remove the requirement of waiters to carry orders to the kitchen. It refers to 'Inamo', a restaurant in London, which offers similar dining table projections for ordering food and playing games, but this system has limitations, i.e. it uses hardware like mouse trackpads and pushbuttons placed on the dining table, which result in higher cost in addition to susceptibility to equipment damage. They also discuss technologies like smart-plates and smart-spoons using which a customer can keep track of their eating habits. Technologies like Augmented Reality (AR) & Virtual Reality (VR) have also been discussed in this research, using immersive feelings that supplement the dining experience that can be introduced, e.g., playing sounds of the sea and sea birds while enjoying seafood. Bankar and Suresh [3] designed Arduino Mega based touchscreen module for order placement with PayPal platform for online payment and a Line Following Robot was introduced for order serving to the customers to reduce the workload on waiters, but the system is limited in terms of the damage to the touchscreen module and drawing multiple lanes for Line Following Robot because it is not practical in a real scenario and has a high cost. Torres [4] designed a smartphone application which contains both the text and image information of food for order placement to reduce the customer and wait staff gap but again those touchscreen gadgets are likely to damage. The android-based restaurant automation system is proposed in [5] in which the order is placed from the android application and order information is conveyed through the Bluetooth module to the kitchen for order placement. The traditional food ordering devices are replaced [6] with the constant tablet on each table of a restaurant for food ordering. The digital restaurant is introduced [7], in which the order is placed from the customer's smartphone by accessing an online mobile application. And as a result when the food is ready, customer can pick it from the conveyer belt and the payment is made online through the card swipes. The food ordering system through a tablet is proposed [8], in which android and PHP programming is used to design the application but the system is limited to high cost and might get damaged from the improper usage by the customer. Jaffee et al. [9] deal with the study of 3D object projection and interactivity but the mechanism used to deal with interactivity is a 15-inch Gvision touchscreen, which is similar to touch projectors, resulting in issues like high cost and risk of equipment safety. Use of Personal Digital Assistants (PDAs) or Android devices by waiters/customers to wirelessly communicate orders to the kitchen is proposed [10-13]. Badariah et al. [14] and Hashim et al. [15] propose the use of keypads at dining tables with Liquid Crystal Display (LCD) in the kitchen communicating using Zigbee and Bluetooth respectively to deliver low cost, low power, and low latency communication. Soon Nyean Cheong et al. [16] developed a multi-touchable e-restaurant management system, with an interactive dining menu to manage orders and billing. They have also developed a GUI and a network management system, but the limitation of a physical hardware placed on the table still remains an issue. Wahab et al. [17] propose the implementation of a Network-based Smart Order System using keypads, microcontroller, display panels, and transmission cables. They have used Microsoft Access and Visual Basic for Database management and GUI for design. Our research aims at eliminating the use of such hand-held or table-installed expensive gadgets, microcontrollers, and keypads to get the job done using the dining table itself as a touch-sensitive medium, thereby increasing the ease of use, efficiency, and safety in addition to drastically reducing the risk of equipment damage. Onsite restaurant interactive self-service (ORISST) is used by top leading quick service restaurants like McDonalds, where tabletop tablets and self-service kiosks are used to enhance their customers' dining experience along with making and customizing their own orders. The research also implies that utilitarian and delightful display are both should be embedded for successful implementation of ORISST [18]. Research also found that as technologies tested in Technology Acceptance Model studies have been improved by highly interactive systems, increased capability and a more user-friendly interface, examining perceived interactivity of technology has become more important for advance robot acceptance models [19]. Technologies play a vital role in recent times as research proposes an AI enabled interactive order recommender system that could recommend a suitable menu based on the customers' historical consumption details to create an interactive dining experience [20]. Another research shows that the quality of experience and service for food industry consumers is highly affected interruptions in flow rate of service by employees which can be eased by introducing self-service technologies [21].

As the food industry advances, it is crucial to attract customers with different strategies and menu presentation is one of them and one such system is eCarte, a two-user interface interactive restaurant menu that facilitates the customers with feedback collection as well [22]. Moreover, the existing solutions are summarized in Table 1.

Table 1
Summarized existing solutions

S. No.	Existing Solutions	Literature
1.	Arduino Mega based touch screen panel was designed for food ordering. Touch panel may get damage by the end users.	[3]
2.	Smartphone application-based food ordering system is proposed. Again, it needs smart gadgets for food ordering so they may be damaged by the end users.	[4], [5]
3.	Tablets were used for food ordering using online mobile application. Again, tablets may be damaged by the improper usage.	[6]
4.	Online application is proposed for food ordering, but every end user may not have smartphone to access that online application.	[7]
5.	Tablets based food ordering is proposed in which android mobile application is designed to place the order, but tablets may be damaged by the end users.	[8]
6.	15-inch Gvision touchscreen is used for order placement, but it is sensitive to water and dust.	[9]
7.	Proposed the use of Personal Digital Assistants (PDAs) or Android devices by waiters/customers to wirelessly communicate orders to the kitchen. Again, they may be damaged.	[10] – [13]
8.	Proposed the use of keypads at dining tables with Liquid Crystal Display (LCD) in the kitchen communicating using Zigbee and Bluetooth. But physical hardware on the table may be damaged.	[14], [15]
9.	Developed a multi-touchable e-restaurant management system, with an interactive dining menu to manage orders and billing but physical hardware on the table may be damaged.	[16]
10.	Proposed the implementation of a Network-based Smart Order System using keypads, microcontroller, display panels, and transmission cables. The physical hardware on the table may be damaged.	[17]
11.	Interactive displays were used for food ordering. But they are not as interactive as the proposed system is.	[18]

3. Methodology

This section provides the details pertaining to the hardware and software used in the development of the Interactive Restaurant Menu Ordering System. It also covers the experimental setup and the algorithm deployed to achieve the task.

3.1 Hardware

In order to be able to develop a real-time, standalone, fast, and accurate system, Lenovo Legion has been chosen for the deployment of this system. A multimedia projector has also been used for projections of the menu on the restaurant's table and an IP camera has been used to detect user's touch input. A remote UDP receiver is also deployed in the kitchen to receive orders from different tables in the restaurants. This section briefly discusses the aforementioned hardware.

3.1.1 Workstation and smart phone

The specifications of the workstation and phone used in the development process are given in Table 2.

Table 2

Hardware Specification

Hardware	
Workstation	
Model	Lenovo Legion Y545
CPU	Intel i7-9750H 2.6G (9th Generation)
GPU	Nvidia GeForce GTX 1660Ti 6 GB
RAM	16GB
Storage	1TB HHD + 512GB SSD
OS	Windows 10 Home
Smart Phone	
Model	Infinix HOT4 LTE
CPU	Quad-core 1.3 GHz Cortex-A7
GPU	Mali-400MP2
RAM	2GB
Storage	16GB
OS	Android 6.0 (Marshmallow)

3.1.2 Camera

An IP camera and a webcam were used in the development of this project. In addition, the camera of a smartphone was also used as an IP camera with the help of the IP Webcam android application, as shown in Fig. 1 for development and testing purposes. The resolution used is 1920 x 1080.

(a) Lenovo 720p HD (b) IP Camera (c) Infinix Hot Camera

Fig. 1. Cameras

3.1.3 Multimedia projector

The projector used in this research is Vivibright GP70 Portable Home Theater LED Projector as shown in Fig. 2. With a decent resolution and 1200 Lumens along with its compact size and low price, this projector served as a reasonable choice for this application.

3.1.4 Mobile device as UDP receiver

We have used Infinix Hot 4 LTE with android version 6.0 running UDP sender/receiver application to serve as a UDP receiver placed in the kitchen to display orders from customers in the restaurant, as shown in Fig. 3.

3.2 Software

The algorithm was developed and tested on Python as the deployment platform using TensorFlow and OpenCV libraries. The GUI has been developed in pyqt5 and QT Designer. The major software and libraries used in this project are listed in Table 3.

Fig. 2. Portable home theater LED projector

Fig. 3. UDP reception on a smartphone

3.1.5 Assembly to hold camera and projector

A temporary table-mounted wooden assembly was used to hold Raspberry Pi-4 B (4GB) and Pi camera during algorithm development and testing for hand detection and gesture recognition on the table, as shown in Fig. 4(b). Another floor-mounted assembly as shown in Fig. 4(a) was constructed with the provision to hold a multimedia projector and a mobile phone to be used as an IP camera. This was later converted to a ceiling-mounted lamp fashioned assembly to keep the equipment from being physically damaged by the customers or food/beverage spillovers as shown in Fig. 4(c). The details of these assemblies are given in section 3.3.

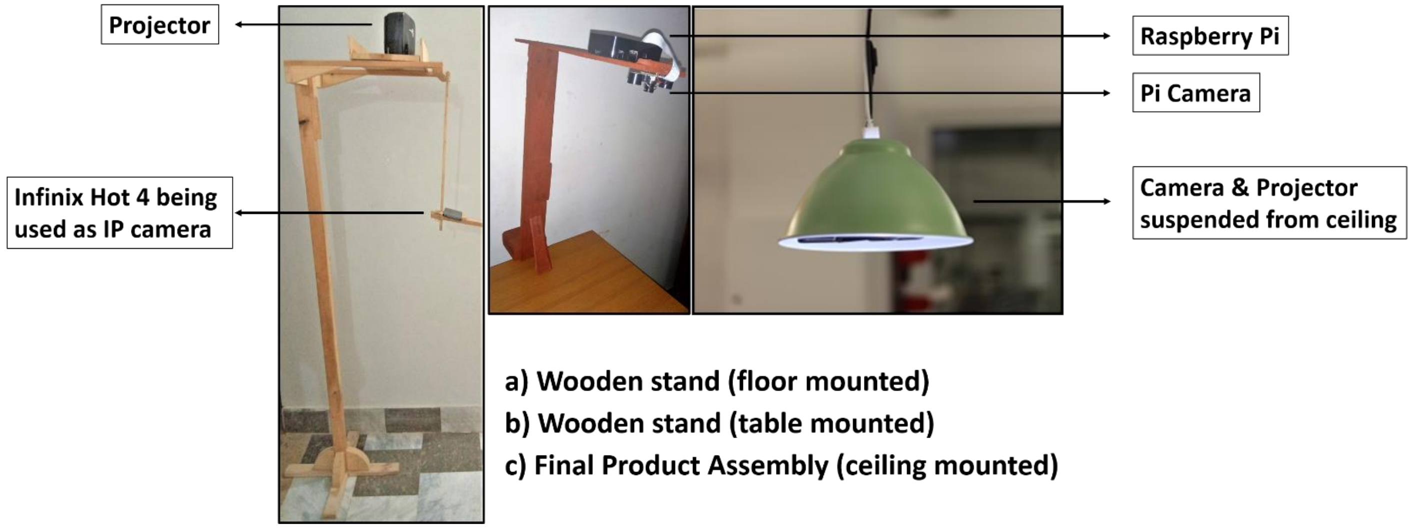


Fig. 4. Camera and projector holders

Table 3

Software and libraries

Software	Version
Python	3.7.6
PyCharm	2021.1.1
TensorFlow	1.14
TensorFlow-GPU	1.14
TensorBoard	1.14
Nvidia CUDA	10.0
CuDNN	7.4.1
QT Designer PyQt5	5.13.0
Open CV	4.1.2

3.3 Experimental Setup

Various temporary arrangements for holding the camera and projector setup were constructed for experimentation as shown in Fig. 4. The final placement of the camera/projector assembly was placed 4 feet above the dining table as shown in Fig. 5. This ensures adequate functioning of the system in addition to the safety of the equipment. The system was tested in luminous intensities ranging from 40 to 500 lux.

3.4 Proposed Algorithm

The developed algorithm consists of two main sections. The first section deals with the Graphical User Interface design, whereas the other section handles TensorFlow custom object detector for hands and nails detection followed by the resolution mapping for mouse control (mouse movement and click based on the three-dimensional position of the index finger's nail). These two sections have been developed separately and then used simultaneously with the help of parallel processing.

Fig. 5. Equipment away from physical harm

3.4.1 PyQt5 graphical user interface

A fully functional GUI, spanning at 1920 x 1080 resolution, as shown in Fig. 6, accepts user clicks on menu items (food and beverages) and appends the items and their respective prices in the lists. It also calculates the total bill on the fly. Any item from the ordered list can be selected and canceled using the 'Cancel Item' push-button, which also results in the reduction of the total bill. Once the order has been finalized by the customer, the 'Place Order' button communicates the ordered items to the kitchen UDP transmission as shown in Fig. 7.

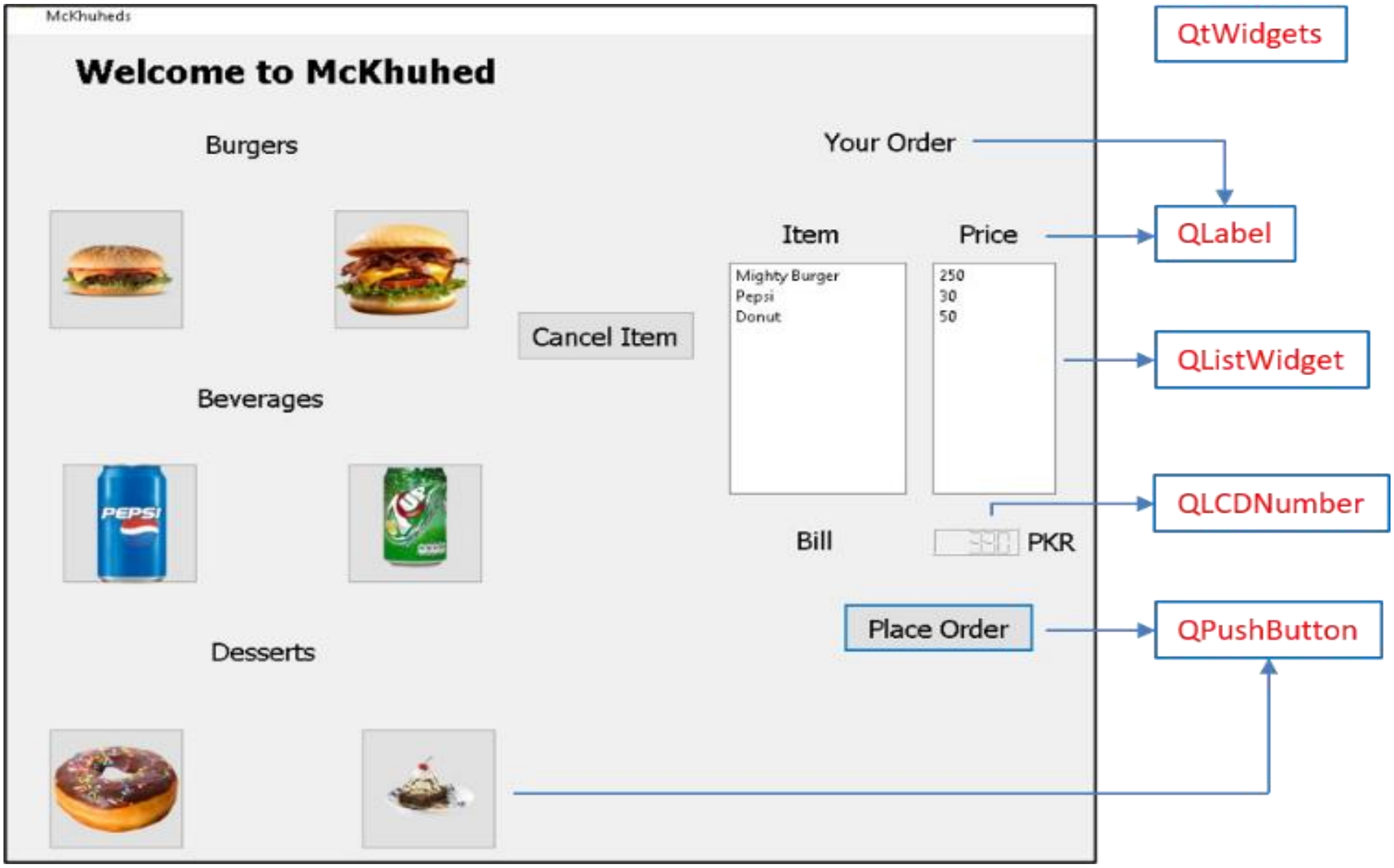


Fig. 6. GUI

The android application used in this research for UDP communication is UDP Sender/Receiver. The libraries used to achieve these tasks are pyqt5 and socket.

3.4.2 TensorFlow custom object detection

We have trained a TensorFlow custom object detection model using a pre-trained TensorFlow model (faster_rcnn_inception_v2_coco) to detect hands and nails. For training the model, we have made our own dataset consists of 1400 images split into training (70%) and testing (20%) and validation (10%) sets, a snippet of it is shown in Fig. 8.

The images are then labeled with 'hand' and 'nail' identifiers as shown in Fig. 9. The .xml files for every image in training and testing datasets created in this step are converted to cumulative .csv files which are then passed on for creating TensorFlow records. With the configured training pipeline, the record files, and the label map (.pbtxt file containing indices and labels), the model is trained and the inference graph is exported which is later used for real-time hand/nail detection. The entire training process can be monitored using TensorBoard. The use of GPU exponentially expedites the process.

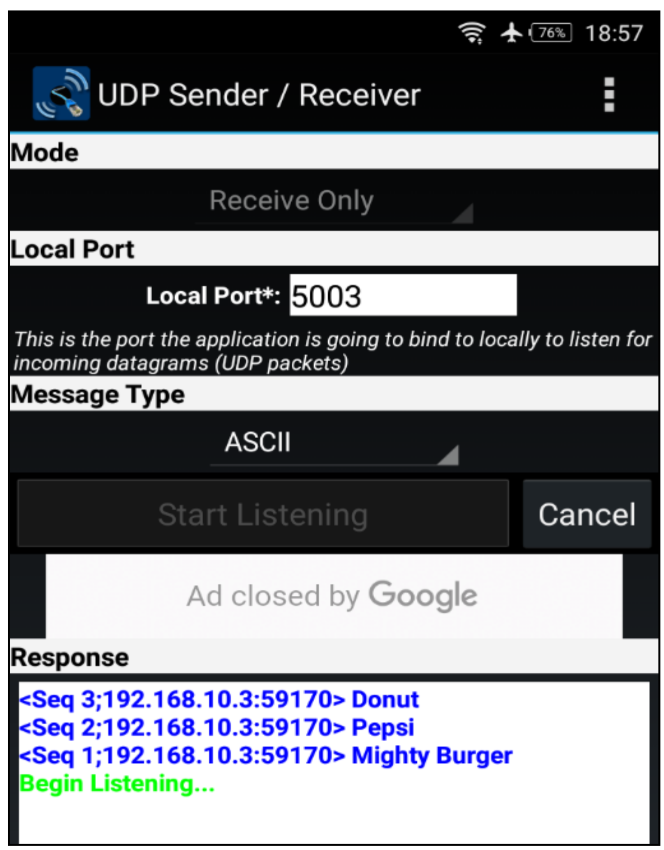


Fig. 7. UDP receiver placed in a kitchen

Fig. 8. Database for training

Fig. 9. Labeled image

The TensorFlow training process is depicted in Fig. 10(a). The total loss graph on the tensor board is given in Fig. 10(b). This is the state after training at approximately 40k steps. The loss value, step number, and time taken per step are shown in Fig. 10(c).

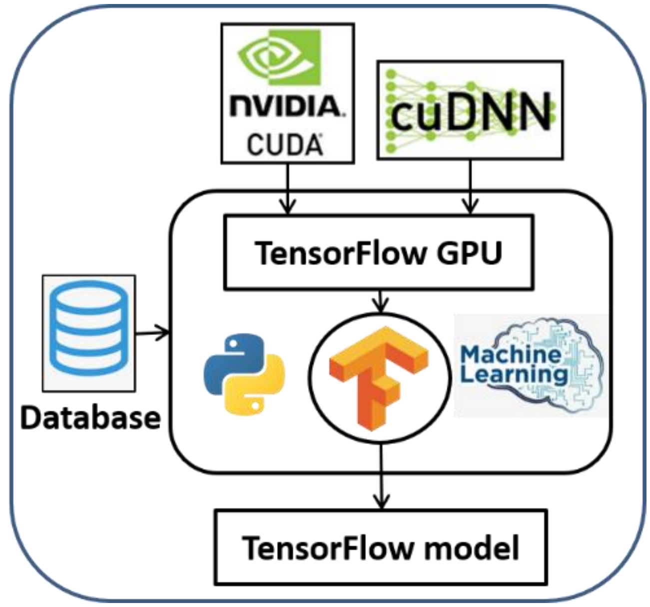


Fig. 10(a). Training block diagram

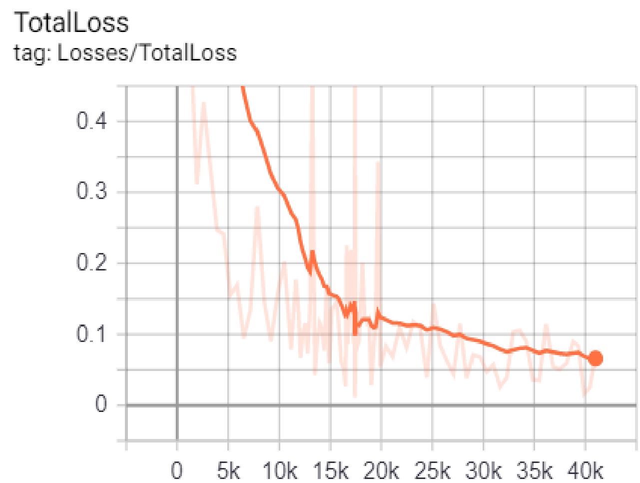


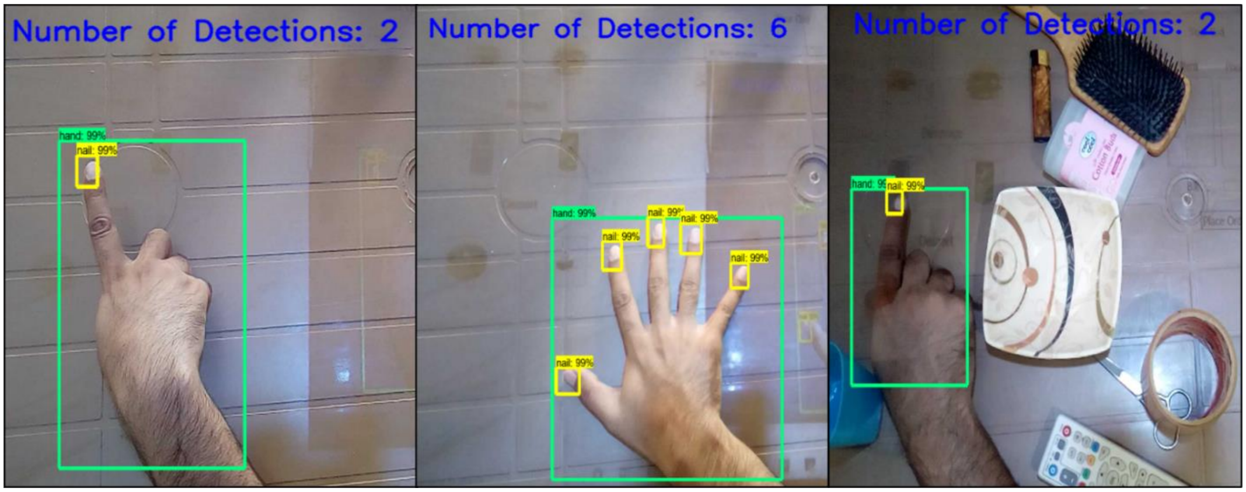
Fig. 10(b). Total Loss (TensorBoard)

```
I0507 13:43:59.940041  9000 learning.py:507] global step 41038: loss = 0.0249 (0.203 sec/step)
INFO:tensorflow:global step 41039: loss = 0.0811 (0.219 sec/step)
I0507 13:44:00.158790  9000 learning.py:507] global step 41039: loss = 0.0811 (0.219 sec/step)
INFO:tensorflow:global step 41040: loss = 0.0638 (0.203 sec/step)
I0507 13:44:00.361816  9000 learning.py:507] global step 41040: loss = 0.0638 (0.203 sec/step)
INFO:tensorflow:global step 41041: loss = 0.0483 (0.222 sec/step)
I0507 13:44:00.584722  9000 learning.py:507] global step 41041: loss = 0.0483 (0.222 sec/step)
INFO:tensorflow:global step 41042: loss = 0.0142 (0.186 sec/step)
I0507 13:44:00.771810  9000 learning.py:507] global step 41042: loss = 0.0142 (0.186 sec/step)
INFO:tensorflow:global step 41043: loss = 0.0660 (0.204 sec/step)
```

Fig. 10(c). Training steps

The results of the nail detector TensorFlow model are given in Fig. 11. Fig. 11(a) shows results under high luminance intensities whereas Fig. 11(b) in low luminance intensities, with clear and cluttered table-top scenarios. This figure also shows the abnormal behaviors of the detector, which have been mostly observed in low-light conditions. The results of the detector are further discussed in section 4.

Luminance: 500 Lux



Gender: Male - **Age:** 35 **Result:** Accurate on Empty Table	**Gender:** Male **- Age:** 35 **Result:** Accurate on Cluttered Table

Fig. 11(a). Hand/Nail detection under bright room light

Luminance: 40 Lux

Gender: Female - **Age:** 27(L) - 5(R) **Result:** Accurate on Empty Table	**Gender:** Female - **Age:** 5 **Result:** Inaccurate on Empty Table (Hand Detection poor accuracy 56%)	**Gender:** Female - **Age:** 27 **Result:** Inaccurate on Cluttered Table (Pair of Scissors appears as a false positive for the nail)

Fig. 11(b). Hand/Nail detection under dim room light

3.4.3 Mouse control

Python package pynput.mouse has been used to control mouse operations. The normalized coordinates of the bounding boxes returned by TensorFlow Object Detector are multiplied by the image's respective dimensions to achieve the coordinates on the actual image.

The 4 entries of each bounding box are [ymin, xmin, ymax, xmax]. The mouse pointer is moved to the center of the nail bounding box calculated using Eq. 1.

$$Centroid = \left(xmin + \frac{xmax-xmin}{2}, + \frac{ymax-ymin}{2}\right) \quad (1)$$

For the mouse click, the Average Nail Area (ANA) with the hand placed on the table is calculated using

Eq. 2. This is computed when the hand is initially placed on the table by the customer at the request of the software for calibration.

$$ANA = \frac{[(xmax_1 - xmin_1)*(ymax_1 - ymin_1)] + \cdots + [(xmax_n - xmin_n)*(ymax_n - ymin_n)]}{n(Number\ of\ Nails\ detected)} \quad (2)$$

This average is later used to detect a table-touch by the finger. When a single hand and nail is later detected at run-time, its Nail Area is computed as given in Eq. 3 and then compared with the Average Nail Area. If the computed Nail Area is less than or equal to the Average Nail Area for 5 consecutive frames, a click is registered and executed at the centroid of the nail coordinates. This is further discussed in section 4.7.

$$NailArea = (xmax - xmin) * (ymax - ymin) \quad (3)$$

3.4.4 System flow with multi-processing

At the start, the user is requested to place his/her hand on the table. The hand image is then captured using either a webcam (module: opencv) or an IP camera (module: requests). This image is then passed on to the TensorFlow object detection model (module: tensorflow) which detects the hands and nails and places bounding boxes around them. The area of each nail's bounding box is then calculated, and the average is computed and labeled as Average Nail Area (ANA). This is the initial calibration step required to detect the touch on the table. After computing ANA, parallel processing initiates. Two processors are simultaneously running (modules: os, multiprocessing), one being the GUI (module: pyqt5) and the other being continuous monitoring of hands and nails. To keep things simple and accurate, mouse movement and click is enabled only if one hand and one fingernail are detected in an image (assuming the customer only uses one finger to point and touch the GUI on the table). If the detected nail's aspect ratio is equal to ANA, this implies that the finger is touching the table (if it is greater than the ANA, this implies that the finger is not touching the table and is hovering somewhere above it, and hence nothing needs to be done). In this case, the mouse pointer is moved to the coordinates of the centroid of the nail bounding box with adequate mapping, and the mouse's left button is clicked. This results in the pressing of the corresponding GUI button and the GUI's on-click functions respond accordingly. When the user has selected the required food items from the GUI using this touch method, he/she finally presses the 'Place Order' button on the GUI which results in the UDP datagram (module: socket) containing the user's order being sent to the kitchen. The entire system flowchart is given in Fig. 12. This is an extremely accurate and fast process as illustrated in section 4.

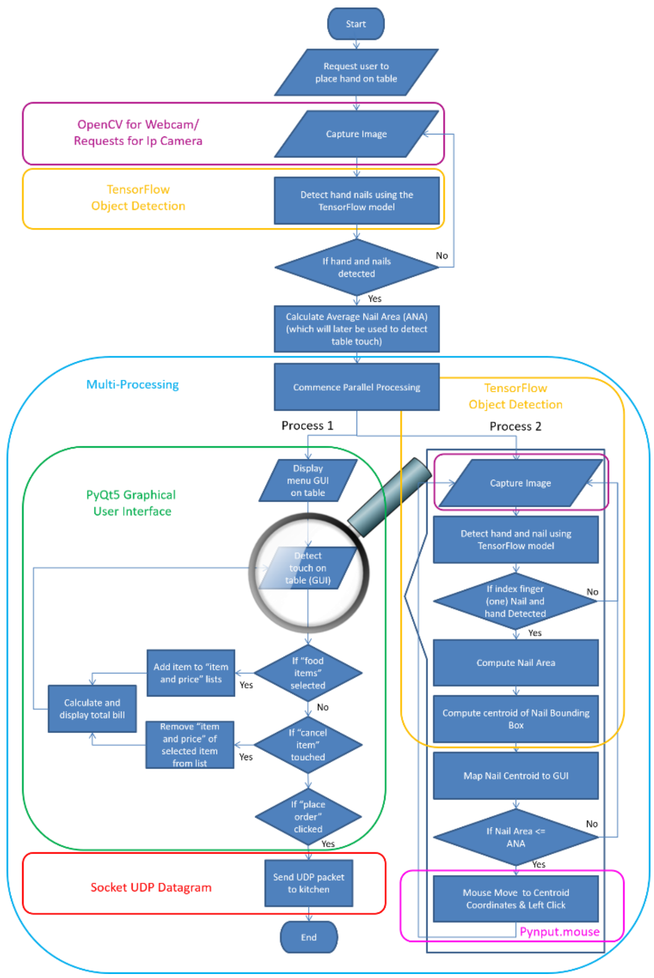


Fig. 12. System flow chart

4. Results and Discussion

This section deals with the performance evaluation of the developed system with respect to speed and accuracy.

4.1 Speed

The TensorFlow model training was carried out on a GPU (Nvidia GeForce GTX 1660Ti 6 GB) and the real-time testing is carried out by deploying the trained model in Raspberry Pi 4 B (4GB CPU). The major time-consuming portion of the software is the nail detector (TensorFlow). Fig. 13 shows the time taken to perform nail detection in 250 consecutive frames. It can be observed that the software is very efficient in performance with respect to speed with a capability to process frames at a rate of 20 fps approximately, keeping in view the average frame processing time of about 50 ms.

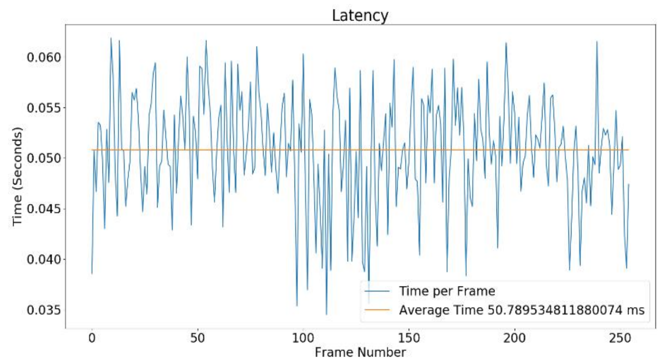


Fig. 13. Processing time per frame

4.2 Hand/Nail Detection

The performance of our custom trained object detector has been tested under different scenarios involving varying room light intensities, cluttered and clear table-tops, and with hands and nails of 10 different subjects. The results are compiled in Table 4. The confidence score set to treat a detected object as a valid hand/nail has been set to 80%, which reduces the chances of false detections. Also, the decision to perform mouse operations is taken only when one hand along with one nail is detected within it. This condition also eliminates the false positives for nails if detected in the cluttered table scenarios.

Table 4

Hand/ Nail Detection Accuracy

Test Subjects: 10
Age Range: 5 to 40 years
Iterations per Subject: 100

Light	Dim Light (40 Lux)	Bright Light (500 Lux)
Clean Table	Attempts: 250 Success: 244 Failure: 06 Accuracy: 97.6%	Attempts: 250 Success: 250 Failure: 00 Accuracy: 100%
Cluttered Table	Attempts: 250 Success: 238 Failure: 12 Accuracy: 95.2%	Attempts: 250 Success: 248 Failure: 02 Accuracy: 99.2%

4.3 Confusion Matrix

The trained model was tested on the validation dataset which contains 140 images, split equally into hands and nails. The performance analysis of the trained model is given by the confusion matrix which is represented in Fig. 14. It was observed that from 70 images of hands-only, 2 False Negative (FN) outcomes were found and from 70 images of nails, only 3 False Positive (FP) outcomes were found which represents that the model has an average of 96% detection accuracy.

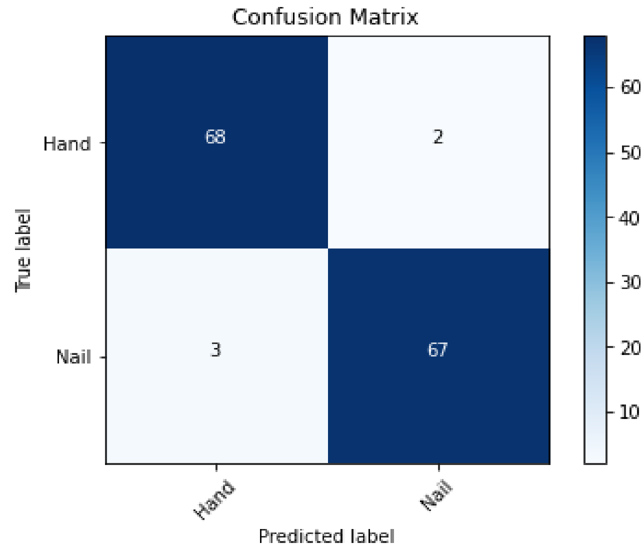


Fig. 14. Confusion matrix

4.4 Classification Report

The classification report is given in Fig. 15 which represents the overall performance parameters such as precision, recall, f1-score, and the accuracy of the trained model on the validation dataset and shows that the overall accuracy of the trained model is 96%.

```
In [9]:  print(classification_report(truth, prediction))
              precision    recall  f1-score   support

        Hand       0.96      0.97      0.96        70
        Nail       0.97      0.96      0.96        70

    accuracy                           0.96       140
   macro avg       0.96      0.96      0.96       140
weighted avg       0.96      0.96      0.96       140
```

Fig. 15. Classification report

4.5 Intersection over Union

The intersection over union (IoU) is an evaluation metric that is used to measure the accuracy of the detection of the trained model. The IoU is defined as the area of the overlap (intersection) between the predicted bounding box, B_p, and the ground-truth bounding box, B_{gt}, divided by the area of their union, which is calculated by using equation 4 and it is illustrated in Fig. 16.

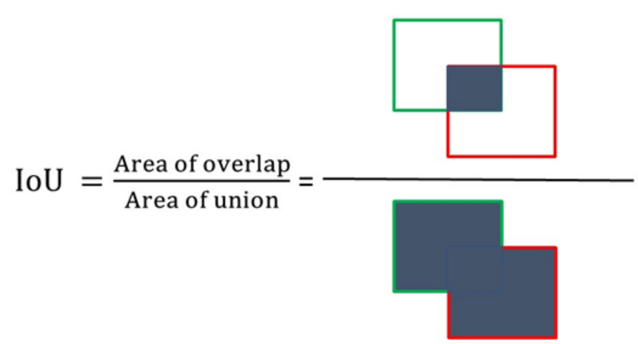


Fig. 16. Intersection Over Union (IoU)

$$IoU = \frac{Area(B_p \cap B_{gt})}{Area\ (B_p \cup B_{gt})}' \quad (4)$$

When the IoU = 1, it shows that two detected bounding boxes have a perfect match and when the IoU = 0, it means that the two bounding boxes have zero match. The value of IoU closer to 1 is considered as better detection. Fig. 17 represents the IoU of the proposed trained model in which the IoU of hands and nails are taken under different lighting environments with different people. The ground truth bounding boxes are represented by green rectangles and predicted bounding boxes are represented by red rectangles. The average predicted IoU = 0.9043, which is closer to 1 and hence the trained model has adequate accuracy for the detection of hand and nail.

Fig. 17 contains 8 images of hands and nails in which the detection is considered as a hand or nail only if its IoU is larger than 0.8 as shown in Table 5.

Fig. 17. IoU of hands and nails

Table 5

IoU values, considering an IoU threshold = 0.8

S. No.	Bounding Box	IoU	IoU>0.80
1.	Hand	0.9886	Yes
2.	Hand	0.9286	Yes
3.	Hand	0.9237	Yes
4.	Hand	0.8626	Yes
5.	Nail	0.9132	Yes
6.	Nail	0.8732	Yes
7.	Nail	0.9229	Yes
8.	Nail	0.8219	Yes

4.6 Mouse Movement

The deviation in the two locations (Nail Centroid and mouse-move) with respect to different positions on the GUI (Region of Interest) was found to be not more than 10 x 10 units, which is pretty decent as a ±10-pixel units' error cannot adversely impact the performance as the minimum area occupied by the (smallest) pushbutton on the GUI is 131 x 41 units. Placement of pushbuttons in the lower-most rows and right-most columns (dead-zone) were avoided in the 1920 x 1080 resolution GUI because in the absence of hand (from the camera view), only detected nail cannot control the mouse as an accuracy enhancement feature. This has been done to avoid invalid movement/click of mouse in the case of a false nail detection.

4.7 Mouse Click

A click is registered only when the Nail Area suggests that the finger has been touching the table for 10 consecutive frames. This is done to avoid false clicks based on abrupt changes in the bounding box dimensions of a nail. It is also a time-efficient process since a frame needs a processing time of 50 ms on an average (as shown in section A) which implies that the total time taken between a finger placed on a GUI button and the click to execute is going to be a quarter of a second which is reasonably fast response time and hence won't annoy the customer with unnecessary delays. In addition, if a false positive hand and nail are detected simultaneously in one frame, the efficiency of the system will not be affected. Fig. 18(a) shows the Nail Area when touching the table, whereas (b) shows the area when the nail is above the table, so a threshold of 1900 worked reasonably well in this particular example with no false clicks.

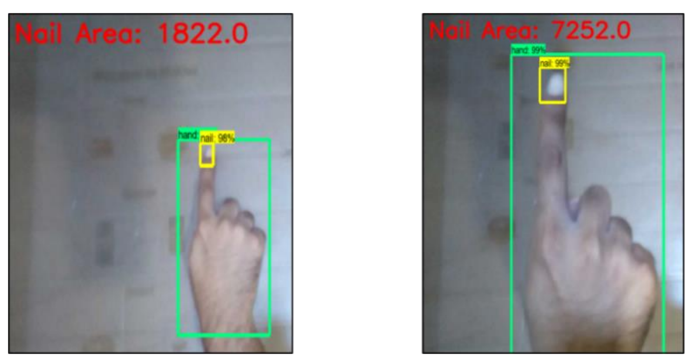


Fig. 18. Nail Area (a) Touching the table (b) Hovering above the table.

4.8 Comparative Analysis of the Proposed System with the Existing Systems

The comparative analysis of the proposed system with the existing systems in terms of accuracy, cost, time and usability is given in the Table 6.

Table 6

Comparative analysis of the proposed system with the existing systems

S. No.	Reference	Accuracy	Cost	Time	Usability
1.	[23]	High	High	Fast	Better
	Proposed	High	Low	Fast	Best
2.	[24]	High	High	Fast	Good
	Proposed	High	Low	Fast	Best
3.	[25]	High	High	Fast	Good
	Proposed	High	Low	Fast	Best
4.	[26]	High	High	Fast	Good
	Proposed	High	Low	Fast	Best

The comparative analysis of the proposed system with the existing systems in market is carried out which shows that mostly the interactive tables available in the market are touch based. If touch-based tables are not properly used, then they may get faulty and they are quite sensitive to water and dust. But the proposed system gives a multimedia based interactive table which cannot get faulty by any rough usage because it is based on the principle of projection of online menu. So, it is found that in terms of usability and life, the proposed system performs the best and moreover it is a cheaper solution.

5. Conclusion

The developed product as a result of this research is a low-cost attraction for restaurant customers which can result in increased revenue turn up at the restaurant. The high speed and accuracy of the software can provide the diners a brisk and pleasant experience by eliminating the delays in waiting for someone to take the order. This also reduces the expenses of the restaurant owners due to a decrease in the investment in tablets or touchscreens traditionally used to take orders, in addition to the reduction in the number of waiters. In addition, running interactive animations and games can later be added to this project to entertain the diners till their food is served. The addition of online billing can also serve as a beneficial feature to this product.

6. Acknowledgements

The authors are thankful to Professor Emeritus Dr. Abdul Qadeer Rajput, Professor Emeritus Dr. Bhawani Shankar Chowdhry, and Dean Faculty of Electrical, Electronic & Computer System Engineering, Dr. Mukhtiar Ali Unar for their support. We are thankful to Dr. Attiya Baqai and Engineer Azam Rafique, Department of Electronic Engineering, MUET, for their valuable input throughout this research.

7. References

[1] M. Dixon, S. E. Kimes, and R. Verma, "Customer preferences for restaurant technology innovations", Cornell Hospitality Report, vol. 9, no. 7, pp. 6–16, 2009.

[2] C. Spence and B. Piqueras-Fiszman, "Technology at the dining table", Flavour, vol. 2, no. 1, pp. 1–13, 2013.

[3] A. Bankar and S. S. Suresh, "Intelligent restaurant - menu ordering system", vol. 5, no. 5, pp. 47–53, 2015.

[4] A. M. Torres, "Electronic Menu and Ordering Application System: A Strategic Tool for Customer Satisfaction and Profit Enhancement", International Journal of u- and e- Service, Science and Technology, vol. 9, no. 4, pp. 401–410, 2016.

[5] R. V. Ravi, N. R. Amrutha, E. Amritha, P. Haneena, and T. Jaseena, "An Android Based Restaurant Automation System with Touch Screen", Proceedings of the 3rd International Conference on Inventive Systems and Control, pp. 438–442, 2019.

[6] S. Khedikar, S. Baghele, V. Muley, and P. Shah, "Digital food ordering system for restaurants", vol. 3, no. 10, pp. 182–185, 2020.

[7] European Food Safety Authority, "Scientific Opinion on the evaluation of allergenic foods and food ingredients 2 for labelling purposes EFSA Panel on Dietetic Products, Nutrition and Allergies (NDA)", EFSA Journal, p. 277, 2014.

[8] N. Tembhekar, P. Singh, A. Mate, K. Gurve, and M. Rahate, "Digital food ordering system for restaurant", Journal of Engineering Research and Application, vol. 10, no. 4, pp. 10–14, 2020.

[9] S. D. Jaffee, L. M. Leventhal, J. Ringenberg, and G. Michael Poor, "Interactive 3D objects, projections, and touchscreens", ACM International Conference Proceeding Series, 2018, doi: 10.1145/3183654.3183669.

[10] K. Khairunnisa, J. Ayob, H. A. W. Mohammed, M. E. Ayob, M. I. Ayob, and M. A. Ayob, "Microsoft Word - 18.doc - Wireless_Food_Ordering_System .PDF", MASAUM Journal of Computing, vol. 1, no. 2, p. 7, 2009.

[11] A. Bhargave, N. Jadhav, A. Joshi, P. Oke, and S. R. Lahane, "Digital ordering system for restaurant using android", International Journal of Scientific and Research Publications, vol. 3, no. 4, pp. 1–7, 2013.

[12] S. Shashikant Tanpure, P. R. Shidankar, and M. M. Joshi, "Automated food ordering system with real-time customer feedback", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 2, p. 2277, 2013.

[13] N. A. Samsudin, S. K. A. Khalid, M. F. A. M. Kohar, Z. Senin, and M. N. Ihkasan, "A customizable wireless food ordering system with realtime customer feedback", IEEE Symposium on Wireless Technology and Applications, pp. 186–191, 2011.

[14] A. N. Badariah, N. N. Azreena, and S. Zulkifli, "Zigbee-based smart ordering system (S.O.S)", International Journal of Computer Trends and Technology, vol. 11, no. 5, pp. 206–209, 2014.

[15] N. M. Z. Hashim, M. R. Anuar, A. Jaafar, M. Z. A. A. Aziz, A. Salleh, and A. S. Ja'afar, "Graphical user interface for wireless patient monitoring system using Zigbee communication", ARPN Journal of Engineering and Applied Sciences, vol. 9, no. 9, pp. 1554–1558, 2014.

[16] S. N. Cheong, H. C. Ling, P. L. Teh, P. K. Ahmed, and W. J. Yap, "Encrypted quick response scheme for hotel check-in and access control system", International Journal of Engineering Business Management, vol. 9, pp. 1–9, 2017.

[17] M. H. A. Wahab, H. A. Kadir, N. Ahmad, A. A. Mutalib, and M. F. M. Mohsin, "Implementation of network-based smart order system", Proceedings - International Symposium on Information Technology 2008, ITSim, vol. 1, pp. 2–8, 2008.

[18] Y. Xu, E. H. Jeong, A. E. Baiomy, and X. Shao, "Investigating onsite restaurant interactive self-service technology (ORISST) use: customer expectations and intentions", International Journal of Contemporary Hospitality Management, vol. 32, no. 10, pp. 3335–3360, 2020.

[19] H. Go, M. Kang, and S. B. C. Suh, "Machine learning of robots in tourism and hospitality: interactive technology acceptance model (iTAM) – cutting edge", Tourism Review, vol. 75, no. 4, pp. 625–636, 2020.

[20] R. Leung and P. T. Loo, "Co-creating interactive dining experiences via interconnected and interoperable smart technology", Asian Journal of Technology Innovation, pp. 1–23, 2020.

[21] X. Wu, D. Gursoy, and M. Zhang, "Effects of social interaction flow on experiential quality, service quality and satisfaction: moderating effects of self-service technologies to reduce employee interruptions", Journal of Hospitality Marketing and Management, vol. 30, no. 5, pp. 571–591, 2021.

[22] K. A. R. Romano, R. S. Masilungan, L. Cacho, and U. M. Melendres, "eCarte: An interactive restaurant menu with feedback collection", Mindanao Journal of Science and Technology, vol. 18, no. 1, pp. 129–144, 2020.

[23] "Interactive Restaurant Table", https://itrestaurant.net/table, [Accessed on 19 Feb 2022].

[24] "21.5 Inch LCD Digital Interactive Smart Touch Screen Coffee Table With Computer Forclass/ Restaurant Game", Alibaba.com, https://www.alibaba.com/product-detail/ Interactive-Table-Restaurant-Interactive-Restaurant-Table_62144422371.html?spm=a2700.7724857.normal_offer.d_title.1d75c99aPlI9TA&s=p, [Accessed on 19 Feb 2022].

[25] "Digital Menu for Restaurant | Tablet Digital Menu", https://www.voolsyscreen.com/industries/ digital-menu-for-restaurant, [Accessed on 19 Feb 2022].

[26] "Candypop Touch Interactive Table, For Indoor, Size/Dimension: 43", https://www.indiamart.com/ proddetail/candypop-touch-interactive-table-20741188348.html?pos=4&pla=n, [Accessed on 19 Feb 2022].