TOUCHLESS ECOSYSTEM USING HAND GESTURES

Soham Patil[1], Sakshi Kodarlikar[2], Shruti Marathe[3], Deepti Vijay Chandran[4]

[1-3]Students, Dept. of Computer Science, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India.
[4]Professor, Dept. of Computer Science, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India.

Abstract: In COVID-19, touchless interfaces are becoming increasingly important. For businesses to resume safely, it is important to reduce or eliminate touchpoints. The pandemic has accelerated the development and deployment of touchless technologies whose value proposition is now re-shaped by the pandemic. Among the top factors driving demand for touchless technology are contactless interaction and health concerns. Humanity has been inundated with touchless interfaces and gesture controls as the pandemic spreads. They enable users to interact with devices without physically touching them. In order to reduce the spread of the virus, COVID-19 has created a need for alternatives to common diurnal practices. With the current pandemic of Coronavirus, it's understandable why some people are unwilling to use touchscreens in public areas. Since touch-free technology is capable of preventing the spread of viruses, it makes sense in this case. In order to minimize virus spread, people tend to use touchless systems as much as possible. These days, airlines use self-service check-in, which uses a touch-screen system. At present, the service is not safe and touchless technology has never existed as much as it does now. In this project, we will be using Tensor Flow to detect hand points and to recognize the different gestures.

Keywords: Hand gestures, Mediapipe, tensorflow.js, hand pose model, touchless ecosystem

I. INTRODUCTION

At present, the world is experiencing the crisis of the COVID - 19 virus, therefore touching any random object can prove to be harmful to anyone. Nowadays, people try to avoid touching anything as much as possible in order to minimize the spread of viruses. Almost all airlines have implemented self-service check-in, which usually revolves around touch-based technology. There are a lot of people who come in contact with the machine in the current scenario, so it's not safe to use that service. Interactive surfaces that use gestures have become widespread over the last few years. The goal of our project is to apply Machine Learning to develop a gesture-based user interface that can be controlled without touching the machine.

Since touch-based systems can spread diseases, they are no longer safe. In addition to ATMs and Airport Check-ins, Touchless ecosystems include restaurant ordering and bill-paying in malls or stores.

In a Check-In kiosk, users can interact with the services through hand gestures. Examples include selecting seats, entering boarding pass data, and printing boarding pass. Security levels of the ecosystem are higher than those of ATMs, which require a high level of security. To access his/her account and make transactions, the user has to first scan the QR code provided by the bank, after which an OTP Verification email has to be cleared, and finally his/her pin has to be entered. Multi-system ecosystems can also be added based on necessity. With the help of the Tensor flow hand pose model, this touch-less hand gesture-based model is very accurate and capable of performing all steps. Our check-in system has therefore been made intuitive and easy-to-use by combining deep learning and an user-friendly interface.

II. LITERATURE SURVEY

The use of Deep Learning over Machine Learning is done because Machine Learning shows output accuracy in the form of 0 or 1, whereas Deep Learning provides better numerical accuracy results that range from 0 to 1. Since the proposed project demands a high need for accuracy in Classification Algorithms it is better to make use of Deep Learning. Just like Data Classification is the most important aspect of Data Science, similarly, Image Classification plays a very big role in Computer Vision. Image Classification first deals with image preprocessing, then it executes image segmentation and then it deals with key feature extraction that helps for finding uniqueness in different trained images, and lastly executes matching identification.

The implementation consists of four main steps: 1. Image Enhancement and Segmentation 2. Orientation Detection 3. Feature Extraction 4. Classification. This work was focused on the above four categories but the main limitation was that the change of color was happening very rapidly by the change in the different lighting conditions, which may cause an error or even failures. For example, due to insufficient light conditions, the existence of the hand area is not detected but the non-skin regions are mistaken for the hand area because of the same color. Involves three main steps for the hand gesture recognition system: 1. Segmentation 2. Feature Representation 3. Recognition Techniques. The system is based on Hand gesture recognition by modeling

the hand in the spatial domain. For modeling, the system uses various geometric and non-geometric 2D and 3D models.

Hand gesture segmentation based on skin color may be interrupted by objects with similar colors to the hands, such as the human face. For this reason, hand gesture segmentation using model features is adopted after detecting skin color. A large number of hand gesture samples are then used to extract the hand gesture features, which are then used to train a classifier to distinguish between the hand area and the rest of the image. The paper adopts the AdaBoost classifier based on Haar feature. Haar feature reflects the image grayscale value change. The feature model is composed of black and white rectangles. In the feature model, the pixel sums under white areas are subtracted from the pixel sums under the black areas. and express the feature value of objects. As a result of traversing all rectangle feature areas when calculating Haar feature values, many calculations will have to be repeated, which wastes a lot of time. By combining two integral images, you can quickly calculate the rectangle feature and it is the main concept to convert the original image into an integral image.

III. MODULES/FRAMEWORKS USED

A) HANDPOSE MODEL

Handpose is a tensorflow.js model based on the google mediapipe model, which detects hands in real-time using TensorFlow. Based on an input image or video stream, it draws a bounding box around the hand, and then, after processing, it returns twenty-one 3D landmarks pointing to the hand's features. Different landmarks define different locations on each finger joint and on the palm. The key points are then used to train different gestures, such as pinching.

Fig. 1 Different 3D landmarks plotted on a hand

B) DJANGO

Django is a Python framework for creating MVT pages in a web application. With Django, we can create multiple applications that can be used to design multiple systems in our ecosystem. The project connects the frontend to the database and provides a user - friendly admin page to manage and view the data.

C) TENSORFLOW.JS

TensorFlow.js is a machine learning framework used to train and deploy models in web frameworks or node.js. It provides tools, libraries, and community resources to help developers build and deploy ML-powered applications. Handpose is implemented in tensorflow.js, which can be easily implemented and used in web frameworks such as Django.

IV. METHODOLOGY

A) HOW THE HANDPOSE MODEL WORKS

With TensorFlow, 21 3D landmarks of a hand are detected from a live video stream using the handpose model. Handpose is an application that is based on the Google media pipe hands and is enabled in TensorFlow by embedding them in the web-based tensorflow.js. The model predicts whether an input contains a hand. This model returns the location of each finger joint along with the palm if it contains, as well as coordinates for the bounding box around the hand. Rather than training a hand detector, we train a palm detector since the estimation of bounding boxes of rigid objects like palms and fists is more straightforward than handling articulated fingers. The non-maximum suppression algorithm also works well for two-hand self-occlusions, such as handshakes, since palms are smaller objects. In addition, palms are often modeled using square bounding boxes (anchors in ML terminology), which ignore other aspect ratios, reducing the number of anchors by a factor of 3 or 4. Furthermore, the encoder-decoder feature extractor is used to make larger scenes contextually aware of small objects as well. Lastly, we minimize the focal loss during training to support an outsized number of anchors due to the high scale variance. This is then used to simulate a mouse cursor using the centroid of the hand. As the centroid moves along the screen, the x and y distances are recorded and a cursor appears on the screen that the user can use to navigate around the UI. A click function is trained by using the Handsfree module, where variations of the hand are captured and stored, and two pinches of the thumb and index finger are made, which can then be implemented within the web application as a new gesture. In order to simulate a mouse click, the cursor is used in conjunction with the index finger and thumb being pinched twice within an 80-millisecond interval to implement the click function.

B) HAND DETECTION

In this system, the user uses hand gestures to perform tasks that would otherwise need touch input to perform, with the system being focused on user-data interaction. In the handheld sensor, the hand position of each marker's fingers is tracked in real-time, and gestures are used to interact based on that location. A bounding box is plotted around the hand using the handpose model. In order to plot the various 3D landmarks on the hand, the bounding box image is further processed. You can then use these key points to train different gestures using the graph on the hand.

V. SECURITY FEATURES

Multiple levels of security are necessary to secure systems such as ATMs or malls where transactions will take place. In order to make an ecosystem secure, the following security precautions can be taken. To make a system more secure, one or several measures can be implemented.

A) QR CODE

A unique QR code containing the user's information is provided to the user in this type of security. In order to log in to the user, the system accesses the information when the card is scanned. QR codes have the following advantages:

1. Fast scanning and Omnidirectional: QR codes can be scanned Omni directionally from any angle, so the scanner does not have to be situated as per the code symbol, allowing for much faster scanning.

2. Smaller size: QR codes are much smaller than barcodes. It is possible to store the same amount of data with QR Codes in only a tenth of the space of 1-D barcodes.

3. Massive Storage Capacity: QR codes offer massive storage capacity.

4. Data Types: QR codes can handle binary data, numbers, alphabetic characters, and Japanese, Chinese, or Korean characters.

5. Wide Range of Uses: QR codes have many potential uses. They also can be implemented to extend the experience of user in-store, restaurants, websites, etc.

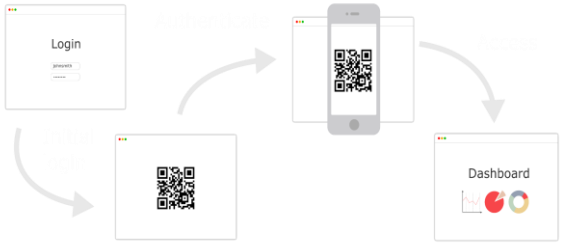

Fig. 2 QR Code Authentication

B) OTP VERIFICATION

A one-time password will be sent to the user by email or phone. A user is logged into the system upon entering the correct OTP. One-time passwords are generated by the OTP system creator by using the user's secret passphrase, as well as a seed entered from the server, to rehash a secure hash function multiple times. As soon as a successful authentication is performed, one secure hash function repetition is removed. An entirely new password sequence is created as a result. A secure hash function is calculated on time and compared with the one-time password selected prior to the submission by the creator on the server.

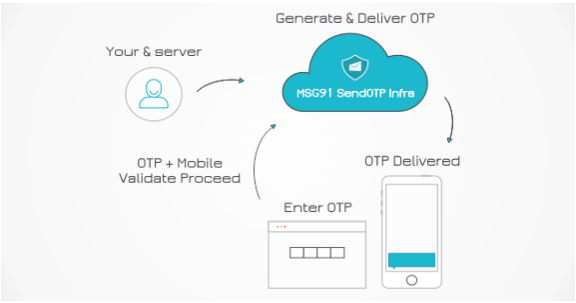

Fig. 3 OTP verification working

C) FACE DETECTION

The user's face must be detected and recognized by a high-quality camera for this type of security. When the two faces match, the system logs in to the user. Using a database of stored faces, the input image is compared with the picture in order to locate the person in it. The method for face recognition is influenced by factors such as shape, size, pose, occlusion, and illumination. It has two main applications: basic and advanced. There are several factors that define major face recognition, such as the width of the nose, the width of the eyes, the depth and angle of the jaw, the height of cheekbones, as well as the distance between the eyes.

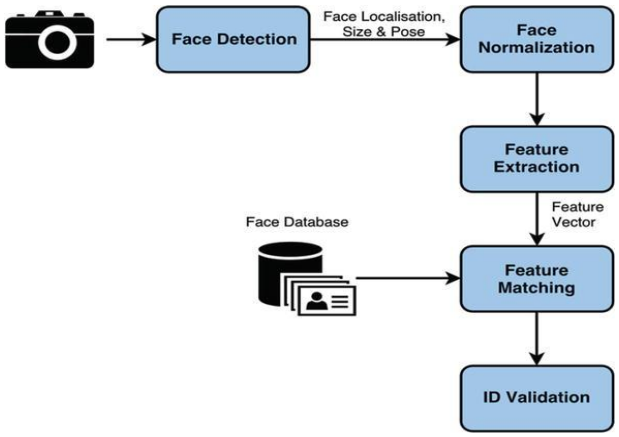

Fig. 4 Face Detection working

A unique numerical code is created by using this information. The numerical code allows a system to compare one image with another and distinguish how they are similar. In face recognition, video camera signals and preexisting images are used to determine the provenance of the image. This can be done by detecting faces, extracting features, and then recognizing people.

D) PALM VEIN AUTHENTICATION

Authentication is based on vein patterns on individuals' palms, and a user's palm pattern must match one in the system's database to be verified. A near-infrared light maps the vein lines on the hand using near-infrared light. This information is then saved by the system. Authentication data is stored in the palm, so it would be difficult to forge the palm. This is the most secure way of scanning.

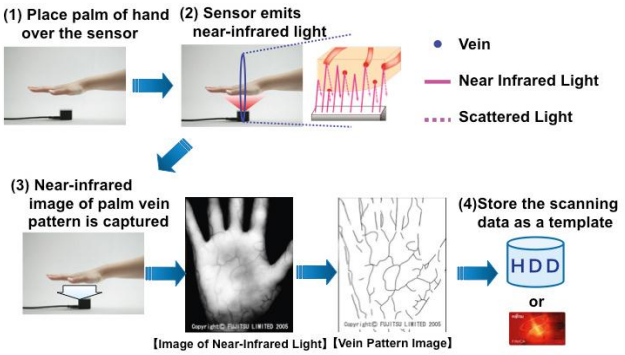

Fig. 5 Palm vein authentication system

E) IRIS SCANNING

To obtain high-resolution clear images of the iris, a high-quality scanner along with a camera are needed, which are then compared to the user's data gathered in the system. Logging in to the user is automated if the iris is matched. Acquisition, segmentation, normalization, feature extraction, and feature matching are the main steps in iris recognition. To recognize an iris, methods use machine learning models like DCNN, and CNN; the result from the method is compared to the results from the database.

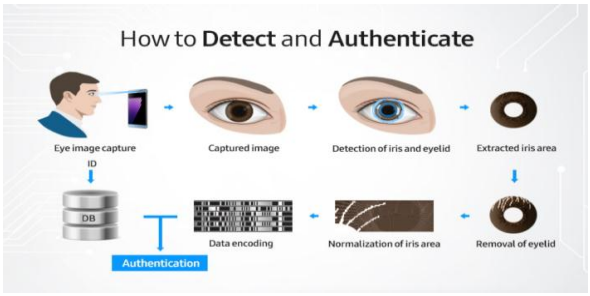

Fig. 6 Iris scanning

VI. SYSTEMS

A) AIRPORT CHECK-IN

Using this system, customers can print their boarding passes instead of using traditional airport check-in kiosks. The user can print out their boarding passes, select their desired seats on the plane, and print their boarding passes with this touchless system. Simple hand gestures can be used to type the booking ID, select the seats, and navigate the UI. There is an on-screen keyboard that is designed to be used with gestures so it can be used with ease.

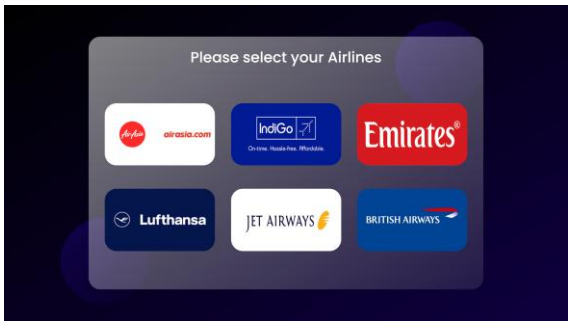

Fig. 7 Airport Checkin

B) TOUCHLESS ATM

The second part of the ecosystem that we implemented, replacing the ATM machine, which is one of the most frequently used systems. Several security features have been implemented to enhance the level of security provided by this Touchless system.

Additionally, this system offers multiple levels of security, including QR code, OTP verification, Face detection, and more. In order to avoid card contact, the QR code is implemented thus providing each customer with a unique code. This code can then be scanned to retrieve all of the

customer's information. The user is requested to enter a one-time password (OTP) after scanning via an email or phone number provided on the scanning page. The OTP is sent via email or phone number provided by the user. The ATM requires a security code to be entered by the user to further enhance security. The system can be further enhanced with additional security features as needed. This system and other ones in the ecosystem can be utilized by using the same easy-to-use gestures.

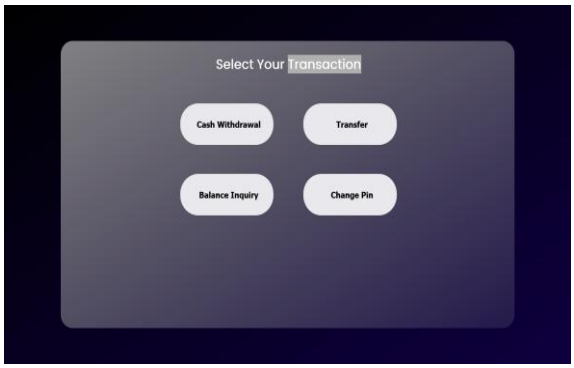

Fig. 8 Touchless ATM

C) RESTAURANT/CAFE ORDERING

This system is made to be used in restaurants or cafes by the customers to place orders. The customers can use the same hand gestures as in other touchless systems to navigate the UI and place orders. The system is based on Django as the backend which keeps a record of all the orders placed by the customers along with the current status of their order. When the order is ready, the customer is notified of the same. Razor pay is also implemented as a payment option in the system which can be used by the customers to pay for their orders. It supports multiple payment options along with a secure and user-friendly way to complete the payment. This system can be used to replace the currently used touch-based kiosks in many places for ordering food.

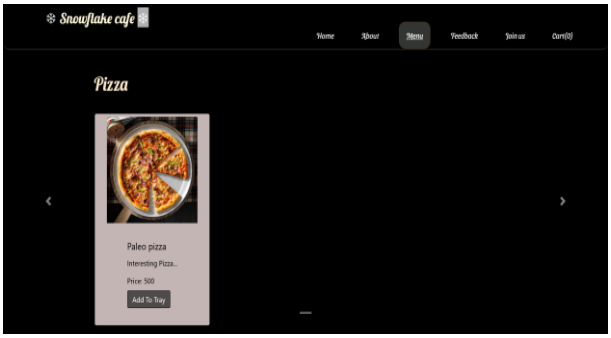

Fig. 9 Restaurent Ordering

D) MOVIE TICKET BOOKING

This system is designed to be used in malls or theaters for booking movie tickets by using touchless systems. Customers can select the movies from the home page which can be updated regularly by the theaters. Along with booking the tickets, customers are provided with a UI to select the seats they desire from the ones that are available. Seats can easily be selected or deselected as per requirement and further can be used to pay for the number of tickets. Customers are given the option to use razor pay which supports multiple payment options along with a secure and user-friendly way to complete the payment. After successful payment, the tickets can be collected from the kiosk. This can be used in place of the traditional ways to book a ticket from the theaters.

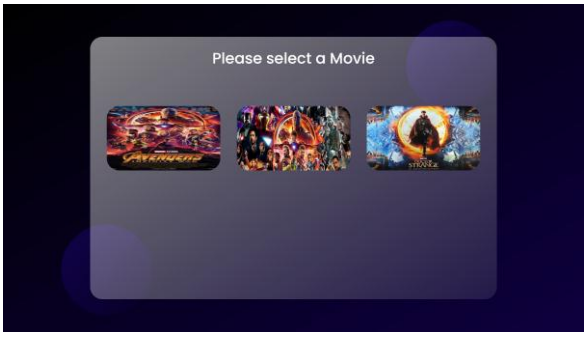

Fig. 10 Movie ticket booking

VII. RESULT

The handpose model we are using gives highly accurate results even in many different lighting conditions which solves the major issue in hand recognition where the results produced are inaccurate if there is a frequent change in lighting conditions or shift in colors. The gestures are also tracked accurately and produce the required results.

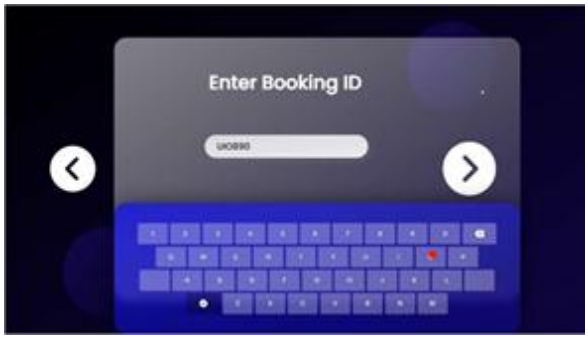

Fig. 11 Typing using the hand gestures

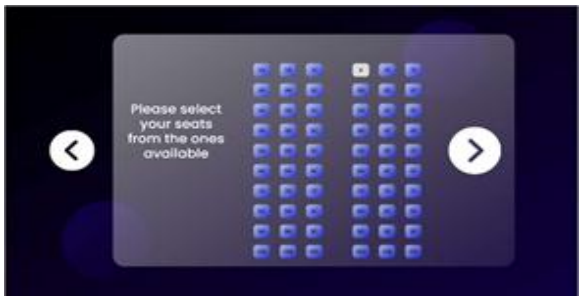

Fig. 12 Selecting seats using the hand gestures

Multiple security systems can be implemented to make the payments and transactions more secure and easy to perform. We have implemented a QR code that can be scanned to verify the user details. Further, an OTP verification is implemented where the user gets a code on his provided mobile number or email which can be used to login into the account.

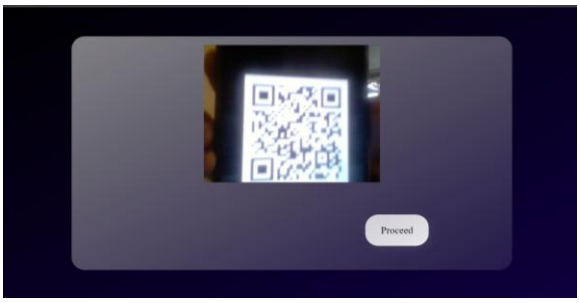

Fig.13 Scanning the QR Code

VIII. CONCLUSION

Touch-based systems aren't safe anymore, as they can contribute to the spread of disease. This Touchless ecosystem ensures that all of its gesture-based systems are safer to use as compared to the systems. With the help of hand gestures, the user can easily navigate the UI and perform the same operations as in traditional systems but touch-free. This touch-less hand gesture-based model is very accurate and capable of performing all the steps with the help of the Tensor flow hand pose model along with robust security systems that can be implemented to make the transactions etc as secure as possible. Thus, using deep learning models, multiple security systems, and the very user-friendly UI we have successfully achieved our objectives and implemented an ecosystem that can perform the tasks of the most used systems by users touch-free.

IX. FUTURE WORK

Any new system can be added to the ecosystem which needs touch-free interaction. Adding more security features like Iris scanning, Face detection, etc. Optimizing the algorithm to work faster.

REFERENCE

[1] Dharmik Bhanushali, Dipanshu Someshwar, Vismay Chaudhari, Ms. Bhargavi Dalal (2020). Hand Gestures Recognition using DeepLearning.

[2] B., Abhishek & Krishi, Kanya & M., Meghana & Daaniyaal, Mohammed & S., Anupama. (2020). Hand gesture recognition using machine learning algorithms. Computer Science and Information Technologies. 1. 116-120. 10.11591/csit.v1i3.p116-120.

[3] Fan Zhang Valentin Bazarevsky Andrey Vakunov Andrei Tkachenka George Sung Chuo-Ling Chang Matthias Grundmann.(2020). MediaPipe Hands: On-device Real-time Hand Tracking

[4] Nhu-Tai Do , Soo-Hyung Kim *, Hyung- Jeong Yang and Guee-Sang Lee(2020).Robust Hand Shape Features for Dynamic Hand Gesture Recognition Using Multi-Level Feature LSTM

[5] M. H. Eldefrawy, K. Alghathbar and M. K. Khan, "OTP-Based Two-Factor Authentication Using Mobile Phones," 2011 Eighth International Conference on Information Technology: New Generations, 2011, pp. 327-331, doi: 10.1109/ITNG.2011.64.

[6] Shally, & Aujla, Gagangeet & Aujla, Singh. (2014). A REVIEW OF ONE-TIME PASSWORD MOBILE VERIFICATION. International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR). 4. 113-118.

[7] Alapetite, Alexandre. (2010). Dynamic 2D-barcodes for multi-device Web session migration including mobile phones. Personal and Ubiquitous Computing. 14. 45-52. 10.1007/s00779-009-0228-5.

[8] Nithya, Alice & Lakshmi, C.. (2015). Iris recognition techniques: A Literature Survey. International Journal of Applied Engineering Research. 10. 32525-32546.

[9] Jain, Tarshi & Kumar, Rajendra. (2019). A STUDY OF VEIN RECOGNITION SYSTEM. Acta Informatica Malaysia. 3. 13-15. 10.26480/aim.01.2019.13.15.

[10] Guimarães, Nuno. (1991). Building generic user interface tools. ACM SIGPLAN Notices. 26. 89-96. 10.1145/117954.117961.

[11] Kumar, Ashu & Kaur, Amandeep & Kumar, Munish. (2019). Face Detection Techniques: A Review. Artificial Intelligence Review. 52. 10.1007/s10462-018-9650-2.