AN EFFICIENT LIGHT WEIGHT CONVOLUTIONAL NEURAL NETWORK FOR HAND GESTURE RECOGNITION

A Project Report

Submitted by

Ms. AYSHA REGA S

REG NO : TKM20MEAI04

SEMESTER : IV

In partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
Mechanical Engineering (Artificial Intelligence)

Under the guidance of
Prof. CHINNU JACOB

Thangal Kunju Musaliar College of Engineering
Kerala

JULY 2022

DECLARATION

I undersigned hereby declare that the project report "An efficient light weight Convolutional Neural Network for Hand Gesture Recognition" submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Chinnu Jacob. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam
Date: AYSHA REGA S

Thangal Kunju Musaliar College of Engineering, Kollam
Centre for Artificial Intelligence

C E R T I F I C A T E

This is to certify that,this report titled ***AN EFFICIENT LIGHT WEIGHT CONVOLUTIONAL NEURAL NETWORK FOR HAND GESTURE RECOGNITION*** is a bonafide record of the **Project** presented by **AYSHA REGA S (TKM20MEAI04)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M.Tech Mechanical Engineering (Artificial Intelligence)** in **APJ Abdul Kalam Technological University** .

Internal Supervisor	Project Coordinator	Head of the Department
Prof. Chinnu Jacob Assistant Professor Centre for AI	Prof. Sumod Sundar Assistant Professor Centre for AI	Dr. Imthias Ahamed T P Professor & Head Centre for AI

Internal Examiner	External Examiner

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A Shahul Hameed**, Principal of T.K.M College of Engineering for giving me an opportunity to present my project. I would like to thank **Dr. Imthias Ahamed T P**, Professor and Head of the Department, Centre for Artificial Intelligence, TKMCE, for her constant support and encouragement throughout the project work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my internal supervisor **Prof. Chinnu Jacob**, Assistant Professor, Centre for Artificial Intelligence, TKMCE and project coordinator **Prof. Sumod Sundar**, Assistant Professor, Centre for Artificial Intelligence, TKMCE for their expert guidance, cooperation. With a profound sense of gratitude, I would like to thank **Dr. Santhi Natarajan**, Honorary Professor, for her immense encouragement. I would like to express my gratitude to **Mr. Manusubramanian S**, Mentor, Scientist/Engineer SE, Liquid Propulsion System Centre, Indian Space Research Organisation(ISRO), for his expert guidance, and cooperation. I also extend my thanks to the entire staff of the Centre for Artificial Intelligence, TKMCE, who has encouraged me throughout this work.

I also express my thanks to my loving parents, brother and friends, for their support and encouragement in the successful completion of this project work.

AYSHA REGA S

Abstract

Following the breakout of COVID-19, there has been a significant increase in demand for gesture sensing applications, which allow users to manage gadgets with simple hand gestures rather than physically touching them. In comparison to expressions, actions, and other interaction techniques, gestures are more intuitive, straightforward, and natural. For Smart AI assistants, a cognitive vision system is essential for enabling seamless interaction with humans. Among the various features in the vision system, the ability to detect and recognize different hand gestures provides significant value addition. Systems employing hand gesture recognition technology are capable of distinguishing specific gestures such as victory sign, thumbs up, wave, peace sign, rock sign, number counting, etc. Real-world systems designed for human-computer interaction struggle to recognise and categorize hand gestures because 1) People perform gestures in a wide variety of ways, based on their cultural difference, 2) Variability of input lighting, distance limits, 3) Requirements of larger datasets etc. The wearable glove-based sensor technique and the camera vision-based sensor approach are the two main strategies for hand gesture recognition research. In this work, images captured using camera sensors are used as the input and fed into the proposed shallow convolutional neural network for classification and prediction of hand gestures. The proposed low weight convolutional neural network achieved faster training results by using fewer parameters and training epochs. Three well-known pre-trained models, including VGG16, ResNet50, and Mobilenet, are also taken into consideration for comparison. These models are applied on Fingers number count and LeapGestRecogn dataset and evaluation measures such as F1 score, recall, accuracy, and precision are computed to analyze the performance of the models.The experimental results indicate that the proposed model has achieved a recognition rate of 99.9% and 99.79% on LeapGestRecogn and Fingers number count dataset respectively. Furthermore, the proposed model outperformed the state-of-the-art methods with better accuracy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Background

Communication between humans and machines has always been difficult. User interaction with a keyboard and mouse has gradually replaced other methods as the norm in the industry.[1]. The majority of our computer interaction is performed with our fingers and eyes, while other portions of our bodies, such as our legs, arms are grossly underutilised. The introduction of the touch screen revolutionised human-computer interface, making it more natural and tactile. With the use of multi-touch technology, we could use several points of contact with a surface (such a trackpad or touchscreen) simultaneously [2]. The user can now zoom in, zoom out, scroll, rotate, and toggle between user interface elements thanks to the development of multi-finger gestures. A greater variety of gestures can help people communicate more effectively. Communication can be improved by using hand gestures. Movements provide an additional layer of engagement to the machine interface, much as subtle facial expressions and gestures augment spoken communication. Over the years, deep learning approaches for voice and facial recognition technology have made significant development. But gesture recognition has always been more challenging and has only recently become more popular [3]. Gesture detection is now not only possible, but also more accurate than ever with deep learning architectures.
Nonverbal communication is vital in our lives since it delivers roughly 65 percent of messages, compared to only 35 percent of our conversations through verbal communication. Human body motions, primarily hand movements, are used to interact with computers [4]. A camera analyses the human body's movements and sends the information to a computer, which then uses the gestures as input to operate objects or applications. When a person claps their hands together in front of a camera is transmitted through a computer, the sound of cymbals crashing together can be produced. One application of gesture recognition is to assist the physically disabled in interacting with computers, such as through translating sign language. Traditional modes of interaction between humans and machines have changed as a result of the quick development of Deep Learning and Artificial Intelligence technology. Hand gestures can symbolise a variety of human-machine communication since they are the most adaptable part of the human body. They're commonly utilised to communicate between humans and computers or other technological gadgets like smartphones,Robotics, vehicle infotainment systems, and so on. Human-computer interaction may be replaced by gesture recognition [5]. Interacting can be done via wired or touch-controlled input devices.

Dynamic gesture recognition belongs to the field of video classification because the dataset is primarily composed of video frames [6]. The outcome is that both spatial and temporal features are used. Dynamic gesture identification is a challenging procedure since the images created from video recordings do not have constant pixels, the camera is not fixed in one place, and each person performs the same movement in a different way [7]. It is challenging for an algorithm to predict gestures with high accuracy since motions include a range of backgrounds as well as continuous hand and arm movement.
The goal of a hand gesture-based Human Computer Interaction framework is to collect raw data, which can be done in two ways: sensor-based or vision-based [8]. The sensor-based methodology necessitates the use of devices or sensors that are physically attached to the user's arm or hand in order to retrieve data. While vision-based designs necessitate the use of a still or video camera to capture images or recordings of hand motions [9].

1.2 Objectives

- The design and development of a deep learning model for accurate classification of different hand gestures.
- Implementation of hand gesture recognition in real time.

Chapter 2

Related Works

The emergence of Deep Learning has revolutionized the field of Hand Gesture Recognition. Most of works in the literature employs a combination 2D and 3D Convolutional Neural Networks. Rehman et al [12] 2022 proposed a 3D- Convolutional Neural Network and LSTM Networks for Dynamic Hand Gesture Recognition. The suggested architecture avoids excessive computing while extracting spatial-temporal information from input video sequences. Additionally, it picks up on the order of the time series frames. Utilizing 3D-CNN, spectral and spatial characteristics are extracted before being sent into the LSTM network, which performs classification. There are many processes in architecture, including data loading, data augmentation, training, and testing with only 3.7 million training parameters in this lightweight architecture. The model provides better results than MobileNetv2 + LSTM. To create a lighter model for use in practical applications, they must however lower the intricate, expensive computing parameters.

Liu Y et al [13] 2021 proposed a 3D Convolutional Neural Network (CNN) combined with attention mechanism algorithm for Dynamic Gesture Recognition. The entire gesture is captured by a 3D Convolutional Neural Network in terms of spatial displacement and spatiotemporal correlation. In order to achieve the conspicuous expression of features, invalid features are repressed and significant spatiotemporal characteristics are enhanced when 3D CNN and an attention mechanism (CBAM) are coupled. To boost efficiency, multimodal data input must recognise the two modes' complementary features. The properties of the item can be made more obvious and a network can be trained to perform better using the data fusion method. On the EgoGesture dataset, 3D CNN with an attention mechanism achieves a recognition accuracy of 72.4%.The model could only identify gestures in videos that contained a single action, thus they need to identify numerous movements more accurately. Another issue is the usage of a high number of parameters, which results in poorer network prediction.

Liu C and Szirányi T [14] 2021 For on-board UAV rescue, deep learning was used to implement person identification and gesture recognition in real-time. The live video recorded by the drone's camera serves as the system's input. Humans are first detected using Yolo3-tiny, and after that, pose estimation using OpenPose, which recognises the human skeleton and collects skeletal data from various body gestures, is used to forecast the user's rescue gesture using deep neural networks. Deep SORT is used for number counting in the human

tracking for the multiple individuals scenario. The skeleton is used as the primary feature for identifying body rescue gestures. The algorithm moves on to the final stage of hand gesture recognition after recognising the Attention gesture. If it is cancelled, turn it off immediately. The CNN model can distinguish five pre-trained gestures: Help, Ok, Nothing (i.e., when none of the aforementioned gestures are input), Peace, and Punch. Since the number of motions that can be identified is limited, the system can also record and define new gestures in order to create a new model by retraining the CNN. Drone rescue is more thorough and efficient when identification of both hand gesture and total body gesture are combined. However, they must address the issue of erroneous skeletal information when a person is not fully present in front of the camera. They might conduct research to adapt the technology to actual wilderness circumstances.

Xu C et al [15] 2021 presented a semi-supervised joint learning from a single color image for Hand Gesture Recognition. They implemented a ResNet for feature extraction of hand gestures. Semi supervised learning is applied to solve the problem of lacking proper annotation of different hand gestures in single video. However, they need to evaluate the efficiency of the results and also recommend to implement in real time for more dynamic applications.

Ahmed S et al [16] 2020 developed a hand gesture recognition IR-UWB Radar using an Inception Module-Based Classifier. The IR-UWB radar signal of gesture is first transformed to a grayscale image, which is then mapped into a three-dimensional (3D) RGB image. The pattern contained in the photographs was then examined using the inception module-based version of GoogleNet in order to recognise distinct hand motions. To test the model's resilience, eight different hand gestures were performed, and data was collected from many human subjects. To make data collection more realistic, they'll need a separate algorithm to recognise and distinguish between gesture and non-gesture signals. Furthermore, the gestures were recorded manually over a specified period of time, and the motions of the gestures were made within the boundary.So, they need to overcome the challenge of time-frame.

Neethu PS et al [17] 2020 implemented Convolutional Neural Networks enabling 96% accurate hand motion detection and identification. CNN separates the finger tips from the image of the hand gesture, and the finger tips are subsequently sent as input to the CNN classifier. A CNN classification technique is trained on the test hand gesture image that was taken from an open access image collection. The suggested hand gesture detection and identification methodology is rated according to its sensitivity, specificity, accuracy, and rate of recognition. In future work, they need to implement in real time on larger datasets.

The advantages and limitations of 5 recent related works are summarized in table 2.1.

Table 2.1: Review of recent related works

Reference	Technique used	Advantages	Disadvantages
[12]	Feature extraction and classification:- 3D CNN and LSTM [2022]	This model gives superior results as compared to MobileNetv2 + LSTM.	The model can't handle extensive computation
[13]	3D-CNN, Convolution Block Attention Module, Data fusion, Multimodal input [2021]	Convolution Block Attention Module suppresses invalid features with 72.4% accuracy	Only videos with a single action were recognised by this model, and a high number of parameters leads to poorer network prediction.
[14]	Human detection:- Yolo3 Tiny, Pose estimation:- OpenPose, Number counting:- Deep SORT algorithm, Classification:- DNN, CNN [2021]	In particular environments, users can efficiently and briefly communicate with drones. Rescue operations in remote areas can operate without intervention from the outside world.	When a person isn't fully present in front of the camera, the skeleton information is inaccurate. System cannot be used in the actual wildness. The UAV has a confined flying position.
[15]	ResNet, Region proposal Network, Semi supervised learning [2021]	To solve the problem of lacking proper annotation.	The proposed system cannot analyze in real time.
[16]	Feature selection:- Impulse-radio ultra-wideband (IR-UWB) radars, Classification:- GoogleLeNet [2020]	Model is well-fitted with 95% accuracy.	Gestures were acquired manually within a fixed time duration

Chapter 3

Hand Gesture Recognition

Static gestures and dynamic gestures are the two types of hand gesture recognition. In static hand gestures, just the hand is maintained stationary with a particular stance over time, however in dynamic hand gestures, the arm with the hand moves and there are a variety of poses that change depending on the time interval. The distinction between posture and gesture is that the former places more emphasis on the shape of the hand while the latter does so on the movement of the hand. Due to this distinction, there are two ways to approach hand gesture recognition: wearable glove-based sensor approach and camera vision-based sensor technique.Different acquisition tools, such as surface electromyography (sEMG) sensors for wearable technology and RGB camera, time of flight (TOF) camera, infrared cameras, or night vision cameras for real-time, are used to capture hand gestures. The features that are extracted for precise and successful gesture prediction include skin tone, appearance, motion, skeleton, depth, 3D model.

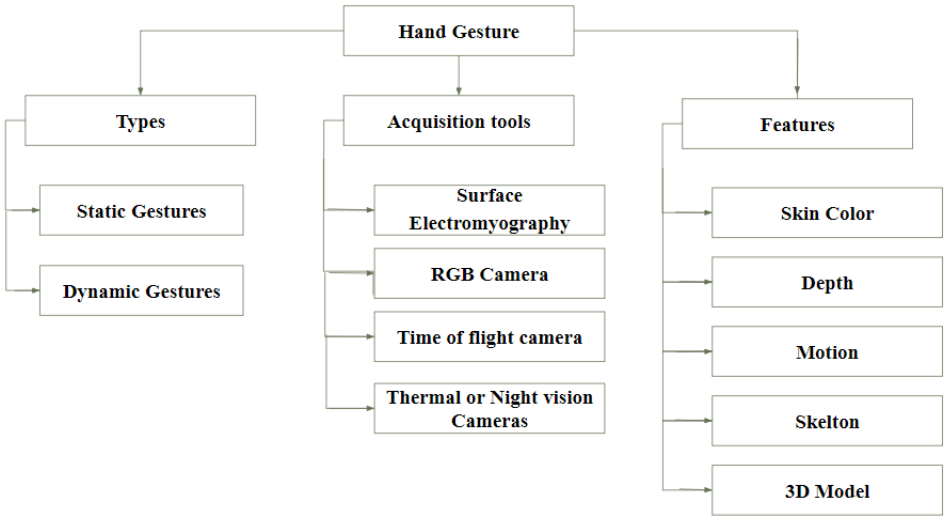

Figure 3.1: Different types, acquision tools and features taken for Hand gesture recognition

3.1 Relevance of Gesture Recognition

The Global Gesture Recognition and touchless sensing market is expected to reach 37.6 billion USD by 2026 at a CAGR of 22.6% [21]. In contrast to verbal communication, which makes up little more than 35% of our encounters, nonverbal communication carries roughly 65% of information [4].An effective human-computer interface is possible via gesture recognition in which the receiver recognises the user's motions. It is becoming more important in the modern paradigm of interactive, intelligent computing. Touch or wired-controlled input devices could be replaced with gesture recognition to minimize human-computer interaction. One significant benefit of the new touchless technology is the lack of the need to repair hardware-related issues.

Chapter 4

Methodology

4.1 Proposed technique for Hand Gesture Recognition

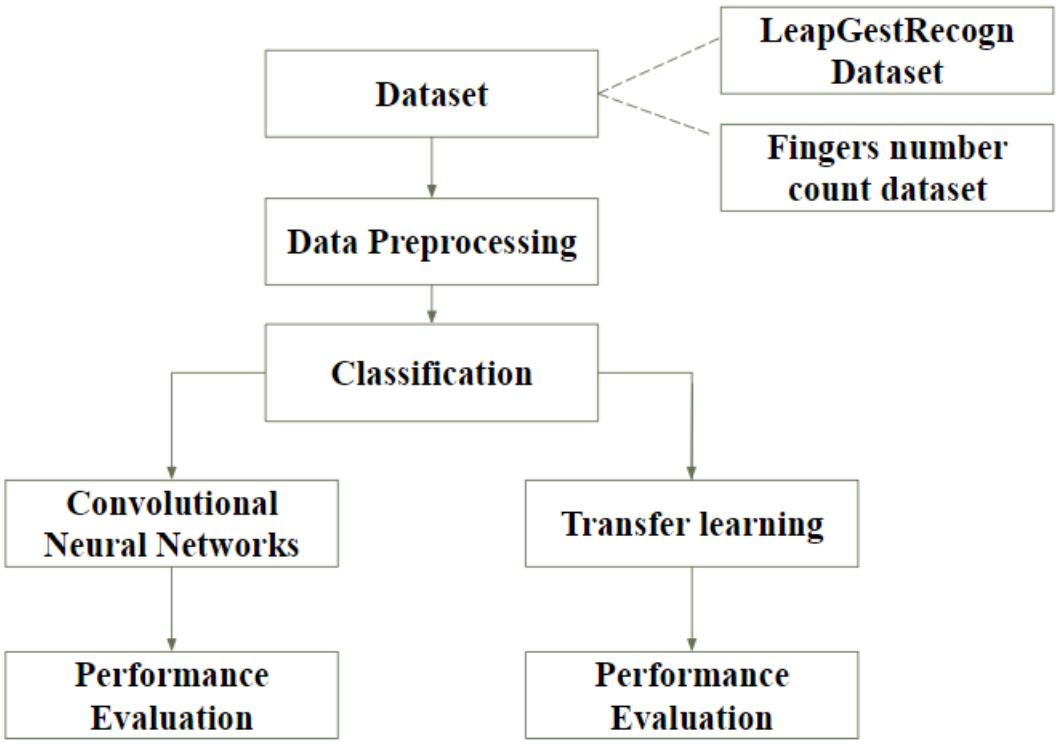

Figure 4.1: Proposed technique for Hand Gesture Recognition

The overeall work of proposed system is illustrated in Fig.4.3. Initially dataset containing different hand gestures like thumbs up, palm, thumbs down, ok etc. (LeapGestRecogn dataset) and gestures like number count from 0 to 5 (Fingers number count dataset) is

used. The different preprocessing techniques such as annotation arrangement of images, normalisation has applied for the dataset. The input image of size 240x640 which is converted into 320x120 pixels for faster processing. Dataset is then splitted into 70% for training and 30% for testing and is given to the proposed Convolutional Neural Network model. Different Pre-trained models such as VGG 16, MobileNet, ResNet50 etc are employed for comparison with the proposed model. Finally, the efficiency of the model is analysed using performance metrics like accuracy, precision, recall, F1-score.

4.2 Dataset Description

4.2.1 LeapGestRecogn dataset

Figure 4.2: Samples of LeapGestRecogn dataset

LeapGestRecogn dataset [18] composed by a set of near infrared images acquired by the Leap Motion sensor. Dataset includes 10 different hand-gestures that were performed by 10 different subjects. The subjects which include randomly picked both men and women (5 men and 5 women). The dataset is arranged as folders and subfolders, where the subjects are taken seperately as each folder from 00 to 09. Within the subject's folder includes 10 different gestures namely palm, I, fist, fist moved, thumbs up, index, ok, palm moved, C, down. Within each gesture subfolders of each subject includes 200 near infrared images of the gesture captured by leap motion sensor. The dataset contains total of 20000 images which is distributed as 2000 for each subject 00-09 with 200 images each for 10 different gestures. The folder name 00, 01, 02 .... 09 are taken as identifier of each subject, as well as 01_palm, 02_I.... 10_down are taken as identifier of different gestures within the subject.

4.2.2 Fingers number count dataset

For finger number count dataset, consists of 21700 images which include both left and right hands fingers where the images are 128 by 128 pixels. In fingers number count dataset composed by the gesture of numbers from 0 to 5.

For the convienence for differentiating between left and right hand gestures, the dataset is arranged as folders and subfolders, where the numbers are taken seperately as each folder from 00 to 05. Within the number's folder has 2 subfolders which includes seperate left and right hand number count. The dataset is arranged as seperate folders for numbers- 00, 01, 02, 03, 04, 05 and within this folders includes the 1800 left hand gestures images and 1800

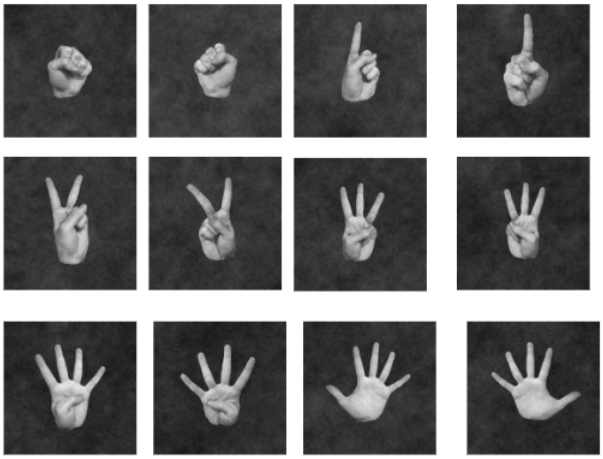

Figure 4.3: Samples of Fingers number count dataset

right hand gestures images.

4.3 Data Preprocessing

For the Fingers number count dataset, the entire dataset has been divided into six files, each including six distinct number count gestures. Two folders are made for left- and right-handed gestures in each folder. The images in both the datasets are resized to input dimension 120x320 for rapid processing.

4.4 Classification

A shallow convolutional neural network is proposed for classification of LeapGestRecogn dataset and Fingers number count dataset. Three well-known pre-trained models including VGG16, ResNet50 and Mobilenet, are also taken into consideration for comparison.

4.4.1 Classification using Convolutional Neural Networks (CNN)

A deep learning neural network designed for analysing organised arrays of data, such as images, is known as a convolutional neural network, or CNN. In the input image, design elements like lines, gradients, circles, or even eyes and faces are extremely successfully picked up by CNN. Convolutional neural networks are extremely effective for computer vision because of this feature. CNN does not require any preprocessing and may be applied directly to an underdone image. The convolutional layer, a specific kind of layer, is what gives convolutional neural networks their strength. Each of the numerous convolutional layers that make up CNN is capable of identifying more complex shapes. These layers are stacked on top of one another. Machines can perceive the environment similarly way humans do and to

apply that understanding for a variety of tasks, including picture and video recognition, image inspection and categorization, media reconstruction, recommendation systems, natural language processing, etc. Convolutional neural networks (CNNs) are employed as a classifier in this work because they are capable of extracting higher-level features from an image set without the need for manual intervention or adjustment. Three main layers make up this architecture. Convolutional layers expose the input image to a predetermined number of convolution filters. The layer conducts a set of mathematical operations and then combines the image with a convolutional filter. The activation function adds nonlinearities to the model in order to emphasise the positive elements and lessen the impact of negative ones. As a result, the layer generates a feature map. Pooling layers are crucial for reducing processing time since they compress the image data that convolutional layers have extracted. To extract subregions of the square-type feature map and maintain only their maximum value while discarding all other values,max pooling layers are used. On the features that the preceding layers have extracted, dense layers classify the data. Every node in a dense layer is linked to every node in the preceding layer.

4.4.2 Proposed Convolutional Neural Network on LeapGestRecogn Dataset

Initially loaded LeapGestRecogn dataset with seperate images and labels. Then the dataset is passed through the Convolutional Neural network classification model which is a Sequential Model with input layer, three 2D convolutional layer, three max-pooling layers, flatten layer and two dense layers. The input layer is fed with images in the shape of 120x320 pixels and Relu activation function is used. The output layer consists of 10 neurons with softmax activation function because it is a multi class classification problem. Convolutional Neural network model is compiled with sparse categorical cross entropy and Adam optimizer. Sparse categorical cross entropy is used when the classes are mutually exclusive. Here the dataset contains many sample gestures which belong to same class. Here, categorical crossentropy is chosen beacause one sample can have multiple labels. Adaptive Moment (ADAM) Estimation is an optimization algorithm for gradient descent which can tackle the problem of handling large number of parameters. It is highly efficient and it takes less memory space. Adaptive Moment (ADAM) Estimation is a combination of the momentum based gradient descent and the Root mean square propagation algorithm. Each block of layers in Convolutional Neural network ensures that the prominent features are passed as input to the subsequent block.In this way, the model will ensure the effectiveness of the output. The model is fitted with 10 epochs and obtained an accuracy of 99.98%. Fig. 4.4 shows the proposed Convolutional Neural Network on LeapGestRecogn Dataset.

4.4.3 Proposed Convolutional Neural Network on Fingers number count Dataset

Here,Fingers number count dataset is loaded with seperate images and labels. Then the dataset is passed through the Convolutional Neural network classification model which is a Sequential Model with input layer, 2D convolutional layer, 3 max-pooling layers, flatten layer and 2 dense layers. The input layer is fed with images in the shape of 120x320 pixels and Relu activation function is used. The output layer consists of 12 neurons with softmax activation function because it is a multi class classification problem. Convolutional Neural network model is compiled with sparse categorical cross entropy and Adam optimizer. Each block

of layers in Convolutional Neural network ensures that the prominent features are passed as input to the subsequent block.In this way, the model will ensure the effectiveness of the output. The model is fitted with 10 epochs and obtained an accuracy of 99.79%. Fig.4.5 shows the proposed Convolutional Neural Network on Fingers Number Count Dataset.

4.4.4 Transfer Learning

Transfer learning is the process of using a model that has already been learned to solve a new issue. Due to its ability to train deep neural networks using a tiny quantity of data, it is well-adopted in deep learning. In computer vision, neural networks typically target the first layer's detection of edges, the middle layer's detection of shapes, and the latter layers detection of task-specific properties. Transfer learning only uses the early and centre layers; the later layers are only retrained. It uses the labelled data from the task that served as its training ground. When we don't have enough annotated data to train our model with, when there is a pre-trained model that has been trained on the same data and tasks, or when there is a large amount of data, transfer learning offers a number of benefits, the most significant of which are reduced training time, improved neural network performance, and the absence of a large amount of data. Different transfer learning models including MobileNet, ResNet50, VGG16, etc. were trained in this study using the leapGestRecogn dataset.

VGG16 [19] is a straightforward and frequently used Convolutional Neural Network (CNN). A fixed-size RGB image with 224 by 224 pixels is used as the convnets' input during training. The only pre-processing applied here is to each pixel by subtracting the mean RGB value calculated on the training set. The image is run through a stack of convolutional layers using filters with an extremely narrow receptive field, such as 3x3 (the smallest size to capture the concepts of left/right, up/down, and centre and has the same effective receptive field as one 7 x 7). It has fewer parameters, more non-linearities, and is deeper. The spatial resolution is maintained after convolution since the convolution stride and the spatial padding of the convolution layer input are set to 1 pixel for 3 x 3 convolutional layers. Following some of the convolutional layers are five max-pooling layers, which aid in spatial pooling. Over a 2x2 pixel window, max-pooling is carried out using stride 2. Following a stack of convolutional layers (these have varying depths in varying architectures), there are three Fully-Connected (FC) layers: the first two have 4096 channels each, while the third performs 1000-way ILSVRC classification and so comprises 1000 channels (one for each class). The soft-max layer is the last one. In all networks, the fully connected layers have the same configuration.

ResNet50 [20] is a ResNet model version having 48 Convolution layers, 1 MaxPool layer, and 1 Average Pool layer. The floating point operations are 3.8 x 109. ResNet50 is used for computer vision tasks like image classification, object localization, and object identification, but it may also be used for non-computer vision tasks to add depth and lower computational costs.

MobileNets are compact, low-latency, and low-power models that may be customised to fit a variety of use cases' resource requirements. On top of them, segmentation, embeddings, and classification can be developed. The key distinction between the MobileNet design and a conventional CNN is that the former uses a batch norm and ReLU after each 3x3 convolu-

tion layer. Convolution was divided into a 3x3 depth-wise convolution and a 1x1 pointwise convolution by Mobile Nets.

4.5 Performance Metrics

An outcome where the model properly predicted the positive class is referred to as a true positive. Similar a true negative is a result for which the model accurately sees the negative class. A false positive is a result when the model forecasts the positive class inaccurately. A false negative is a result where the model forecasts the negative class inaccurately. The percentage of correctly classified data instances over all data instances is known as accuracy. Accuracy in multi-class categorization is described as follows:

$$\text{Accuracy} = \frac{\text{Correctly predicted value}}{\text{Total number of samples}} \tag{4.1}$$

The frequency with which a model correctly predicted the positive class is known as precision. A good classifier should have precision that is 1 (high). Precision only increases to 1 when the numerator and denominator are equal; as a result, the false positive rate is zero. As the number of false positives rises, the denominator value rises while the accuracy value falls.

$$\text{Precision} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Pos}} \tag{4.2}$$

For a good classifier, recall should ideally be 1 (high). Recall only increases to 1 when the numerator and denominator are equal, hence false negetive is 0. Denominator value rises as false negetive increases, exceeding numerator value, and recall value falls.

$$\text{Recall} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Neg}} \tag{4.3}$$

Therefore, the ideal precision and recall for a good classifier are one, which also indicates that the number of false positives and false negatives is zero. As a result, we require a measure that considers both precision and recall. The definition of the F1-score, a metric that considers both precision and recall:

$$\text{F1}score = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

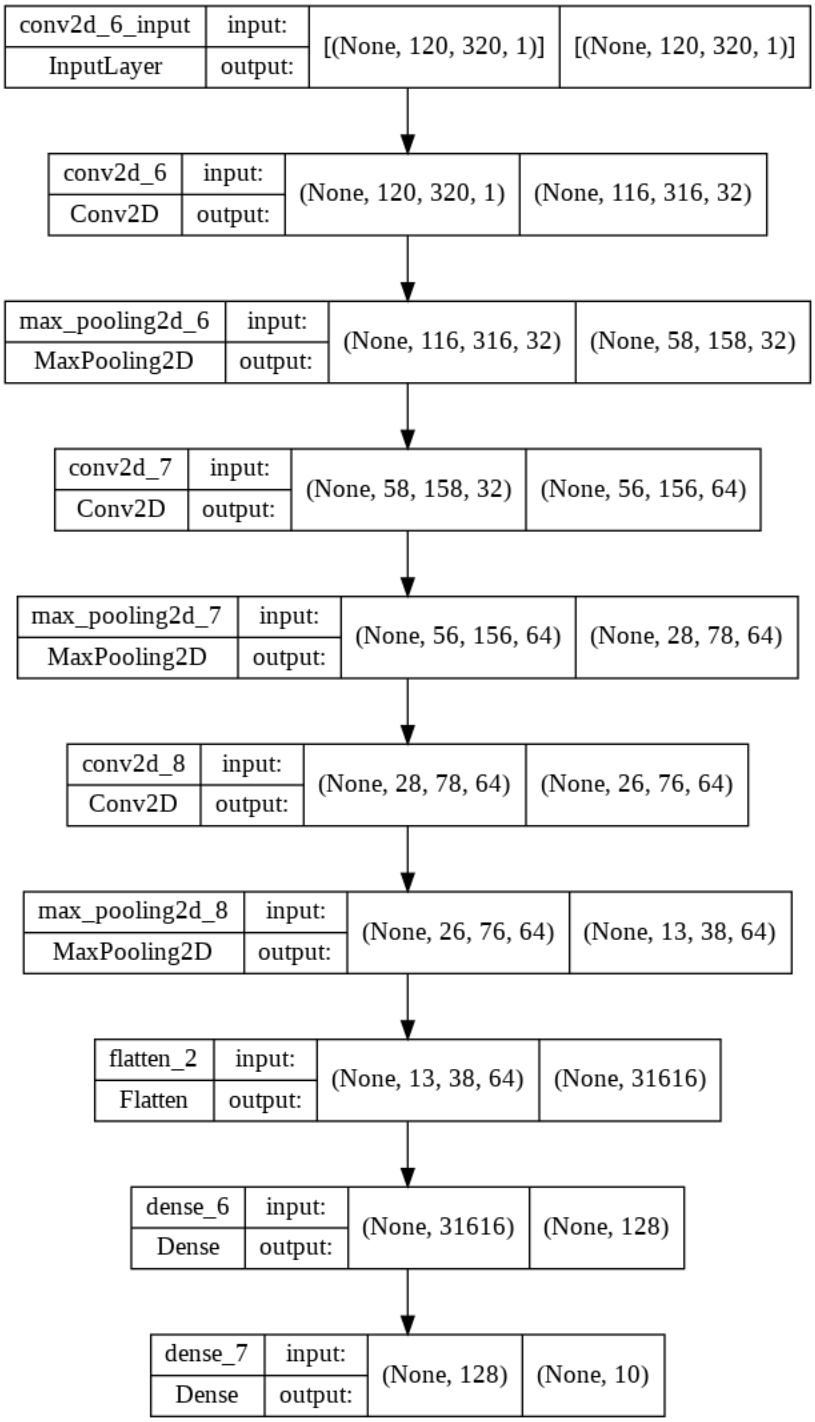

Figure 4.4: Proposed CNN Model on LeapGestRecogn Dataset

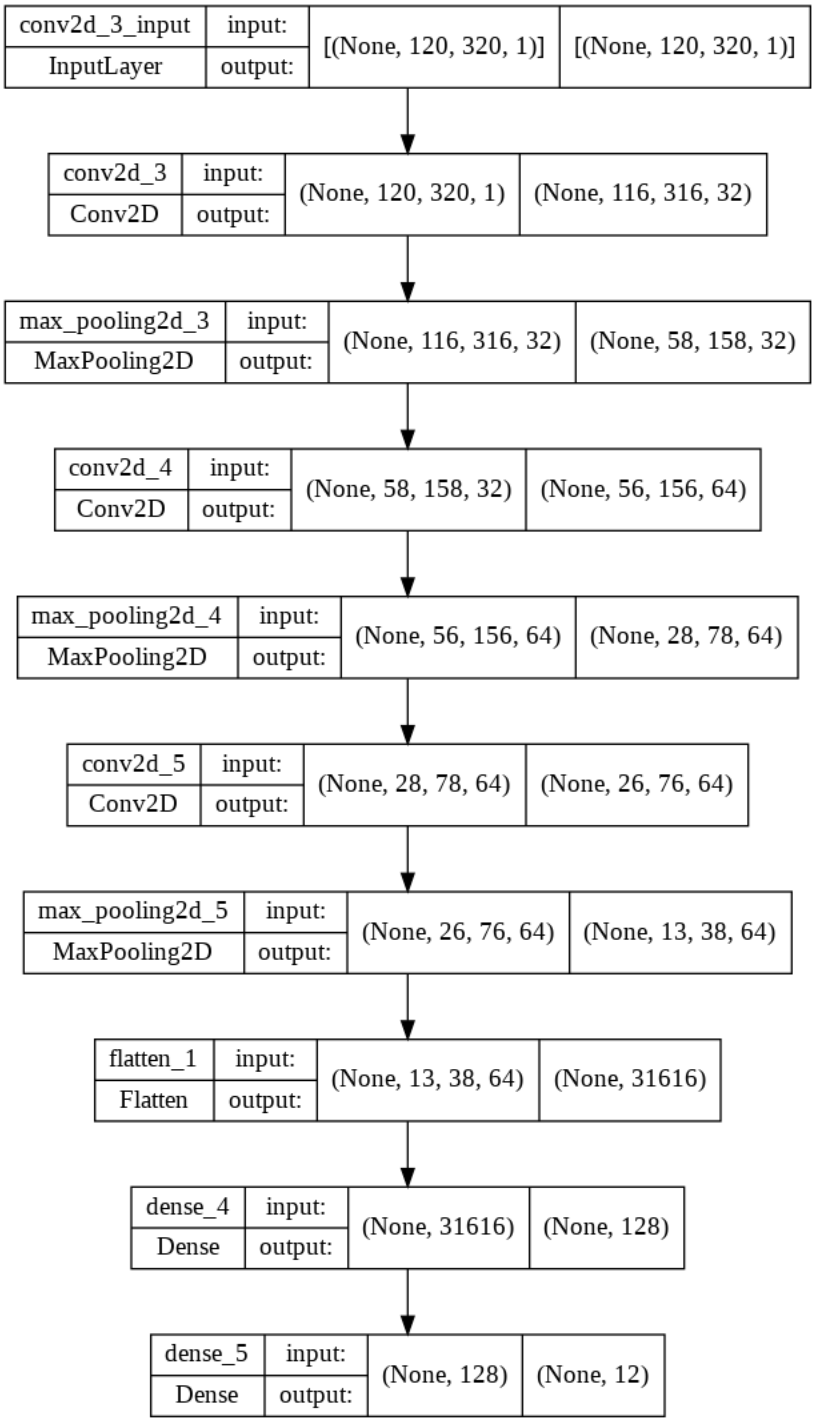

Figure 4.5: Proposed CNN Model on Fingers Number Count Dataset

Chapter 5

Results and Discussion

5.1 Hardware and experimental environment

Classification models are implemented in Windows 10 Pro OS, 64-bit operating system, x64-based processor, Intel Core i7 (7th Gen) 7600U @ 2.8 GHz (3.9 GHz). The experimental environment was prepared by using Python 3.6 programming language. The framework used is Keras with TensorFlow as backend in google Collaboratory. Machine learning and deep learning libraries include - NumPy, Pandas, Matplotlib, cv2 and Scikit learn. Performance analysis is performed to identify the best model that have the highest detection rate. The general evaluation metrics such as Accuracy, Precision, Recall, F1 score and Confusion matrix are used.

5.2 Data preprocessing

The LeapGestRecogn Dataset and Fingers Number count dataset is splitted into training and testing (70% for training and 30% for testing). For Fingers number count dataset, rearranged the whole dataset manually, by seperating the dataset gestures into different folders and then created 2 folders with the folder of each gesture and annotated as left and right, which represents the left hand gesture and right hand gesture.

Table 5.1: Training and testing distribution of dataset

Dataset	Training	Testing	Total
LeapGestRecogn Dataset	14000	6000	20000
Fingers number count dataset	15190	6510	21700

5.3 Experimental Results

The proposed Convolutional Neural Network is implemented on Fingers number count dataset and obtained an accuracy of 99.79%. To validate the proposed system, applied the proposed Convolutional Neural Network on LeapGestRecogn dataset. Table 5.2 shows the performance analysis of proposed Convolutional Neural Network on different datasets.

Table 5.2: Performance analysis of proposed Model on different datasets

Dataset	Accuracy(%)	Precision	Recall	F1-Score
LeapGestRecogn Dataset	99.98	0.9999	0.9998	0.9999
Fingers number count dataset	99.79	0.998	0.999	0.998

The Confusion matrix shows each class's contribution of the correct and wrong predictions is reported using count values. In Fig. 5.1 and Fig. 5.2 depicts the Confusion Matrix of Proposed Convolutional Neural Network Model on LeapGestRecogn dataset and Fingers number count dataset.

Table 5.3: Confusion Matrix of Proposed model on LeapGestRecogn dataset

	T. down	Palm	I	Fist	Fist_mv	Thumbs up	Index	Ok	Palm_mv	C
T. down	604	0	0	0	0	0	0	0	0	0
Palm	0	605	0	0	0	0	0	0	0	0
I	0	0	600	0	0	0	0	0	0	0
Fist	0	0	0	610	0	0	0	0	0	0
Fist_mv	0	0	0	0	591	0	0	0	0	0
Thumbs up	0	0	0	0	0	568	0	0	0	0
Index	0	0	0	0	0	0	596	0	0	0
Ok	0	0	0	0	0	0	0	586	0	0
Palm_mv	0	0	0	0	0	0	0	0	618	0
C	0	0	0	0	0	0	0	0	0	621

Table 5.4: Confusion Matrix of Proposed model on Fingers number count dataset

	0	1	2	3	4	5
0	1114	0	0	0	0	0
1	0	648	0	0	0	0
2	0	0	810	0	0	0
3	0	0	0	970	2	0
4	0	0	0	0	1076	0
5	0	0	0	0	0	1055

The predicted gesture and true gesture comparison is shown in Fig. 5.3 and Fig. 5.4 of Proposed Convolutional Neural Network Model on LeapGestRecogn dataset and Fingers number count dataset. Here it exhibits the efficiency of the model for prediction of hand gestures.

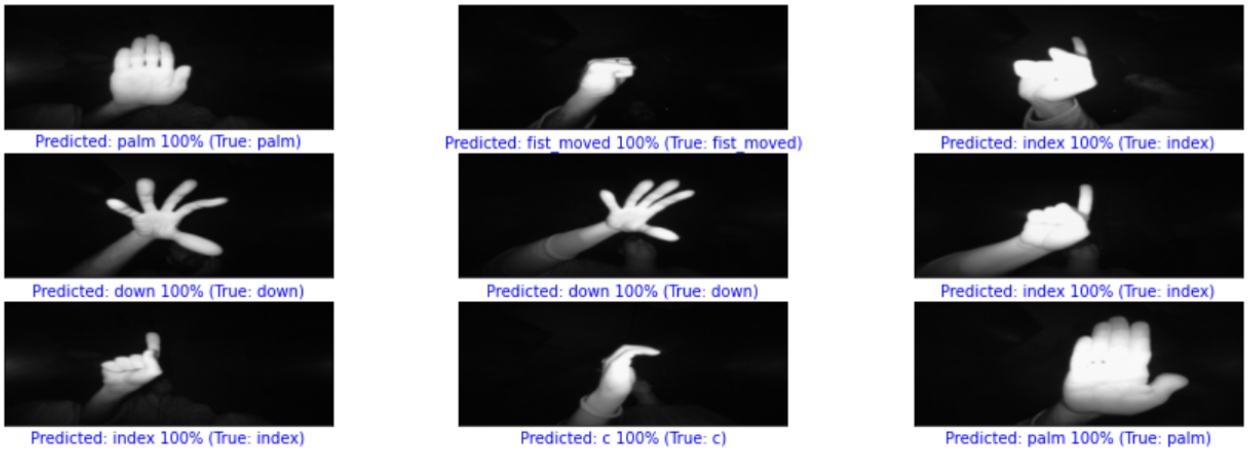

Figure 5.1: Prediction using proposed model on LeapGestRecogn dataset

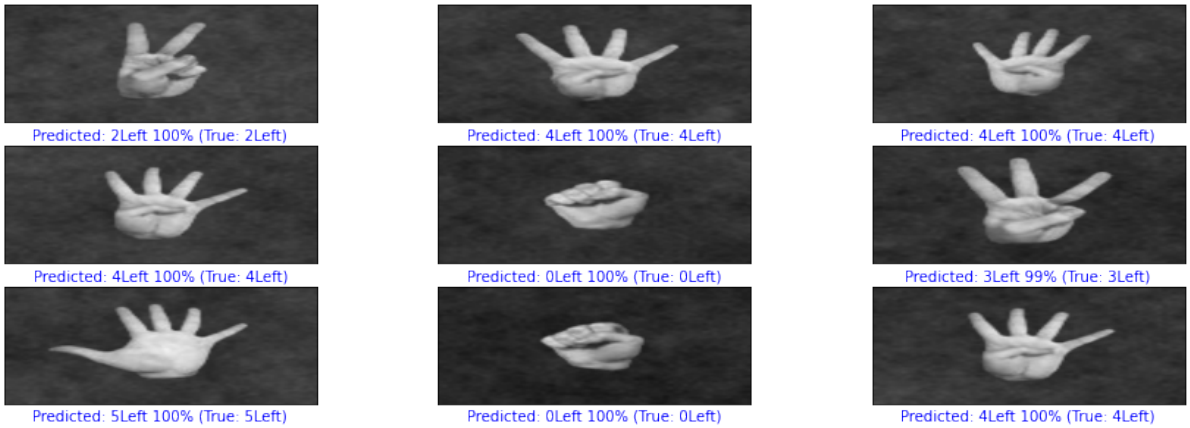

Figure 5.2: Prediction using proposed model on Fingers number count dataset

The loss plot depicts how well the proposed Convolutional Neural Network is performing on LeapGestRecogn dataset and Fingers number count dataset. The accuracy plots depicts the efficiency in prediction by comparing the proposed Convolutional Neural Network model with true value. It is plotted verses the number of epochs.

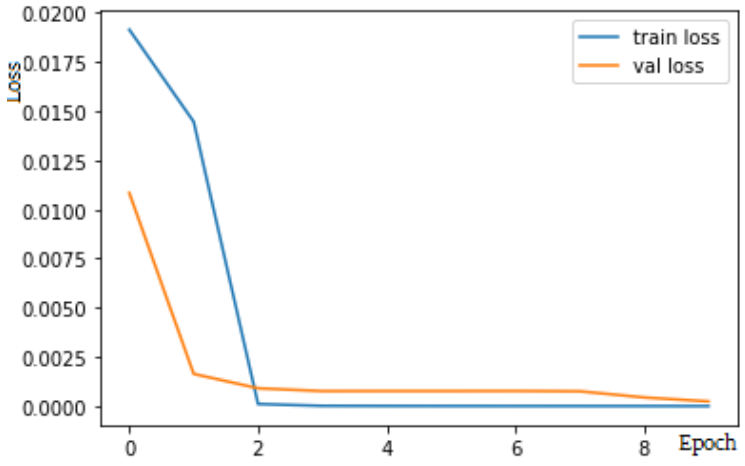

Figure 5.3: Loss Graph of proposed model on LeapGestRecogn dataset

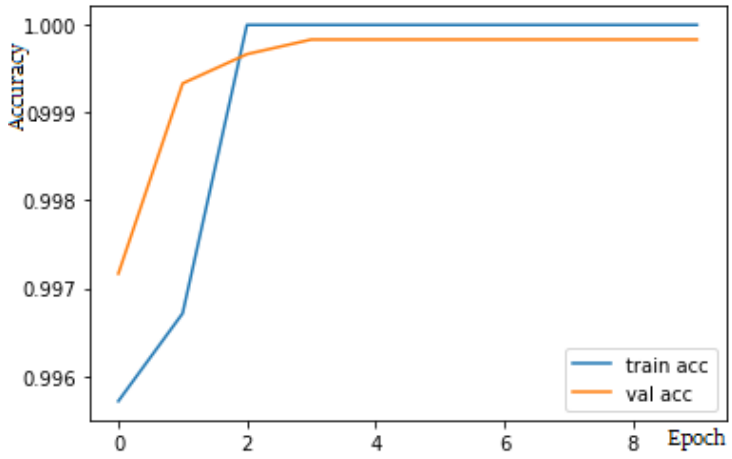

Figure 5.4: Accuracy Graph of proposed model on LeapGestRecogn dataset

In Fig. 5.5 and Fig. 5.6 shows the loss and accuracy graphs of Proposed Convolutional Neural Network model on LeapGestRecogn dataset.

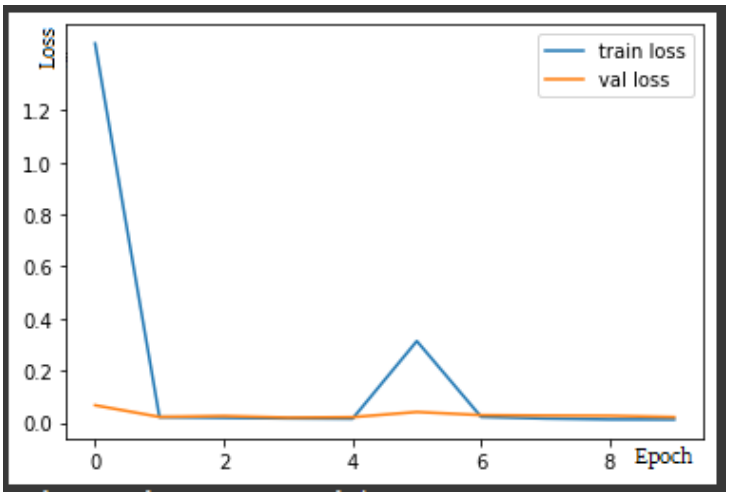

Figure 5.5: Loss Graph of proposed model on Fingers number count dataset

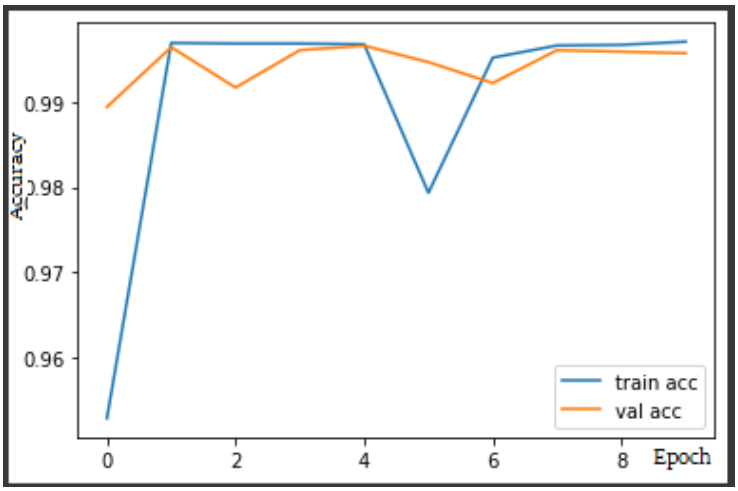

Figure 5.6: Accuracy Graph of proposed model on Fingers number count dataset

In Fig. 5.7 and Fig. 5.8 shows the loss and accuracy graphs of Proposed Convolutional Neural Network model on Fingers number count dataset.

5.4 Comparison of proposed model with the existing techniques

In table 5.4, different existing works is compared with the proposed Model on LeapGestRecogn Dataset. Can C et al. (2021) [21] and Enikeev D et al. (2020) [22] has implemented convolutional neural network and obtained an accuracy of 98.9% and 95% whereas the proposed convolutional Neural Network outperforms with an accuracy of 99.9% accuracy in LeapGestRecogn Dataset.

Table 5.5: Comparison of proposed technique with the existing techniques (LeapGestRecogn Dataset)

Work	Technique	Accuracy (%)
Can C et al. (2021) [21]	Convolutional Neural Network	98.9
Enikeev D et al. (2020) [22]	Convolutional Neural Network	95
Proposed work	**Convolutional Neural Network**	**99.98**

5.5 Performance with different pretrained Models

Transfer learning increases efficiency and resource conservation when developing new models. Since the majority of the model has already been trained, it is suitable for training on existing unlabeled datasets. Various pretrained models including VGG-16, ResNet 50 and MobileNet were analysed on LeapGestRecogn dataset. Table 5.3 shows the performance analysis of various transfer learning techniques on LeapGestRecogn dataset.

Table 5.6: Performance analysis of various transfer learning techniques on LeapGestRecogn dataset

Classifier	Accuracy(%)	Precision	Recall	F1-Score
VGG-16	35	0.32	0.64	0.43
ResNet 50	47	0.40	0.70	0.53
MobileNet	65	0.64	0.510	0.601
Proposed CNN	**99.98**	**0.999**	**0.9998**	**0.9999**

In a CNN, the number of parameters affects both training time and computation speed. A smaller number of parameters results in performing fewer mathematical calculations during testing, which speeds up the process. Additionally, training will occur more quickly with less number of epoch. The number of parameters to be learnt are summarized in Table 5.7. Amongst the four networks, VGG16 model required the highest number of parameters while proposed CNN required the least number of parameters to be learnt. A smaller number of parameters results in performing fewer mathematical calculations during testing, which speeds up the process. Additionally, training will occur more quickly with less number of epochs.

Table 5.7: Number of trainable parameters

Classifier	Total number of layers	Number of trainable parameters (in millions)
VGG16	16	138
ResNet50	50	23
MobileNet	28	13
Proposed CNN	**8**	**4**

Chapter 6

Conclusion

An automated hand gesture recognition system analyse the different gesture with effective detection. It is important to have a reliable feature selection and classification methods since the wrong prediction and detection of gestures are very crucial while creating an AI assistant system. This work presented an light weight Convolutional Neural network based model for efficient detection and classification of different hand gestures. Different performance metrics, including accuracy, precision, recall, and F1 Score, were used to assess the model's performance. The proposed Convolutional Neural network attained the highest classification accuracy of 99.98% and 99.79% in the LeapGestRecogn and Fingers number count datasets respectively. Pre-trained models such as VGG-16, ResNet-50 and MobileNet are employed on same datasets for comparison. The results indicate that the suggested model surpassed the pre-trained models in all evaluation measures. The real-time gesture recognition systems are then implemented for cognitive AI assistant systems to help recognise gestures more effectively with less parameters.

Chapter 7

Future Scope

In future, a real time hand gesture recognition model with better performance metrics and low latency for space applications has to be developed. The model which recognize multiple gestures in complex environments will bring reliability to the system. For the model to perform better, should be trained on large datasets with many more different gestures. Additionally, the issue of distance restrictions and input lighting variability needs to be resolved.

References

[1] Tsai TH, Huang CC, Zhang KL. Design of hand gesture recognition system for human-computer interaction. Multimedia tools and applications. 2020 Mar;79(9):5989-6007.

[2] Shin S, Kim WY. Skeleton-based dynamic hand gesture recognition using a part-b ased GRU-RNN for gesture-based interface. Ieee Access. 2020 Mar 11;8:50236-43.

[3] Sarma D, Bhuyan MK. Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review. SN Computer Science. 2021 Nov;2(6):1-40.

[4] Oudah M, Al-Naji A, Chahl J. Hand gesture recognition based on computer vision: a review of techniques. journal of Imaging. 2020 Aug;6(8):73.

[5] T. Mantecón, C.R. del Blanco, F. Jaureguizar, N. García, Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller, Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, pp. 47-57, 24-27 Oct. 2016. (doi: 10.1007/978-3-319-48680-2_5)

[6] Li, X. Liu, M. Zhang and D. Wang, Spatio-temporal deformable 3D convnets with attention for action recognition, Pattern Recognition, vol. 98, pp. 107037, 2020.

[7] F. Obaid, A. Babadi and A. Yoosofan, Hand gesture recognition in video sequences using deep convolutional and recurrent neural networks, Applied Computer Systems, vol.

[8] N. L. Hakim, T. K. Shih, S. P. Kasthuri Arachchi, W. Aditya, Y.-C. Chen et al., Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model, Sensors, vol. 19, no. 24, pp. 5429, 2019.

[9] P. Nguyen and T. N. Luong, Two-stream convolutional network for dynamic hand gesture recognition using convolutional long short-term memory networks, Vietnam Journal of Science and Technology, vol. 58, no. 4, pp. 514, 2020.

[10] O. Köpüklü, A. Gunduz, N. Kose and G. Rigoll, Real-time hand gesture detection and classification using convolutional neural networks, in 14th IEEE Int. Conf. on Automatic Face Gesture Recognition, Lille, France, pp. 1–8, 2019.

[11] Y. Zhang, L. Shi, Y. Wu, K. Cheng, J. Cheng et al., Gesture recognition based on deep deformable 3D convolutional neural networks, Pattern Recognition, vol. 107, pp. 107416, 2020.

[12] Rehman MU, Ahmed F, Khan MA, Tariq U, Alfouzan FA, Alzahrani NM, Ahmad J. Dynamic Hand Gesture Recognition Using 3D-CNN and LSTM Networks. CMC-COMPUTERS MATERIALS CONTINUA. 2022 Jan 1;70(3):4675-90.

[13] Liu Y, Jiang D, Duan H, Sun Y, Li G, Tao B, Yun J, Liu Y, Chen B. Dynamic gesture recognition algorithm based on 3D convolutional neural network. Computational Intelligence and Neuroscience. 2021 Aug 17;2021.

[14] Liu C, Szirányi T. Real-Time Human Detection and Gesture Recognition for On-Board UAV Rescue. Sensors. 2021 Jan;21(6):2180.

[15] Xu C, Jiang Y, Zhou J, Liu Y. Semi-Supervised Joint Learning for Hand Gesture Recognition from a Single Color Image. Sensors. 2021 Jan;21(3):1007.

[16] Ahmed S, Cho SH. Hand gesture recognition using an IR-UWB radar with an inception module-based classifier. Sensors. 2020 Jan;20(2):564.

[17] Neethu PS, Suguna R, Sathish D. An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. Soft Computing. 2020 Oct;24(20):15239-48.

[18] Parveen N, Roy A, Sai Sandesh D, Sai Srinivasulu JY, Srikanth N. Human computer interaction through hand gesture recognition technology. International Journal of Scientific and Technology Research. 2020;9(4):505-13.

[19] Deng J, et al. Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition,2009. CVPR 2009. IEEE;

[20] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016 pp. 770–778;

[21] Can C, Kaya Y, Kılıç F. A deep convolutional neural network model for hand gesture recognition in 2D near-infrared images. Biomedical Physics Engineering Express. 2021 Jul 9;7(5):055005.

[22] Enikeev D, Mustafina S. Recognition of sign language using leap motion controller data. In2020 2nd international conference on control systems, mathematical modeling, automation and energy efficiency (SUMMA) 2020 Nov 11 (pp. 393-397). IEEE.

[23] https://www.grandviewresearch.com/industry-analysis/gesture-recognition-market

[24] Deng J, et al. Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition,2009. CVPR 2009. IEEE;

[25] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016 pp. 770–778;