IOT BASED SMART SECURITY SOLUTION FOR UNIQUE AUTHENTICATION

1. Samarpan Das [mailme.samarpandas@gmail.com], 2. Mili Chaturvedi [mc4399@srmist.edu.in],
3. Matura Ganesh [mg6779@srmist.edu.in], 4. Mrs. A. Bhavani [bhavania@srmist.edu.in]
Department of Electronics and Communication Engineering,
SRM Institute of Science and Technology, Chennai, India.

***Abstract*- To ensure the confidentiality of our home stay, office, banks etc., we use several security measures. Thus, making security the primary facet of our approach. This can help ensure that we keep unauthorized access out of our way. The use of biometrics-based authentication like fingerprint recognition and face recognition for unique authentication is of paramount importance as it provides a high level of reliability. It is very difficult to imitate fingerprints and faces of an individual. This paper proposes an IOT based unique authentication of fingerprint and face for opening locks in smart homes. To strengthen the traditional door lock system at home or authentication system a fingerprint scanner and a camera is installed. The database will keep a record of anyone trying to access the lock system and the buzzer will go off when the match is not found. The system is controlled by a Raspberry pi 3 processor and an arduino module. Access to the door will only be provided when a match for both face and fingerprint is found.**

***Keywords*- Fingerprint sensor, Raspberry Pi-3 B+, Raspberry pi camera, SVM Algorithm, CNN, Inception Neural Network version-1, Inception Neural Network version-2, VNC viewer, Wnetwatcher.**

I. INTRODUCTION

Although there exist many kinds of smart home lock systems such as keypad lock systems where a code is punched in and access is provided if the code is correct, or the voice recognition lock system, there still is scope to potentially enhance these security solutions. Having two factor authentication ensures a better approach to providing a smarter solution for user authentication.

One of main inspirations for us to experiment with a new standard was an existing 2 factor authentication system approach for verification. This model included finger ridges and voice recognition to authenticate the user.

In this paper a model that includes fingerprint and face recognition for authentication will be developed.

Including two biometric factors makes the system even more secure as it is difficult to impersonate both the face and fingerprint of an individual with accuracy. Thus, it will reduce the access risk to unauthorized individuals by fraud means.

This project has three phases. The first phase includes fingerprint sensor code and testing of it with a database created consisting of a certain number of datasets.

The second phase includes creating face recognition models, and comparing them to get the most efficient model so that we can implement them and test with the dataset first to integrate it with the hardware so that working with real time data can be tested.

The third and last phase includes integration of all the components together to test the final working of the model.

II. IMPLEMENTATION

Real-world worries about the safety of material assets and identity theft are becoming more prevalent each day. There are already mechanisms in place, such as pins to punch in to access a secure workplace or biological patterns to authenticate or keys for locker systems. However, these existing models have issues such as forgotten pins, carrying a physical device (key or RFID card) which could be lost. Our two-factor authentication system irradiates the need to carry any physical chip - based device by the user or the need for the user to remember a pin / password, thus reducing chances of human error.

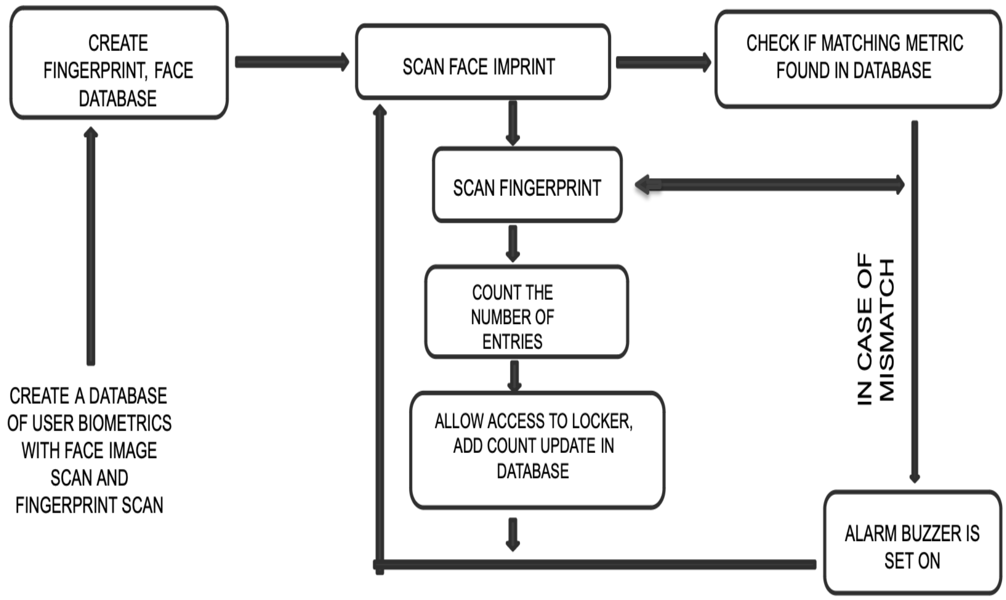

Fig. 1: Proposed Workflow

The project aims to propose an improved version of existing security systems by increasing the number of layers in authentication and by using enhanced authentication systems. The addition of face recognition along with existing systems of fingerprint. Implement and identify efficient machine learning and deep learning models against the target dataset and find which performs best by projecting the accuracy scores.

III. HARDWARE

1. Fingerprint Sensor:

   This optical fingerprint sensor contains a number of LEDs that are used by the CMOS or CCD sensors to illuminate both the fingertip areas and the reflected light waves. Currently available sensor versions have the ability of detecting wet fingers and come in sizes as small as 1mm. It takes a finger impression and compares it to previously scanned patterns.

2. Raspberry pi 3 B+ :

   On-board is a Broadcom 1.4GHz quad-core 64-bit processor with low energy. 4 USB 2.0 ports are present. The power supply required is via Micro USB or Gpio at 5V/2.5A DC. The interface for the graphics card is already built-in. It contains an expanded 40-pin GPIO header, IEEE 802.11 B/G/N/Ac wireless LAN at 2.4GHz and 2.4GHz, and Bluetooth 4.2.Its use is in word processing, spreadsheets, high definition video, games, and programming due to the quick processing it offers.

   With the aid of this module, the fingerprint and the face database are interfaced to meet our needs. Python is used for programming it. This module is connected to the buzzer, fingerprint reader, face reader, and locking system.

3. Alarm / Buzzer :

   Any inconsistency in the entries is signaled by the buzzer. The signal's primary function is to transform audio to sound. It frequently uses DC voltage for electricity. The color is black. The range of frequencies is 3,300 Hz. operating voltages of 3 to 24 volts direct current.

4. Central Lock Actuator :

   When an interruption is activated, a low-voltage solenoid in a solenoid lock pulls the latch back into the door. The latch will remain in place until the interruption is activated. The raspberry pi 4 model b module is attached to the solenoid lock. The Raspberry Pi's GPIO Pins can supply external power, which is utilized to activate the lock via the relay. The lock is unlocked when the fingerprint and face of the individual attempting to get access match the pre-stored fingerprint and face.

5. Arduino UNO :

   This is an ATmega328-based, compact, and breadboard-friendly board.

   Instead of a regular USB cable, it functions using a mini-B version.

   It can be used to interact with a computer, another Arduino, or another microcontroller, among other things.

   It is used in our system to communicate with the Raspberry Pi model that is being used.

6. Raspberry Pi Camera :

In order to use the camera for face recognition, desired picture quality must be achieved. The camera module might be easily connected to the Raspberry Pi board using an existing connector. It has a 5 MP lens and is equipped with CSI (Camera Serial Interface).

IV. HARDWARE AND SOFTWARE INTERFACING

To interface hardware with the software we have used softwares. First the code to save the database of fingerprints was completed in C++. The face recognition code that is written in python is integrated with the raspberry pi 3 B+ model.

For the purpose of programming in the Raspberry pi we used the vnc viewer software.

A delay for 4 seconds is provided for the fingerprint recognition part. A delay of 5 seconds is provided for face recognition. The arduino board plays the role of a database storage system. The raspberry pi camera and the fingerprint sensor are connected to the usb ports in the raspberry pi board.

The processing happens in the raspberry pi board.

Using basic linux commands the calls are given in the raspberry pi terminal window to input data and get the final output.

The lock is connected to the arduino board, after the processing in raspberry pi, when the command goes to the arduino depending upon the match or not the lock and buzzer connected to it will respond.

The final result of the connections may look like in the picture attached below.

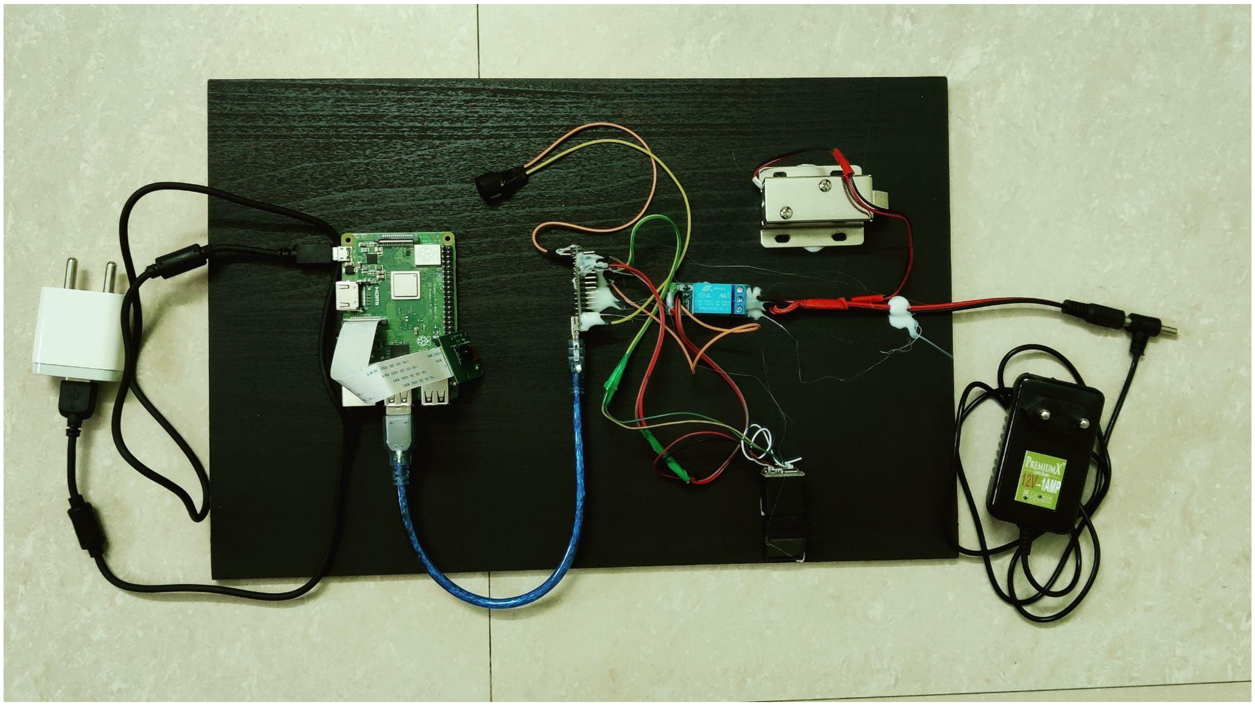

fig.2- Final hardware connection illustration

V. RESULTS

After trying to implement the Algorithms of Machine Learning and Deep Learning, it was found that the Deep learning models performed better and gave better accuracy results.

For illustrating the observation below attached are the performance of all the algorithms implemented in a tabular and graphical form and at last to project the performance of each model we have a single comparison bar graph.

Epoch	Loss	Accuracy	Validation loss	Validation accuracy
1/50	0.539	0.848	0.3475	0.913
10/50	0.662	0.898	0.332	0.919
30/50	0.422	0.987	0.408	0.926

40/50	0.329	0.990	0.454	0.921
50/50	0.019	0.993	0.387	0.928

Table-1.Epoch Accuracy data for Inception Neural Network V1

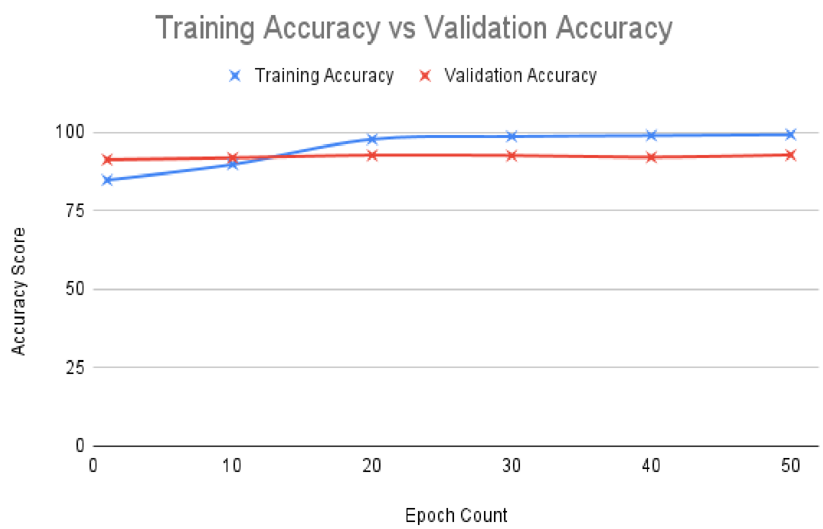

fig.3(a)- Graph plotted for Training accuracy vs Validation accuracy of Inception Neural Network V1

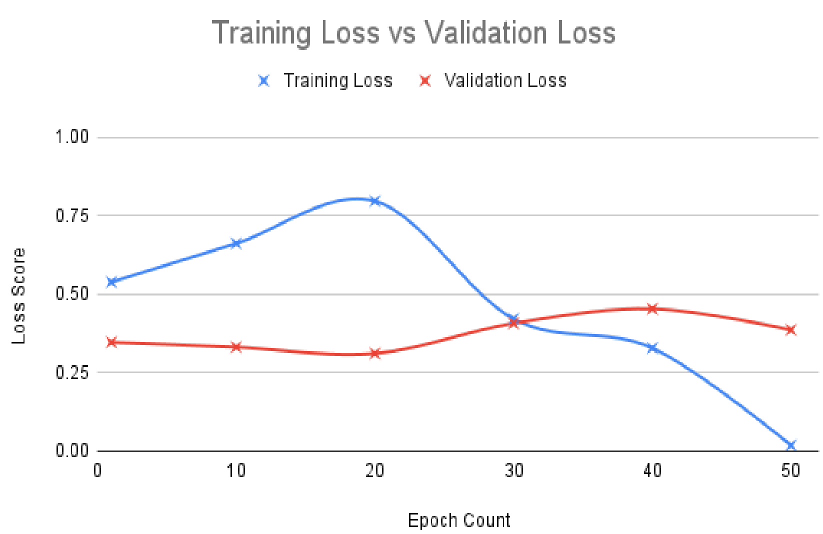

fig.3(b)- Graph plotted for Training loss vs Validation loss of Inception Neural Network V1

Epoch	Loss	Accuracy	Validation loss	Validation accuracy
1/50	0.8508	0.7522	0.3478	0.8957
10/50	0.0661	0.9794	0.2007	0.9459
20/50	0.0325	0.9902	0.0.2518	0.9459
30/50	0.0271	0.9915	0.2128	0.9454
40/50	0.0240	0.9937	0.2035	0.9666
50/50	0.0238	0.9946	0.2259	0.9577

Table-2- Epoch Accuracy data for Inception Neural Network V2

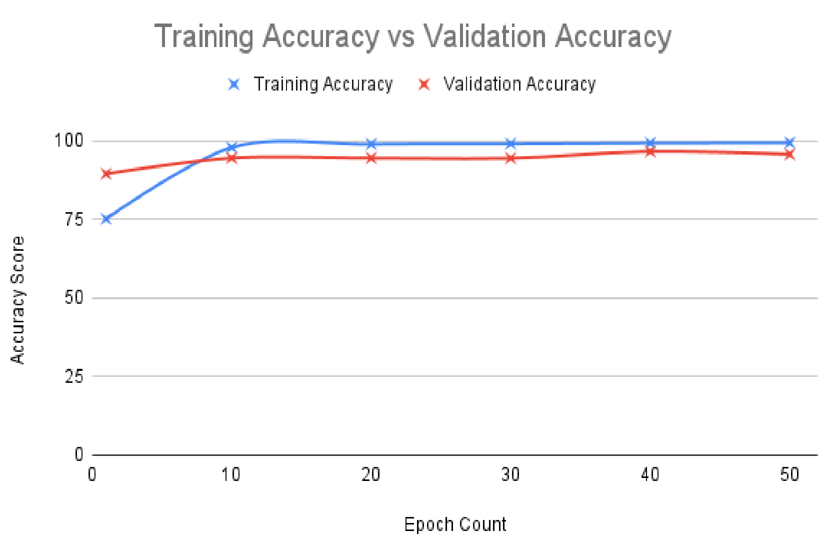

fig.4(a) - Graph plotted for Training accuracy vs Validation accuracy of Inception Neural Network V2

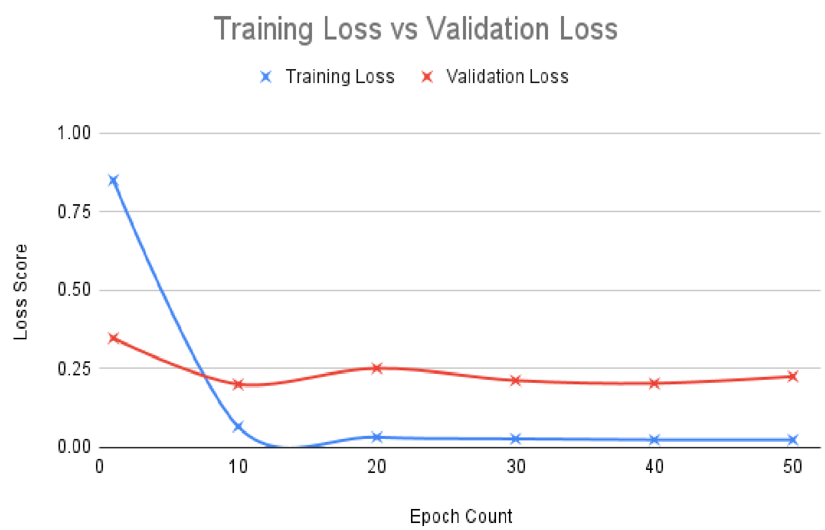

fig.4(b) - Graph plotted for Training loss vs Validation loss of Inception Neural Network V2

Epoch	Loss	Accuracy	Validation loss	Validation accuracy
1/50	0.6101	0.7993	1.9106	0.7125
10/50	0.0457	0.9834	0.8488	0.8303
30/50	0.0085	0.9971	0.9031	0.8210
40/50	0.0049	0.9977	0.8564	0.8395
50/50	0.0031	0.9977	0.9171	0.8314

Table-3- Epoch Accuracy data for Support Vector Machine

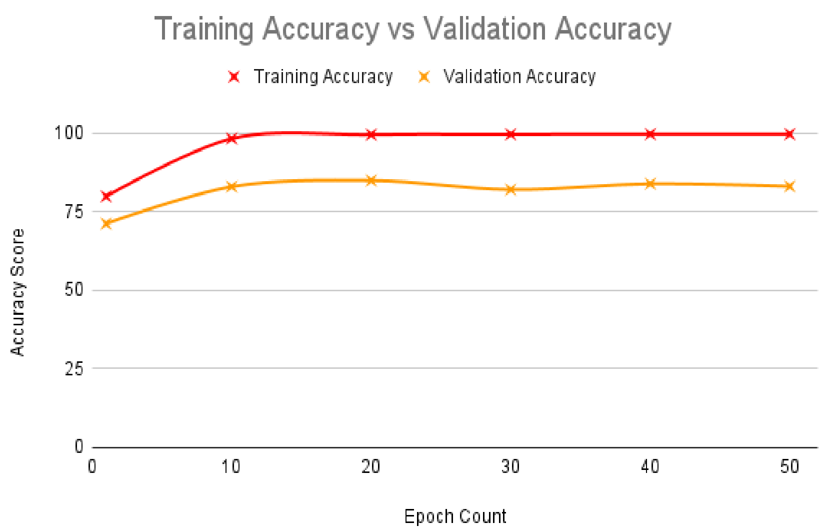

fig.5(a)- Graph plotted for Support Vector Machine Training accuracy vs Validation accuracy

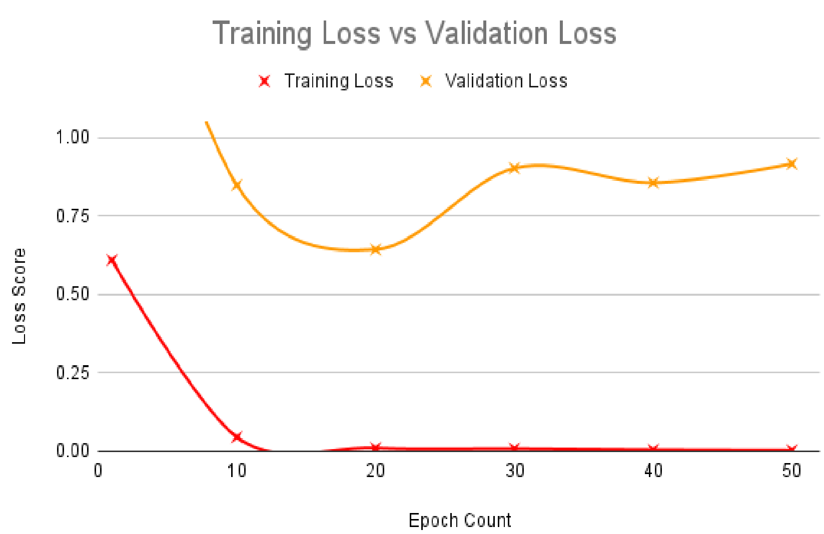

fig.5(b)- Graph plotted for Training loss vs Validation loss of Support Vector Machine

Epoch	Loss	Accuracy	Validation loss	Validation accuracy
1/50	2.4870	0.3895	2.6569	0.4366
10/50	0.0892	0.9742	1.5141	0.6598

20/50	0.0288	0.9916	1.6755	0.6613
30/50	0.0174	0.9943	1.8187	0.6593
40/50	0.0146	0.9953	2.0363	0.6481
50/50	0.0104	0.9964	2.1411	0.6270

Table-4- Epoch Accuracy data for Decision Tree Classifier

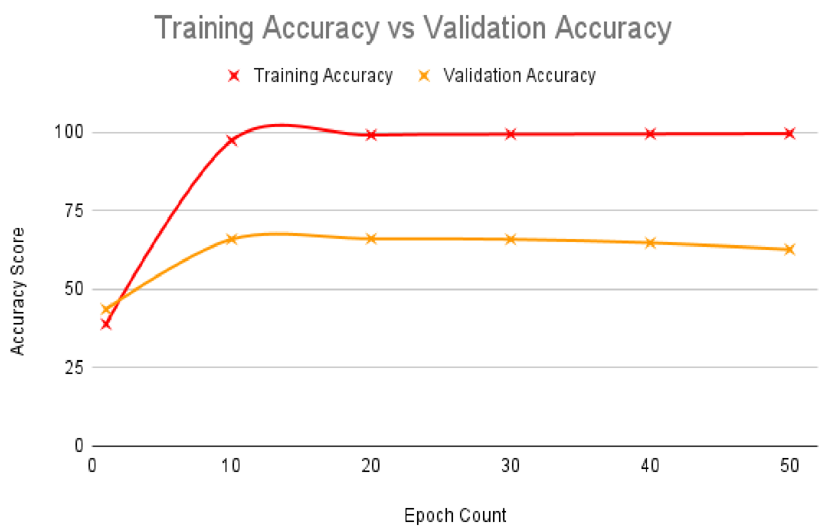

fig. 6(a)- Graph plotted for Training accuracy vs Validation accuracy of Decision Tree Classifier

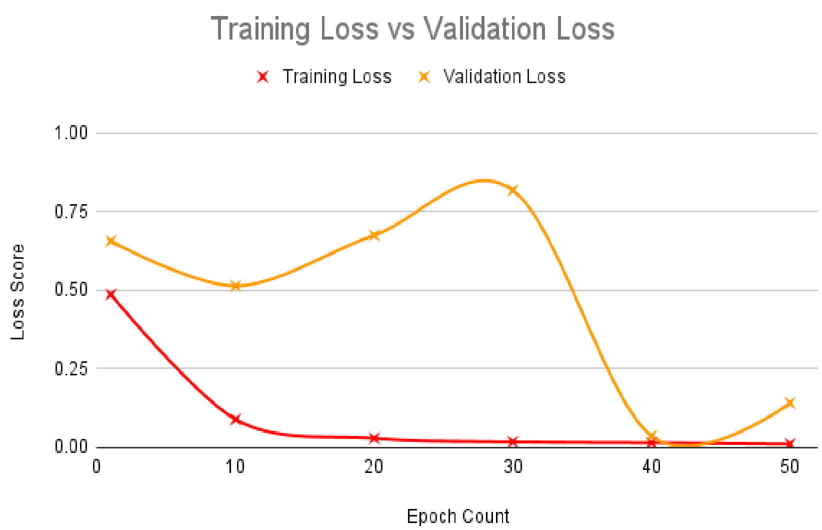

fig.6(b)- Graph plotted for Training loss vs Validation loss of Decision Tree Classifier

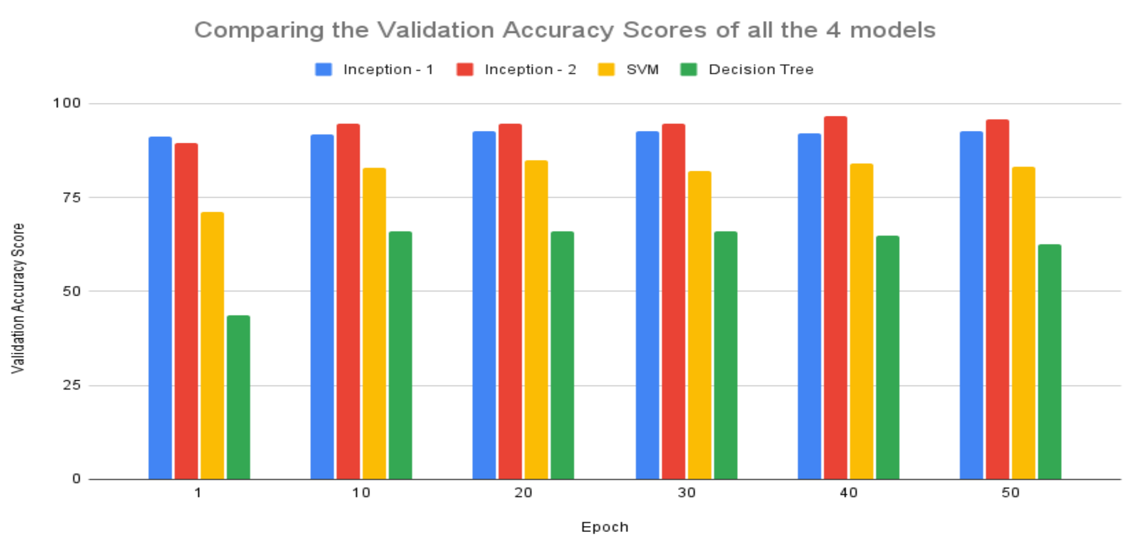

fig.7 - Graph for compassion of the performance of the 4 models examined.

By plotting the graphs we compared the accuracy of these two selected deep learning models so as to get a final model to use and implement for our project.

From the figures 3, 4, 5, 6 and tables 1, 2, 3, 4 we can conclude that the version 2 of the Inception Neural Network with an accuracy of 95.77% is best for use.

VI. CONCLUSION AND FUTURE WORKS

After all the connections are successfully plotted, the working of the model will be tested. The expected output is; after the face image of the user is taken as an input, if it matches with the database, then the fingerprint will be sensed and if a match is found, the lock will be unlocked. If a match is not found then the buzzer would go off.

The efficiency and the accuracy of the system would however only become clear after the test is completed.

This system can also include a third gateway for authentication by including an OTP verification as the last step.

The system then would work as Fingerprint matching, Face recognition, OTP verification. If all three is a match then the lock would open, if not a message can be sent to the user on the registered mobile number, that an attempt was made to access the system.

VII. REFERENCES

[1] X. Yang, S. Yang, J. Liu, C. Wang, Y. Chen and N. Saxena, "Enabling Finger-Touch-Based Mobile User Authentication via Physical Vibrations on IoT Devices," in IEEE Transactions on Mobile Computing, vol. 21, no. 10, pp. 3565-3580, 1 Oct. 2022, doi: 10.1109/TMC.2021.3057083.

[2] D. Breitenbacher, I. Homoliak, Y. L. Aung, Y. Elovici and N. O. Tippenhauer, HADES-IoT: A Practical and Effective Host-Based Anomaly Detection System for IoT Devices (Extended Version)," in IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9640-9658, 15 June 15, 2022, doi: 10.1109/JIOT.2021.3135789.

[3] C. Choi and J. Choi, "Ontology-Based Security Context Reasoning for Power IoT-Cloud Security Service," in IEEE Access, vol. 7, pp. 110510-110517, 2019, doi: 10.1109/ACCESS.2019.2933859.

[4] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf and Y. A. Bangash, "An In-Depth

Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10250-10276, Oct. 2020, doi: 10.1109/JIOT.2020.2997651.

[5] J. Srinivas, A. K. Das, M. Wazid and A. V. Vasilakos, "Designing Secure User Authentication Protocol for Big Data Collection in IoT-Based Intelligent Transportation System," in IEEE Internet of Things Journal, vol. 8, no. 9, pp. 7727-7744, 1 May 1, 2021, doi: 10.1109/JIOT.2020.3040938

[6] S. Batool, A. Hassan, M. A. K. Khattak, A. Shahzad and M. U. Farooq, "IoTAuth: IoT Sensor Data Analytics for User Authentication Using Discriminative Feature Analysis," in *IEEE Access*, vol. 10, pp. 59115-59124, 2022, doi: 10.1109/ACCESS.2022.3178635.

[7] L. Zhang, L. Zhang and D. Zhang, "Finger-knuckle-print: A new biometric identifier," *2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, 2009, pp. 1981-1984, doi: 10.1109/ICIP.2009.5413734.

[8] M. Rukhiran, S. Wong-In and P. Netinant, "IoT-Based Biometric Recognition Systems in Education for Identity Verification Services: Quality Assessment Approach," in *IEEE Access*, vol. 11, pp. 22767-22787, 2023, doi: 10.1109/ACCESS.2023.3253024.

[9] J. J. Robertson, R. M. Guest, S. J. Elliott and K. O'Connor, "A Framework for Biometric and Interaction Performance Assessment of Automated Border Control Processes," in *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 983-993, Dec. 2017, doi: 10.1109/THMS.2016.2611822.

[10] K. M. Alashik and R. Yildirim, "Human Identity Verification From Biometric Dorsal Hand Vein Images Using the DL-GAN Method," in *IEEE Access*, vol. 9, pp. 74194-74208, 2021, doi: 10.1109/ACCESS.2021.3076756.