Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy , Thrissur,
Kerala 679531

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

Mudra Classification

MAIN PROJECT REPORT

SAURAV MUNDANATT SATHEESH KUMAR	**(JEC17CS091)**
MUHAMMED AFTHAB V U	**(JEC17CS070)**
SANDRA DAVID	**(JEC17CS084)**
SRUTHI ELSA SHAJI	**(JEC17CS100)**

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Under the guidance of

Ms. NINU FRANCIS

JANUARY 2021

Department of Computer Science & Engineering

Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy , Thrissur,
Kerala 679531

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

Mudra Classification

MAIN PROJECT REPORT

SAURAV MUNDANATT SATHEESH KUMAR (JEC17CS091)
MUHAMMED AFTHAB V U (JEC17CS070)
SANDRA DAVID (JEC17CS084)
SRUTHI ELSA SHAJI (JEC17CS100)

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Under the guidance of

Ms. NINU FRANCIS

JANUARY 2021

Department of Computer Science & Engineering

Department of Computer Science and Engineering

JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY

THRISSUR 679 531

JANUARY 2021

BONAFIDE CERTIFICATE

This is to certify that the main project report entitled **Mudra Classification** submitted by **Saurav Mundanatt Satheesh Kumar (JEC17CS091), Muhammed Afthab V U(JEC17CS070), Sandra David(JEC17CS084) and Sruthi Elsa Shaji (JEC17CS100)** in partial fulfillment of the requirements for the award of **Bachelor of Technology** degree in **Computer Science and Engineering** of **A P J Abdul Kalam Technological University** is the bonafide work carried out by them under our supervision and guidance.

Ms. Ninu Francis	**Dr. Swapna B Sasi**	**Dr. Vinith R**
Project Guide	Project Coordinator	Head of The Dept.
Assistant Professor	Associate Professor	Associate Professor
Dept. of CSE	Dept. of CSE	Dept. of CSE

Jyothi Engineering College
NAAC Accredited college with NBA Accredited programmes*
Approved by AICTE & Affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKKATTIRI P.O., CHERUTHURUTHY, THRISSUR PIN-679531 PH: +91-4884-259000, 274423 FAX : 04884-274777
www.jecc.ac.in | E-mail : info@jecc.ac.in

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

COLLEGE VISION

Creating eminent and ethical leaders through quality professional education with emphasis on holistic excellence.

COLLEGE MISSION

- To emerge as an institution par excellence of global standards by imparting quality engineering and other professional programmes with state-of- the-art facilities.
- To equip the students with appropriate skills for a meaningful career in the global scenario.
- To inculcate ethical values among students and ignite their passion for holistic excellence through social initiatives.
- To participate in the development of society through technology incubation, entrepreneurship and industry interaction.

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

DEPARTMENT VISION

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

DEPARTMENT MISSION

- To create technically competent and ethically conscious graduates in the field of Computer Science & Engineering by encouraging holistic learning and excellence.
- To prepare students for careers in Industry, Academia and the Government.
- To instill Entrepreneurial Orientation and research motivation among the students of the department.
- To emerge as a leader in education in the region by encouraging teaching, learning, industry and societal connect

PROGRAMME OUTCOMES (POs)

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-Long Learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. The graduates shall have sound knowledge of Mathematics, Science, Engineering and Management to be able to offer practical software and hardware solutions for the problems of industry and society at large.
2. The graduates shall be able to establish themselves as practising professionals, researchers or Entrepreneurs in computer science or allied areas and shall also be able to pursue higher education in reputed institutes.
3. The graduates shall be able to communicate effectively and work in multidisciplinary teams with team spirit demonstrating value driven and ethical leadership.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. An ability to apply knowledge of data structures and algorithms appropriate to computational problems.
2. An ability to apply knowledge of operating systems, programming languages, data management, or networking principles to computational assignments.
3. An ability to apply design, development, maintenance or evaluation of software engineering principles in the construction of computer and software systems of varying complexity and quality.
4. An ability to understand concepts involved in modeling and design of computer science applications in a way that demonstrates comprehension of the fundamentals and tradeoffs involved in design choices.

COURSE OUTCOMES (COs)

C410.1 The students will be able to analyse a current topic of professional interest and present it before an audience.

C410.2 Students will be able to identify an engineering problem, analyse it and propose a work plan to solve it.

C410.3 Students will have gained thorough knowledge in design, implementations and execution of Computer science related projects.

C410.4 Students will have attained the practical knowledge of what they learned in theory subjects.

C410.5 Students will become familiar with usage of modern tools.

C410.6 Students will have ability to plan and work in a team.

ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all respected personalities who had guided, inspired and helped us in the successful completion of this interim project. First and foremost, we express our thanks to **The Lord Almighty** for guiding us in this endeavour and making it a success.

We take immense pleasure in thanking the **Management** of Jyothi Engineering College and **Dr. Sunny Joseph Kalayathankal**, Principal, Jyothi Engineering College for having permitted us to carry out this project. Our sincere thanks to **Dr. Vinith R**, Head of the Department of Computer Science and Engineering for permitting us to make use of the facilities available in the department to carry out the interim project successfully.

We express our sincere gratitude to **Mr. Shaiju Paul & Dr. Swapna B Sasi**, Project Coordinators for their invaluable supervision and timely suggestions. We are very happy to express our deepest gratitude to our mentor **Ms. Ninu Francis**, Assistant Professor,Department of Computer Science and Engineering, Jyothi Engineering College for her able guidance and continuous encouragement.

Last but not least, we extend our gratefulness to all teaching and non-teaching staff who directly or indirectly involved in the successful completion of this interim project work and to all friends who have patiently extended all sorts of help for accomplishing this undertaking.

ABSTRACT

Indian classical dance has been part of Indian Culture from around 200 BC. A *hasta mudra* is a symbolic gesture of hand which is used as a support for visual communication. This work inquires into the possibility of identifying hasta mudras in various classical dance forms of India. Not everyone are expertise in understanding the core idea presented in the classical dance forms and know what each pose portrays. The work attempts to find the feasibility of identifying the hasta mudras depicted in a classical dance form video and define its meaning. Here. machine learning and deep learning concepts are explored to frame the best solution. The Faster R-CNN is is used for the classification. Inception acts as the backbone of network. The core theme of the classical dance forms is rightfully conveyed to the viewer. The classical art forms could thus find a place in the minds of people. The viewer can appreciate the artist and glorify the art form. The Indian classical dance forms get proper recognition when more people get to appreciate the art forms.

CONTENTS

List of Figures

List of Abbreviations

CNN	: *Convolutional Neural Network*
SVM	: *Support Vector Machine*
HOG	: *Histogram of Oriented Gradients*
ReLU	: *Rectified Linear Unit*
MAT	: *Medial Axis Transformation*
GMM	: *Gaussian Mixture Model*
SSIM	: *Structural Similarity Index Measure*
ANN	: *Artificial Neural Network*
R – CNN	: *Region based Convolutional Neural Network*

(1)

CHAPTER 1
INTRODUCTION

1.1 Overview

Gesture recognition means the identification of different expressions of human body parts to express the idea, thoughts and emotion. It is a multi-disciplinary research area. The application areas of gesture recognition have been spreading very rapidly in our real-life activities including dance gesture recognition. Dance gesture recognition means the recognition of meaningful expression from the different dance poses. Today, research on dance gesture recognition receives more and more attention throughout the world. The automated recognition of dance gestures has many applications Instead of whole body movement, we consider human hands because human hands are the most flexible part of the body and can transfer the most meaning. The work focuses on the classification of hasta mudras from the video of an Indian classical dance.

Real-time recognition of dynamic hand gestures from video streams is a challenging task since there is no indication when a gesture starts and ends in the video. Th performed gestures should only be recognized once, and the entire architecture should be designed considering the memory and power budget. In the work, these challenges are addressed by exploring the possibility of identifying the hasta mudras in various dance forms.

Not everyone are expertise in understanding the core idea represented in the classical dance forms and know what each pose portrays. The work attempts to find the feasibility of identifying the mudras depicted in a classical dance form video and define its meaning. The core theme of the classical dance forms is rightfully conveyed to the viewer. The classical art forms could thus find a place in the minds of people. The viewer can appreciate the artist and glorify the art form. The heritage of India can be protected. Since the work seeks to identify and provide brief description on the hasta mudras of dance forms, the machine learning concepts are explored to frame the best solution. The first task is to capture the data images. As the area of consideration is video, there comes the need to extract images frame by frame from the video. The need for populating a dataset with the images of hasta mudras is essential to create an accurate model for classification.

Generally it is very difficult for a common man to understand the theme of various dances because of its complicated hand gesture language structure and dance movements. Since different combination of mudras in certain ways create complex meanings, it is difficult for one to understand and appreciate this art, unless one is well versed with these mudras. The classification attempt is implemented by Faster R-CNN. Inception acts as the backbone of network.

1.2 Objectives

The project is intended for users who wish to identify and learn the various Indian classical dance hasta mudras. The various actions include single hand mudra, double hand mudra, leg alignment, hip movement, eye movement, facial expression, and leg posture. Each dance form has unique gestures. The objective of the work is to identify the mudras depicted and grasp its meaning. The core theme of the classical dance forms is rightfully conveyed to the viewer. The viewer can appreciate the artist and glorify the art form.

1.3 Data Description

The video data of hand mudras of various classical dance forms are collected from internet. Then image segmentation and feature extraction are done followed by CNN algorithm and output is received as text messages. The data-set as a whole is generally divided into three categories: training, testing and validation. As in the case with a usual learning problem, training the model would be using pre-trained dataset and evaluating the performance with the test dataset.

1.4 Organization of the project

The report is organised as follow:

- **Chapter 1:Introduction** Gives an introduction to "Mudra Classification using Faster R-CNN".
- **Chapter 2:Literature Survey** Summarizes the various existing techniques that helps in achieving the desired result.
- **Chapter 3: Problem Statement** Discuses about the need for the proposed system
- **Chapter 4:Project Management** Contains the effective project management model to be used for the project.

- **Chapter 5:Proposed System** Describes the various steps involved to produce this project.
- **Chapter 5:System Requirements & Specification**Describes the various technologies needed for implementation.
- **Chapter 6:Conclusion** Concludes with the future scope of implementation.
- **References** Includes the references for the project.

CHAPTER 2
LITERATURE SURVEY

2.1 Classification of Kathakali Hand Gestures Using SVM & CNN

2.1.1 Abstract

Indian classical dance such as Kathakali is composed of complex hand gestures, body moments, facial expressions and background music. Due to the complexities involved in its hand-gesture language, it is often difficult to understand kathakali mudras.The paper proposed an SVM model and CNN model which classifies the images into 24 different classes.This work inquires into the possibility of identifying mudras & compare the performance of machine learning algorithms & deep learning algorithms.Results show that the deep learning algorithms gave up to 74% accuracy. To the best of our knowledge, this is the first attempt to generate a dataset of Kathakali hand gestures, explore data pre-processing techniques for machine learning techniques and applying deep learning techniques for classification of Kathakali hand gestures[2][6].

2.1.2 Methodology

- **Dataset preparation & data preprocessing**

  In this work,first build a dataset of kathakali hand gestures.Build a new dataset of 654 images of Kathakali hand gestures in which each mudra has 27 images. The images are taken under natural light and room light. The mudra images are taken by both left and right hands and from different persons by considering multiple factors including different positions,background color,hands by different people etc in generating dataset.For pre-processing of kathakali mudra classification,explore four feature extraction methods like Haar wavelet features,Histogram of oriented gradients[6],Contour extraction and canny edge detection[7].

- **Classification**

  SVM classification

  Proposing a multiclass SVM model to classify kathakali hand gesture image dataset.

SVM classifier will take features as input and classify those features into different classes. SVM classification can be done in two stages[6]. The first stage of SVM classification is pre-processing and feature extraction and the second stage is the classification stage. In the first step,tried four different methods such as Haar Wavelet features, Histogram of Oriented features, contour extraction and canny edge detection. The extracted features are given as the input to the SVM classifier and finally classifying the images using SVM classifier.

CNN classification

CNN is one of the best methods for a high level representation of image data[6]. CNN learns how to extract features from image pixel's data which has been given as input and tries to return the inference about pixels. CNN processes the input image and classify it in to different categories. Basically, each input image undergoes several convolutional layers of the CNN model. The convolutional layer contains filters, max or min pooling layers, Rectified linear unit (ReLU) layers and fully connected layers. Here designed a multi-layer Convolutional Neural Network model with two convolutional layers, two ReLU layers and two max pooling layers,used a batch normalization technique for calculating mean and standard deviation. The proposed CNN architecture is given below. In the first stage the images are undergone through preprocessing stage. In this stage the images are resized into 56*56*3. Resizing the input images will actually increase the computational capacity. After resizing, the images are then converted to gray scale images.

The second step is feeding the preprocessed images to the Convolution Neural Network. The CNN model will learn from the features that have been extracted from the input images. Out of total 654 images, 480 images (20 images of each of the 24 classes) are taken as training set and 168 images (7 images of each of the 24 classes) are taken as testing set of images. The training set of images is divided into training set (388 images) and validation set (96 images).

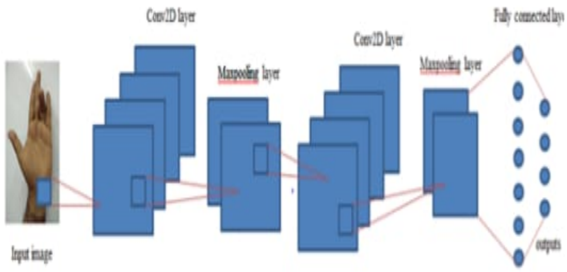

Figure 2.1: CNN Architecture

2.1.3 Results:

- SVM Results

  This paper explores four feature extraction methods such as Haar wavelet features,Histogram of oriented features,Contour extraction and canny edge detection.The SVM classification is performed in two steps. The first step is to perform pre-processing and extracting the features. The second step is to classify the images based on the features extracted by using SVM classifier[9].The result of haar wavelet feature extraction are given below.

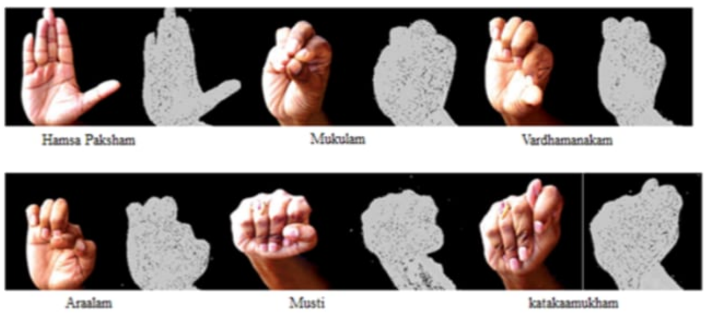

Figure 2.2: Haar wavelet features

The feature set of Histogram of Oriented Gradients are shown in below.

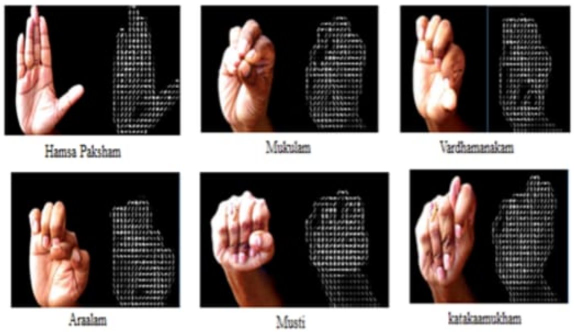

Figure 2.3: Histogram of Oriented features

This paper explored Canny edge detection and Contour extraction methods as edge detection methods for detecting edge features of dataset.The results of canny edge detection and contour extraction methods are shown in figure below.

Figure 2.4: Contour extraction & Canny edge detection

In case of Canny edge detection and contour extraction, instead of detecting edges, it is detecting many other lines and hence features are varying. Thus the classification method based on these features are not giving the best results.There are several parameters for analysing the SVM classifier such as precision, recall, F1-score and accuracy.The confusion matrix obtained for SVM classifier identifies that for 6 classes or mudras shows 100% which is the highest possible precision.The lowest precision is noticed for 4 classes.The rest of the classes have the average precision and the overall average accuracy is 39% .Based on these values ,we can clearly tell that SVM model fails in the kathakali mudra classification problem.

No	*Mudra*	*Precision*	*Recall*	*f1-score*
1	Pathaaka	0.60	0.60	0.60
2	Mudraakhyam	0.75	0.43	0.55
3	Katakam	0.25	1.00	0.40
4	Mushti	1.00	0.12	0.22
5	Kartharee Mukham	1.00	0.09	0.17
6	Sukathundam	0.17	1.00	0.29
7	Kapidhakam	0.33	0.25	0.29
8	HamsaPaksham	0.33	0.25	0.29
9	Sikharam	0.50	0.50	0.50
10	Hamsaasyam	0.17	0.20	0.18
11	Anjaly	0.75	0.50	0.60
12	Ardhachandram	0.29	0.29	0.29
13	Mukuram	0.29	0.40	0.33
14	Bhramaram	1.00	0.40	0.57
15	Soochimukham	1.00	0.17	0.29
16	Pallayam	1.00	0.25	0.40
17	Thripathaaka	0.09	0.50	0.15
18	Mrigaseersham	0.27	0.50	0.35
19	Sarpasirassu	0.50	0.25	0.33
20	Vardhamanakam	0.33	1.00	0.50
21	Araalam	0.33	1.00	0.50
22	Oornanabham	0.50	0.67	0.57
23	Mukulam	1.00	0.60	0.75
24	Katakaamukham	0.50	0.50	0.50

Figure 2.5: Confusion Matrix

- CNN Results

  In case of CNN architecture, The training images are divided into training and validation images. Among 20% of the training images are taken as validation set and train the model with these input images.As the number of epochs are increasing the training loss is decreasing because for each epoch the CNN model will acquire new set of features and classify the images based on them. The plot of validation loss is raising as the number of epochs are increasing because a new set of images are trained and tested by the CNN model.In case of CNN,we got an accuracy of up to 74%.

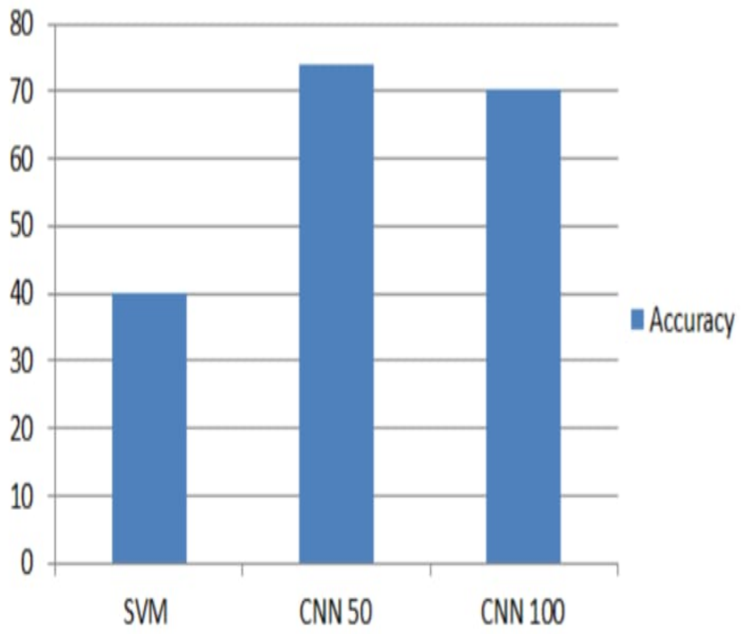

Figure 2.6: Comparison of SVM & CNN

2.1.4 Conclusion

This is an method for the classification of kathakali hand gestures using SVM & CNN.The features extracted from the kathakali image dataset are given to the SVM classifier as input .The classifier classifies the images into different classes of mudras and got an accuracy of 40%.In case of CNN,got an accuracy of 74%.This is the first attempt of this kind to create dataset for Kathakali images, identifying pre-processing techniques for machine learning classification[2] and classification of mudras using deep learning techniques. Several possible future works include, generating larger dataset, identifying mudras shown in both hands, interpreting the mudra meaning from a video, identifying mudras from video streams etc

2.2 Two-level classification scheme for single-hand gestures of Sattriya dance

2.2.1 Abstract

Hand gestures is an important factor of every dance forms.Every mudra contributes a specific meaning and the use of hand gestures is to project different emotions the dance requires.Mudras are of two kinds:single hand gestures(Asamyukta) & double hand gestures(Samyukta).This paper introduces a simple twolevel classification method for asamyukta hastas of Sattriya dance which is an Indian classical dance form. In the first level, twenty nine classes of hastas are categorized into three groups based on their structural similarity. Then, in the next level hastas are individually recognized from the database within the group. Moreover, the proposed method extracts Medial Axis Transformation (MAT) from the captured images to identify the groups in the first level[4].

2.2.2 Methodology

- Preprocessing:

  The preprocessing step is done in two sub steps: background removal and Gaussian filtering. Background removal is done by using GMM on RGB images. Then, a Gaussian filter approach has been used to make the images smooth and noise free. Thereafter, the RGB images are converted into gray images by using MATLAB function .

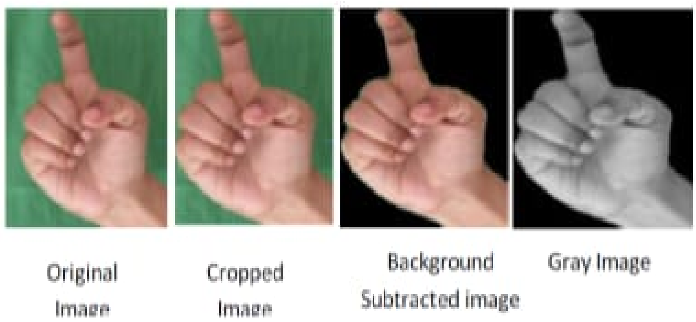

Figure 2.7: preprocessing

- Feature Extraction:

  The main aim of feature extraction is to transform the input image into set of numeric values which is also known as feature vector. The extracted features are used to find out the meaning of gestures. In this paper, geometrical features like centroid, eccentricity, orientation, bounding box, major axis length etc.. are used. Features are:

- **Centroid:**

  Centroid of the image can be defined as center of mass of the object

- **Eccentricity:**

The eccentricity feature of the image can be defined as the ratio of the distance between the foci of the object to its major axis length.

- **Orientation:**

  The orientation feature vector represents the angle (in degrees) between the x-axis and the major axis of the object.

- **Bounding Box:**

  The bounding box feature returns the smallest rectangle that can contain the object.

- **Major Axis Length:**

  The major axis length feature returns the longest diameter of the object.

- **Minor Axis Length:**

  The minor axis length feature returns the shortest diameter of the object.

- **Aspect Ratio:**

  The aspect ratio can be measured by finding the ratio of the width to the height of the object.

- **Perimeter:**

  The perimeter feature can be calculated by finding the distance between each adjoining pair of pixels around the border of the region.

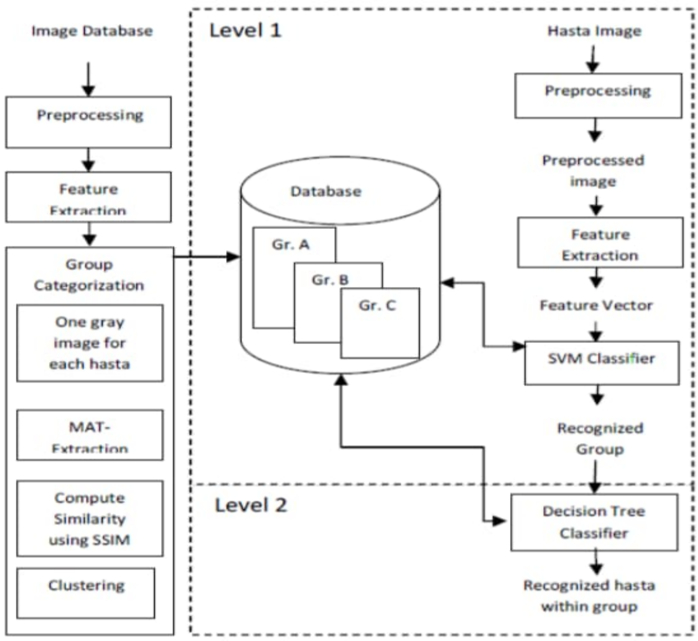

Figure 2.8: Work Flow Diagram

Grouping Similar Hastas

This section mainly focuses on the group categorization. To classify the hasta images in the database into three groups, the following steps are performed:

- Take one image for each type of hasta.
- Extract the Medial Axis Transformation (MAT) for each image.
- Apply Structural Similarity Index Method with window size 11X11.
- Create 29X29 similarity matrix and covert into distance matrix.
- Apply hierarchical agglomerative clustering algorithm on distance matrix.
- Draw a complete linkage dendrogram for clustering.
- Cut at threshold point as per requirement of number of group.

Working

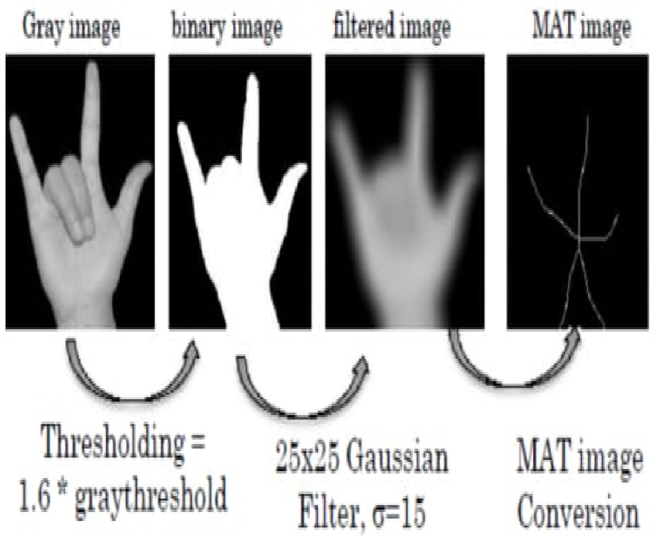

Figure 2.9: Steps for MAT extraction from image

- Initially, take one image for each hasta and extract its Medial Axis Transformation(MAT). The Medial Axis Transformation (MAT) finds out the closest boundary points for each point in an object and finally gives the skeletal of the images. To extract MAT, at first it convert the gray image to binary using threshold value determined by multiplying 1.6 with automatic gray threshold value.
- In the next step, apply 25x25 Gaussian filter with average mask sigma=15 to make the images smooth.
- The next step of this process is implementation of Structural Similarity Index (SSIM) Method. The SSIM works with a square window that moves pixel by pixel over the entire image and calculates the local statistics like mean intensity and standard deviation for each step of movement. The SSIM method is experimented on both gray image dataset and MAT image dataset of the Gaussian weighting function with varying window size of 8x8, 9x9, 10x10 and 11x11.
- Finally, from this experiment,come to conclusion that MAT image dataset is more robust than gray image dataset for group identification.

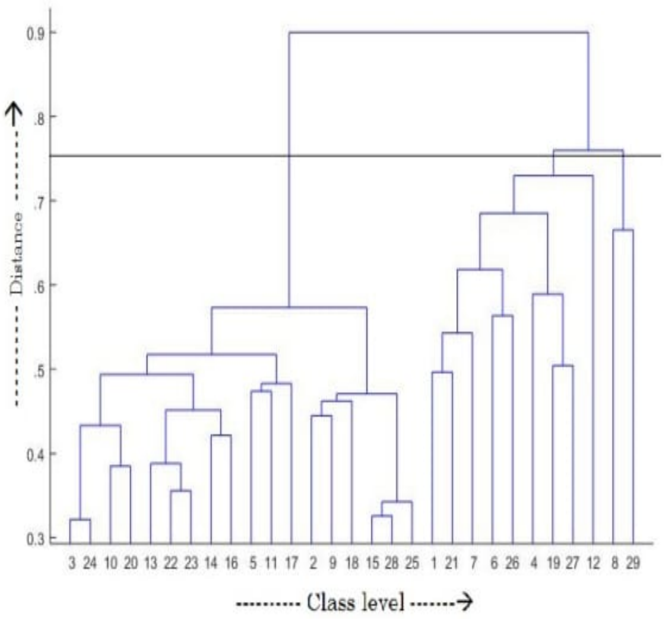

Figure 2.10: Creation of Groups on MAT- image dataset

Outcome of this experiment is 29x29 similarity matrix with values varying from 0-1. In this experiment, it was observed that the complete linkage gives the best clustering result.. Finally, to determine the groups, an optimal threshold point of 0.75 is chosen. At this threshold point, three groups are determined from the dendogram. After categorizing the twenty nine hastas into three groups, the datasets are trained for three group classification.

Classification

Support Vector Machine[6][9] is used for this first level classification. The SVM classifier is used as it gives better result compared to other classifiers. Similarly, decision tree classifier is used at the second level as best results are observed with this classifier at this level. In this level, the classification is narrowed down to the group to which the image matches the most as identified in the first level classification. And with the help of decision tree classifier the image is classified within its relevant group.

2.2.3 Results

The overall classification accuracy achieved for asamyukta hastas of Sattriya dance for first level classification is shown in figure.

Classifier	Total no. of Instances	Correctly classified instance	Accuracy (%)
K-nn(n=5)	1015	704	71.68
Bayesian Network	1015	639	62.95
Decision Tree	1015	818	80.59
Support Vector Machine	1015	987	97.24

Figure 2.11: First level classification

Among all the classifiers, SVM gives the best results with 97.24% accuracy using RBF kernel with 10 fold cross validation based on extracted feature sets. The kernel parameter C=15 and gamma=0.09 is obtained by varying C and gamma value within a range. So,here chosen a SVM classifier with RBF kernel at the first level classification.

	a	b	c
a	428	0	2
b	15	192	7
c	4	0	367

Figure 2.12: First level classification with SVM

After knowing the group at the first level classification, Decision tree classifier is used to recognize the Hasta at the second level. The recognition accuracies obtained at this level are: 71.62% for Group A, 75.39% for group B and 79.15% for group C. The average accuracy obtained at the second level classification is 75.45%.

Group	Classifier	Total no. of Instances	Correctly classified instances	Accuracy (%)
GroupA	Decision Tree	430	307	71.54
GroupB	Decision Tree	214	161	75.29
GroupC	Decision Tree	371	295	79.51

Figure 2.13: Second level classification

2.2.4 Conclusion

In this paper, a two-level classification method for recognition of single-hand gestures of Sattriya dance is proposed.In the first level,SVM is used to classify an unknown hasta image into one of three groups and at this level, 98% accuracy is achieved . In second level,decision tree classifier is used to recognize hasta within the group.At t, the observed recognition accuracies are 71.54%, 75.29% and 79.51% for group A, Group B and Group C respectively.

2.3 Artificial neural network based identification and classification of images of Bharatanatya gestures

2.3.1 Abstract

Here is an attempt to identify the hand gestures using Image-processing and also Pattern Recognition methods. An attempt in computer aided recognition of Bharatnatya Mudras is made using Image classification and processing techniques using Artificial Neural Network[1]. The entry that gives the least difference to the feature of a mudra is the match for the input image considered. Finally, the system also provides the health benefit of the identified mudra.By using mathematical algorithms, human gestures can be interpreted. This is referred to as Gesture Recognition[12].

2.3.2 Working:

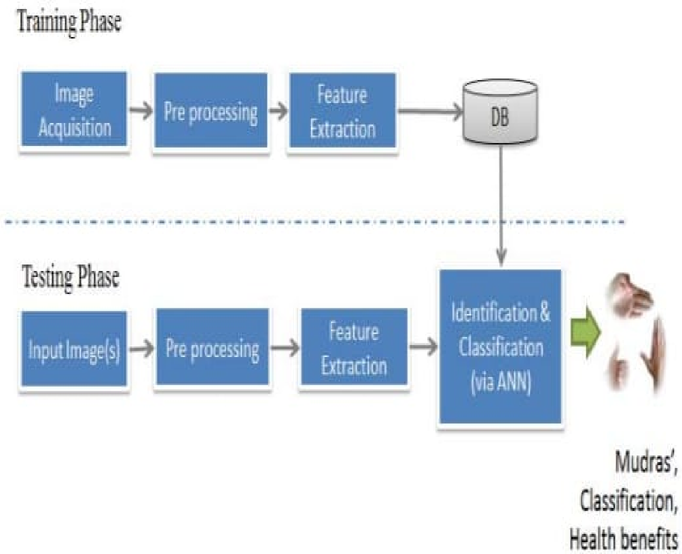

Figure 2.14: Architecture

This is an system which identifies and classifies the captured gestures using image processing techniques. Thus, main objectives are:

1. To collect the various forms of mudras.
2. To pre-process the images that is captured by applying image processing techniques.
3. To extract the features of the captured images.
4. To classify the images based on positioning of the fingers.

The following list of activities define the process in which the static recognition of gestures is carried out:

- Image capture
- Preprocessing
- Feature extraction

- Classifying

2.3.3 Proposed Methodology

The initial phase is to capture the static single and double hand gestures images using a standard camera system. In second stage, the images will be cropped and resized to a standard size of 113x150. Also the background is changed to black and white. In the next stage, perform feature extraction, in which the input data is transformed to get a set of features. The objective of extracting features is to understand what the images convey. The features aid in reliable recognition of the images. All the 28 shape features extracted are stored in a database in the upcoming phase. The same process is performed on every image in the database and the resulting features are appended to the database along with the meaning and properties calculated for each gesture. In the last stage, images are classified using ANN classifier[1].

Classification is the level where a feature vector or a set of features is used to identify the gesture by assigning it to a predefined class. To recognize the input gesture and to display the name and its health benefit is main task of the classification phase. Out of multiple techniques available to classify, here implemented a ANN technique.

2.3.4 Results

97.82% is the accuracy obtained as the result of applying precision formula. The following table represents the entire result analysis which consists of the total image set, the samples that are correctly classified, the ones that are not and accuracy of the system.

Sample Size	No. of Samples accurately classified (Tr.p)	No. of samples Misclassified (Fa.p)	Accuracy in %
230	225	5	97.82%

Figure 2.15: Result Analysis

The result for identification of single-handed mudras is 98.46and for double-handed mudras is 96%. On an average, the result for identification for the whole system is 97.82%.

2.3.5 Conclusion

This paper presents an ideal approach to represent the classification and recognition of different hand gestures of the Bharatnatya dance form. Image processing techniques is used to classify and recognise the gesture collected in the database. Then the ANN classifier is used to classify the images and display the name and health benefits of the hand gestures. The system is designed to work well with both single and double handed gestures. The system has an advantage of being used to teach and improve young dancers.

2.4 Bharatanatyam hand gesture recognition using polygon representation

Bharatanatyam' is one of the popular dances and widely performed in world-wide. It originated from Tamil Nadu, a south Indian state. This traditional dance form signifies elegance, purity and grace. An essential element of 'Bharatanatyam' is its 'Hastas' (hand 'mudras' or movements). 'Hastas' signifies different hand symbols that a dancer use to communicate. 'Hastas' can be of two types 'Asamyukta Hastas' (single hand) and 'Samyukta Hastas' (double hand). 'Asamyukta Hastas' consists of 28 mudras and 'Samyukta Hastas' consists of 24 mudras.With the increasing popularity of internet, it is very much useful to identify hand gestures of the 'Bharatanatyam' dance using computers.

The objective of this paper is to develop a simple system for promoting e-learning of 'Bharatanatyam' dance.Thus the 'Bharatanatyam' learning becomes an interactive program between the dancer and the computer[8].Also e-learning is a fast and cost-effective way to spread 'Bharatanatyam' dance world-wide. Flexibility and accessibility are the other two important advantages of elearning.In this proposed algorithm, we have designed a four level system.

2.4.1 Boundary Extraction

One of the important aspects of gesture recognition is to identify the hand of the dancer from the background. In , the authors are using skin-color based segmentation for separating the hand from the background. But it has two drawbacks already stated. So texture base segmentation is applied to eliminate unnecessary information about the background.The input RGB image is first converted to grey-scale image. Here the values are in the range of 0 (for black) and 1 (for white). Then we apply texture based segmentation in this image and converted to binary image (BW) by calculating entropy.

For texture based segmentation entropy calculation is used . This stage is the first preprocessing stage for hand gesture recognition. It requires finite neighbourhood for processing. First the gray-level co-occurrence matrix is obtained using a displacement vector d. Here it is specified as (1,1), which means one pixel to right and one pixel below.

$$d = (dx, dy) = (1,1)$$

P[i,j] is not necessarily symmetric. Each element of P[i,j] is divided by total no of pixel pairs to obtain normalize value. This normalized P[i,j] is known as probability mass function. Entropy is calculated based on P[i,j].

$$Entropy = -\sum_i \sum_j P[i,j] \log p[i,j]$$

2.4.2 Straight Line Approximation

The boundary of the hand gesture is a curved line, but for generation of chain code by polygon representation, we need a straight line approximated boundary . For this reason, the two end points of the image are extracted. Then we start from one end point (which is at the left side from the other end point) and its adjoining 8 pixels are examined. This continues until the other end point is reached. The entire boundary is represented using connected straight lines. This is done by first calculating the slopes of adjacent pixels and then detecting the point where there is a change of slope. Then these points are connected using straight lines. Thus we get a straight line representation of the image where the shape of the boundary is kept intact.

Our goal is to represent the boundary using minimum possible straight lines so as to reduce the noise by keeping the shape of the image undistorted. To smooth the image further, at first those points which are very near to each other, i.e., those points which are joined by straight lines having very small length are discarded, i.e., straight lines having length less than a particular threshold value are discarded. This value is chosen such that it discards maximum possible straight lines without affecting the shape of the boundary. In this case the threshold value is empirically chosen as 15. Next the slope of each straight line is determined. Starting from the first straight line, its slope is compared to its adjacent connected straight lines. If the difference between the slopes is within a particular range then the latter is discarded and the next straight line is compared to the first one. Here also, the range is chosen such that it discards the maximum possible number of straight lines keeping the shape of the boundary more or less intact. In this case the range is taken to be -40o to +40o, which is also determined experimentally.

2.4.3 Polygon Representation and Chain Code Generation

Arbitrary geometric structures can be encoded by a number of ways, each having its own particular advantages and disadvantages. One such method is to represent the straight lines by the edges of a regular polygon. Here we are using regular decagon to generate chain code. More will be the side of the regular polygon; complexity of the code will be more, thus increasing timing complexity. But with lesser no of sides of regular polygon, information about different slopes will be lost. The straight lines which have a large difference in slope will be approximated by same side of polygon. Thus we are taking 10-sided polygon for representing the straight line approximated boundary by a chain code.

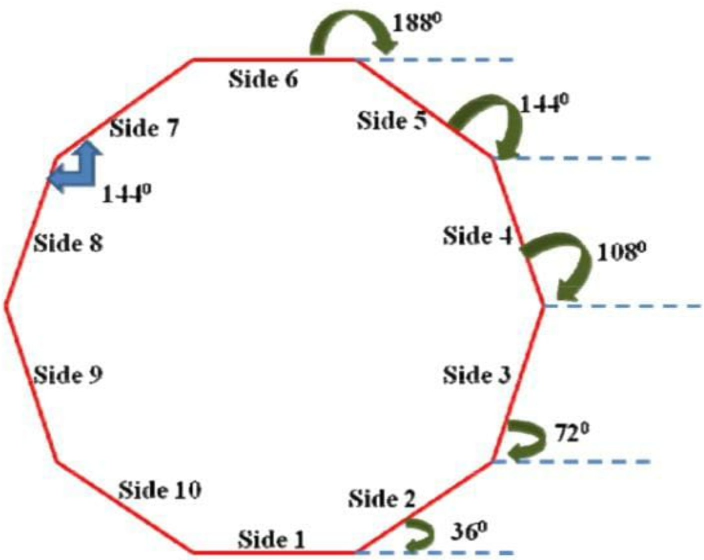

Figure 2.16: Decagon showing internal and external angles

2.4.4 Matching of Hand Gestures

Whenever, an unknown posture is detected, its boundary is calculated and chain code using polygon approximation is produced. Now the chain code is scaled to obtain equal length code like the plots from the database. For scaling purpose, all the code lengths are made to 31. As we have found the largest code length is 31 for all the experimented images. If a code obtain is of length x, then (31x) no of zeros are added to the end of the code to form equal length chain codes. The unknown code is matched with all the 28 codes from the database. The difference of the chain codes is taking as the error function. The best matched gesture is having least error value. Now this best matched gesture is taken as the output.

$$S = \arg\min \forall_{p=28} \left[\sum_{q=1}^{31} \{ Code_{unknown} - Code_{known} \} \right]$$

where S is the similarity function, which returns argument based on unknown code (Codeunknown) and known code (Codeknown) for 28 hand primitives already present in the database. 31 is the maximum code length possible, which is known empirically.

2.4.5 Proposed Algorithm

Step 0 Create an initial database with polygon represented chain codes, each having 31 numbers for the 28 basic hand gesture primitives of 'Bharatanatyam'.

BEGIN

Step 1A Apply texture based segmentation on the unknown hand gesture.

Step 1B Detect edge by Sobel edge detector.

Step 1C Smooth uneven thickness of the boundary using morphological shrink operation.

Step 2A Obtain two endpoints of the boundary.

Step 2B Start the straight line approximation code from the end point which has less column value.

Step 2C Remove the straight lines of the boundary which are less than 15 pixels, keeping the shape intact.

Step 2D Discard the straight lines which slopes are in the range -400 to +400, make them a single straight line. After this we obtain straight line approximated boundary with less no of straight lines.

Step 3A Segment the straight lines with segment length 30 pixels.

Step 3B Take regular decagon for generation of chain code.

Step 3C Calculate the interior angle of the polygon.

Step 3D Determine the polygon represented chain code for each straight line.

Step 4A Form equal length chain codes 31 length by zero-padding. Step 4B Calculate the similarity operator for 28 hand primitives already present in the dataset.

Step 4C Best match gesture is taken as the output, i.e., the code having minimum error value.

END

2.5 Heterogeneous hand gesture recognition using 3D dynamic skeletal data

2.5.1 Abstract

Hand gestures come to us naturally. These are the easiest and quickest way of non-verbal communication. The hand gesture communication is effective for both human to human interaction as well as human-computer interaction (HCI). Hence, the field of hand gesture recognition has great relevance. Hand gesture recognition has gradually become an active field of research. The field of study opens up several applications in different aspects of the life. Thus, hand gesture recognition systems urged popularity, as an inspiring field of research. A number of papers have been published and a variety of approaches have been proposed, regarding the subject. Advances in the technology have promoted and improved the approach of 3D hand gesture detection and recognition. The paper aimed to explore the possibility of hand gesture recognition using 3D dynamic skeletal data of hand. The approach relies on the structure of hand topology to extract the effective descriptors from the gesture sequence. The statistical representation method, Fisher Vector representation and the temporal representation method, temporal pyramid are used for the encoding of the descriptors. Finally, an SVM classifier is used for the classification of the gesture.

2.5.2 Methodology

The hand occupies relatively a small portion of our body, but has a complex topology. There are endless possibilities for performing a gesture. Some gestures are identified by the hand shape whereas, some other by the hand motion. In the static approaches, we mostly consider the hand silhouette from a single image. Unlike static approaches, dynamic approach defines the gesture as a sequence of hand shapes, considering the aspects of hand motion. It helps to describe both the motion and the hand shape of the gesture. The first step is to capture the gesture using motion capture devices. The recently popular devices such as, Intel Real Sense or Leap Motion Controller provides precise skeletal data of hand with 22 joints. Then comes the role of feature extraction to extract the most appropriate features from a whole set of features. A statistical representation method, FV and temporal representation method, TP are used to simplify complex calculations and improve performance. The classification is performed using an SVM classifier.

- **Capturing 3D Gestures**

  Motion capture devices which have sensors attached to a glove are the most reliable tools for capturing 3D hand gestures. But they have some drawbacks like cost, naturalness of hand, etc. The effective and inexpensive depth sensors, like the Microsoft Kinect, are popular today. Depth images also offer a great opportunity for the purpose. Shotton et al.[11] proposed a method to predict the 3D positions of 20 body joints, from depth images, called body skeleton. The most recent advancement is the device, such as Intel Real Sense or the Leap Motion Controller (LMC). This provides precise information of the full 3D skeleton corresponding to 22 joints. Since there are different possibilities for obtaining the precise skeletal data, it is an excellent choice to rely upon.

Figure 2.17: 22 Joints in hand

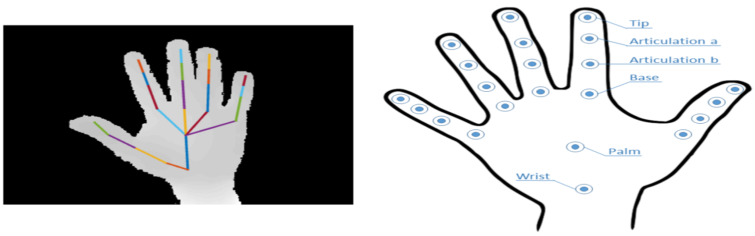

Figure 2.18: Depth and hand skeletal data returned by the Intel Real Sense camera

A dynamic gesture is a time series of hand skeletons. The hand shape and its motion is described along these series of input.

- **Feature Extraction**

  Motion Features Some gestures, for instance swipes are expressed almost only by the way in which the hand moves in space. In such cases, we define a direction vector for each time frame in the sequence. To take into consideration how the hand moves during the gesture, the rotation of thewrist is analyzed. For each time frame, to get the rotational information a rotational vector from wrist node to palm node is computed.

- **Hand Features**

  A descriptor based on joints is defined to represent shape of hand. It is called as Shape of Connected Joints (SoCJ). Here, a normalization phase is considered. Firstly, in order to take into account the differences of hand size between performers, we estimate the average size of each bone of the hand skeleton using all hands in the dataset. Firstly, using all hands in the dataset, an average of size of each bone is estimated. Secondly, the size is changed by their respective average size but keep the angles between bones unchanged. It is important that the estimations are consistent with the translation and rotation transformation. Hence, consider a reference hand Hf with joints [0 0 0], called as root joint with its front facing the camera. The input image is translated and rotated so as to align it with the reference skeleton. This results in a new hand which is in reference to the base hand with joints [0 0 0]. The translation and the rotation with reference to the base hand is computed on for only the first hand of the sequence of the gesture. This calculation is applied to all other skeletons in the sequence of hand skeletons in a gesture sequence.

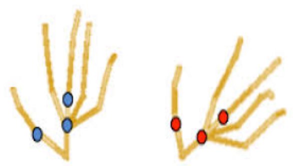

Figure 2.19: Before translation and rotation

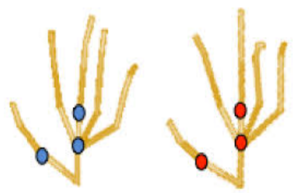

Figure 2.20: After translation and rotation

The use of Intel Real Sense camera provides with information of 22 joints of hand skeleton S. Theoretically, C(22,5)= 26334 different SoCJs for hand skeleton can be calculated, where C is a binomial coefficient.

- **Statistical Representation**

The Fisher Vector (FV) coding method is relied upon for the purpose of statistical representation of data. The FV method was meant for large scale image classification when it was first introduced. Its superiority against the Bag-Of-Word (BOW) method has been analyzed in the image classification [10].

Fisher Vector encoding method characterizes a sample by its deviation from the generative model GMM.The deviation is measured by computing the gradient of the sample loglikelihood with respect to the model parameters (w,m,s)[13]. Gaussian mixture models are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning [5].

- **Fisher Representation**

  Now comes the stage to take into consideration the dynamic nature of hand. The use of a Temporal Pyramid (TP) representation is relied. The principle of the TP is to divide the sequence into a number of sub sequences. Suppose the pyramid has n levels. Each i th level of the pyramid will have i sub sequences. The three descriptors and their respective statistical representations are calculated for each sub sequence and the results are concatenated. More number of levels to the pyramid guarantees more temporal precision but increases computing time.

- **Classification**

  Classification is a predictive modelling problem where the model predicts the class of a random input. The classification process suggested here is through an SVM classifier. In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis [14]. SVM with a linear kernel is used as the data high dimensional. A one-vs-rest strategy of SVM is used. That is, classifying a multi classification problem using a binary classification strategy. We group classes as one class and the second group with all left classes.

2.5.3 Result

The The Intel Real Sense short range depth camera can be used to populate the dataset. Each frame gives coordinates of 22 hand joints in the camera space. To encode the descriptor, there is a need to fix the number of levels in temporal pyramid. For a hand skeleton with 22 joints, one can compute 26334 joints, But using all of them is not mandatory. The task is to choose the feature set as combination of the most relevant SoCJs. The first step is to intuitively select a SoCJ set anf then use a selection algorithm called Sequential Forward Floating Search (SFFS).

The next task is to perform the tasks step by step.

The approximate accuracy percentage in the recognition system with and without FV and TP are given in the table below.

Features + SVM	76.89
Features + FV + SVM	79.76
Features + TP + FV + SVM	86.86

Figure 2.21: The approximate accuracy

Each additional step in approach is to improve the accuracy of recognition. Here, the use of statistical representation method Fisher Vector and temporal representation method Temporal Pyramid improves the efficiency of the system.

CHAPTER 3
PROBLEM STATEMENT

The project "Mudra classification" aims to identify and provide brief description on the hasta mudras of dance forms. This work is done using Faster R-CNN & Inception V2 architecture. Not everyone are expertise in understanding the core idea represented in the classical dance forms and know what each pose portrays. The work attempts to find the feasibility of identifying the mudra depicted in a classical dance form and defining its meaning. The classical art forms could thus find a place in the minds of people. Generally it is very difficult for a common man to understand the core idea of various dance forms, because of its complicated hand gesture language and body movements.

CHAPTER 4
PROJECT MANAGEMENT

4.1 Introduction

Project management is the discipline of planning, organizing, securing, managing, leading, and controlling resources to achieve specific goals. A project is a temporary endeavor with a defined beginning and end (usually time-constrained, and often constrained by funding or deliverables), undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with business as usual (or operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the management of these two systems is often quite different, and as such requires the development of distinct technical skills and management strategies.

In our project we are following the typical development phases of an engineering project

1. Initiation
2. Planning and Design
3. Execution and Construction
4. Monitoring and Controlling Systems
5. Completion

4.1.1 Initiation

The initiating processes determine the nature and scope of the project. The initiating stage should include a plan that encompasses the following areas :

1. Analysing the business needs/requirements in measurable goals
2. Reviewing of the current operations
3. Financial analysis of the costs and benefits including a budget
4. Stakeholder analysis, including users, and support personal for the project

5. Project charter including costs, tasks, deliverables, and schedule

4.1.2 Planing and design

After the initiation stage, the project is planned to an appropriate level of detail (see example of a flow-chart). The main purpose is to plan time, cost and resources adequately to estimate the work needed and to effectively manage risk during project execution. As with the initiation process, a failure to adequately plan greatly reduces the project's chances of successfully accomplishing its goals.

- Determining how to plan
- Developing the scope statement
- Selecting the planning team
- Identifying deliverables and creating the work breakdown structure
- Identifying the activities needed to complete those deliverables
- Developing the schedule
- Risk planning

4.1.3 Execution

Executing consists of the processes used to complete the work defined in the project plan to accomplish the project's requirements. The execution process involves coordinating people and resources, as well as integrating and performing the activities of the project in accordance with the project management plan. The deliverables are produced as outputs from the processes performed as defined in the project management plan and other frameworks that might be applicable to the type of project at hand.

4.1.4 Monitoring & controlling

Monitoring and controlling consists of those processes performed to observe project execution so that potential problems can be identified in a timely manner and corrective action can be taken, when necessary, to control the execution of the project. The key benefit is that project performance is observed and measured regularly to identify variances from the project management plan.

4.2 System Development Life Cycle

The Systems development life cycle (SDLC), or Software development process in systems engineering, information systems, and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system.

The SDLC phases serve as a programmatic guide to project activity and provide a flexible but consistent way to conduct projects to a depth matching the scope of the project. Each of the SDLC phase objectives is described in this section with key deliverables, a description of recommended tasks, and a summary of related control objectives for effective management. The project manager must establish and monitor control objectives during each SDLC phase while executing projects. Control objectives help to provide a clear statement of the desired result or purpose and should be used throughout the entire SDLC process.

4.2.1 Spiral Model

We have used the Spiral model in our project. The Spiral model incorporates the best characteristics of both- waterfall and prototyping model. In addition, the Spiral model also contains a new component called Risk Analysis, which is not there in the waterfall and prototype model. In the Spiral model, the basic structure of the software product is developed first. After the basic structure is developed, new features such as user interface and data administration are added to the existing software product. This functionality of the Spiral model is similar to a spiral where the circles of the spiral increase in diameter. Each circle represents a more complete version of the software product. The spiral is a risk-reduction oriented model that breaks a software project up into main projects, each addressing one or major risks. After major risks have been addressed the spiral model terminates as a waterfall model. Spiral iteration involves six steps:

1. Determine objectives, alternatives and constraints.
2. Identify and resolve risks.
3. Evaluate alternatives.
4. Develop the deliverables for the iteration and verify that they are correct.
5. Plan the next iteration.

6. Commit to an approach for the next iteration.

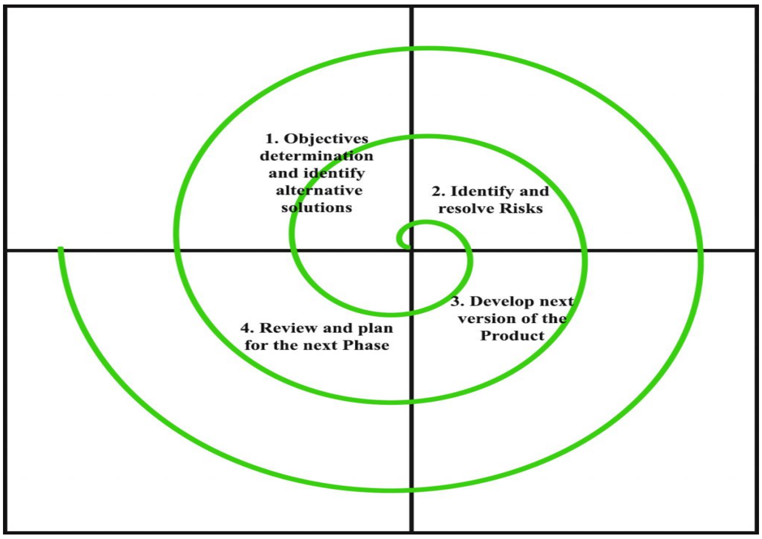

Figure 4.1: Spiral Model

CHAPTER 5
METHODOLOGY

5.1 System Requirements & Specifications

5.1.1 Tensorflow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

5.1.2 Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, equations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.

5.1.3 Python 3.6.2

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

Python runs on Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines. Python is distributed under an OSI-approved open source license that makes it free to use, even for commercial products.

5.1.4 Google Colab

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

5.1.5 Windows 10

Windows 10 is a series of operating systems developed by Microsoft and released as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, released nearly two years earlier, and was released to manufacturing on July 15, 2015, and broadly released for the general public on July 29, 2015. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10, which are available to Windows Insiders.

5.2 Proposed System

Modules

5.2.1 Data Acquisition Module

In this phase the image is captured using dataset. Images are saved in a folder and divided the images into 2 sections like testing and training. In our project data set we build a new dataset of 600 images of 80% training section and 20% testing section.

5.2.2 Data Preprocessing Module

. The step of data preprocessing plays a very important role in contributing to the accuracy of any training model. Pre-processing is a common name for operations with images at the lowest level of abstraction. The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, translation). Here, the image is resized to fit into the algorithm. In this project, we mainly perform,

1. **Annotation:**

   Image annotation is the human-powered task of annotating an image with labels. These labels are predetermined and are chosen to give the computer vision model information about what is shown in the image. The most commonly used and simplest type of image annotation is the bounding box. This form of annotation requires to draw a box as close as possible to the edges of key objects within the image. Image annotation is one of the most crucial tasks in computer vision.

5.2.3 Identification and Classification Module

This module identifies and classifies the data given. The main aim of the proposed work is to enrich the true meaning of gesture by Faster R-CNN with Inception V2 architecture. Inception V2 acts as back bone of this network. Here the captured image can determine the true meaning of that particular image and display it. Also, it can classify the different type of same label images using Faster R-CNN method. In the first stage, the images are undergone through preprocessing stage. Our application takes hand inputs of the dance form and it is pre processed. The second step is extracting the features from the pre-processed images. It is extracted using the convolutional neural network loaded with the Inception V2 filter banks..

Faster R-CNN is then used for hand gesture recognition and it will learn from the features that have been extracted from the input images. Faster R-CNN is known to have less prediction timing and shows greater accuracy. The comparison among many classification methods shows that Faster R-CNN shows best performance.

NEURAL NETWORK IDENTIFIED

Neural network identified for classification is Faster R-CNN

ALGORITHM USED: Faster R-CNN

Faster R-CNN is a third iteration of the R-CNN "Rich feature hierarchies for accurate object detection and semantic segmentation". R stands for regions and CNN stands for convolutional neural networks. The main objective of faster R-CNN is to reduce the computational expenses and to work at able to work at real time.

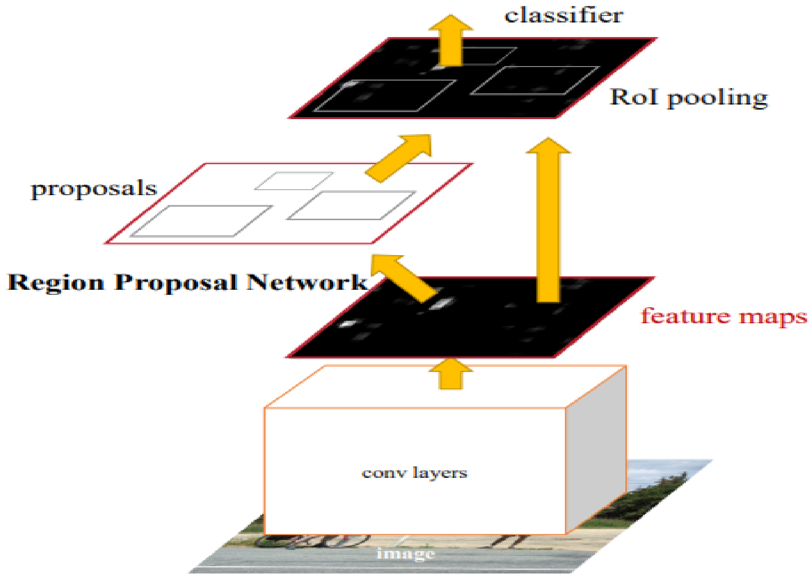

Figure 5.1: Faster R CNN

The architecture is comprised of two modules:

1. **Module 1:** Region Proposal Network.

   A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score. It is a fully convolutional network that generates proposals with various scales and aspect ratios. The RPN implements the terminology of neural network with attention to tell the object detection (Fast R-CNN) where to look. The RPN module is responsible for generating region proposals. It applies the concept of attention in neural networks, so it guides the Fast R-CNN detection module to where to look for objects in the image. The region proposals are now generated using a network that could be trained and customized according to the detection task. Because the proposals are generated using a network, this can be trained end-to-end to be customized on the detection task. Thus, it produces better region proposals compared to generic methods like Selective Search and EdgeBoxes. Due to sharing the same convolutional layers, the RPN and the Fast R-CNN can be merged/unified into a single network. Thus, training is done only once.

2. **Module 2:** Fast R-CNN.

   Convolutional neural network for extracting features from the proposed regions and outputting the bounding box and class labels

   Faster R-CNN is the modified version of Fast R-CNN. The major difference between them is that Fast R-CNN uses selective search for generating Regions of Interest, while Faster R-CNN uses "Region Proposal Network" RPN. RPN takes image feature maps as an input and generates a set of object proposals, each with an objectness score as output.

   The below steps are typically followed in a Faster R-CNN approach:

   - We take an image as input and pass it to the ConvNet which returns the feature map for that image.
   - Region proposal network is applied on these feature maps. This returns the object proposals along with their objectness score.
   - A RoI pooling layer is applied on these proposals to bring down all the proposals to the same size.
   - Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

Inception V2

The Inception network was an important milestone in the development of CNN classifiers. It is the second generation of Inception convolutional neural network architectures which notably uses batch normalization. To train the CNN for gesture recognition, the features are extracted from the pre-processed images. The features of the proposed region in an image is extracted using the convolutional neural network loaded with the Inception V2 filter banks. The same process is applied for all the images in the database to generate a training pattern. The performance of the neural network is better when the dimensions of the input are not altered drastically by convolutions. The larger convolutions are computationally expensive. Too much reduction in the dimensions of input results in loss of information, known as a "representational bottleneck". The inception V2 model is designed to reduce the dimensionality of its feature map, passing the resulting feature map through a Relu activation function, and then performing the larger convolution. [3]

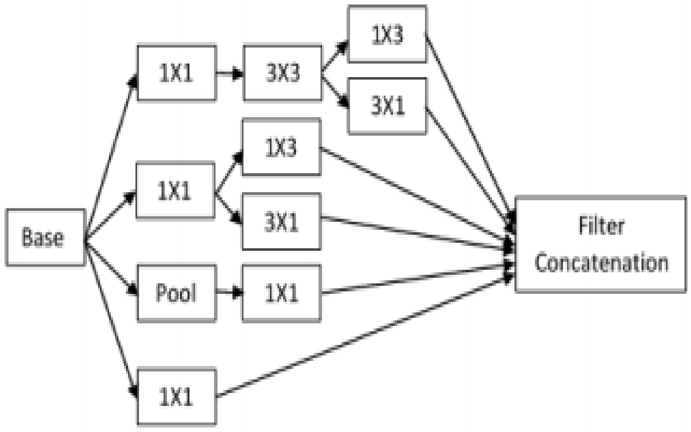

Figure 5.2: Inception V2 module with wider filter banks

Other changes include dropping dropout and removing local response normalization, due to the benefits of batch normalization. Prior to its inception, most popular CNNs just stacked convolution layers deeper and deeper, hoping to get better performance. Inception uses a lot of tricks to push performance; both in terms of speed and accuracy. Its constant evolution lead to the creation of several versions of the network. Here, Inception V2 is an upgraded version of Inception v1, which increased the accuracy and reduced the computational complexity. It Factorize 5x5 convolution to two 3x3 convolution operations to improve computational speed. Moreover, it also factorize convolutions of filter size nxn to a combination of 1xn and nx1 convolutions, thus making it cheaper.

5.3 Data Flow Diagrams

5.3.1 Data Flow Diagram- Level 0

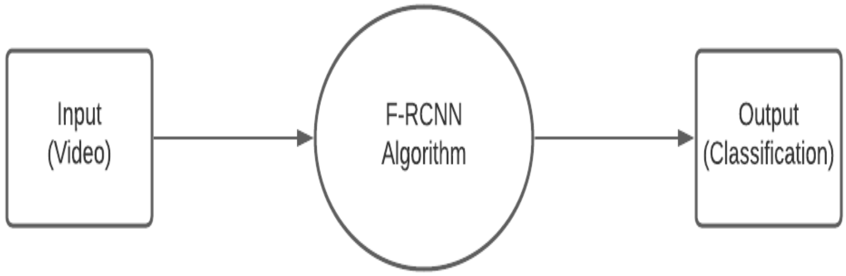

Figure 5.3: DFD- Level 0

5.3.2 Data Flow Diagram- Level 1

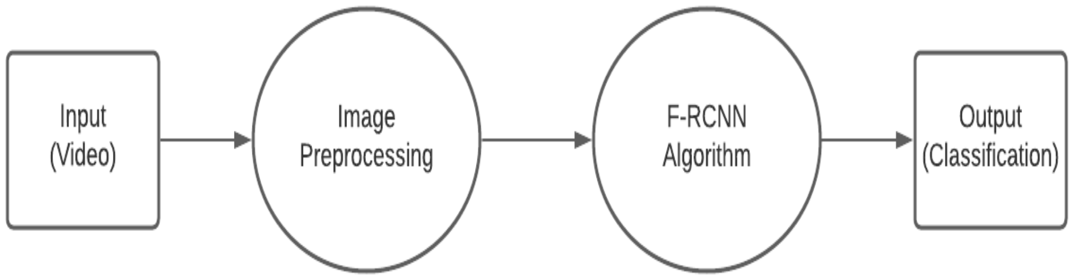

Figure 5.4: DFD- Level 1

5.3.3 Data Flow Diagram- Level 2

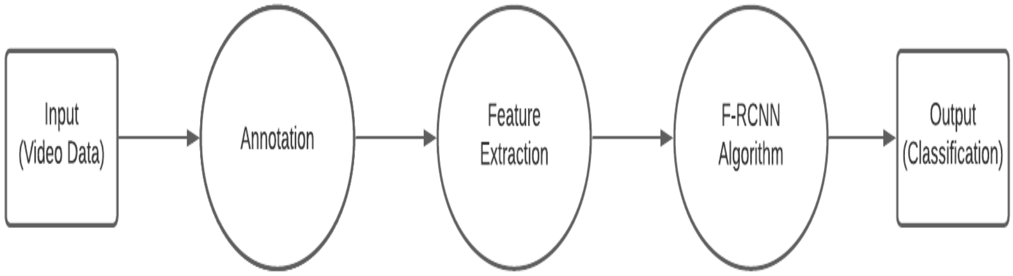

Figure 5.5: DFD- Level 2

5.4 Use Case Diagram

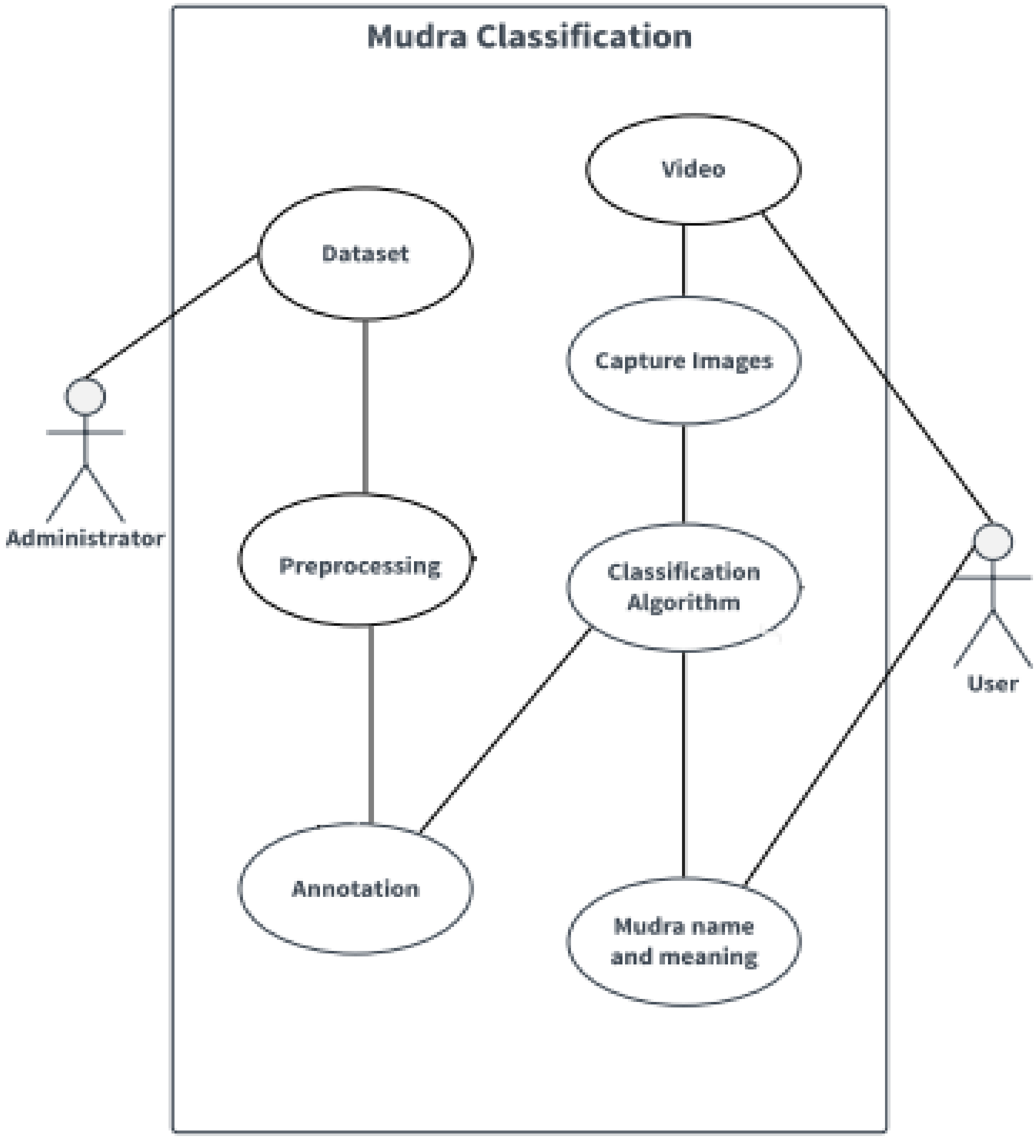

Figure 5.6: Use Case

5.5 Architecture

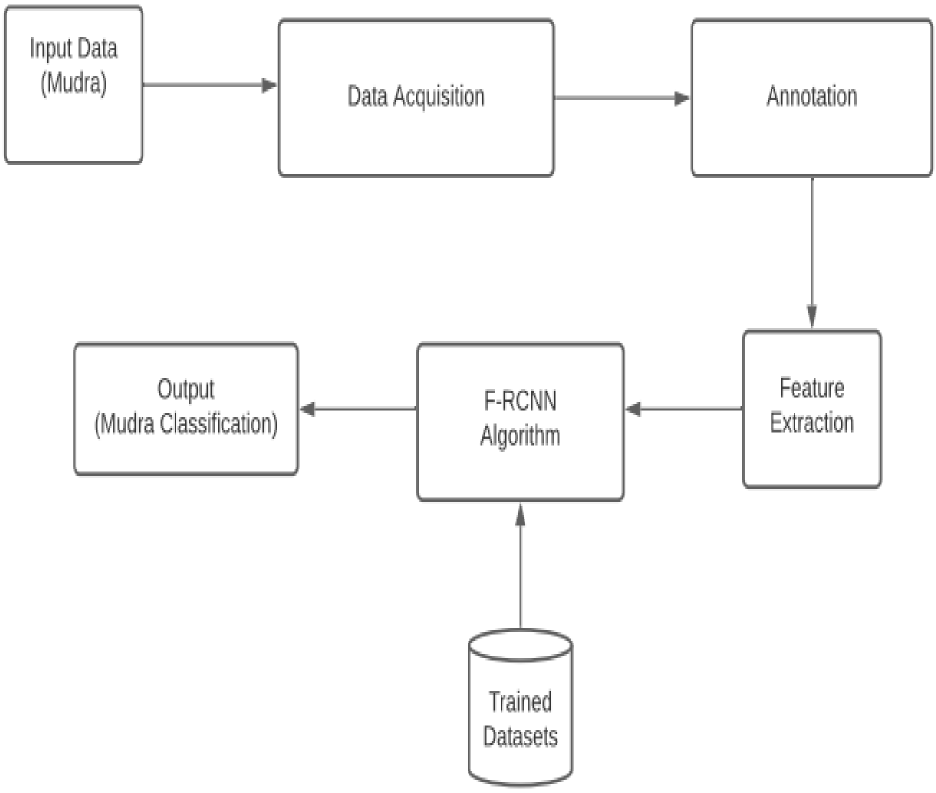

Figure 5.7: Architectural Diagram

CHAPTER 6
CONCLUSION AND FUTURE WORKS

Machine learning is widely used in solving many computing problems. The machine learning algorithms use data as input to predict the right output values. The project suggests a machine learning solution for the classification of hasta mudras in classical dance forms. India has a numerous amount of classical dance forms as part of its rich culture. But the truth is that a common person who wish to enjoy the art forms feels difficult to understand the theme portrayed in an art form. Through the implementation of project, it is expected that one can understand the dance forms better and appreciate the art form as well as the artist.

We have outlined the design of the proposed project, which aims to identify algorithms and features that can best identify and classify the different hasta mudras of classical dances. It uses the data for classification. The features are extracted using Inception V2 architecture and classified using Faster R-CNN. In the near future, the work can be extended to narrate the exact theme of the dance.

REFERENCES

[1] Basavaraj S Anami and Venkatesh A Bhandage. A comparative study of suitability of certain features in classification of bharatanatyam mudra images using artificial neural network. *Neural Processing Letters*, 50(1):741–769, 2019.

[2] Lakshmi Tulasi Bhavanam and Ganesh Neelakanta Iyer. On the classification of kathakali hand gestures using support vector machines and convolutional neural networks. In *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, pages 1–6. IEEE, 2020.

[3] MIT Campus. Hand gesture recognition using faster r-cnn inception v2 model. 2019.

[4] Mampi Devi and Sarat Saharia. A two-level classification scheme for single-hand gestures of sattriya dance. In *2016 International Conference on Accessibility to Digital World (ICADW)*, pages 193–196. IEEE, 2016.

[5] Andrei Dobre John McGonagle, Geoff Pilling and 5 others contributed. Gaussian mixture model: https://brilliant.org/wiki/gaussian-mixture-model/: :text=gaussian

[6] PVV Kishore, KVV Kumar, E Kiran Kumar, ASCS Sastry, M Teja Kiran, D Anil Kumar, and MVD Prasad. Indian classical dance action identification and classification with convolutional neural networks. *Advances in Multimedia*, 2018, 2018.

[7] KVV Kumar and PVV Kishore. Indian classical dance mudra classification using hog features and svm classifier. *International Journal of Electrical & Computer Engineering (2088-8708)*, 7(5), 2017.

[8] Sriparna Saha, Amit Konar, Deblina Gupta, Anasuya Ray, Abhinaba Sarkar, Pritha Chatterjee, and Ramadoss Janarthanan. Bharatanatyam hand gesture recognition using polygon representation. In *Proceedings of The 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, pages 563–567. IEEE, 2014.

[9] Sriparna Saha, Amit Konar, and Jayashree Roy. Single person hand gesture recognition using support vector machine. In *Computational advancement in communication circuits and systems*, pages 161–167. Springer, 2015.

[10] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[11] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011.

[12] CV Soumya and Muzameel Ahmed. Artificial neural network based identification and classification of images of bharatanatya gestures. In *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 162–166. IEEE, 2017.

[13] Gonzalo Vaca-Castano. Fisher vector encoding: http://www.cs.ucf.edu/courses/cap6412/spr2014/papers/fisher-vector-encoding.pdf.

[14] Wikipedia. Support vector machine: https://en.wikipedia.org/wiki/support-vector-machine.