National College of Ireland

IMPLEMENTATION OF TOUCH-LESS HAND GESTURE RECOGNITION ATM BASED ON DEEP LEARNING APPROACH

MSc Research Project
Data Analytics

Ibrahim Rinub Babu
Student ID: X19207387

School of Computing
National College of Ireland

Supervisor: Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ibrahim Rinub Babu
Student ID:	X19207387
Programme:	Data Analytics
Year:	2021-2022
Module:	MSc Research Project
Supervisor:	Giovani Estrada
Submission Due Date:	31/01/2022
Project Title:	IMPLEMENTATION OF TOUCH-LESS HAND GESTURE RECOGNITION ATM BASED ON DEEP LEARNING APPROACH
Word Count:	3488
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

IMPLEMENTATION OF TOUCH-LESS HAND GESTURE RECOGNITION ATM BASED ON DEEP LEARNING APPROACH

Ibrahim Rinub Babu
X19207387

Abstract

Many people are concerned that they will contract coronavirus disease if they use ATM's during the outbreak. Corona and other air-born viruses have been shown to linger on ATM screens or buttons for up to 72 hours, with the potential to trigger a pandemic outbreak. The goal of this research is to review all the current architectures and methodologies to come up with a better Deep learning neural network architecture for hand gesture recognition. The best model is combined with an ATM online application simulator, which allows all financial transactions to be completed using gestures and financial transactions to be authenticated using the user's facial recognition, allowing for complete contactless management of the ATM application. In this study, Custom CNN models and transfer learning models such as VGG-16, ResNets50, and MobileNet were created and evaluated using various metrics with the precision of 92.99 percent, 99.04 percent, 99.14 percent, and 99.65 percent. The ResNet50 with the processing time of 26ms with the real-time validation precision of 100

1 Introduction

Researchers are working on numerous critical technologies to make public places safer during the COVID-19 pandemic. Despite the fact that new variants of illness have already emerged in most of the areas, many cities and countries throughout the world have reopened their doors. Many scientists are attempting to reduce surface pollution in public places. Because airborne respiratory droplets that settle on surfaces are the most common source of infection. This research concentrates on enhancing the gesture control precision of DNN algorithms and tracking position accuracy, as well as integrating it into ATMs to offer a contact-less experience. This innovative touch-less gesture recognition technology which gets the input as hand gestures or signs makes the automated teller machine improve the human-computer interaction for financial transactions in a better and safe way. To perform specific financial tasks such as verifying the user account information, withdrawal of cash, or depositing cash to user account, user's doesn't need to contact the screen or controls buttons, which lowers virus transmission.. A Deep Neural Network algorithm is utilized to locate and recognize the hand in order to make users feel natural and at ease while utilizing the touch-less hand-gesture recognition ATM service. Although it does not have totally connected layers, the Deep Neural Network architecture is developed on the transfer learning model based on CNN composition.

Because of its high quality, which has a positive impact on image classification tasks, and its widespread use in areas where transfer learning could aid improve outcomes, the transfer learning model was chosen as the foundation network. The classification of numerous hand gestures is subsequently performed in order to complete the duties in the ATM.

1.1 Motivation and Project Background

Glove-based device control integration introduction marked the beginning of hand gesture detection control for computer and other electronic devices. Researchers discovered that sign language-inspired movements can be utilized to provide simple orders for a computer interface. With the introduction of many precise accelerometers, infrared cameras, and even fibre-optic bend-sensors, this steadily evolved. Some of these advancements in glove-based systems finally allowed for computer vision-based recognition without the use of any sensors on the glove.

This work by (Chae et al.; 2019)) offers a wearable wireless surface electromyogram incorporated gateway that acquires and recognizes signals using a proposed analogy pseudo-wavelet pre-processor. This wearable interface wirelessly transfers real-time EMG and APWP data to a laptop or a sensor hub, where they are further analysed in the MATLAB environment using the pseudo-wavelet transform. A wearable tech module and its wireless system prototype have been built based on this ROIC to distinguish five different forms of real-time hand-gesture motions, with battery usage reduced even lower by using low-power technologies.

This research by (Sharma et al.; 2019) describes a recognition system that can assist a blind person. In this study, a hand gesture recognition system and a facial recognition system have been created, allowing for the completion of a variety of activities. Various actions, such as turning on the fan or lights, can be done based on the motion recognized. Face detection and recognition are carried out using Haar Cascade Classifiers and the LBPH recognizer, respectively. Vibrant images are made by extracting dynamic images from a dynamic video and then processing them with specific algorithms. This technique has been used to recognize and identify a variety of hand motions and human faces.

A soldier identification intelligent recognition system was intended to address the issue of combatants being unable to converse face to face while performing combat tasks. The system first performs data standardization and endpoint detection on the acquired gesture data. The processed data in this study is used to extract characteristics such as mean, maximum average, and RMS values. Following that, in this study by (Zhu et al.; 2019), the dynamic time warping technique is employed to determine the relationship between the test gesture and the real-time action, and a recognition outcome for the retrieved feature is obtained. The experimental results reveal that the system has high recognition accuracy, excellent real-time performance, and high adaptability to individual variances.

1.2 Research Question

RQ: "How far a deep neural network architecture be used to produce a accurate contact-free hand gesture detection control interface for ATMs using images generated by OpenCV?".

Sub RQ: "How exact the hand gesture detection DNN algorithm built in this research would perform precisely classifying all the gestures to perform specific task while

deploying it in real time using OpenCV ?"

This experimentation concentrates to improve the precision and location tracking accuracy of the hand gesture detection deep neural network algorithms, as well as to integrate it into OpenCV to provide a contact-free environment. After choosing the optimum and appropriate model for real-time application, the ATM prototype will be displayed as a web application.

1.3 Research Objectives

The figure 1 illustrates the objective of the project in each development phase.

Objectives	Description
Objective 1	A critical review of the present literature work done in the same field and identified gaps (2014-2019)
Objective 2	Modofied methodology approach used
Objective 3	Architecture and process flow diagram
Objective 4	Data Generated using OpenCV
Objective 4.1	Data preprocessing and Augmentation
Objective 4.2	Exploratory Data Analysis to get insight about feature for handgesture and face authentication dataset
Objective 5	Implementation, Evaluation of DNN model
Objective 5.1	Integration the developed DNN model and the web application using flask framework
Objective 6	To perform face authentication and handgesture classification to carry out financial transaction in real time ATM web application prototype.

Figure 1: Project management objective

2 Critical Review of Hand gesture recognition using different approaches

2.1 Introduction

Hand gesture recognition technologies enable people to communicate with machines that are more mortal, creative, easy to use. Contact with the human-machine, gestures, immersive game technology, and other applications are all covered by hand gesture recognition. There are several research in progress to improvise the precision of hand gesture

recognition. This study by (Nair et al.; 2020) uses region filling in the gesture or item of interest. The moments are employed in the KNN algorithm for feature extraction and classification.

This study by (Wang and Rai; 2020) hopes to merge the greatest aspects of image recognition with an embedded control system in Jie Wang's research. The system then completes the image pre-processing and recognizes the gesture data, sending the identification results back to the Raspberry Pi via Bluetooth. Five alternative motions are available for controlling the car's forward, backward, turn left, turn right, and stop. The pre-processed gesture picture's geometric moment feature and histogram feature are extracted, and the first two components of the seven feature components, as well as the compactness of the pre-processed gesture image, are chosen as feature values. A neural network is built using Python's Anaconda distribution.

2.2 Literature Review of hand gesture recognition based on statistical and PCA approaches

The proposed combination method is illustrated by (Golovanov et al.; 2020) research paper. There are also descriptions of the employed hand detector and gesture recognition algorithms. Equations for evaluating prospective performance increases are presented, as well as experimental findings. The suggested system is put to the test using publicly available gesture databases as well as video sequences created by the authors. The experimental results support theoretical predictions and show the advantages of the suggested gesture recognition system architecture.

This study by (Ahuja and Singh; 2015) proposes a technique for database-driven hand gesture recognition based on a skin color model approach and thresholding approach, as well as an effective template matching using PCA, for vision-based hand gesture recognition. Thresholding is used to distinguish the foreground and background in the next stage. Finally, for recognition, a template-based matching technique is constructed utilizing Principal Component Analysis. The system is put to the test with four movements, each with five possible positions, and 20 photos per gesture from four subjects.

2.3 Literature Review of approaches based on Trajectory recognition

In this publication by (Kavyasree et al.; 2020), the optical flow is used to track information sights in the video and record the monitored motion as photographs. Trajectory-based images refer to the practice of recording actions as images as described above. The pictures are then categorized using a VGG16 architecture in the next phase. Because of its inherent ability to recover trustworthy and precise information for categorization, this deep learning method is used for feature extraction and detection.

This article by (Kabir et al.; 2019) offers information needed for a successful HRI interaction in a sensitive live application. In this experimented hand sign recognition system, palm photographs as well as structural framework information are collected for the Kinect sensor. The video sequence is separated from the palm region of both hands. This is done using the skin color segmentation approach. The feature detection detects the open, close postures of the hand by calculating the region of the palm and the ultimate angle. The proposed model has a 94.5 percent prediction accuracy on average.

2.4 Literature Review of Hand-Gesture Recognition Based on EMG

This research by (Chanu et al.; 2017) demonstrates two alternative computer vision hand sign detection systems as well as a data glove-based solution. The static hand gesture recognition technique and the real-time hand gesture recognition approach are both vision-based techniques. The input photos are processed using MATLAB software in both ways, and no dataset is used for decision making, making this system more accurate than existing system designs. The glove used in the data glove approach has five flex sensors. All three procedures are tested on ten participants and the results are compared to determine which is the most accurate. Both vision-based strategies were shown more accuracy when compared to data glove-based strategy accurate.

The fundamental goal of (Jaramillo and Benalcázar; 2017) research is to develop a real-time hand gesture recognition model for diverse applications in medicine and engineering that has a higher recognition accuracy and a larger number of gestures to recognize than real-time models proposed in the scientific literature. Acquisition of EMG data, pre-processing, feature extraction, classification, and post-processing are the five stages of the proposed model. The author uses machine learning identification methodology to classify hand gestures with EMG utilizing Machine Learning.

2.5 Literature Review of Gesture Recognition using a unified framework

This paper by (Alon et al.; 2009) proposed a unified architecture for performing picture geographical segmentation, periodic segmentation, and detection all at the same time. This method can be applied to video frames with busy, dynamic backgrounds. This classifier-based trimming method enables for speedier and more exact rejection of subpar gesture modeling solutions. Two gesture recognition methods are used in this study work for a successful outcome. The first is a continuous real-time digit identification system that determines whether users are wearing shorter sleeves and whether there are less than four individuals in the backdrop (at least in one of the test sets). Users might find fascinating motions in a continuous video framework in the second phase.

2.6 Literature Review of Gesture Recognition using a Transfer Learning methodology

The main objectives of the scheme are to create a new dataset of 2D single hand gestures belonging to 27 classes that were collected from Google search, YouTube videos, and professional artists under staged environment constraints, to investigate the effectiveness of Convolutional Neural Networks in identifying and classifying single hand gestures by optimizing hyperparameters, and to evaluate the impacts of transfer learning and double transfer learning models on the dataset are explained in this article (Parameshwaran et al.; 2019). This architecture's base model is heavily designed to boost the model's processing time when implemented in a real-time application.

This research (Venkatesh et al.; 2021) shows how to use a fine-tuned MobileNet Convolutional Neural Network model to capture and categorize fruit kinds. With 88 layers, the original MobileNet model is more computationally demanding and contains more parameters. The entirely adhered layer can be removed. The MobileNet CNN model,

which has been fine-tuned, performs effectively, with improved fruit categorization accuracy and lower computing costs. The validation accuracy of the suggested model is around 98.60 percent, with a loss rate of about 0.38 percent.

The modulation pattern of the constellation is identified in this article (Tian and Chen; 2019) utilizing a variety of standard CNN, including VGG-16, VGG-19, Inception, Xception, and Resnet-50. Experiments demonstrate that the Resnet-50 network recognizes constellations the best. The Resnet-50 convolutional neural network is then used to distinguish the two modulation modes as well as the constellation's signal-to-noise ratio concurrently.

The proposed custom CNN and pre-trained CNN such as mobilnetv2 are analyzed in this article (Bousbai and Merah; 2019) using several performance factors in this work for identifying hand motions using ASL data set. Both models are trained and tested using 1815 images separated by color and set on a black background, as well as five participants' static hand motions, which include changes in scale, lighting, and noise. The proposed custom convolution neural network architecture had a recognition accuracy of 98.9% respectively, according to the data. A CNN augmented using Pre-trained CNN techniques performed 97.06 percent better.

2.7 Conclusion

After reviewing all the approaches it is evident that the EMG and unified learning has limitation while implementing it in the real-time application. Since to achieve gesture recognition using Electromyography the user should wear a glove that classifies different minute electric pulse signals which is the major drawback for this approach. And some of the approaches using custom CNN and transfer learning model, the authors mainly focus on achieving accuracy with the random dataset from different sources without focussing on the training period and the processing time of classifying the gestures. Because while implementing the model in the real-time application the processing time of the deep learning architecture to give out the response is important. In some research, the model architecture is very heavy that if it's exported as h5 to get integrated with real-time applications such as web or mobile app's, the size will exceed more than 600Mb which can make it impossible to deploy it. In this research, all these limitations will be addressed by focusing on deploying a model that will show better validation accuracy, less processing time, and be easy to integrate with the real-time application

3 Methodology Approach

3.1 Introduction

The three key components of the methodology must be discussed to understand the Research Methodology of hand gesture recognition ATM. The importance of comprehending the data cannot be overstated. This part will look at the data used to train the model to classify gestures used to complete certain financial transactions and to authenticate the financial transaction using facial recognition generated programmatically using OpenCV, as well as the information contained in the data. The data must next be pre-processed, converted, and enhanced before being used to train the DNN model. For a better understanding of how the DNN model integrated with ATMs web application prototype to provide a contactless experience for the users by executing all financial transactions

via hand gestures, see Fig 3. These steps are broken down into subsections. In this research, the project development phase and the critical decision making in this study, KDD was utilized to manage the project management and development phases of the complex process of identifying interesting, valid, and valuable patterns from the generated image collection. It usually entails several steps, including using OpenCV to create and select a subset of data from a randomly generated image dataset, data pre-processing and transformation (data augmentation), selecting an appropriate Data Mining technique for extracting patterns from the image dataset for classification, and feature extraction. After all of these processes are done, the model is fine-tuned and executed, evaluated and interpreted, and deployed for real-time use.

3.2 Data Collection

3.2.1 Hand gesture Dataset

The OpenCV framework is used to create the hand gesture data set for training the DNN model to perform certain tasks in the ATM. The webcam's input video stream source for hand movements is read and parsed into individual images. It collects 600 photos of each hand gesture from the background for a single execution of code with the wait key one. The display window will appear, and the photographs will begin to be captured. The hand gesture dataset is created in the directory where the path was allocated. Six directories were created according to the label allocated to them such as "Back", "Cancel", "Check_Account_Balance", "Deposit_cash", "Next", "Withdraw_Cash", which can be seen in the figure 2.

Hand Gesture Recognition ATM	
GESTURES	**PERFORMANCE**
Check_Account_Balance	Shows the amount of money you have available in your checking or savings account.
Deposit_cash	To deposit cash at a bank using ATM
Withdraw_Cash	To withdraw cash from a bank using ATM
Back	Go back in the financial transaction process
Next	Go forward in the financial transaction process
Cancel	Cancel the Financial transaction process

Figure 2: Hand geture dataset

3.2.2 Face Authentication Dataset

More than 13,000 pictures of faces were created programmatically using openCV in this data set. The name of the individual pictured has been labeled on each image. In the data

set, there are 400 different images of two of the people pictured. This dataset will be used to train the model for employing facial recognition to authenticate financial transactions at ATMs. The developed DNN model evaluates many aspects of your face, such as eye placement and nose width, and integrates all this information into a unique code that identifies and is used for authentication.

3.3 Data Pre-processing

Image pre-processing is used to improve the quality of programmatically generated data for neural networks and to get improved model prediction in real-time without misclassification loss. Some of the techniques that are carried out in this research for data pre-processing for the purpose of improving inputs for neural networks are interpolation, rescale, dilation, opening, closing, and checking if the data format such as RGB channel is coming first or last and then input shape will be fed accordingly. Keras API is utilized to perform all the pre-processing using inbuilt parameters.

3.4 Data Augmentation

Data augmentation is performed to prevent the neural network from learning irrelevant features and to diversify and increase the generated datasets. Since the model will be integrated in real-time and it is a financial transaction-based machine, side-on transformations techniques are carried out. The learning rate is kept constant when training the Custom DNN and the proposed transfer learning network for feature extraction to see the change in accuracy only due to augmentation. When we utilize augmentation to train the network, a new transformation of each hand gesture and Face Authentication image is generated for each epoch. As a result, the model sees the same number of photos in each epoch as there are in the original training data, even though each epoch contains a fresh version of those images. As a result, the number of photos the model has seen grows with each epoch.

3.5 Data Exploratory Analysis

The below matrix of images plots the hand gesture dataset and face authentication dataset that is generated automatically using the OpenCV framework.

It took 20 frames per second. The below plots visualize some of the glimpses of the dataset with its class name in the figure 3.

Since the image is a 3-channel image with a goal size of 244x244 pixels, the histogram below in the figure 5 explains the distribution of color distribution and brightness levels of each primary color in the image, such as RGB. The color becomes deeper and richer as you move ahead in pixels towards the right. It seems that there is more combination of green and red pixel value which is elevated in the distribution histogram.

4 Design Specification

In pursuance of implementing the proposed design and deploying the ATM prototype as a web application to check the built model performance in real-time, the following tools and frameworks are utilized.The figure 6 explains the design specification diagram

Figure 3: Hand gesture dataset

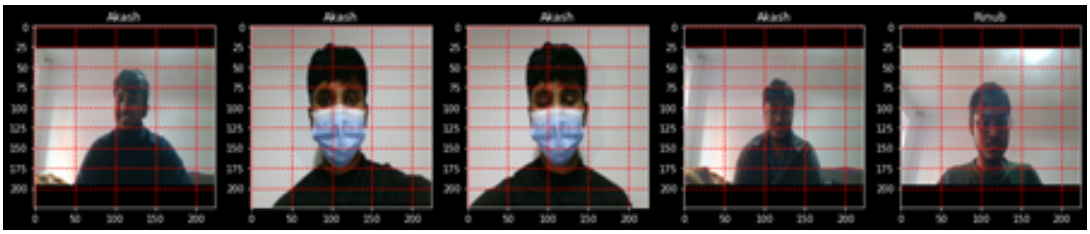

Figure 4: Face Authentication dataset

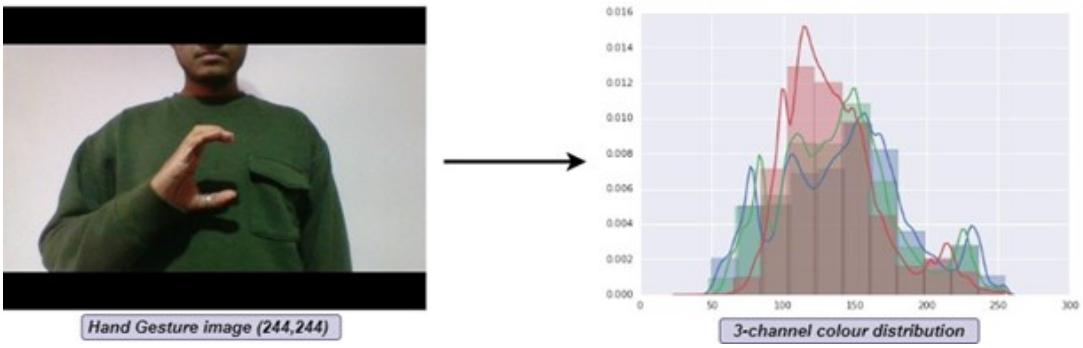

Figure 5: Hand gesture dataset

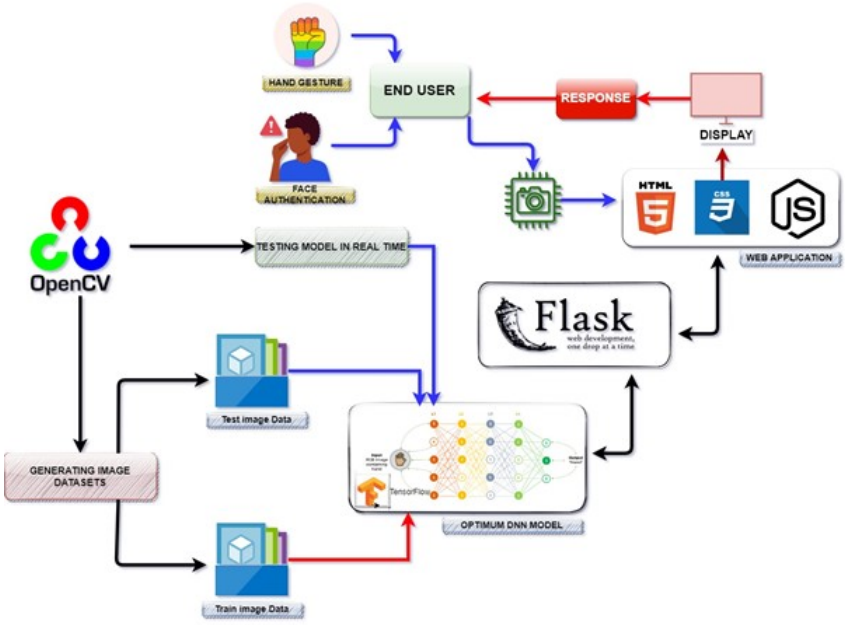

Figure 6: Design Specification

Keras, a deep learning application interface based on Python that runs as a front-end of the Deep learning platform TensorFlow, would be used to implement the proposed design. This helps to swiftly put the concept of contact-less hand-gesture recognition ATM into reality, which is critical for successful research. TensorFlow-2 is an open-source Deep learning framework that operates from start to finish on the CPU and GPU, efficiently executing low-level tensor operations.

HTML serves as the backbone for the ATM interface. This interface is enhanced and modified using other tools such as CSS and JavaScript. CSS is used to control the appearance, style, and structure of a website. JavaScript governs the behavior of individual components.

Flask is a python web framework that integrates the front-end of the ATM with the exported deep learning model to construct the gesture recognition ATM prototype. To handle HTTP response requests, we import the 'Response' and 'request' modules from the flask. The HTML file shown previously is rendered using render template '. OpenCV is a module that allows you to execute any computer vision operations as well as access the camera on your device.

5 Implementation

Phase 1: The image dataset of hand gestures is framed and generated using the python OpenCV framework. The open CV framework access the laptop's inbuilt camera. A rapid 100 photos were captured and stored in the given path. By repeating this process five different hand gestures image datasets were created which hold 2000 images. All these images were separated into different folders concerning different hand gestures.

Phase 2: All these images were pushed to google drive to Programmatically access them in google colab Pro. Because Google Colab Pro is an outstanding tool for deep learning

tasks, it is used for this project development phase and exporting an optimum model to integrate it with the web application. It's a hosted Jupyter notebook that doesn't require installation and includes a great version that offers access to Google processing resources like GPUs and TPUs to train and build a deep learning model faster with a large image dataset.

Phase 3: After accessing the image datasets, a python framework called TensorFlow is utilized which holds libraries to develop multi-layer large-scale neural networks and to perform image data pre-processing, data augmentation. Four models such as Custom CNN, VGG16, MobileNet, ResNet50 were built to extract features from the images and classify the gestures accordingly to perform specific tasks in the ATM. All these models were evaluated using different metrics such as Accuracy, Precision, Recall, Validation Accuracy, Validation Precision, Validation Recall. The optimum model is exported in the 'h5' file format for integration.

Phase 4: The front end of the web application is designed using HTML, CSS, and Javascript. To access the device camera a separate class in javascript is implemented. The integration between the front end and the exported optimum deep learning model is established using the Python Flask framework which is clearly explained in the Figure 6.

Phase 5: After integrating the exported optimum deep learning model with the front end. The web application server is hosted and UAT is performed to check the real-time execution of the model. Initially, face authentication is performed to access the user account to perform financial transactions using hand gestures. The javascript class for accessing the device camera is programmed to capture 4 frames per second which is shown in the Figure 6. After capturing the images, the flask framework sends the request for a prediction to the machine learning model. The result from the model is sent back to the front end as a response to perform financial transactions by the end-user.

5.1 Implementing diffractive deep neural network for hand gesture recognition

After generating enough data to train and validate the custom CNN and transfer learning models such as ResNet50, MobileNet, and VGG16. The dataset undergoes explanatory analysis. After understanding the image dataset carefully before building the model. Data is preprocessed where resizing and changing all the images to the same target size, which can be seen in the figure 7.

For training, the custom CNN model the image is transformed to a constant size of (64, 64) and for training and validating the transfer learning models such as ResNet50, MobileNet, and VGG16. The image is then augmented using different parameters such as 'rotation range', 'rescale', 'zoom range' 'horizontal flip', 'width shift range', 'height shift range', and the image in the dataset is increased. After building all four models, the performance of each DNN model using different factors and parameters such as the number of epochs, size of the dataset, run time of the DNN model, accuracy, precision, recall, validation accuracy, validation precision, and validation recall. Finally, in which processing unit the model is trained is also taken into consideration.

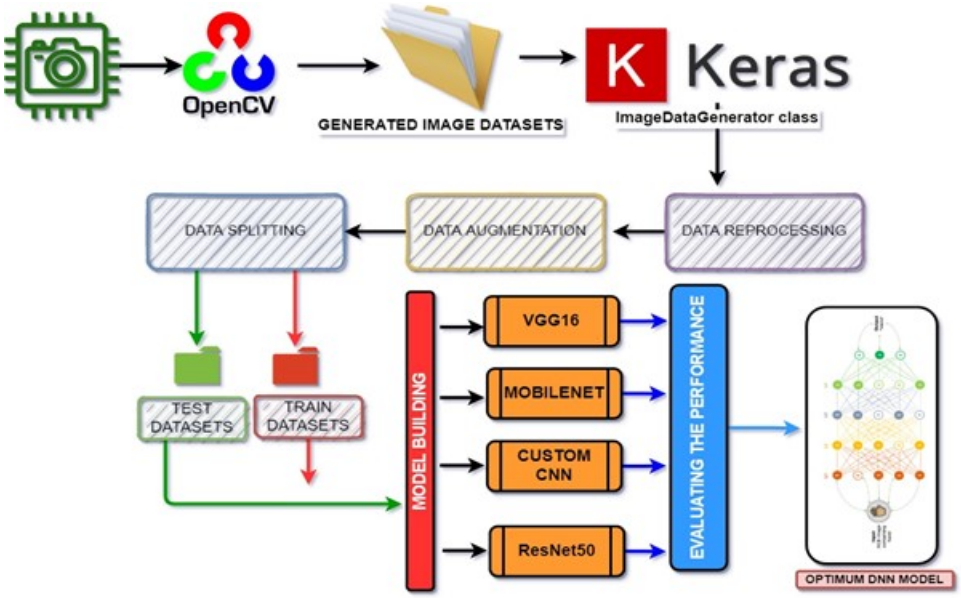

Figure 7: Model Selection

5.1.1 Custom CNN diffractive DNN architecture for hand gesture recognition

The Figure 8 is the custom CNN architecture, which is built so that every Conv2D and MaxPooling2D layer outputs a 3D tensor of shape with height, width, and channels. It has three channels since it generates colored visuals. Every image in the dataset is converted to the intended size of 64 x 64 before being introduced to the input layer. When traveling farther into the network, the width and height measurements begin to shrink. Units and the activation layer of the final dense layer control the number of output channels for each Conv2D layer. The last dense layer in this architecture has 5 units to fire via the Softmax activation function, which performs better for polynomial classifications with multiple polynomials. To decrease the computation, more units were not added when the width and height of each Conv2D layer shrank.

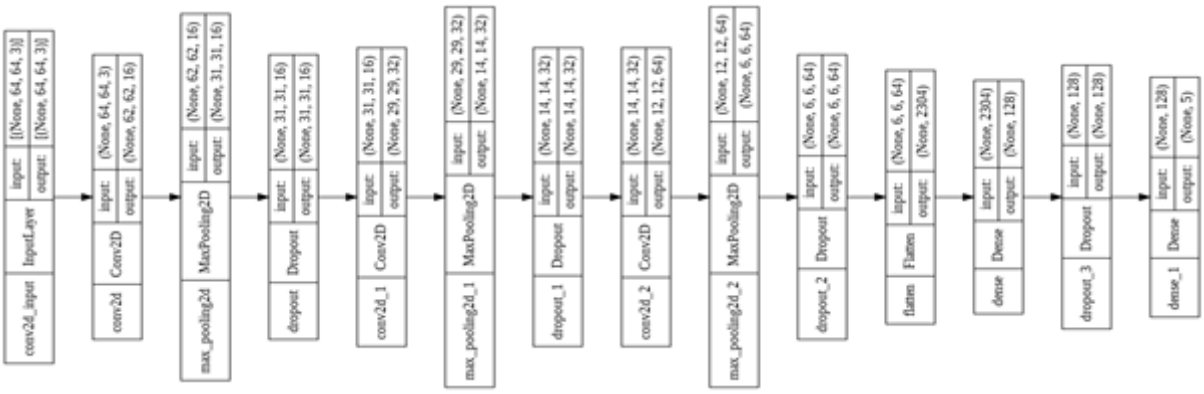

Figure 8: Architecture of CNN

5.1.2 VGG16 Transfer learning DNN architecture for hand gesture recognition

The network's input is a two-dimensional image (224 x 224 x 3) which is shown in the figure 9. The first two layers have the same padding and 64 channels with a 3*3 filter size. After that, two layers of convolution layers with 256 filter size and filter size are added after a stride (2 x 2) max pool layer (3 x3). Following it is a stride (2 x 2) max-pooling layer, which is identical to the previous layer. Then there are 256 filters and two convolution layers with filter sizes of 3 and 3. Then there are two sets of three convolution layers, as well as a max pool layer. Each filter has 512 filters of the same size (3 x 3) and the same padding. This image is then fed into a two-layer convolution stack. It also employs 1 x 1 pixels in some of the layers to adjust the number of input channels. After each convolution layer, 1-pixel padding is used to prevent the image's spatial information from being lost, that can be shown in the Figure 9.

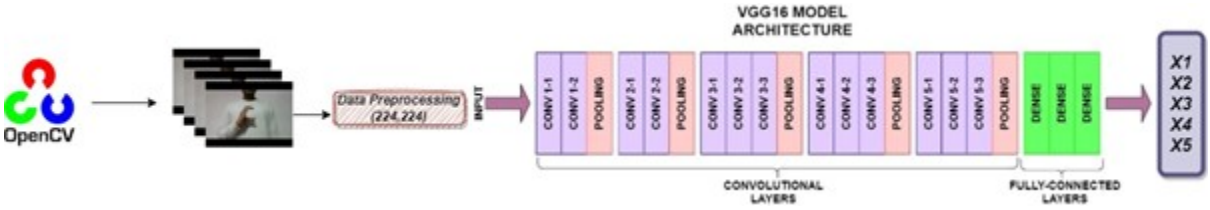

Figure 9: Architecture of VGG16

5.1.3 MobileNet Transfer learning DNN architecture for hand gesture recognition

MobileNet is a reduced design that uses depth-wise separable convolutions to generate less computational deep convolutional neural networks in the Figure 10. The mobile network is based using depth-wise separable convolutions, with the exception of the first layer, as previously indicated. The first layer is a complete convolutional layer. All layers are subjected to batch normalization and the Rectified linear activation function. The final layer, on the other hand, is a completely linked layer that feeds into the softmax for classification and has no non-linearity is shown in the Figure 10.

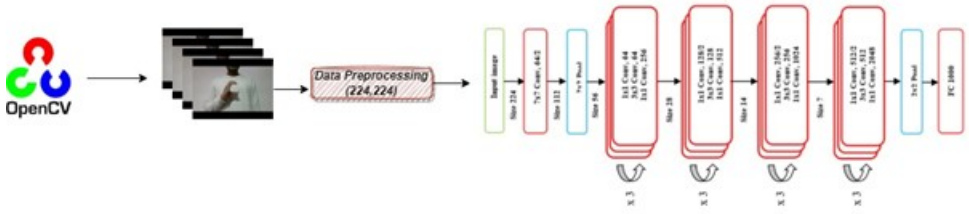

Figure 10: Architecture of MobileNet

5.1.4 ResNet50 Transfer learning DNN architecture for hand gesture recognition

Forty-eight Conv2d layers, a MaxPool layer, and an Average Pool layer make up the ResNet50 transfer learning model. There are 3.8 x 109 floating-point operations in it. There was a minor alteration in the connectivity between the neural connections, which previously skipped two layers but now skip three layers, referred to as a bottleneck layer, and there were also 1 * 1 convolution layers. The number of parameters and matrix multiplications are reduced when a bottleneck is used. To increase the depth and have fewer parameters, the objective is to make residual blocks as thin as feasible. They are utilized in deeper ResNets50 and were introduced as part of the ResNet design.

5.1.5 Feature extraction and engineering

The DNN network employed in this study allows a picture to propagate to the final max-pooling layer before the fully connected layers and extract the activations there. The volume form of the output from the max-pooling layer, which we flatten into a feature vector with a dim of 21,055. This technique is performed for all photos in a collection of 2000 images, resulting in a total of 2000 $\times$ 21,055-dim feature vectors.

5.2 Implementing diffractive Transfer learning model for Facial Authentication in ATM

To protect the user's account and access it to perform financial transactions, facial authentication is used. To achieve financial authentication in the initial sign-in page of the application transfer learning deep learning model was built and compared to finalize the optimum performing model. The facial data set of two people were generated using the OpenCV. These datasets were processed and separated as testing and training datasets. So, such as VGG19 were used where the weights are downloaded from ImageNet. The final layer was capped and connected with the 2 dense layers to classify which person is trying to access the ATM. VGG19 is a VGG model version that consists of nineteen layers, including sixteen convolution layers, three fully linked levels, five MaxPool layers, and one SoftMax layer. This model is made up of roughly 19.6 billion FLOPS. This model has a more complex design than VGG16 and the other models used in this study.

5.2.1 Feature extraction and engineering for Facial Authentication model

The ATM is authenticated using transfer learning models, while training the models all the facial images generated using OpenCV propagate to the final max-pooling layer before the fully connected layers and extract activations from there. The output of the max-pooling layer in volume form, which we flatten into a feature vector of dimension 21,055. This approach is used to generate 1500 x 21,055 dimensional feature vectors from a set of 2000 images.

5.3 Evaluating the performance of the diffractive deep neural network

Cross-validation is one of the key validation approaches utilized in this study while training the CNN and transfer learning models such as ResNet50, VGG16, and MobileNet.

It provides a reliable assessment of a model's performance on unseen data. The dataset is divided into two subsets, models are trained on all but one of the subsets, and model performance is evaluated on the held-out validation dataset. The method is repeated until the held-out validation set is allowed to be selected from all subsets shown in the figure figure 11. After then, the performance metric is averaged across all the models constructed. This research used several parameters such as accuracy, precision, recall, validation accuracy, validation precision, validation recall. Additionally, the number of epochs, processing unit, data size, and time is taken to train the model are also taken into consideration while choosing the final model for integration. After exporting the optimum model after comparing all the model performance

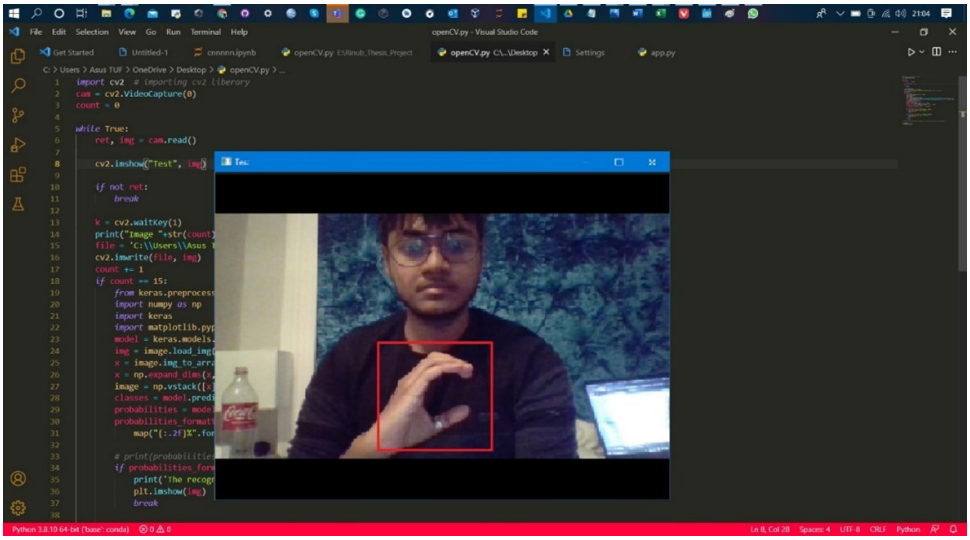

Figure 11:

5.3.1 Evaluating Custom CNN diffractive DNN architecture for hand gesture recognition

While training the custom CNN model which is composed of 3 convolution layers with the data size of 630 MB. The time took to train the model with 20 epochs was 180 seconds which is shown clearly in the figure 12. The google colab pro was used to run the model to access faster GPU. The models show the accuracy, precision, and recall of 87.68%, 92.99%, 76.22%. Some of the other parameters were also calculated such as validation accuracy, validation precision, and validation recall of 85.76%, 92.08%, 76.74% which are the most important parameters which show the performance of the model in real-time. The below line graph shows the loss and accuracy during every epoch while training the model.

Fine-tuning the model: To reduce the saturation and overfit of the model parameters such as call back, Early Stopping, ReduceLROnPlateau are implemented. When learning becomes stagnant, certain parameters, such as reduced learning rate, are used to benefit the model by reducing the learning rate by a factor of 2-10. This callback analyzes a quantity and reduces the learning rate if no improvement is noticed after a 'patience' number of epochs.

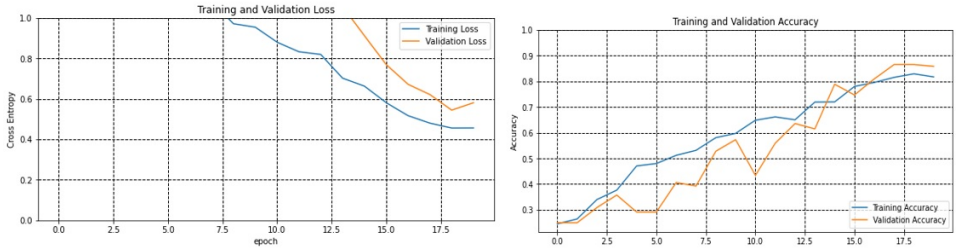

Figure 12: Validation loss and Validation accuracy for CNN

5.3.2 Evaluating VGG16 Transfer learning DNN architecture for hand gesture recognition

With a data size of 630 MB and 2000 photos, the proprietary transfer learning VGG16 model, which is composed of convolution and max pool layers, was trained consistently across the architecture. It took 120 seconds to train the model with five epochs which is shown clearly in the figure 13. The model was run on the Google Colab Pro to have access to a faster GPU. The accuracy, precision, and recall of the models are 99.04 percent, 99.04 percent, and 98.96 percent, respectively. Other metrics such as validation accuracy, validation precision, and validation recall of 99.62 percent, 99.82 percent, and 98.52 percent were also calculated, which are the most essential parameters that reflect the model's performance in real-time. The loss and accuracy throughout each period of training the model are depicted in the line graph below. **Fine-tuning the model:** Image augmentation was used to fine-tune the VGG-16 model to test if it would increase model accuracy. Unfreeze the fifth convolution block while keeping the first four blocks frozen, using the same VGG-16 model object stored in the transfer model variable from our previous model.

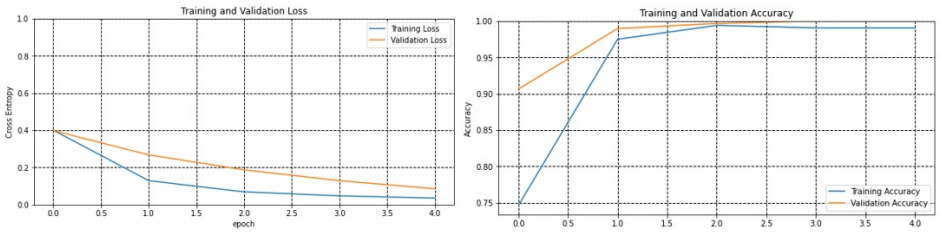

Figure 13: Validation loss and Validation accuracy for VGG16

5.3.3 Evaluating MobileNet Transfer learning DNN architecture for hand gesture recognition

The proprietary transfer learning MobileNet model, which is based on a streamlined architecture, was trained uniformly across the architecture with a data set of 630 MB and 2000 photographs. The model took 112 seconds to train with five epochs. The model was run on the Google Colab Pro to make use of a faster GPU. The models' accuracy, precision, and recall shown in the figure 14 are 99.74 percent, 99.14 percent, and 99.65 percent, respectively. Other metrics, such as validation accuracy, precision, and recall

of 97 percent, 98.45 percent, and 98.86 percent, were calculated, as these are the most important parameters that indicate the model's performance in real-time. The line graph below depicts the loss and accuracy at each period of training the model. **Fine-tuning the model:** After running the model with the downloaded weights and by freezing some of the base layers. Some of the other parameters are tuned such as training with the updated weights and unfreezing the base layer for training to achieve higher accuracy. This allows the higher-order feature representations of the fundamental model to be fine-tuned, making them more relevant for identifying hand motions.

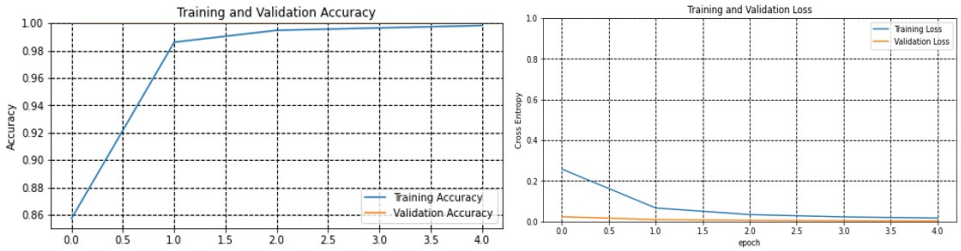

Figure 14: Validation loss and Validation accuracy for Mobile-Net

5.3.4 Evaluating ResNet50 Transfer learning DNN architecture for hand gesture recognition

With a data set of 630 MB and 2000 pictures, the proprietary transfer learning ResNet50 model, which is based on a streamlined architecture, was trained evenly across the architecture. With five epochs, the model took 60 seconds to train. To make use of a faster GPU, the model was run on the Google Colab Pro. The accuracy, precision, and recall of the models shown in the figure 15 are 99.74 percent, 99.65 percent, and 99.83 percent respectively. Other metrics were generated, including validation accuracy, precision, and recall of 100%, 100%, 100%, as they are the most essential parameters that show the model's performance in real-time. When compared to all other models, this model took the shortest time to train (60 seconds) and has greater accuracy in classifying gestures. The line graph below depicts the model's loss and accuracy over each phase of training. **Fine-tuning the model:** The ImageDataGenerator is used to modify the data and generate new images depending on the samples. The average epoch time using a Tesla K80 GPU was roughly 10 seconds, which is nearly 2 times faster than all other transfer learning models built up for the same purpose.

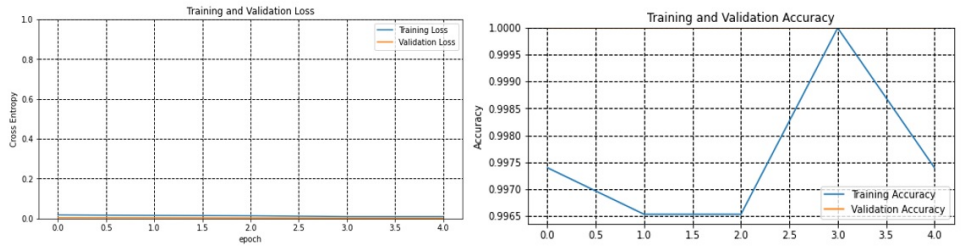

Figure 15: Validation loss and Validation accuracy for ResNet50

6 Comparison of Developed Models and Discussion

After building the proposed models architecture using the Keras framework and training all the models using the images generated from the OpenCV. All the CNN and Transfer learning models are fine-tuned using several parameters to achieve a state of best performance. All the models in the Figure 16 are compared concerning the above-mentioned matrices and parameters. It is evident from the Run time, epochs, validation accuracy, precision, and recall, ResNet50 performance in real-time was way better and the run time is comparably very low. It achieves 100 percent validation accuracy with the real-time images with a processing time of 26ms.

Model	Algorithm	Image count	Epoch	Processing	Time	Accuracy	Precision	Recall
1	Custom CNN	2000	20	GPU	180	87.68%	92.99%	76.22%
2	VGG16	2000	5	GPU	120	99.04%	99.04%	98.96%
3	Mobile Net	2000	5	GPU	112	99.74%	99.14%	99.65%
4	ResNet50	2000	5	GPU	60	99.74%	99.65%	99.83%

Model	Algorithm	Image_count	Epoch	Processing	Time	Val_Accuracy	Val_Precision	Val_Recall
1	Custom CNN	2000	20	GPU	180	85.76%	92.08%	76.74%
2	VGG16	2000	5	GPU	120	99.62%	99.82%	98.52%
3	MobileNet	2000	5	GPU	112	97%	98.45%	98.86%
4	ResNet50	2000	5	GPU	60	100%	100%	100%

Figure 16: Metrics for Training data

Figure 17: Visualization for evaluation metrics

The ResNet50 model is exported as h5 format and integrated with the open CV to take live images, and these images are loaded to the model using predict class. The value is received in an array format. For each value of the array in the figure 17, a specific financial

task is printed concerning the gesture. The below-mentioned histogram and tabulated performance outcome of all the four models with different factors are visualized. This model is integrated with the front end of the web application ATM simulator using the python flask framework. The figure 17 shows the visualization for evaluation metrics.

7 Conclusion and Future Work

In this experiment, various deep learning models such as custom CNN and other transfer learning models such as VGG16, ResNet50, Mobile Net, VGG19 were built to achieve operating the Automatic Teller Machine using Hand Gestures by the users. This new innovative interaction between the automatic teller machine and the end-user will result in the safe use of the ATM without contacting the surfaces of the ATM such as screen and buttons for control. This will reduce the transmission and spread of the virus such as COVID from the surface of the ATM. In this period of the never-ending cycle of pandemic and lockdown, this technology will be a breakthrough in banking and will contribute to this society by preventing the spread of COVID and another influenza virus that transmits through the contagious surface. The optimum model is selected through various factors as described in the above sections. The ResNet50 model which achieved a training accuracy of 99.74% and validation accuracy of 100% is chosen as the best model. The processing time for the ResNet50 model is 26ms which will make the web application faster to send the request to the model and get a response from the ResNet50 model through the flask framework. To take this research furthermore, in the future instead of classifying the gesture by a single model, parallel processing of multiple DNN models will be achieved simultaneously. The results from all the models are compared and the gesture that is identified by most of the models will be shown as the response in the front end of the web application. This will be more precise can completely avoid the validation loss.

Acknowledgment

I'd like to express my gratitude to the people listed below since I would not have been able to finish this research project or progress this far in my master's degree without them. The National College of Ireland faculty, especially my supervisor, Dr. Giovani Estrada, assisted me through my research with her knowledge and insight into the issue. And a special thanks to my friends and housemates, who motivate me to excel academically.

References

Ahuja, M. K. and Singh, A. (2015). Static vision based hand gesture recognition using principal component analysis, *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, pp. 402–406.

Alon, J., Athitsos, V., Yuan, Q. and Sclaroff, S. (2009). A unified framework for gesture recognition and spatiotemporal gesture segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(9): 1685–1699.

Bousbai, K. and Merah, M. (2019). A comparative study of hand gestures recognition based on mobilenetv2 and convnet models, *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA)*, pp. 1–6.

Chae, H. Y., Lee, K., Jang, J., Park, K. and Kim, J. J. (2019). A wearable semg pattern-recognition integrated interface embedding analog pseudo-wavelet preprocessing, *IEEE Access* **7**: 151320–151328.

Chanu, O. R., Pillai, A., Sinha, S. and Das, P. (2017). Comparative study for vision based and data based hand gesture recognition technique, *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pp. 26–31.

Golovanov, R., Vorotnev, D. and Kalina, D. (2020). Combining hand detection and gesture recognition algorithms for minimizing computational cost, *2020 22th International Conference on Digital Signal Processing and its Applications (DSPA)*, pp. 1–4.

Jaramillo, A. G. and Benalcázar, M. E. (2017). Real-time hand gesture recognition with emg using machine learning, *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–5.

Kabir, R., Ahmed, N., Roy, N. and Islam, M. R. (2019). A novel dynamic hand gesture and movement trajectory recognition model for non-touch hri interface, *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp. 505–508.

Kavyasree, V., Sarma, D., Gupta, P. and Bhuyan, M. (2020). Deep network-based hand gesture recognition using optical flow guided trajectory images, *2020 IEEE Applied Signal Processing Conference (ASPCON)*, pp. 252–256.

Nair, R., Singh, D. K., Ashu, Yadav, S. and Bakshi, S. (2020). Hand gesture recognition system for physically challenged people using iot, *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 671–675.

Parameshwaran, A. P., Desai, H. P., Sunderraman, R. and Weeks, M. (2019). Transfer learning for classifying single hand gestures on comprehensive bharatanatyam mudra dataset, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 508–510.

Sharma, S., Jain, S. and Khushboo (2019). A static hand gesture and face recognition system for blind people, *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 534–539.

Tian, X. and Chen, C. (2019). Modulation pattern recognition based on resnet50 neural network, *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*, pp. 34–38.

Venkatesh, Y, N., Hegde, S. U. and S, S. (2021). Fine-tuned mobilenet classifier for classification of strawberry and cherry fruit types, *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–8.

Wang, J. and Rai, L. (2020). Hand gesture recognition system for parking applications based on machine vision, *2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)*, pp. 734–737.

Zhu, D., Wei, R., Zhan, W. and Hao, Z. (2019). Individual soldier gesture intelligent recognition system, *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 231–235.