DYNAMIC HAND GESTURE RECOGNITION FRAMEWORK FOR HUMAN-COMPUTER INTERACTION

Thesis submitted to Delhi Technological University

in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

In

INFORMATION TECHNOLOGY

By

REENA TRIPATHI

(2K20/PHDIT/02)

Under the Supervision of

DR. BINDU VERMA

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

NEW DELHI-110042, INDIA

NOVEMBER-2024

Acknowledgements

I am profoundly grateful to the many individuals who have contributed to the successful completion of my PhD journey. Their unwavering support, guidance, and encouragement have been invaluable throughout this endeavour. First and foremost, I express my sincere gratitude to my advisor, **Dr. Bindu Verma**. Your mentorship, dedication, and passion for advancing knowledge have inspired me continuously. The insights and expertise you provided shaped my research trajectory and deepened my understanding of the field. I am truly fortunate to have worked under your guidance. I would also like to thank my SRC members: **Prof. Dinesh Kumar Vishwakarma, Dr. Pawan Singh Mehra**, and **Dr. Virender Ranga**, whose valuable guidance and support were instrumental in shaping my thesis. My heartfelt appreciation extends to the esteemed faculty of the IT Department at DTU, including **Prof. Kapil Sharma, Prof. Seba Susan, Dr. Ritu Agarwal, Ms. Anamika Chauhan, Dr. Priyanka Meel, Dr. Rahul Gupta, Dr. Varsha Sisaudia** and **Ms. Geetanjali Bhola**. Additionally, I am deeply grateful to **Mr. Rajesh Dangi**, and **Mr. Bhoop Singh** for their encouragement and assistance.

I would like to dedicate a special part of my acknowledgments to my beloved grandparents, **Late Shri Ratanmani Tripathi** and **Late Mrs. Bhagirathi Devi Tripathi**. Though they are no longer with us, the values they imparted continue to inspire and motivate me. I am deeply grateful to my parents, **Prof. Devi Prasad Tripathi** and **Mrs. Savitri Devi Tripathi** for their unwavering support, sacrifices, and faith in my abilities. Their encouragement has been the foundation of my academic pursuits. I also extend my gratitude to my uncles, **Mr. Rajender Prasad Tripathi** and **Mr. Ramesh Tripathi**; my aunts, **Mrs. Anusuya Devi Tripathi** and **Mrs. Manisha Tripathi**; my elder brother, **Mr. Bhagwati Prasad Tripathi**, and sister-in-law, **Mrs. Sandhya Tripathi**; my elder brothers, **Dr. Durga Prasad Tripathi** and **Mr. Chandrashekhar Tripathi**; my younger brother, **Chandra Bhushan Tripathi**; my sisters, **Ranjana Tripathi, Anjana**

Tripathi, Pradidhi Tripathi, and **Pragya Tripathi**; my niece, **Padmakshi Tripathi**; and my nephews, **Medhesh and Shivesh Tripathi**. Each of you has been a vital source of strength throughout this journey.

I also deeply appreciate the guidance provided by **Dr. Kamal Pathak** at Guru Govind Singh Indraprastha University and **Prof. Kapil Sharma** at Delhi Technological University, who generously offered essential resources for my research. I extend my thanks to **Prof. Indu Prakash Upadhyay**, and **Prof. Ramchandra Pandey** at Banaras Hindu University(BHU) for their constant support, as well as to **Mr. Uday Singh Rawat, Prof. Ashok Thapaliyal** and **Mr. Narendra Kumar Sharma, Er. Abhay Kumar Mishra, Prof. Meenakshi Mishra**, and **Mrs. Neera Upadhyay**. To my friends: **Gitika Rawat, Ishveen Kaur, Shelja Deswal, Avyaqt Raina, Deepak Vashistha, Ikroop Singh, Devansh Upadhyay, Nidhi, Lakshita Agarwal, Sunakshi Mehra, Santosh Kumar Ray, Akanksha Karotia,** and **Abhishek Verma**, for their unwavering support and encouragement. Their presence made challenging times more bearable and moments of success more joyful. I extend my gratitude to everyone whose names may not appear here but who has contributed to my academic journey. Their support, patience, and kindness have been invaluable to my personal and professional growth. This journey has been possible only with the collective support of all these remarkable individuals, for which I am deeply thankful.

(Reena Tripathi)

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road-Delhi-42

Certificate

This is to certify that the thesis entitled **"Dynamic Hand Gesture Recognition Framework for Human-Computer Interaction"**, being submitted by **Reena Tripathi** (Enrollment Number 2K20/PHDIT/02) to the **Department of Information Technology, Delhi Technological University, India,** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Information Technology,** is an authentic record of work carried out by her under the guidance and supervision of **Dr. Bindu Verma**.

This research work has not been submitted, in part or full, to any other University or Institution for the award of any degree or diploma.

Date: 28/11/2024
Place: New Delhi

Reena Tripathi
(Ph.D. Student)

9/12/2024
(Supervisor)
Dr. Bindu Verma
(Assistant Professor)
Department of Information Technology
Delhi Technological University, Delhi- 42

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road-Delhi-42

Declaration

I hereby declare that the Ph.D. thesis entitled **"Dynamic Hand Gesture Recognition Framework for Human-Computer Interaction"** being submitted to the **Delhi Technological University, Delhi,** in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** in **Department of Information Technology,** is an authentic record of work carried out by me under the guidance and supervision of **Dr. Bindu Verma**.

I also mention that the research work is original and has not been submitted by me, in part or full, to any other University or Institution for the award of any degree or diploma.

Reena Tripathi
(Ph.D. Student)
Department of Information Technology,
Delhi Technological University, New Delhi-110042 India

Dedicated to My Parents: for their love, endless support, encouragement, and unwavering cooperation throughout this entire journey...

Abstract

To communicate with one another hand gesture is very important. The task of using the hand gesture in technology is influenced by a very common way humans communicate in the natural environment. In the early days of interaction with a computer, the user uses a keyboard, mouse, pen. Similar type of communication can be possible using hand gesture that replaces the hardware devices and reduces the cost of hardware. Due to the advancement of technologies and the digital era the need of human-computer interaction(HCI) techniques needs to grow. Hand gesture recognition is a one of the possible way that makes human interaction with the computers. Hand gestures have numerous applications in daily life, ranging from controlling automatic vehicles to enhancing smart home development and human-robot interaction. They are used in clinical operations where surgeons can handles MRI or X-ray scans through hand gestures. In sign language recognition, hand gestures enable communication among the deaf community. In robotics, dynamic hand gestures control robot movements, 3D hand gesture recognition facilitates real-time human-computer interaction. Hand gestures also play a crucial role in home automation, controlling appliances like lights, fans, and security systems. For computers and tablets, gestures are used to drag, drop, and move files, improving human-computer interaction. The recognizing and finding gesturing hand comes under the area of hand gesture analysis. To find out the gesturing hand is very difficult than finding the another part of the human body because the smaller size of the hand. The hand has greater complexity and more challenges due to various factor such as hand occlusion, background clutter, lighting illumination and inter and intra-class variation. These factors affect the accuracy of dynamic hand gesture. Real-time recognition of dynamic hand gesture is difficult because the algorithm can't determine with accuracy where a gesture starts and ends in a video feed.

Dynamic hand gesture recognition (DHGR), which involves understanding gestures in motion over time, poses various challenges. These include variations in lighting, occlusions, complex backgrounds, and similarities between gestures, within the

same category (Intra-class) and across different categories (Inter-class), making detection and recognition difficult. Traditional models often find it challenging to address these issues, particularly when working with a small or limited dataset. Further, integrating dual-modality and multi-modality where RGB data, skeletal data, and depth information integrated in the model makes more challenging. The afford mentioned challenges motivated us to work in the field of dynamic hand gesture recognition addressing the various research gaps such as solving the problem of hand detection and tracking, inter and intra-class variation, hand occlusion and efficient and genic framework.

This thesis aims to develop efficient models that handle these issues, work well with limited data, and perform reliably under diverse conditions. Initially, In first framework, we solve the challenge of hand detection and tracking where RGB videos are used to extract the features using CLIP model. The CLIP-BLSTM model is specifically designed to address challenges associated with small hand sizes and changing lighting conditions, proving to be efficient with fewer training samples and parameters. Overall, it performs effectively in different lighting environments, establishing it as an accurate hand gesture recognition system. Further, extraction of skeleton data from RGB data and use of skeleton data in the proposed models overcome the challenges of background clutter and gesturing hand tracking. Same gesture may perform differently by different persons arises the concern on inter-class and intra-class variation problem. To tackle inter-class and intra-class variation, DDA Loss is employed to enhance within-class similarity gesture and reduce the between-class similarity gesture. In this work, skeleton data is used to create skeleton point trajectories, and DDA loss is used to enhance the feature learning so that intra-class similarity increases and inter-class similarity decreases.

In the literature we analyze that use of multiple modalities compare to the single modality performs well on the deep learning models and boost the performance. Thus, we also work on dual-modality and multiple-modality. In the next model we combined skeletal data with RGB data to recognize the dynamic hand gesture. Our proposed model offers a bidirectional gated recurrent unit (Bi-GRU) model based

hand gesture recognition system that is computationally effective than Bi-LSTM. This method is designed to attain high-speed performance while being capable of working successfully even with limited training samples. This dual-feature extraction method allows the model to achieve a more robust understanding of hand gestures, improving overall performance in diverse environment. However, limited research has been conducted on multi-modal fusion as combination of multiple modalities can boost the performance. In the next work we developed a hybrid framework that integrates RGB, depth, and skeleton data to create an efficient system for dynamic hand gesture recognition. An extensive experimental study conducted on various standard datasets such as SKIG, DHG14/28, NWUHG, FPHA, LISA, 26-Gestures, NTU, NTU120, and CHG illustrates the effectiveness of our all proposed frameworks.

Author Research Publications

Journals

- **Reena Tripathi**, and Bindu Verma. "Survey on vision-based dynamic hand gesture recognition." *The Visual Computer*(2023): 1-29 **(SCIE Indexed, IF: 3) DOI: https://doi.org/10.1007/s00371-023-03160-x** ***(Published)***

- **Reena Tripathi**, and Bindu Verma. "Motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recognition." *Signal, Image and Video Processing* (2024): 1-10.. **(SCIE Indexed, IF: 2) DOI: https://doi.org/10.1007/s11760-024-03153-w** ***(Published)***

- **Reena Tripathi**, and Bindu Verma. "Tri-Modal Fusion for Dynamic Hand Gesture Recognition: Integrating RGB, Depth, and Skeleton Data" is communicated in *Journal of Visual Communication and Image Representation* **(SCIE Indexed, IF: 2.6)** ***(Communicated)***

- **Reena Tripathi**, and Bindu Verma. "Ensemble Learning with DDALoss for Inter and Intra Class Variation in Hand Gesture Recognition" is communicated in *Signal, Image and Video Processing* **(SCIE Indexed, IF:2)** ***(Under Review)***

Conferences

- **Reena Tripathi**, and Bindu Verma. "Skeleton Data is all about: Dynamic Hand Gesture Recognition.". *In 2023 Seventh International Conference on Image Information Processing (ICIIP)* (pp. 576-585) (2023, November). IEEE. ***(Published)***

- **Reena Tripathi**, and Bindu Verma. "CLIP-LSTM: Fused Model for Dynamic Hand Gesture Recognition." presented in *In 2023 IEEE 20th India Council International Conference (INDICON)*, pp. 926-931. IEEE, 2023. ***(Published)***

Contents

List of Tables

List of Figures

List of Abbreviations

1DCNN	1D Convolutional Neural Network
2DCNN	2D Convolutional Neural Network
3DCNN	3D Convolutional Neural Network
ANN	Artificial Neural Network
AU	Action Unit (AU)
AUC	Area Under the Curve
BE	Behaviour Encoder
Bi-GRU	Bidirectional Gated Recurrent Units
CHG	Cambridge Hand Gesture
CNN	Convolutional Neural Network
CLIP	Contrastive Language-Image Pre-training
DDA	Discriminant Distribution-Agnostic
DHG	Dynamic Hand Gesture
DHGR	Dynamic Hand Gesture Recognition
DG-STA	Dynamic Graph-Based Spatial-Temporal Attention
DTW	Dynamic Time Warping
FMF	Finger motion features
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FPHA	First-person hand action
FSTA	Fine-grained Spatio-temporal Attention
GDA	Grassmann Discriminant Analysis
GGDA	Grassmann Graph Discriminant Analysis
GEM	Global enhancement model
GMF	Global motion features
GRU	Gated Recurrent Units

HBU-LSTM	Hybrid Bidirectional Unidirectional LSTM
HCI	Human-Computer Interaction
HGR	Hand Gesture Recognition
HGSS	Hand Gesture Skeleton Sequences
HGMF	MC-histogram gradient magnitude frequency
HGSS	hand gesture skeleton sequences
HMM	Hand Movement Map
HOG3D	3D histogram of oriented gradient
HOG	Histogram of Oriented Gradient
HPEV	Hand Posture Evolution Volume
IndRNN	Independently Recurrent Neural Network
KNN	k-nearest neighbor
LISA	Laboratory for Intelligent and Safe Automobiles
LSTM	Long-Short Term Memory
LSTSN	Local Spatial-Temporal Synchronous Network
MHI	Motion History Images
MNN	Multiple Layer Neural Network
MP	Max Pooling
MPM	Motion perception module
NN	Neural Network
NWUHG	North Western University Hand Gesture Dataset
OF	Optical Flow
RBi	Residual Bidirectional
RGB	Red Green Blue
RNN	Recurrent Neural Network
ResNet	Residual Network
ResCLIP	Resudal Contrastive Language-Image Pre-training
ROC	Reciver Operating Characteristic
SAGCN	Self-Attention Graph convolutional Network
sEMG	Surface Electromyography

SKIG	Sheffield Kinect Gesture
SOM	MSelf-organizing Map
STST	spatial-temporal synchronous transformer
SVM	Support Vector Machine
TN	True Negative
TOF	Time of flight
TP	True Positive
TPR	True Positive Rate
TSM	Temporal Shift Modules
TSN	Temporal Segment Networks
VGG	Visual Geometry Group
VGG16	Visual Geometry Group 16
YOLO	You Only Look Once

Chapter 1

Introduction

To communicate with one another hand gesture is very important. The task of using the hand gesture in technology is influenced by a very common way humans communicate in the natural environment [1]. In the early days of interaction with a computer, the user uses a keyboard, mouse, pen. Similar type of communication can be possible using hand gesture that replaces the hardware devices and reduces the cost of hardware. Earlier gloves and sensors-based trackers were there that were used to communicate with the computer but they were not successful due to the cost of wearable devices. Moreover, user needs to wear these device's that hinders the naturalness of the hand gesture and very uncomfortable to wear such type of devices. Then vision-based hand gesture recognition came into picture, where user performs the hand gesture in front of the camera, and corresponding action is triggered. The vision-based hand gesture is the way by which we give a signal to the computer system. It is a non-contact technique for giving input. Hand gesture is of two types (i) Static hand gesture which contains the shape of the hand, palm and fingers (ii) Dynamic hand gesture which contains the movement of the hand with shape and contains spatio-temporal information.

Hand gestures have numerous applications in daily life, ranging from controlling automatic vehicles to enhancing smart home development and human-robot interaction. They are used in clinical operations where surgeons can handles MRI or X-ray scans through hand gestures. In sign language recognition, hand gestures enable com-

munication among the deaf community. In robotics, dynamic hand gestures control robot movements, 3D hand gesture recognition facilitates real-time human-computer interaction. Hand gestures also play a crucial role in home automation, controlling appliances like lights, fans, and security systems. For computers and tablets, gestures are used to drag, drop, and move files, improving human-computer interaction. In gaming, users engage with games through hand and body motions tracked by Kinect sensors, while in automatic vehicles, gestures can control in-vehicle menus such as music systems and navigation. Lastly, smart devices utilize hand gestures for functions like capturing photos or opening doors in smart stores. From the various studies, we witness that there are two methods used to interact between humans and computers using hand gestures, data glove-based, and vision-based approaches. In the data glove-based approach, a sensor is attached to the gloves by electric signals and hand postures are observed. This approach involves the physical connection of humans and computers via cables. Data gloves have various advantages, like they obtain hand joint data, and are suitable for small signal interference. The disadvantage of data gloves is that the user needs to wear these devices, which hinders the naturalness of the hand gesture and makes it very uncomfortable to wear such devices. Also, the cost and maintenance of the data gloves are high. In contrast, the vision-based hand gesture data is captured through the camera, and the gesture is performed in front of the camera. Data captured through the camera can be in the form of RGB data, depth data, and skeleton data. This technique is non-physical and does not require the users to wear any device, and we get the natural and raw image of the hand. The advantage of this technique is that it is computationally efficient, generic robust, and easy to perform.

1.1 Vision-based Hand Gesture

In vision-based, data is captured through the camera or gesture is performed in front of the camera. There are different sensors used to collect the data such as RGB cameras, Kinect sensors, Infra red cameras, webcam etc. These cameras captures the

RGB data, depth data, skeleton data, marker data and 3D data. Then that data is processed and corresponding gesture is recognized. This technique is non-physical and users do not need to wear any such device and we get the natural image of the hand. Advantage of this technique is this is computationally efficient, generic robust, and easy to perform. Moreover, vision-based hand gesture recognition systems can be used in many real-time applications such as operating TV menus, gaming, navigation, in medical imaging and sign language recognition etc. Following we have discussed various modality based dynamic hand gesture recognition.

1.1.1 Color-based(RGB) Recognition

For feature extraction, RGB video data is passed through the model. RGB based model has several benefits it gives color information about the gesturing hand, including specifics like clothing, skin tone, and objects used in the gesture. RGB data also provides spatial information that helps to interpret the shape of the gesturing hand, its texture, and other relevant features of the hand. Overall, the RGB data actual visualization of the hand gesture and gives textual and visual information. RGB data may encounter challenges such as illumination variations, occlusion, and background clutter, which can hinder hand gesture recognition. In contrast, skeleton data overcome these challenges.

1.1.2 Skeleton-based Recognition

Skeleton data can be captured by the Kinect V2 camera and skeleton data gives the 21 land mark points of the hand gesture. The construction of a skeleton model of the hand structure includes key joints including the wrist, palm, and finger joints such as the knuckles and fingers. Based on the spatial configuration of the hand joints geometric features can be extracted and helps in understanding the performed gestures. Skeleton-based recognition is based on the features of the skeleton data. Skeleton information gives the basic structure of the hand shape and hand articulation points in a 2D or 3D space. It makes easier to precisely track hand gesture movements by

giving information about the finger tip of the gesturing hand. Because skeleton data is not dependent on background information, it overcomes the challenges of dynamic hand gesture recognition such as occlusion, cluttered backgrounds, and changing illumination.

1.1.3 Depth-based Recognition

A depth map gives a distance of the scene object from the camera viewpoint. The depth map can be used to segment the hand or detect the hand by giving the depth range. The depth information can also be used to reduce the noise in the data, remove the shadow, and use for background subtraction. Each frame of the depth information gives the model the ability to understand the gesture's spatial configuration in three dimensions. Depth data is more dependable under a range of lighting situations because it is less affected by illumination than RGB data. Additionally, depth data mitigates issues related to occlusion since it directly captures physical object distances from the sensor, bypassing potential obstructions. Depth data also helpful in the segmentation of the gesturing hand.

1.1.4 Multi-Modality Recognition

In the multi-modality based hand gesture recognition a combination of multiple modalities is used to recognize the dynamic hand gesture. Multiple modality combinations can be, a combination of RGB and depth, depth and skeleton, skeleton and RGB etc. When multiple modalities are used together, accuracy is frequently increased as compared to a single modality. This is so that the model may use a variety of information sources to help it make a better feature learning. Multi-modal fusion enhances the model's ability to generalize across different environments, lighting conditions, and hand orientations, leading to better performance in real-world scenarios.

The various stages of gesture detection, include data acquisition, hand detection

and tracking across frames, feature extraction, and classification. During the data acquisition stage, hand gestures are performed in front of sensors, which can include an infrared camera, RGB camera, depth camera, or skeleton camera. In hand gesture classification, the traditional approach is set a milestone, that fails only when the images are noisy and cluttered. In such scenario, the hand tracking of each frame is a tough task and hand crafted features may not be good enough to propose a generic and robust systems and degraded the accuracy of the model. In current state, deep learning based methods shows good recognition accuracy in various gesture recognition system. Due to the advancement of computing devices and introduction of deep neural networks, various deep learning framework has been proposed in the field of dynamic hand gesture recognition and these framework gives tremendous accuracy on the various bench mark datasets. In the deep learning based hand gesture recognition system have only two stages that is data pre-processing and hand gesture classification. In pre-processing image enhancement, noise removal, and image resizing and data augmentation can be done. Then the particular images/videos passed to the deep network in which automatic feature extraction and classification is done using various deep learning networks.

1.2 Research Gaps, Challenges and Motivation

We surveyed the current state of the art methods for dynamic hand gesture recognition, and enumerate some research gaps, challenges and motivation to work in the field of dynamic hand gesture recognition. Though the research on vision-based hand gesture recognition has been extensive and many valuable achievements have been made, to date, the reliability and practicality of these hand gesture recognition systems are still a challenging.

- The main challenge in hand gesture recognition is accurately tracking hand movements to recognize dynamic gestures under poor lighting and cluttered backgrounds. Among the wide variety of vision based hand gesture recognition methods, some can achieve very good recognition rates in certain restrictive

environments, but may not be applicable yet to real world situations. Detecting and tracking the hand is difficult due to its small size and complex structure compared to the whole body.

- Background interference which occurs during the segmentation process, such as lighting, brightness, similar colours, overlapping areas, or similar objects in the background, can generate very divergent segmentation results. Human eyes are able to differentiate between foreground and background easily while machine can not.

- Inter-class variation refers to differences between different gestures that may have similar patterns, making them hard to distinguish. An intra-class variation involves differences within the same gesture. Both types of variation complicate accurate gesture recognition.

- Camera viewpoints and gesture occlusion is also a problem due to the natural motion of the hand, certain gestures inherently include partial occlusion which affects the systems recognition accuracy. Moreover, motion blur and quiescent camera conditions pose problems, the first occurs while tracking a dynamic gesture in motion.

- Hand articulation and occlusion make hand gesture detection challenging, while the hand tracking process faces challenges like complex background, dynamic background, and illumination variation.

To make a robust gesture recognition system, hand detection, and tracking steps must be performed flawlessly to propose a generic system. Segmenting the gesturing hand in cluttered, complex, or dynamically changing backgrounds is challenging due to issues like image resolution, clothing, and lighting variations. Finally, extracting relevant features from the hand and accurately defining gestures is one of the biggest challenges in hand gesture recognition. Due to the aforementioned challenge motivates us to work in the area of dynamic hand gesture recognition. We have proposed deep

learning based dynamic hand gesture recognition framework that uses single and multiple modalities.

1.3 Problem Definition

Hand gestures give individuals a simple and intuitive way to communicate with technology, opening up opportunities in virtual reality, gaming, smart home automation, etc. Dynamic hand gesture recognition (DHGR), which involves understanding gestures in motion over time, poses various challenges. These include variations in lighting, occlusions, complex backgrounds, and similarities between gestures, within the same category (Intra-class) and across different categories (Inter-class), making detection and recognition difficult. Traditional models often find it challenging to address these issues, particularly when working with a small or limited dataset, or when integrating dual-modality models and multi-modal inputs, such as RGB images, skeletal data, and depth information, is required. This thesis aims to develop efficient models that handle these issues, work well with limited data, and perform reliably under diverse conditions. Initially, we have proposed, the CLIP-BLSTM model to overcome the problem of hand tracking and also efficient with fewer training samples and parameters. Further, extraction of skeleton data from RGB data and use of skeleton data in the proposed models overcome the challenges of background clutter and various lighting illumination. Further, same gesture may perform differently by different persons arises the concern on inter-class and intra-class variation problem. To tackle inter-class and intra-class variation, DDA Loss is employed to enhance within-class similarity gesture and reduce the between-class similarity gesture. In this work, skeleton data is used to create skeleton point trajectories, and DDA loss is used to enhance the feature learning so that intra-class similarity increases and inter-class similarity decreases. In the literature we analyze that use of multiple modalities compare to the single modality performs well on the deep learning models and boost the performance. Thus, we also work on dual-modality and multiple-modality. In the next model of the we combined skeletal data with RGB data to recognize the dynamic

hand gesture. However, limited research has been conducted on multi-modal fusion as combination of multiple modalities can boost the performance. In the next work we developed a hybrid framework that integrates RGB, depth, and skeleton data to create an efficient system for dynamic hand gesture recognition.

1.3.1 Research Objectives

OBJECTIVE 1: To develop a generic framework using RGB videos for dynamic hand gesture recognition.
OBJECTIVE 2: To design a hand gesture recognition framework that handles inter-class and intra-class variations.
OBJECTIVE 3: To investigate existing algorithm for extracting skeleton information of moving hand and proposed a framework to overcome hand tracking problem.
OBJECTIVE 4: To develop an efficient hybrid framework that will use RGB, depth, and skeleton data.

1.4 Contributions in the Thesis

In this thesis, we focus on dynamic hand gesture recognition and briefly address the few research research gaps in the field of dynamic hand gesture recognition. In this thesis four research objective were defined as mention in Section1.3.1 and we address all these objective one by one discussed below as contribution of the thesis:

(I) OBJECTIVE 1: *To develop a generic framework using RGB videos for dynamic hand gesture recognition.*

We proposed a method for dynamic hand gesture detection that uses the RGB videos to extract the features using the CLIP model and BLSTM model is used for sequence to sequence learning and recognize the hand gesture. Use of CLIP for feature extraction from RGB video solve the problem of hand detection and tracking. In the proposed model, the features extracted from CLIP are

fed to the pipeline of a neural network that consists of two Conv1D layers, a Bidirectional LSTM layer, and LSTM layer. The output of the Bidirectional LSTM layer is then fed to another LSTM layer. The model recognizes the dynamic hand gesture by adding a dense layer with a predetermined number of output classes and a softmax activation function. The proposed CLIP-BLSTM model is efficient with fewer training samples and parameters and experimental results shows that the model performs well under different lighting conditions, making it a rapid and accurate hand gesture recognition system.

(II) OBJECTIVE 2: *To design a hand gesture recognition framework that handles inter-class and intra-class variations.*

To address the problem of inter-class and intra-class variation, DDA loss is used that increases the with-in class similarity features and decreases the between class similarity features. In this work skeleton data is used to create a skeleton point trajectory, which helps overcome challenge of background clutter and illumination variations. However, the previous method performed well on RGB data but data may encounter challenges such as occlusion, and background clutter, which can hinder hand gesture recognition. Thus, in the proposed model skeleton data is used to create a skeleton point trajectory image, which helps overcome challenges of illumination variations, and complex backgrounds. Features are extracted from skeleton point trajectory image parallelly using VGG16, Inception V3 and DenseNet121 and then feature vectors from all models are then ensembled together and passed through the DDA loss function. The DDA loss combines center loss and compute the total loss and train the feature learning. The DDA loss function encourages features to be close to their respective class centers while being distinct from other class centers. The proposed model is an ensemble of three pre-trained neural networks (VGG16, InceptionV3, and DenseNet121) trained with DDA loss, which enhances within-class similarity for gesture accuracy, thereby improving overall performance

(III) OBJECTIVE 3: *To investigate existing algorithm for extracting skeleton in-*

formation of moving hand and proposed a framework to overcome hand tracking problem.

As in previous objectives, both RGB and skeletal data are crucial for extracting meaningful information necessary for reliable and accurate gesture recognition. To boost the performance of the model, the third model of this thesis combined skeletal data and RGB data. Skeleton data is extracted from RGB videos using Media-pipe. OpenPose Lib, Media Pipe, Hourglass network were available in the literature and we found that media pipe extract skeleton data properly despite of occlusion, illumination and rotation invariant. The proposed model is pipe-lined in two streams and carried out concurrently. In the first pipeline, skeleton data is used to create the skeleton point trajectory video. The advantage of using skeleton point video is that it overcomes the challenges of illumination, and complex background. In the second pipeline from RGB/Depth data optical flow video is calculated. The advantage of calculating the optical flow video is that it captures the hand motion and discards the stationary background. After calculating the skeleton and optical flow, video features are extracted using Xception-Net and represented in the form of Finger motion features(FMF) and Global motion features(GMF) matrix. Then these features are passed to the Bi-GRU unit for sequence-to-sequence learning. The output of both Bi-GRU units is averagely fused and is flattened at a fully connected layer. In the last SoftMax layer with cross-entropy loss is applied to get the final probability score.

(IV) OBJECTIVE 4: *To develop an efficient hybrid framework that will use RGB, depth and skeleton data.*

As we have seen in previous objective using dual modality features boost the performance of the modal. RGB and skeleton data helps in to extract the visual and geometrical features respectively and combining these features boost the performance of the model. Further, as depth data is less affected by the illumination and occlusion gives a most discriminating features. Thus, combin-

ing RGB, depth, and skeletal data creates a more robust and reliable gesture recognition framework. Accordingly, in the fourth model, we propose a computationally efficient ResCLIP-LSTM-based hand gesture recognition system. This approach is designed to achieve high-speed performance and can operate effectively with a smaller number of training samples. The proposed model begins by taking RGB, Depth, and Skeleton data as an input. The CLIP model is combined with the residual block for feature extraction of sequential data. First, features are extracted using the CLIP model individually, and the features are passed through the residual block. The sequential learning model followed by the processing of each modality through two Conv1D layers and an LSTM layer. After processing, the outputs from all modalities are concatenated into a single layer, set to 0.5 dropout regularization to avoid over-fitting. Finally, a Dense layer with SoftMax activation predicts the class probabilities.

1.5 Outlines of the Thesis

This thesis is divided into Seven chapters and three appendices.

1. **Chapter 1 Introduction**

   In this chapter, we present an introduction of thesis, including the problem statement and the motivation. We briefly address the challenges in hand gesture recognition. We also discuss our contributions and provide an outline of the thesis in this chapter.

2. **Chapter 2 Literature Survey**

   In this chapter, we present an overview of the state-of-the-art methods in dynamic hand gesture recognition. First, we have discussed the survey on traditional hand gesture and deep learning hand gesture recognition. Further, we have discussed the dynamic hand gesture with single, double, and triple modalities. We have also done detailed analysis of the state-of-the art methods and represented statistically in the form of pie-chart and tabular.

3. **Chapter 3 Develop an Efficient and Generic Framework using RGB Videos for Hand Gesture Recognition**

   This chapter introduces a novel framework called CLIP-LSTM model for dynamic hand gesture recognition using RGB videos. The CLIP-LSTM model is designed to overcome the challenges of hand detection and tracking. The proposed model used CLIP for feature extraction from RGB data. Additionally, BLSTM is used for classification, it is efficient with fewer training samples and parameters. The model performs well under different lighting conditions, making it an accurate hand gesture recognition system.

4. **Chapter 4 Develop a Hand Gesture Recognition Framework that will Reduce the Inter and Intra-Class Variation**

   In this chapter, we have presented an ensemble learning with a unique loss function called Discriminant Distribution-Agnostic(DDA) Loss. To address the problem of inter-class and intra-class variation, DDA loss is used that increases the with-in class similarity features and decreases the between class similarity features. In the proposed model, the features are extracted individually via VGG16, InceptionV3, and DenseNet121. Then ensemble of models, trained with the advanced loss function, the DDALoss function encourages features to be close to their respective class centers while being distinct from other class centers. It improves classification accuracy by combining different models' strengths in extracting features and using a strong loss calculation method to ensure effective training.

5. **Chapter 5 Motion Feature Estimation using Bi-Directional GRU for Skeleton-based Dynamic Hand Gesture Recognition**

   This chapter presents a hybrid deep-learning model called motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recognition. This method is designed to overcome challenges like illumination, cluttered background, and occlusions. The method improves hand gesture recognition accuracy by using skeleton data and optical flow, which helps to

overcome issues like occlusion and background clutter.

6. **Chapter 6 Hybrid Framework for Dynamic Hand Gesture Recognition using Multiple Modalities**

   In this chapter, we present a fusion of multiple-modality concepts in our proposed work, and each modality has its advantage. Fusion of multiple modality features boost the performance of the model. This approach is designed to achieve high-speed performance and can operate effectively with a smaller number of training samples.

7. **Chapter 7 Conclusion**

   This chapter summarizes the key findings and main contributions of the thesis. Along with that, future research directions are also discussed in this chapter.

8. **Appendix A Long Short-Term Memory(LSTM)**

   In this appendix, we explain in detail about Long Short-Term Memory(LSTM).

9. **Appendix B VGG16, Densenet121, Inception Net V3**

   In this appendix, we give a detailed explanation of VGG16, Densenet, and Inception Net.

10. **Appendix C Bi-GRU**

    In this appendix, we provide the details of the Bi-GRU model along with its formulation and architecture.

Chapter 2

Literature Survey

Reena Tripathi, and Bindu Verma. "Survey on vision-based dynamic hand gesture recognition." *The Visual Computer*(2023): 1-29 **(SCIE Indexed, IF: 3) DOI: https://doi.org/10.1007/s00371-023-03160-x** ***(Published)***

In this chapter, we review the state-of-the-art methods in dynamic hand gesture recognition on traditional and deep learning based dynamic hand gesture recognition. Furthermore, we have shown the detailed analysis of the state-of-the-art-methods in the form of pie-chart and tabular form.

In the literature hand gesture recognition systems proposed using traditional methods as well using deep neural methods. In traditional methods the process of hand gesture recognition method categories into the various stages such as data acquisition, image pre-processing, hand detection and tracking, feature extraction and classification. The input to the dynamic hand gesture is a videos which is a continuous sequence of frames as input to the model. While in deep learning based approaches pre-processing steps and deep neural network for automatic feature extraction and classification. There are various traditional as well as deep learning based methods has been proposed by research community to recognize the hand gesture.

2.1 Literature Survey on Traditional Hand Gesture Recognition Methods

Here, we have covered all the approaches used to recognize dynamic hand gestures using traditional methods. After collecting the data, move to the pre-processing stage, where it removes the noise and cleaning of data. Also, pre-processing steps are required to prepare the data based on the model. Many algorithms such as spatial filter [2], temporal median filter [3], Gaussian mixture model [4], and adaptive background mixture model [5] are used to extract the noise from the data. After pre-processing next step is hand segmentation and tracking of the gesturing hand for extracting the features. Hand segmentation involves identifying the pixels in an image that make up a hand and tracing the hand's trajectory across a video is known as hand tracking [6].

Due to the development of technologies and the digital era, the need for human-computer interaction (HCI) techniques needs to grow. The dynamic hand gestures movements are collected through optical flow [7], motion history images(MHI) [8] and dynamic image network [9]. The motion of moving objects is captured by these methods without the need to segment moving objects. However, feature extraction plays a vital role in recognizing hand gestures. Due to the image resolution and illumination variations, segmenting the gesturing hand from the background is challenging. Author's Verma et al. [10] segment the gesturing hand by using depth data and calculating the model hand trajectories and geometrical features. Then Grassmann discriminant analysis framework was used for gesture recognition. Similarly, Nguyen et al. [11] and Zhou et al. [12] also used manifold learning to recognize the dynamic hand gesture. In another work, Verma et al. [13] used a 3D histogram of oriented gradient for extracting hand-crafted features and then used Grassman discriminant analysis for gesture classification by substituting feature vector as a subspace on the manifold. Oh-Bar et al. [14] extracted global features with different histogram-of-Oriented Gradient (HOG) variations for automatic gesture recognition. Likewise, to represent the hand pose, Smedt et al. [15] used a hand skeleton data descriptor and

connected joint's shape. The author Shotton et al. [16] introduces a fast and robust method for predicting the human body poses. The researchers focus on the 3D technology without knowing about the preceding frame's information. They performed an experiment on a large dataset of image pairs, applied the machine learning technique, and obtained high accuracy on synthetic and real test sets. Similarly, the author Conseil et al. [17] worked on the shape of the hand; they used a Fourier descriptor to perform the shape of the hand, and match patterns for hand pose recognition. The SVM classifier is used for gesture recognition. The researcher aimed to combine two algorithms to get better accuracy. The author also worked on multi-scale curvatures and correlations of data. They include fingertip angles and the distance between two desired points. The authors, Kollorz et al. [18] propose a new gesture recognition classification technique, using Time of flight(TOF) sensors that are used for hand segmentation. By using depth features and x-y projection coordinates of the image, classify data quickly and simplistically. Lalit et al. [19] use depth matrix and adaptive Bayes classifiers for finding dynamic gestures. The researchers use a depth matrix and 1-nearest neighbor for recognition purposes. A Naïve Bayes classifier is used and gesture is operated via two methods, state-level and sequence level. Tang et al. [20] establish two new datasets for hand gesture purposes a) Hand gesture dataset and b) Action 3D dataset. They used image entropy (feature extraction method) for fast and better recognition with less chance of error. The author Chengjin et al [21] proposed the DGS Subspace Pursuit algorithm and the dynamic group sparsity model, which isolated the gesture features and reduced noise components. The experiment used the support vector machine (SVM) classifier, with a sparsity level of 48. In comparison to an OMPbased system, the testing findings revealed a 3.3% increase in recognition rate, and in a small dataset, they outperformed a CNN-based method. To replace dense optical flow estimation in multimodal techniques for hand gesture recognition (HGR), the author Gibran et al. [22] proposed a more affordable solution by combining hand segmentation masks and RGB frames. To increase the recognition rate of two real-time HGR method, named as Temporal Shift Modules (TSM) and Temporal Segment Networks (TSN), the authors uses a lightweight semantic seg-

mentation technique known as FASSDNet. The author Huiet al. [23] proposed vision based marker less methods for hand gesture recognition method taking depth image sequences as input . the proposed method include the extraction of temporal and spatial features from the input sequences, and emphasis on hand parsing and 3d finger tip for localization for hand gesture recognition. The authors Huiyue et al. [24] Initially decomposed a dynamic gesture and used hybrid model composed of hidden Markov model for time-series modeling and fuzzy neural network for fuzzy inference.

Various classifiers and feature extraction techniques are used in the traditional dynamic hand gesture recognition approach. In gesture representation and feature extraction, researchers need to extract features that should be invariant to affine transformation. The features can be geometric features, texture and pixel value, 2D and 3D model-based features, and motion trajectory features. We can use joint angles, hand location, surface textures, and surface illumination for spatial features. These features have limitations, such as hand occlusion, making hand representation difficult. If there is a distortion in motion trajectory, extracted features may not be distinguishable. The components extracted using a histogram of the oriented gradient may be affected by various lighting conditions and illumination. Due to the above-mentioned issues, the traditional method's accuracy is not up to the mark. In hand gesture classification, traditional methods have been effective but struggle with noisy and cluttered images, making hand tracking difficult and reducing accuracy. Deep learning-based methods have shown improved recognition accuracy in these scenarios, offering more robust and reliable systems.

2.2 Literature Survey on Deep Learning Based Methods

2.2.1 Dynamic Hand Gesture Recognition using Single Modality

In the literature, many authors have proposed deep learning-based frameworks using single and multiple modalities. The author Chen et al. [25] used single modality RGB data with short-term and long term features to classify the dynamic hand gesture. Similarly, author [26] also focuses on a single RGB modality as an input and finds the spatio-temporal features to classify the dynamic hand gesture using deep learning model.

The author Salih et al. [27] use surface electromyography (sEMG) technique for hand gesture recognition. They focused on the benefits of both EMG signals and depth vision for real-time recognition by using MNN(Multiple Layer Neural Network) as a classifier. According to the researcher, the experiment was done in two parts a) HSOM clustering that automatically labels the data and b) MNN classifier as a result MNN gives better accuracy as compared to others. The author(s) Hiroomi et al. [28] uses gesture spotting for hand gesture recognition. The researchers converted the input videos into feature vectors. Self-organizing Map(SOM)-Hebb was used as a classifier. Sharma et al. [7] proposed 2D-CNN to extract features from optical flow motion template and 3D-CNN to extract the features from RGB sequences. Then in the last, before the classification layer, features are fused to boost the classification accuracy. Mujahid et al. [29] proposed a lightweight model for gesture recognition based on YOLO (You Only Look Once) V3 and DarkNet-53 convolutional neural networks that do not require additional pre-processing, picture filtering, or image enhancement.

2.2.2 Dynamic Hand Gesture Recognition using multiple Modalities

Using a deep learning framework, skeleton and depth-based hand gesture detection also perform exceptionally well. To classify the performed hand gesture, authors Lai et al. [30] and Chen et al. [31] both use skeleton data and bidirectional recurrent neural networks (RNNs). In the literature, a different RNN and LSTM combination was proposed. Bi-directional LSTM is used by Li et al. [32] to identify the gestures. For activity and hand gesture recognition, Juan et al. [33] use a Long Short-Term Memory (LSTM) recurrent network along with a Convolutional Neural Network (CNN). With the use of skeleton data, Chen et al. [34] proposed dynamic graph-based spatial-temporal attention (DG-STA). The authors Devineau et al. [35] uses skeletal data to classify the hand gestures with the help of a new convolutional neural network. The author(s) Yangke et al. [36] proposed a novel approach for dynamic hand gesture recognition based on skeleton data. They focus on the global enhancement model(GEM) and Motion perception module(MPM) for improving feature maps and x,y, and z coordinates axis. Researchers use DHG 14/28 dataset and combine 2D-CNN and 3D-CNN to recognize hand gestures. Similarly, the author Yong et al. [37] also work upon dynamic hand gestures and focus on the problems that are created by hand gesture recognition like the joints connectivity. The proposed model was a hand gesture graph convolutional network which is an advanced version of spatial-temporal graph CNN of dynamic hand gesture recognition system. However, the RGB data can be affected by lighting conditions, leading to inconsistent results. To address this, some researchers used depth features, skeleton features and a fusion of RGB and depth features [38] or RGB and skeleton [39] features in their work.

In the literature multiple modality used to recognize the dynamic hand gesture. Author Zhang et al. [40], extract Spatio-temporal features using 3DCNN and ConvLSTM to classify the gestures. The author Verma et al. [10] uses skeleton and depth data information for fingertips and creates trajectories and Grassmann Graph Discriminant

Analysis(GGDA) is applied for gesture recognition. Similarly, author's Tripathi et al. [41] used skeleton trajectories extracted from the RGB data and optical flow information for RGB video with GRU model to classify the dynamic hand gesture. Zhang et al. [42] used Recurrent 3DCNN to classify the dynamic hand gesture, and Gibran et al. [43] used 3DCNN to classify the dynamic hand gesture. Qing et al. [44] used 3DCNN plus ConvLSTM to recognize dynamic hand gestures using 3D hand pose, depth, RGB, and skeleton data. The author Kankana et al. [45] used RGB and depth data with sparse low-rank scores for hand action recognition, it includes four main modules including CNN and RNN, that address frame level and video level classification. The author Mucha et.al [46] proposed two novel 2D hand posture estimation models for an egocentric view. These models aim to address challenges in dynamic hand gesture recognition, such as overlapping hand occlusion. The author(s) Daniel et al. [47] used hand gestures in the field of sign language and proposed framework for various applications like human-computer interaction, robotics, health-care unit, etc. In this paper, the researchers use thermal images as an input in the CNN model to classify the gesture. The authors Muneer et al. [48] proposed multiple deep neural network such as 3DCNN that learns a local and global features, sequence feature to recognize the dynamic hand gesture. The author(s) Yangke et al. [49] proposed a novel approach for dynamic hand gesture recognition based on skeleton data using DHG 14/28 dataset. They focus on the global enhancement model(GEM) and Motion perception module(MPM) for improving feature maps and x, y, and z coordinates axis by using 2D-CNN and 3D-CNN.

2.3 Review on Dynamic Hand Gesture Applications

Many Authors introduced a framework for different applications using dynamic hand gestures. The author's Bin et al. [50] use a hand gesture system for UAV (unmanned aerial vehicles) flight control systems. The prominent component of the system is based upon deep learning neural network. The leap motion devices are given as input, and a deep network is used to recognize the hand gesture. Mishra et al. [51]

focus on detecting the infant's hand and tracking the infant's hand using a recurrent neural network. Daniel et al. [52] used hand gestures in sign language and proposed a framework for various applications like human-computer interaction, robotics, health, care unit, etc. Andrea et al. [53] introduce a natural user interface(NUIs) platform based on dynamic hand gestures. The researchers aimed to develop a system that reduces driver distraction and works on the automotive condition using a convolutional neural network. Noorkholis et al. [54] worked on the application of hand gestures in the field of electronic gazettes like intelligent TV. By combining RGB and depth data as input for deep learning models, the authors used 3DCNN and LSTM for extracting the Spatio-temporal features and Finite State Machine to control the class decision for real-time applications. Nadia et al. [55] uses EMG sensors for gaming purposes and introduces a new architecture conv-GRU architecture for gesture recognition. The author Mohamed et al. [56] proposed a framework of GRU and 1DCNN for real-time application for hearing-impaired persons. They use media-pipe to locate hand-skeleton important key points. In order to offer a reliable representation of dynamic gestures, the author Rahul et al. [57] proposed an approach for encoding a depth gesture's videos into an encoded motion image (EMI), which is a single dynamic image. They use VGG16 and 2DCNN to classify the hand gestures. Author verma et al. [13] explain the working of intelligent vehicles through hand gestures. According to the authors, the hand gesture is utilized for controlling and monitoring in-vehicle task such as operating music, navigation, answering phone calls, switching the music menus, etc. Author Mahmud et al. [58] proposed a framework for English Capital Alphabet (ECA) recognition drawn by index finger. Dynamic Time Warping distances were determined between a template ECA and a test ECA for each ECA.

2.4 Detailed Analysis of the State-of-the-art-Methods

Due to the availability of the large size of the dataset and high-capacity hardware, deep learning emerged rapidly and gave excellent results in the field of dynamic hand gesture. After doing a literature review on detailed analysis we find that hand ges-

tures are challenging in tracking, shape changes, illumination variation, and cluttered background. A detailed description of the deep learning method of dynamic hand gestures is given in Table 2.1. As we can see, around 50% research work on dynamic hand gesture recognition is done using deep learning-based model and achieved a good accuracy. Most of the deep learning model focused on the skeleton data as compared to the RGB or depth data.

Table 2.1: The deep learning methods of dynamic hand gesture recognition

Paper	Dataset	Algorithm	Description
Sharma et al. [7]	Palm's Graffiti Digits and &self-collected in-house dataset	2DCNN &3DCNN	Motion template guided by optical flow is encoded into a an image for each video. C3D CNN model is used to process the RGB videos and 2DCNN model is used to process the motion template and in last both pipeline features are used to classify the dynamic hand gesture.
Li et al. [59]	SKIG, VIVA and & NV Gesture	3DCNN& Spatial attention mechanism	Proposed 3D ConvNet model for effective feature extraction and positive knowledge transfer framework from strong modality to low modality to classify the dynamic hand gesture.
Chuankun et al. [60]	DHG-14/28 & FPHA	SAGCN & RBi-IndRNN	Main focus on extracting short-term and long-term temporal information. Self-attention-based graph convolutional network mainly focus on the hand joints.
Li et al. [32]	ChaLearn LAP 2014 & SKIG	BLSTM	Desnly connected BLSTM used to classify the dynamic hand gesture.
Chunyong et al. [61]	DHG-14/28	Kalman filter-LSTM	Proposed a nested interval unscented Kalman filter (UKF) with LSTM framework for noisy hand gesture data.
Ameur et al. [62]	Leap gesture & RIT dataset	HBU-LSTM	Proposed Hybrid Bidirectional Unidirectional LSTM (HBU-LSTM) that handles sequential data generated by leap motion controller device.
Zhang et al. [63]	SKIG & LSA 14	3D CNN & ConvLSTM	Proposed alternate fusion of 3D CNN and ConvLSTM, which is called as the Multiple extraction and Multiple prediction (MEMP) network. MEMP network retains more spatial-temporal feature information through multiple information extraction and prediction of feature maps.
Chen et al. [31]	DHG-14/28	LSTM	Extract finger motion features that represent finger movement and global motion features that represent hand shape movement and in last features are fused and classified using BLSTM. Proposed model suitable when skeleton data is available.
Lai et al. [30]	DHG-14/28	CNN&RNN	The author used a combination of CNN and RNN where CNN used to extract the features and RNN used for sequence to sequence learning. This framework designed to handle occlusion and illumination issues by considering skeleton and depth data only.
Lei et al. [64]	DHG-14/28, SHREC17	De coupled spatial-temporal attention network (DSTA-Net)	Proposed a four de-coupled stream and trained separately such as spatial-temporal stream (original data),spatial stream, fast-temporal stream and slow-temporal stream. In last classification scores are averaged to obtain the final result.
Lui et al. [65]	DHG-14/28, SHREC17, FPHA	HMM-Net & HPEV-Net	Proposed two separate pipeline one for one for hand posture variations and other one for hand movements trajectory. Recognition results can be obtained by fusing the predictions of the two networks.

Continued on next page

Table 2.1 – continued from previous page

Paper	Dataset	Algorithm	Description
Devineau et al. [35]	DHG-14/28	CNN	They focus on mainly hand-skeleton joints for hand gesture classification where 2DCNN used to process sequences of hand-skeletal joints by parallel convolution. A limitation of the gesture recognition system is that it only functions on complete sequences and only on skeleton data.
Yangke et al. [36]	SHREC'17 & DHG-14/28	TP stream & SP stream	In SP-Stream proposed a novel compact joints encoding method to represent the geometric shape characteristics of the hand gesture and TP-Stream, proposed the motion perception module to capture the significant motion features of the hand gesture. Features are fused to get the final prediction.
Li et al. [37]	DHG-14/28& SHREC'17	RNN	proposed hand gesture graph convolutional networks (HG-GCN) model, and focused on learning of more semantic data.
Adam et al. [66]	SHREC'17, DHG-14/28,&	MMEGRN	Proposed a new deep learning approach multi-model ensemble gesture recognition network (MMEGRN) to overcome the problem low recognition rate due to the noisy and complex skeleton sequences.
Peng et al. [67]	FPHA & SHREC'17	ResGCN	Authors designed an efficient and lightweight graph convolutional network, named ResGCNeXt that overcome the problem of high parameter and high computation cost. ResGCNeXt learns rich features from skeleton information and achieves high accuracy with less number of model parameters
Okan et al. [3]	EgoGesture & NV Gesture	3DCNN	Author's proposed a novel two-model hierarchical architecture for online dynamic hand gesture recognition systems. The proposed architecture is designed considering resource efficiency, early detections and single time activations, which are critical for online gesture recognition applications
Gibran et al. [43]	NVIDIA & IPN	3DCNN	3DCNN is used to classify the dynamic hand gesture in real-world scenario. They generated a new new benchmark dataset for continuous HGR that includes real-world issues.
Qing et al. [44]	DHG-14/28 and &SHREC'17	3DCNN + ConvLSTM	Proposed 2D hand pose estimation using OpenPose and 3DCNN + ConvLSTM is used to classify the dynamic hand gesture. Features of RGB, depth and 3D skeleton data is fused for final prediction.
Noorkholis et al. [54]	Self-defined gestures dataset	3DCNN + LSTM	Create a 24 dynamic hand gesture to control the real-time smart TV environment using deep learning model.
Chen et al. [25]	Ego gesture	ConvLSTM	applied an attention mechanism to extract features from a collection of hand gestures. They proposed RPCNet, which is made up of R2plus1D and ConvLSTM which are combined in parallel.
Zhang et al. [68]	DHG-14/28& SHREC2017	Graph Convolutional Neural Network	Proposed two-stream graph attention convolutional network with spatial–temporal attention for hand gesture recognition. In one stream hand pose is processed and in another one skeleton graph of current and previous frame is passed.

We have reviewed various research papers on different methods for hand gesture recognition. These include papers focusing on color detection, appearance-based methods, motion-based methods, skeleton-based approaches, depth-based techniques, 3D detection methods, and deep learning-based approaches. The details of few papers are summarized in Table 2.1. The pie chart in Figure 2-1 illustrates the progress of

research in the field of dynamic hand gesture recognition.

The pie chart shows that most research in dynamic hand gesture recognition focuses on deep learning methods (42.11%) because they are very effective at understanding complex spatial and temporal patterns. Skeletal-based approaches (14.91%) use joint locations to identify gestures, whereas depth-based approaches (14.04%) use depth sensor data to increase accuracy. These methods have gained popularity because they effectively address challenges and produce better results.

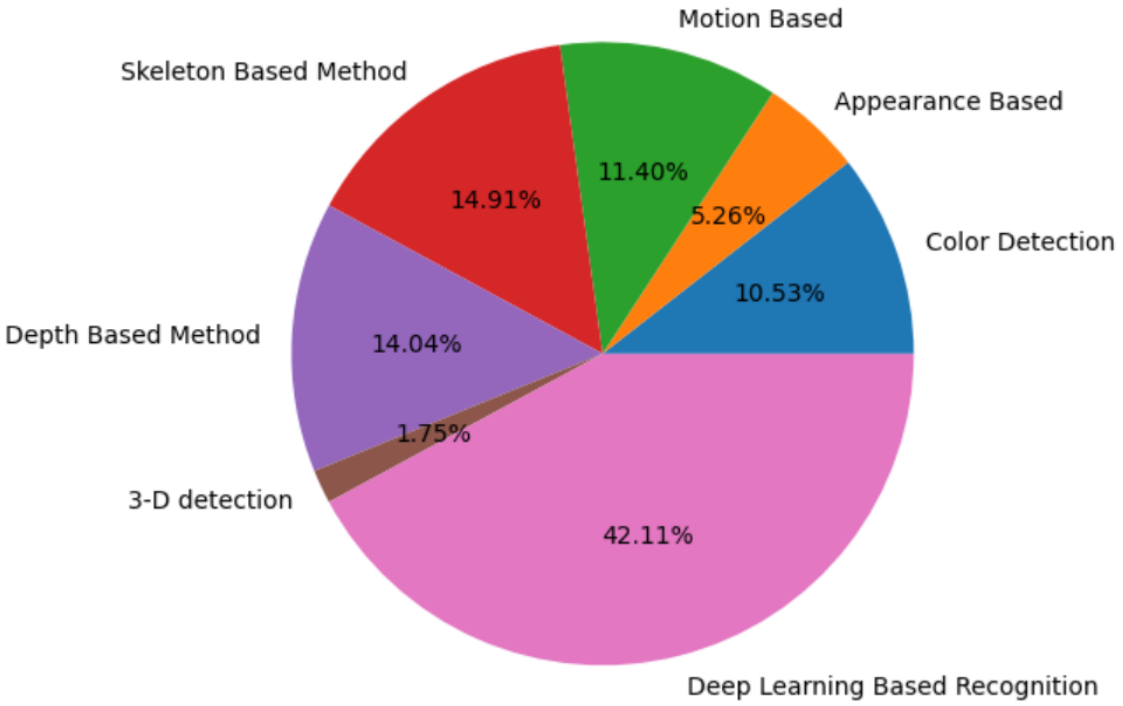


Figure 2-1: Pie chart depicting the development of research in the area of dynamic hand gesture recognition

2.5 In Thesis Prospective:

This thesis proposed four frameworks for dynamic hand gesture recognition. We address the difficulties of hand tracking in the first framework by using RGB videos for hand gesture recognition. It is very challenging to detect the hand for tracking because of its complex structure and smaller size as compared to the whole body. The CLIP-LSTM model is designed to overcome the challenges that arise due to the small size of

the hand and varying lighting conditions. A proposed model uses CLIP(Contrastive Language-Image Pre Training) model to extract features from RGB data. We use 2D skeleton trajectories and skeletal data in the second framework. Which deals with inter-class variation and intra-class variation both complicate accurate gesture recognition. We discover that both the skeletal and RGB data are useful and have benefits over one another. Therefore, in order to increase recognition accuracy, we combine the skeletal data with optical flow data in our third framework. We use skeleton data because the skeleton data does not contain background information. Therefore, skeleton-based models will not be affected by complex background information. Additionally, they are unaffected by occlusion and changes in illumination. Similarly, the creation of an optical flow video contains the movement of the hand irrespective of any background information. Thus, it filters out irrelevant data and concentrates on the gesturing hand that helps in extracting temporal features. In our fourth framework and each modality has its advantages. The first modality utilizes RGB data. It gives spatial information that helps interpret the gesturing hand's shape, texture, and color information. The second modality employs depth data, which records gesture motion. The third modality incorporates skeleton data. The challenges of complex backgrounds and occlusion are resolved by using skeletal data. In our pipeline, features are extracted individually from each modality using the CLIP model and sequential learning model followed by the processing of each modality through two Conv1D layers and an LSTM layer. This approach is designed to achieve high-speed performance and can operate effectively with a smaller number of parameters and also address the challenges of dynamic hand gesture recognition.

Chapter 3

Develop an Efficient and Generic Framework using RGB Videos for Hand Gesture Recognition

Reena Tripathi, and Bindu Verma. "CLIP-LSTM: Fused Model for Dynamic Hand Gesture Recognition." presented in *In 2023 IEEE 20th India Council International Conference (INDICON)*, pp. 926-931. IEEE, 2023. *(Published)*

3.1 Introduction

In this chapter we present our work on dynamic hand gesture recognition using CLIP as a feature extractor and BLSTM for sequence-to-sequence learning. We use hand gestures in a variety of contexts every day, including automatic vehicles, intelligent homes, controlling and monitoring home appliances, human-robot interaction, 3D modeling, 3D space sensors, sign language, and, artificial intelligence. Dynamic hand gesture recognition includes analyzing the hand motion and fingers, as well as the overall shape and position of the hand, to identify specific gestures. In a dynamic hand gesture, spatio-temporal information is required to recognize any particular gesture. Automated hand gesture is a challenging due to the smaller size of the

hand compare to the whole body. Furthermore, detection and segmentation of the hand from the background, detecting the shape of the hand is challenging. Further, robustly tracking the hand across frames and extracting important features so that the gesture is well-defined is also challenging. Various illumination conditions also make hand detection and tracking difficult. Thus, in the work we have used CLIP model to extract the features that overcome the challenge of hand detection and tracking.

In this chapter, we proposed the CLIP-BLSTM model, which uses a CLIP model to extract features from RGB data. The obtained features are fed into the BLSTM model for sequence-to-sequence learning and dynamic hand gesture is recognized. The novelty of our work consists of the use of a CLIP model, through which RGB video data is passed to extract the features. We test the proposed model on two datasets CHG(Cambridge hand gesture dataset) and LISA. We compare our proposed model with state-of-the-art methods and demonstrate that our proposed CLIP-BLSTM model outperforms existing approaches, achieving an accuracy of 86.0%.

3.2 Literature Survey

In deep learning-based hand gesture recognition, the hand gesture recognition system has only two stages that are data pre-processing and hand gesture classification. The pre-processing stage consists of image enhancement, noise removal, and image resizing. The authors Yimin et al. [69] presented a novel technique for feature extraction. The researchers use a weighted radial projection algorithm to detect each finger of the hand. Researchers use two methods for gesture recognition in their studies a) edge feature-based matching and b) gesture silhouette-based matching and angular projection were used for obtaining wrist angles, finger length, and orientation. Lalit et al. [19] used depth matrix and adaptive Bayes classifiers to recognize the dynamic hand gestures. The researchers use a depth matrix and 1-nearest neighbor for recognition purposes. In this paper, a Naïve Bayes classifier is used by the user and the gesture is operated via two methods, a) state-level and b) sequence level. A spatio-

temporal attention-based 3D-CNN was suggested by the authors Huang et al. [70] to gather high-quality information for categorizing multi model dynamic gestures. According to the author, the multi model had certain drawbacks, namely that the data acquisition settings were not ideal. The Grassmann graph is used by the author Verma et al. [13] to classify the gesture. The distance between classes and the test gesture is calculated using k-nearest neighbor and embedding discriminant analysis the gesture is then allocated to the class with which it has the smallest distance. Rubin et al. [71] proposed Hybrid Single Stage Recognition (hybrid-SSR) based on CNN for hand gesture recognition and enhanced Xception CNN model for feature extraction. Ameur et al. [62] Proposed Hybrid Bidirectional Unidirectional LSTM (HBU-LSTM) that handles sequential data generated by leap motion controller device. Due to the availability of the large size of the dataset, and high-capacity hardware deep learning emerged rapidly and gives excellent results. However, feature extraction plays a vital role in recognizing hand gestures.

3.3 Proposed Architecture

The proposed architecture is depicted in Figure 3-1. In the proposed model, the features extracted from CLIP are fed to the pipeline of a neural network that consists of two Conv1D layers, a Bidirectional LSTM layer, and LSTM layer. The function applies two Conv1D layers with the same number of filters, "row hidden," and a kernel size of 3×3. When the padding parameter is set to "same," the layers' output shape will have the same dimensions as their input. The output of the Bidirectional LSTM layer is then fed to another LSTM layer. This LSTM produces a single vector. The model recognizes the dynamic hand gesture by adding a dense layer with a predetermined number of output classes and a softmax activation function.

3.3.1 Contrastive Language-Image Pre-training(CLIP)

A modern machine-learning model called CLIP has attracted a lot of attention in the AI field because of its extraordinary capacity to comprehend the relationship between

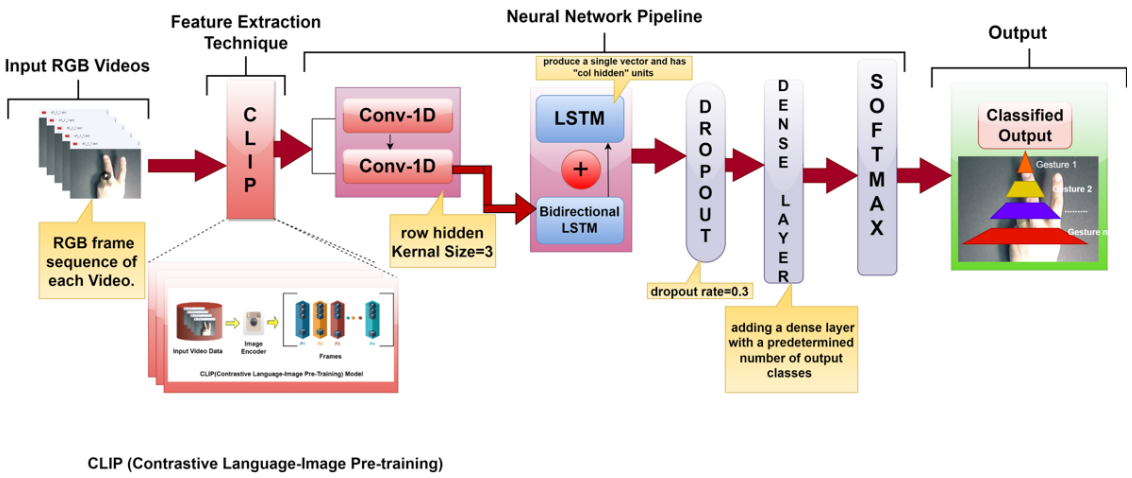


Figure 3-1: Proposed model(CLIP-BLSTM) architecture: Features extracted from CLIP are processed through a neural network comprising two Conv1D layers, a Bidirectional LSTM layer, and an LSTM layer. The Conv1D layers, configured with identical filters and kernel size (3×3) and "same" padding, preserve input dimensions. The Bidirectional LSTM output is passed to the LSTM layer, which generates a single vector. Finally, a dense layer with a softmax activation function classifies dynamic hand gestures into predefined classes.

images. We used CLIP image encoder to extract the features from each frame of a video. The CLIP model is shown in Figure 3-2.

CLIP feature extractor extracts the feature of each video and stored it into a 1-D vector of dimension 512. Let T_f represent the feature vector of 1 video as shown in Equation 3.1. If there are 'n' number of videos in one class, the size of the feature vector matrix for one class will be in Equation 3.2. Like-wise next class feature vector matrix will be appended. In last for a dataset having 'C' classes the feature vector matrix will be $(C \times n) \times T_f$. The total number of features for one video will be represented by T_f.

$$T_f = f_0, f_1, f_2, f_3.....f_{511} \tag{3.1}$$

$$EF = P_i \times T_f \qquad \forall i = 1, 2, 3...m \tag{3.2}$$

Where, the features matrix for one class will be represented by 'EF'. P_i represented gestures of class C_m.

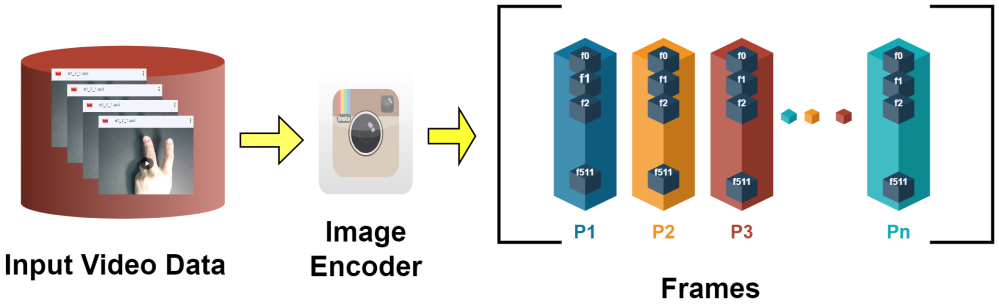


Figure 3-2: CLIP (Contrastive Language-Image Pre-training) Model

$$Output_{(C\times n)\times T_f} = \begin{bmatrix} C_1P_1\{f_0 & f_1 & \cdots & f_{511}\} \\ C_1P_2\{f_0 & f_1 & \cdots & f_{511}\} \\ \vdots & \vdots & \ddots & \vdots \\ C_1P_n\{f_0 & f_1 & \cdots & f_{511}\} \\ C_2P_1\{f_0 & f_1 & \cdots & f_{511}\} \\ C_2P_2\{f_0 & f_1 & \cdots & f_{511}\} \\ \vdots & \vdots & \ddots & \vdots \\ C_mP_n\{f_0 & f_1 & \cdots & f_{511}\} \end{bmatrix} \tag{3.3}$$

The Feature matrix for one complete dataset is represented by the $Output_{(C\times n)\times T_f}$. Where, C_1 represents class 1, P_1 represents gesture 1 of class C_1, and $\{f_0, f_1, \ldots, f_{511}\}$ is the feature matrix of video 1 of class 1, and so on.

3.3.2 Convolutional Network

The 1D-CNN stands for the 1-D convolutional neural network is made up of 3 layers. The 3 layers, are the convolutional layer, the pooling layer, and the fully connected layer. The fully connected layer has a number of neurons equivalent to the number of output classes, while the input layer receives the 1D feature signal.

3.3.3 Bi-directional LSTM

Recurrent neural networks struggle with long-term reliance because of the vanishing gradient problem in LSTM networks. Instead of analyzing each data point separately, they can analyze entire data sequences and store pertinent information from prior data in series to assist in processing new data points. Therefore, it is very adept at processing sequential data.

The input gate, output gate, forget gate, and cell gate of an LSTM unit are responsible for controlling the learning process as shown in Figure 3-3. To govern the functioning of the gates throughout the learning process, sigmoid functions are essential. The Cell state refers to the long-term memory in the LSTM. It regulates the data that will be saved in an LSTM cell from the previous stage. The cell gate is modified by the remembering vector, which is known as the forget gate. The forget gate and output state instruct the cell gate whether to maintain the information in the cell state if it is 1 or to forget if it is 0 [72]. Use of LSTM has the main benefit of resolving the vanishing gradient issue. The following equations illustrate the working of LSTM in our model [72].

$$i_t = \sigma(A_t w_{xi} + H_{t-1} w_{Hi} + b_{t-1} w_{bi} + w_{ibias}) \tag{3.4}$$

Where, "i_t" represents the input gate at time step "t". 'A_t" is the input vector and "w_{xi}" is its weight matrix. " H_{t-1}" is the hidden state from the previous time step with "w_{Hi}". "b_{t-1}" and "w_{bi}" are the bias term and its weight matrix, respectively. "w_{ibias}" is an additional bias term for the input gate.

$$f_t = \sigma(A_t w_{xf} + h_{t-1} w_{Hf} + b_{t-1} w_{bf} + w_{fbias}) \tag{3.5}$$

Where, "f_t" represents the output gate. "h_{t-1}", another notation for the hidden state from the previous time step. "w_{Hf}" is the weight matrix for the hidden state to the forget gate, and "w_{bf}" is the weight matrix for the bias to the forget gate.

$$C_t = tanH(A_t w_{AC} + h_{t-1} w_{HC} + w_{zbias}) \tag{3.6}$$

Where, "C_t" represents the candidate cell state. "w_{HC}" is the weight matrix for the hidden state to the cell state and "w_{zbais}" is the bias term. In Equation 3.7 the "b_t" represents the cell state.

$$b_t = C_t \otimes i_t + b_{t-1} \otimes i_t \tag{3.7}$$

$$o_t = \sigma(A_t w_{xo} + H_{t-1} w_{Ho} + b_{t-1} w_{bo} + w_{obais}) \tag{3.8}$$

Where, "o_t" represents Output gate. Controls the output from the cell state.

$$H_t = o_t + tanH(b_t) \tag{3.9}$$

Where,"H_t" represents the hidden state.

Equations 3.7, 3.8, and 3.9 are the standard formulas for output, forget gates, and hidden state. The "b_t", "H_t" represents output memory activation function at time interval t.

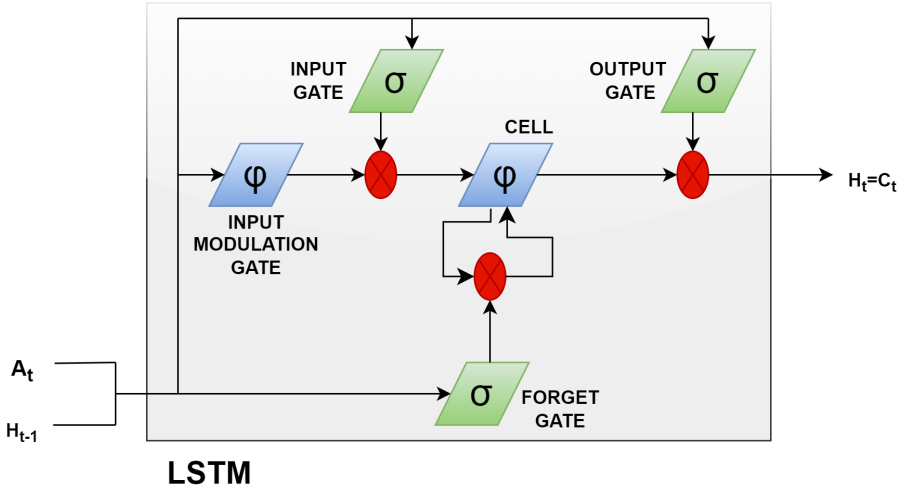


Figure 3-3: LSTM model.

Bidirectional LSTM is frequently used for sequential data processing applications like voice and natural language processing. The primary characteristic of Bidirectional LSTM is that it uses two different LSTM layers that are used to process

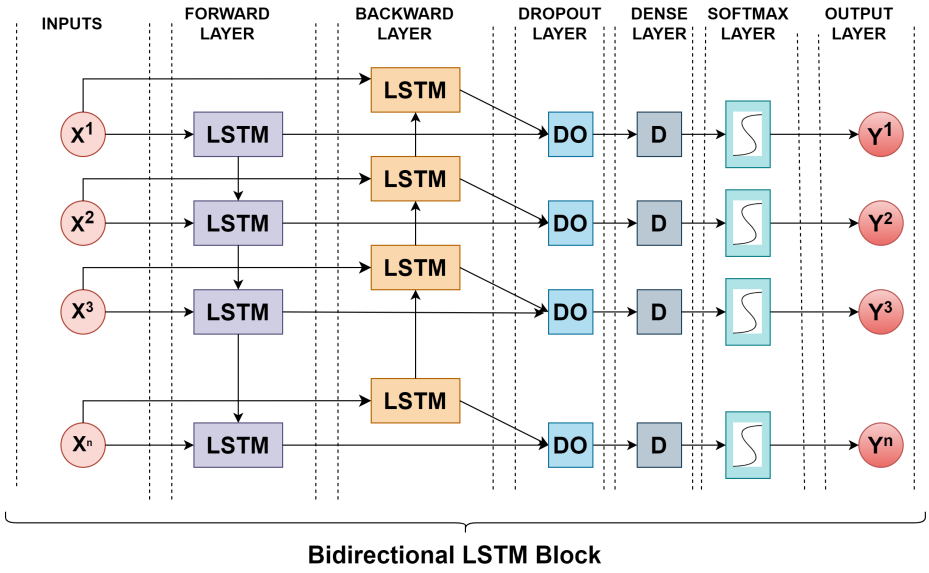


Figure 3-4: Bidirectional LSTM block.

both the forward and backward directions of the input sequence as depicted in Figure 3-4. Concatenating the output of each layer results in the output feature vector, which retains both the past and future context of input data. Bidirectional LSTM, in contrast to LSTM, can comprehend movements captured before and after the present point as it can utilize forward information and backward information both. Because of the flow of information in both directions, the BLSTM Long-term dependencies between signal patterns are captured. As compared to unidirectional networks, bidirectional LSTM is much superior [73]. In our proposed model the output of the second Conv1D is fed into this network.

3.3.4 Hand gesture classification

The extracted features using CLIP are fed into the proposed model. Then we combine the outputs of two layers and add a dropout layer with a 0.3 dropout rate to prevent overfitting. After that, the concatenated LSTM layer's output is mapped using the dense layer, and using the SoftMax layer class-wise probability is calculated below we have discussed all the layers of the model.

The algorithm of the proposed model is shown in Agorithm 3.1.

Algorithm 3.1 The proposed model's algorithm

Input: RGB data
Feature Extraction:
    Let the total number of features represented by $T_f = \{f_0, f_1, f_2, f_3, \ldots, f_{511}\}$
    The feature size EF is defined as: $EF = P_i \times T_f$
    Output matrix:

$$Output_{(C\times n)\times T_f} = \begin{bmatrix} C_1P_1\{f_0, f_1, \ldots, f_{511}\} \\ C_1P_2\{f_0, f_1, \ldots, f_{511}\} \\ \vdots \\ C_1P_n\{f_0, f_1, \ldots, f_{511}\} \\ C_2P_1\{f_0, f_1, \ldots, f_{511}\} \\ C_2P_2\{f_0, f_1, \ldots, f_{511}\} \\ \vdots \\ C_mP_n\{f_0, f_1, \ldots, f_{511}\} \end{bmatrix}$$

Pipeline:
    Step 1: Define Input Layers
        $input\_layer\_rgb \leftarrow Input(shape = input\_shape\_rgb)$
    Step 2: Process Modality
function PROCESS_MODALITY(in_layer)
        $x \leftarrow Conv1D(kernel\_size = 3, padding =' same')(in\_layer)$
        $x \leftarrow Conv1D(kernel\_size = 3, padding =' same')(x)$
        return LSTM(col_hidden)(x)
end function
    $processed\_rgb \leftarrow process\_modality(input\_layer\_rgb)$
    Step 3: Concatenation and Output Layer
        concatenate([first_read, trans_read])
    $concatenated\_layers \leftarrow Dropout(0.5)(concatenated\_layers)$
    $output\_layer \leftarrow Dense(num\_classes, activation =' softmax')(concatenated\_layers)$
    Step 4: Compile the Model
    Step 5: Train the Model
    Step 6: Evaluate the Model
    Step 7: Plot Results

Dropout Layer

Dropout is a regularization method used to stop deep learning models from overfitting. It is frequently employed in LSTM networks to enhance the model's performance. To reduce overfitting, we use dropout 0.3 on each LSTM layer's output. Dropout makes LSTM units more durable and helps avoid overfitting.

Dense Layer

The dense layer, which has the ability to conduct nonlinear transformations on the LSTM layer's output, is frequently used after the LSTM layer in neural network models. Each hidden state in the LSTM layer's output represents the memory at a particular step. These hidden states can be mapped to output using the dense layer. In the proposed model, the concatenated LSTM layer's output is mapped using the dense layer to the final output of given classes, using softmax activation. In order to avoid overfitting, the output of the concatenated LSTM layer is first passed through a Dropout layer. Next, the Dense layer is applied to create the final output.

Softmax Function

In our proposed model, we employ LSTM for classification and a layer that is dense and has a softmax activation function which generates the output probabilities for each class. A typical loss function, especially for multi-class classification tasks, is categorical cross-entropy. It calculates the difference between the actual labels and the predicted probability distribution. Each class predicted probabilities, are calculated using a Softmax activation function. In order to ensure that the projected values are both non-negative and add up to 1, Softmax creates a probability distribution over the classes.

$$Y_L = -\sum_{i=1}^{C}(x_{\text{true}}(j) \cdot \log(x_{\text{pred}}(j))), \qquad \text{for C classes} \tag{3.10}$$

In Equation 3.10 the summation is applied to all classes C and Y_L is represented as a cross-entropy loss. The true label is represented by the $x_{\text{true}}(j)$, while the predicted probabilities are represented by $x_{\text{pred}}(j)$ for j^{th} class. Elements of the expected prob-

ability are applied individually to the logarithm. It calculates the difference between actual class labels and expected probability. By adding the element-wise product of the true labels and the logarithm of the predicted probabilities for each class, the loss is determined. To reduce the loss during training, the negative sign is applied. The calculated probability distribution tends to resemble the real distribution when the cross entropy loss is kept to a minimum, which increases the classification accuracy of the model.

3.4 Experimental Analysis

All the experiments were conducted on intel Core i7 processor with 8GB RAM, and 8GB NVIDIA GETFORCE GTX graphics card. Implementation is done in Tensorflow 2.8 with PyTorch libraries. Adam is applied as an optimization function and using the SoftMax layer class-wise probability is calculated. The proposed model uses pre-trained CLIP to extract the features. Following that, the 1DCNN and BLSTM network receive the extracted features and perform sequence-to-sequence learning. The entire input sample is split into 90% for training and 10% for testing. For 200 epochs, we trained the proposed model using a batch size of 8 and a learning rate of 0.001.

3.4.1 Experiment on Various Hand Gesture Datasets

Cambridge Hand Gesture Dataset (CHG) [74]

CHG dataset performed by five persons with three different hand shapes and each gesture performed 20 times.The CHG dataset includes 900 image sequences of 9 hand gesture classes, each of which is represented by 3 basic hand shapes with 3 motions. 100 image sequences (5 various illuminations 10 arbitrary motions of 2 persons) are provided in each class. The gesture inventory contains 9 classes i.e. ‘flat-contract’,‘ flat right’, ‘flat-left’, ‘spread-contract’ ‘spread-left’, ‘spread-right’, ‘V-shape-contract’, ‘V-shape-left’, ‘V-shape-right’. The Cambridge Hand Gesture dataset is shown in

Figure 3-5. The Cambridge hand gesture dataset RGB videos are passed into the

Figure 3-5: Shows a few samples from the Cambridge Hand Gesture (CHG) dataset, illustrating various hand gesture categories. Each row represents a distinct type of gesture, highlighting the dynamic movements of the hand, which are useful for evaluating gesture recognition systems. The red arrows indicate the direction of motion, where applicable.

CLIP model for feature extraction. The features are further divided into 90% training sets and 10% testing sets, and fed into the proposed model. The classwise accuracy of a proposed model on CHG dataset is shown in Figure 3-6, the model achieved 97% average accuracy.

Figure 3-7 displays a confusion matrix on the CHG dataset. The confusion matrix shows that the following: classes 'spread-right' and 'V-shape-left' are mostly misclassified with 'spread-left and 'V-shape-right' due to the same motion movement and same hand shape. The performance of our system for a particular set of test data, we calculate the precision(P), recall(R), and F1-score(f1) values provided in Table 3.1. Our experiments show that the F1-Score is greater than 94% for all classes, and the macro average accuracy of our proposed model is 97%, proving that our proposed model achieves a high level of balance and performs remarkably well across all classes. We calculate the precision(P), recall(R), and F1-score(f1) values represented

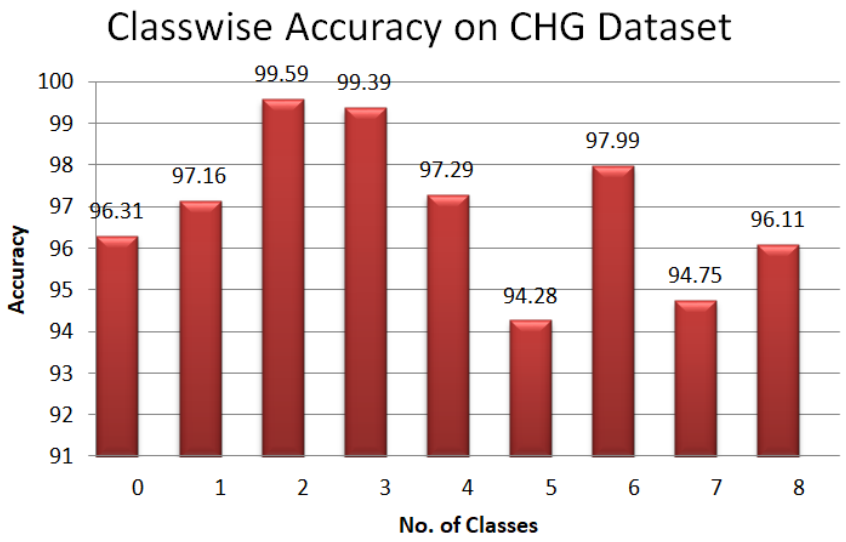


Figure 3-6: For the Cambridge Hand Gesture (CHG) dataset, the class accuracy is greater than 94% for all classes, and the average accuracy of our proposed model is 97%.

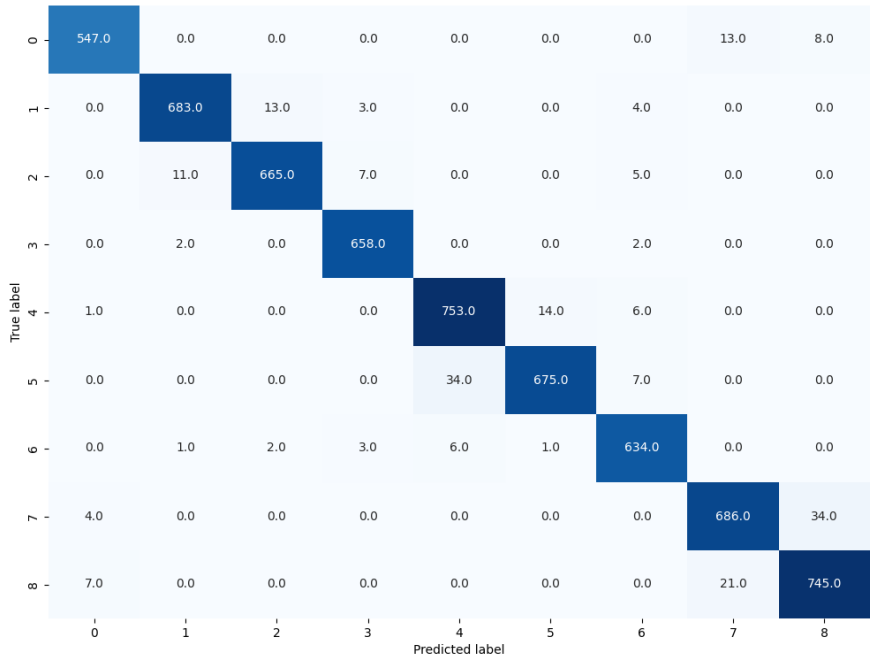


Figure 3-7: Shows a confusion matrix on the CHG dataset. The confusion matrix shows that the following: classes(5) spread-left' and class (7) V-shape-left' are mostly misclassified with class(6) spread-right' and class(8) V-shape-right' due to the same motion movement and same hand shape.

Table 3.1: All performed classes Precision, Recall, and F1-Score on CHG dataset.

Class No.	Class	P	R	f1
0	'flat-contract'	0.97	0.98	0.97
1	'flat right'	0.99	0.96	0.97
2	'flat left'	0.96	0.98	0.97
3	'spread-contract'	0.99	0.99	0.99
4	'spread-left'	0.98	0.91	0.95
5	'spread- right'	0.96	0.98	0.95
6	'V-shape-contract'	0.98	0.99	0.98
7	'V-shape-left'	0.97	0.94	0.94
8	'V-shape-right'	0.97	0.97	0.95

in Equations 3.11, 3.12, and 3.13.

$$\text{Precision} = \frac{tp}{tp + fp} \tag{3.11}$$

where tp stands for "true positives," and fp for "false positives,"

$$\text{Recall} = \frac{tp}{tp + fn} \tag{3.12}$$

the number of false negatives is indicated by fn. The F1-Score is the harmonic mean of recall and precision, with the most effective value being 1 and the poorest being 0. Equation 3.13 is utilized to determine the F1-Score.

$$F1\text{-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.13}$$

Experiments on LISA dataset [14]

LISA dataset contains 32 gesture sequences with three different hand shapes single-finger, two-finger, and three-finger gestures. These gestures are 'swipe left', 'swipe right', 'scroll up', 'scroll down', 'single tap', 'double tap', 'zoom in', 'zoom out', 'swipe up shape change', 'swipe down shape change', 'expand', and 'pinch' 'clockwise rotation', 'anticlockwise rotation', 'reverse Z', 'rotate', 'scroll left', and 'scroll right'. The LISA dataset is shown in Figure 3-8.

First, we conducted an experiment using 32 different gestures and obtained 81%

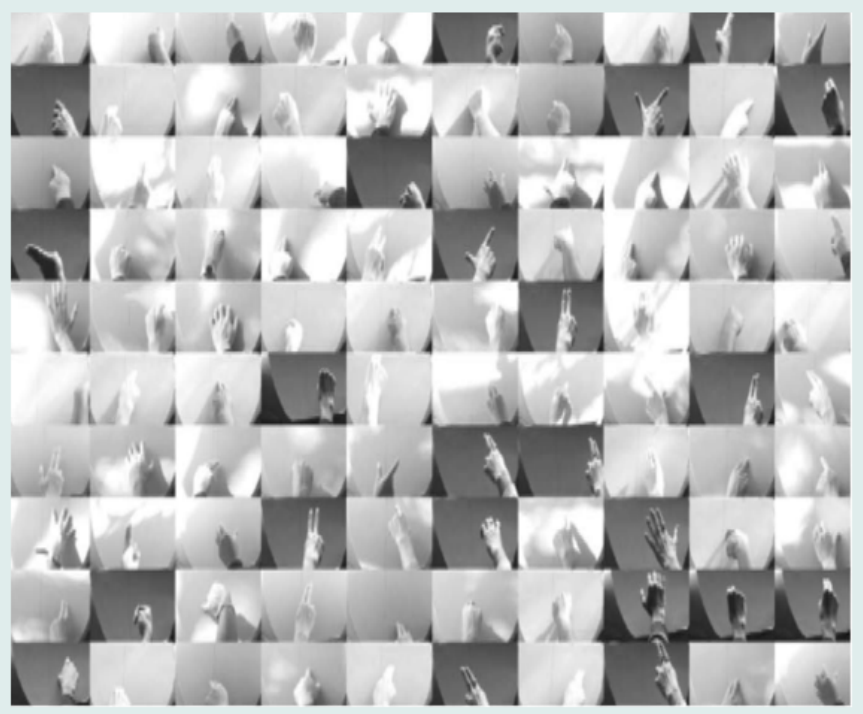


Figure 3-8: LISA Dataset

accuracy, since some movements such as scrolling left and shifting left, have similar movement patterns. Similar confusion exists between the gestures scroll plus and scroll right, shift right and scroll X, and one tap and two taps. The accuracy increased to 86% after we combined a similar kind of gesture. The features extracted from the CLIP are further divided into 90% training sets and 10% testing sets and fed into LSTM for feature learning. The class-wise accuracy is shown in Figure 3-9.

Figure 3-10 displays a confusion matrix. From the confusion matrix, we can see that class 5 performs a zigzag gesture using 2 fingers, class 7 performs a swipe X gesture using two fingers, and class 21 is a merged class of three gestures like single tap using one finger, single tap using two fingers, and double tap using 3 fingers have less than 75% accuracy because of similar tracks and Similar confusion exists between the gestures scroll right and shift right, scroll plus and scroll X, and one tap and two taps.

The recall(R), precision(P), and F1-Score(f1) measures of the LISA dataset, are provided in Table 3.2. Our experiments show that the F1-Score is greater than 76%

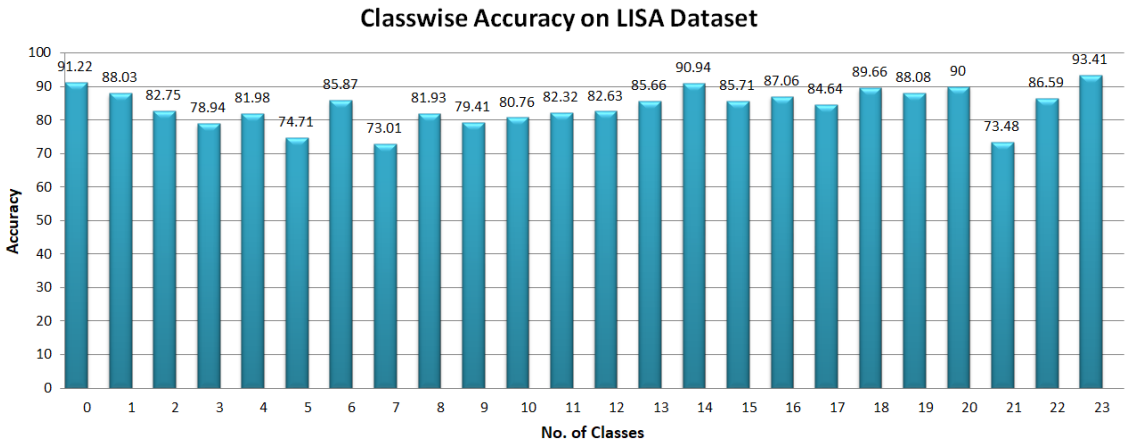


Figure 3-9: For the LISA dataset, the class accuracy is greater than 73% for all classes, and the average accuracy of our proposed model is 86%.

for all classes, and the macro average accuracy of our proposed model is 86%, proving that our proposed model achieves a high level of balance and performs remarkably well across all classes of challenging datasets.

Comparison with State-of-the-Art Methods

Comparison of CHG Dataset with State-of-the-Art Methods

We compare the experimental results with state-of-the-art on the CHG dataset. Table 3.3 presents a comparison between the proposed model and existing approaches. In this paper [75], the author presents a key frame extraction method for gesture videos using high-level feature representation. They utilize a multi-channel gradient magnitude frequency histogram (HGMF-MC) descriptor based on VGG16. The author [13] uses the Grassmann graph for gesture classification, employing k-nearest neighbors and embedding discriminant analysis to assign gestures to the nearest class. Similarly, the author [74] introduces tensor canonical correlation analysis (TCCA) to capture joint space-time relationships in video data for action classification, combined with feature selection and a nearest neighbor classifier. The author [20] focuses on improving the speed of hand gesture recognition by using image details and clustering to pick keyframes in videos, thus speeding up data processing. The author [76] enhances dense trajectory features by introducing a new hand segmentation tech-

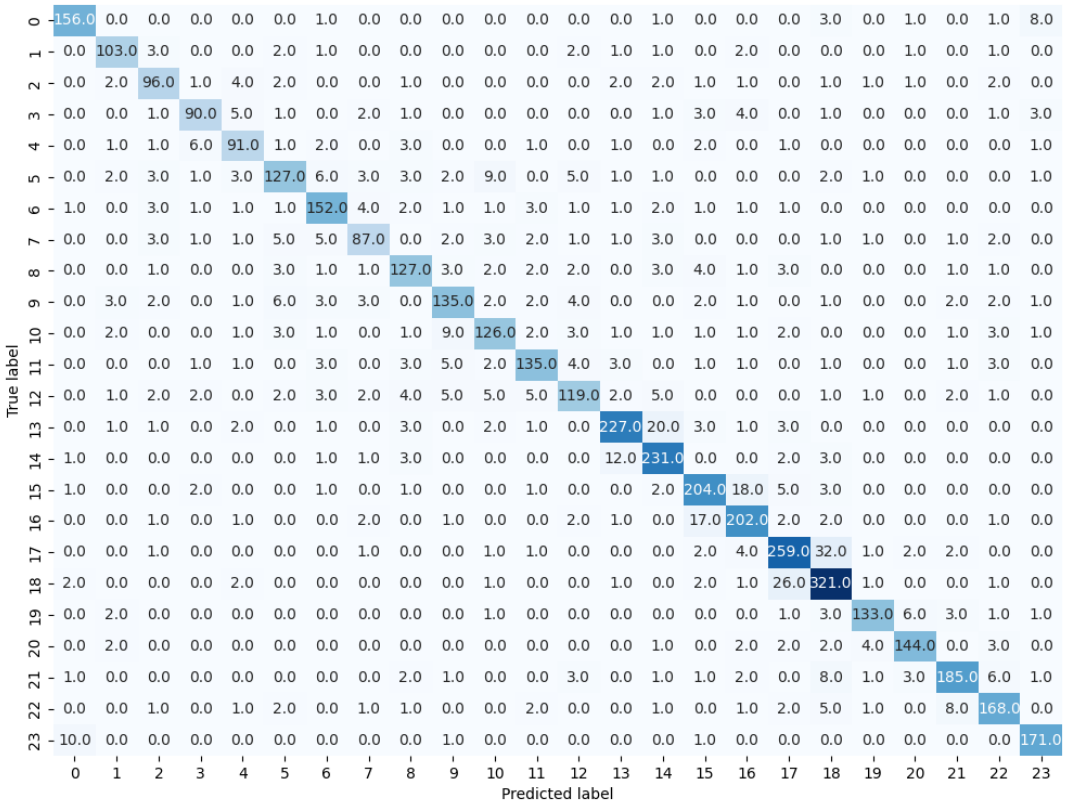


Figure 3-10: Confusion Matrix on LISA dataset. The matrix highlights that only gestures like class 5 (zigzag gesture with two fingers), class 7 (swipe X gesture with two fingers), and class 21 (a merged class of single and double taps) achieve less than 75% accuracy due to similar motion tracks. Confusions occur between gestures like scroll right and shift right, scroll plus and scroll X, and one tap versus two taps. The rest of the classes achieved more than 75% accuracy. The overall accuracy of the proposed model is 86%.

Table 3.2: All performed classes Precision, Recall, and F1-Score on LISA dataset.

Class No.	Class	P	R	f1
0	Pinch	0.90	0.93	0.92
1	Swipe right	0.90	0.92	0.91
2	Swipe left	0.92	0.83	0.87
3	Swipe down	0.85	0.84	0.84
4	Swipe up	0.87	0.83	0.85
5	Zig Zag	0.80	0.77	0.79
6	Swipe V	0.82	0.80	0.81
7	Swipe X	0.79	0.73	0.76
8	Swipe +	0.79	0.73	0.76
9	Clockwise rotation	0.86	0.80	0.83
10	counter clockwise	0.83	0.79	0.81
11	Reverse z	0.85	0.81	0.83
12	Rotate	0.78	0.84	0.81
13	Scroll left Scroll left(3SCL)	0.79	0.80	0.79
14	Scroll right Scroll right(3SCR)	0.82	0.85	0.83
15	Scroll down Scroll down(3SCD)	0.80	0.85	0.82
16	Scroll up Scroll up(3SCU)	0.81	0.90	0.85
17	Single Tap(1ST) Single Tap(2ST) Double tap(3ST)	0.86	0.85	0.85
18	Double Tap(1DT) Double Tap(2DT) Double Tap(3DT)	0.82	0.86	0.84
19	Zoom in	0.88	0.87	0.87
20	Zoom out	0.90	0.92	0.91
21	Swipe up shape change	0.93	0.87	0.90
22	Swipe down shape change	0.91	0.87	0.89
23	Expand	0.90	0.84	0.86

nique that leverages superpixels and spatial-temporal coherence. The results, shown in Table 3.3, demonstrate that the proposed CLIP-BLSTM model outperforms other state-of-the-art methods on the CHG dataset, achieving 97.0% accuracy.

Table 3.3: Comparison of recognition accuracy on the CHG dataset with state-of-the-art methods.

Methods	Accuracy (%)
2D-DWT-PH [75]	90.0
3DHOG+GGDA [13]	89.7
TCCA+k-NN [74]	85.5
LBP-TOP [20]	60.8
Dense Trajectories + Hand Seg [76]	94.0
CLIP-BLSTM	**97.0**

Table 3.4: Comparison of classification accuracy on the LISA dataset with state-of-the-art methods.

Methods	Accuracy (%)
STG Embedding + Spline Modelling [77]	82.0
CNN with the HRN and LRN [78]	77.0
HOG3DVV + GGDA [13]	83.5
CNN-HMM Hybrid [79]	57.5
CLIP-BLSTM	**86.0**

Comparison of LISA Dataset with State-of-the-Art Methods

We compare our experimental results with state-of-the-art on the LISA dataset as shown in Table 3.4. In the paper [77], the author presents a method for recognizing actions in videos by creating graphs based on video data and matching them with new video clips. In papers [79] and [78], the authors focus on improving traditional dynamic hand gesture recognition methods by training HMMs with complex features like HOG and CNN, while also exploring techniques such as dimensionality reduction and data augmentation to prevent overfitting. The Grassmann graph is used by the author in [13] to classify gestures, where the distance between classes and the test gesture is calculated using k-nearest neighbor and embedding discriminant analysis. For the LISA dataset, we compare our proposed model with state-of-the-art methods and demonstrate that our proposed CLIP-BLSTM model outperforms existing

approaches, achieving an accuracy of 86.0% the result shows in Table 3.4.

3.5 Conclusion

To create a rapid and accurate hand gesture system, the chapter proposed a CLIP-BLSTM model that is computationally effective on fewer training samples and uses fewer parameters. Our experimental findings demonstrate that our model operates well under various illuminations and is comparable to SOTA techniques on challenging datasets from CHG and LISA. The experimental accuracy on the CHG dataset is 97% and on the LISA dataset is 86% superior performance to cutting-edge techniques while requiring fewer parameters.

The novelty of our work lies in utilizing the CLIP model to extract features from RGB video data. The CLIP-BLSTM model is specifically designed to address challenges associated with small hand sizes and hand tracking, proving to be efficient with fewer training samples and parameters. Overall, it performs effectively in different lighting environments, establishing it as an accurate hand gesture recognition system.

Chapter 4

Develop a Hand Gesture Recognition Framework that will Reduce the Inter and Intra-Class Variation

Reena Tripathi, and Bindu Verma. "Ensemble Learning with DDALoss for Inter and Intra Class Variation in Hand Gesture Recognition" is communicated in *Signal, Image and Video Processing* (SCIE Indexed, IF:2) *(Under Review)*

4.1 Introduction

In this chapter, we have presented an ensemble learning with a Discriminant Distribution-Agnostic(DDA) Loss. To address the problem of inter-class and intra-class variation, DDA loss is used that increases the with in class similarity features and decreases the between class similarity features. In this work skeleton data is used to create a skeleton point trajectory, which helps to overcome challenges such as hand occlusion, illumination variations, and complex backgrounds. However, the previous method of chapter 3 performed well on RGB data but it data may encounter challenges such as illumination variations, occlusion, and background clutter, which can hinder hand gesture recognition. In contrast, skeleton data overcome these challenges because

skeleton data is not dependent on background information and foreground information. Further, plotting of skeleton trajectory and processing with DDA loss solve the problem of inter and intra class variations. Inter-class variation is the variation in performing the gestures of different classes. Intra-class variation, on the other hand, describes variations within the same gesture class. This variation in gesture arises from the fact that the same gesture might be performed in various ways by several individuals or even by the same individual at different times. Thus, plotting of the gesture trajectory that shows these variations and feature extraction with various deep learning model then DDA loss feature learning solve the inter and intra-class variation problem. The DDA loss function encourages features to be close to their respective class centers while being distinct from other class centers. It improves classification accuracy by combining different models' strengths in extracting features and using a strong loss calculation method to ensure effective training.

The proposed model, utilizes three pipelines, each using skeleton data to create a skeleton point trajectory, addressing challenges like hand occlusion and illumination variations. For complex structure and to get the high discriminating features deeper deep learning model required. Features are extracted individually using VGG16, InceptionV3, and DenseNet121, and then ensembled. The DDA loss function combines center loss and DDA loss to compute the total loss, encouraging features to be close to class centers while distinct from others. DDA loss, increases the similarity within class (Intra-class) and decreases the similarity in different classes (Inter-class), and improves the performance of the hand gesture recognition system. This loss is used for back-propagation, updating model weights iteratively with the Adam optimizer. After training, the ensemble of models makes predictions by aggregating outputs, improving classification accuracy by leveraging the strengths of each model and robust loss calculation. This combination makes the model more robust to diverse hand gestures and varying conditions. While keeping a significantly lower computational cost, the proposed model outperformed with other state-of-the-art methods on benchmark datasets.

4.2 Literature Survey

Hand detection is challenging due to inter-class and intra-class variations in gesture performance, as well as moving backgrounds and illumination changes. Many authors have used skeleton data with the deep learning model to classify the dynamic hand gesture. The author Wenbin et al. [80] proposed a model for inter-class and intra-class constraints in Novel class discovery (NCD) using symmetric Kullback-Leibler divergence (SKLD). Similarly, other authors [81] work on inter and intra-class variation using central loss and triplet loss to reduce intra-class variation and increase inter-class variation. The author Verma et al. [10] uses skeleton and depth data information for fingertips and creates trajectories and Grassmann Graph Discriminant Analysis(GGDA) is applied for gesture recognition. Similarly, author Chen et al. [31] proposed motion feature-augmented RNN for dynamic hand gesture recognition using skeleton data. In this paper, finger and global motion features are extracted, to enhance the RNN performance. The author Smedt et al. [15] proposed a new skeleton-based approach for 3D hand gesture recognition, utilizing the geometric shape of the hand to extract descriptors from hand skeleton joints. Similarly, author's Tripathi et al. [41] used skeleton trajectories extracted from the RGB data and optical flow information for RGB video with GRU model to classify the dynamic hand gesture. To recognize hand gestures, the author Liu et al. [65]proposed a technique that divides hand gestures into two categories, hand movements and hand posture variations. It presents an end-to-end two-stream network that uses a 2D CNN for hand movement features and a 3D CNN for hand posture development to learn from these components. The author Caputo et al. [82] proposed a 3D gesture recognizer model based on trajectory matching of a single hand. In another paper [83]authors used 3D trajectory gestures, and few trajectory options were taken for comparison. The author Sheng et al. [67] proposed an effective Graph Convolutional Network (GCN) model for dynamic hand gesture recognition using skeleton data. Li et al. [84] proposed the MVHANet method for single-hand gesture recognition by finding a suitable distribution of angles in skeleton data. The author Deng et al. [85] proposed a multistream

network (MM-Net), utilizing skeleton data for action recognition.

4.3 Proposed Architecture

The architecture of the proposed model is shown in Figure 4-1. The model is organized into three pipelines, each utilizing a different model. In all pipelines, skeleton data is used to create a skeleton point trajectory, which helps in overcome the challenges such as hand occlusion, illumination variations, and complex backgrounds. After calculating the skeleton points trajectory, features are extracted individually via VGG16, InceptionV3, and DenseNet121. These models extract features from the trajectory images plotted using skeleton data of the performed gesture. These feature vectors from all models are then ensembled and passed through the DDA loss function, which combines center loss and DDA loss to compute the total loss. The DDA loss function encourages features to be close to their respective class centers while being distinct from other class centers. This combined loss is used for back propagation to update the model weights. Thus, DDA loss helps in to increase the with in class similarity and decrease the between class similarity. The models are trained iteratively using the Adam optimizer. After training, predictions are made by aggregating outputs from all models in the ensemble, and the final predicted labels are produced. The proposed model is an ensemble of three pre-trained neural network models (VGG16, InceptionV3, and DenseNet121) for a multi-class classification task, enhanced by a custom loss function, Discriminant Distribution-Agnostic Loss(DDA loss). The ensemble approach aims to leverage the strengths of each model for better performance. The ensemble of models, trained with this advanced loss function, works to improve classification accuracy by combining different models' strengths in extracting features and using a strong loss calculation method to ensure effective training [86].

4.3.1 Ensemble Learning

In ensemble learning, we combine multiple models to improve the overall performance and robustness of the system. Each model (VGG16, DenseNet121, and InceptionV3)

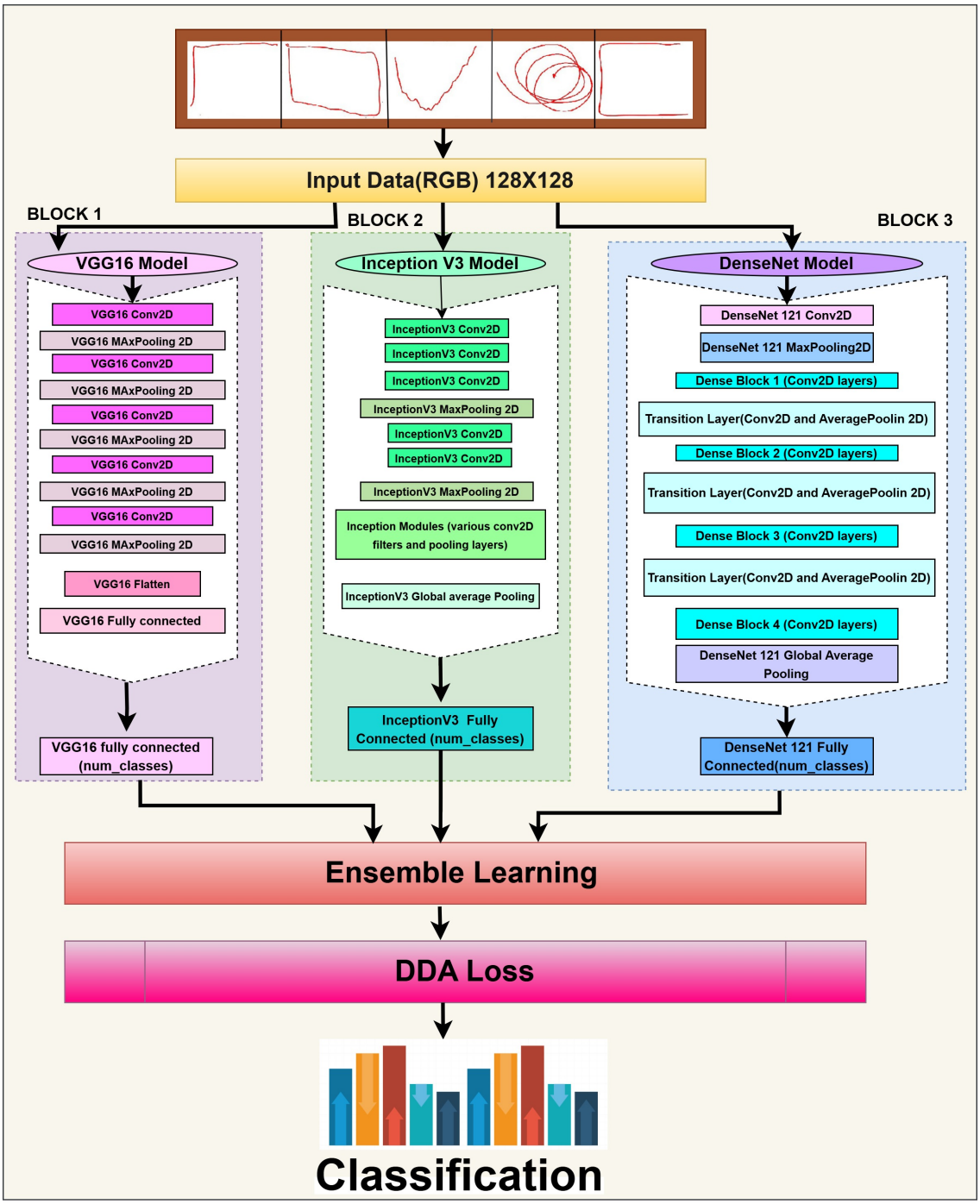


Figure 4-1: For hand gesture recognition, the proposed architecture comprises three deep learning architecture models: VGG16, InceptionV3, and DenseNet121. Through the use of their individual layers, each model separately processes the input image in order to extract features. After the features are collected, they are put together in an ensemble layer and then processed with the DDA loss function that controls class variations to increase recognition accuracy, predictions are made by aggregating outputs from all models in the ensemble.

will learn different features and patterns from the input data, and their predictions will be aggregated to produce the final prediction. This approach can help in reducing overfitting, especially when working with a small dataset, as the different models can generalize better together.

- VGG16 [75]: In our proposed model we load the VGG16 model with pre-trained weights for feature classification. The model's layers are frozen to prevent their weights from being updated during training. A flattened layer is added to convert the 2D feature maps to 1D feature vectors. A Dense layer with 256 units and ReLU activation is then included. The output layer is a Dense layer with the number of classes specified by num_classes and a softmax activation for classification. The architecture of VGG16 is shown in Figure 4-1BLOCK 1. VGG16 is a deep neural network with 16 layers, consisting of 13 convolutional layers to detect patterns such as edges, and textures. With 3x3 filters and 5 max-pooling layers, extracting features from images will help in preserving spatial information while reducing computational complexity. It ends with fully connected layers for classification, where the final layer produces probabilities for classes.

- InceptionV3 Model [87]: By using several convolutions of various sizes in parallel, InceptionV3 is able to capture multi-scale information. The pre-trained weights are used by the model to load InceptionV3. The feature maps are transformed into a single vector per picture by a GlobalAveragePooling2D layer, which is followed by a Dense layer that contains the softmax activation and number of classes for classification. The architecture of inception V3 is shown in Figure 4-1BLOCK 2. Inception V3 is a deep neural network that uses multiple parallel convolution layers (1x1, 3x3, 5x5) and pooling operations to efficiently capture features at different scales. It uses smaller, split-up convolutions to make the model faster and more efficient. It also adds extra classifiers during training to help the model learn better and prevent overfitting, making it very good at recognizing images.

- DenseNet Model [88]: DenseNet121 enhances the flow of information and gradients by connecting each layer to every other layer in a feed-forward fashion. Global Average Pooling2D is used to reduce the spatial dimensions of the feature maps before the fully connected layers. The architecture of DenseNet121 is shown in Figure 4-1BLOCK 3. DenseNet121 connects each layer to all previous layers, improving feature reuse and reducing the number of parameters. It consists of 4 dense blocks, each separated by transition layers that downsample the feature maps. The output of each layer is ensembled with the output from all previous layers in the same block, resulting in dense connectivity. This architecture enhances model efficiency and performance in deep learning tasks.

Ensemble the outputs of the three models, we can integrate the predictions and determine the final prediction for the class with the highest average probability. The detailed algorithm of the proposed model is outlined in Algorithm 4.1.

Algorithm 4.1 The proposed model's algorithm.

Input: 2D Skeleton trajectory plots data
Step 1: Split data into $X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}}$
Step 2: Create DataLoaders
    train_loader $\leftarrow$ DataLoader($X_{\text{train}}, y_{\text{train}}$)
    test_loader $\leftarrow$ DataLoader($X_{\text{test}}, y_{\text{test}}$)
Step 3: Feature Extraction - Each model separately extracted the features of an input images(VGG16, InceptionV3, DenseNet121)
Step 4: Define each Model individually
    VGG16Model, InceptionV3Model, DenseNet 121Model
Step 5: Ensemble Layer and Output Layer
    Ensembled Layer - Fuse the outputs from VGG16, InceptionV3, and DenseNet 121 fully connected layers
Step 6: Define DDA loss
    Initialize DDA loss with parameters
    Forward pass: Compute center loss and DDA loss
Step 7: Apply DDA loss
    Apply DDA Loss
Step 8: Compile the Model
Step 9: Train the Model
Step 10: Evaluate the Model
Step 11: Plot Results
Output the final classification result

4.3.2 Discriminant Distribution-Agnostic Loss (DDA loss)

DDA loss is a loss function used in deep learning model's specially when dealing with unbalanced data or noisy labels. DDA loss maximises the class separability thus, solve the problem of inter and intra-class variation. In DDA loss, the Euclidean distance is used to measure how close a feature vector is to its true class center and how far it is from other class centers. The loss minimizes this distance for the similar classes while maximizing for different classes, thereby improving intra-class compactness and inter-class separation. DDA loss is better at handling imbalanced data than traditional losses like cross-entropy. Training with both softmax loss and center loss helps create well-separated clusters of features. Softmax loss emphasizes the angular separation between features of different classes, but it doesn't work well when the dataset is challenging. Center loss focuses only on reducing the distance between features and their class centers, ignoring other classes. The center loss works very effectively on challenging datasets and imbalanced datasets, especially in real-world scenarios. It creates clearer boundaries between different gesture classes, which is helpful when gestures look similar. These features make DDA loss more accurate and reliable for recognizing dynamic hand gestures compared to traditional methods.

The Euclidean distance between deep feature vectors and class centers is used by the DDA loss to overcome both intra-class and inter-class variances. How DDA loss handles these variances is provided below:

Inter-Class and Intra-Class Variations

In dynamic hand gesture recognition, inter-class variations describes the distinctions between different classes of gestures. To avoid miss-classification, effective loss functions will ensure that features from various classes are well-separated. DDA loss [89] can be calculated using Equation 4.1.

$$L_{DDA} = -\frac{1}{2n}\sum_{j=1}^{n}\log\frac{e^{-\|p_j - c_{q_j}\|^2/2}}{\sum_{v=1}^{R_v} e^{-\|p_j - c_v\|^2/2}} \tag{4.1}$$

where:

- n is the number of samples in the batch.
- p_j is the deep feature of the i-th sample.
- c_{q_j} is the center of the class to which p_j belongs.
- R_v is the number of classes.
- c_v is the center of the v-th class.

The SoftMax function can be represented as:

$$p_C(p_j \in C_v \mid v) = \frac{e^{-\|p_j - c_v\|^2/2}}{\sum_{v=1}^{R_v} e^{-\|x_j - c_v\|^2/2}} \tag{4.2}$$

Where, In Equation 4.2. p_j is the deep feature of the j-th sample. c_v is the center of the v-th class and R_v is the number of classes. The network effectively separates features of distinct classes by learning to maximize this probability for the correct class by minimizing DDA loss.

On the other hand, In dynamic hand gesture recognition, intra-class variation describes the variations that take place inside the same class of gestures. These variations arise from the same gesture being performed differently by many people, or even by the same person at various times. Different hand shapes, sizes, orientations, speeds, and trajectories are among the factors that cause intra-class variations.

The Class Center Attraction(CCA) represents the distance between the data point and its class center. The loss function makes sure that features belonging to the same class are tightly packed around the center of that class by maximizing the log probability as shown in Equation 4.3.

$$CCA = e^{-\|x_i - c_{y_i}\|^2/2} \tag{4.3}$$

$$\text{Total Loss} = \lambda \cdot \text{Center Loss} + \gamma \cdot \text{DDA loss} \tag{4.4}$$

where “λ" is the weight for the center loss, and “γ" is a weight for the DDA loss. This total loss is computed during the forward pass of the DDA loss function and

used for backpropagation to update the model weights. Unlike center loss, which only considers the distance to the correct class center, DDA loss considers the distances to all class centers. DDA loss improves both inter-class separation and intra-class compactness by considering Euclidean distances to all class centers. Combines center loss and DDA loss using the weights as shown in the Equation 4.4, ensuring a balance between intra-class compactness and inter-class separation.

4.4 Experimental Analysis

4.4.1 Training Details

An Intel Core i7 processor with 8GB of RAM and an 8GB NVIDIA GeForce GTX graphics card was used to perform the experiments. TensorFlow 2.8 was used for the implementation, along with Keras libraries. Adam served as the optimization function, while the categorical cross-entropy is utilized as the loss function. The scores from the three deep learning architectures are ensembled at the ensemble learning layer and then passed to DDA loss layer to form the final prediction. Categorical cross-entropy and DDA loss are used in the proposed model, The DDA loss is used to find the final prediction. A model undergoes 150 epochs of training with 4 batch sizes and a 0.0001 learning rate. Our experimentation encompassed various batch sizes, loss functions, and optimizers, ultimately selecting a batch size of 4 and DDA loss as the loss function based on experimental results. We Perform experiments on the 26-Gesture Dataset [82], DHG14/28 [15], and a subset of NTU RGB+D3 [90] and NTU RGB+D 120 [91] benchmark dataset.

4.4.2 Experimental Evaluation Across Different Datasets

26-Gestures Dataset

There are 26 different gesture trajectory 3D points in the 26-Gestures dataset [1] [82]. A set of 3D skeletal points captured using a Leap Motion controller by 14 dis-

[1]https://github.com/davidespano/3cent-dataset

tinct people are included in the dataset. The gesture set consists of 3D symbols (‘caret’, ‘check’, ‘curly-bracket-right‘, ‘delete’, ‘pigtail’, ‘curly bracket-left’, ‘square-bracket-right’, ‘star’, ‘v’, ‘x’, ‘square-bracket-left’), semi-circular arcs (‘arc3Dright’, ‘arc3Dleft’), a ‘3D spiral’, simple geometric figures (‘zig-zag’, ‘circle’, ‘left-swipe’, ‘right-swipe’, ‘rectangle’), and 3D polygonal chains (‘poly3Dxyz’, ‘poly3Dyxz’, ‘poly3Dzxy’, ‘poly3Dxzy’, ‘poly3Dyzx’, ‘poly3Dzyx’). The 2D skeleton trajectories of few classes of 26-Gestures dataset is show in Figure 4-2. Figure 4-3 presents the confusion

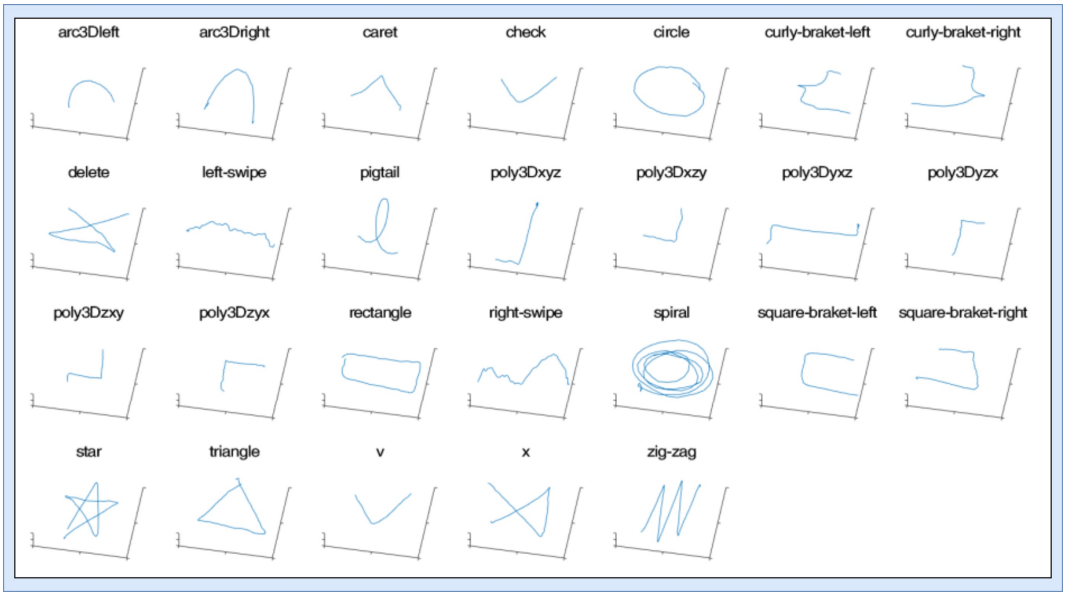


Figure 4-2: Shows samples from the 2D skeleton points trajectories of 26-Gestures dataset [82], illustrating different hand gesture patterns such as Spiral, Circle, Triangle, Zig-Zag, etc.

matrix of the proposed model on the 26-Gesture dataset. The model achieved 100% accuracy in most of the classes, with an overall average accuracy of 99.8%. To assess the classification accuracy of our system for a particular set of test data, precision(P), recall(R), and F1-score for both datasets as shown in Table 4.1.

The class-wise accuracy of the 26-Gestures dataset, as shown in Figure 4-4, indicates that the model performed exceptionally well across all classes, with only a few misclassified. Most of the classes achieved 100% accuracy, demonstrating the model's excellent classification performance on the 26-Gestures dataset.

The Table 4.1 shows performance metrics (Precision, Recall, F1-score) for the 26-

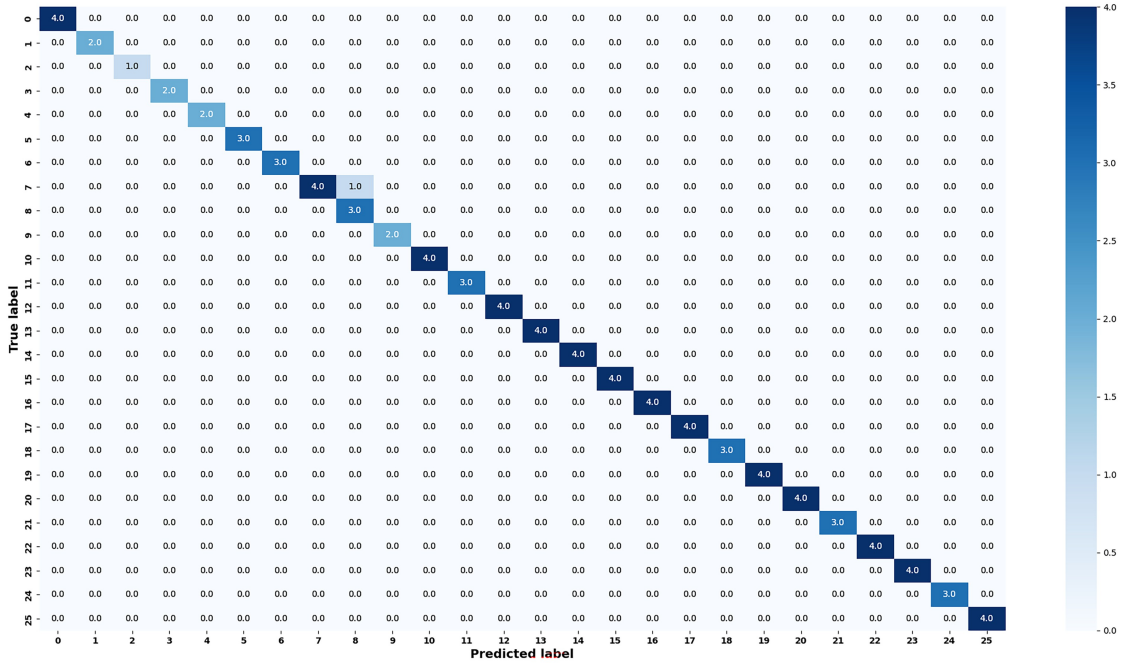


Figure 4-3: Confusion matrix of 26-Gestures dataset, the proposed model achieved 100 percent accuracy on most classes. The overall accuracy of the model is more than 99.80 percent.

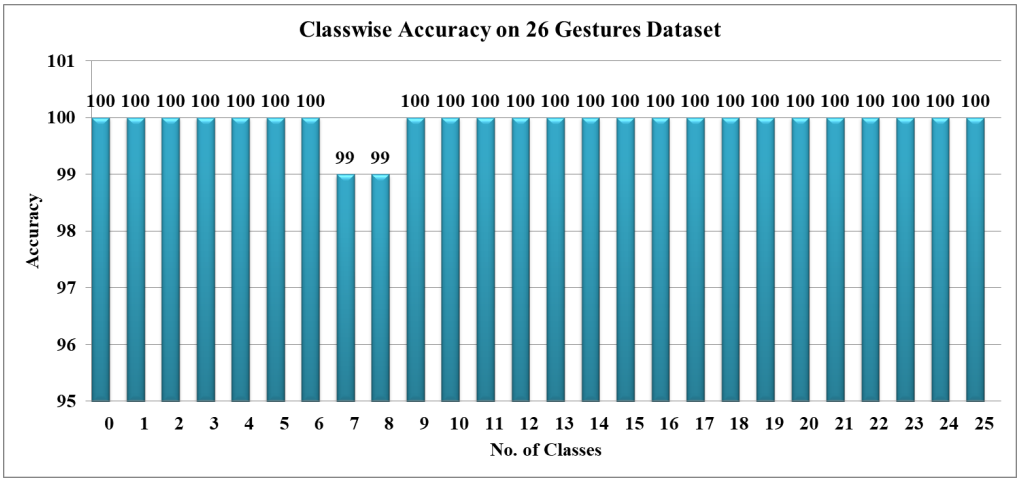


Figure 4-4: For the 26-Gestures dataset, the class accuracy is greater than 99% for all classes, and the average accuracy of our proposed model is 99.8%.

Gestures dataset. It exhibits high classification accuracy across most classes, with a few exceptions where performance dropped. In the 26-Gestures dataset, Class 4 has marginally lower metrics (P: 0.82, R: 0.90, F1: 0.86). This indicates that the model produces false positives and minimizes true positives, leading to less performance in accurately identifying the target class 4.

Table 4.1: F1, Precision(P), and Recall(R) values for 26-Gestures dataset and DHG14/28 dataset

Class	26G			DHG			Class	26G			DHG		
	P	R	F1	P	R	F1		P	R	F1	P	R	F1
0	1.00	1.00	1.00	1.00	1.00	1.00	13	1.00	1.00	1.00	1.00	0.95	0.97
1	1.00	1.00	1.00	1.00	1.00	1.00	14	1.00	1.00	1.00	-	-	-
2	1.00	1.00	1.00	1.00	0.95	0.97	15	1.00	1.00	1.00	-	-	-
3	1.00	0.79	0.88	1.00	1.00	1.00	16	1.00	1.00	1.00	-	-	-
4	0.82	1.00	0.90	1.00	1.00	1.00	17	1.00	1.00	1.00	-	-	-
5	1.00	1.00	1.00	0.99	0.95	0.95	18	1.00	1.00	1.00	-	-	-
6	1.00	1.00	1.00	1.00	0.95	0.98	19	1.00	1.00	1.00	-	-	-
7	1.00	1.00	1.00	0.95	0.95	0.95	20	1.00	1.00	1.00	-	-	-
8	1.00	1.00	1.00	0.95	1.00	0.98	21	1.00	1.00	1.00	-	-	-
9	1.00	1.00	1.00	0.95	1.00	0.98	22	1.00	1.00	1.00	-	-	-
10	1.00	1.00	1.00	1.00	0.95	0.97	23	1.00	1.00	1.00	-	-	-
11	1.00	1.00	1.00	1.00	1.00	1.00	24	1.00	1.00	1.00	-	-	-
12	1.00	1.00	1.00	1.00	1.00	1.00	25	1.00	1.00	1.00	-	-	-

Figure 4-6 show how different model combinations perform with and without DDA loss on 26-Gestures dataset. Each graph shows that incorporating DDA loss consistently improves the accuracy across all model combinations, highlighting the effectiveness of the DDA loss function in enhancing classification performance. The biggest improvements are seen when combining multiple models, demonstrating that DDA loss helps in making the models better at classifying the data correctly.

DHG14/28

The DHG14/28[2] [15] dataset comprises depth data and skeleton data of hand joints. It includes 2800 gesture sequences across 14 different hand gesture classes. These gestures are performed by 20 participants, using either a single finger or the whole hand, with each gesture being performed 5 times. The skeleton data captures 22 hand

[2]http://www-rech.telecom-lille.fr/DHGdataset/

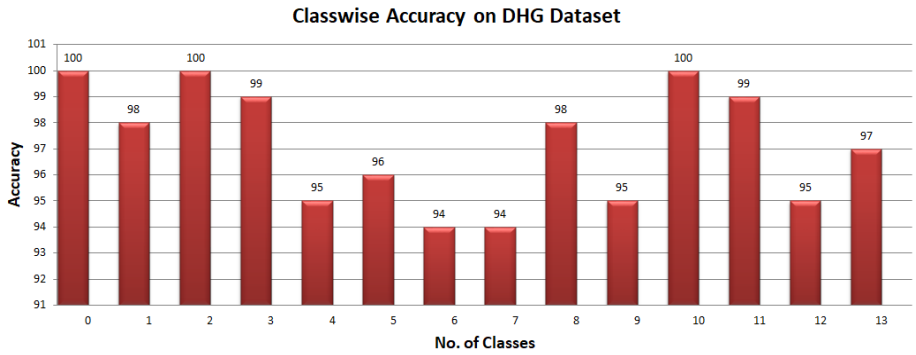


Figure 4-5: Shows class-wise accuracy on the DHG14/28 dataset, the model's performed excellent with most classes achieving 100% accuracy. The average accuracy of the model is 97.1%.

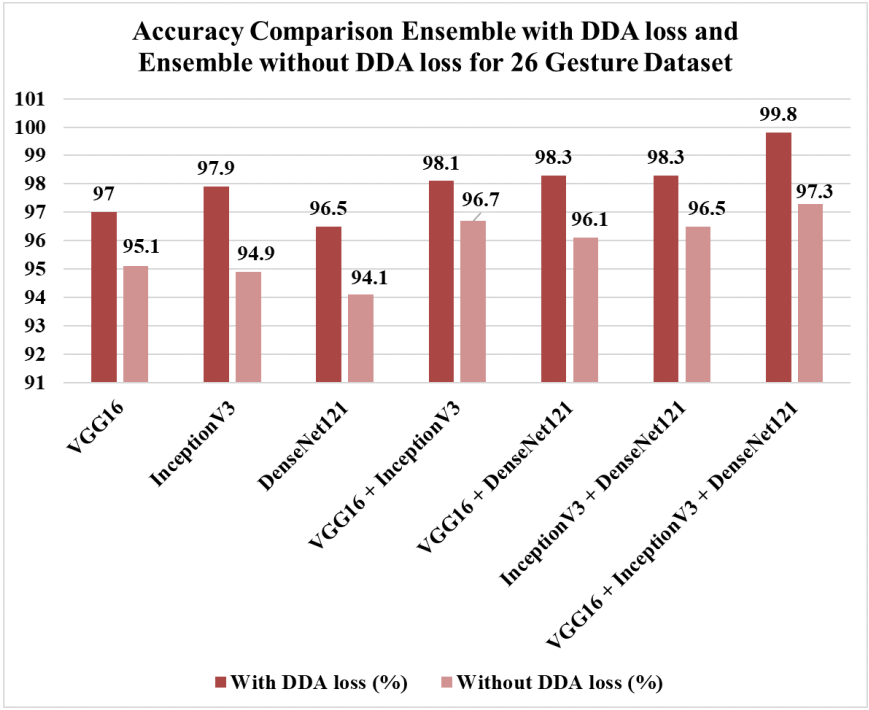


Figure 4-6: Accuracy comparison with or without DDA loss on 26-Gestures dataset. DDA loss improves accuracy for all model combinations. The biggest improvements are seen when combining multiple models.

joint points, representing the hand's shape, and is utilized for both 2D and 3D hand shape analysis. The information is gathered via the Intel RealSense short-range depth camera. There are two categories of gestures in the gesture inventory: 'Coarse' and 'Fine'. The 'Coarse' gestures include 'Tap', 'Swipe Up', 'Swipe Down', 'Swipe left', 'Swipe Right', 'Swipe V', 'Shake' 'Swipe X', and 'Swipe +'. The gestures like 'Rotation Clock Wise', 'Rotation anti-Clock Wise', 'Pinch', 'Expand', and 'Grab' come under 'Fine' gesture category. The class"0" represents as "Tap", class"1" represents as "swipe Up", class"2" as "swipe Down" and so on.

The 2D trajectory using a single finger of the hand shape of the DHG 14/28 gesture dataset is fed as an input to the proposed model. Figure 4-7 presents the confusion matrix of the proposed model on the DHG14/28 dataset. The model achieved 100% accuracy in most classes, with an overall accuracy of 97.1%. To assess the classification accuracy of our system for a particular set of test data, we calculate the precision(P), recall(R), and F1-score as shown in Table 4.1 shows performance metrics (Precision, Recall, F1-score) for the DHG14/28 datasets. This indicates that the model produces true positives and minimizes false positives, leading to good performance in accurately identifying the target class with a few exceptions like Classes 4 and 6 have slightly lower Recall (0.95) and F1-scores (0.97 and 0.98, respectively). The class-wise accuracy of the DHG14/28 dataset is shown in Figure 4-5. The model performed exceptionally well across all classes, with only a few instances of misclassification. Most classes achieved an accuracy of more than 94%, and several classes attained 100% accuracy. This demonstrates that the model achieved excellent classification performance on the DHG14/28 dataset across all classes.

Our experiments show that the F1-Score is greater than 99% for all classes, and the macro average accuracy of our proposed model is 97.1%, for some classes our approach attains 100% accuracy, proving that our proposed model achieved better results and performs remarkably well across all classes.

Figure 4-8 show how different model combinations perform with and without DDA loss on DHG14/28 dataset. They clearly show that using DDA loss improves accuracy for all model combinations. The biggest improvements are seen when combining

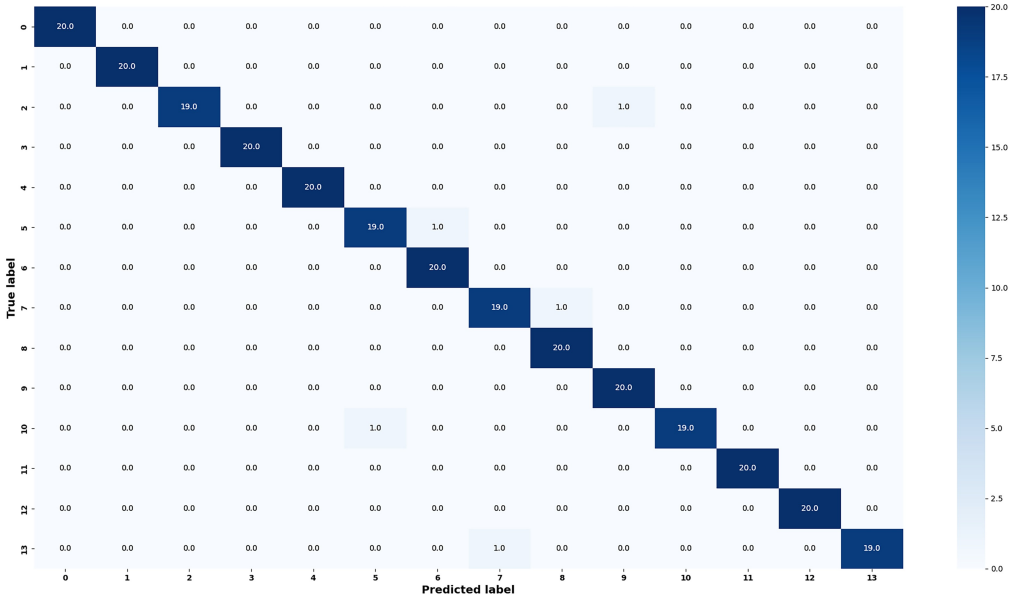


Figure 4-7: Confusion matrix of DHG14/28 dataset, the proposed model achieved 100 percent accuracy on most classes. The overall accuracy of the model is more than 97.1 percent.

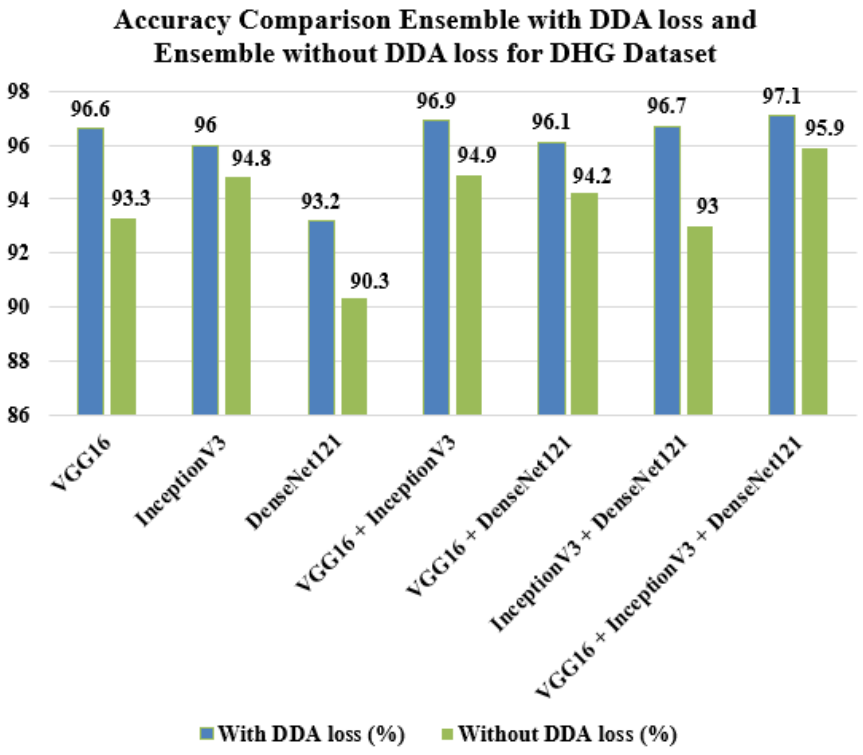


Figure 4-8: Accuracy comparison ensemble with DDA loss and ensemble without DDA loss on DHG14/28 dataset. DDA loss improves accuracy for all model combinations. The biggest improvements are seen when combining multiple models.

multiple models, demonstrating that DDA loss helps in making the models better at classifying the data correctly.

NTU RGB+D and NTU RGB+D 120

The NTU RGB+D[3] [90] and NTU RGB+D 120[4] [91] are the largest multi-modal action recognition datasets, containing skeletal, RGB, depth, and infrared data. The proposed model uses the 3D trajectory plots from both datasets as input. NTU RGB+D contains 60 classes of different actions. However, we have selected only 7 classes where participation of hand gesture involved to perform the action. Such classes are 'drinking water', 'clapping', 'writing', 'hand waving' 'type on keyboard', 'rub two hands', and 'put palms together'. NTU RGB+D 120 contains 120 classes; out of that we have selected 13 classes where hand gesture involved. Such classes are 'drinking water', 'clapping', 'writing', 'hand waving' 'type on keyboard', 'rub two hands', 'put palms together' 'thumb up', 'thumb down', 'make OK sign', 'make a victory sign', 'cutting nails', and 'snap fingers. The model achieved more than 90% class-wise accuracy on the mentioned classes of NTU RGB+D dataset are shown in Figure 4-9, the overall average accuracy on the proposed model is 98.71% The Figure 4-10 illustrates the model's classwise accuracy on mention classes of NTU RGB+D 120 dataset with an average accuracy of 96.23%.

Impact of the Proposed Model on Real-World Scenario

Our proposed model improves class separability and feature robustness, ensuring higher accuracy in gesture recognition. This is important in real-world settings where gestures can change due to lighting, angle, or speed. Our method contributes to the development of models that are capable of handling real-world problems by enhancing the model's performance in various scenarios. There are a wide range of possible uses for hand gesture recognition in both industry and research. In industries like healthcare or manufacturing, where minimising physical contact is essential, it can be

[3]https://rose1.ntu.edu.sg/dataset/actionRecognition/
[4]https://rose1.ntu.edu.sg/dataset/actionRecognition/

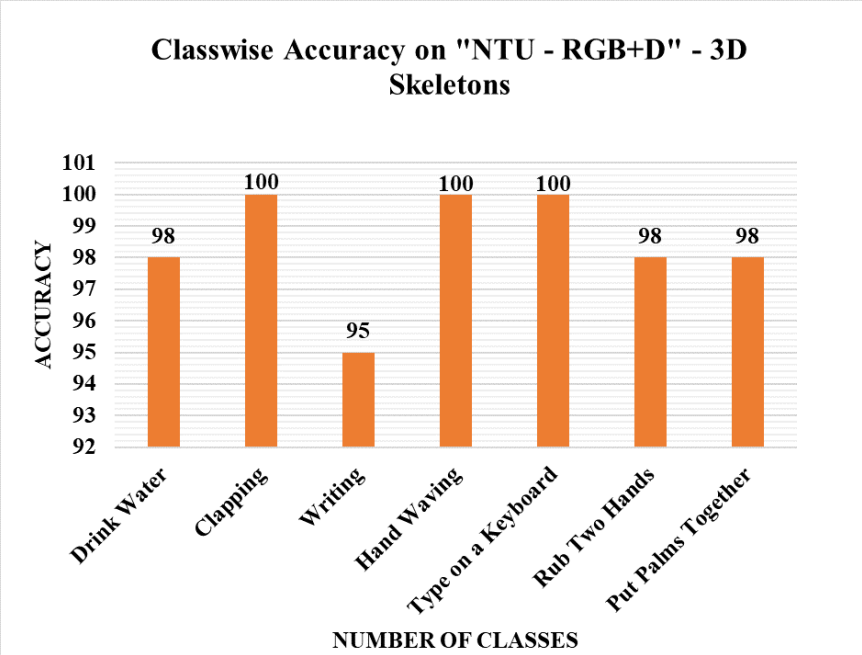


Figure 4-9: Shows class-wise accuracy of NTU RGB+D dataset. The class accuracy is greater than 95% for all classes, and the average accuracy of our proposed model is 98.71%.

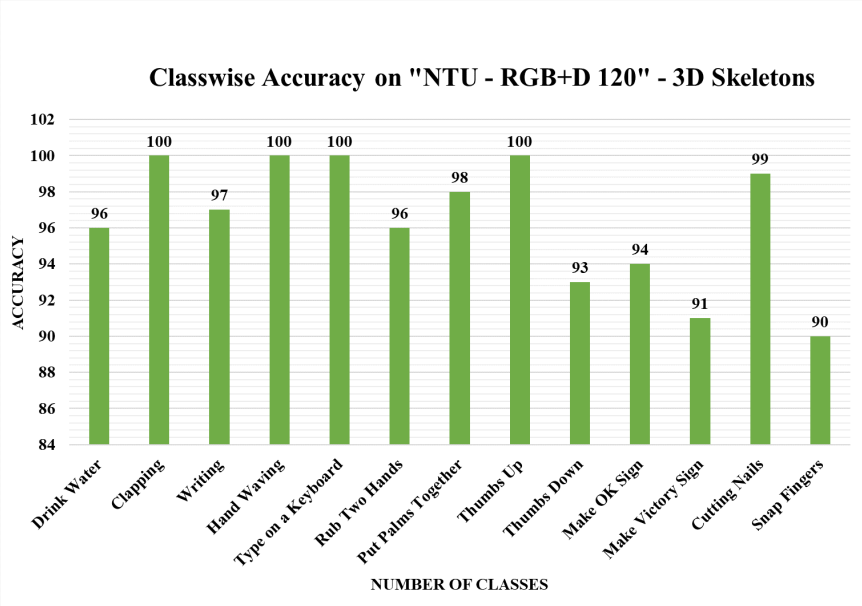


Figure 4-10: Shows class-wise accuracy of NTU RGB+D 120 dataset. The class accuracy is greater than 90% for all classes, and the average accuracy of our proposed model is 96.23%.

utilised in human-computer interaction systems to enable touch less control. It can also be used in games, virtual and augmented reality, and sign language recognition, which makes technology more user-friendly and accessible.

4.4.3 Ablation Study

Detecting and tracking gesturing hands remains a challenging task. To address this issue, an ensemble model with DDA loss is introduced which incorporates three modalities (VGG16 + InceptionNet V3 + DenseNet121) that can be utilized to improve gesture classification. The 26-Gestures dataset and the DHG14/28 dataset have skeleton information that helps in plotting 2D trajectory using a single finger to capture the hand gesture trajectory. We have discussed impact of with DDA loss and without DDA loss on various datasets.

- Ensemble without DDA Loss: Analyze the impact of ensemble learning using categorical cross-entropy as the loss function, without incorporating DDA loss. The result revealed that the final accuracy was 2.15% to 2.31% lower as compare to the ensemble learning with DDA loss. Our experiments, conducted on the 2D skeleton data of the 26-Gestures dataset and DHG14/28 datasets, clearly demonstrate that the Ensemble with DDA loss significantly outperforms its counterpart. These findings are detailed in Table 4.2 for DHG14/28 dataset and Table 4.3 for 26-Gestures dataset.

- Ensemble with DDA Loss: The effect of Ensemble learning with DDA loss was analyzed. The results show the final accuracy being 2.15% to 2.31% greater than that of the ensemble learning without DDA loss. Experimental results are evaluated over the 2D skeleton data of the 26-Gesture dataset and DHG14/28 dataset. As a result, the Ensemble with DDA loss outperforms the Ensemble without DDA loss as shown in Table 4.2 for DHG14/28 dataset and in Table 4.3 for 26 Gestures dataset and .

- Impact of Dataset split into 70-30, 80-20 and 90-10: The experiment was performed on two benchmark datasets, splitting each dataset into $70 - 30\%$,

$80-20\%$ and $90-10\%$ and results are shown in Table 4.3 for 26-Gestures dataset and Table 4.2 for DHG14/28 dataset. As we can see in the table in both cases Ensemble with DDA loss and Ensemble without DDA loss, outperforms best in $90-10\%$ split as compared to the 70-30 split and 80-20 split.

Table 4.2: Ablation study with DDA loss and without DDA loss on DHG14/28 dataset

S.No.	Models			Without DDA Loss (A%)			With DDA Loss (A%)		
	V	I	D	70-30	80-20	90-10	70-30	80-20	90-10
1	✓	×	×	89.9	91.9	93.3	94.2	95.1	96.6
2	×	✓	×	91.1	92.9	94.8	95.4	95.8	96.0
3	×	×	✓	87.9	89.9	90.3	91.6	92.7	93.2
4	✓	✓	×	92.4	93.2	94.9	95.8	96.1	96.9
5	✓	×	✓	92.6	93.0	94.2	95.6	95.8	96.1
6	×	✓	✓	91.1	92.7	93.0	94.1	95.7	96.7
7	✓	✓	✓	93.0	94.8	95.9	95.7	96.8	97.1

V=VGG16, I=InceptionNet V3, D=DenseNet121, A=Accuracy

Table 4.3: Ablation study with DDA loss and without DDA loss on 26-Gestures dataset

S.No.	Models			Without DDA Loss (A%)			With DDA Loss (A%)		
	V	I	D	70-30	80-20	90-10	70-30	80-20	90-10
1	✓	×	×	93.1	94.9	95.1	94.7	96.2	97.0
2	×	✓	×	93.1	93.9	94.9	94.1	96.3	97.9
3	×	×	✓	92.7	92.8	94.1	93.9	95.8	96.5
4	✓	✓	×	93.5	94.2	96.7	95.9	97.3	98.1
5	✓	×	✓	93.5	94.9	95.8	96.0	97.5	98.3
6	×	✓	✓	94.1	95.1	96.5	96.3	97.5	98.3
7	✓	✓	✓	95.1	96.0	97.3	96.8	98.6	99.8

V=VGG16, I=InceptionNet V3, D=DenseNet121, A=Accuracy

4.4.4 Comparison with Literature

We compare the experimental results with state-of-the-art on the 26- Gestures dataset and DHG14/28 dataset. Table 4.4 presents a comparison between the proposed model and current approaches. In this paper [10], the author uses fingertip skeleton information to create trajectories for hand gesture recognition. Similarly, another author [82]

also uses skeleton data and proposes a 3D gesture recognizer model based on trajectory matching of a single hand. In this paper [31], the author proposed motion feature-augmented RNN for dynamic hand gesture recognition using skeleton data. Similarly, in this paper [15], the author uses the geometric shape of the hand to extract descriptors from hand skeleton joints also uses skeleton data for 3D hand gesture recognition, utilizing the geometric shape of the hand to extract descriptors from hand skeleton joints. other authors [61] [65], uses LSTM and 3DCNN models for dynamic hand gesture recognition while integrating RGB and skeleton modalities. These models aim to address challenges in dynamic hand gesture recognition, such as inter and intra-class variation. In a proposed model, we implemented an ensemble model with DDA loss. The results shown in Table 4.4 demonstrate that the proposed model surpasses other state-of-the-art methods on the 26-Gestures dataset and DHG14/28 dataset, achieving an accuracy of 97.1% on DHG14/28 dataset and an accuracy of 99.8% on 26-Gestures dataset.

Table 4.4: Results of comparing the 26-Gestures dataset and DHG14/28 Dataset's classification accuracy(%) with SOTA

Papers	**26G**	**D**	**Acc(%)**
Grassmann Manifold [10]	✓	×	99.3
3D algorithm [82]	✓	×	96.9
GGDA [10]	×	✓	88.4
RNN [31]	×	✓	84.6
SoCJ+HoHD [15]	×	✓	82.2
HoWR+SoCJ+HoHD [15]	×	✓	88.3
LSTM [61]	×	✓	84.5
3DCNN [65]	×	✓	94.8
CNN [92]	×	✓	94.6
Ensemble+DDA loss	×	✓	**97.1**
Ensemble+DDA loss	✓	×	**99.8**

26G=26 Gestures dataset,D=DHG14/28 dataset, Acc=Accuracy

4.5 Conclusion

In this chapter, we emphasize ensemble learning with the DDA loss model, a new framework for dynamic hand gesture recognition that effectively addresses inter-class and intra-class variations challenges. Hand gesture recognition has several uses in

both industry and research. In manufacturing and healthcare, it can reduce physical contact by enabling touch less control in human-computer interaction systems. Additionally, it enhances user-friendliness in sign language recognition, automation and gaming. The experiment is performed on the DHG 14/28 and 26-Gesture dataset. The skeleton 2D trajectory images are fed as input to the proposed model. Each layer of the CNN model's VGG16, Inception V3, and DenseNet121 extracts the features separately. The output from each model is combined in the ensemble layer. After feature extraction, they are put together in an ensemble learning layer and processed with the DDA loss function that controls class variations to increase recognition accuracy and classify gestures. Our model performs competitively on the 26-Gestures and DHG14/28 benchmark datasets, achieving more than 99% accuracy on the 26-Gestures dataset and more than 97.0% accuracy on the DHG 14/28 dataset, matching state-of-the-art methods.

The proposed hand gesture recognition framework increases gesture recognition accuracy and efficiently handles intra and inter-class variability in hand gesture recognition by integrating ensemble learning with a DDA loss. Use of skeleton data also overcome the challenge of hand detection in occlusion and cluttered background.

Chapter 5

Motion Feature Estimation using Bi-Directional GRU for Skeleton-based Dynamic Hand Gesture Recognition

Reena Tripathi, and Bindu Verma. "Motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recognition." *Signal, Image and Video Processing* (2024): 1-10.. (SCIE Indexed, IF: 2) DOI: https://doi.org/10.1007/s11760-024-03153-w *(Published)*

Reena Tripathi, and Bindu Verma. "Skeleton Data is all about: Dynamic Hand Gesture Recognition.". *In 2023 Seventh International Conference on Image Information Processing (ICIIP)* (pp. 576-585) (2023, November). IEEE. *(Published)*

5.1 Introduction

In this chapter, we introduce a hybrid deep-learning model called motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recog-

nition, with skeleton and optical flow data fusion. As demonstrated by our previous frameworks discussed in Chapters 3 and 4, both RGB and skeleton data are essential for extracting meaningful information needed for reliable and accurate gesture recognition. Therefore, in order to increase recognition accuracy, we combine the skeleton data with optical flow data in the proposed model. Furthermore, the geometric features used in our work in Chapter 4 perform effectively when fingertips are accurately detected, restricting the system to cases where the gesturing hand does not self-occlude—unless skeleton data is also available. Similarly, while RGB data performs well in Chapter 3, it falls short in scenarios with highly cluttered backgrounds or significant variations in illumination makes hand tracking difficult. Due to different lighting conditions, cluttered background, self co-articulation, noisy images, and occlusion [93] [94] hand detection and tracking may not be accurate. Due to the aforementioned issues, it is very difficult to detect and track the gesturing hand. To make a robust gesture recognition system, hand detection, and tracking steps must be performed flawlessly to propose a generic system.

The main motivation of this chapter is to propose a generic framework that does not require detection and tracking of the gesturing hand and recognize the dynamic hand gesture with high accuracy. We have created skeleton trajectory video using skeleton data and optical flow video using RGB/Depth data. Skeleton data was used because it does not contain background information. Therefore, skeleton-based models will not be affected by complex background information. Similarly, the creation of an optical flow video contains the movement of the hand irrespective of any background information. Thus, it filters out irrelevant data and concentrates on the gesturing hand that helps in extracting temporal features. For each skeleton trajectory video, features are extracted using Xception-Net [87] called as a finger motion feature (FMF) and features extracted from optical flow videos are global motion features (GMF). Features extracted from a single modality is not sufficient enough to classify the dynamic hand gesture, thus we proposed a fusion of FMF and GMF that gives better accuracy compared to the single modality. Recognizing the dynamic hand gesture requires sequential learning of spatio-temporal features. Thus recurrent

neural networks can be used for sequential learning but they suffered from vanishing gradient problems due to back-propagation. Due to cell memories and gates to learn the sequential data, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) were introduced that solve the vanishing gradient problem. Thus, this paper used a Bi-GRU network for sequential learning, and SoftMax function with crass entropy loss is used to classify the dynamic hand gesture. Both FMF and GMF features are passed for sequence-to-sequence learning.

5.2 Literature Survey

Due to the advancement in computing devices and an increasing amount of data deep learning-based method performs tremendously in many fields such as image classification, object detection, activity recognition, and dynamic hand gesture recognition. The author Liu et al. [65] proposed a decoupled two-stream network where in one stream 2DCNN is used to process the Hand Posture Evolution Volume (HPEV) and in another stream, 3DCNN is used to process the Hand Movement Map (HMM) techniques. The features of both streams are fused to predict the final accuracy. The lightweight model for gesture detection put forth by Mujahid et al. [29] relies on DarkNet-53 and YOLO (You Only Look Once) V3 convolutional neural networks. The author Lie et al. [95] used AlexNet to extract features and SVM for the classification. Similarly, the author Jimin et al. [96] proposed a dynamic gesture recognition technique that addresses the problems of complexity, computational difficulty, and sluggish training. Authors Zheng et al. [97] proposed a depth motion map (DMM) to represent the hand shape and features are extracted using DLEH2 and dynamic hand gesture are classified. The proposed DLEH2 is robust against illumination, and cluttered background. The multiview hierarchical aggregation network, which was proposed by the author Shaochen et al. [84], solves the issue of difficulties in locating hand gestures because of the complex structure of the hand. The author used virtual cameras to capture hand skeletons from different angles and CNN was used for feature extraction.

Skeleton-based recognition added a breakthrough in the field of dynamic hand gesture recognition that overcome the challenges of occlusion, illumination, and background clutter. Skeleton information gives the basic structure of the hand shape and hand articulation points and helps in creating a pseudo skeleton image. For the recognition of dynamic hand gestures, authors Lei et al. [64, 65] used skeleton data and proposed a 3DCNN model coupled with a Hidden Markov Model. The author Xinghao et al. [98] uses skeletal data in motion feature augmented network (MFA-Net) to recognize the dynamic hand gesture and used a variational autoencoder to extract the features. In addition to a temporal encoding of the gesture dynamics, the author Smedt et al. [99] developed a method using three gestural features to capture the hand shape and motion information. Li et al. [36] included a spatial perception stream (SP-Stream) that uses the convex hull of the hand to encode skeletal images and a temporal perception stream (TP-stream) that records hand gestures. Other authors Adam et al. [66] proposed a multi-model ensemble gesture recognition network (MMEGRN), a sophisticated ensemble architecture for skeleton-based gesture recognition. A bi-stream activity has recently caught researcher's interest in the field of computer vision, including hand gesture recognition [100]. The author Mehran et al. [100] proposed DeepGRU a new deep network model for gesture and action recognition. Instead of using LSTM, DeepGRU uses raw skeleton, posture, or vector data with Bidirectional Gated Recurrent Units (Bi-GRU) to speed up computation and perform well with sparse training data. A bidirectional gated recurrent unit (Bi-GRU) model for hand gesture recognition was proposed by the authors Bindu et al. [101]. They proposed two stream model where optical flow video and RGB videos are inputted into GoogleNet to extract the feature and Bi-GRU for sequence to sequence learning. Sharma et al. [7] used optical flow motion templates and used 2DCNN and 3DCNN to extract the features from RGB videos and in last features are fused to classify the dynamic hand gesture. The two modules BE (Behaviour Encoder) and FSTA (Fine-grained Spatio-temporal Attention) was proposed by the author Wenwei et al. [102]. The BE module prioritizes behavioral inputs and minimizes unnecessary information, which improves hand gesture recognition accuracy. Authors Miki

et al. [103] proposed low computational spiking neural networks (SNNs) that process temporal information. A sequence of depth gestures is passed to the SNN model and gestures are classified based on the firing frequency. The author Sataya et al. [92] proposed a method for real-time hand gesture recognition using a skeleton-based approach. The proposed model uses an LSTM network and a multi-channel CNN to record hand gestures in both space and time. Machine learning models struggle with accuracy and complexity problems in dynamic hand gesture recognition to overcome this issue the author Yun et al. [104] proposed a new method that integrates a CNN and Transformer model using attention mechanisms, enhancing spatial and temporal feature extraction.

5.3 Proposed Architecture

The architecture of the proposed model is shown in Figure 5-1. The proposed model is pipe-lined in two streams and carried out concurrently. In the first pipeline if skeleton data is available we directly used skeleton data to plot the skeleton point trajectory video. If no skeleton data is available, the media pipe library discussed in Section 5.3.2 is used to extract the skeleton point, and skeleton point videos are generated. The advantage of using skeleton point video is that it overcomes the challenges of hand occlusion, illumination, and complex background. In the second pipeline from RGB/Depth data optical flow video is calculated as discussed in Section 5.3.1. The advantage of calculating the optical flow video is that it captures the hand motion and discards the stationary background. The detailed algorithm of the proposed model is shown in Algorithm 5.1. After calculating the skeleton and flow video features are extracted using 2DCNN Xception-Net as discussed in Section 5.3.3 and represented in the form of FMF and GMF matrix. Then these features are passed to the Bi-GRU unit for sequence-to-sequence learning. The output of both Bi-GRU units is averagely fused using Equation 5.13 and is flattened at a fully connected layer. In the last SoftMax layer with cross-entropy loss is applied to get the final probability score using Equation 5.14.

Algorithm 5.1 Algorithm of the Proposed Model.

Input:
Feature Extraction
a) Finger Motion Feature(FMF):
(i) Generate skeleton points video either using skeleton data/ using media pipe.
(ii) Extract features using Xception-Net
Sequence Matrix $FMF = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & . & . & . & X_{1n} \\ X_{21} & X_{22} & X_{23} & X_{14} & . & . & . & X_{2n} \\ . & . & . & . & . & . & . & . \\ & . & . & . & . & . & . & . \\ & . & . & . & . & . & . & . \\ X_{m1} & X_{m2} & X_{m3} & X_{m4} & . & . & . & X_{mn} \end{bmatrix}$
b) Global Motion Feature(GMF):
Generated optical flow video(OFV) using RGB/depth data
Extracted features using Xception-Net
Compute the output matrix as follows:
Sequence Matrix $GMF = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & . & . & . & Y_{1n} \\ Y_{21} & Y_{22} & Y_{23} & Y_{14} & . & . & . & Y_{2n} \\ . & . & . & . & . & . & . & . \\ & . & . & . & . & . & . & . \\ & . & . & . & . & . & . & . \\ Y_{m1} & Y_{m2} & Y_{m3} & Y_{m4} & . & . & . & Y_{mn} \end{bmatrix}$
Pipeline: Input to Bi-GRU is both the matrix.
(i) FMF input to Bi-GRU for sequence learning.
(ii) GMF input to Bi-GRU for sequence learning.
Decision level fusion using eq.5.13
Flatten the output at the fully connected layer.
Classify using softmax represented in eq.5.14

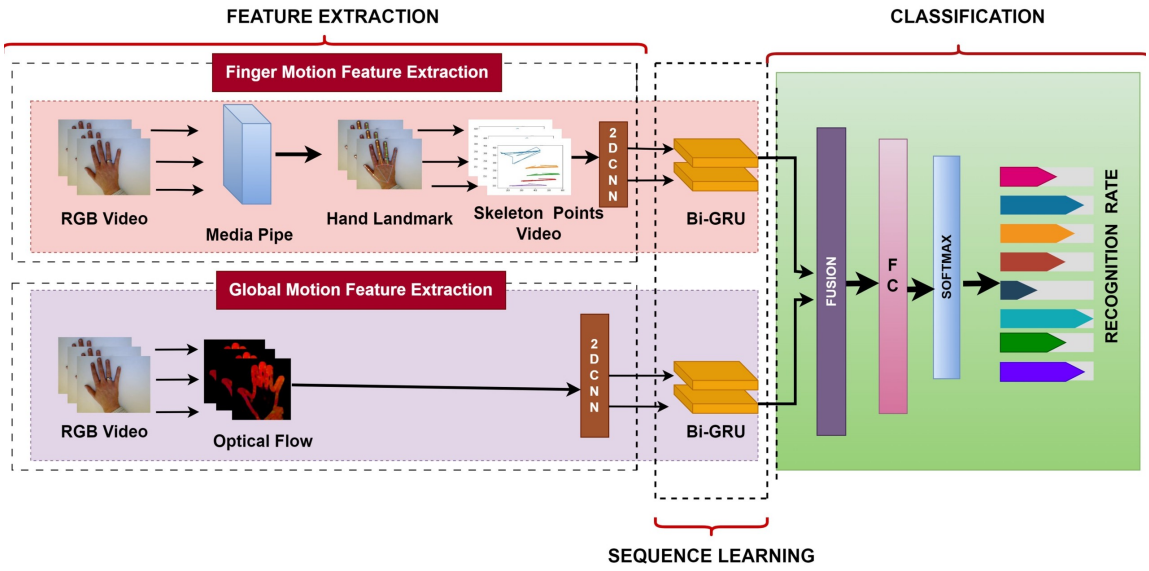


Figure 5-1: Shows a two-stream pipe-lined 2DCNN with a Bi-GRU for sequential learning. In order to extract features from each video frame, first, the skeleton points plot videos and dense optical flow sequences are input into two 2DCNN concurrently. Second, for sequential learning input layer of Bi-GRU received the extracted FMF and GMF features. Features are fused from both the pipe-line and flattened at FC layer and Softmax with cross-entropy loss is used to obtain the final prediction.

5.3.1 Global Motion Feature

For each RGB/Depth video optical flow video is generated that captured the motion of the moving hand and discards the unnecessary background. Thus, it captures the hand shape, and movement of the hand along with finger movement, and features extracted from these video frames are global motion features (GMF). For each optical flow video, features are extracted using 2DCNN Xception-Net called global motion feature and represented using the matrix 5.7.

Optical flow

The motion of each pixel in a video frame between any two frames is determined by dense optical flow [105]. In our proposed work, we computed the dense optical flow videos using well-known Gunnar Faneback's approach. It determines the displacement of each pixel between frames based on the presumption that adjacent pixels in a picture have comparable motion patterns. An optical flow video is generated by

using optical flow to record the hand gesture's movements. For each RGB/Depth video, we calculated the optical flow video as shown in Figure 5-2. The Figure 5-2 a) displays a few RGB frames from a video clip, Figure 5-2 b) depicts a skeleton point that depicts finger motion, and Figure 5-2 c) depicts corresponding optical flow video that represents global motion. As we can see in Figure 5-2 c), the background portion has been eliminated, leaving only the moving object to be recorded. The recognition accuracy of the proposed model increases by incorporating both optical flow videos and skeleton videos.

Assuming that at time T, pixel P(X, Y, T) represents a pixel in a picture with the coordinates X and Y, and that pixel P moves to $\triangle X, \triangle Y$ in the following frame after $\triangle T$ time.

$$P(X, Y, T) = P(X + \triangle X, Y + \triangle Y, T + \triangle T) \tag{5.1}$$

The displacement Equation 5.1 can be written using the Taylor approximation as :

$$\frac{\triangle P}{\triangle X}\delta X + \frac{\triangle P}{\triangle Y}\delta Y + \frac{\triangle P}{\triangle T}\delta T = 0 \tag{5.2}$$

after dividing Equation 5.2 by δT we obtained :

$$\frac{\Delta P}{\Delta X}W + \frac{\Delta P}{\Delta Y}Z + \frac{\Delta P}{\Delta T} = 0 \tag{5.3}$$

where W, and Z are referred to as the flow vectors, $W = \frac{\delta X}{\delta T}$ and $\frac{\triangle P}{\triangle X}$, known as an image gradient along the horizontal axis, and $Z = \frac{\delta Y}{\delta T}$ and $\frac{\triangle P}{\triangle Y}$, known as an image gradient along the vertical axis, and $\frac{\triangle P}{\triangle T}$ image gradient with time. Calculating the flow vectors W and Z are as follows:

$$\begin{bmatrix} W \\ Z \end{bmatrix} = \begin{bmatrix} \sum_a f_{X_a}^2 & \sum_a f_{X_a} f_{Y_a} \\ \sum_a f_{X_a} f_{Y_a} & \sum_a f_{Y_a}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_a f_{X_a} f_{T_a} \\ \sum_a f_{Y_a} f_{T_a} \end{bmatrix} \tag{5.4}$$

where $f_X = \frac{\triangle P}{\triangle X}$ and $f_Y = \frac{\triangle P}{\triangle Y}$ are the image gradient of all the point in the image. The intensity variations for each point in the image are represented in this Equation

by the image gradients f_X and f_Y. where,

$$f_X = \Delta P / \Delta X \tag{5.5}$$

and

$$f_Y = \Delta P / \Delta Y \tag{5.6}$$

which, respectively, show the variations in intensity along the horizontal and vertical axes. The optical flow's magnitude and direction are determined using the flow vector. The image is color-coded to show the optical flow, with the hue value representing the direction and the value plane in the HSV image space representing the magnitude. Each RGB video is also converted into a dense optical flow video, which is then used by a 2DCNN to determine frame-level characteristics.

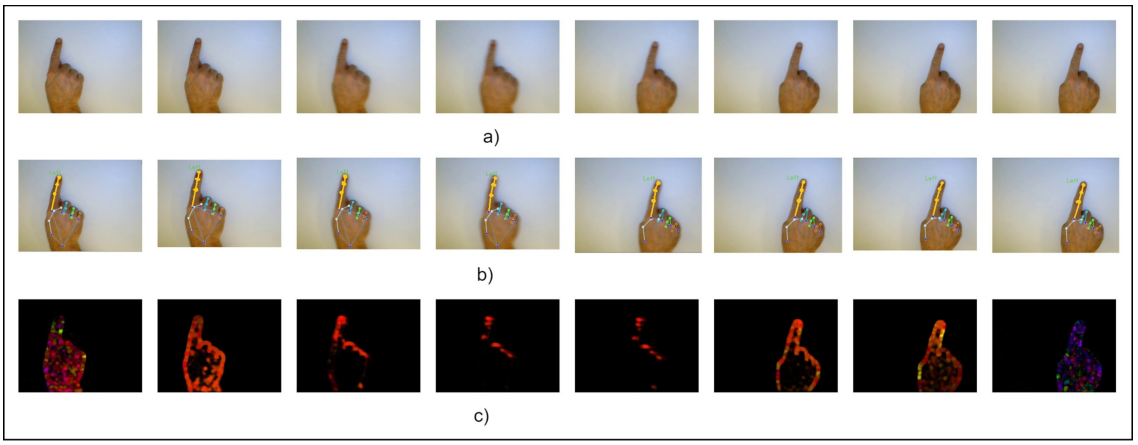


Figure 5-2: a) Shows RGB frames (b) Shows frames of skeleton point video and (c) Shows frames of optical flow motion video.

5.3.2 Finger Motion Features

If skeleton data is available in the dataset, no need to extract the skeleton point using the media pipe library. Skeleton data is directly used to plot the hand trajectory and generation of skeleton video. If only RGB data is available then for each RGB video we have to find the skeleton points using Media Pipe library [106] and track the finger movement across the frames and a trajectory is plotted. All fingertips are tracked

and the trajectory is plotted as shown in Figure 5-3. For each skeleton video features are extracted using 2DCNN called a finger motion feature (FMF) and represented using matrix 5.7.

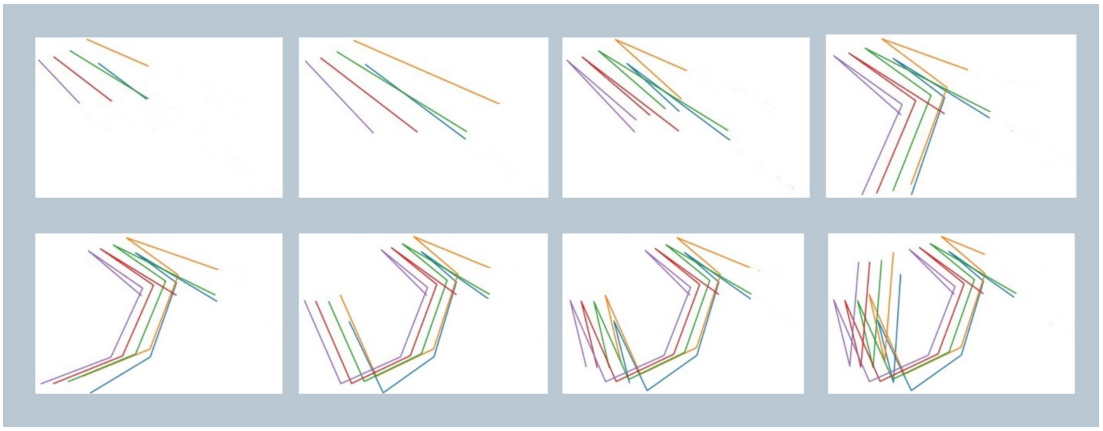


Figure 5-3: Circle trajectory formation of fingertips using skeleton points.

Skeleton Point Extraction using Media Pipe

An open-source framework called Media Pipe [106] is created by Google. In our proposed model for detecting the hand's skeleton points, we used a media pipe, which is a pre-trained model for detecting hand landmarks in real-time. The hand landmark is a task that finds the locations of hands in still photos and video frames that have been decoded. Several parameters are provided in the setup for locating the hand landmark, including running mode, the number of hands that can be detected in total, the minimum scores for hand detection, and the minimum tracking confidence. For hand landmarks detection it is necessary to find out palm detection.

The hand landmark detection model locates 21 hand-landmark coordinates as key points inside the identified hand regions. The general key points are named as: 'WRIST',' THUMB_CMC', 'THUMB_MCP', 'THUMB_IP', 'THUMB_TIP', 'INDEX_FINGER_MCP', 'INDEX_FINGER_PIP', 'INDEX_FINGER_DIP', 'INDEX_FINGER_TIP', 'MIDDLE_FINGER_MCP', 'MIDDLE_FINGER_PIP', 'MIDDLE_FINGER_DIP', 'MIDDLE_FINGER_TIP' , 'RING_FINGER_MCP', 'RING_FINGER_PIP', 'RING_FINGER_DIP', 'RING_FINGER_TIP' , 'PINKY

_MCP', 'PINKY_PIP', 'PINKY_DIP', 'PINKY_TIP' is shown in Figure 5-5. The media pipe's operation is depicted in Figure 5-4. Two methods are used to recognize hand gestures.

- The palm detector method: The image is processed using a palm detector model, which creates an orientated boundary line surrounding the hand.
- The Hand landmark detection method: In order to identify 3D hand key points, a hand landmark model analyses the cropped boundary line image.
- Skeleton Points Video: The fingertip of the obtained skeleton points are plotted further to make a skeleton points video.

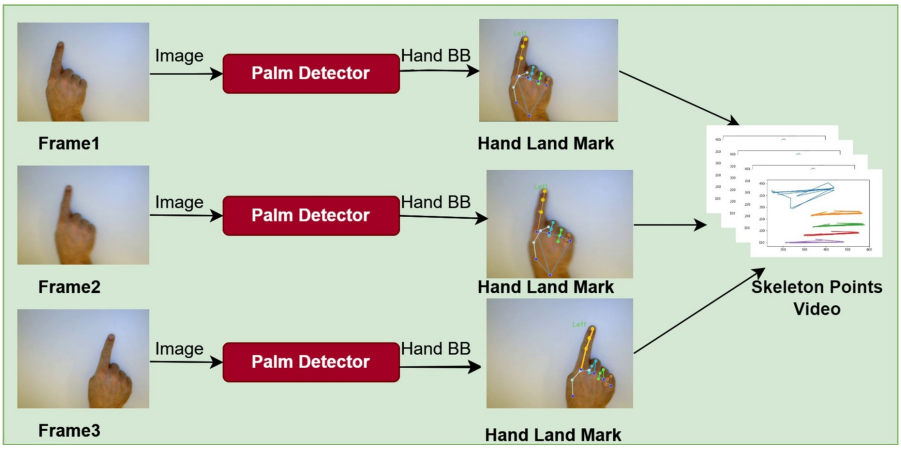


Figure 5-4: Shows working of media pipe, using palm detector and hand landmark detection method.

The selection of skeleton key points detection model is based upon the requirement of the need of the dataset used. In this paper, we are using a large sequential dataset. Whereas media pipe's architecture is designed to handle scalability, making it suitable for applications that may require processing multiple streams or handling large amounts of data concurrently. Media pipe offers pre-trained models and a user-friendly API, simplifying the integration process. The Media pipe framework's performance can not be affected by rotation, speed, and scale of input data. With some little interference from motion blur, MediaPipe performed better in detecting skeleton key points than other detection models like OpenPose. In comparison to

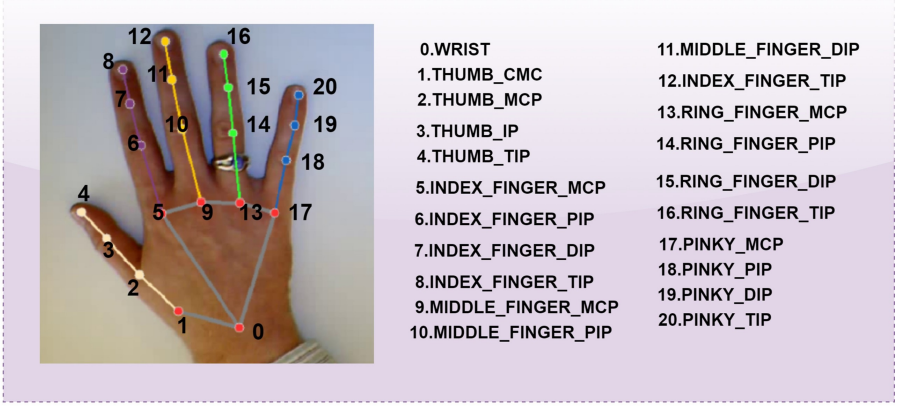


Figure 5-5: Shows 21 skeleton points of hand (hand landmarks).

MediaPipe, using alternative detection techniques like open pose proved extremely slow. Even on decent equipment, processing videos requires a lot of computing power and time taking. As a result, we use MediaPipe for detecting skeleton key points in our proposed work [107]. Media pipe detects skeleton points in low and varying illumination conditions as shown in Figure 5-6. Media pipe also detects skeleton points of occluded hand as shown in as shown in Figure 5-7.

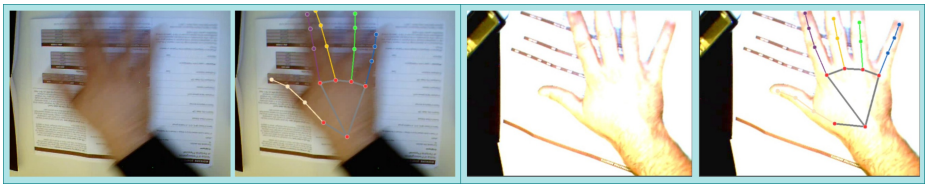


Figure 5-6: Hand in low and varying illumination conditions and corresponding skeleton key points.

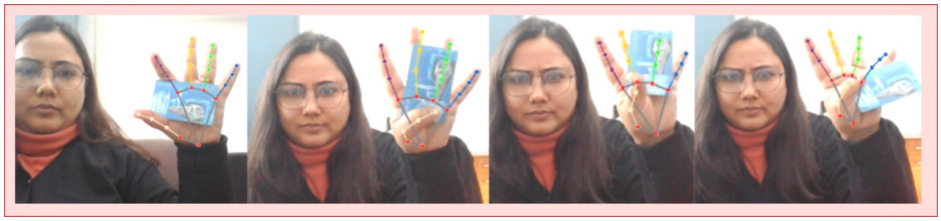


Figure 5-7: Skeleton key points detected in occluded hand.

5.3.3 2D-Convolutional Neural Network

Xception-Net [87] is a deep convolutional neural network (CNN) architecture that has faster training and a significantly improved version of the Inception-V3 model. We used a pre-trained Xception-Net model to extract the finger motion and global motion features. Each video frames run through the Xception-Net model as shown in Figure 5-8, and the average pooling layer's features are then retrieved and saved as a feature vector. The features at the average pooling layer are kept in a features vector X_V={ X_1, X_2, X_3, X_4, ..., X_n} for a given gesture sequence G with frames { F_1, F_2, F_3, F_4, ..., F_n}. The size of the feature vector X_1 is 1024 and for each optical flow video having n frame is $1024 \times n$ matrix. Likewise, features of all the optical flow videos are stacked (row-wise) and form a 3D matrix of size $m \times 1024 \times n$. Where m is the number of videos in the dataset. In addition, all of the gesture features are gathered and saved in a sequence matrix, GMF/FMF as shown in the matrix 5.7, and this matrix is passed to Bi-GRU for sequence-to-sequence learning.

$$GMF/FMF = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & . & . & . & X_{1n} \\ X_{21} & X_{22} & X_{23} & X_{14} & . & . & . & X_{2n} \\ . & . & . & . & . & . & . & . \\ & . & . & . & . & . & . & . \\ & . & . & . & . & . & . & . \\ X_{m1} & X_{m2} & X_{m3} & X_{m4} & . & . & . & X_{mn} \end{bmatrix} \tag{5.7}$$

5.3.4 Sequential Learning: Bi-directional GRU(Bi-GRU)

Recurrent neural networks, Gated Recurrent Units (GRU) as shown in Figure 5-9, and Bi-directional GRU (Bi-GRU) as shown in Figure 5-10 are frequently utilized in applications involving sequential data processing and natural language processing. Cho et al. [108] introduced the GRU (Gated Recurrent Unit), a form of recurrent neural network, in 2014. The GRU cell is more computationally efficient since it has fewer parameters comparable to the LSTM cell. The GRU selectively updates

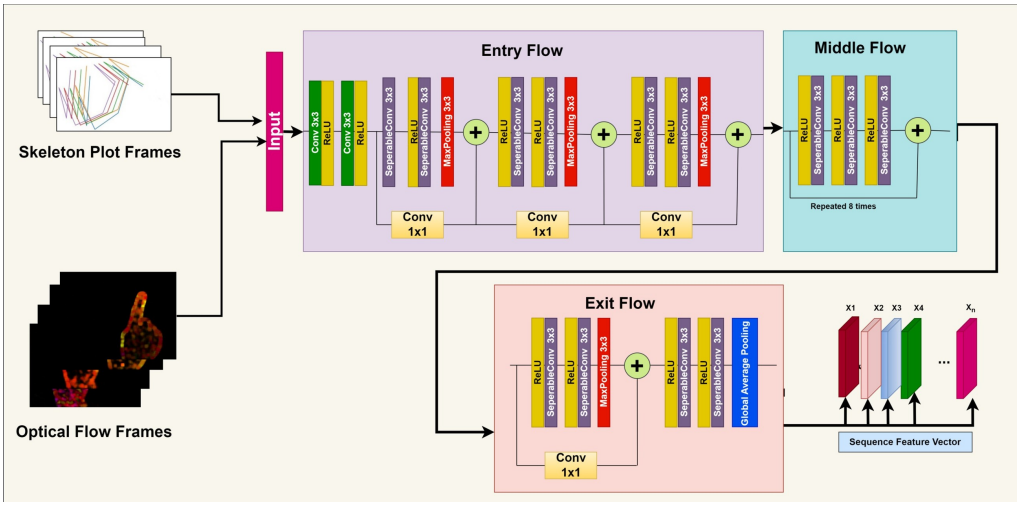


Figure 5-8: Xception-Net architecture [87]

the hidden state and memory cell using gating mechanisms as a result it solves the problem of vanishing gradients that can happen in conventional recurrent neural networks. The reset gate and the update gate are both components of the GRU cell. The update gate regulates the amount of fresh candidate activation utilized in the current time step, whereas the reset gate regulates the amount of the prior hidden state used in the current time step. Based on the current input and prior hidden layer output, the update gate utilizes a sigmoid neural layer to selectively add or delete information from the input. Equation 5.8 is used to determine the update gate's function.

$$U_T = \sigma((w^{(U)} X_T + B^U) + (q^{(U)} H_{T-1} + B^U)) \tag{5.8}$$

Where U_T represents as update gate, H_{T-1} as the output of the hidden layer. X_T is a current input that has been inserted into a network unit and multiplied by its own weight $w^{(U)}$ and biases are included, H_{T-1} is a prior time stamp information that has been multiplied by its original weight $q^{(U)}$ and biases.

$$R_T = \sigma((w^{(R)} X_T + B^R) + (q^{(R)} H_{T-1} + B^R)) \tag{5.9}$$

Similarly, in Equation 5.9, R_T represents the reset gate, which selects the exact amount of the prior information to forget.

In Equation 5.10 the reset gate R_T is used in the next steps to determine the memory content H'_T in order to obtain the necessary information from the past.

$$H'_T = tanh((wX_T + B) + R_T \odot (qH_{T-1} + B)) \tag{5.10}$$

$$H_T = U_T \odot H_{T-1} + (1 - U_T) \odot (H'_T) \tag{5.11}$$

The combined findings from both steps are applied in the last stage, followed by tanh activation and H_T is finding out to keep the most recent information and transmit it throughout the network. In Equation 5.11 U_T is multiplied with H_{T-1} to determine what information needs to collect from the previous step.

An extension of the GRU, known as the Bi-GRU (Bidirectional Gated Recurrent Unit), is an additional set of hidden states that are generated in the opposite direction. As a result, the model is able to include data from the input sequence's past and future. The Bi-GRU cell Equations are as follows:

$$\overrightarrow{H_T} = GRU_{fwd}(X_T, \overrightarrow{H_{T-1}})\overleftarrow{H_T} = GRU_{bwd}(X_T, \overleftarrow{H_{T+1}})H_T = \overrightarrow{H_T} \oplus \overleftarrow{H_T} \tag{5.12}$$

where $\oplus$ denotes the act of concatenating two vectors, $\overrightarrow{H_T}$ denotes the state of the forward GRU, and $\overleftarrow{H_T}$ denotes the state of the backward GRU.

5.3.5 Classification

Spatio-Temporal Features Fusion

In finger motion, feature extraction features are extracted from the trajectory video formed by detecting the fingertip and tracking the fingertip across the frames. Similarly, in parallel Global motion features are extracted using the 2DCNN. In the global motion pipeline input to the 2DCNN is an optical flow video and frame-wise features

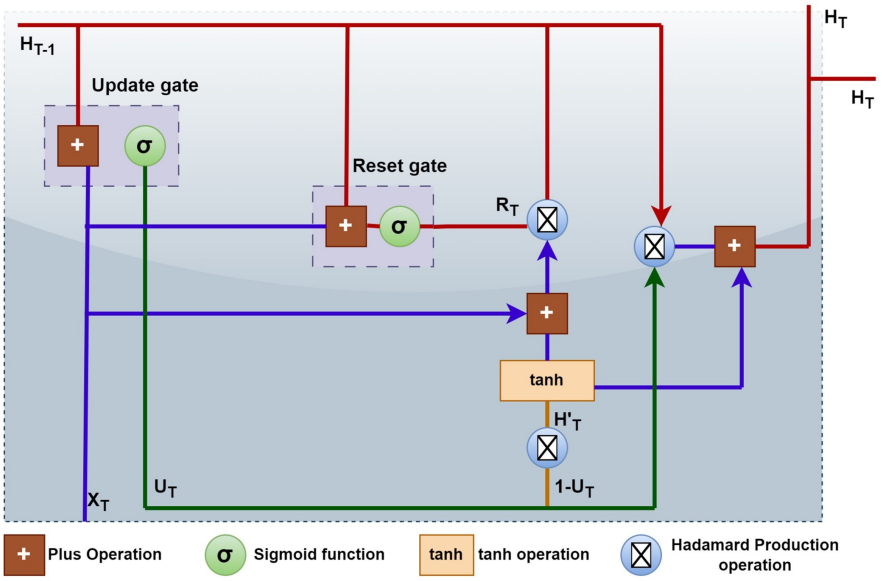


Figure 5-9: GRU architecture

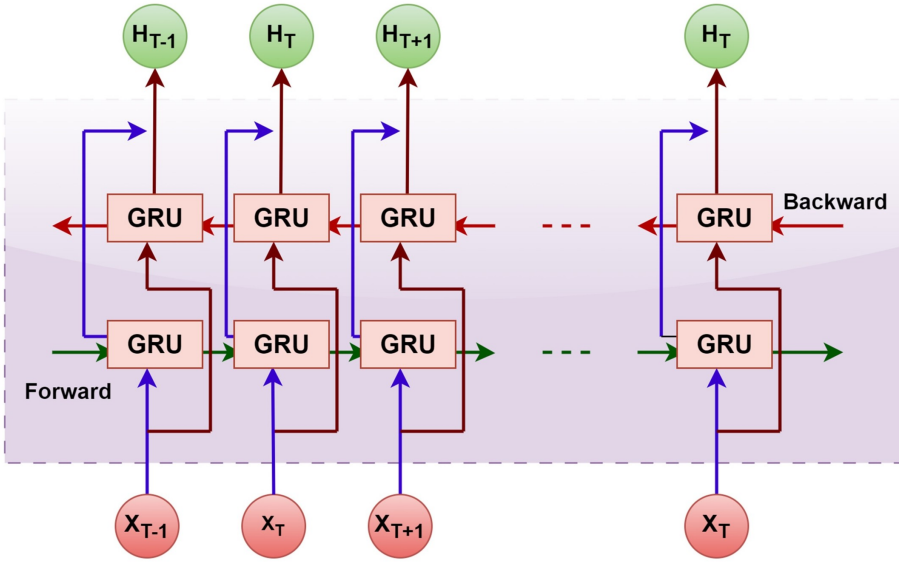


Figure 5-10: Bidirectional-GRU architecture

are calculated. Both pipeline features are passed in the Bi-directional GRU model for sequence-to-sequence learning. After sequence learning, output from the max pooling layer is averagely fused together followed by the fully connected layer and soft-max layer for the final class prediction. We perform the fusion of the features after the max pooling layer called feature level fusion where features from both the Bi-GRU model are averagely fused. Average fusion at the max pooling layer is most frequent and most widely used to fuse the high-level features [109].

$$F_{features} = avg(FMF_{bgru} + GMF_{bgru}) \tag{5.13}$$

$$\hat{Y} = softmax(FC(F_{features})) \tag{5.14}$$

We also used decision-level fusion to evaluate the model's precision. The FC layer receives the outputs from the FMF_{bgru} and GMF_{bgru} separately and uses a softmax classifier to produce the probability distribution of class labels.

$$\hat{Y} = avg(softmax(FC(FMF_{bgru})) + softmax(FC(GMF_{bgru}))) \tag{5.15}$$

Softmax layer output of both the network averagely fused together to get the final prediction as shown in Equation 5.15. Feature level fusion strategy discussed in Equation 5.13 and 5.14 gives better accuracy and followed the same to perform the experiments on the various datasets.

Categorical Cross Entropy

A typical loss function, especially for multi-class classification tasks, is categorical cross-entropy. It calculates the difference between the actual labels and the predicted probability distribution. Each class predicted probabilities, are calculated using a Softmax activation function. In order to ensure that the projected values are both non-negative and add up to 1, Softmax creates a probability distribution over the classes.

$$Y_L = -\sum_{i=1}^{C}(x_{\text{true}}(j) \cdot \log(x_{\text{pred}}(j))), \qquad \text{for C classes} \tag{5.16}$$

In Equation 5.16 the summation is applied to all classes C and Y_L is represented as a cross-entropy loss. The true label is represented by the $x_{\text{true}}(j)$, while the predicted probabilities are represented by $x_{\text{pred}}(j)$ for j^{th} class. Elements of the expected probability are applied individually to the logarithm. It calculates the difference between actual class labels and expected probability. By adding the element-wise product of the true labels and the logarithm of the predicted probabilities for each class, the loss is determined. To reduce the loss during training, the negative sign is applied. The calculated probability distribution tends to resemble the real distribution when the cross entropy loss is kept to a minimum, which increases the classification accuracy of the model.

5.4 Experimental Analysis

5.4.1 Training Details

All the experiments are conducted on intel Core i7 processor with 8GB RAM, and 8GB NVIDIA GETFORCE GTX graphics card. Implementation is done in Tensorflow 2.8 with PyTorch libraries. Adam is applied as an optimization function and categorical cross entropy is used as a loss function. The score of both networks is fused to get the final prediction. Categorical cross-entropy is used to calculate the final prediction loss, and loss is back-propagated in both networks. The batch size is 8 and the model is trained up to 200 epochs. Initially learning rate is 0.0001 and is reduced by a factor of 10 once the learning starts. We perform our experiment on different batch sizes and loss function and experimentally decided batch size 8 and categorical cross entropy as a loss function. We choose the Adam optimizer for our proposed model, it covers high-dimensional parameter spaces and works well with smaller datasets. It adjusts learning rates for each parameter adaptively, making convergence faster and more efficient. The hyper-parameters are shown in Table 5.1

Table 5.1: Hyper parameters of the proposed model.

Name	Details
Batch Size	8
Epochs	200
Learning rate	0.0001
Optimizer	Adam
Loss	Categorical Cross entropy

5.4.2 Experimental Analysis on Different Datasets

Experiments are performed on two different benchmark datasets such as North Western University Hand Gesture (NWUHG) dataset and Dynamic Hand Gesture (DHG-14/28) dataset. Input to our proposed model is an optical flow video and skeleton trajectory plot video. Features extracted from optical flow video are called as global motion features (GMF) that captures the shape of the moving hand. Further features extracted from the skeleton video is called as a finger motion features (FMF) which represent the movement of fingers along with change in the hand shape.

North Western University Hand Gesture Dataset(NWUHG)

NWUHG dataset [110] performed by 15 persons doing 10 dynamic hand gestures in 7 various poses. There are $7 \times 15 = 105$ RGB video sequences available for each move. RGB videos for gestures like 'move right,' 'move left,' 'rotate up,' 'rotate down,' 'move down-right,' 'move right-down,' 'clockwise circle,' 'counter-clockwise circle,' 'Z,' and 'cross' are included in the gesture inventory. Few samples of 'move right' and 'rotate down' gesture is shown in Figure 5-11. The NWUHG contains a total of 1050 samples.

In NWUHG dataset only RGB video is available thus, skeleton videos are calculated using the media pipe library, and optical flow videos are formed using the RGB video only. If in the input data only RGB video is available then skeleton points can be extracted by the media-pipe library and optical flow video can be formed by RGB data. If the data skeleton points is available no need to use the media pipe library to form the skeleton video. Both FMF and GMF features are extracted using pre-trained 2DCNN called as Xception-Net in two parallel pipelines. Then both

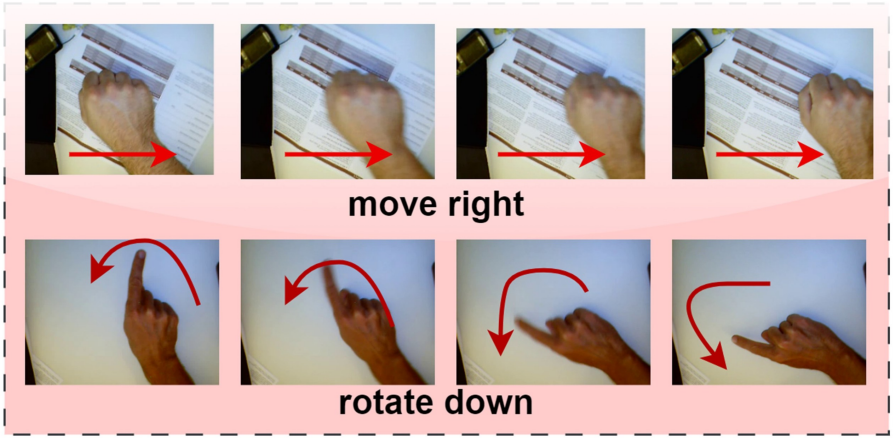


Figure 5-11: North Western University Hand Gesture Dataset(NWUHG)

features are passed to Bi-GRU for sequence-to-sequence learning and just before thee fully connected both pipeline features are averagely fused and then fused features are passed to the FC layer and then SoftMax layer for the final prediction. Figure 5-12

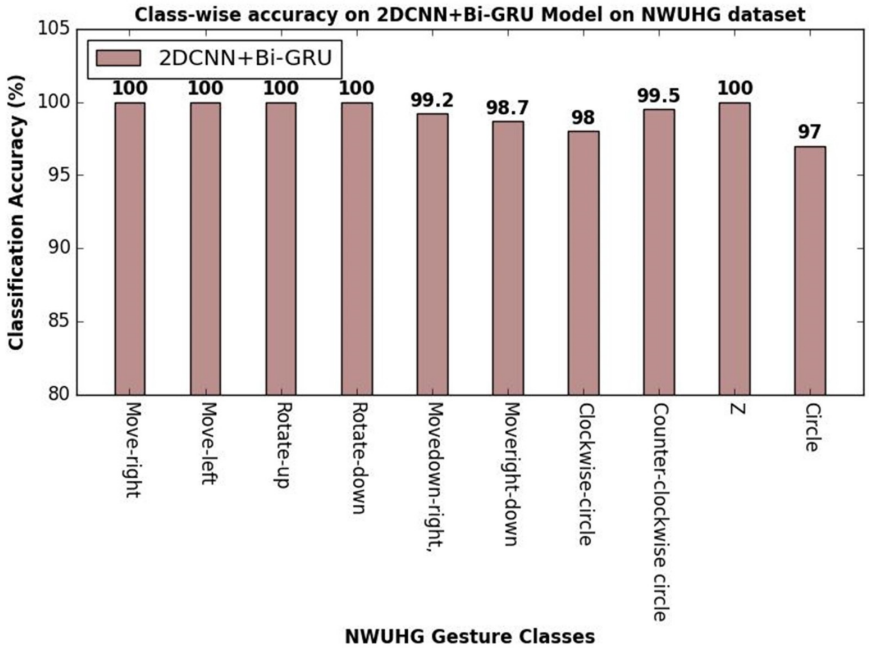


Figure 5-12: Shows class-wise accuracy of Bi-GRU architecture NWUHG dataset. The average accuracy of the proposed model is 99.2%.

show the class-wise accuracy on NWUHG dataset. As we can see that in most of the classes we got 100% accuracy and in few classes less than 100%. This variation is because due to the similar type of motion track we got in few classes and that's

why there is confusion between gesture classes. The average accuracy on NWUHG is around 99.2%.

5.4.3 Experiments on DHG-14/28 Dataset

The DHG14/28[1] [15] contains the depth data and skeleton data of the hand joints captured by Intel Real Sense Short-Range Depth Camera. It includes 2800 gesture sequences having 14 classes of hand gestures. 20 people took part in making gestures using full hand shape and using just one finger. Each participant repeated each gesture five times. The few samples of the 'expand' gesture are shown in Figure 5-13. The skeleton information contains 22 hand joint points which resemble the hand

Figure 5-13: Shows depth images of DHG-14/28 dataset.

skeleton. The gesture inventory contains two categories:'Fine' and 'Coarse' gestures. The 'Fine' gestures are 'Grab,' 'Expand,' 'Pinch,' 'Rotation Clock Wise,' and 'Rotation Counter Wise,' whereas the 'Coarse' gestures are 'Tap', 'Swipe Right,' 'Swipe Left,' 'Swipe Up', 'Swipe Down', 'Swipe X', 'Swipe V', 'Swipe +', and 'Shake'. In DHG-14/28 depth and skeleton points are available. Therefore, the optical flow video is calculated using the depth data, and the skeleton video is formed using the skeleton points. DHG-14/28 gesture contains depth and skeleton data having 14 gesture classes performed by two hand shape: using the whole hand and using only one finger. In DHG-14, gesture performed by the whole hand and a single finger is considered as belonging to the same class, thus it has only 14 classes. Similarly, in DHG-28, a gesture performed by whole hand and a single finger is considered as belongs to the different classes, thus it has total 28 gesture classes. We have reported the class-wise accuracy for 14-gesture classes and 28-gesture classes in the Figure 5-14 and

[1]http://www-rech.telecom-lille.fr/DHGdataset/

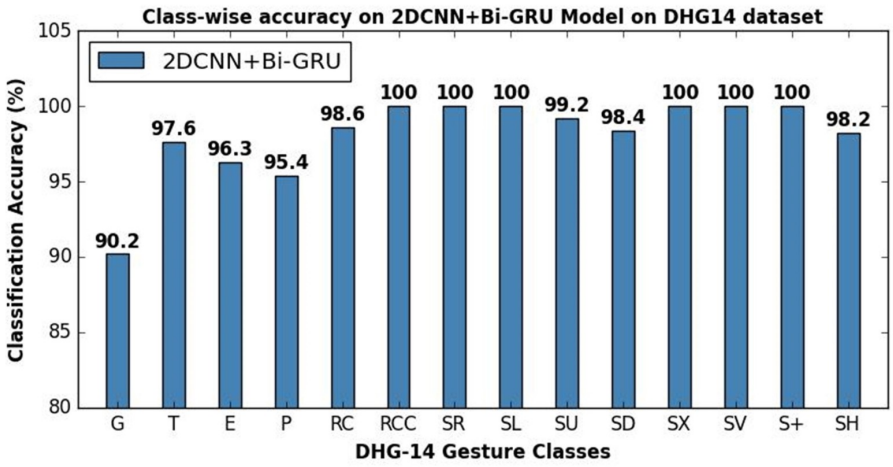


Figure 5-14: Shows class-wise accuracy of Bi-GRU architecture on DHG14 dataset. The average accuracy of the proposed model in 98.2%.

Figure 5-15. In DHG-14 gesture classes we got average accuracy of 98.2% and in DHG-28 gesture classes average accuracy of 94.2%.

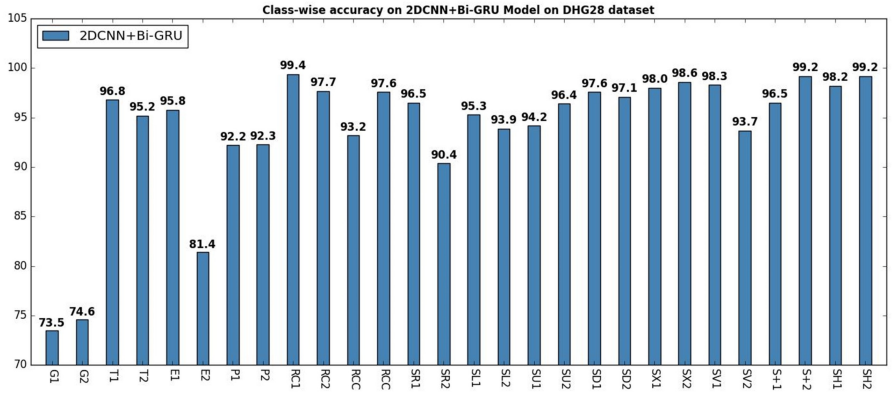


Figure 5-15: Shows class-wise accuracy of Bi-GRU architecture on DHG28 dataset. The average accuracy of the proposed model is 94.2%.

5.4.4 Ablation Study

The success of the various model can't be denied using only RGB data or using only skeleton data but there is a scope to extract complimentary features from different modalities and accuracy can be enhanced as well. Many approaches have shown considerable improvements when using a fusion of features from multiple modalities in dynamic gesture recognition. Specifically, depth maps, normal maps, IR images,

and optical flow images have been used in various feature fusion techniques and have shown varying degrees of improvement. The fusion of other modalities with the skeleton features is still an unexplored area. Gesturing hand detection and tracking is still a challenging task and to avoid this issue skeleton data can be used surpassing hand detection and tracking. Moreover, additional modality creation of optical flow video using RGB or depth data also reduces the challenge of hand detection and capturing the motion of a moving hand. Both modalities helps in the extraction of the spatio-temporal features. We evaluate the proposed two pipeline architecture using the 2DCNN+Bi-LSTM model and the 2DCNN+Bi-GRU model for sequence-to-sequence learning. Experimentally we witness that 2DCNN+Bi-GRU performs better compare to the 2DCNN+Bi-LSTM. Bi-GRU has lesser tensor operation and efficiently faster compare to the LSTM. Bi-GRU performs better when training samples are less. Two deep architecture for sequence learning is trained on various input data such as (i) using only finger motion features (FMF) (ii) using only global motion features (GMF) and fusion of FMF and GMF at feature level fusion and decision level fusion. The deep architecture as shown in Figure 5-1 is trained on the two fusion strategies. Two score fusions are examined using Equation 5.13 and Equation 5.15 feature level fusion and decision level fusion respectively. Both fusion strategies fuse the finger motion and global motion features and results are shown in Table 5.2 and decision level fusion is shown in Table 5.3. Equation 5.13 and Equation 5.15 show two fusion

Deep Architecture	**Modality**	**Accuracy**
Bi-LSTM	Finger Motion Features (FMF)	93.5
Bi-LSTM	Global Motion Feature (GMF)	95.2
Bi-LSTM (Feature)	FMF + GMF	97.2
Bi-GRU	Finger Motion Features (FMF)	96.4
Bi-GRU	Global Motion Feature (GMF)	98.3
Bi-GRU(Feature)	FMF + GMF	99.2

Table 5.2: Performance of two different architecture using different modality on Northwestern University Hand Gesture dataset with feature level fusion.

strategies: feature-level fusion and decision-level fusion respectively. One is to fuse the spatio-temporal feature fusion for the classification and the other one is to use the prediction score fusion using average. The accuracy 97.2% and 99.2% shows the

Deep Architecture	Modality	Accuracy
Bi-LSTM	Finger Motion Features (FMF)	93.3
Bi-LSTM	Global Motion Feature (GMF)	95.2
Bi-LSTM(Decision)	FMF + GMF	96.7
Bi-GRU	Finger Motion Features (FMF)	95.8
Bi-GRU	Global Motion Feature (GMF)	96.2
Bi-GRU(Decision)	FMF + GMF	98.4

Table 5.3: Performance of two different architecture using different modality on Northwestern University Hand Gesture dataset with decision level fusion.

superiority of spatio-temporal feature fusion over decision-level fusion using Bi-LSTM and Bi-GRU respectively. Apart from performance feature-level fusion also reduces the computation cost of the fully connected layer compared to the twice computation of the fully connected layer at the decision level. Thus, early fusion at the feature level fusion is superior to the prediction level fusion. From the experiments, we observed that feature-level fusion with Bi-GRU architecture performs best and followed the same experimental analysis pattern on all the other datasets. Figure 5-16 shows the

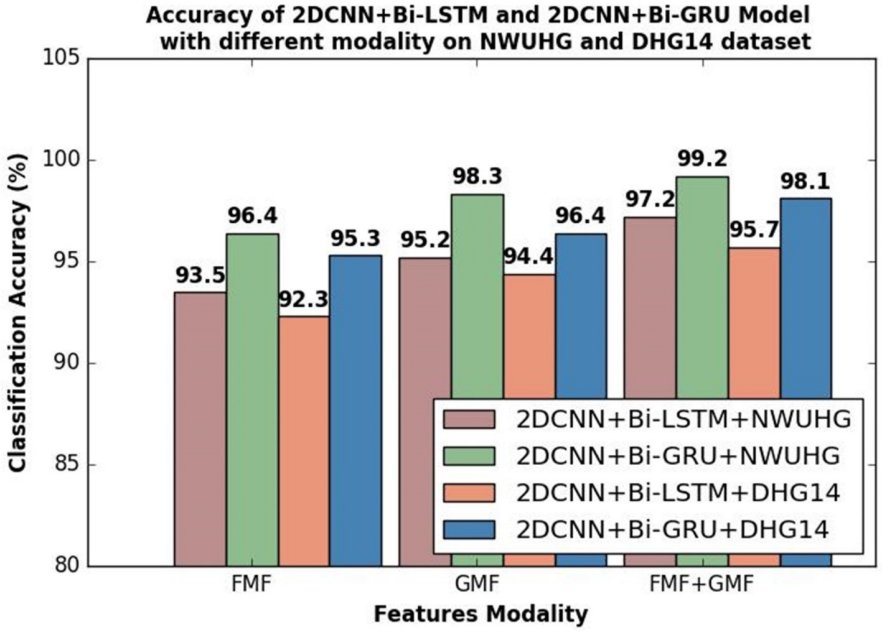


Figure 5-16: Shows accuracy of Bi-LSTM and Bi-GRU architecture with different features on NWUHG and DHG14 dataset.

accuracy of different architecture with different feature modality on various dataset. As we can see that accuracy using 2DCNN + Bi-GRU on NWUHG dataset is superior with feature fusion than a single modality i.e 99.2%. Similarly, accuracy using 2DCNN

+ Bi-GRU on DHG14 dataset is superior with feature fusion than single modality features i.e 98.1%

5.4.5 Comparison with State-of-the-art

On the DHG14/28 and NWUHG data sets, we compare experimental results to the state-of-the-art. Discussion and analysis of the findings are given below:

Comparison with State-of-the-Art-Methods on DHG14/28 Dataset

The proposed model is compared in Table 5.4 with existing state-of-the-art methods. The accuracy rate for the spatial-temporal synchronous transformer(STST) technique [111] are 97.6% and 95.8%. In this technique, a two-stage network focuses on employing transformers to capture spatial-temporal correlations in hand gesture skeleton sequences (HGSSs). In paper [60] proposed self-attention graph convolutional network (SAGCN) + residual bidirectional(RBi) - independently recurrent neural network(IndRNN) for extracting the temporal information over the long and short term features. Similarly, in paper [64] the author proposes DSTA-Net for skeleton-based gesture recognition. Some other papers [65], [100], [11], [34] use deep learning approaches for hand gesture recognition and the implemented data type is either skeleton, RGB, depth, or fusion of RGB and depth data, but in our proposed model we used a fusion of optical flow and skeleton data types. Which overcomes problems like occlusion, illumination, and complex background captures the hand motion, and discards the stationary background. Thus as compared to other SOTA methods we obtained a higher recognition rate except [111]. In this paper [111], the author proposed a technique to encode hand gesture skeleton sequences by dividing them into chunks. These chunks help capture joint relationships over time. The encoded features are processed by transformer modules to enhance the understanding of both local and global spatial-temporal information in the sequences. The results listed in Table 5.4 show that the proposed XceptionNet+Bi-GRU outperforms other state-of-the-art methods on DHG-14/28 dataset achieved 98.1% accuracy, except in

paper [111] for DHG-28 gesture and the reason behind this is our model get confused on identical gestures made with two different hand shape.

Table 5.4: DHG14/28 dataset's classification accuracy comparison(%) results.

Methods	**14 G(%)**	**28 G(%)**
DG-STA [34]	91.9	88.0
ST-TS-HGR-NET [11]	94.2	89.4
DeepGRU [100]	94.5	91.4
HPEV-Net+HMM-Net [65]	92.5	88.8
DSTA-Net [64]	93.8	90.9
CNN+BGRU+RGB+OFV [101]	97.8	92.1
SAGCN+RBi-IndRNN [60]	96.3	94.0
STST [111]	97.6	95.8
LSTSN [112]	95.6	92.2
XceptionNet+Bi-GRU	**98.2**	**94.2**

Comparison with State-of-the-Art-Methods on North Western University Hand Gesture (NWUHG) Dataset

We conduct an experimental assessment with state-of-the-art on the NWUHG dataset, and the findings are mentioned in Table 5.5. The accuracy rate for the key frames+feature fusion [20] is 97.9%. In this paper, the author proposed a new approach for key frame extraction combining image entropy and density clustering. In paper [96] the author proposed Key-frames splicing + feature fusion techniques for the dynamic method of hand gesture recognition and obtained a 97.6% recognition rate. Some paper based on a traditional method of hand gesture like [113] uses a genetic algorithm, to combine machine-learned spatio-temporal descriptors for gesture detection and paper [110] worked on Motion divergence fields. While in paper [95] the author proposed $AlexNet^2$ features for transfer learning-based extraction of spatial-temporal features. Similarly in a paper, [101] the author proposed CNN+BGRU+RGB+OFV techniques for deep learning-based hand gesture recognition. The author find optical videos from RGB and combine RGB and optical flow video capabilities for gesture classification and obtained 98.6% accuracy on the northwestern hand gesture dataset. In comparison with these papers [110], [20], [113], our proposed method performs better because they used hand tracking and feature extraction which itself is a challenging due to various illumination condition and complex background. Similarly

from [96], [101], and [95] papers our proposed method performed well and obtained good accuracy due to the fusion of optical flow and skeleton features Which overcomes problems like occlusion, illumination, and complex background captures the hand motion, and discards the stationary background. For the NWUNG dataset, we compare our proposed model with the state-of-the-art methods and show that our proposed XceptionNet+Bi-GRU model outperforms the literature and achieved 99.2% accuracy.

Table 5.5: NWUHG dataset's classification accuracy comparison(%) results.

Methods	**Accuracy(%)**
Motion divergence fields [110]	95.8
Key frames + Feature fusion [20]	97.9
Key frames splicing + feature fusion [96]	97.6
CNN+BGRU+RGB+OFV [101]	98.6
AlexNet^2 [95]	96.9
Genetic programming [113]	96.1
XceptionNet+Bi-GRU	**99.2**

5.5 Conclusion

Our proposed model offers a bidirectional gated recurrent unit (Bi-GRU) model-based hand gesture recognition system that is computationally effective than Bi-LSTM. This method is designed to attain high-speed performance while being capable of working successfully even with limited training samples. The advantage of our proposed model is that it overcomes the challenges of hand occlusion, illumination, and complex background and it captures the hand gestures and discards the stationary background. The creation and integration of optical flow video and skeleton trajectory video features increases the accuracy of the model and proposed framework can be used in real-time applications. Experiments on two benchmark datasets having accuracy more than 98% on the DHG-14 gesture and more than 94% on the DHG-28 gesture and more than 99% accuracy on NWUHG dataset outperforming with state-of-art methods.

Our proposed model offers a bidirectional gated recurrent unit (Bi-GRU) mod-

elbased hand gesture recognition system that is computationally effective than Bi-LSTM. This method is designed to attain high-speed performance while being capable of working successfully even with limited training samples. This dual-feature extraction method allows the model to achieve a more robust understanding of hand gestures, improving overall performance in diverse environments.

Chapter 6

Hybrid Framework for Dynamic Hand Gesture Recognition using Multiple Modalities

Reena Tripathi, and Bindu Verma. "Tri-Modal Fusion for Dynamic Hand Gesture Recognition: Integrating RGB, Depth, and Skeleton Data" is communicated in *Journal of Visual Communication and Image Representation* (SCIE Indexed, IF: 2.6) *(Communicated)*

6.1 Introduction

In this chapter, we introduce a multimodal fusion approach in our proposed work, leveraging the unique advantages of each modality. Multi-modal fusion enhances the model's ability to generalize across different environments, lighting conditions, and hand orientations, leading to better performance in real-world scenarios. In previous Chapter 5, we proposed a hybrid multi-modal fusion network that combined skeleton and optical flow features. As we have seen in previous chapter using dual modality features boost the performance of the modal. RGB and skeleton data helps in to extract the visual and geometrical features respectively and combining these features boost the performance of the model. Further, as depth data less affected by the

illumination and occlusion gives a most discriminating features. Thus, using RGB, depth, and skeletal data together creates a more reliable gesture recognition framework. When multiple modalities are combined, accuracy often improves compared to using a single modality. This is so that the model may use a variety of information sources to help it make a better recognition.

The primary goal of this work is to present a general framework that can accurately identify dynamic hand gestures without the need for hand gesture detection and tracking. We proposed a fusion of three modalities (RGB, Depth, and Skeleton). RGB data contributes various insights such as the color of the image and spatial characteristics of hand gestures, encompassing hand shape, texture, and appearance. On the other hand, depth data provides depth information that is crucial for understanding the spatial configuration of hand gestures. Nevertheless, RGB data may encounter challenges such as illumination variations, occlusion, and background clutter, which can hinder hand gesture recognition. In contrast, skeleton data overcome these challenges. By integrating these modalities, we achieve more robust gesture recognition system, that enhances model accuracy. The proposed algorithm starts with inputs from RGB, Depth, and Skeleton data. Since it is a, sequential data, first, we extract the features using the CLIP model individually, then pass these features through the residual block. Input layers for each modality are processed through two Conv1D layers and LSTM layer. Outputs from all modalities are concatenated for hand gesture classification. To classify the dynamic hand gesture, this paper used a LSTM network for sequential learning using the Softmax function with entropy loss. The features of all modalities RGB, depth, and skeleton are passed for sequence-to-sequence learning.

6.2 Literature Survey

In the literature many author proposed deep learning based framework using single and multiple modalities. The author Chen et al. [25] used single modality RGB data with short-term and long term features to classify the dynamic hand gesture. Similarly, author [26] also focuses on a single RGB modality as an input and finds

the spatio-temporal features to classify the dynamic hand gesture using deep learning model.

The author Kankana et al. [45] used RGB and depth data with sparse low-rank scores for hand action recognition, it includes four main modules including CNN and RNN, that address frame level and video level classification. Similarly, Yang et al. [114] and Yilin et al. [39] fused skeleton and RGB modalities, and used transformer model to classify the dynamic hand gesture. Verma et al. [101] used RGB and depth data where optical flow video is calculated for both modalities and 2DCNN with GRU model for dynamic hand gesture recognition. Dexu et al. [32] proposed a multimodal gesture recognition technique that merges densely connected convolutional networks (DenseNet) with bidirectional long short-term memory (BLSTM) networks. This approach combines RGB and depth features to classify the dynamic hand gesture. The author Tang et al. [115] proposed an architecture that combines two networks, the ResC3D network and Convolutional LSTM. They use RGB and depth data for a dynamic selection method called Selective Spatio-temporal feature learning (SeST) to classify the gesture. By overcoming the problem of occlusion different lighting condition, and cluttered background, skeleton-based recognition has made significant progress in the field of dynamic gesture detection [116]. To identify dynamic hand gestures, the authors Zhou et al. [12], Nguyen et al. [11] and Alberto et al. [117] recognize dynamic hand gesture recognition using manifold learning. Skeletal data is used to represent hand joints, and a Gaussian aggregation network is used to encode the spatial and temporal relationships between hand joints. The author Liu et al. [65] proposed an end-to-end two-stream network that uses a 2D CNN for hand movement features and a 3D CNN for hand posture development to learn from these components.

The author Sheng et al. [67] proposed an effective Graph Convolutional Network (GCN) model for dynamic hand gesture recognition using skeleton data. Li et al. [84] proposed the MVHANet method for single-hand gesture recognition by finding a suitable distribution of angles in skeleton data, and Xuan et al. [11] proposed a model that learns a dis-criminative SPD matrix encoding the first and second-order statistics for skeleton-based hand gesture recognition. Liu et al. [65] proposed the

Hand Posture Evolution Volume (HPEV) model that uses 2DCNN in one stream and 3DCNN for Hand Movement Map (HMM). Similarly, the author Satya et al. [92] uses 1DCNN and 2DCNN in a multi-scale model for feature extraction using skeleton data for dynamic hand gesture recognition.

Inspired from the literature that combination of modalities shows a good performance compare to the single modality. Thus, in our proposed model we have used all three modalities together to classify the dynamic hand gesture that overcome the challenge of illumination, occlusion, and background clutter.

6.3 Proposed Architecture

The proposed model architecture is shown in the Figure. 6-1. The proposed model

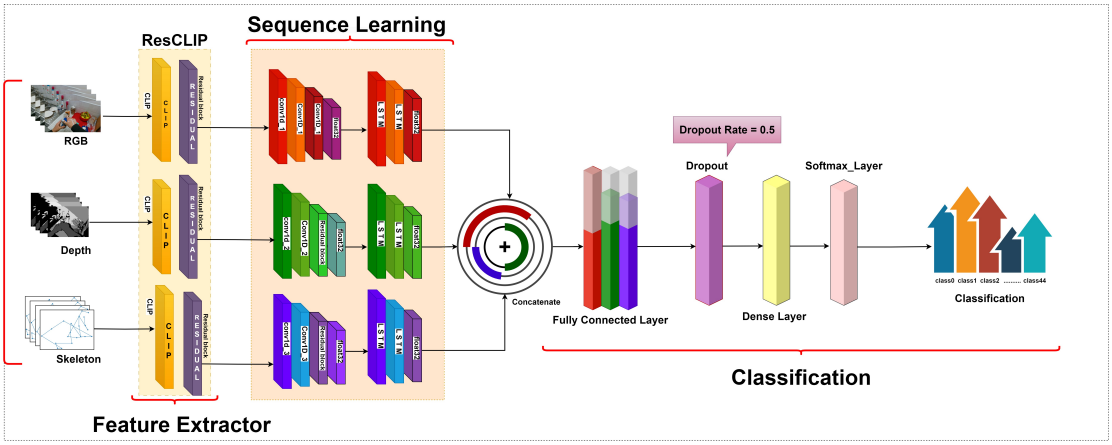


Figure 6-1: Shows a three-stream pipe-lined 1D CNN with LSTM for sequential learning. First, using ResCLIP, the features are extracted from each video frame: the RGB videos, depth videos, and the skeleton point plot video sequences. Second, for sequential learning, the input layer of the extracted features is fed into two 1D CNNs and LSTM. The features are fused from all three pipelines and Concatenated at the FC layer, and SoftMax with cross-entropy loss is used to obtain the final prediction.

begins by taking RGB, Depth, and Skeleton data as an input. It then proceeds with feature extraction, defining the total number of features and generating an output matrix representing feature vectors for each sample. The CLIP model is combined with the residual block for feature extraction of sequential data. First, features are extracted using the CLIP model individually explained in Section 6.3.1, and the

features are passed through the residual block. Residual blocks allows more effective feature learning by mitigating the vanishing gradient problem and allowing networks to converge faster by focusing on learning residuals.

In our pipeline, features are extracted individually from each modality using the CLIP model. These features are then refined through a residual block before being further processed for sequential learning, ensuring a robust and comprehensive feature representation. The sequential learning model followed by the processing of each modality through two Conv1D layers given in Section 6.3.2 and an LSTM layer. The Conv1D layers apply a kernel size of 3x3 with the same number of filters, and padding is set to "same" to maintain input shape. After processing, the outputs from all modalities are concatenated into a single layer, set to 0.5 dropout regularization to avoid over-fitting. Finally, a Dense layer with SoftMax activation predicts the class probabilities. The detailed steps of the proposed model are outlined in the Algorithm. 6.1.

6.3.1 Feature Extractor: ResCLIP

The CLIP (Contrastive Language-Image Pre-training) model [118] is a machine learning development that has attracted a lot of interest in the artificial intelligence community because of its amazing capacity to understand the complex relationships within images. We used CLIP image encoder to extract the features from each frame of a video.The CLIP model is shown in Figure 6-2. CLIP feature extractor extracts the feature of each video and stored it into a 1-D vector of dimension 512. Let T_f represent the feature vector of 1 video as shown in Equation 6.1. If there are 'n' number of videos in one class, the size of the feature vector matrix for one class will be in Equation 6.2. Like-wise next class feature vector matrix will be appended. In last for a dataset having 'C' classes the feature vector matrix will be $(C \times n) \times T_F$. The total no of features represented by T_F will be

$$T_F = F_0, F_1, F_2, F_3.....F_{511} \tag{6.1}$$

$$EF = Fr_i \times T_F \qquad \forall i = 1, 2, 3...m \tag{6.2}$$

Where, the features matrix for one class will be represented by 'EF'. Fr_i represented gestures of class C_m.

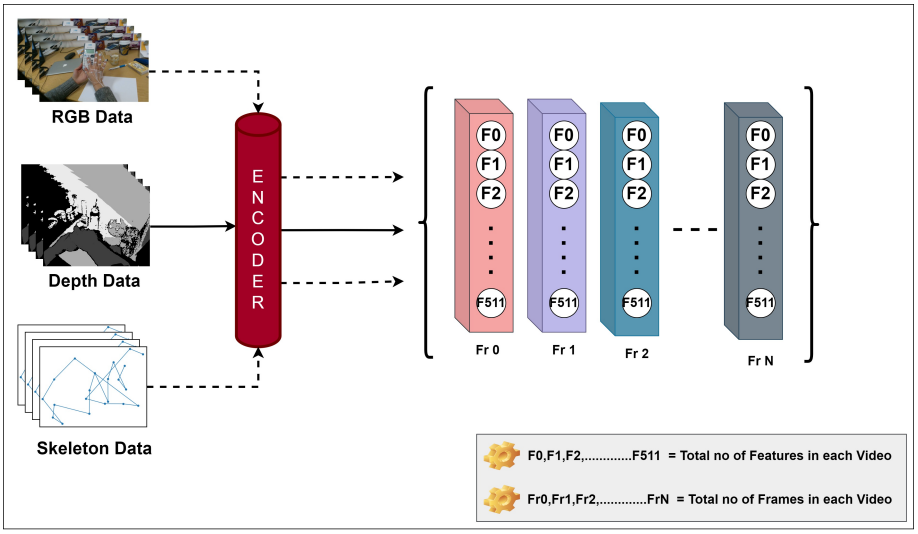


Figure 6-2: CLIP(Contrastive Language-Image Pre-training) model.

$$Output_{(C\times n)\times T_F} = \begin{bmatrix} C_1Fr_1\{F_0 & F_1 & \cdots & F_{511}\} \\ C_1Fr_2\{F_0 & F_1 & \cdots & F_{511}\} \\ \vdots & \vdots & \ddots & \vdots \\ C_1Fr_n\{F_0 & F_1 & \cdots & F_{511}\} \\ C_2Fr_1\{F_0 & F_1 & \cdots & F_{511}\} \\ C_2Fr_2\{F_0 & F_1 & \cdots & F_{511}\} \\ \vdots & \vdots & \ddots & \vdots \\ C_mFr_n\{F_0 & F_1 & \cdots & F_{511}\} \end{bmatrix} \tag{6.3}$$

The Feature matrix for one complete dataset is represented by the $Output_{(C\times n)\times T_f}$. Where, C_1 represents class 1, Fr_1 represents gesture 1 of class C_1, and $\{F_0, F_1, \ldots, F_{511}\}$ is the feature matrix of video 1 of class 1, and so on.

RGB Features Extraction

In the proposed model the RGB features are extracted using the CLIP model and fed into the proposed model via residual block, as shown in Figure 6-3. This modality has several benefits it gives color information about the gesturing hand, including specifics like clothing, skin tone, and objects used in the gesture. RGB data also provides spatial information that helps interpret the shape of the gesturing hand, its texture, and other relevant features of the hand. Overall, the RGB data helps the model better understand the environment where hand gestures happen.

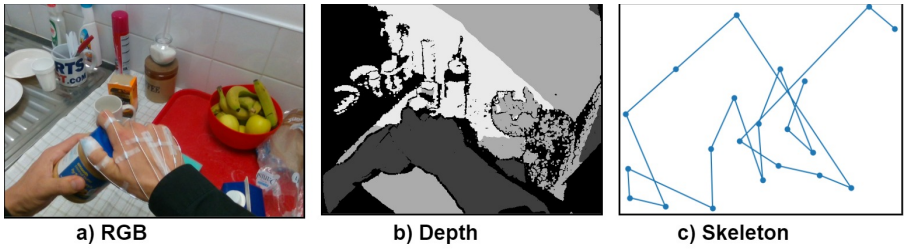


Figure 6-3: FPHA dataset: capturing hand gesture through RGB, depth, and skeleton modalities for comprehensive gesture recognition

Depth Features Extraction

In the proposed model, the depth features are extracted using the CLIP model and fed into the proposed model via residual block, as shown in Figure 6-3. There are distinct benefits associated with the second modality, which is sequential depth data. Each frame's depth information gives the model the ability to understand the gesture's spatial configuration in three dimensions. Depth data is more dependable under a range of lighting situations because it is less affected by illumination than RGB data. Additionally, depth data mitigates issues related to occlusion since it directly captures physical object distances from the sensor, bypassing potential obstructions.

Skeleton Features Extraction

In the proposed model, the skeleton features are extracted using the CLIP model and fed into the proposed model via residual block, as shown in Figure 6-3. It makes it easier to precisely track hand gesture movements by giving information about the numerous hand skeleton joints. Because skeleton data is not dependent on background information, it overcomes the challenges of dynamic hand gesture recognition like occlusion, cluttered backgrounds, and changing illumination.

The proposed model can take advantage of RGB, depth, and skeletal data, leading to more reliable gesture recognition framework. When multiple modalities are used together, accuracy is frequently increased as compared to a single modality. This is so that the model may use a variety of information sources to help it make a better selection. Multi-modal fusion enhances the model's ability to generalize across different environments, lighting conditions, and hand orientations, leading to better performance in real-world scenarios.

Residual Block

In our proposed model, we combine the CLIP model with residual block [119] for feature extraction of sequential data. First, we extract features using the CLIP model, and then we pass these features through the residual block. As a result, CLIP can capture more information from video data. The residual block enables deeper networks and enhances and refines the extracted features from CLIP. It also learns more robust features that are less sensitive to noise and variations in input data. Since we use CLIP and residual block together, we call it ResCLIP.

6.3.2 Sequential Learning: LSTM

Since LSTM networks are designed to solve the vanishing gradient and exploding gradient problem, recurrent neural networks have trouble being dependable over the long term. An LSTM recurrent unit uses "gates" with different activation functions to "remember" important past information and "forget" irrelevant data. It also maintains

an Internal Cell State, which holds the information from previous LSTM units. The framework of an LSTM unit is composed of the input gate, output gate, forget gate, and cell gate, which are responsible for controlling the learning process, as shown in Figure 6-4. Sigmoid functions are essential for control the functioning of the gates throughout the learning process. The cell state represents the long-term memory in the LSTM and regulates which data from earlier periods will be saved in an LSTM cell. The cell gate is modified by the forget gate, whose output determines whether the information in the cell state should be retained (if it is 1) or forgotten (if it is 0) [72]. The following equations illustrate how LSTM works in our model.

$$u_t = \sigma(P_t w_{xu} + Q_{t-1} w_{Qu} + b_{t-1} w_{bu} + w_{ubias}) \tag{6.4}$$

Where, "u_t" represents the input gate at time step "t". 'P_t" is the input vector and "w_{xu}" is its weight matrix. " Q_{t-1}" is the hidden state from the previous time step with "w_{Qu}". "b_{t-1}" and "w_{bu}" are the bias term and its weight matrix, respectively. "w_{ubias}" is an additional bias term for the input gate.

$$r_t = \sigma(P_t w_{xr} + h_{t-1} w_{Qr} + b_{t-1} w_{br} + w_{rbias}) \tag{6.5}$$

Where, "r_t" represents the output gate. "h_{t-1}", another notation for the hidden state from the previous time step, is similar to "Q_{t-1}"."w_{Qr}" is the weight matrix for the hidden state to the forget gate, and "w_{br}" is the weight matrix for the bias to the forget gate.

$$B_t = tanH(P_t w_{PB} + h_{t-1} w_{QB} + w_{zbias}) \tag{6.6}$$

Where, "B_t" represents the candidate cell state. "w_{QB}" is the weight matrix for the hidden state to the cell state and "w_{zbias}" is the bias term. In Equation 6.7 the "b_t" represents the cell state.

$$b_t = B_t \otimes i_t + b_{t-1} \otimes i_t \tag{6.7}$$

$$v_t = \sigma(P_t w_{xv} + Q_{t-1} w_{Qv} + b_{t-1} w_{bv} + w_{obias}) \tag{6.8}$$

Where, "v_t" represents Output gate. Controls the output from the cell state.

$$Q_t = v_t + tanH(b_t) \tag{6.9}$$

Where,"Q_t" represents the hidden state.

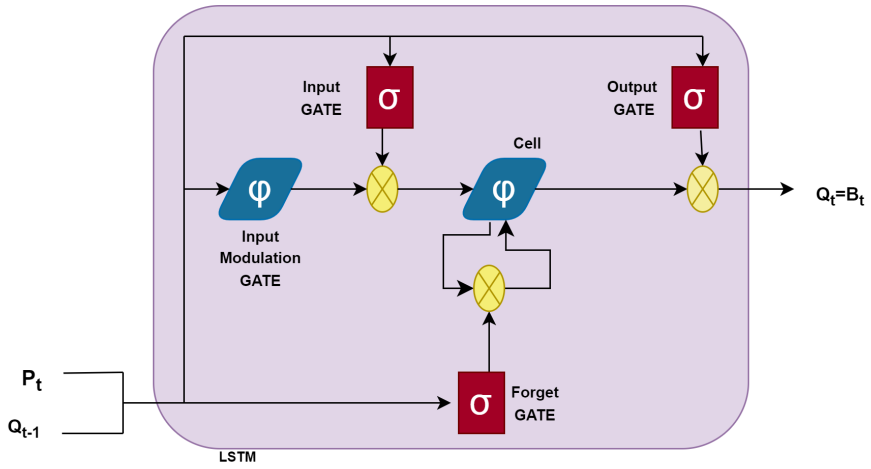


Figure 6-4: Block diagram of the long shot term memory (LSTM) model architecture.

6.3.3 Concatenation of Spatio-Temporal Features and Classification

The features of all three modalities are passed to the proposed model for sequence-to-sequence learning. The max pooling layer's output is averaged for the final class prediction after sequence learning, followed by the FC layer and softmax layer. Feature fusion is performed after the max pooling layer, a process known as feature-level fusion, where features from all three modalities are concatenated averagely. The most common and extensively utilized method for fusing the high-level features is average fusion at the max pooling layer.

$$F_{features} = \text{avg}(RGB_{ResCLIPrnn} + Depth_{ResCLIPrnn} + Skeleton_{ResCLIPrnn}) \tag{6.10}$$

$$\hat{Y} = softmax(FC(F_{features})) \tag{6.11}$$

To assess the accuracy of the model, decision-level fusion was also employed. The FC layer employs a softmax classifier to generate the probability distribution of class labels after receiving the outputs from the $RGB_{ResCLIPrnn}$, $Depth_{ResCLIPrnn}$ and $Skeleton_{ResCLIPrnn}$ independently.

$$\begin{aligned} \hat{Y} = avg(&softmax(FC(RGB_{ResCLIPrnn}))+ \\ &softmax(Depth_{ResCLIPrnn})+ \\ &softmax(Skeleton_{ResCLIPrnn}))) \end{aligned} \tag{6.12}$$

The softmax layer outputs of both networks are fused together by averaging to obtain the final prediction, as depicted in Equation 6.12. The feature-level fusion strategy discussed in Equations 6.10 and 6.11 yields improved accuracy, and we followed the same approach in conducting experiments on FPHA datasets. The proposed model employs a categorical cross-entropy loss function, as illustrated in Equation 6.13.

$$Y_{Loss} = -\sum_{I=1}^{q}(x_{\text{true}}(J) \cdot \log(x_{\text{pred}}(J))), \quad \text{for “}C\text{” classes} \tag{6.13}$$

In Equation 6.13, the summation is applied to all classes C and Y_{Loss} denotes the cross-entropy loss. Here, $x_{\text{true}}(J)$ represents the true label for the J^{th} class, while $x_{\text{pred}}(J)$ represents the predicted probabilities.

6.4 Experimental Analysis

6.4.1 Training Details

An Intel Core i7 processor with 8GB of RAM and an 8GB NVIDIA GeForce GTX graphics card was used to perform the experiments. TensorFlow 2.8 was used for the implementation, along with Keras libraries. Adam served as the optimization function, while the categorical cross-entropy was utilized as the loss function. The scores

from the three modalities are concatenated to form the final prediction. Categorical cross-entropy is employed to compute the loss for the final prediction, and this loss is back-propagated across all modalities. With a batch size of 64, the model undergoes training up to 750 epochs. We began with an initial learning rate of 0.0001 as shown in Table 6.1. Our experimentation encompassed various batch sizes, loss functions, and Adam as an optimizer, ultimately selecting a batch size of 64 and categorical cross-entropy as the loss function based on experimental results. In the model we chose the Adam optimizer because it is good at negotiating high-dimensional parameter spaces, especially with fewer datasets. It increases overall efficiency by accelerating convergence through adaptively adjusting the learning rates for each parameter.

Table 6.1: Training details of the proposed model.

S.no	Training Details	Values
1	Dropout	0.5
2	Optimizer	Adam
3	NOP LSTM	1.3M
4	NOP GPU	1.0M
5	NOP RNN	0.95M
6	NOP Residual block	1.47
7	Learning rate	0.0001
8	Batch Size	64(FPHA) & 32(SKIG)
9	Loss Function	Categorical Cross Entropy

Where, NOP = Number of Parameter

6.4.2 Experimental Analysis on Different Datasets

The experiments are performed on two benchmark datasets such as the First-Person Hand Action (FPHA) [120] dataset and the Sheffield Kinect Gesture (SKIG) dataset [121].

Input to our proposed model is RGB videos, Depth videos, and skeleton trajectory plot videos. The ResCLIP model is used to extract the features from RGB videos, depth videos, and skeleton trajectory videos. If skeleton videos are not available, we have extracted them using a media pipe as shown in Figure 6-5.

Algorithm 6.1 The proposed model's algorithm

Input: RGB data, Depth data, Skeleton data
Feature Extraction:
  Let the total number of features represented by $T_F = F_0, F_1, F_2, \ldots, F_{511}$
  The feature embedding size EF is defined as: $E \times F = Fr_i \times T_F$
  Output matrix:

$$Output_{(C\times n)\times T_f} = \begin{bmatrix} C_1Fr_1\{F_0 & F_1 & \cdots & F_{511}\} \\ C_1Fr_2\{F_0 & F_1 & \cdots & F_{511}\} \\ \vdots & \vdots & \ddots & \vdots \\ C_1Fr_n\{F_0 & F_1 & \cdots & F_{511}\} \\ C_2Fr_1\{F_0 & F_1 & \cdots & F_{511}\} \\ C_2Fr_2\{F_0 & F_1 & \cdots & F_{511}\} \\ \vdots & \vdots & \ddots & \vdots \\ C_mFr_n\{F_0 & F_1 & \cdots & F_{511}\} \end{bmatrix}$$

Pipeline:
  Step 1: Define Input Layers
    $input\_layer\_rgb \leftarrow Input(shape = input\_shape\_rgb)$
    $input\_layer\_depth \leftarrow Input(shape = input\_shape\_depth)$
    $input\_layer\_skeleton \leftarrow Input(shape = input\_shape\_skeleton)$
  Step 2: Process Each Modality
function PROCESS_MODALITY(in_layer)
    $x \leftarrow Conv1D(kernel\_size = 3, padding =' same')(in\_layer)$
    $residual \leftarrow x$
    $x \leftarrow Conv1D(kernel\_size = 3, padding =' same')(x)$
    $x \leftarrow Add()([x, residual])$
    return LSTM(col_hidden)(x)
end function
  $processed\_rgb \leftarrow process\_modality(input\_layer\_rgb)$
  $processed\_depth \leftarrow process\_modality(input\_layer\_depth)$
  $processed\_skeleton \leftarrow process\_modality(input\_layer\_skeleton)$
  Step 3: Concatenation and Output Layer
  $concatenated\_layers \leftarrow Concatenate()([processed\_rgb, processed\_depth, processed\_skeleton])$
  $concatenated\_layers \leftarrow Dropout(0.5)(concatenated\_layers)$
  $output\_layer \leftarrow Dense(num\_classes, activation =' softmax')(concatenated\_layers)$
  Step 4: Compile the Model
  Step 5: Train the Model
  Step 6: Evaluate the Model
  Step 7: Plot Results

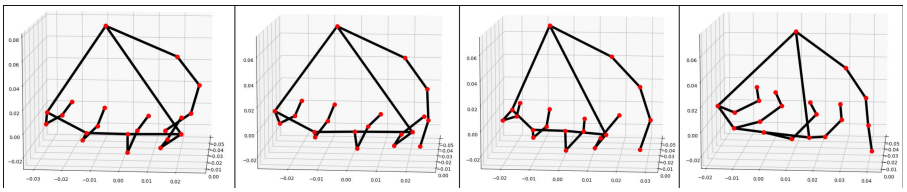


Figure 6-5: This image shows the skeleton of a person performing a wave gesture from the SKIG dataset. The skeletal structure is retrieved using Mediapipe. The red dots indicate key joint positions, connected by black lines to illustrate the gesture of the hand

Experiments on FPHA Dataset

FPHA[1] (First-person hand action) [120] The dataset consists of 1175 action videos with depth and RGB data and skeleton joints. There are forty-five action videos in all, with six actors acting out three different scenarios. These hand actions encompass activities in the kitchen, office, and social settings, involving various objects like book pages, wallet, juice bottle, “liquid soap bottle", “pour milk" and more, as shown in Figure 6-6. The lengths of the videos range from 7 to 1151 frames, with 21 skeletal joints captured in each frame. More details about the data set are given in the paper [120].

The FPHA dataset contains a total of 45 classes of gestures sequence, In this paper 0 class represents “charge_cell_phone", 1 class represents “clean_glasses" and so on. The names of all the classes are given below: “charge cell phone", “clean glasses", “close juice bottle", “close liquid soap", “close milk", “close peanut butter", “drink mug", “flip pages", “flip sponge", “give card", “give coin", “handshake", “high five", “light candle", “open juice bottle", “open letter", “open liquid soap", “open milk", “open peanut butter", “open soda can", “open wallet", “pour juice bottle", “pour liquid soap", “pour milk", “pour wine", “prick", “put salt", “put sugar", “put tea bag", “read letter", “receive coin", “scoop spoon", “scratch sponge", “sprinkle", “squeeze paper", “squeeze sponge", “stir", “take letter from envelope", “tear paper", “toast wine", “unfold glasses", “use calculator", “use flash", “wash sponge", “write".

Figure 6-7 displays a confusion matrix of the proposed model on the FPHA

[1]https://guiggh.github.io/publications/first-person-hands/

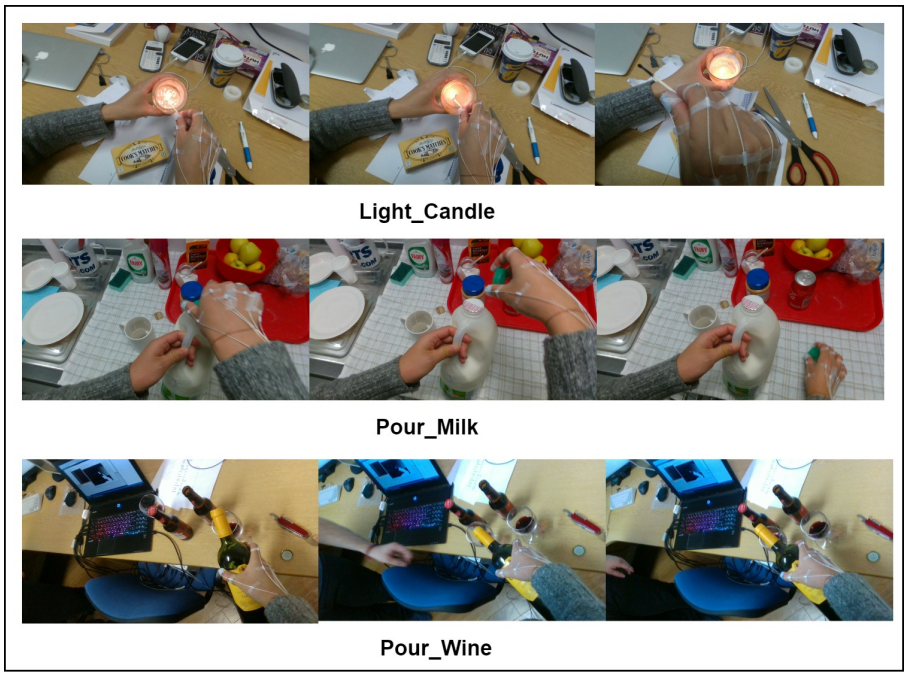


Figure 6-6: First Person Hand Action(FPHA) Dataset

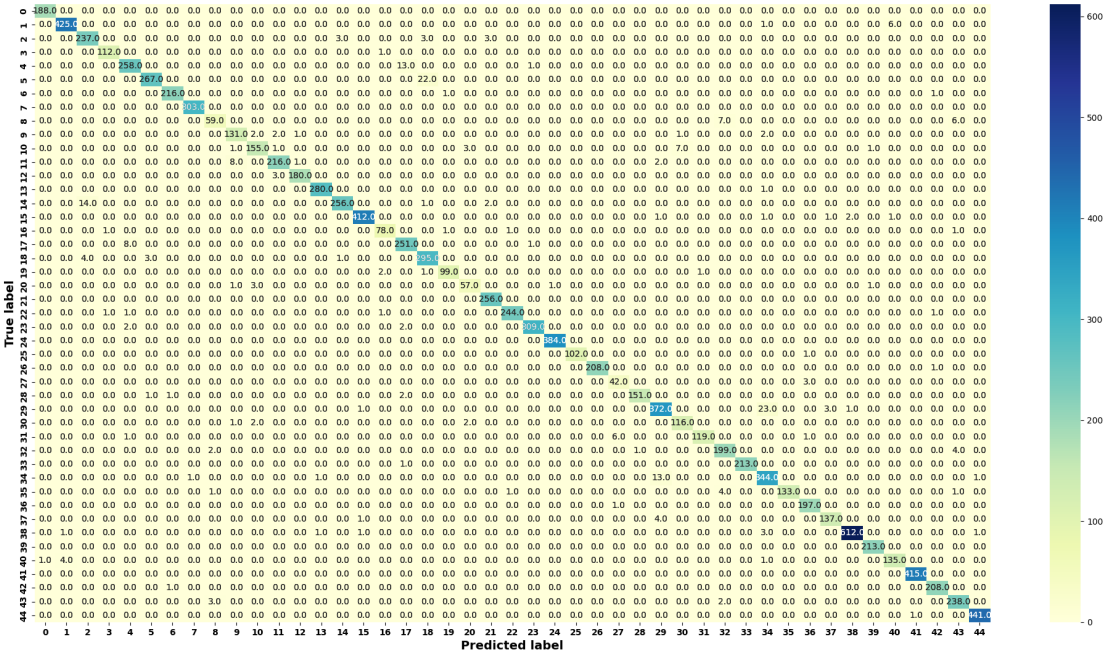


Figure 6-7: Confusion matrix of FPHA dataset, the proposed model achieved 100 percent accuracy on seven classes. The overall accuracy of the model is more than 98 percent.

dataset. Our model achieved over 84% accuracy across all classes and reached 100% accuracy in seven classes. Some gestures are misclassified because of similar gesture patterns. For example, class 2 "close juice bottle" is misclassified with class 3 "close liquid soap" because the skeleton trajectory is quite similar in both cases. Similarly, class 8 “flip sponge" is misclassified with class 35 “squeeze sponge” having similar type of trajectory and activity performed. To assess the classification accuracy of our system for a particular set of test data, we calculate the ROC, precision(P), recall(R), and F1-score as shown in Table 6.2 and the micro-average ROC curve (pink dotted line) shown in Figure 6-9 on FPHA dataset, aggregates the contributions of all classes to compute the average performance of the classifier. Our experiments show that the F1-Score is greater than 85% for all classes, and the macro average accuracy of our proposed model is 98.10%. Our proposed model achieved better results and performs remarkably well across all classes as shown in Table 6.2. Figure 6-8 illustrates the

Table 6.2: F1, Precision, and Recall values for different classes using LSTM, GRU, and RNN on FPHA dataset.

Class	RNN			GRU			LSTM			Class	RNN			GRU			LSTM		
	F1	P	R	F1	P	R	F1	P	R		F1	P	R	F1	P	R	F1	P	R
0	0.99	0.99	0.99	0.99	0.98	0.99	1.00	0.99	1.00	23	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	24	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.92	0.97	0.98	0.95	0.91	0.99	0.95	0.93	0.96	25	0.99	0.98	1.00	0.99	0.99	0.99	1.00	1.00	0.99
3	0.96	0.97	0.96	0.96	0.95	0.97	0.99	0.98	0.99	26	0.99	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00
4	0.96	0.94	0.97	0.94	0.93	0.96	0.95	0.96	0.95	27	0.93	0.91	0.96	0.97	0.96	0.98	0.88	0.84	0.93
5	0.94	0.89	0.99	0.95	0.93	0.98	0.95	0.99	0.92	28	0.98	0.99	0.98	0.98	0.97	0.99	0.98	1.00	0.96
6	0.99	0.99	1.00	0.99	1.00	0.99	0.99	0.99	0.99	29	0.93	0.93	0.93	0.94	0.92	0.96	0.94	0.95	0.94
7	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00	30	0.94	0.93	0.93	0.93	0.96	0.90	0.94	0.92	0.96
8	0.84	0.90	0.78	0.83	0.88	0.79	0.86	0.91	0.82	31	1.00	1.00	0.99	0.99	0.98	1.00	0.96	0.99	0.94
9	0.93	0.96	0.89	0.95	0.95	0.95	0.93	0.92	0.95	32	0.94	0.92	0.97	0.93	0.92	0.95	0.95	0.94	0.97
10	0.94	0.96	0.92	0.93	0.95	0.92	0.94	0.96	0.92	33	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	0.96	0.92	1.00	0.97	0.96	0.98	0.96	0.97	0.95	34	0.91	0.94	0.88	0.92	0.97	0.98	0.93	0.91	0.96
12	0.98	0.98	0.98	0.99	0.99	0.98	0.99	0.99	0.98	35	0.97	1.00	0.94	0.95	0.97	0.94	0.97	1.00	0.95
13	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	1.00	36	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0,97	0.99
14	0.94	0.98	0.90	0.96	0.98	0.94	0.96	0.98	0.94	37	0.94	0.93	0.96	0.97	0.97	0.97	0.96	0.97	0.96
15	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	38	0.98	0.97	0.99	0.99	0.99	1.00	0.99	1.00	0.99
16	0.96	0.95	0.96	0.97	0.96	0.98	0.95	0.95	0.96	39	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00
17	0.96	0.96	0.95	0.94	0.93	0.95	0.95	0.93	0.97	40	0.98	0.98	0.98	0.97	0.96	0.98	0.95	0.95	0.96
18	0.92	0.97	0.87	0.93	0.96	0.90	0.94	0.97	0.92	41	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
19	0.97	0.96	0.98	1.00	1.00	1.00	0.98	0.99	0.96	42	0.99	1.00	0.99	0.99	0.99	0.99	0.99	0.99	1.00
20	0.90	0.89	0.90	0.93	0.88	0.98	0.92	0.92	0.92	43	0.95	0.94	0.96	0.95	0.96	0.94	0.97	0.95	0.98
21	0.99	1.00	0.98	0.99	0.99	1.00	0.99	0.98	1.00	44	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
22	0.99	0.99	1.00	0.99	1.00	0.99	0.99	0.99	0.98										

class-wise accuracy on the FPHA dataset. We observe most of the classes achieve more than 93% accuracy, some classes achieved 100% accuracy. This variability arises from similarities in motion tracks among certain classes, leading to confusion between gesture classes. The average accuracy on FPHA stands at approximately 98.10%.

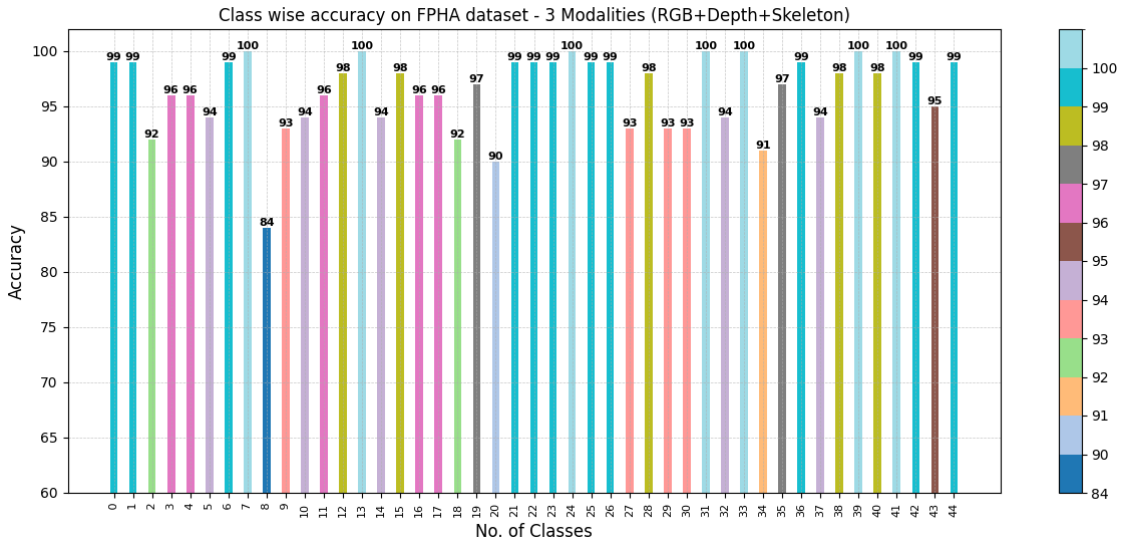


Figure 6-8: The class-wise accuracy of the FPHA dataset using three modalities: RGB, depth, and skeleton. The model achieves high accuracy for most of the classes, with several reaching 100%. The overall accuracy of the proposed model is more than 98%.

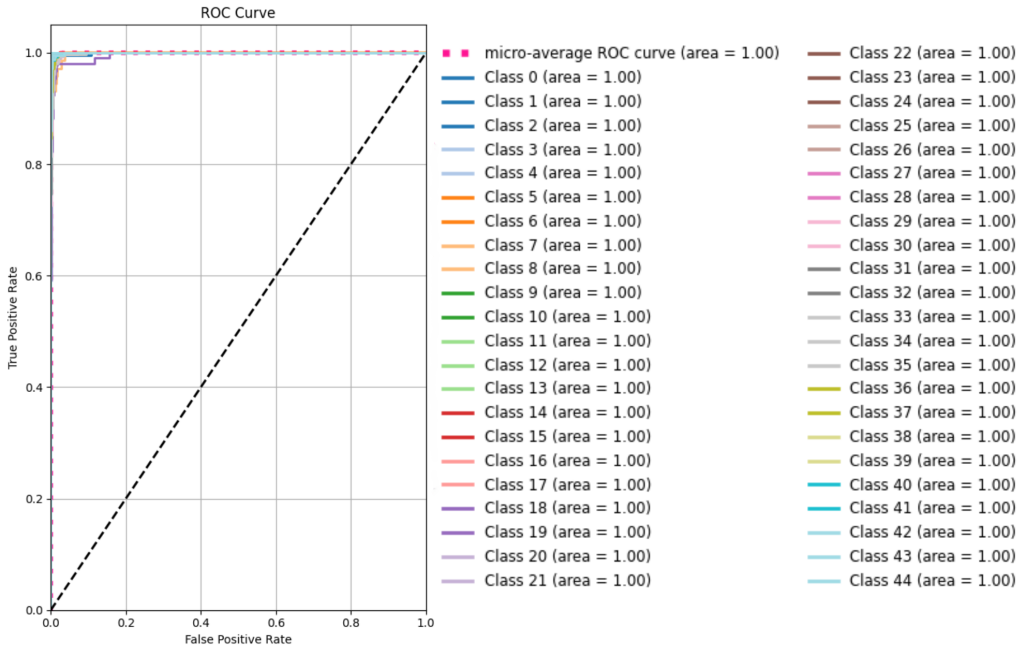


Figure 6-9: For every class, the model exhibits remarkable performance with minimum mis-classification errors. This is shown by the area Under the Curve(AUC) of 1.0 for all classes. This Receiver Operating Characteristics (ROC) curve analysis demonstrates that the model used for the FPHA dataset achieves perfect classification performance across all classes

Experiments on SKIG Dataset

The SKIG (Sheffield Kinect Gesture)[2] dataset comprises 1080 RGB sequences and 1080 depth sequences, totaling 2160 hand gesture sequences. This dataset comprises 10 distinct classes of hand gestures, which are performed by 6 distinct individuals on 3 distinct backgrounds under 2 illumination circumstances using 3 distinct hand shapes. Thus, a total 108 video sequences for each gesture classes. The few SKIG gesture sequence such as "turn around", "Wave", "Come-here", "Cross", "Line", "Circle", "Z", "Triangle", "Up-down", and "Pat" shown in Figure 6-10.

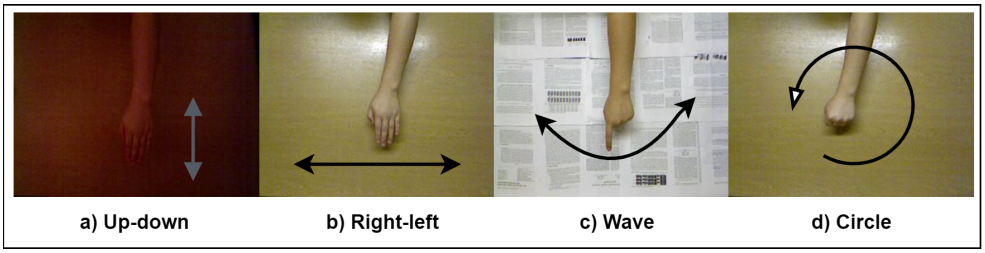


Figure 6-10: Shows SKIG hand gestures dataset images in different lighting conditions and backgrounds a) Up-Down and b) Right-Left c) Wave and d) Circle

Figure 6-11 displays a confusion matrix of the SKIG dataset. In most of the classes we achieved 100% accuracy. some classes are misclassified due to performing almost the same type of gesture, such as class 3 "cross" is misclassified with class 7 "triangle". The overall accuracy of the model is more than 99%. To assess the classification accuracy of our system for a particular set of test data, we calculate the ROC, precision(P), recall(R), and F1-score as shown in Table 6.3 and ROC on the SKIG dataset shown in Figure 6-13. Our experiments shows that the F1-Score is greater than 99% for all classes, and the macro average accuracy of our proposed model is 99.83%, for some classes our approach attains 100% accuracy, proving that our proposed model achieved better results and performs remarkably well across all classes as shown in Table 6.3.

Figure 6-12 illustrates the class-wise accuracy on the SKIG dataset. We observe that while most classes achieved 100% accuracy, and some classes achieved 99% accuracy. This variability arises from similarities in motion tracks among certain classes,

[2]http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm

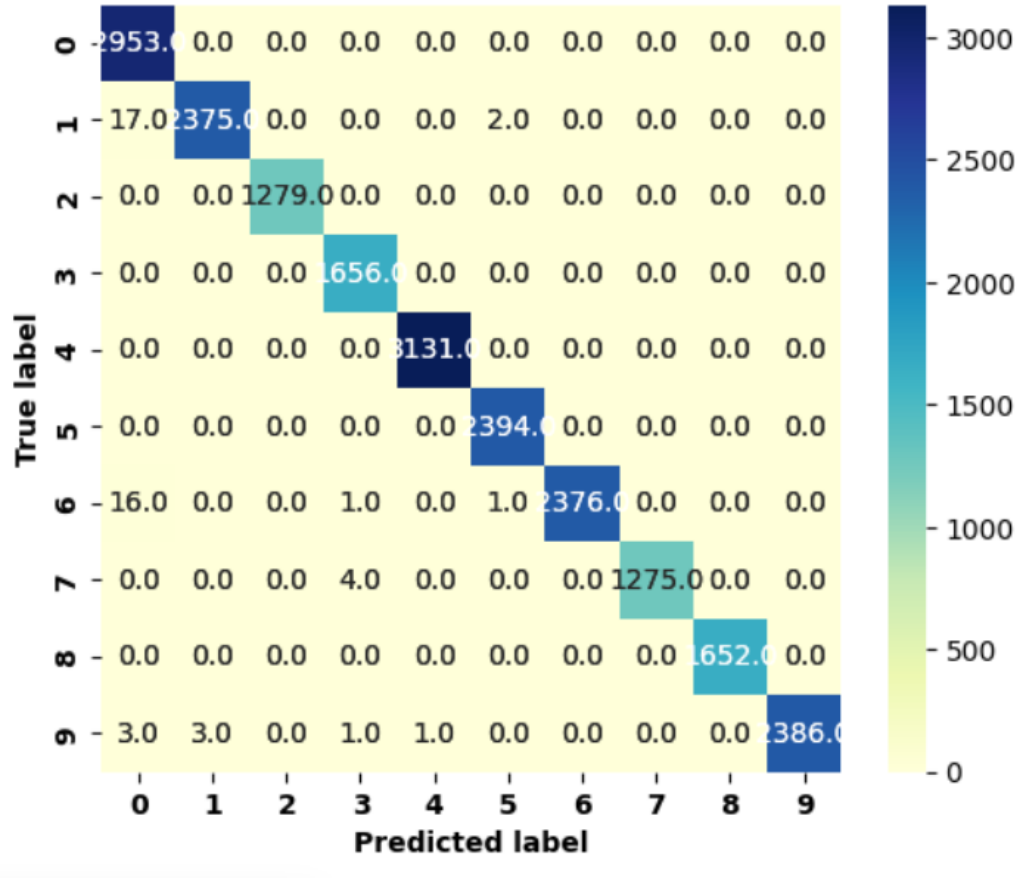


Figure 6-11: The confusion matrix of the SKIG dataset shows that most classes in the proposed model achieved 100% performance. The overall accuracy of the model is more than 99%.

Table 6.3: F1, Precision, and Recall values for different classes using LSTM, GRU, and RNN of SKIG dataset.

Class	RNN			GRU			LSTM		
	F1	P	R	F1	P	R	F1	P	R
0	0.99	0.99	1.00	0.99	0.98	1.00	0.99	0.98	1.00
1	0.99	1.00	0.98	0.99	0.99	0.98	0.99	0.99	0.99
2	0.99	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00
3	1.00	0.99	1.00	1.00	0.99	1.00	0.99	0.99	1.00
4	0.99	1.00	0.99	0.99	1.00	0.99	1.00	0.99	1.00
5	0.99	0.99	0.99	0.99	1.00	0.99	1.00	0.99	1.00
6	0.99	0.98	0.99	0.99	0.99	0.99	1.00	0.99	0.99
7	0.99	0.99	1.00	0.99	1.00	0.99	0.99	1.00	0.99
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	1.00	0.99	1.00	1.00	1.00	0.99	1.00	1.00	0.99

leading to confusion between gesture classes. The average accuracy on SKIG stands at approximately 99.83%.

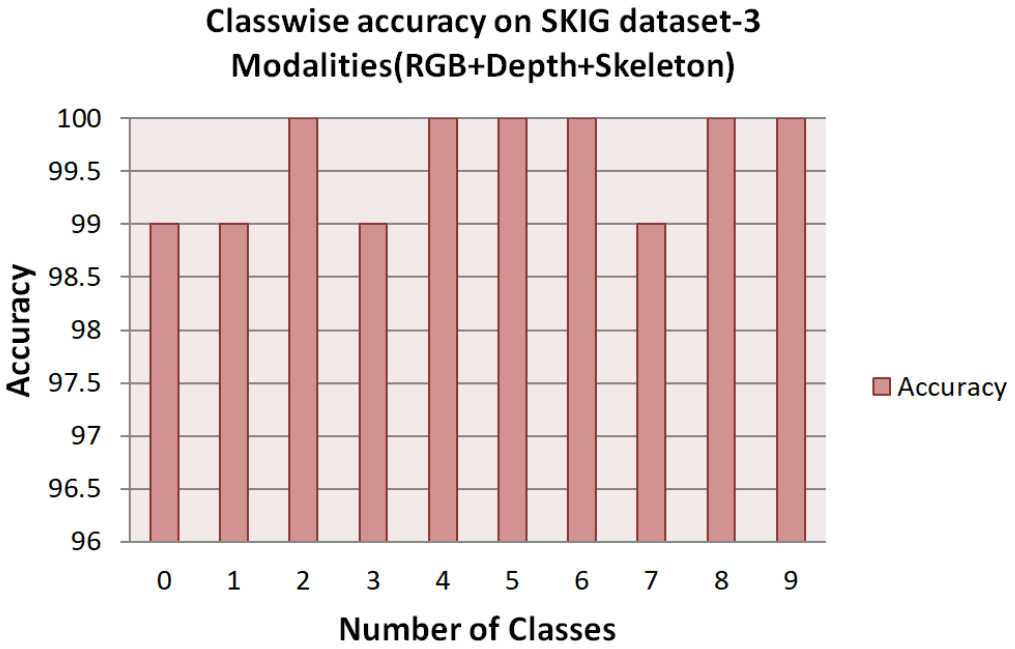


Figure 6-12: Class-wise accuracy of SKIG dataset using three modalities: RGB, depth, and skeleton. The model achieved higher accuracy for most of the classes with several reaching 100%. The overall accuracy of the proposed model is more than 99%.

6.4.3 Ablation Study

In literature, mostly single modality is used. However, combining features from multiple modalities can further improve accuracy. Detecting and tracking gesturing hands remains a challenging task. To address this issue, a fused model incorporating three modalities (RGB + depth + skeleton) can be utilized to improve gesture classification. The FPHA and SKIG datasets already include RGB and depth frames of the videos. Additionally, the skeleton trajectory videos are generated using the provided skeleton joint information from the FPHA dataset. In the case of the SKIG dataset, we used MediaPipe to extract skeleton key points from the RGB frames. We then plot the skeleton trajectory for each frame and make a skeleton video for the SKIG dataset as shown in Figure 6-5. The proposed architecture, which integrates

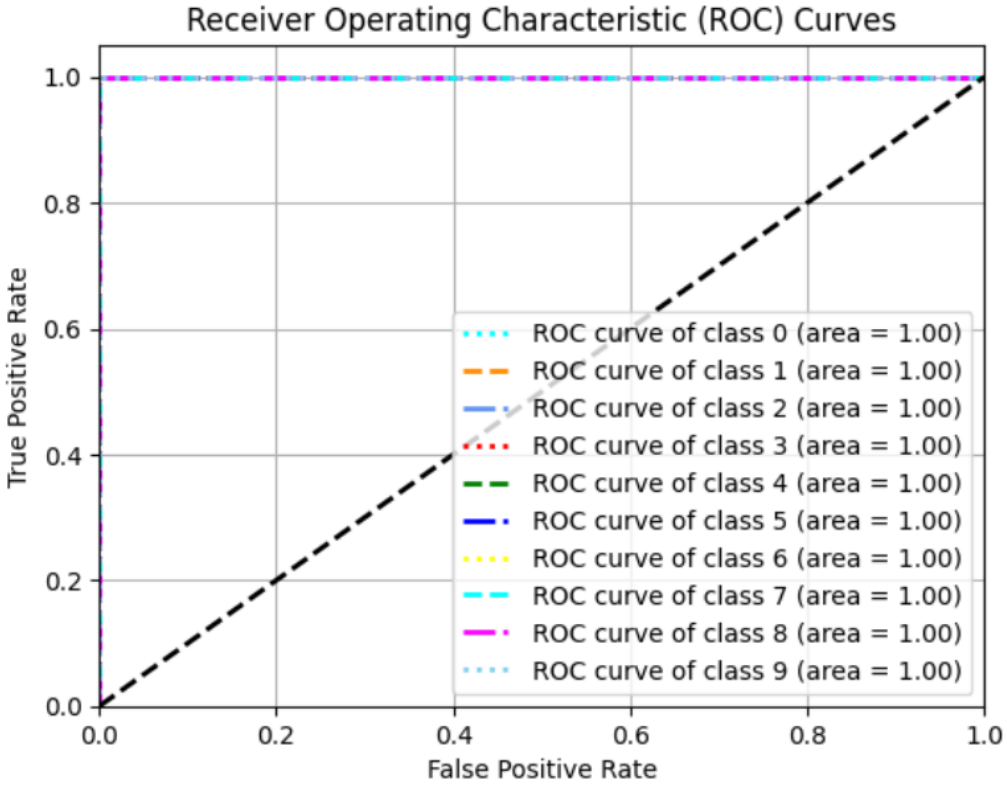


Figure 6-13: Area Under the Curve(AUC) of 1.00 for all classes shows that the model performs exceptionally well across all classes with few mis-classification errors. ROC curve analysis demonstrates that the model used for the SKIG dataset achieved excellent classification performance across all classes

three modalities, using three different models for sequence-to-sequence learning is: 1DCNN+ResCLIP-LSTM, 1DCNN+ResCLIP-GRU, and 1DCNN+ResCLIP-RNN.

- Effect of ResCLIP+RNN model: We analyzed the effect of an RNN model on the proposed work. The main drawback of RNNs is their vanishing gradient problem due to which the accuracy of the model is decreased, which results in the final accuracy being 1% to 2% less with GRU and LSTM. Experimental results are evaluated over the SKIG and FPHA datasets, as depicted in Table 6.4.

- Effect of ResCLIP+GRU model: We analyzed the effect of a GRU model on the proposed work. The results show the final accuracy being 0.5% to 1% less than that of the LSTM. Experimental analysis are evaluated over the SKIG and FPHA datasets, both of which are complex datasets. As a result, the LSTM outperforms the GRU because the proposed model involves complex sequential data, as depicted in Table 6.4.

- Effect of ResCLIP+LSTM model: Our experimental findings reveal that ResCLIP-LSTM outperforms both ResCLIP-RNN and ResCLIP-GRU. ResCLIP-LSTM exhibits advantages such as reduced tensor operations, it removes the vanishing gradient problem and exploding gradients problem leading to faster efficiency compared to RNN and GRU models. Particularly, ResCLIP+LSTM demonstrates superior performance when training parameters are limited compared to the other state-of-the-art methods. As we can see the accuracy using ResCLIP + LSTM on the FPHA and SKIG dataset is superior with feature fusion than a single modality as shown in Table 6.4.

In our proposed model RNN has $0.95M$ parameters while GRU and LSTM have $1.03M$ and $1.17M$ parameters respectively. Additionally, it demands fewer computational resources, making it more suitable for tasks or environments with constrained computing capabilities. RGB, depth, and skeleton information are input data for sequence-to-sequence learning.

Equations 6.10 and 6.12 illustrate feature-level fusion and decision-level fusion, respectively. By using Equation 6.12, the final gesture recognition accuracy of the

proposed model is determined. Fusion strategies involving three modalities and the corresponding results at the decision level are presented in Table 6.4.

Table 6.4: Comparison of recognition accuracy for ablation study on FPHA and SKIG datasets on different strategies with varying modalities.

Strategy	RGB	Depth	Skeleton	Accuracy(FPHA)	Accuracy(SKIG)
	✓	×	×	96.38	97.24
	×	✓	×	92.00	91.16
	×	×	✓	91.14	89.90
ResCLIP+RNN	✓	✓	×	96.99	98.48
	✓	×	✓	96.00	98.15
	×	✓	✓	92.00	92.23
	✓	✓	✓	96.42	98.80
	✓	×	×	96.30	99.01
	×	✓	×	91.33	92.37
	×	×	✓	91.89	90.56
ResCLIP+GRU	✓	✓	×	97.19	99.53
	✓	×	✓	96.83	99.31
	×	✓	✓	93.12	93.00
	✓	✓	✓	97.94	99.25
	✓	×	×	97.27	99.34
	×	✓	×	92.43	92.85
	×	×	✓	92.32	92.21
ResCLIP+LSTM	✓	✓	×	97.29	99.52
	✓	×	✓	96.92	99.53
	×	✓	✓	94.43	94.65
	✓	✓	✓	98.10	99.87

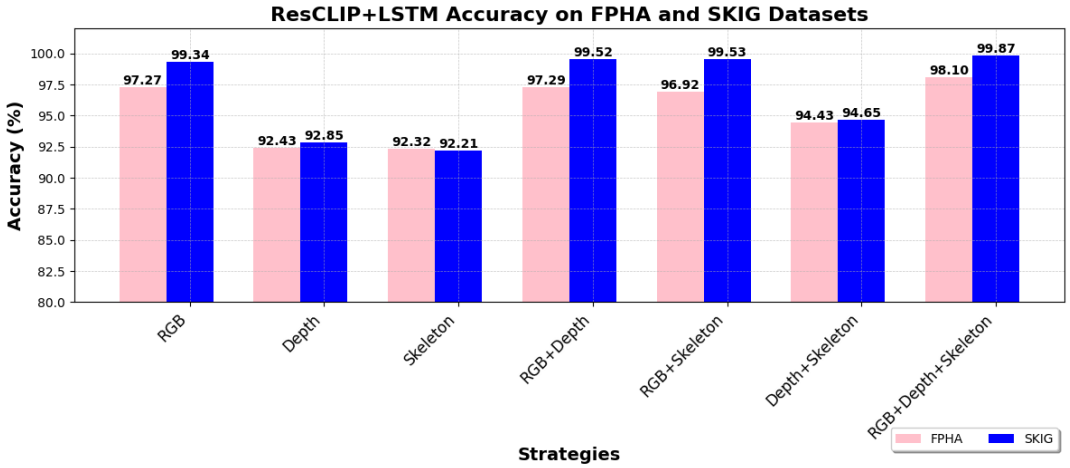


Figure 6-14: The graph shows that the accuracy of the ResCLIP+LSTM model is generally increased when multiple data types are combined; the greatest improvement is shown when all three data types—RGB, Depth, and Skeleton—are used together. Across every methodology, the SKIG dataset consistently outperforms the FPHA dataset

This Figure 6-14 illustrates the accuracy of different modalities on the proposed model for the FPHA and SKIG datasets. To gain a deep insight into the proposed model (ResCLIP+LSTM), we evaluated two additional experiments for an ablation study, as indexed in Table 6.5. This part focuses on validating the effectiveness of the proposed model by examining the impact of different dropout layers and the effect of adding one more Conv1D layer.

- Effect of the Dropout layer: To examine the impact of the dropout layer, we evaluated the results by using two dropout rates: 0.3 and 0.5. From the results, we observed that a dropout rate of 0.3 resulted in lower accuracy compared to a dropout rate of 0.5, as shown in Table 6.5. A 0.5 dropout rate provides stronger regularization than a 0.3 dropout rate, reducing the risk of overfitting. By dropping out 50% of the neurons, the model becomes more robust as it is forced to learn redundant representations. This redundancy can improve the model's ability to handle noise and variations in the input data. For our proposed model, we prefer a 0.5 dropout rate for hand gesture recognition.

- Effect of adding one more Con1D layer: First, we analyzed the effect of three Conv1D layers and found that it created overfitting during training. Then, we experimented with two and three layers of Conv1D and observed that consideration of two layers avoids overfitting, resulting in 2.21% higher accuracy on the FPHA dataset, and 1.56% higher accuracy on the SKIG dataset, as shown in Table 6.5.

Table 6.5: Comparison of recognition accuracy for ablation study on FPHA and SKIG datasets using the proposed model (ResCLIP+LSTM).

R	**D**	**S**	**D(0.5)**	**D(0.3)**	**FPHA(CD2)**	**FPHA(CD3)**	**SKIG(CD2)**	**SKIG(CD3)**
✓	✓	✓	✓	×	98.10	-	99.87	-
✓	✓	✓	×	✓	-	95.89	-	98.31

R=RGB, D=Depth, S=skeleton, D=Dropout, CD2=Two layers of Conv1D, CD3=Three layers of Conv1D

6.4.4 Comparison with Literature

Comparison of FPHA Dataset with State-of-the-Art Methods

We compare the experimental results with state-of-the-art on the FPHA dataset. Table 6.6 presents a comparison between the proposed model and current approaches. In the paper [45], the author employs CNN and RNN for video classification and presents a novel method for hand action identification called sparse low-rank scores. Similar to Yang et al. [114], [84] and Yilin et al. [39], these authors also used transformer models for dynamic hand gesture recognition while integrating RGB and skeleton modalities. The paper [46] proposed two novel 2D hand posture estimation models for an egocentric view. These models aim to address challenges in dynamic hand gesture recognition, such as overlapping hand occlusion.

We implemented a fusion of RGB, depth, and skeleton features in our proposed model. While some other researchers have also used deep learning methods with different modalities such as RGB, skeleton, or depth data only or combination of these modalities [122], [45], [67]. The results shown in Table 6.6 demonstrate that the proposed ResCLIP+LSTM model surpasses other state-of-the-art methods on the FPHA dataset, achieving an accuracy of 98.10% except the paper [122] but having 1.17M parameters and computationally efficient model.

Table 6.6: FPHA dataset's classification accuracy comparison(%) Results.

Methods	R	D	S	Acc	Par	B.S	L.R
LSTM [122]	✓	×	×	98.40	135M	32	0.00042
ResNet-152+RGB [45]	✓	×	×	87.06	311.1M	64	0.005
ResNet-152+RGB [45]	×	✓	×	84.31	311.1M	64	0.005
HMM + 2DCNN [65]	×	×	✓	90.96	-	40	3e-4
GCN [67]	×	×	✓	89.41	1.41M	10	0.001
TransformerCNN [84]	×	×	✓	87.32	-	64	1e-3
Conv2D [11]	×	×	✓	93.22	-	30	0.01
YOLOv7 [46]	✓	×	✓	94.43	21.45M	64	0.001
RNN [45]	✓	✓	×	91.59	341.1M	64	0.0005
Resnet [114]	✓	×	✓	85.22	-	4	0.00001
Transformer [39]	×	✓	✓	86.36	11.17M	2	0.00003
CNN [117]	✓	×	✓	89.93	-	64	-
ResCLIP+LSTM	✓	✓	✓	**98.10**	1.177M	64	0.0001

R=RGB, D=Depth, S=skeleton, Acc=Accuracy, Par=Parameter, B.S=Batch Size

Comparison of SKIG dataset with State-of-the-Art Methods

We compare our experimental results with state-of-the-art on the SKIG dataset as shown in Table 6.7. In the paper [25], the author employs 3DCNN and LSTM for video classification and they use RGB data as input for their model. Similarly, the author [26] focuses on a single RGB modality as an input and finds the spatio-temporal features. The authors Dexu et al. [32] proposed a multimodal gesture recognition technique that integrates DenseNet with BLSTM networks. This approach combines RGB and depth data, similar to the work of other authors Verma et al. [123] and Tang et al. [115], who have utilized these features for gesture recognition. In our proposed model, we implemented a fusion of RGB, depth, and skeleton features. In contrast, some researchers used either a single modality or a combination of RGB and depth for hand gesture recognition [32], [115], [123]. The results shown in Table 6.7 demonstrate that the proposed ResCLIP+LSTM model surpasses other state-of-the-art methods on the SKIG dataset, achieving an accuracy of 99.87% with a smaller number of parameters, i.e., 0.77 million parameters.

Table 6.7: SKIG dataset's classification accuracy comparison(%) results.

Methods	R	**D**	**S**	**Acc**	**Par**	**B.S**	**L.R**
3DCNN+LSTM [25]	✓	×	×	99.65	159M	8	0.000001
ResNet-18 [26]	✓	×	×	98.70	3.35M	16	0.01
DensNet,BLSTM [32]	✓	✓	×	99.07	-	8	-
3DCNN, LSTM. [115]	✓	✓	×	99.63	4.73M	8	0.001
MHI+VGG16 [123]	✓	✓	×	99.12	-	-	0.01
ResCLIP+LSTM	✓	✓	✓	**99.87**	0.77M	32	0.0001

R=RGB, D=Depth, S=skeleton, Acc=Accuracy, Par=Parameter, B.S=Batch Size

6.5 Conclusion

Our proposed model introduces a ResCLIP-LSTM-based hand gesture recognition system that is computationally efficient. This approach is designed to achieve high-speed performance and can operate effectively with a smaller number of training samples. By utilizing a fusion of RGB video, depth video, and skeleton trajectory

video, our model achieves high accuracy. Our proposed model uses a fusion of RGB, depth, and skeleton data. Features are extracted using a pre-trained ResCLIP model, and processed with LSTM units. Features from all three modalities are concatenated and classified using a Softmax classifier. Our model performs competitively on the FPHA and SKIG benchmark datasets and achieves more than 99% on the SKIG dataset and more than 98% on the FPHA dataset, matching state-of-the-art methods.

The ResCLIP-LSTM model integrates CLIP with residual blocks to enhance feature extraction across these data types, optimizing performance with fewer training samples and mitigating vanishing gradient issue.

Chapter 7

Conclusion

Due to the advancement of technologies and the digital era the need of human-computer interaction(HCI) techniques needs to grow. Hand gesture recognition is a one of the possible way that makes human interaction with the computers. However, challenges arise due to the small size and complexity of hands, making gesture recognition and pose estimation difficult. Deep learning techniques address these issues by capturing temporal information and complex hand dynamics. Researchers also face challenges such as time-consuming data processing and difficulty in extracting the region of interest. The growing need for hand gesture recognition has led to various applications in robotics, smart homes, gaming, autonomous vehicles, and healthcare, minimizing interference and reducing communication costs.

7.1 Summary and Contribution of the Thesis

In this thesis, we proposed four frameworks for vision-based dynamic hand gesture recognition that address various challenges such as lighting variations, occlusion, and complex backgrounds and inter and intra class variation. These models are designed to efficiently handle the complexities of dynamic hand gestures, ensuring high-speed performance and adaptability across different conditions. They are also optimized to work effectively with a smaller number of training samples, making them suitable for practical applications with limited data availability.

- In first framework, we solve the challenge of hand detection and tracking where RGB videos are used to extract the features using CLIP model. RGB video data is useful for extracting features, as it provides the visual information about the hand, as well as spatial information that helps identify the hand's shape and texture. This enhances the model's understanding of the gesture context. Further, we have used BLSTM model for sequence to sequence learning and classify the dynamic hand gesture. Experimental results on CHG dataset with an accuracy of 97% and LISA dataset with an average accuracy of 86% shows the prominence of the proposed model.

  The novelty of our work lies in utilizing the CLIP model to extract features from RGB video data. The CLIP-BLSTM model is specifically designed to address challenges associated with small hand sizes and hand tracking, proving to be efficient with fewer training samples and parameters. Overall, it performs effectively in different lighting environments, establishing it as an accurate hand gesture recognition system.

  However, during experiments we witnessed that RGB approach faces challenges such as occlusion, and background clutter, which can impede recognition rate. Thus, in next frameworks we have tried to solve these issues using the skeleton data also solve the problem of inter-class and intra-class variation.

- In second frame work, we have used skeleton data to plot the hand gesture trajectory. If skeleton data is available, trajectory can be plotted using skeleton points without bothering about the hand occlusion, background clutter. Further, inter and intra-class variation problem resolved using the DDA loss. We extracted the features of the plotted trajectory using the VGG16, DenseNet121 and InceptionV3 and ensemble learning used to find the best suitable features. Then DDA loss used to refine the extracted features by increasing the closeness of with-in class similarity features and decreases the between class similarity features. Further, gestures are classified using the SoftMax activation function. The Experimental results demonstrate that the proposed model surpasses other

state-of-the-art methods DHG14/28 dataset with an average accuracy of 97.1%, and 99.8% average accuracy on 26-Gestures dataset.

The proposed hand gesture recognition framework increases gesture recognition accuracy and efficiently handles intra and inter-class variability in hand gesture recognition by integrating ensemble learning with a Discriminant Distribution-Agnostic Loss. Use of skeleton data also overcome the challenge of hand detection in occlusion and cluttered background.

- In third framework, we have proposed a hybrid deep-learning model where skeleton and optical flow videos are calculated in two pipelines parallely and Bi-GRU used for sequence to sequence learning. The advantage of using skeleton point video is that it overcomes the challenges of hand occlusion and complex background. The advantage of calculating the optical flow video is that it captures the hand motion and discards the irrelevant data and stationary background. For each skeleton trajectory video, features are extracted using Xception-Net called as a finger motion feature (FMF) and features extracted from optical flow videos are global motion features(GMF). Features extracted from a single modality is not sufficient enough to classify the dynamic hand gesture, thus, we proposed a fusion of FMF and GMF that gives better accuracy compared to the single modality. The proposed model achieved competitive results on benchmark datasets such as on DHG14/28 dataset with an accuracy of 98.2%, on NWUHG dataset with an accuracy of 99.2% with other state-of-the-art models while maintaining a considerably lower computational complexity.

  Our proposed model offers a bidirectional gated recurrent unit (Bi-GRU) model-based hand gesture recognition system that is computationally effective than Bi-LSTM. This method is designed to attain high-speed performance while being capable of working successfully even with limited training samples. This dual-feature extraction method allows the model to achieve a more robust understanding of hand gestures, improving overall performance in diverse environments.

- In the fourth framework, we present a multimodal hybrid framework that utilizes the different strength of each modality. Use of all three modalities RGB, depth, and skeleton data boost the performance of the proposed model. Proposed work introduced ResCLIP-LSTM-based hybrid method where CLIP is used to extract the features from each modality individually and the features are refined using the residual block and then LSTM used for sequence to sequence learning. Experimental results on FPHA dataset with an accuracy of 98.10% and SKIG dataset with an average accuracy of 99.87% shows the prominence of the proposed model.

  The ResCLIP-LSTM model integrates CLIP with residual blocks to enhance feature extraction across these data types, optimizing performance with fewer training samples and mitigating vanishing gradient issue.

7.2 Future Directions

- As skeleton data have shown a very promising form of dataset that can be used in graph convolutional neural networks as a hand gesture recognition. Use of skeleton data in graph convolutional neural network may boost the performance and can solve many challenges encounter in the hand gesture recognition.

- Another future work can be to create a more robust framework that emphasizes key frames in gesture sequences, which can enhance recognition accuracy and efficiency by focusing on the most informative moments in a gesture.

- Another future work can be to propose a real-time applications for hand gesture recognition, enabling instant feedback and interaction in various context using the framework given in this thesis.

- Future work can be to integrate gesture recognition systems with artificial intelligence and Internet of Things (IoT) devices, which can enhance user experiences in smart home environments, allowing for intuitive control of devices through hand gestures.

- In order to enable more precise and natural gesture recognition for applications such as virtual reality (VR), augmented reality (AR), and robots, 3D dynamic hand gesture recognition can be developed.

- In the future, we can explore DDA loss with graph convolutional neural networks for dynamic hand gesture recognition. To enhance adaptability, we can investigate the extension of our models for egocentric hand gesture for daily hand action activities.

- In the future, as a prospective directional we want to create more accurate feature extractions which can handle every kind of gesture data without relying on the feature chosen for the specific kind of gestures.

References

[1] M. V. Jadeja, M. R. Davda, M. C. Patel, M. J. Solanki, and M. M. Zala, "Hand gesture recognition approach: A survey," *International Journal for Innovative Research in Science & Technology (IJIRST), ISSN*, pp. 2349–6010.

[2] H.-M. Zhu and C.-M. Pun, "Hand gesture recognition with motion tracking on spatial-temporal filtering," in *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, 2011, pp. 273–278.

[3] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll, "Online dynamic hand gesture recognition including efficiency analysis," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 2, pp. 85–97, 2020.

[4] G. Oliveira, L. L. Minku, and A. L. Oliveira, "Tackling virtual and real concept drifts: An adaptive gaussian mixture model approach," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[5] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 246–252.

[6] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in *2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication*. IEEE, 2012, pp. 411–417.

[7] D. Sarma, V. Kavyasree, and M. K. Bhuyan, "Two-stream fusion model for dynamic hand gesture recognition using 3d-cnn and 2d-cnn optical flow guided motion template," *arXiv preprint arXiv:2007.08847*, 2020.

[8] B. Verma and A. Choudhary, "Affective state recognition from hand gestures and facial expressions using grassmann manifolds," *Multimedia Tools and Applications*, vol. 80, no. 9, pp. 14 019–14 040, 2021.

[9] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2799–2813, 2017.

[10] B. Verma and A. Choudhary, "Grassmann manifold based dynamic hand gesture recognition using depth data," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 2213–2237, 2020.

[11] X. S. Nguyen, L. Brun, O. Lézoray, and S. Bougleux, "A neural network based on spd manifold learning for skeleton-based hand gesture recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 036–12 045.

[12] L. Zhou, X. Bai, X. Liu, J. Zhou, and E. R. Hancock, "Learning binary code for fast nearest subspace search," *Pattern Recognition*, vol. 98, p. 107040, 2020.

[13] B. Verma and A. Choudhary, "Framework for dynamic hand gesture recognition using grassmann manifold for intelligent vehicles," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 721–729, 2018.

[14] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE transactions on intelligent transportation systems*, vol. 15, no. 6, pp. 2368–2377, 2014.

[15] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–9.

[16] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[17] S. Conseil, S. Bourennane, and L. Martin, "Comparison of fourier descriptors and hu moments for hand posture recognition," in *2007 15th European Signal Processing Conference*. IEEE, 2007, pp. 1960–1964.

[18] E. Kollorz, J. Penne, J. Hornegger, and A. Barke, "Gesture recognition with a time-of-flight camera," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3-4, pp. 334–343, 2008.

[19] L. Kane and P. Khanna, "Depth matrix and adaptive bayes classifier based dynamic hand gesture recognition," *Pattern Recognition Letters*, vol. 120, pp. 24–30, 2019.

[20] H. Tang, H. Liu, W. Xiao, and N. Sebe, "Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion," *Neurocomputing*, vol. 331, pp. 424–433, 2019.

[21] C. Zhang, Z. Wang, Q. An, S. Li, A. Hoorfar, and C. Kou, "Clustering-driven dgs-based micro-doppler feature extraction for automatic dynamic hand gesture recognition," *Sensors*, vol. 22, no. 21, p. 8535, 2022.

[22] G. Benitez-Garcia, L. Prudente-Tixteco, L. C. Castro-Madrid, R. Toscano-Medina, J. Olivares-Mercado, G. Sanchez-Perez, and L. J. G. Villalba, "Improving real-time hand gesture recognition with semantic segmentation," *Sensors*, vol. 21, no. 2, p. 356, 2021.

[23] H. Liang, J. Yuan, D. Thalmann, and Z. Zhang, "Model-based hand pose estimation via spatial-temporal hand parsing and 3d fingertip localization," *The Visual Computer*, vol. 29, pp. 837–848, 2013.

[24] H. Wu, J. Wang, and X. Zhang, "Combining hidden markov model and fuzzy neural network for continuous recognition of complex dynamic gestures," *The Visual Computer*, vol. 33, pp. 1265–1278, 2017.

[25] G. Chen, Z. Dong, J. Wang, and L. Xia, "Parallel temporal feature selection based on improved attention mechanism for dynamic gesture recognition," *Complex & Intelligent Systems*, vol. 9, no. 2, pp. 1377–1390, 2023.

[26] X. Xiaoyan, C. Panyu, and Z. Zhaozhe, "A dynamic gesture recognition method based on encoded video," in *Proceedings of the 2022 5th International Conference on Artificial Intelligence and Pattern Recognition*, 2022, pp. 711–716.

[27] S. E. Ovur, X. Zhou, W. Qi, L. Zhang, Y. Hu, H. Su, G. Ferrigno, and E. De Momi, "A novel autonomous learning framework to enhance semg-based hand gesture recognition using depth information," *Biomedical Signal Processing and Control*, vol. 66, p. 102444, 2021.

[28] H. Hikawa, Y. Ichikawa, H. Ito, and Y. Maeda, "Dynamic gesture recognition system with gesture spotting based on self-organizing maps," *Applied Sciences*, vol. 11, no. 4, p. 1933, 2021.

[29] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliūnas, and K. H. Abdulkareem, "Real-time hand gesture recognition based on deep learning yolov3 model," *Applied Sciences*, vol. 11, no. 9, p. 4164, 2021.

[30] K. Lai and S. N. Yanushkevich, "Cnn+ rnn depth and skeleton based dynamic hand gesture recognition," in *2018 24th international conference on pattern recognition (ICPR)*. IEEE, 2018, pp. 3451–3456.

[31] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2881–2885.

[32] D. Li, Y. Chen, M. Gao, S. Jiang, and C. Huang, "Multimodal gesture recognition using densely connected convolution and blstm," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3365–3370.

[33] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018.

[34] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, "Construct dynamic graphs for hand gesture recognition via spatial-temporal attention," *arXiv preprint arXiv:1907.08871*, 2019.

[35] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 106–113.

[36] Y. Li, D. Ma, Y. Yu, G. Wei, and Y. Zhou, "Compact joints encoding for skeleton-based dynamic hand gesture recognition," *Computers & Graphics*, vol. 97, pp. 191–199, 2021.

[37] Y. Li, Z. He, X. Ye, Z. He, and K. Han, "Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, pp. 1–7, 2019.

[38] T. Do, K. Vuong, and H. S. Park, "Egocentric scene understanding via multimodal spatial rectifier," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2832–2841.

[39] Y. Wen, H. Pan, L. Yang, J. Pan, T. Komura, and W. Wang, "Hierarchical temporal transformer for 3d hand pose estimation and action recognition from egocentric rgb videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 243–21 253.

[40] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1011–1021, 2018.

[41] R. Tripathi and B. Verma, "Motion feature estimation using bi-directional gru for skeleton-based dynamic hand gesture recognition," *Signal, Image and Video Processing*, pp. 1–10, 2024.

[42] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3278–3289, 2018.

[43] G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, and K. Yanai, "Ipn hand: A video dataset and benchmark for real-time continuous hand gesture recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4340–4347.

[44] Q. Gao, Y. Chen, Z. Ju, and Y. Liang, "Dynamic hand gesture recognition based on 3d hand pose estimation for human-robot interaction," *IEEE Sensors Journal*, 2021.

[45] K. Roy, "Multimodal score fusion with sparse low rank bilinear pooling for egocentric hand action recognition," *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.

[46] W. Mucha and M. Kampel, "In my perspective, in my hands: Accurate egocentric 2d hand pose and action recognition," *arXiv preprint arXiv:2404.09308*, 2024.

[47] D. S. Breland, S. B. Skriubakken, A. Dayal, A. Jha, P. K. Yalavarthy, and L. R. Cenkeramaddi, "Deep learning-based sign language digits recognition from thermal images with edge computing system," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 10 445–10 453, 2021.

[48] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, T. S. Alrayes, H. Mathkour, and M. A. Mekhtiche, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192 527–192 542, 2020.

[49] Y. Li, D. Ma, Y. Yu, G. Wei, and Y. Zhou, "Compact joints encoding for skeleton-based dynamic hand gesture recognition," *Computers & Graphics*, vol. 97, pp. 191–199, 2021.

[50] B. Hu and J. Wang, "Deep learning based hand gesture recognition and uav flight controls," *International Journal of Automation and Computing*, vol. 17, no. 1, pp. 17–29, 2020.

[51] S. Mishra, "Infant hand detection and tracking," 2021.

[52] D. S. Breland, S. B. Skriubakken, A. Dayal, A. Jha, P. K. Yalavarthy, and L. R. Cenkeramaddi, "Deep learning-based sign language digits recognition from thermal images with edge computing system," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 10 445–10 453, 2021.

[53] A. D'Eusanio, A. Simoni, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, "Multimodal hand gesture classification for the human–car interaction," in *Informatics*, vol. 7, no. 3. Multidisciplinary Digital Publishing Institute, 2020, p. 31.

[54] N. L. Hakim, T. K. Shih, S. P. Kasthuri Arachchi, W. Aditya, Y.-C. Chen, and C.-Y. Lin, "Dynamic hand gesture recognition using 3dcnn and lstm with fsm context-aware model," *Sensors*, vol. 19, no. 24, p. 5429, 2019.

[55] N. Nasri, S. Orts-Escolano, and M. Cazorla, "An semg-controlled 3d game for rehabilitation therapies: Real-time time hand gesture recognition using deep learning techniques," *Sensors*, vol. 20, no. 22, p. 6451, 2020.

[56] M. S. Abdallah, G. H. Samaan, A. R. Wadie, F. Makhmudov, and Y.-I. Cho, "Light-weight deep learning techniques with advanced processing for real-time hand gesture recognition," *Sensors*, vol. 23, no. 1, p. 2, 2022.

[57] R. Jain, R. K. Karsh, and A. A. Barbhuiya, "Encoded motion image-based dynamic hand gesture recognition," *The visual computer*, vol. 38, no. 6, pp. 1957–1974, 2022.

[58] H. Mahmud, R. Islam, and M. K. Hasan, "On-air english capital alphabet (eca) recognition using depth information," *The Visual Computer*, vol. 38, no. 3, pp. 1015–1025, 2022.

[59] J. Li, R. Liu, D. Kong, S. Wang, L. Wang, B. Yin, and R. Gao, "Attentive 3d-ghost module for dynamic hand gesture recognition with positive knowledge transfer," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, 2021.

[60] C. Li, S. Li, Y. Gao, X. Zhang, and W. Li, "A two-stream neural network for pose-based hand gesture recognition," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 4, pp. 1594–1603, 2021.

[61] C. Ma, A. Wang, G. Chen, and C. Xu, "Hand joints-based gesture recognition for noisy dataset using nested interval unscented kalman filter with lstm network," *The visual computer*, vol. 34, pp. 1053–1063, 2018.

[62] S. Ameur, A. B. Khalifa, and M. S. Bouhlel, "A novel hybrid bidirectional unidirectional lstm network for dynamic hand gesture recognition with leap motion," *Entertainment Computing*, vol. 35, p. 100373, 2020.

[63] X. Zhang and X. Li, "Dynamic gesture recognition based on memp network," *Future Internet*, vol. 11, no. 4, p. 91, 2019.

[64] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[65] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan, "Decoupled representation learning for skeleton-based gesture recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5751–5760.

[66] A. A. Mohammed, J. Lv, M. S. Islam, and Y. Sang, "Multi-model ensemble gesture recognition network for high-accuracy dynamic hand gesture recognition," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2022.

[67] S.-H. Peng and P.-H. Tsai, "An efficient graph convolution network for skeleton-based dynamic hand gesture recognition," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[68] W. Zhang, Z. Lin, J. Cheng, C. Ma, X. Deng, and H. Wang, "Sta-gcn: two-stream graph convolutional network with spatial–temporal attention for hand gesture recognition," *The Visual Computer*, vol. 36, pp. 2433–2444, 2020.

[69] Y. Zhou, G. Jiang, and Y. Lin, "A novel finger and hand pose estimation technique for real-time hand gesture recognition," *Pattern Recognition*, vol. 49, pp. 102–114, 2016.

[70] J. Huang, W. Zhou, H. Li, and W. Li, "Attention-based 3d-cnns for large-vocabulary sign language recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2822–2832, 2018.

[71] S. R. Bose and V. S. Kumar, "In-situ recognition of hand gesture via enhanced xception based single-stage deep convolutional neural network," *Expert Systems with Applications*, p. 116427, 2021.

[72] M. Ur Rehman, F. Ahmed, M. Attique Khan, U. Tariq, F. Abdulaziz Alfouzan, N. M Alzahrani, and J. Ahmad, "Dynamic hand gesture recognition using 3d-cnn and lstm networks," *Computers, Materials & Continua*, vol. 70, no. 3, 2021.

[73] K. Nguyen-Trong, H. N. Vu, N. N. Trung, and C. Pham, "Gesture recognition using wearable sensors with bi-long short-term memory convolutional neural networks," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 15 065–15 079, 2021.

[74] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[75] H. Yang, Q. Tian, Q. Zhuang, L. Li, and Q. Liang, "Fast and robust key frame extraction method for gesture video based on high-level feature representation," *Signal, Image and Video Processing*, vol. 15, pp. 617–626, 2021.

[76] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara, "Gesture recognition in ego-centric videos using dense trajectories and hand segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 688–693.

[77] Y. Yuan, H. Zheng, Z. Li, and D. Zhang, "Video action recognition with spatio-temporal graph embedding and spline modeling," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 2422–2425.

[78] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Proc. ieee conf. computer vision and pattern recognition workshops," 2014.

[79] N. Deo, A. Rangesh, and M. Trivedi, "In-vehicle hand gesture recognition using hidden markov models," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2179–2184.

[80] W. Li, Z. Fan, J. Huo, and Y. Gao, "Modeling inter-class and intra-class constraints in novel class discovery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3449–3458.

[81] C. Li, Z. Liu, J. Ren, W. Wang, and J. Xu, "A feature optimization approach based on inter-class and intra-class distance for ship type classification," *Sensors*, vol. 20, no. 18, p. 5429, 2020.

[82] F. M. Caputo, P. Prebianca, A. Carcangiu, L. D. Spano, A. Giachetti *et al.*, "A 3 cent recognizer: Simple and effective retrieval and classification of mid-air gestures from single 3d traces." in *STAG*, 2017, pp. 9–15.

[83] F. M. Caputo, P. Prebianca, A. Carcangiu, L. D. Spano, and A. Giachetti, "Comparing 3d trajectories for simple mid-air gesture recognition," *Computers & Graphics*, vol. 73, pp. 17–25, 2018.

[84] S. Li, Z. Liu, G. Duan, and J. Tan, "Mvhanet: multi-view hierarchical aggregation network for skeleton-based hand gesture recognition," *Signal, Image and Video Processing*, pp. 1–9, 2023.

[85] Z. Deng, Q. Gao, Z. Ju, and X. Yu, "Skeleton-based multifeatures and multistream network for real-time action recognition," *IEEE Sensors Journal*, vol. 23, no. 7, pp. 7397–7409, 2023.

[86] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, pp. 241–258, 2020.

[87] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[88] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[89] A. H. Farzaneh and X. Qi, "Discriminant distribution-agnostic loss for facial expression recognition in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 406–407.

[90] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

[91] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

[92] S. Narayan, A. P. Mazumdar, and S. K. Vipparthi, "Sbi-dhgr: Skeleton-based intelligent dynamic hand gestures recognition," *Expert Systems with Applications*, p. 120735, 2023.

[93] J. Singha, A. Roy, and R. H. Laskar, "Dynamic hand gesture recognition using vision-based approach for human–computer interaction," *Neural Computing and Applications*, vol. 29, no. 4, pp. 1129–1141, 2018.

[94] K. S. Yadav, S. Misra, R. H. Laskar, T. Khan, M. Bhuyan *et al.*, "Removal of self co-articulation and recognition of dynamic hand gestures using deep architectures," *Applied Soft Computing*, vol. 114, p. 108122, 2022.

[95] Y. Liu, S. Song, L. Yang, G. Bian, and H. Yu, "A novel dynamic gesture understanding algorithm fusing convolutional neural networks with hand-crafted features," *Journal of Visual Communication and Image Representation*, vol. 83, p. 103454, 2022.

[96] J. Yu, M. Qin, and S. Zhou, "Dynamic gesture recognition based on 2d convolutional neural network and feature fusion," *Scientific Reports*, vol. 12, no. 1, p. 4345, 2022.

[97] J. Zheng, Z. Feng, C. Xu, J. Hu, and W. Ge, "Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition," *Multimedia Tools and Applications*, vol. 76, pp. 20 525–20 544, 2017.

[98] X. Chen, G. Wang, H. Guo, C. Zhang, H. Wang, and L. Zhang, "Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data," *Sensors*, vol. 19, no. 2, p. 239, 2019.

[99] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Heterogeneous hand gesture recognition using 3d dynamic skeletal data," *Computer Vision and Image Understanding*, vol. 181, pp. 60–72, 2019.

[100] M. Maghoumi and J. J. LaViola, "Deepgru: Deep gesture recognition utility," in *Advances in Visual Computing: 14th International Symposium on Visual Computing, ISVC 2019, Lake Tahoe, NV, USA, October 7–9, 2019, Proceedings, Part I 14*. Springer, 2019, pp. 16–31.

[101] B. Verma, "A two stream convolutional neural network with bi-directional gru model to classify dynamic hand gesture," *Journal of Visual Communication and Image Representation*, vol. 87, p. 103554, 2022.

[102] W. Song, W. Kang, and L. Lin, "Hand gesture authentication by discovering fine-grained spatiotemporal identity characteristics," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[103] D. Miki, K. Kamitsuma, and T. Matsunaga, "Spike representation of depth image sequences and its application to hand gesture recognition with spiking neural network," *Signal, Image and Video Processing*, pp. 1–9, 2023.

[104] Y. Zhang and F. Wang, "Handformer: A dynamic hand gesture recognition method based on attention mechanism," *Applied Sciences*, vol. 13, no. 7, p. 4558, 2023.

[105] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*. Springer, 2003, pp. 363–370.

[106] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[107] P. Radzki, "Detection of human body landmarks-mediapipe and openpose comparison," 2022.

[108] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[109] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4041–4049.

[110] X. Shen, G. Hua, L. Williams, and Y. Wu, "Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields," *Image and Vision Computing*, vol. 30, no. 3, pp. 227–235, 2012.

[111] D. Zhao, H. Li, and S. Yan, "Spatial-temporal synchronous transformer for skeleton-based hand gesture recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[112] D. Zhao, Q. Yang, X. Zhou, H. Li, and S. Yan, "A local spatial–temporal synchronous network to dynamic gesture recognition," *IEEE Transactions on Computational Social Systems*, 2022.

[113] L. Liu and L. Shao, "Synthesis of spatio-temporal descriptors for dynamic hand gesture recognition using genetic programming," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. IEEE, 2013, pp. 1–7.

[114] S. Yang, J. Liu, S. Lu, M. H. Er, and A. C. Kot, "Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 769–786.

[115] X. Tang, Z. Yan, J. Peng, B. Hao, H. Wang, and J. Li, "Selective spatiotemporal features learning for dynamic gesture recognition," *Expert Systems with Applications*, vol. 169, p. 114499, 2021.

[116] R. Tripathi and B. Verma, "Survey on vision-based dynamic hand gesture recognition," *The Visual Computer*, pp. 1–29, 2023.

[117] A. Sabater, I. Alonso, L. Montesano, and A. C. Murillo, "Domain and viewpoint agnostic hand action recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7823–7830, 2021.

[118] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[119] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022.

[120] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419.

[121] L. Liu and L. Shao, "Learning discriminative representations from rgb-d video data," in *Twenty-third international joint conference on artificial intelligence*, 2013.

[122] H. Cho, C. Kim, J. Kim, S. Lee, E. Ismayilzada, and S. Baek, "Transformer-based unified recognition of two hands manipulating objects," in *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 4769–4778.

[123] B. Verma and A. Choudhary, "Dynamic hand gesture recognition using convolutional neural network with rgb-d fusion," in *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, 2018, pp. 1–8.

[124] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 96–99.

Appendix A

Long Short-Term Memory(LSTM)

In this appendix, we explain the LSTM in detail.

A.1 Long Short-Term Memory(LSTM)

Recurrent neural networks struggle with long-term reliance because of the vanishing gradient problem the LSTM networks are created. Instead of analyzing each data point separately, they can analyze entire data sequences and store pertinent information from prior data in series to assist in processing new data points. Therefore it is very adept at processing sequential data.

The framework is composed of the input gate, output gate, forget gate, and cell gate of an LSTM unit and are responsible for controlling the learning process as shown in Fig. A-1. To govern the functioning of the gates throughout the learning process, sigmoid functions are essential. The Cell state refers to the long-term memory in the LSTM. It regulates the data that will be saved in an LSTM cell from earlier periods. The cell gate is modified by the remembering vector, which is known as the forget gate. The forget gate output state instructs the cell gate whether to maintain the information in the cell state if it is 1 or to forget it if it is 0 [72]. Implementing LSTM has the main benefit of resolving the vanishing gradient issue. The following given below equations illustrates the working of LSTM [72].

$$i_t = \sigma(A_t w_{xi} + H_{t-1} w_{Hi} + b_{t-1} w_{bi} + w_{ibais}) \tag{A.1}$$

Where, "i_t" represents the input gate at time step "t". 'A_t" is the input vector and "w_{xi}" is its weight matrix. " H_{t-1}" is the hidden state from the previous time step with "w_{Hi}". "b_{t-1}" and "w_{bi}" are the bias term and its weight matrix, respectively. "w_{ibias}" is an additional bias term for the input gate.

$$f_t = \sigma(A_t w_{xf} + h_{t-1} w_{Hf} + b_{t-1} w_{bf} + w_{fbais}) \tag{A.2}$$

Where, "f_t" represents the output gate. "h_{t-1}", another notation for the hidden state from the previous time step, is similar to "H_{t-1}"."w_{bf}" is the weight matrix for the hidden state to the forget gate, and "w_{bf}" is the weight matrix for the bias to the forget gate.

$$C_t = tanH(A_t w_{AC} + h_{t-1} w_{HC} + w_{zbais}) \tag{A.3}$$

Where, "C_t" represents the candidate cell state. "w_{HC}" is the weight matrix for the hidden state to the cell state and "w_{zbais}" is the bias term. In equation A.4 the "b_t" represents the cell state.

$$b_t = C_t \otimes i_t + b_{t-1} \otimes i_t \tag{A.4}$$

$$o_t = \sigma(A_t w_{xo} + H_{t-1} w_{Ho} + b_{t-1} w_{bo} + w_{obais}) \tag{A.5}$$

Where, "o_t" represents Output gate. Controls the output from the cell state.

$$H_t = o_t + tanH(b_t) \tag{A.6}$$

Where,"H_t" represents the hidden state.

Equations A.3,A.5, and A.6 are the standard formulas for output, forget gates, and hidden state. The "b_t", "H_t" represents output memory activation function at time interval t.

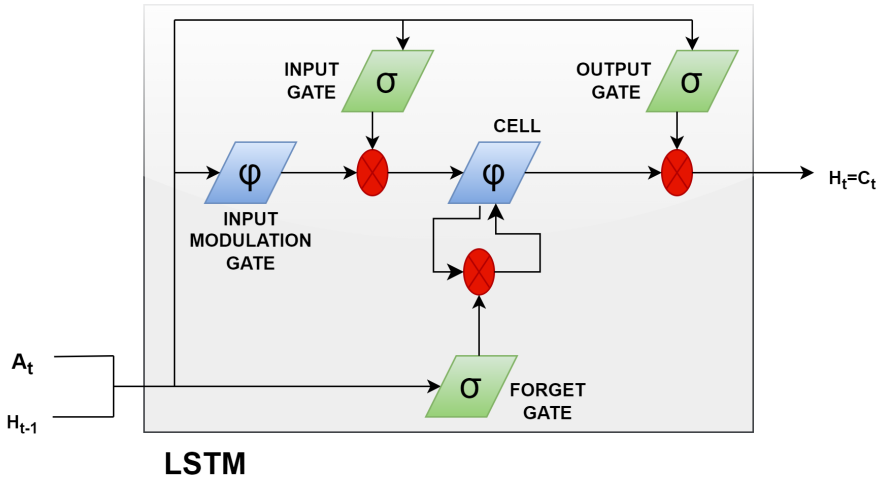


Figure A-1: LSTM Model.

A.2 Bidirectional-Long Short-Term Memory(Bi-LSTM)

Similarly, an example of a recurrent neural network is Bidirectional LSTM is frequently employed for sequential data processing applications like voice and natural language processing. The primary characteristic of Bi-directional LSTM is that it uses two different LSTM layers are used to process both the forward and backward directions of the input sequence as depicted in Fig. A-2. Concatenating the output of each layer results in the output feature string, which retains the both past and future context of each piece in the input pattern. Bidirectional LSTM, in contrast to LSTM, can comprehend movements captured before and after the present point as it can utilize forward information and backward information. Because of the flow of information in both directions, the bidirectional LSTM Long-term dependencies between signal patterns are captured. as compared to unidirectional networks, bi-directional LSTM is much superior [73].

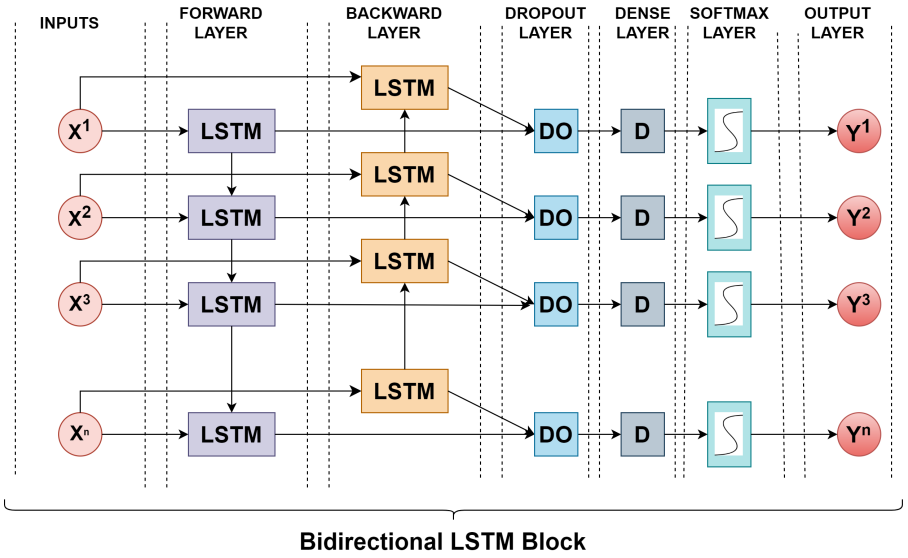


Figure A-2: Bidirectional LSTM Block.

Appendix B

VGG16, Densenet, Inception Net

This appendix explains the VGG16, DenseNet121, and Inception Net architectures in detail.

B.1 VGG16

Convolutional neural networks like VGG Net are very good at visual identification tasks since they were trained on the ImageNet dataset, which has over a million labeled images in different categories. There are two main variations of the network, however, VGG16 is one of the most widely used because of its simplicity and depth. The model's layers are frozen to prevent their weights from being updated during training. A flattened layer is added to convert the 2D feature maps to 1D feature vectors. A Dense layer with 256 units and ReLU activation is then included. The output layer is a Dense layer with the number of classes specified by num_classes and a softmax activation for classification.

VGG16 is a deep neural network with 16 layers, organized in a uniform structure with repeated blocks of convolutional and pooling layers, as shown in Figure **??**. It consists of 13 convolutional layers with 3x3 filters to capture patterns such as edges and textures, interspersed with 5 max-pooling layers that reduce computational complexity while preserving spatial information. The network concludes with fully connected layers that produce probabilities for class predictions, enabling strong

performance in image classification.

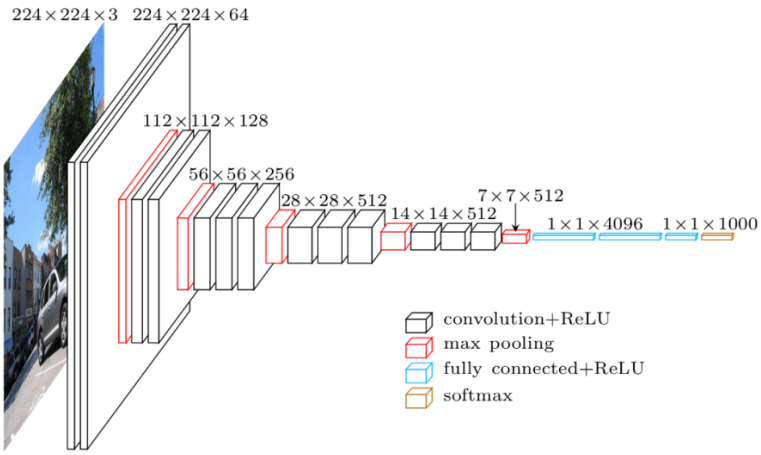


Figure B-1: VGG16 [124]

B.2 DenseNet

CNNs of the DenseNet (Densely Connected Convolutional Network) architecture are distinguished by their novel feed-forward connection strategy between each layer and every other layer. Its unique structure makes it possible to employ parameters more effectively and improves feature propagation, which makes it very successful for computer vision applications like segmentation and picture classification.

Each layer in DenseNet builds dense connections by passing its own feature maps to each succeeding layer and receiving inputs from all preceding layers. The network may access a greater variety of data from earlier layers because to this connectivity architecture, which removes the need to relearn redundant features. Because every layer in DenseNet has direct access to the feature maps produced by every layer before it, the architecture encourages feature reuse. When compared to conventional convolutional neural networks (CNNs), this feature reuse results in a significant reduction in the number of parameters. Furthermore, by improving gradient flow during backpropagation, the dense connections successfully address the vanishing gradient issue and facilitate the training of deeper networks. With substantially fewer parameters, DenseNet achieves great performance by optimizing feature reuse and decreasing

redundancy.

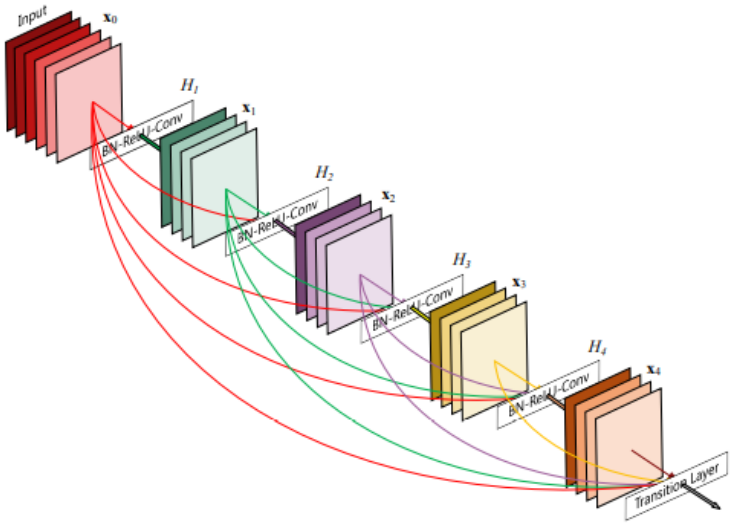


Figure B-2: DenseNet [88]

B.2.1 DenseNet121

A particular DenseNet setup with 121 layers is called DenseNet121. It is a popular variation that is extensively utilized in applications where accuracy and efficiency are crucial. DenseNet121 enhances the flow of information and gradients by connecting each layer to every other layer in a feed-forward fashion. Global Average Pooling2D is used to reduce the spatial dimensions of the fea- ture maps before the fully connected layers. DenseNet121 connects each layer to all previous layers, improving feature reuse and reducing the number of parameters. It consists of 4 dense blocks, each separated by transition layers that down sample the feature maps. Dense connection is achieved by concatenating each layer's output with the output of every preceding layer in the same block. In deep learning tasks, this design improves model performance and efficiency.

B.3 Inception Net

Google first unveiled the Inception network, commonly referred to as GoogLeNet, in 2014. It presented the idea of an Inception module, which enables the network to use numerous filter sizes inside the same layer to capture various information levels. This architecture is renowned for its great accuracy and computational efficiency on picture classification tasks, such as the ImageNet Challenge. The Inception module, which is the central component of the Inception network, uses several convolutional layers with varied sizes of filters to acquire various facets of a source picture. The filters are like 1x1, 3x3, 5x5. The network reduces dimensionality using 1x1 convolutions before applying larger filters, which lowers the number of parameters and preserves computational efficiency. It is feasible to employ deeper networks without using excessive computer resources by using smaller filters, which lower the total number of parameters and processing cost.

B.3.1 InceptionV3

An upgraded version of the original Inception architecture, Inception V3 added a number of improvements to increase efficiency and performance. By using several convolutions of various sizes in parallel, InceptionV3 is able to capture multi-scale information. The pre-trained weights are used by the model to load InceptionV3. In order to lessen overfitting and model size, it employs global average pooling rather than fully connected layers. The feature maps are transformed into a single vector per picture by a GlobalAveragePooling2D layer, which is followed by a Dense layer that contains the softmax activation and number of classes for classification. Inception V3 is a deep neural network that uses multiple parallel convolution layers (1x1, 3x3, 5x5) and pooling operations to efficiently capture features at different scales. It uses smaller, split-up convolutions to make the model faster and more efficient. It also adds extra classifiers during training to help the model learn better and prevent overfitting, making it very good at recognizing images.

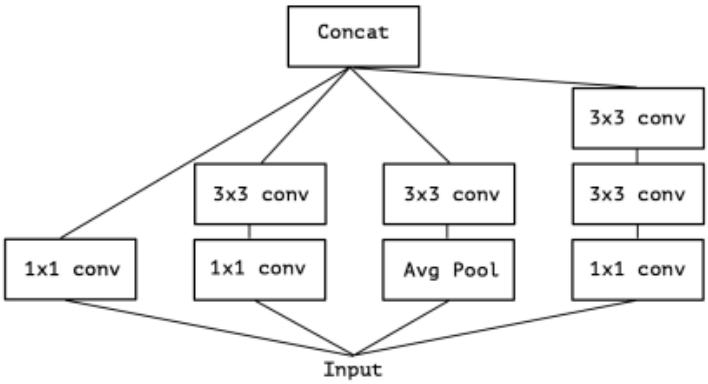


Figure B-3: Inception V3 [87]

Appendix C

Bidirectional Gated Recurrent Unit(Bi-GRU)

In this appendix, we provide detailed information on the Bi-GRU model, including its formulation and architecture.

C.1 Gated Recurrent Units (GRU)

Recurrent neural networks, Gated Recurrent Units (GRU), and Bi-directional GRU (Bi-GRU) are frequently utilized in applications involving sequential data processing and natural language processing. Cho et al. [108] introduced the GRU (Gated Recurrent Unit), a form of recurrent neural network, in 2014. The GRU cell is more computationally efficient since it has fewer parameters comparable to the LSTM cell. The GRU selectively updates the hidden state and memory cell using gating mechanisms as a result it solves the problem of vanishing gradients that can happen in conventional recurrent neural networks. The reset gate and the update gate are both components of the GRU cell. The update gate regulates the amount of fresh candidate activation utilized in the current time step, whereas the reset gate regulates the amount of the prior hidden state used in the current time step. Based on the current input and prior hidden layer output, the update gate utilizes a sigmoid neural layer to selectively add or delete information from the input. Equation C.1 is used to

determine the update gate's function.

$$U_T = \sigma((w^{(U)}X_T + B^U) + (q^{(U)}H_{T-1} + B^U)) \tag{C.1}$$

Where U_T represents as update gate, H_{T-1} as the output of the hidden layer. X_T is a current input that has been inserted into a network unit and multiplied by its own weight $w^{(U)}$ and biases are included, H_{T-1} is a prior time stamp information that has been multiplied by its original weight $q^{(U)}$ and biases.

$$R_T = \sigma((w^{(R)}X_T + B^R) + (q^{(R)}H_{T-1} + B^R)) \tag{C.2}$$

Similarly, in equation C.2, R_T represents the reset gate, which selects the exact amount of the prior information to forget.

In equation C.3 the reset gate R_T is used in the next steps to determine the memory content H'_T in order to obtain the necessary information from the past.

$$H'_T = tanh((wX_T + B) + R_T \odot (qH_{T-1} + B)) \tag{C.3}$$

$$H_T = U_T \odot H_{T-1} + (1 - U_T) \odot (H'_T) \tag{C.4}$$

The combined findings from both steps are applied in the last stage, followed by tanh activation and H_T is finding out to keep the most recent information and transmit it throughout the network. In Equation C.4 U_T is multiplied with H_{T-1} to determine what information needs to collect from the previous step.

C.2 Bidirectional Gated Recurrent Unit(Bi-GRU)

An extension of the GRU, known as the Bi-GRU (Bidirectional Gated Recurrent Unit), is an additional set of hidden states that are generated in the opposite direction. As a result, the model is able to include data from the input sequence's past and

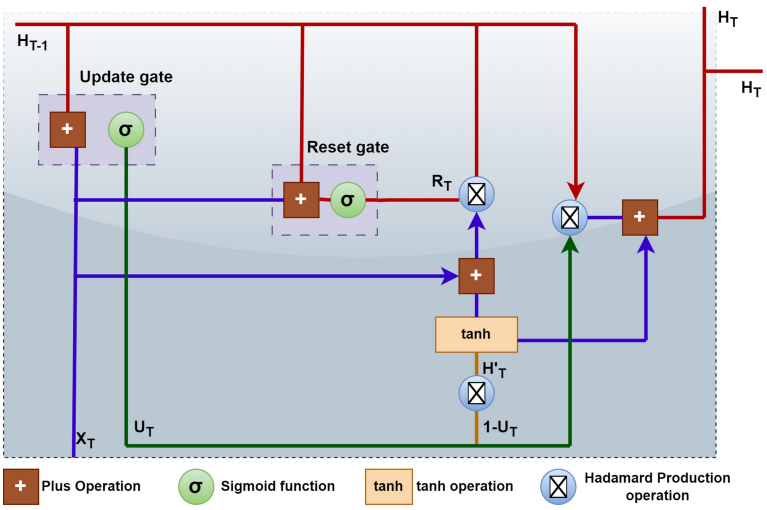


Figure C-1: GRU Architecture

future. The Bi-GRU cell equations are as follows:

$$\overrightarrow{H_T} = GRU_{fwd}(X_T, \overrightarrow{H_{T-1}}) \overleftarrow{H_T} = GRU_{bwd}(X_T, \overleftarrow{H_{T+1}}) H_T = \overrightarrow{H_T} \oplus \overleftarrow{H_T} \tag{C.5}$$

where $\oplus$ denotes the act of concatenating two vectors, $\overrightarrow{H_T}$ denotes the state of the forward GRU, and $\overleftarrow{H_T}$ denotes the state of the backward GRU.

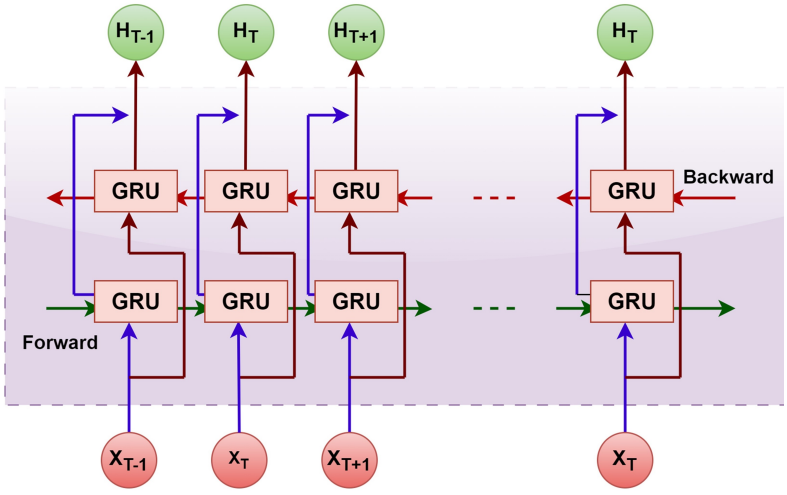


Figure C-2: Bidirectional-GRU Architecture

LIST OF PUBLICATION AND THEIR PROOFS

LIST OF JOURNALS

Journal Paper: 1

Tripathi, R. and Verma, B., 2023. Survey on vision-based dynamic hand gesture recognition. The Visual Computer, pp.1-29.
https://doi.org/10.1007/s00371-023-03160-x ***(Published, I.F=3)***

The Visual Computer (2024) 40:6171–6199
https://doi.org/10.1007/s00371-023-03160-x

SURVEY

Survey on vision-based dynamic hand gesture recognition

Reena Tripathi[1] · Bindu Verma[1]

Accepted: 29 October 2023 / Published online: 9 December 2023

Abstract
To communicate with one another hand, gesture is very important. The task of using the hand gesture in technology is influenced by a very common way humans communicate with the natural environment. The recognizing and finding pose estimation of hand comes under the area of hand gesture analysis. To find out the gesturing hand is very difficult than finding the another part of the human body because the hand is smaller in size. The hand has greater complexity and more challenges due to differences between the cultural or individual factors of users and gestures invented from ad hoc. The complication and divergences of finding hand gestures will deeply affect the recognition rate and accuracy. This paper emphasizes on summary of hand gestures technique, recognition methods, merits and demerits, various applications, available data sets, and achieved accuracy rate, classifiers, algorithm, and gesture types. This paper also scrutinizes the performance of traditional and deep learning methods on dynamic hand gesture recognition.

Keywords Dynamic hand gesture · Deep learning · Image Processing · Video processing · Classification · Survey on dynamic hand gesture

1 Introduction

In real life, hand gestures are very important in communicating with one another. The task of using the hand gesture in technology is influenced by the way humans communicate in the natural environment [1]. The user uses a keyboard, mouse, and pen to communicate with a computer. A similar type of communication can be possible using hand gesture that replaces the hardware devices and reduces the hardware cost. Earlier gloves and sensor-based trackers came into picture that are used to communicate with the computer, but they are not successful due to the cost of wearable devices. Moreover, users need to wear these devices, which hinders the naturalness of the hand gesture and makes it very uncomfortable to wear such devices. Then, vision-based hand gesture recognition comes into picture, where a user performs the hand gesture before the camera, and the corresponding action is triggered. The hand gesture is the way by which we give a signal to the computer system. It is a non-contact technique for giving input [2]. Hand gesture is of two types: (i) static hand gesture that contains hand's shape, fingers and palm and (ii) dynamic hand gesture that contains hand movement along with shape changes and has spatio-temporal information.

From the various studies, we witness that there are two methods used to interact between humans and computers using hand gestures, and these are:

- **Data Glove-based hand gesture:** In this process, a sensor is attached to the gloves by electric signals and hand postures are observed. This approach involves the physical connection of humans and computers via cables, as shown in Fig. 1. Data gloves have various advantages, like they obtain hand joint data and are suitable for small signal interference. The disadvantage of data gloves is that the user needs to wear these devices, which hinders the naturalness of the hand gesture and makes it very uncomfortable to wear such devices. Also, the cost and maintenance of the data gloves are high.
- **Vision-Based hand gesture:** In vision-based, data are captured through the camera, and gesture is performed in front of the camera. Data captured through the camera can be in the form of RGB data, depth data, skeleton data, and 3D landmarks data, as shown in Fig. 2. Then that data are processed, and the corresponding gesture is

✉ Bindu Verma
bindu.cvision@gmail.com

Reena Tripathi
8june.reena@gmail.com

[1] Department of Information Technology, Delhi technological University, New Delhi, India

Journal Paper: 2

Tripathi, R., Verma, B. Motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recognition. SIViP (2024). https://doi.org/10.1007/s11760-024-03153-w ***(Published, I.F=2)***

Signal, Image and Video Processing (2024) 18 (Suppl 1):S299–S308
https://doi.org/10.1007/s11760-024-03153-w

ORIGINAL PAPER

Motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recognition

Reena Tripathi[1] · Bindu Verma[1]

Received: 13 January 2024 / Revised: 21 February 2024 / Accepted: 13 March 2024 / Published online: 7 April 2024

Abstract
Dynamic hand gesture recognition continues to be an interesting field in computer vision applications. Occlusion and background clutter make dynamic hand gesture recognition challenging. In this study, we proposed two parallel pipelines. The first pipeline uses skeleton data to generate a skeleton point trajectory video where the fingertips are tracked across the frame and a trajectory video is created. The use of skeleton data overcomes the challenges of occlusion and complex background. Similarly, in the second pipeline optical flow videos are calculated from RGB/Depth data that capture the motion information of the moving hand. Creation of an optical flow video filters out irrelevant data and concentrates on the gesturing hand that helps in extracting spatio-temporal information. Then, features are extracted parallelly from both pipelines using pre-trained Xception-Net. The created feature vector is passed to the Bi-GRU unit for sequence-to-sequence learning. At the feature level, features of both Bi-GRU networks are averagely fused and flattened at the FC layer and the Softmax classifier is used to classify the gesture. We tested our proposed model on two benchmark datasets, namely NWUHG dataset and the DHG-14/28 dataset. The proposed model achieved 99.2% accuracy on NWUHG, 98.1% on DHG-14, and 94.2% on DHG-28, i.e., comparable with the state-of-the-art methods.

Keywords Dynamic hand gesture recognition · Bi-GRU · Video processing · Deep learning

1 Introduction

In real life, hand gestures are very important in communicating with one another. The task of using the hand gesture in technology is influenced by the way humans communicate in the natural environment. The user uses a keyboard, mouse, and pen to communicate with a computer. A similar type of communication can be possible using hand gesture that replaces the hardware devices and reduces the hardware cost. Hand gesture recognition is increasingly important for several applications, including robot control, sign language recognition, HCI, virtual game control, etc. To recognize the dynamic hand gesture, one needs to detect and track the gesturing hand, and then spatio-temporal feature extraction and classification of the dynamic hand gesture. Due to different lighting conditions, cluttered background, self-co-articulation, noisy images, and occlusion [1, 2] hand detection and tracking may not be accurate. Due to the aforementioned issues, it is very difficult to detect and track the gesturing hand. To make a robust gesture recognition system, hand detection and tracking steps must be performed flawlessly to propose a generic system. HandSNet and YOLO-H [3] are models excellent at solving issues like background interference and accurately localizing bare hands by merging detection and tracking. Moreover, a semantic segmentation model is implemented in the literature to overcome the above-mentioned challenges [4].

Numerous framework/real-time applications have been proposed to recognize dynamic hand gesture. In computer vision, various traditional and deep learning methods for hand gesture recognition have been proposed. In the traditional method, hand gesture recognition goes into various stages such as data pre-processing, hand segmentation and tracking [5–7], spatio-temporal feature extraction [6–9] and classification [5, 6, 10, 11]. In hand gesture classification, the traditional approaches set a milestone that fails only when the images are noisy and cluttered. In such a scenario, the hand

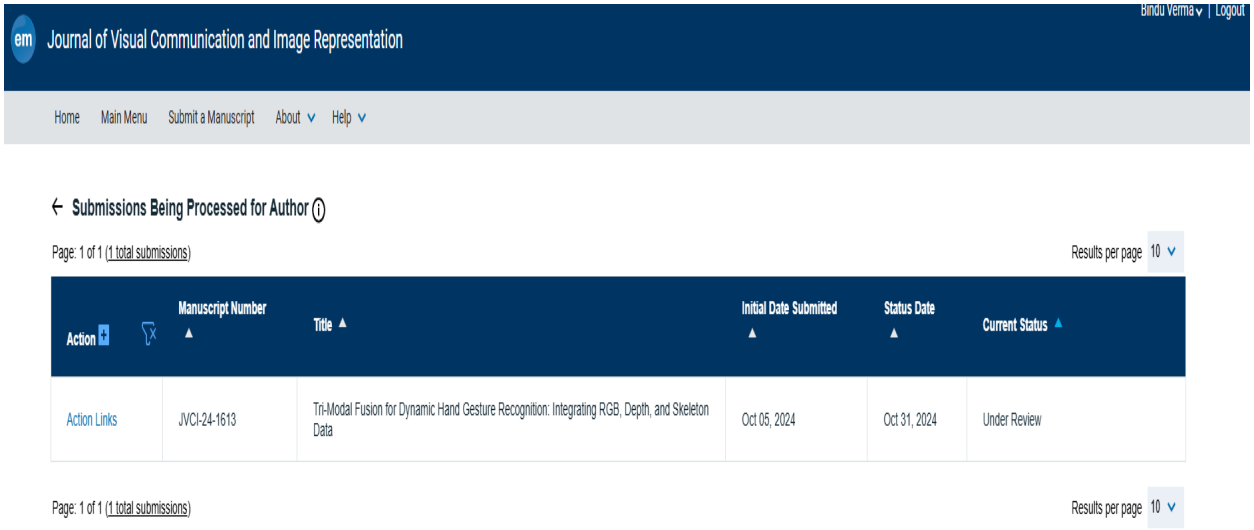
✉ Bindu Verma
bindu.cvision@gmail.com

Reena Tripathi
8june.reena@gmail.com

[1] Department of Information Technology, Delhi Technological University, Delhi, India

Journal Paper: 3

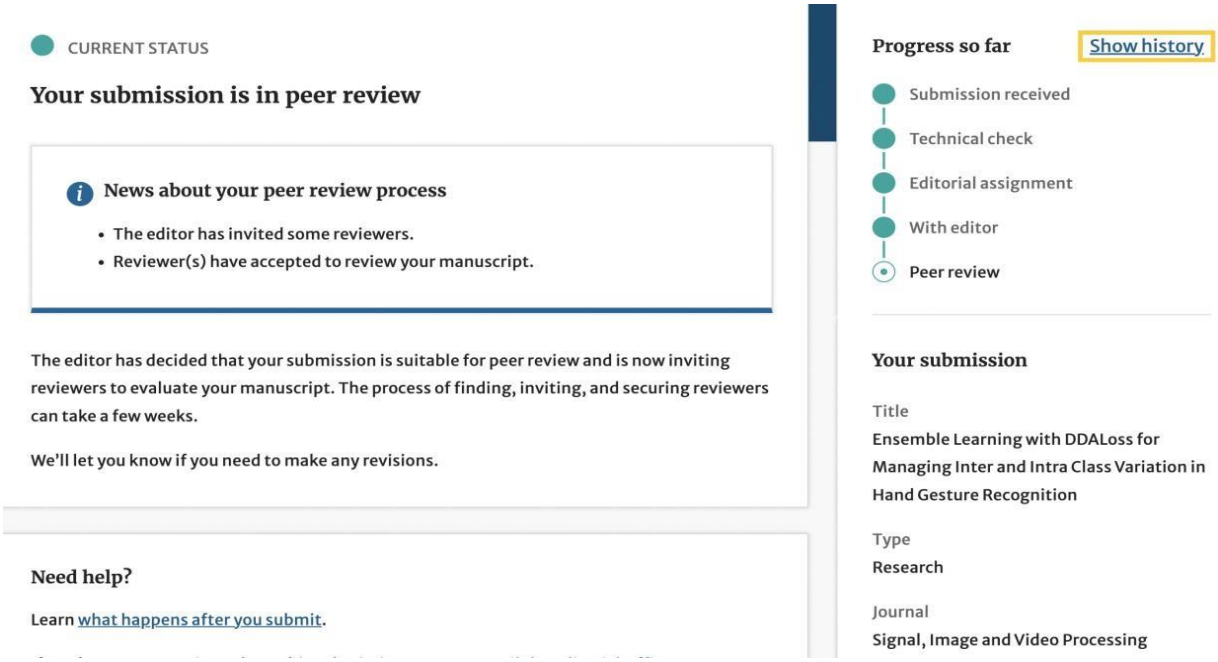
Reena Tripathi, and Bindu Verma. “Tri-Modal Fusion for Dynamic Hand Gesture Recognition: Integrating RGB, Depth, and Skeleton Data” is communicated in Journal of Visual Communication and Image Representation (SCIE Indexed, IF: 2.6) ***(Communicated)***



.....

Journal Paper: 4

Reena Tripathi, and Bindu Verma. “Ensemble Learning with DDALoss for Inter and Intra Class Variation in Hand Gesture Recognition” is communicated in Signal, Image and Video Processing. (SCIE Indexed, IF:2) ***(Communicated under revision)***



LIST OF CONFERENCES

Conference: 1

Tripathi, R. and Verma, B., 2023, December. CLIP-LSTM: Fused Model for Dynamic Hand Gesture Recognition. In 2023 IEEE 20th India Council International Conference (INDICON) (pp. 926-931). IEEE. 10.1109/INDICON59947.2023.10440820 ***(Published)***

Conferences > 2023 IEEE 20th India Council ...

CLIP-LSTM: Fused Model for Dynamic Hand Gesture Recognition

Publisher: IEEE | Cite This | PDF

Reena Tripathi ; Bindu Verma All Authors

114 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related Work

III. Proposed Work

IV. Experimental Analysis

V. Conclusion and Future Direction

Authors

Figures

References

Keywords

Metrics

Abstract:

The computer vision field continues to find dynamic hand gesture detection to be an intriguing subject. The algorithm is unable to determine accurately whether a gesture begins or ends in a video feed therefore, recognizing dynamic hand movements in real time is challenging. Real-time dynamic hand gesture detection has several applications, and numerous researchers are working on it. In this paper, we have used the CLIP model to extract the features of hand gestures. Then the extracted features passed into the BLSTM model to classify the dynamic hand gestures. Using the CLIP model for feature extraction overcomes the problem of hand detection and tracking. The various illumination makes hand detection and tracking challenging and CLIP is used to extract the features of each video. We conduct an experiment with fewer parameters on a challenging dataset. Experimental results on the CHG and LISA dataset with 97% accuracy on CHG and 86% accuracy on LISA datasets shows that our proposed model outperforms the state-of-the-art methods (SOTA).

Published in: 2023 IEEE 20th India Council International Conference (INDICON)

Date of Conference: 14-17 December 2023

DOI: 10.1109/INDICON59947.2023.10440820

Date Added to IEEE *Xplore*: 27 February 2024

Publisher: IEEE

ISBN Information:

Conference Location: Hyderabad, India

ISSN Information:

SECTION I.

Introduction

Conference 1: Certificate

Conference: 2

Tripathi, R. and Verma, B., 2023, November. Skeleton Data is all about: Dynamic Hand Gesture Recognition. In 2023 Seventh International Conference on Image Information Processing (ICIIP) (pp. 576-585). IEEE. 10.1109/ICIIP61524.2023.10537708 ***(Published)***

Conferences > 2023 Seventh International Co...

Skeleton Data is all about: Dynamic Hand Gesture Recognition

Publisher: IEEE | Cite This | PDF

Reena Tripathi ; Bindu Verma **All Authors**

49 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related Work

III. Skeleton Data in Dynamic Hand Gesture Recognition

IV. Extracting Skeleton Key Points

V. Recognition Methods and Performance Indicators

Show Full Outline

Authors

Figures

References

Keywords

Metrics

Abstract:

It's crucial to use hand gestures when communicating with one another: The task of utilizing hand gestures in technology is impacted by one of the most prevalent ways that people interact with their surroundings. Hand gesture analysis includes recognizing and position estimation of hands. Because the hand is smaller than other parts of the body, it is much harder to locate the pointing hand than other parts. The hand has greater complexity and more challenges due to differences between the cultural or individual factors of users and gestures invented from ad-hoc. The complications and divergences of finding hand gestures will deeply affect the recognition rate and accuracy. This paper emphasizes on summary of skeleton-based dynamic hand gesture technique, skeleton points extraction methods, merits and demerits, various applications, available data sets, and skeleton-based recognition methods. This paper also scrutinizes the performance of hand gesture recognition systems using skeleton data.

Published in: 2023 Seventh International Conference on Image Information Processing (ICIIP)

Date of Conference: 22-24 November 2023

DOI: 10.1109/ICIIP61524.2023.10537708

Date Added to IEEE *Xplore*: 28 May 2024

Publisher: IEEE

ISBN Information:

Conference Location: Solan, India

ISSN Information:

SECTION I.
Introduction

Conference 2: Certificate

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis: Dynamic Hand Gesture Recognition Framework for Human-Computer Interaction

Total Pages: 158

Name of the Scholar: Reena Tripathi

Supervisor: Dr. Bindu Verma

Department: Information Technology

This is to report that the above thesis was scanned for similarity detection. Process and outcomeare given below:

Software used: Turnitin **Similarity Index:** 9% **Word Count:** 39,588 Words

Date: 28/11/2024

(Dr. Bindu Verma)

Candidate's Signature **Signature of Supervisor**

reena Thesis

Ph_D_Thesis__Reena_Tripathi____Introduction_Future_work_ (V2).pdf

My Files

My Files

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:72490229

Submission Date

Nov 28, 2024, 8:52 AM GMT+5:30

Download Date

Nov 28, 2024, 9:06 AM GMT+5:30

File Name

Ph_D_Thesis Reena_Tripathi__Intro_Future_work_ (V2).pdf

File Size

13

158 Pages

39,588 Words

208,104 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Small Matches (less than 10 words)

Exclusions

- 2 Excluded Sources

Match Groups

174 Not Cited or Quoted 6%
Matches with neither in-text citation nor quotation marks

42 Missing Quotations 1%
Matches that are still very similar to source material

30 Missing Citation 1%
Matches that have quotation marks, but no in-text citation

Reena Tripathi (Ph.D. Student)

(Supervisor)

Reena Tripathi

Ph.D. Scholar, Department of Information Technology

Delhi Technological University(DTU), Rohini, Delhi, India- 110016

Email: 7june.reena@gmail.com, reenatripathi_2k20phdit02@dtu.ac.in

Contact Number: + 91 9013814040

Education

Delhi Technological University (DTU), Delhi | PURSUING — **Delhi, India**
PhD, Information Technology (*Department of Information technology*) — *2020 - Present*

Guru Govind Singh Indreprastha University (GGSIPU)| 75.04% — **Delhi, India**
M.Tech, Computer Science and Engineering — *2020*

Bhagat Phool singh Mahila Vishwavidyalaya (BPSMV)| 70.79% — **Haryana, India**
B.Tech, Electronic and Telecommunication — *2015*

Kendriya Vidyalaya JNU | — **Delhi, India**
12th Class(CBSE) — *2010*

Kendriya Vidyalaya JNU | — **Delhi, India**
10th Class(CBSE) — *2008*

Research Experience

Ph.D — **DTU Delhi**

***Advisor**: Dr. Bindu Verma* — *2020-Present*

***Dynamic Hand Gesture recognition*:**

M.Tech — **GGSIPU, Delhi**

***Advisor**: Dr. Rahul Johari* — *2017-2020*

Routing on SENSEnuts TestBuds in IoT Networks: *The research work, focused on the effort has been made to demonstrate the effective routing between nodes in an IoT drove wireless network on real-time testbeds of SENSEnuts nodes introduced by Eigen Technologies and the results have been positive and Encouraging.*

B.Tech **BPS University**

***Advisors**: Prof A L Vyas (IIT Delhi)* *2009-2013*
Major Project: Device Controller using DTMF
Minor Project: Code Locker, Embedded C, C++

Professional Experience

Worked as a Graduate Trainee Engineer in GAURI TELE COMMUNICATION *Feb, 2017*

Project, IIT Delhi, India *Nov, 2017*

Teaching Assistant(TA)/ Research Scholar, Department of Information Technology DTU *August 2020-Present*

International Journal

(1) Tripathi, Tripathi, R. and Verma, B., 2023. Survey on vision-based dynamic hand gesture recognition. The Visual Computer, pp.1-29.
DOI: https://doi.org/10.1007/s00371-023-03160-x | Impact factor: 3

(2) Tripathi, R., Verma, B. Motion feature estimation using bi-directional GRU for skeleton-based dynamic hand gesture recognition. SIViP (2024).
DOI: https://doi.org/10.1007/s11760-024-03153-w | Impact factor: 2

Communicated

(3) Tripathi, Reena, and Bindu Verma. "Ensemble Learning with DDALoss for Inter and Intra Class Variation in Hand Gesture Recognition" is communicated in Signal, Image and Video Processing (SCIE Indexed, IF:2) (*Submitted*).

(4) Tripathi , Reena, and Bindu Verma. "Tri-Modal Fusion for Dynamic Hand Gesture Recognition: Integrating RGB, Depth, and Skeleton Data" is communicated in Applied Intelligence (SCIE Indexed, IF: 3.5) (*Submitted*).

International/ National Conference

(1) Tripathi, Reena, and Bindu Verma. "Skeleton Data is all about: Dynamic Hand Gesture Recognition." In *2023 Seventh International Conference on Image Information Processing (ICIIP)*, pp. 576-585. IEEE, 2023.

(2) Tripathi, R. and Verma, B., 2023, December. CLIP-LSTM: Fused Model for Dynamic Hand Gesture Recognition. In 2023 IEEE 20th India Council International Conference (INDICON) (pp. 926-931). IEEE

(3) Johari, R., Kaur, I., Tripathi, R. and Gupta, K., 2020, October. Penetration testing in IoT network. In 2020 5th International Conference on Computing, Communication and Security (ICCCS) (pp. 1-7). IEEE.

Book Chapters

1) Johari, R., Tripathi, R., Kaur, I., Gupta, K. and Singh, R.K., 2020. ROSET: Routing on SENSEnuts Testbeds in IoT Network. In *IoT Security Paradigms and Applications* (pp. 43-67). CRC Press.

2) Tripathi, Reena, and Bindu Verma. "Comparative Analysis of PSO and WOA-Based Segmentation of Brain Tumor MRIs." In *Applied Intelligence for Medical Image Analysis*, pp. 43-58. Apple Academic Press, 2024.

Teaching Experience

Teaching Assistant (TA) DTU
Department of Information Technology *2020 - present*

Technical Reports and Project Work Experience

(1) Done six weeks of training in Project "Wireless Body Area Network for Health Monitoring" at Instrument Design Development Centre IIT delhi, under the guidance of Prof. A.L Vyas in year 2013.

(2) Six weeks of training in "Embedded System Technology". Organized by DUCAT Gurgaon, in year 2013.

(3) Six weeks Industrial training in MTNL under the Guidence of Jayant Chaudhary in year 2014.

Workshops Participations/Technical Talks/ Reviewer

(1) Worked with university placement and counseling cell (UPACC) as a class representative. Organized by Department of ECE in BPS Mahila Vishwavidyalaya, for session 2012-2013.| *Served as a volunteer.*

(2) Successfully completed the program "Embedded System Technology". Organized by DUCAT Gurgaon, 18th Dec, 2013- 2nd Feb, 2014.| *Participant.*

(3) "Embedded System Design and Simulation Tools". Organized by Department of Electronics and communication Engineering (ECE) in BPS Mahila Vishwavidyalaya ,17th February,2014.| *Participant / Organizer.*

(4) "INDUSTRY-ACADEMIACONCLAVE on Employability Skills: Bridge The Gape Between Industry & Academia". Organized by university placement & counseling cell in association with faculty of commerce & management studies, BPS Mahila Vishwavidyalaya ,11th -12th Mar,2014.| *Participant / Organizer/ Speaker.*

(5) Participated as student organizer during counseling camp, organized by university placement & counseling cell, BPS Mahila Vishwavidyalaya ,on 5th -6th Feb,2014.| *Participant / Organizer.*

(6) Successfully completed 06 weeks Industrial Training, organized by Institute of Telecom Technology & Management (ITTM), MTNL Sanchar Bhawan, Shadipur, Delhi ,with effect from 9th June,2014.| *Participant .*

(7) 2nd National Conference on "Machine Intelligence and Research Advancement". Organized by Department of Electronics and communication Engineering (ECE) in BPS Mahila Vishwavidyalaya ,on 19th -20th Mar,2015.| *Participant*

(8) "Intellectual Property Rights (IPRs) and IP management for Startup" Organized by Department of Information Technology in collaboration with IIC. Delhi Technological University (DTU), 23rd May, 2023.| *Served as a volunteer.*

(9) Recent Trends in Machine Learning and Deep Learning for AI applications. Member of organizing committee in one week short term course. Organized by Department of Information Technology in Delhi Technological University, 5th -9th June, 2023 | *Participant / Organizer.*

(10) 2nd International Conference on Communication, Security and Artificial Intelligence(ICCSAI). Technically co-sponsored by IEEE UP SECTION. Organised by Galgotias University, Greater Noida(U.P), 23rd -25th Nov, 2023 | *Reviewer.*

(11) 2nd International Conference on Communication, Security and Artificial Intelligence(ICCSAI). Technically co-sponsored by IEEE UP SECTION. Organised by Galgotias University, Greater Noida(U.P), 23rd -25th Nov, 2023 | *Member of Technical Program Committee.*

Awards and Honors

DTU Fellowship — *DTU, Delhi, India*
To pursue PhD — *2020 - Present*

Participation — *Kendriya Vidyalaya, JNU*
Received several certificates for different activities, including drawing, calligraphy, one-act plays, art and craft, and the Vivekananda Prashnottari Pratiyogita, achieving 1st, 2nd, and 3rd positions at the school level.

Best Guide Award — *Kendriya Vidyalaya, JNU*
Bharat Scout and Guide — *2003*

Gold Medal in Kumite — *Martial Art Association of Delhi*
4th All India Martial Art Championship-2004

3rd Position in Tackwondo — *Kendriya Vidyalaya Sangathan*
Regional Spoorts Meet 2004-2005

Certificate of successful participation — *Science Olympaid Foundation*
7th National Science Olympiad — *2004*

Leadership Experience

***Head of Fine Arts Committee**, BPS's University, Haryana, India* — *2014*

***Sub-Head of Fine Arts Committee**, BPS's University, Haryana, India* — *2013*

***Group Leader of Fine Arts committee,** BPS's University, Haryana, India* — *2013 - 2014*

***Team Leader** (Bharat Scout and Guide), KVJNU, Delhi, India* — *2003 - 2008*

***Head Girl**, KVJNU, Delhi, India* — *2008 - 2010*

Personal Details

Date of Birth:	*10-sep-1992*
Gender:	*Female*
Father's Name:	*Prof. D.P.Tripathi*
Mother's Name:	*Mrs. Savitri Tripathi*
Nationality:	*Indian*
Languages Known:	*Hindi, English*

References

Dr. Bindu Verma

(Assistant Professor)

Department of Information Technology (DTU), Delhi

Email: bindu.cvision@gmail.com

Dr. Rahul Johari

(Assistant Professor)

Guru Gobind Singh Indraprastha University

Sec-16 C, Dwarka, New Delhi-110078

Email: rahul@ipu.ac.in