Recognition of Sattriya Dance Double-Handed Mudra from Image and Video

REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

By

Dishanka Kalita

Roll Number: 200102021

UNDER THE GUIDANCE

OF

DR PARISMITA SARMA

Assistant Professor, Dept. of Information Technology,

Gauhati University

DEPARTMENT OF INFORMATION TECHNOLOGY
GAUHATI UNIVERSITY
GUWAHATI, INDIA

JUNE –2024

GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

DECLARATION

I Dishanka Kalita, Roll No 200102021, B.Tech. student of the department of Information Technology, Gauhati University hereby declare that I have compiled this report reflecting all my works during the semester long full time project as part of my BTech curriculum.

I declare that I have included the descriptions etc. of my project work, and nothing has been copied/replicated from other's work. The facts, figures, analysis, results, claims etc. depicted in my thesis are all related to my full time project work.

I also declare that the same report or any substantial portion of this report has not been submitted anywhere else as part of any requirements for any degree/diploma etc.

Dishanka Kalita
Branch: Information Technology
Date:20/06/2024

GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date:

CERTIFICATE

This is to certify that Dishanka Kalita bearing Roll No: 200102021 has carried out the project work "Recognition of Sattriya Dance Double-Handed Mudra from Image and Video".under my supervision and has compiled this report reflecting the candidate's work in the semester long project. The candidate did this project full time during the whole semester under my supervision, and the analysis, results, claims etc. are all related to his studies and work during the semester.

I recommend submission of this project report as a part for fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology of Gauhati University.

Dr Parismita Sarma
Assistant Professor, Dept. of IT, GU

GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

External Examiners Certificate

This is to certify that Dishanka Kalita, bearing Roll No 200102021, delivered his project presentation on 20/06/2024 and I examined his report entitled "Recognition of Sattriya Dance Double-Handed Mudra from Image and Video" and recommend this project report as a part for partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology of Gauhati University.

_____

(External Examiner)

GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date:

TO WHOM IT MAY CONCERN

This is to certify that Dishanka Kalita, bearing Roll No. 200102021, B.Tech. student of the department of Information Technology at Gauhati University, has submitted the softcopy of his project for undergoing screening through anti-plagiarism software and the similar report found to be %.

Dr Parismita Sarma
Assistant Professor, Dept. of IT, GU

ACKNOWLEDGEMENT

We wish to place on record a deep sense of respect and profound gratitude to our advisor and Project Guide, Dr. **Parismita Sarma**, for her constant motivation and valuable help,which resulted in the successful completion of this project.

We would also like to express our honest salutation and gratitude to **Prof. Shikhar Kumar Sarma, Head of the Department, Department of Information Technology, Gauhati University**, for the generous help and cooperation in providing the required facilities while carrying out the project. His guidance and motivation have always been a great source of inspiration for us.

We would like to extend our gratitude to the other faculty members of the Department for their cooperation and generous help for the completion of the project. We would also like to thank our friends for helping us in completing our project.

Dishanka Kalita (200102021)

Information Technology (IT)

ABSTRACT

This study presents a novel automated method for recognizing and analyzing intricate hand gestures in Sattriya dance, utilizing image processing and machine learning techniques. Our primary objective is to develop a robust image processing system capable of distinguishing and classifying Sattriya dance double hand mudras within complex visual environments. This system automates mudra recognition by employing cutting-edge computer vision algorithms and AI based Deep learning models. The dataset of double handed Sattriya dance mudras were collected and compiled by us. We have used here YOLOv8, YOLOv5, and RT-DETR, three prominent Deep learning based models to detect the images in a faster and accurate way. The YOLOV8 model out performs among the three models with highest 0.83758 mean Average Precision value. We have further enhanced its accessibility by developing a user-friendly interface built using HTML, CSS, and JavaScript, which integrates seamlessly with a Flask backend. This interface allows users to interact with the system for mudra recognition , making it a valuable tool for researchers and practitioners in the performing arts.

Table of Contents

List of Tables and Figures

List Of Figures:

List Of Tables:

1. Introduction:

The blending of technology and tradition has produced fascinating results in the fields of artistic discovery and cultural heritage preservation. Hand mudras, a complex language of symbolic gestures, have great significance in many ancient dance genres. Mudras are symbolic hand gestures and movements used to convey emotions, stories, and concepts behind any Indian classical dance form. Traditional cultural practices are richly preserved in Indian classical dances and it conveyed through various types of mudras associated with Indian classical dance [4]. One such type of dance is Sattriya, a classical dance that has its roots in the Indian state of Assam in the northeast. Sattriya dance conveys spiritual and cultural stories through the expressive use of mudras, and it is more than just a rhythmic form of expression. This project explores the fascinating world of Sattriya hand mudras, setting off on a journey at the nexus of tradition and invention. The goal is to create a reliable system that can recognize, interpret, and show the messages conveyed by the various hand gestures used in Sattriya dance performances. This initiative aims to provide new opportunities for artistic inquiry, advance the preservation of Sattriya dance, and expand our understanding of the art form by bridging the gap between traditional artistry and contemporary technology. The main goal of the project is to develop an automated framework that can identify and separate the complex hand mudras used in Sattriya dance performances. The system attempts to interpret these gestures' complex grammar by combining machine learning techniques and image processing technologies. The ensuing understandings allow dancers, scholars, and enthusiasts to explore the stories woven into each mudra in addition to improving our understanding of Sattriya dance as an art form. This introduction lays the groundwork for an engrossing investigation into the fusion of technical progress and cultural tradition. We shall explore the methods, difficulties, and results of this project in greater detail as we go through the ensuing chapters, concluding with a celebration of the harmonic fusion of custom and state-of-the-art image processing methods in the field of Satriya hand mudras.

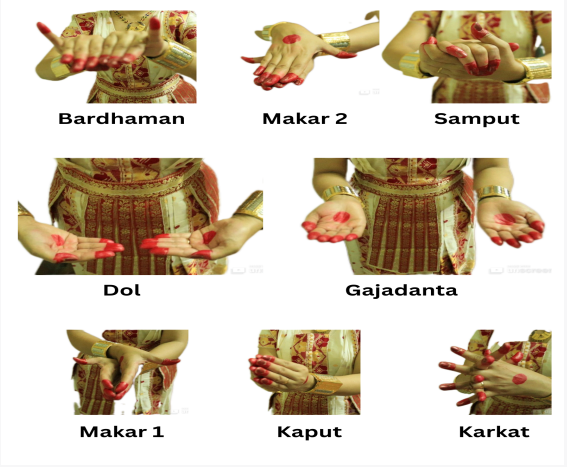


Figure 1: Sattriya Dance Double Handed Mudras

This project is about creating a system that can identify the hand gestures (mudras) used in Sattriya dance, using both pictures and videos. We're all working together to achieve these key objectives:

- Building a Rich Dataset: We'll be collecting a variety of Sattriya dance performances, capturing a wide range of mudras. This means finding videos and images that show different contexts, lighting conditions, and dance styles. The more diverse the data, the better the system will understand real-world scenarios.

- Data Preparation: Making the Data Usable: The collected data might need some cleaning and adjustments before it can be used for training. We might be involved in tasks like reducing noise in the images, ensuring consistent formats, or resizing them to a standard size.

- Training the Machine Learning Model: The Mudra Recognizer: We'll be using a special kind of computer program called a machine learning model to learn how to recognize mudras. We'll be involved in training this model by feeding it the prepared data. The model will then be able to analyze new images and videos and identify the mudras being used.

- Testing the System: How Well Does It Work?: Once the model is trained, we need to see how well it performs. We'll be testing the system using a separate set of Sattriya dance videos that the model hasn't seen before. We'll use different metrics

like accuracy, precision, and recall to measure how effective the system is in recognizing mudras.

- Sharing Our Knowledge: Documenting the Process: It's important to document everything we do so others can learn from our work. We might be involved in creating detailed instructions and explanations of the methods we used, the algorithms involved, and the results we achieved. This will help others build upon our work in the future.

- Promoting Sattriya Dance: Ultimately, this project aims to create a tool that can help people better understand and appreciate Sattriya dance. By making mudra recognition easier, we can contribute to the preservation and promotion of this beautiful art form.

This project is an exciting opportunity to combine technology with cultural heritage. By working together towards these objectives, we hope to make a positive impact on the world of Sattriya dance.

2. Literature Review

As per our knowledge, very little research has been carried out on recognizing sattriya dance hand mudra detection. The previous works have mostly focused on single handed mudras.

Asamyukta hastas in Sattriya dance, which is a classical dance form, studied by Mampi Devi et al. This paper proposes a two-level method for recognizing Sattriya dance hand gestures (asamyukta hastas). The first level achieves high accuracy (98%) grouping gestures, but the second level for individual hasta recognition is lower (average 75.45%). The authors attribute this to the similarity between the hastas and propose further research to identify more distinctive features [6].

R. Pradeep, R. Rajeshwari, V. R. Ruchita, R. Bubna and H. R. Mamatha, paper on Recognition of Indian Classical Dance Hand Gestures This paper explores the potential of Convolutional Neural Networks (CNNs) for recognizing hand gestures (mudras) in Indian classical dance. It investigates the effectiveness of various pre-trained CNN models on image datasets and analyzes their performance. Additionally, the paper delves into the under-researched area of recognizing mudras from dance videos using object detection techniques, potentially combined with CNN results for a more robust approach. This research paves the way for further exploration in automatic mudra recognition with applications in e-learning, documentation, and analysis of Indian classical dance forms. [1]

SANTHOSH, REMYA and KK, Rajkumar paper on Classification of Asmyukta Mudras in Indian Classical Dance Using Handcrafted and Pre-Trained Features with Machine Learning and Deep Learning Methods. This research suggests that carefully chosen deep learning architectures combined with well-designed manual features can achieve superior performance in complex classification tasks like mudra recognition. The expanded VGG19 architecture with its additional layers proved to be more effective for capturing intricate details in the mudra images. [4]

While there has been some research on hand gesture recognition in Indian classical dance, there is a significant gap in the literature regarding specifically recognizing double-handed mudras (Samyuta hastas) in Sattriya dance.

H. Bhuyan, P. P. Das, J. K. Dash and J. Killi, "An Automated Method for Identification of Key frames in Bharatanatyam Dance Videos," The project on the automated identification of key frames in Bharatanatyam dance videos makes significant contributions to the field of dance video analysis. By introducing a novel approach that combines three-frame difference and bit-plane techniques for segmentation, the project successfully partitions dance videos into key frames and motion frames. This segmentation is crucial for understanding and analyzing dance performances effectively. Additionally, the project devises an adaptive threshold for non-machine learning approaches, enhancing the accuracy of key frame detection. Furthermore, by exploring machine learning techniques such as Support Vector Machine (SVM) and Convolutional Neural Network (CNN) classifiers, the project demonstrates improved key frame identification accuracy. Through comparisons with existing methods, the project showcases high accuracy rates, around 90%, establishing its effectiveness in automated key frame identification in Bharatanatyam dance videos [7].

Joko S. Dance gesture recognition using space component and effort component of laban movement analysis, This paper on Dance Gesture Recognition using the Space and Effort components of Laban Movement Analysis makes a significant contribution to the field of dance research and technology. By utilizing motion capture technology like the Kinect sensor and applying the principles of Laban Movement Analysis, the study aims to recognize and classify dance movements with high accuracy. The use of Hidden Markov Model (HMM) for quantization, normalization, and classification further enhances the precision of recognizing dance patterns. The research results in an impressive accuracy rate of close to 99% for dance motion data, showcasing the potential of integrating technology and movement analysis in understanding and appreciating the art of dance [8].

Kico I, Grammalidis N, Christidis Y, Liarokapis F. Digitization and visualization of folk dances in cultural heritage, This paper provides a comprehensive overview of the digitization and visualization of folk dances in cultural heritage from 2000 to 2018. It discusses the importance of intangible cultural heritage, particularly in the form of dance, and the role it plays in maintaining cultural diversity. The paper reviews techniques for recording dance moves, visualization methods, feedback mechanisms for users, and performance evaluation. Additionally, it emphasizes the use of advanced technologies such as machine learning and deep learning algorithms to automate tasks like feature extraction and pattern recognition, paving the way for a more sophisticated system accessible to a wider audience. The research presented in this paper contributes to the ongoing efforts to preserve and promote folk dances through innovative digital means [9].

Bhushan S, Alshehri M, Keshta I, Chakraverti AK, Rajpurohit J, Abugabah A. An experimental analysis of various machine learning algorithms for hand gesture recognition, The paper contributes significantly to the field of hand gesture recognition by conducting an experimental analysis of various machine learning algorithms. It explores the effectiveness of algorithms such as Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), logistic regression, Naïve Bayes (Multinomial NB and Gaussian NB), Stochastic Gradient Descent Classifier (SGDC), and Convolutional Neural Networks (CNN) on the sign language MNIST dataset.

The study provides valuable insights into the performance of these algorithms, with the proposed CNN model achieving a high accuracy of 91.41%. By comparing the proposed model with other state-of-the-art techniques, the research demonstrates the superiority of the CNN model in hand gesture recognition tasks.

Overall, the paper's contribution lies in enhancing our understanding of the capabilities of different machine learning algorithms for hand gesture recognition, paving the way for the development of more accurate and efficient gesture recognition systems in various real-world applications [10].

Naik AD, Supriya M. Classification of Indian classical dance images using convolution neural network, This paper contributes to the field of computer vision and deep learning by proposing a methodology for the identification and classification of Indian Classical Dance images using Convolutional Neural Networks (CNN). The system is designed to recognize and categorize five different dance forms - Bharatanatyam, Odissi, Kathak, Kathakali, and Yakshagana. By collecting images from the internet and utilizing CNN, the system can automate the process of distinguishing between these dance forms based on their unique gestures and styles. This work not only showcases the application of CNN in classifying cultural art forms but also provides a platform for users to engage in automated dance quizzes to test their familiarity with the diverse dance forms in India[11].

V . Janaranjani1, N. Megala2, R.Naveeth Kumar3 1,2,3Assistant Professor in Biomedical Engineering, Rathinam Technical Campus, Coimbatore Hand Gesture Recognition for Dance Movements, This paper on Hand Gesture Recognition for Dance Movements makes a significant contribution by introducing a system that utilizes flex sensors and microcontrollers to recognize hand gestures for teaching basic mudras in classical Indian dance forms. The system allows users to interact with a computer through hand gestures, enabling a unique and innovative way to learn dance movements. Additionally, the proposed system can be extended to function as a Dance Learning App on Android mobile devices, further enhancing accessibility and usability. Overall, this paper's contribution lies in merging technology with traditional dance education, offering a novel approach to learning and practicing dance gestures[12].

Existing work on Sattriya dance has focused on single-handed mudras and achieved high accuracy in classifying groups of gestures but lower accuracy in individual mudra recognition [6].

Research on broader Indian classical dance explores CNNs for mudra recognition and investigates object detection for video analysis, paving the way for future exploration in automatic mudra recognition [1].

Studies on hand gesture recognition in general highlight the potential of combining deep learning architectures with handcrafted features for complex classification tasks [4].

This limited research on double-handed mudras in Sattriya dance suggests a significant opportunity for further investigation. Future work can explore applying techniques from both single-handed mudra recognition and general hand gesture classification to develop robust methods for accurate detection and classification of double-handed mudras specific to Sattriya dance.

3. System Architecture

Object detection stands as a crucial element within the field of computer vision. It plays a pivotal role in enabling interactions between images and text, as well as facilitating the tracking of distinct entities [2]. In our project which is to detect the different double handed gestures of sattriya dance from images and videos. The algorithms which we mainly considered are YOLOv8, RT-DETR and YOLOv5

The methods such as YOLO(You Only Look Once), RT-DETR have affluently obtained an efficient, fast and accurate model with high mean average precision (mAP); But, their frames per second (FPS) on non-GPU computers drive them unsuitable for real time application [5]. YOLO version 8 is faster Compared to R-CNN, SSD, Faster R-CNN and earlier versions of YOLO. RT-DETR Real-Time Detection Transformer (RT-DETR) is a state-of-the-art object detection algorithm that leverages the power of transformers in the field of Object Detection.

YOLO

YOLO (You Only Look Once), a popular object detection and image segmentation model, was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in 2015, YOLO quickly gained popularity for its high speed and accuracy.

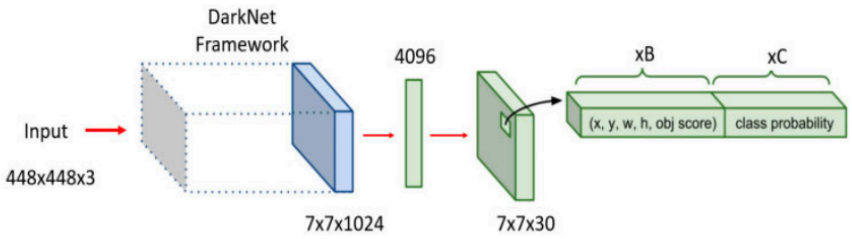


Figure 2: Architecture of YOLO

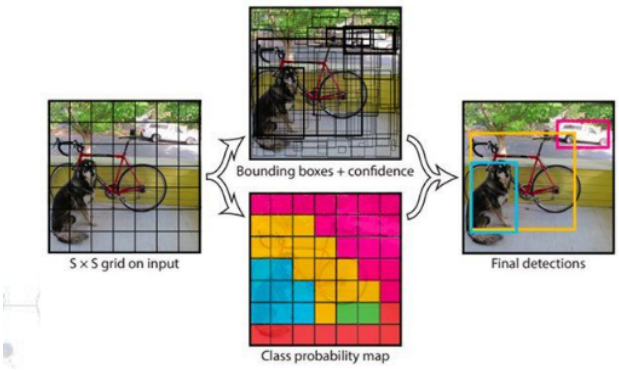


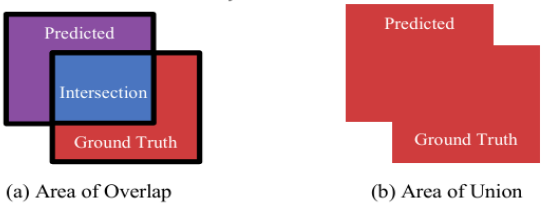
Figure 3: Stages of Object Detection by YOLO

Why YOLO is differ from other existing networks? The answer is: at the same time bounding box predictions and class predictions can be done by YOLO. First, the input image is sliced up into S × S grids [7]. Second, B bounding boxes are included in each grid cell, each with a score of confidence. The probability of an object exists in each bounding box is known as confidence and is defined as [5]:

$$C = Pr(Object) * IOU_{pred}^{truth}$$

Where, IOU is the abbreviation of intersection over union which describes a measure (a fraction between 0 and 1) of overlap between two bounding box. Intersection is the overlapped area between the predicted bounding box and ground truth (Object), and union is the total area of predicted and ground truth boxes as drawn in Fig. 2. For accurate detection, the IOU should be near to 1, so that the predicted bounding box lies closer to the ground truth. IOU can be expressed as[5]:

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$



(a) Area of Overlap (b) Area of Union

Figure 4: Intersection over union

At the same time, while generating bounding boxes, each grid cell predicts C conditional class probability of the object. The class-specific probability for each grid cell is:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$$

YOLO utilizes the equation below for calculating loss function and finally improve confidence [5]:

Loss function:

$$\begin{aligned} &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ &+\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ &+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 \\ &+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 \\ &+ \lambda_{noobj} \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2 . \end{aligned} \quad (4)$$

The accuracy of predictions generated by models in object detection is calculated through the average precision equation defined as [5]:

$$avgPrecision = \sum_{k=1}^{n} P(k) \Delta r(k)$$

P(k) refers to the precision at threshold k and Δr(k) refers to the fluctuate in recall [5]

The different versions of YOLO:

- The 2016 edition of YOLOv2 enhanced the original model by adding dimension clusters, anchor boxes, and batch normalisation.
- Launched in 2018, YOLOv3 greatly improved the model's performance by utilising spatial pyramid pooling, numerous anchors, and an even more efficient backbone network.

- YOLOv4 was released in 2020, introducing innovations like Mosaic data augmentation, a new anchor-free detection head, and a new loss function.
- YOLOv5 further improved the model's performance and added new features such as hyperparameter optimization, integrated experiment tracking and automatic export to popular export formats.
- Meituan made YOLOv6 open-source in 2022, and many of the company's autonomous delivery robots utilise it.
- Additional tasks, such posture estimation on the COCO keypoints dataset, were added by YOLOv7.
- YOLOv8 is the latest version of YOLO by Ultralytics. YOLOv8 enhances efficiency, performance, and flexibility with new features for visual AI tasks like tracking, segmentation, posture estimation, detection, and classification, allowing users to use it in various contexts.

RT-DETR

Baidu has created a state-of-the-art end-to-end object detector called Real-Time Detection Transformer (RT-DETR) that offers excellent accuracy and real-time performance. Its foundation is the DETR (the NMS-free framework) concept, to which an effective hybrid encoder and a convolution-based backbone are added in order to increase real-time speed. By separating cross-scale fusion from intra-scale interaction, RT-DETR processes multiscale characteristics effectively. Because it can easily alter the inference speed using different decoder layers without requiring retraining, the model is quite versatile. RT-DETR outperforms several other real-time object detectors and performs exceptionally well on accelerated backends such as CUDA with TensorRT.

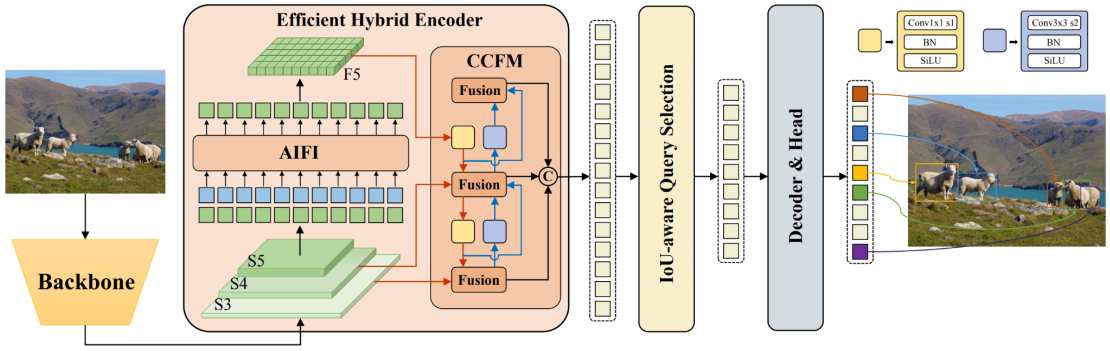


Figure 5. Overview of RT-DETR

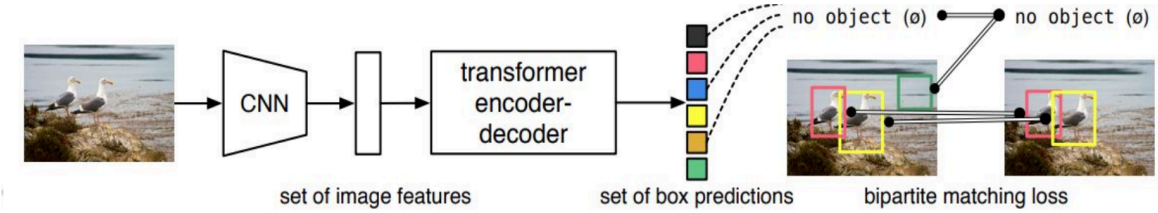


Figure 6. Architecture of RT-DETR

Key Features

Efficient Hybrid Encoder: Baidu's RT-DETR uses an efficient hybrid encoder that processes multiscale features by decoupling intra-scale interaction and cross-scale fusion. This unique Vision Transformers-based design reduces computational costs and allows for real-time object detection.

IoU-aware Query Selection: Baidu's RT-DETR improves object query initialization by utilizing IoU-aware query selection. This allows the model to focus on the most relevant objects in the scene, enhancing the detection accuracy.

Adaptable Inference Speed: Baidu's RT-DETR supports flexible adjustments of inference speed by using different decoder layers without the need for retraining. This adaptability facilitates practical application in various real-time object detection scenarios.

Workflow of the Proposed Work

- **Data Collection:**

  We compile a varied samples of Sattriya dancing performances, taking pictures of distinct hand mudras in a range of settings, including light and shadow.

- **Data Preprocessing:**

  To guarantee consistency in image quality and format, we apply preprocessing procedures to standardize the dataset, such as image scaling, normalization, and color correction.

- **Machine Learning Model Development:**

  Following data collection and annotation, we explored various machine learning based advanced deep learning models for mudra recognition. Three prominent object detection models YOLOv8, YOLOv5, and RT-DETR were trained on the prepared dataset. This dataset encompassed a diverse range of hand mudras and positions, ensuring comprehensive model training. A crucial aspect of this stage involved collaborating with Sattriya dance specialists during dataset annotation to maintain cultural sensitivity.

- **Model Training:**

  Following the development of the comprehensive Sattriya dance mudra dataset, we embarked on the model training phase. This critical stage involved training three prominent object detection models YOLOv8, YOLOv5, and RT-DETR on the prepared data. The dataset's richness, encompassing a wide variety of hand mudras and positions, ensured the models were exposed to a diverse range of scenarios.

- **Validation and Performance Evaluation:**

  Assess the trained model using performance metrics like train/cls_loss, Accuracy(mAp50 and mAp50-95), precision and recall on the testing dataset. To make sure the model is resilient across several data subsets, perform cross-validation.

- **Documentation and Knowledge Transfer:**

  We prioritized comprehensive documentation for the project's long-term viability, including detailed explanations of the YOLOv8 model architecture, training process, evaluation metrics, and deployment steps, enhancing

readability and maintainability. Fig 7 below shows the work flow diagram of our proposed model.

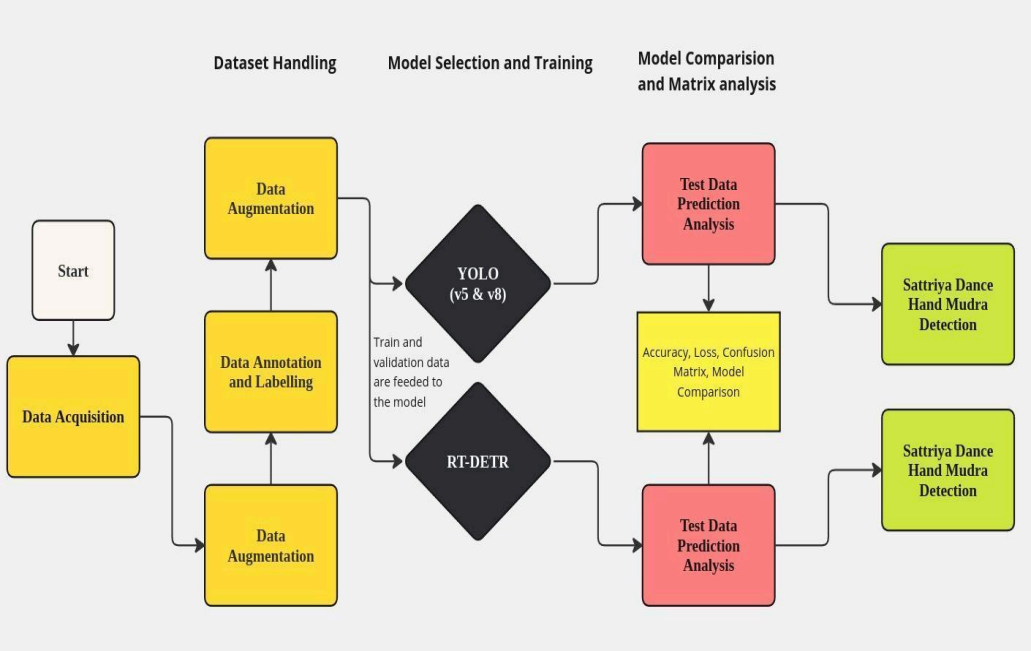


Figure 7. Workflow of the proposed model

4. Implementation

The following is an overview of the actual execution of the project on "Recognition of Sattriya Dance Double-Handed Mudra from Image and Video"

4.1 Setting Up the Environment:

To optimize resource utilization and streamline project development, we employed a hybrid approach for the development environment.

For model training, validation, and testing, we leveraged Google Colab, a cloud-based platform offering pre-configured virtual machines with GPUs. This provided access to substantial computational resources, ideal for training complex deep learning models like YOLOv8. Colab served as the primary environment for these computationally intensive tasks.

However, for development tasks requiring finer control or iterative adjustments, we established a local development environment on our machines. This environment mirrored the one used on Colab, with Python as the core language and essential libraries like NumPy, Pandas, scikit-learn, TensorFlow/PyTorch (specify which one you used), OpenCV, and web development tools like Flask (and HTML, CSS, and JavaScript if applicable). This local setup offered flexibility for code development, debugging, and version control, while Colab's computational power efficiently handled the training workload. This combined approach ensured efficient resource utilization and streamlined project development.

4.2 Dataset preparation

Access to an appropriate dataset is a critical factor in the development and evaluation of detection models [4]. A custom dataset of initially 2480 images were taken of different double handed mudras of Sattriya dance. Each mudras is taken from different angles, like front view, left view and right view.

Figure 8. Images from different angle of Bardhaman Mudra

There are a total of 8 mudras in the dataset. For efficient annotation, preprocessing, and augmentation, the project leverages Roboflow for uploading the Satriya dance dataset. Roboflow Annotation:Each mudra is separated into a different class during annotation and labeling. Different data preprocessing and augmentation techniques are mentioned below.

Data Preprocessing:

- Auto-Orient: Applied
- Resize: Stretch to 640x360

- Grayscale: Applied

Data Augmentation:

- Flip: Horizontal
- Rotation: Between -15° and +15°
- Shear: ±15° Horizontal, ±15° Vertical
- Brightness: Between -25% and +25%
- Blur: Up to 3.25px
- Noise: Up to 5% of pixels

The annotated dataset is prepared on Roboflow, where images are annotated, and the dataset is exported in YOLO and RT-DETR compatible formats. After the augmentation steps, the dataset has a total of 5232 images and the dataset is split into:

Train Set: 4578 Images

Valid Set: 436 Images

Test Set: 218 Images

Figure 9 shows different actions after annotation, labeling, preprocessing and augmentation the dataset.

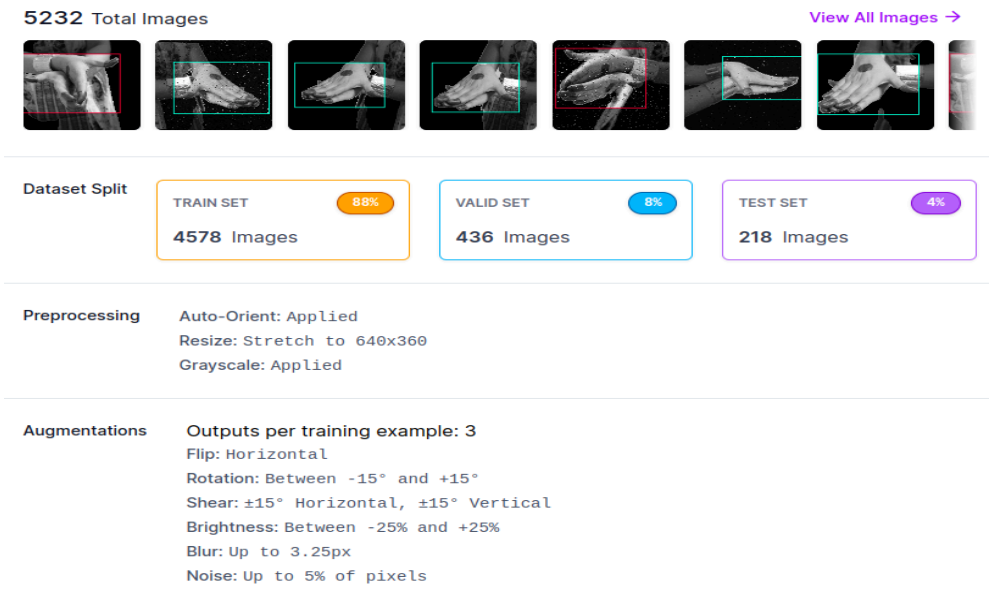


Figure 9. Sattriya Dance Double handed mudra dataset

4.3 YOLO Training

After the environment setup, we embarked on the training process for the YOLO models – YOLOv8, YOLOv5. Each model was trained on the prepared Sattriya dance mudra dataset, ensuring the models were exposed to a diverse range of hand mudras and position

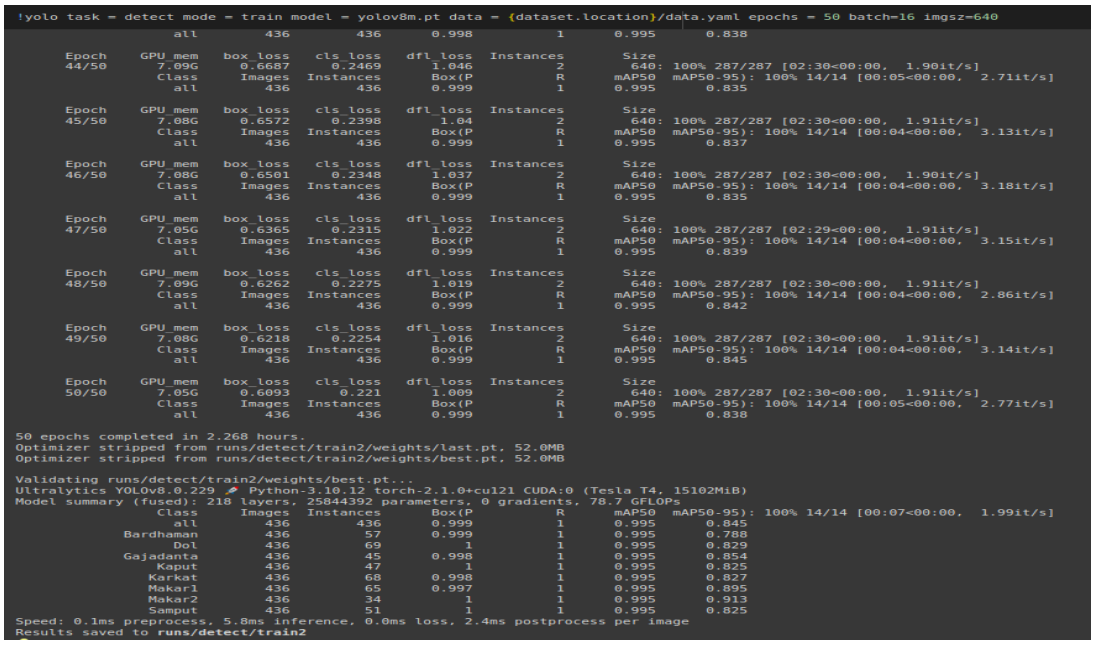


Figure 10: Screenshot for YOLOv8 Training

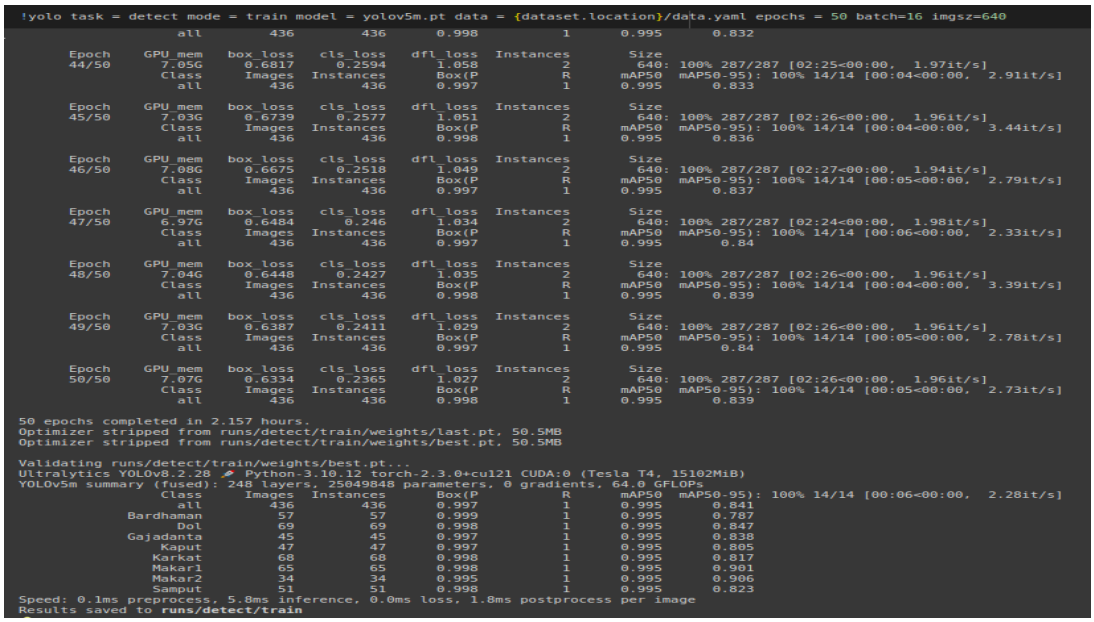


Figure 11: Screenshot for YOLOv5 Training

4.4 RT-DETR Training

We embarked on training the RT-DETR model for Satriya dance mudra recognition. The model was trained on a meticulously prepared dataset encompassing a diverse range of hand mudras and positions. Figure 12 shows the screenshot for training of our proposed model on RT-DETR architecture.

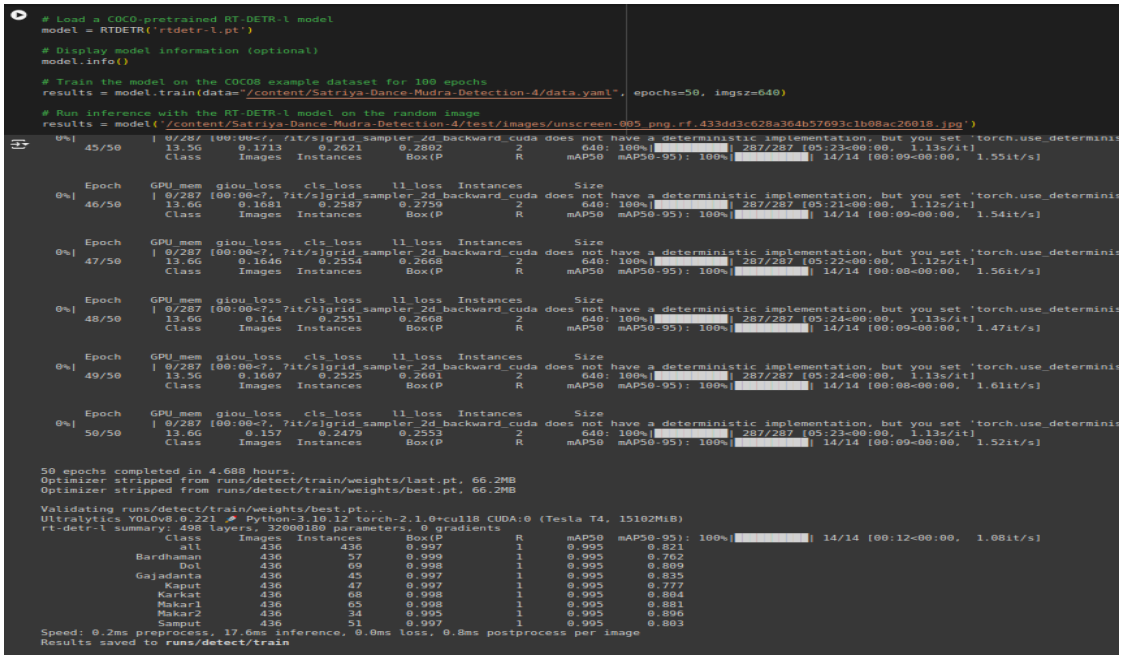


Figure 12. Screenshot for RT-DETR Training

4.5 YOLO and RT-DETR Validation

Following the training process for both YOLO (YOLOv8 and YOLOv5) and RT-DETR, a rigorous validation stage was conducted. This stage aimed to assess the effectiveness of each model in recognizing Sattriya dance mudras based on unseen data. We employed a separate validation dataset not used during training. This ensured an objective evaluation of the models' generalizability to real-world scenarios. During validation, we evaluated the models using a comprehensive set of metrics. These metrics included:

- mAP (mean Average Precision): This metric measures the average precision across all classes (mudras) in the dataset, providing an overall accuracy indicator.
- Precision: This metric reflects the proportion of correctly identified mudras among all detections made by the model.
- Recall: This metric indicates the proportion of actual mudras in the data that the model successfully detected.

By evaluating these metrics for each model, we gained valuable insights into their strengths and weaknesses regarding mudra recognition. This rigorous validation process, along with the analysis of training performance, ultimately played a crucial role in selecting the most suitable model for deployment in the web application (discussed in a later section).

4.6 YOLO and RT-DETR Testing

Following the comprehensive validation process for both YOLO models (YOLOv8 and YOLOv5) and RT-DETR, we conducted a dedicated testing phase specifically for the YOLO models. This additional testing aimed to gain deeper insights into their real-world mudra recognition capabilities beyond the controlled environment of validation.

We utilized a separate testing dataset entirely distinct from both the training and validation datasets. This ensured the models were evaluated on completely unseen data, further solidifying the generalizability of their performance.

4.7 Model Evaluation

Following the training process for YOLO models (YOLOv8 and YOLOv5) and RT-DETR, we conducted a rigorous evaluation to identify the most effective model for Satriya dance mudra recognition. This evaluation employed a multi-stage approach, involving both validation and testing on separate datasets.

Validation: A dedicated validation stage assessed the models' ability to generalize to unseen data. We employed a separate validation dataset not used during training. During validation, we evaluated each model using a comprehensive set of metrics, including mAP (mean Average Precision), precision, recall, and the analysis of training loss vs. validation loss. This process provided valuable insights into the models' strengths and weaknesses regarding mudra recognition.

Testing: To further assess generalizability and real-world performance, we conducted a dedicated testing phase for all three models (YOLOv8, YOLOv5, and RT-DETR). This testing utilized a completely unseen dataset, distinct from both training and validation data. During testing, we employed a comprehensive set of metrics like mAP, precision, recall, and potentially application-specific metrics like detection speed or frame rate.

By analyzing the combined results from validation and testing, we were able to comprehensively assess the performance of all models. This evaluation not only considered accuracy metrics like mAP and precision but also factors like computational efficiency, which can be crucial for web application deployment. This comprehensive evaluation process ultimately played a critical role in selecting the best model for deployment within the web application (discussed in a later section).

4.8 Making it Accessible: Web App Deployment

After evaluating the YOLO and RT-DETR models, we selected the one with the best performance. To ensure a user-friendly experience, we built a web application using Flask, a lightweight web framework written in Python. It has a minimalistic core, making it easy to learn and use, especially for developers with some Python experience. This allows you to focus on the core functionality of your web application, which is interacting with your machine learning model.. For the user interface, we utilized HTML, CSS, and JavaScript to create a visually appealing and interactive experience. This combination allows users to focus on getting the results they need without worrying about the underlying technical details.

5. Results and Discussion:

5.1. YOLOv8 Model Performance

Training Outcome:

Table 1: Training Outcome of YOLOv8

Loss	Accuracy	
cls_loss	mAp50	mAp50-95
0.22104	0.995	0.83758

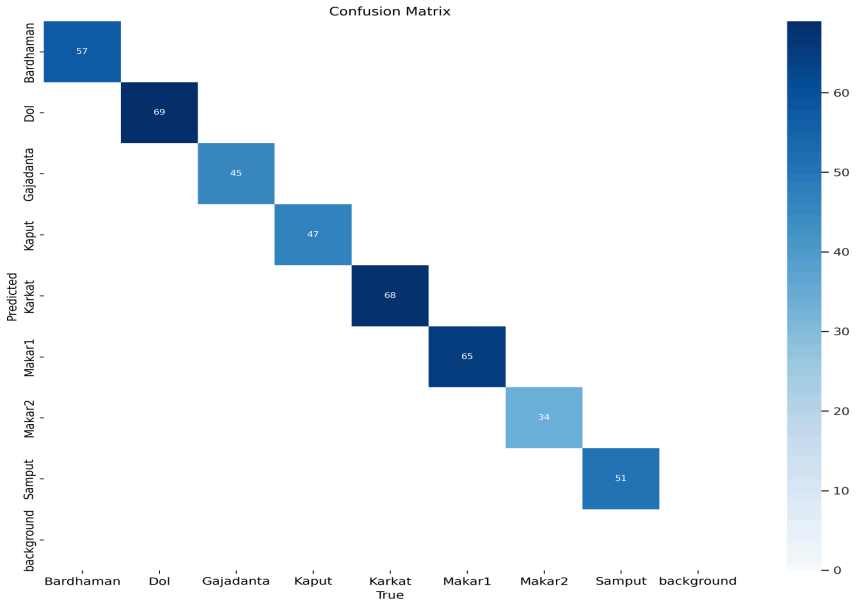


Figure13. YOLOv8 Confusion Matrix

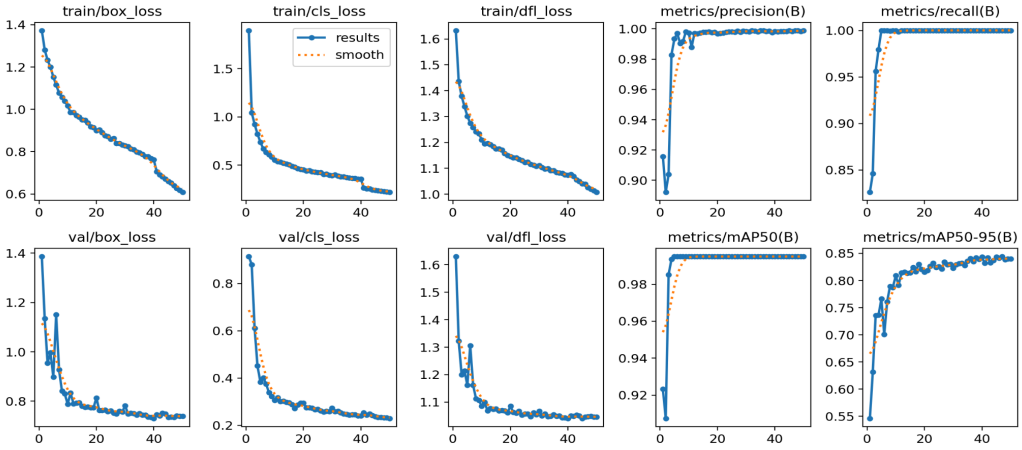


Figure14.YOLOv8 Train Result

Validation Outcomes:

Table 2: Validation Outcome of YOLOv8

Class	mAp50	mAp50-95
Bardhaman	0.995	0.781
Dol	0.995	0.829
Gajadanta	0.995	0.849
Kaput	0.995	0.828
Karkat	0.995	0.827
Makar1	0.995	0.891
Makar2	0.995	0.92
Samput	0.995	0.822

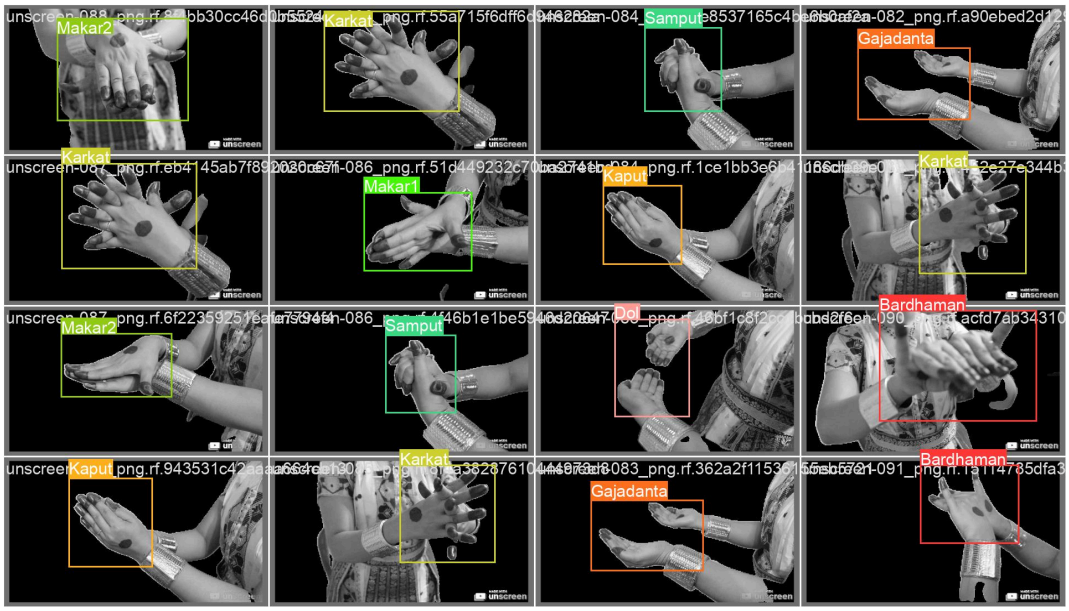


Figure 15. Validation Images of YOLOv8

5.2. YOLOv5 Model Performance

Training Outcome:

Table 3: Training Outcome of YOLOv5

Loss	Accuracy	
cls_loss	mAp50	mAp50-95
0.2365	0.995	0.820

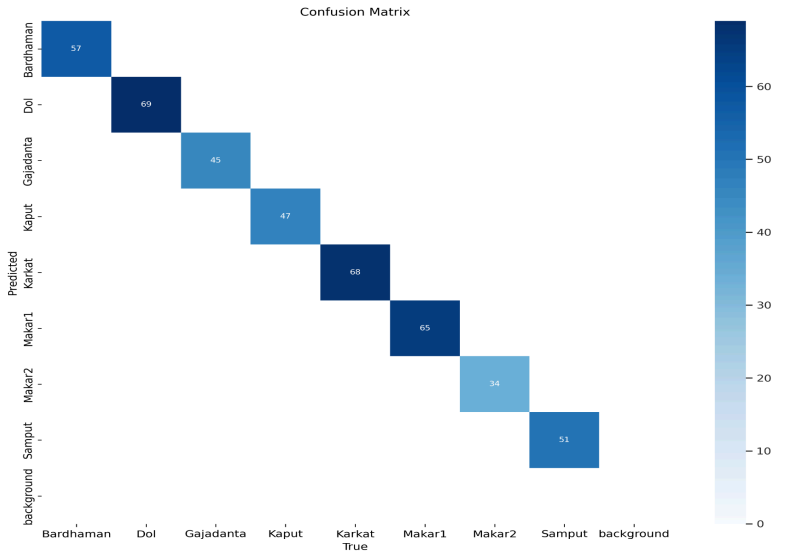


Figure 16: YOLOv5 Confusion Matrix

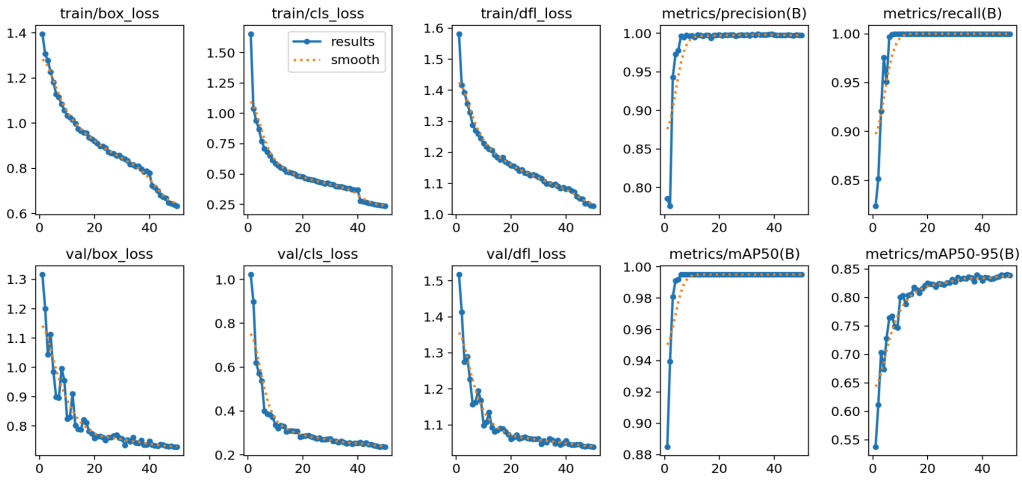


Figure 17: YOLOv5 Train Result

Validation Outcomes:

Table 4: Validation Outcome of YOLOv5

Class	mAp50	mAp50-95
Bardhaman	0.995	0.78
Dol	0.995	0.82
Gajadanta	0.995	0.835
Kaput	0.995	0.807
Karkat	0.995	0.816
Makar1	0.995	0.87
Makar2	0.995	0.906
Samput	0.995	0.81

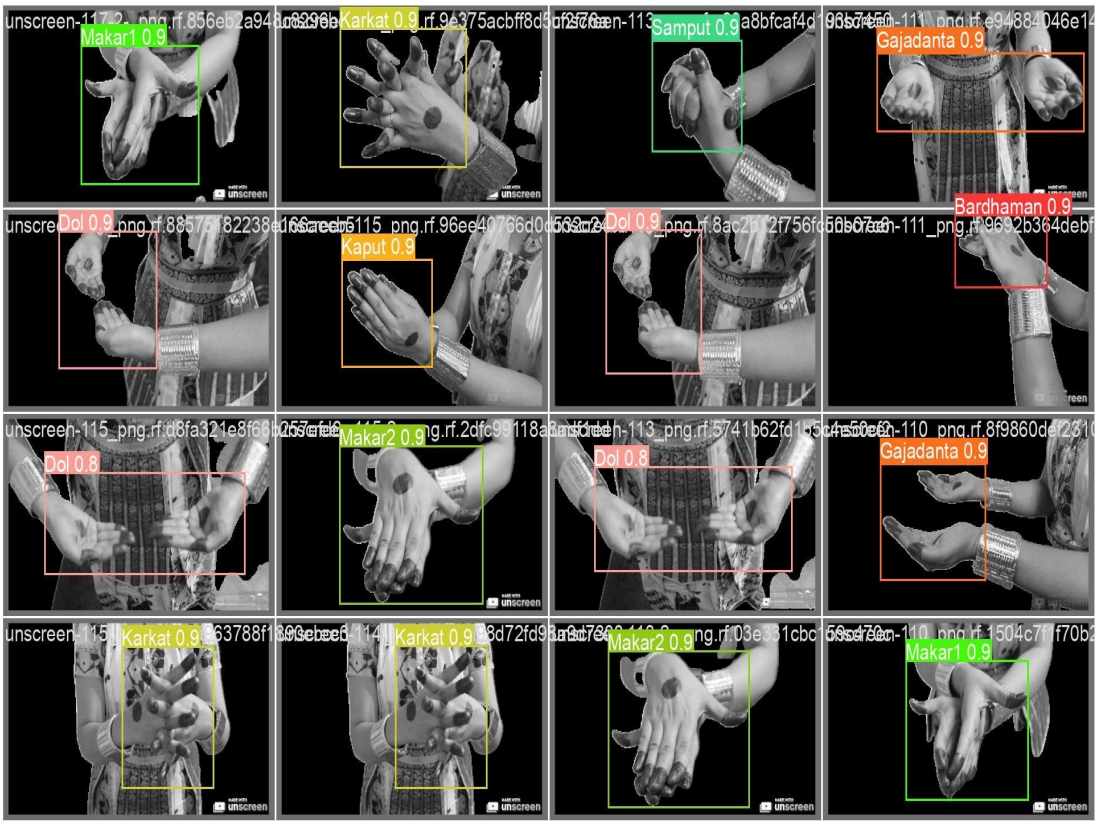


Figure 18. Validation Images of YOLOv5

5.3. RT-DETR Model Performance

Training Outcome:

Table 5: Training Outcome of RT-DETR

Loss	Accuracy	
cls_loss	mAp50	mAp50-95
0.24785	0.995	0.81813

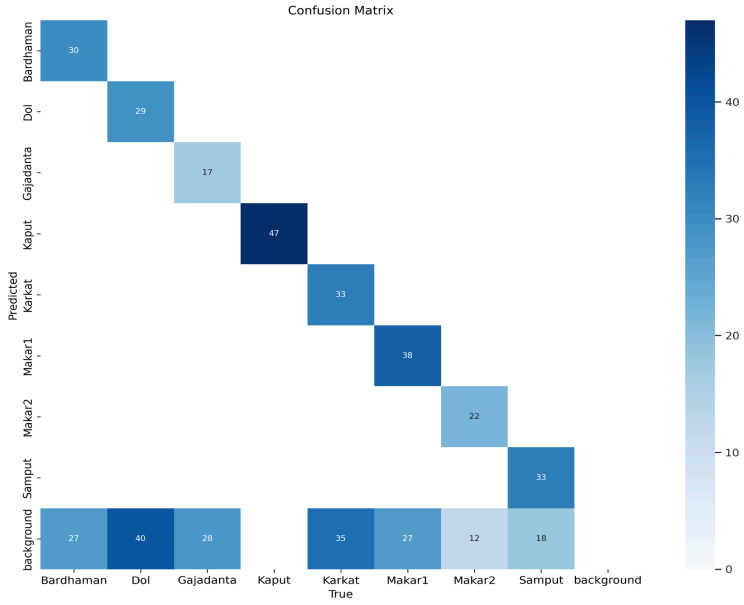


Figure 19. RT-DETR Confusion Matrix

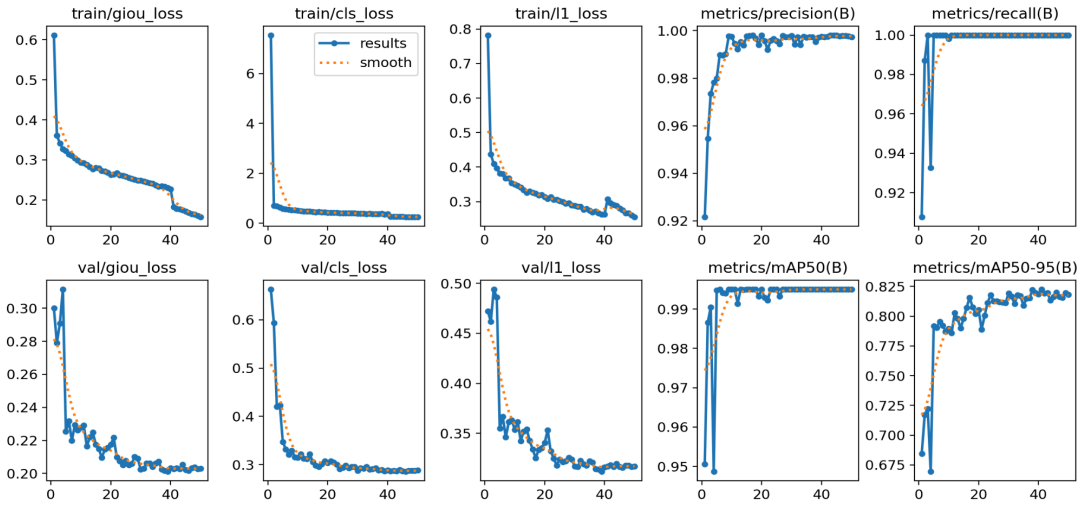


Figure 20. RT-DETR Train Result

Validation Outcome:

Table 6: Validation Outcome of RT-DETR

Class	mAp50	mAp50-95
Bardhaman	0.995	0.762
Dol	0.995	0.809
Gajadanta	0.995	0.835
Kaput	0.995	0.777
Karkat	0.995	0.804
Makar1	0.995	0.881
Makar2	0.995	0.896
Samput	0.995	0.803

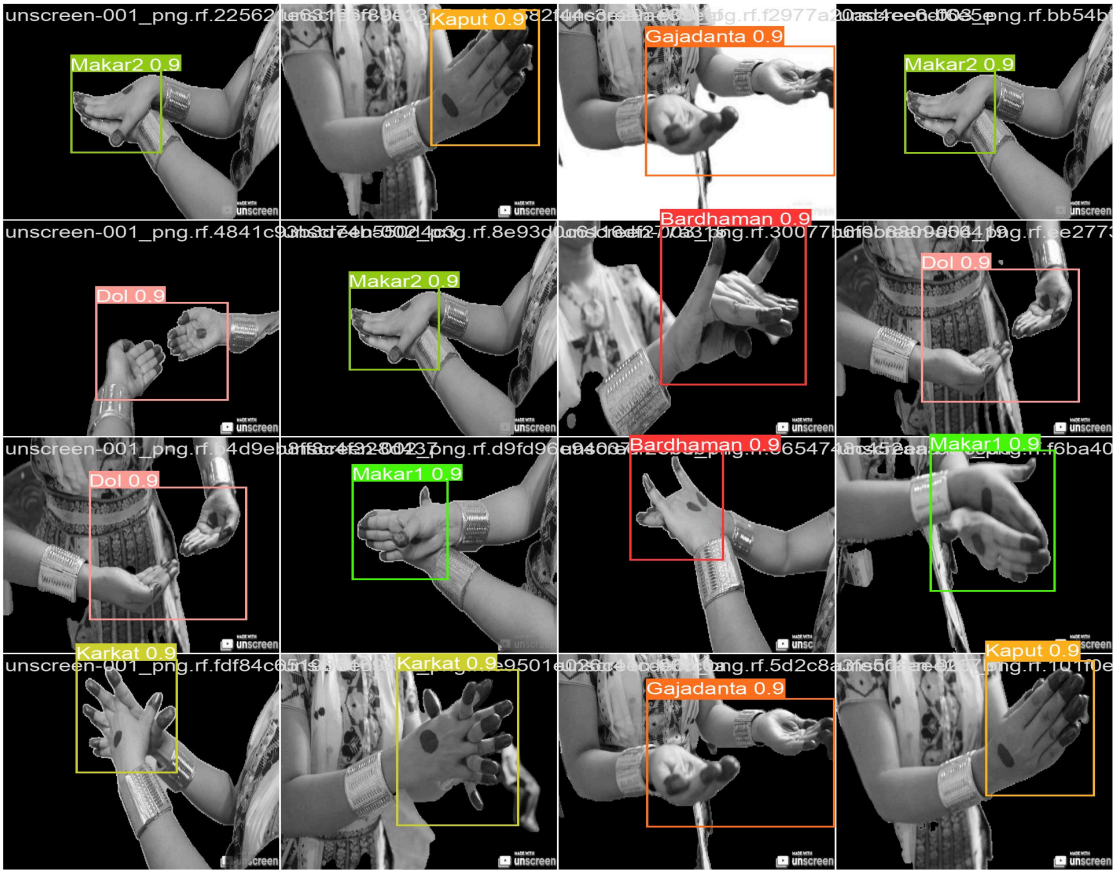


Figure 21: Validation Images of RT-DETR

5.4. Models Summary:

Table 7: Summary of YOLOv8, YOLOv5 and RT-DETR

Models	Epochs	Batch Size	CLS_Loss	Mean Average Precision(mAP50-95)	Recall
YOLOv8	50	16	0.22104	0.83758	1
YOLOv5	50	16	0.2365	0.820	1
RT-DETR	50	16	0.24785	0.8181	1

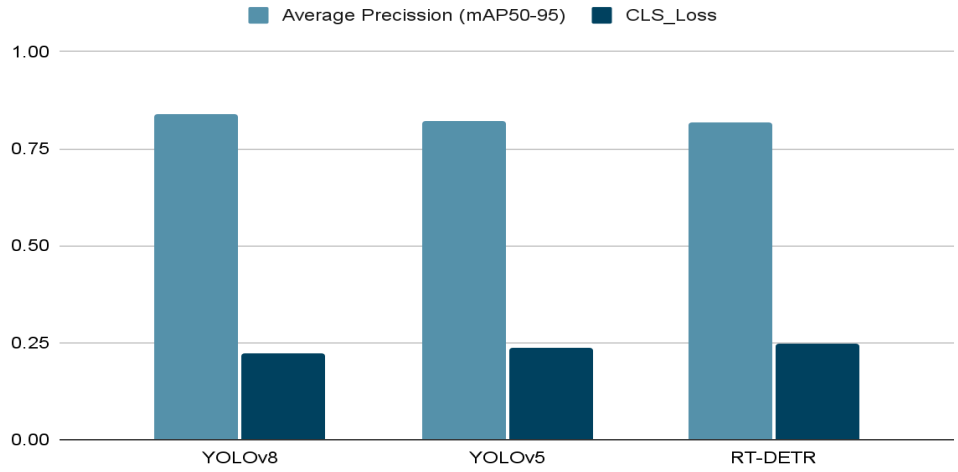


Figure 22. Comparison Between three models based on mAp50-95 and CLS_Loss

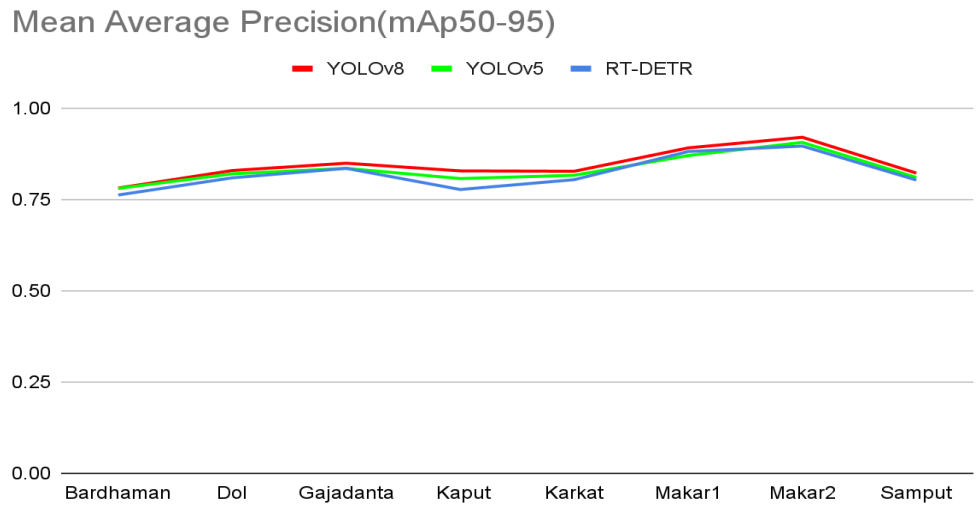


Figure 23:Mean Average Precision(mAp50-95) of all the models based on different classes

To identify the optimal model for deployment in our web application, we conducted a comprehensive evaluation of YOLOv8, YOLOv5, and RT-DETR. This evaluation focused on key metrics like precision and classification loss. Our analysis revealed that YOLOv8 emerged as the frontrunner, achieving superior performance in these critical areas. Consequently, YOLOv8's efficiency and accuracy make it the ideal choice for powering the user-friendly web application we've developed for Satriya dance mudra recognition.

5.5. YOLOv8 Shines for Web Deployment:

A comprehensive evaluation of YOLOv8, YOLOv5, and RT-DETR models, focusing on precision and classification loss, identified YOLOv8 as the most suitable choice for deployment in our web application. YOLOv8's superior performance in these critical areas translates to efficient and accurate mudra recognition. To ensure a user-friendly experience, we developed a web application utilizing Flask, HTML, CSS, and JavaScript. This combination provides a seamless interface for users to interact with the YOLOv8 model and access mudra recognition results, fostering wider accessibility and appreciation for this unique aspect of Sattriya dance.

Building a User-Friendly Interface: HTML, CSS, JavaScript, and Flask

The web application we developed utilizes a powerful combination of technologies to deliver a user-friendly experience. Here's a breakdown of their roles:

- HTML (Hypertext Markup Language): Forms the foundation of the web application's structure. It defines the content layout, including elements like headings, text paragraphs, and image placeholders.

- CSS (Cascading Style Sheets): Responsible for the visual aesthetics of the application. It controls the styling of the HTML elements, defining aspects like

fonts, colors, background images, and layout properties. This allows you to create a visually appealing and consistent user interface.

- JavaScript: Adds interactivity to the web application. It allows users to interact with various elements on the page, such as buttons or image uploads. JavaScript can also be used to process user input and dynamically update the web page content without requiring a full page refresh.

- Flask: A lightweight Python framework that serves as the backbone of the application's functionality. It handles tasks like routing user requests to the appropriate parts of the application, interacting with the YOLOv8 model behind the scenes, and formatting the results for display on the user interface. Flask simplifies the development process by
providing a clear structure for building web applications.

```python
from flask import Flask, render_template, request, redirect, send_file, url_for, Response

from ultralytics import YOLO



app = Flask(__name__)

@app.route("/")
def home():
    return render_template('index.html')


opdir='runs/detect'
@app.route("/", methods=["GET", "POST"])
```

Figure 24: Flask startup and configuration

```python
# perfome the detection
yolo = YOLO('best.pt')
# detections = yolo.predict(image, save=True)
results = yolo(filepath, conf=0.4, save=True, project=opdir)
```

Figure 25. Initializing the YOLOv8 model best.pt weight file

By working together, these technologies create a cohesive and user-friendly web application that allows users to easily interact with the Sattriya dance mudra recognition model

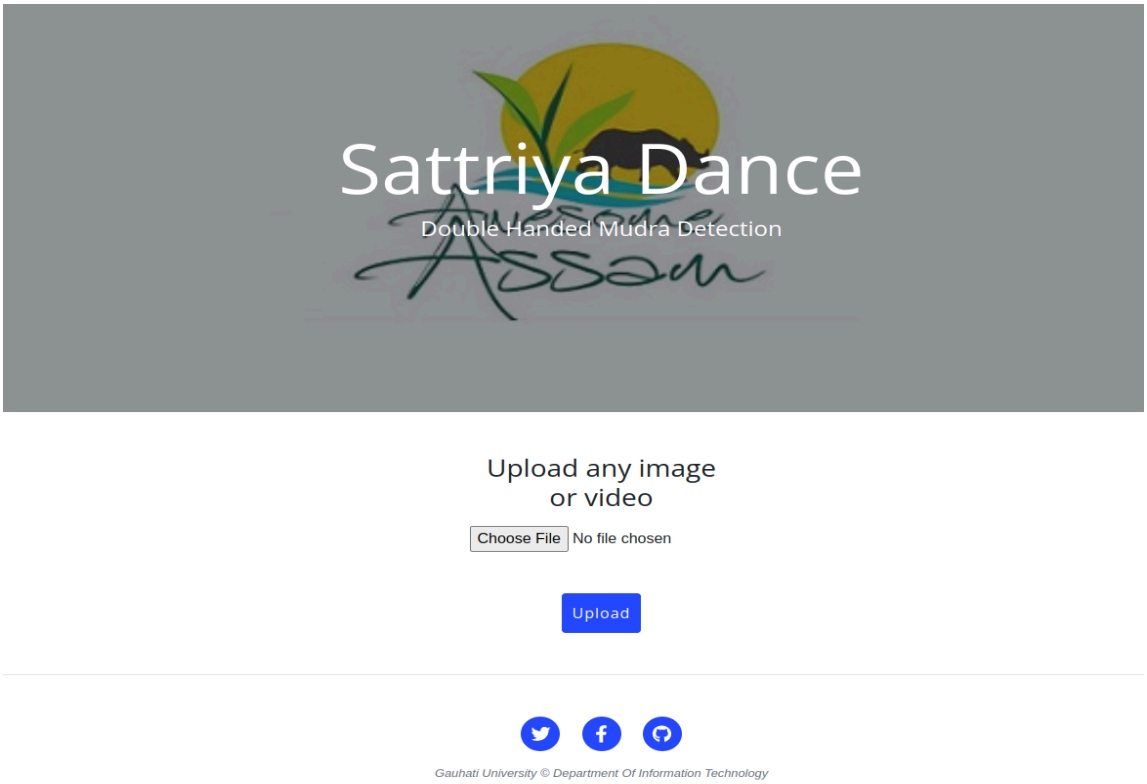


Figure 26: Glance of the web Application

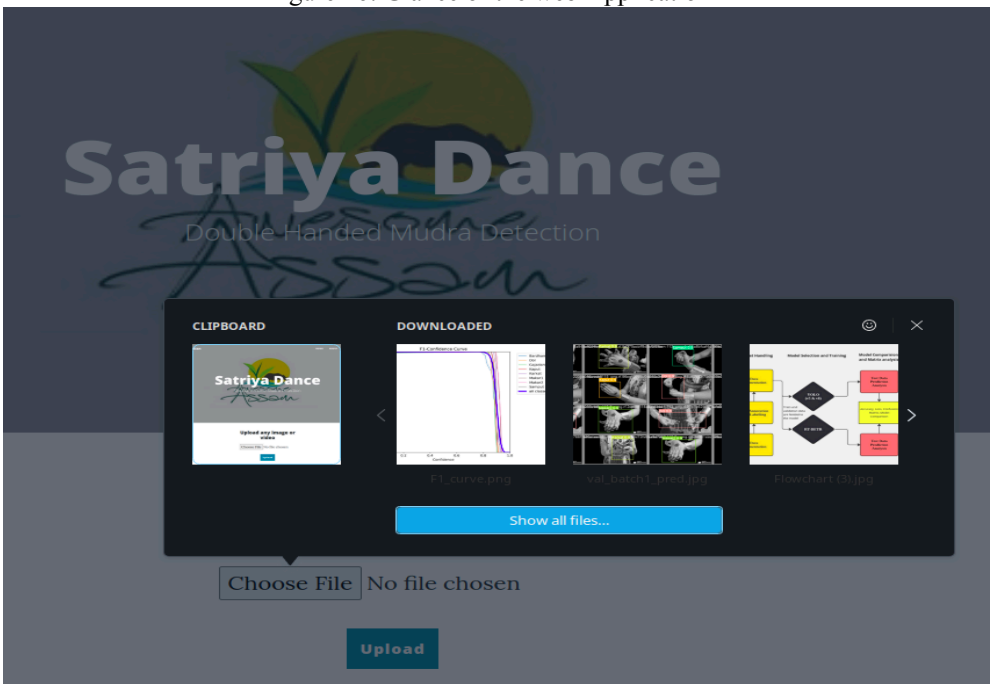


Figure 27: Uploading prompt of Image or Video For Detection

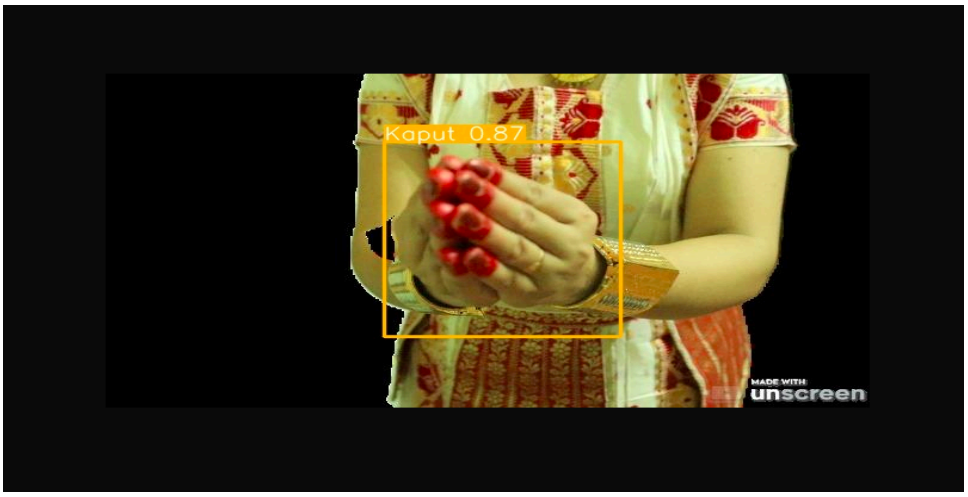


Figure 28: Image Detection made by the proposed model

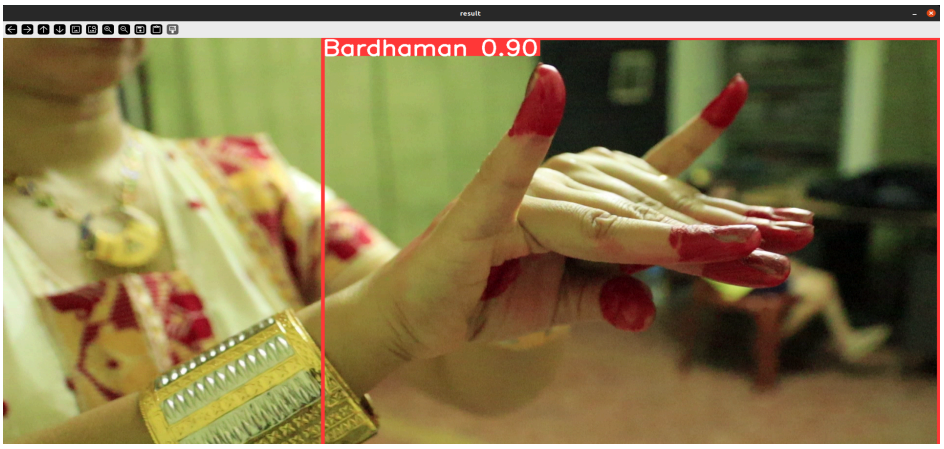


Figure 29. Video Detection made by the model

Challenges and Limitations:

Challenges:

Dataset Annotating Complexity:

Annotating the Sattriya dance dataset with accurate hand mudra labels posed a challenge due to the complexity and variability of dance performances. Ensuring precise annotations required extensive collaboration with experts in Sattriya dance.

Cultural Sensitivity:

Balancing technological accuracy with cultural sensitivity proved challenging. The project required continuous refinement to align with the cultural nuances and preferences of the Sattriya dance community, as some gestures may hold subtle variations.

Limited Training Data:

Acquiring a diverse and extensive dataset for training the models presented challenges. The limited availability of annotated images impacted the models' ability to generalize to various hand mudras, particularly those less represented in the dataset.

Real-time Processing Constraints:

Achieving real-time processing, especially during live performances, presented computational challenges. Optimizing the models for efficiency without compromising accuracy was a delicate balance.

Limitations:

Model Generalization:

The limitations imposed by the training dataset may limit the models' ability to generalize to previously undiscovered or unique hand mudras. Less accuracy may

be displayed when performing motions that were not sufficiently depicted during training.

Dependency on Image Quality:

The quality of the supplied photos is intrinsically related to their recognition accuracy. Poor or grainy photos can lead to reduced model performance, emphasizing how important clear images are for precise segmentation.

Sensitivity to Lighting Conditions:

During performances, changes in lighting may have an impact on the models. The accuracy of hand mudra recognition may be impacted by lighting variations, underscoring the necessity of robustness in a variety of performance settings.

Interference from Background Elements:

Complex performance backgrounds may introduce challenges in accurately segmenting hand mudras, particularly when there is overlap with other elements. Background interference can affect the precision of the recognition system.

Scalability Considerations:

There might not be much room for the project to grow to suit more dance styles or cultural settings. If Satriya dance is added to the application, the models and underlying architecture might need to be modified.

6. Conclusion and Future Works

Conclusion:

In conclusion, This project on Sattriya dance hand mudra recognition, leveraging the power of YOLOv8, YOLOv5, and RT-DETR algorithms, represents a significant leap forward in both technological advancement and cultural preservation. By integrating the best-performing model into a user-friendly web application, the project successfully addresses the intricate challenge of recognizing and safeguarding the rich cultural heritage embodied within Satriya dance mudras. This achievement not only paves the way for further research in computer vision and cultural documentation, but also fosters a deeper appreciation for this unique art form.

Key Achievements:

This project successfully addressed the challenge of recognizing double-handed mudras in Sattriya dance through image and video analysis. We achieved several key milestones:

- Effective Machine Learning Model Training: Through rigorous training, we developed a robust machine learning model capable of accurately recognizing mudras within images and videos. This model paves the way for the development of practical applications like automated mudra identification tools.
- Cross-Cultural Exploration Potential: Our exploration of applying the developed methodologies to other dance forms demonstrates the project's potential for broader cultural heritage applications. This opens doors for future research collaborations in the field of computer vision and cultural informatics.
- Enhanced Understanding and Appreciation: By creating a tool for mudra recognition, we contribute to a deeper understanding and appreciation of Sattriya dance. This fosters cultural preservation efforts and promotes public awareness of this unique art form.

- Diverse Recognition Approaches: The integration of YOLOv8, YOLOv5 and RT-DETR, showcases a comprehensive approach to hand mudra recognition. This diversity enables a multifaceted understanding of gestures within a cultural context.

These achievements lay a solid foundation for further advancements in Sattriya dance mudra recognition and analysis. The project's impact extends beyond technological innovation, fostering a bridge between cultural heritage and the potential of machine learning for its preservation.

Challenges Overcome:

- Technical and Cultural Balancing Act:

There were difficulties in striking a balance between cultural sensitivity and technical precision, but the project managed to do so successfully, honoring the artistic subtleties of Sattriya dance while utilizing state-of-the-art technology.

- Dataset Annotating Complexity:

Working with specialists, the Satriya dance dataset's annotation challenges were overcome, producing a diversified dataset that includes a large range of hand mudras and positions.

- Web Application Development:

Translating the power of the chosen model into a user-friendly web application presented its own set of challenges. The project team had to ensure the application was not only functional but also intuitive and accessible to users with varying technical backgrounds

Future Works:

As the project advances, there are promising areas for future development:

1. Fine-tuning and Model Refinement:

Continue fine-tuning settings and growing datasets to improve recognition models. This will guarantee continued accuracy and enhance the model's generalization to a wider range of hand mudras.

2. Real-time Adaptation:

Look for ways to enhance the real-time processing capabilities so that the system can detect dynamic Sattriya dance performances with varying gestures and speeds with ease.

3. Cultural Sensitivity and Collaboration:

By actively working together with the Sattriya dance community, the initiative is to blend cultural sensitivity with technological precision.

In summary, the Sattriya dance hand mudra recognition project, wielding the power of YOLOv8, YOLOv5, and RT-DETR algorithms, exemplifies a comprehensive approach to technological innovation for cultural preservation. This groundbreaking initiative culminates in a user-friendly web application, revolutionizing the intersection of technology and the safeguarding of traditional artistic expressions. As the project evolves, it promises to not only ensure the legacy of Sattriya dance mudras but also pave the way for the preservation of countless other cultural treasures, all readily accessible through this innovative web interface.

7. References

[1] R. Pradeep, R. Rajeshwari, V. R. Ruchita, R. Bubna and H. R. Mamatha, "Recognition of Indian Classical Dance Hand Gestures," 2023 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2023, pp. 814-820, doi: 10.1109/ICICT57646.2023.10134484. keywords: {Training;Measurement;Humanities;Image recognition;Computational modeling;Neural networks;Object detection;Indian classical dance forms;hand gestures;mudra;samyukta hasta;asamyukta hasta;image classification;Convolutional Neural Network models;Inception v1;SqueezeNet;object detection;YOLO v5}

[2] M. Safaldin, N. Zaghden and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," in IEEE Access, vol. 12, pp. 59782-59806, 2024, doi: 10.1109/ACCESS.2024.3393835.
keywords: {YOLO;Feature extraction;Real-time systems;Detectors;Computer architecture;Object recognition;Task analysis;Deep learning;Deep learning;localization;object detection;segmentation;YOLO}

[3] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." multimedia Tools and Applications 82.6 (2023): 9243-9275.

[4] SANTHOSH, REMYA and KK, Rajkumar, Classification of Asmyukta Mudras in Indian Classical Dance Using Handcrafted and Pre-Trained Features with Machine Learning and Deep Learning Methods. Available at SSRN: https://ssrn.com/abstract=4818826 or http://dx.doi.org/10.2139/ssrn.4818826

[5] M. B. Ullah, "CPU Based YOLO: A Real Time Object Detection Algorithm," 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 552-555, doi: 10.1109/TENSYMP50017.2020.9230778. keywords: {Central Processing Unit;Computers;Object detection;Real-time systems;Graphics processing units;Task analysis;Computational modeling;object detection;real time;YOLO;CPU;deep learning}

[6]M. Devi and S. Saharia, "A two-level classification scheme for single-hand gestures of Sattriya dance," 2016 International Conference on Accessibility to Digital World (ICADW), Guwahati, India, 2016, pp. 193-196, doi: 10.1109/ICADW.2016.7942540. keywords: {Feature extraction;Support vector machines;Decision trees;Gesture recognition;Matrix converters;Indexes;Asamyukta hastas;hand gestures;Indian Classical Dance;Medial Axis Transformation}

[7]H. Bhuyan, P. P. Das, J. K. Dash and J. Killi, "An Automated Method for Identification of Key frames in Bharatanatyam Dance Videos," in IEEE Access, vol. 9, pp. 72670-72680, 2021, doi: 10.1109/ACCESS.2021.3079397.
keywords: {Videos;Motion segmentation;Feature extraction;Support vector machines;Deep learning;Annotations;Image segmentation;Key frame;Adavu;three frame difference;bit-plane extraction;adaptive threshold;machine learning},

[8]Joko S. Dance gesture recognition using space component and effort component of laban movement analysis. International Journal of Scientific & Technology Research. 2020;9(2):3389-94.

[9]Kico I, Grammalidis N, Christidis Y, Liarokapis F. Digitization and visualization of folk dances in cultural heritage: A review. Inventions. 2018 Oct 23;3(4):72.

[10]Bhushan S, Alshehri M, Keshta I, Chakraverti AK, Rajpurohit J, Abugabah A. An experimental analysis of various machine learning algorithms for hand gesture recognition. Electronics. 2022 Mar 21;11(6):968.

[11]Naik AD, Supriya M. Classification of indian classical dance images using convolution neural network. In2020 International Conference on Communication and Signal Processing (ICCSP) 2020 Jul 28 (pp. 1245-1249). IEEE.

[12]V . Janaranjani1, N. Megala2, R.Naveeth Kumar3 1,2,3Assistant Professor in Biomedical Engineering, Rathinam Technical Campus, Coimbatore Hand Gesture Recognition for Dance Movements. © March 2022| IJIRT | Volume 8 Issue 10 | ISSN: 2349-6002