Spatial Structure-Oriented and Angle-Based Human Pose Estimation for Pose Classification

P Charith[1], Prajwal Kumar B R[2], Anitha M[3]

[1,2]Student, Computer science and engineering, Dayananda Sagar college of engineering
[3]Assistant Professor, Computer science and engineering, Dayananda Sagar college of engineering

Abstract

This study provides a detailed analysis of the performance of different pose classification models trained using data from the human pose classification model. The approach involves considering both spatial structure-oriented techniques, which incorporate body part coordinates and their relative positions, and angle-based methods that calculate the angles between joints. This combined spatial and angular data play a crucial role in enhancing the precision of pose classification. It is worth noting that while our primary investigation is based on a yoga pose dataset, the versatility and applicability of our approach extend to other pose datasets, showcasing the broad potential of our spatial and angle-based methodology.

In summary, this research embarks on the integration of Human Pose Estimation with machine learning for yoga pose classification. The outcomes promise not only to advance the field of pose classification but also to yield practical applications in exercise, fitness, and beyond. This research has practical implications, aiming to integrate the developed model into a project we developed titled "**AI-Based Human Pose Detection Tool**. "The tool uses real-time video analysis to track users' movements during workouts, with the Blazepose model detecting key landmarks and assessing metrics. This enhances posture and form assessment, making the tool valuable for fitness enthusiasts.

Keywords: Human pose estimation, computer vision, machine learning, pose classification

1. Introduction

Human Pose Estimation in computer vision is a transformative technology decoding human body language, empowering machines to understand postures and movements. With applications in fitness, sports, medical diagnostics, and gaming, its core focus is characterizing key body part positions, bridging the physical-digital gap, and promising a revolution in perception and interaction.

The synergy between Human Pose Estimation (HPE) and classification techniques further amplifies this technology's significance. Our research centers on integrating the Blazepose model, an HPE model, with machine learning techniques for yoga pose classification. This nuanced approach to interpreting human movement is showcased through the exploration of five fundamental yoga poses: down dog, goddess, plank, tree, and warrior.

The effectiveness of our approach is evident in the utilization of data from human pose estimation models compared to the direct use of raw images. These models excel in abstracting vital pose information by focusing on key joint positions and relationships, streamlining feature extraction. Their robustness to variations in lighting, background, and environmental conditions enhances classification accuracy. The inherent dimensionality reduction, coupled with generalization capabilities and computational efficiency, solidifies their suitability for classification tasks. Moreover, the interpretability of these models provides clearer insights into the classification rationale, a feature often challenging to achieve with raw image data. In summary, our research demonstrates that incorporating data from human pose estimation models optimizes classification through abstraction, robust feature extraction, and enhanced interpretability, ultimately improving accuracy and efficiency in pose classification tasks.

2. Related Works

Our research drew from various papers, offering insights into human pose estimation and diverse dataset normalization techniques, with a focus on methods for improvement. Vivek Anand Thoutam et al. [1] introduced a yoga pose classification method using joint and key point angles. However, they did not address dataset normalization, a critical factor for data standardization and reducing variations in joint positions and angles.

Utkarsh Bahukhandi et al. [3] attained a notable 94% accuracy by employing joint coordinates to train logistic regression and SVM models. However, they did not include joint angles as training parameters and omitted dataset normalization, which are essential for ensuring data consistency and accurate model training. In contrast, Steven Chen et al. [2] introduced key point normalization as part of their work on constructing a pose trainer. This concept shed light on valuable data processing techniques, particularly normalization by reference points, which we have considered for implementation in our research.

Ashish Ohri et al. [5] introduced a pose correction method using Dynamic Time Warping and emphasized the effectiveness of the MediaPipe model, known for its exceptional accuracy, speed, and robustness in real-time human pose estimation. Rohit Srivatsa et al. [4] conducted a comparative analysis of various open-source pose estimation models. This comparative study proved instrumental in guiding our selection of the most suitable model for our specific use case.

Sen Qiao et al. [9] introduced a cost-effective system for grading human gestures using Openpose, focusing on a novel approach based on the spatial distance between joints, potentially improving model accuracy.

Anilkumar et al. [10] proposed a self-practice yoga monitoring system using angle conditions among key joints. Yet, manual angle input may be time-consuming. Incorporating joint coordinates alongside angles could provide a more comprehensive solution. As a part of our research, we have considered 2 datasets, one with only coordinates and the other with coordinates and joint angles, to evaluate the importance of the angle parameter.

Prof. Rupal More et al. [12] solely utilized Logistic Regression for yoga pose classification, but it may not always be the ideal choice. They did not explore potential benefits from model fine-tuning or alternative models. In our research, we have included 4 models for training and analyzed each model's performance. Chhaihuoy Long et al.[11] proposed a transfer learning method for pose classification. Unlike image-based classification, using an HPE model to extract joint coordinates and angles offers more accurate and robust data, improving classification accuracy and resilience to image-related challenges.

These research contributions informed our work on human pose estimation, dataset processing, model evaluation, and performance analysis, prompting us to explore more efficient and effective methods for data processing and model tuning.

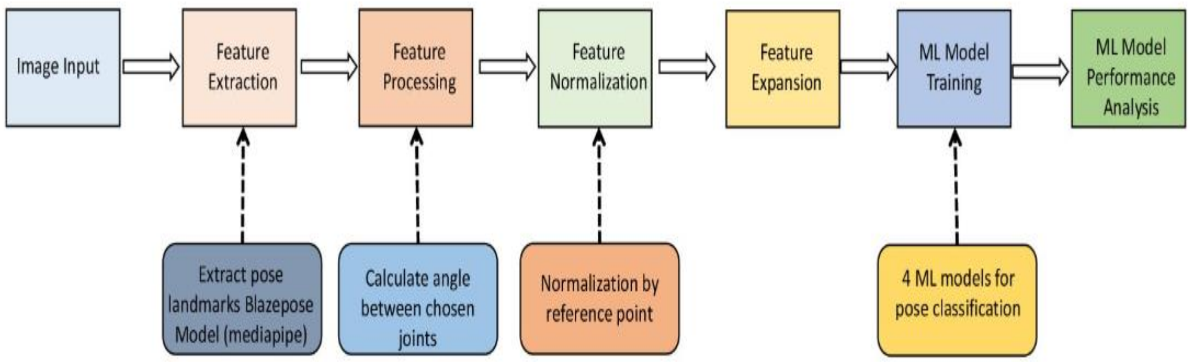
3. Methodology

This research focuses on three core objectives:

1. **Processing Yoga Pose Image Data**: Dataset processing, normalization, and joint angle integration using Blazepose HPE model coordinates for improved classification.
2. **Evaluation of Classification Models**: Rigorous evaluation of machine learning models for yoga pose classification, focusing on accuracy and efficiency with body part coordinates and joint angles.
3. **Analysis of Model Performance**: In-depth analysis of classification models, providing insights into their strengths, limitations, and effectiveness in yoga pose classification.

The machine learning models at the heart of our research include - Multilayer Perceptron (MLP), Random Forest Classifier, Support Vector Machine (SVM), and XGBoost.

Figure 1: The above figure illustrates the overview of the research process involved



The methodology comprises four primary steps:

1. **Feature Extraction:** Feature extraction is the initial step, involving the use of the Blazepose model from MediaPipe to extract the coordinates of 33 body parts from each image. These coordinates serve as the fundamental data for subsequent processing and classification
2. **Feature Processing:** Following feature extraction, the extracted coordinates are further processed. This step includes the calculation of six key angles: 'left_arm_angle,' 'right_arm_angle,' 'left_shoulder_angle,' 'right_shoulder_angle,' 'left_knee_angle,' and 'right_knee_angle.' These angles provide valuable insights into the posture and form of the yoga poses.
3. **Feature Normalization:** Normalization of the extracted coordinates is essential to ensure consistent and uniform data. Here the normalization process is performed with respect to a reference point.
4. **Pose Classification:** The final step in the methodology is pose classification. This stage encompasses the actual classification of yoga poses using the processed and normalized data.

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

3.1 Dataset

The "Yoga Poses Dataset" on Kaggle is a comprehensive resource for yoga pose classification. It features images of individuals performing various yoga poses, including "Downward Dog," "Goddess Pose," "Plank Pose," "Tree Pose," and "Warrior2 Pose." The dataset is well-curated, offering diverse images captured from different angles to ensure dataset completeness. The below table gives an idea of the dataset.

Table 1: Dataset used

Yoga Poses	Images	
	Train	Test
Downward dog pose	223	97
Goddess pose	180	80
Plank pose	266	115
Tree pose	160	69
Warroir2 pose	252	109

Dataset Structure:

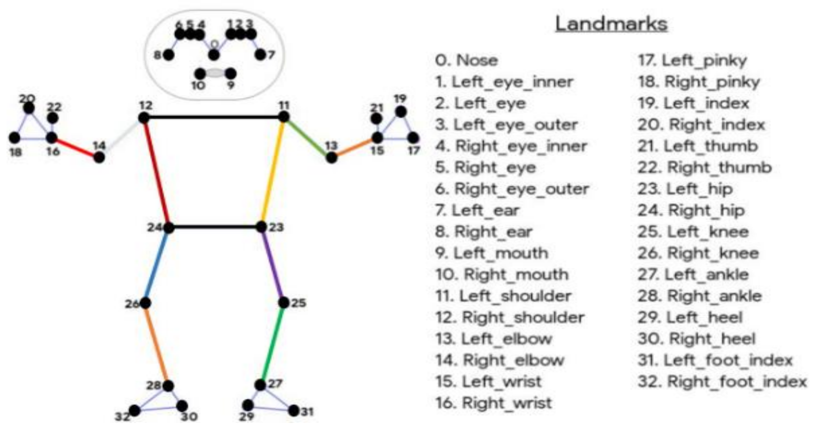
This dataset is divided into "train" and "test" folders, further organized into subfolders corresponding to five distinct yoga pose classes. Merging test and train images within each subfolder creates a unified dataset for machine learning model training and testing.

3.2 Feature Extraction using Pose Estimation Model

During this stage, the Blaze Pose model, provided by MediaPipe, is harnessed to extract 33 pose landmarks, each defined by the following attributes:

1. **x and y**: The coordinates of these landmarks are standardized, falling within the range of [0.0, 1.0], relative to the image's width and height, respectively.
2. **z**: This value represents the depth of each landmark, measured from the reference point of the hips' midpoint. A smaller z value indicates the landmark's proximity to the camera, and this scale is somewhat akin to that of x.
3. **visibility**: Expressed as a numerical value between 0.0 and 1.0, this parameter signifies the probability of a landmark being observable in the image. A higher visibility value implies a greater likelihood that the landmark is both present and unobstructed within the image.

Figure 2: The above diagram shows the body parts for which Blazepose provides pose landmarks



In this stage, the process is executed for each image within the image dataset. In essence, this involves creating a comprehensive dictionary that encompasses the attributes of various body parts, thereby providing a detailed representation of the pose landmarks. Simultaneously, the label, which corresponds to the folder name and signifies the specific yoga pose depicted in the image, is thoughtfully incorporated into this dictionary.

This stage extends beyond pose landmarks and labels, involving the calculation and inclusion of a list of angles. These angles play a crucial role, detailed in the subsequent feature processing stage, enhancing our comprehension of each yoga pose

In this process, images without pose landmarks are excluded from the dataset, while images with landmarks are collected into a list. Each list entry is enriched with pose landmarks and the corresponding label (folder name). This method culminates in a list containing these augmented dictionaries.

3.3 Feature Processing

Following feature extraction, the Feature Processing stage is a pivotal phase in our research. During this stage, the extracted pose landmarks are further enriched by calculating six essential angles associated with specific body parts. Figure 3 shows the angles considered. These angles, namely 'left_arm_angle'(1), 'right_arm_angle'(2), 'left_shoulder_angle'(3), 'right_shoulder_angle'(4), ‘left_knee_angle' (5) and 'right_knee_angle'(6) provide detailed insights into the posture and form of the yoga poses. This step enhances the dataset with valuable information, which is vital for precise classification.

Figure 3: Angles considered

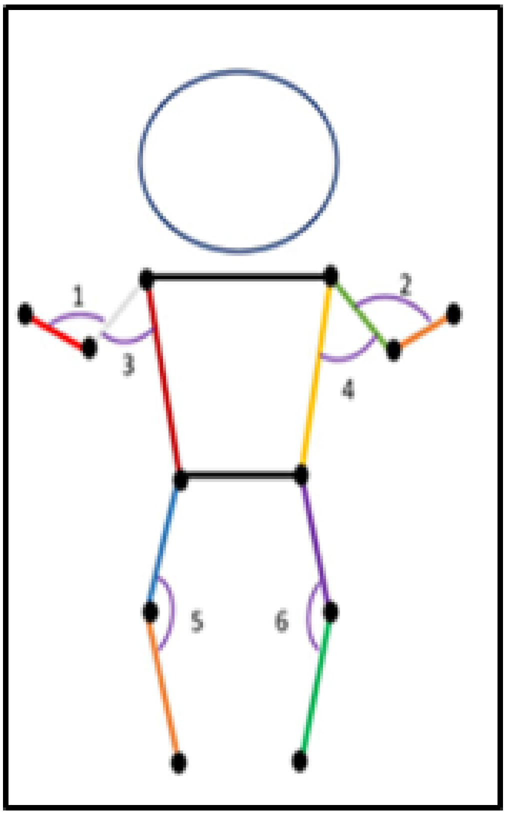
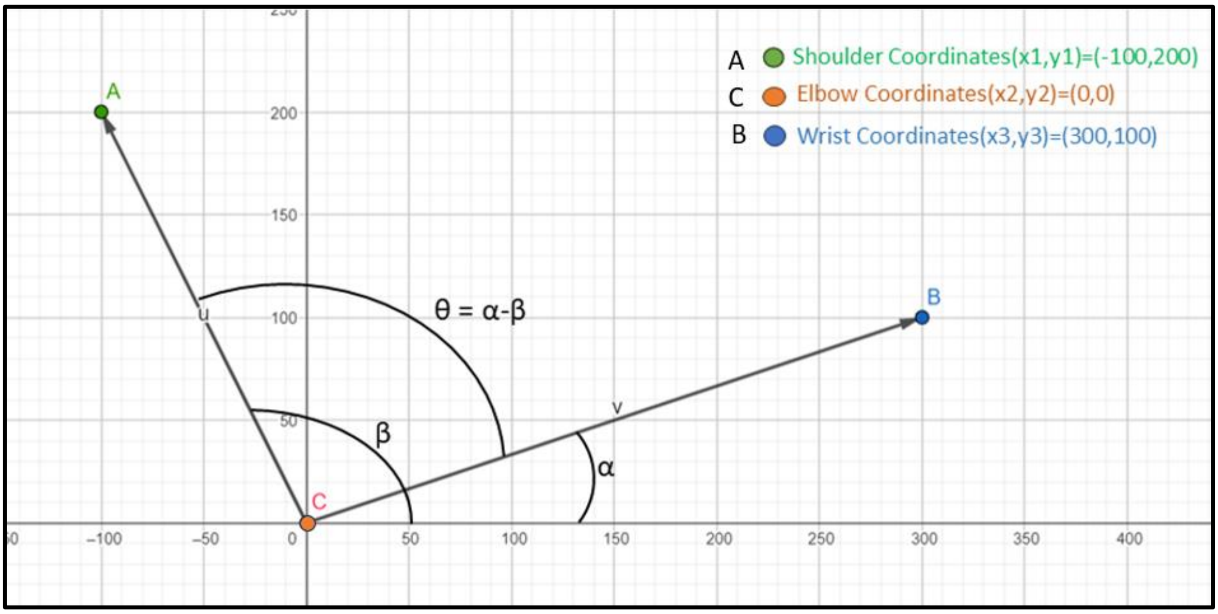


Figure 4: Angle calculation example



To calculate these angles, a trigonometric approach is employed. The formula used to calculate the angle (θ) between three points (x1, y1), (x2, y2), and (x3, y3) in a 2D space is expressed as:

θ_rad = atan2(y3 - y2, x3 - x2) - atan2(y1 - y2, x1 - x2)

In this formula:
- (x1, y1), (x2, y2), and (x3, y3) represent the coordinates of the three points.
- Here, atan2(dy, dx) is the arctangent function that calculates the angle formed by the vector (dx, dy) with respect to the positive x-axis. The subtraction of these arctangents gives the difference in angles between the two-line segments, resulting in the angle θ_rad in radians.

Figure 4 shows the implementation of angle calculation with an example. Three points are considered A(x1,y1)=(-100,200), B(x3,y3)=(300,100) and C(x2,y2)=(0,0). The coordinates are provided in Figure 4.

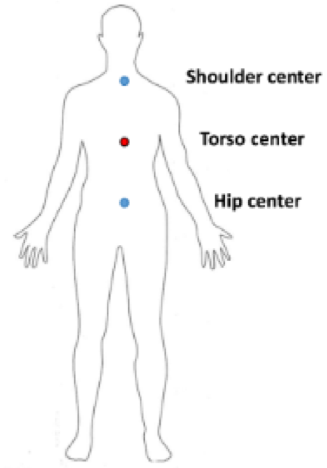
Calculation:
α = atan2(y3-y2,x3-x2) = atan2(100,300) = 0.32 radians = 18.33 degrees
β = atan2(y2-y1,x2-x1) = atan2(-200,100) = -1.10 radians = -63.03 degrees
θ = α – β = 18.33 – (-63.03) ~ 81.4 degrees

The meticulous calculation of these angles contributes to a more comprehensive understanding of the yoga poses. The angles are calculated for each image in the dataset, offering valuable geometric insights into the orientation of body parts. This feature processing stage significantly enhances the dataset, providing both the landmark coordinates and the angles required for accurate pose classification.

3.4 Feature Normalization

Data normalization plays a critical role in preparing pose coordinates for effective classification. It's essential because it ensures that data is consistent and ready for analysis. In the context of pose estimation, normalization standardizes the data and focuses on relative body part positions.

Figure 5: Torso center



We employ a specific method known as "**Normalization by reference point**" In this method, the chosen reference point is the "**torso center**," which is strategically located at the midpoint between the shoulder center and hip center. This reference point is pivotal because it effectively centers and scales the pose data based on a stable, central location within the body.

The advantage of using this method includes:

1. **Centering and Scaling:** By centering the data on the torso center, we eliminate variations arising from the position or size of the subject, thereby making the data consistent and comparable.
2. **Preserving Relative Relationships:** This method is particularly useful in tasks where maintaining the relative relationships between body parts is essential. It ensures that the model can focus on how different body parts are positioned concerning the torso center.

Overall, feature normalization through the use of a reference point like the torso center optimizes the dataset for accurate pose classification. It mitigates potential biases introduced by variations in body size and posture, emphasizing the relative positioning of body parts. This prepares the data for the subsequent stages of the research, ensuring that the machine learning models operate on standardized and meaningful inputs. Figure 6 gives the normalization algorithm implemented.

Figure 6: Normalization by Torso Center Algorith

For each record in data do
   Create a copy of the record as `pose_data`.
   Remove the 'Label' and angles specified in `angle_list` from `pose_data`.

Calculate the 'torso center' as follows:
   Calculate the average x-coordinate (x_center),y-coordinate (y_center) and z-coordinate (z_center) for LEFT_SHOULDER, RIGHT_SHOULDER, LEFT_HIP, and RIGHT_HIP using the following formula:
   x_center = (LEFT_SHOULDER['x'] + RIGHT_SHOULDER['x'] + LEFT_HIP['x'] + RIGHT_HIP['x']) / 4

```
  y_center = (LEFT_SHOULDER['y'] + RIGHT_SHOULDER['y'] + LEFT_HIP['y'] + RIGHT_HIP['y']) / 4
  z_center = (LEFT_SHOULDER['z'] + RIGHT_SHOULDER['z'] + LEFT_HIP['z'] + RIGHT_HIP['z']) / 4

Store these averages in the `torso_center` dictionary: { 'x': x_center, 'y': y_center, 'z': z_center }.
Initialize an empty dictionary `normalized_pose_data`.
For each `body_part` and `coordinates` in `pose_data` do
  Calculate the normalized coordinates as follows:
  normalized_x = x[body_part] - torso_center['x'].
  normalized_y = y[body_part] - torso_center['y'].
  normalized_z = z[body_part] - torso_center['z'].
  Preserve the confidence level associated with the body part: normalized_confidence = coordinates[f"{body_part}_confidence"].
  Store the normalized coordinates in `normalized_pose_data` using `body_part` as the key: { 'x': normalized_x, 'y': normalized_y, 'z': normalized_z, f"{body_part}_confidence": normalized_confidence }.

Update the original record with the `normalized_pose_data`.
End loop (For each record in data).
End Process.
```

3. 5 Feature Expansion

This stage of feature expansion is crucial for enriching the dataset. It involves extending the number of columns for each body part, providing a more comprehensive view of the data. As an example, consider the 'nose' body part, which is expanded into four distinct columns: 'nosex,' 'nosey,' 'nosez,' and 'nose_confidence.' This expansion process is applied uniformly to all body parts, effectively enhancing the dataset with detailed and valuable information.

In the previous stage, the dataset comprised 33 pose landmarks, 6 angles, and a label, totaling 40 columns. However, after completing this stage of feature expansion, the dataset now includes a total of 33 * 4 columns for pose landmarks (one set of four columns for each body part), 6 angle columns, and the label, resulting in a dataset with a grand total of 139 columns. This expanded dataset is better equipped to provide comprehensive insights for accurate pose classification. This data is stored as a CSV file for ML model training.

3.6 Machine Learning Model Training

In the Machine Learning Model Training stage, we employ four distinct models, each offering its unique strengths for the yoga pose classification task:

1. **Multilayer Perceptron (MLP):** Proficient in handling intricate data and capturing complex patterns.
2. **Random Forest Classifier:** Known for robustness and adaptability, key in yoga pose classification.
3. **Support Vector Machine (SVM)**: Contributes data point discrimination abilities for precise pose classification.

4. **XGBoost**: Celebrated for speed and enhanced model efficiency and accuracy.

3.7 ML Model Configurations

3.7.1 Random Forest Classifier

```
RandomForestClassifier(n_estimators=100, random_state=42)
```

3.7.2 Multilayer Perceptron(MLP)

```
keras.Sequential([
keras.layers.Dense(128, activation='relu',   input_shape=(X.shape[1],)),
keras.layers.BatchNormalization(),
keras.layers.Dropout(0.5),
keras.layers.Dense(64, activation='relu'),
keras.layers.BatchNormalization(),
keras.layers.Dropout(0.5),
keras.layers.Dense(32, activation='relu'),
keras.layers.BatchNormalization(),
keras.layers.Dropout(0.5),
keras.layers.Dense(len(label_encoder.classes_), activation='softmax')
])
```

3.7.3 Support Vector Machine(SVM)

```
SVC(kernel='linear', C=1.0, random_state=42)
```

3.7.4 XGBoost

```
XGBClassifier(n_estimators=100, random_state=42)
```

3.8 Next Steps

In this stage, the dataset is divided into training and testing data using Python libraries, ensuring a robust evaluation of model performance.

The Machine Learning Model Training phase is divided into two crucial phases:

Phase 1 - Training ML Models on Expanded Data with Only Coordinates Columns (133 columns): This initial step involves training machine learning models using the expanded dataset, which exclusively contains coordinate columns. This phase assesses the performance of models that rely solely on geometric information.

Phase 2 - Training ML Models on Expanded Data with Both Coordinates and Angles Columns (139 columns): In the subsequent step, the training expands to include the dataset with both coordinate and angle columns. This comprehensive dataset equips models with additional information from the calculated angles, enabling a more nuanced evaluation of performance.

4. Results

In the Results stage, we will delve into the outcomes of the previous stage, specifically the performance of our machine learning models on the datasets. This stage encompasses two essential phases for evaluation:

Phase 1 - ML model performance on Data with Only Coordinates Columns
Phase 2 - ML model performance on Data with Both Coordinates and Angles Columns

For each of these models, precision, recall, and F1 score will be presented.

1. **Precision**: Precision represents the ratio of true positive predictions to the total number of positive predictions. In the context of yoga pose classification, precision tells us how many correctly predicted yoga poses were actually correct.
2. **Recall**: Recall, often called sensitivity, is the ratio of true positive predictions to the total number of actual positive instances. In our scenario, recall indicates how many of the actual yoga poses were correctly predicted.
3. **F1 Score**: The F1 score is the harmonic mean of precision and recall. It provides a balanced assessment of a model's performance, considering both false positives and false negatives. A higher F1 score signifies a model's ability to achieve both high precision and recall.
4. **Accuracy**: Accuracy measures the percentage of correctly predicted instances, considering both true positives and true negatives. While accuracy is a valuable overall metric, precision, recall, and F1 score are essential when dealing with imbalanced datasets or when different misclassification costs exist.

These metrics are essential for understanding how well the models classify the different yoga poses. Additionally, the accuracy for each model (based on performance on testing data) is provided to offer a quantitative assessment of their overall performance. The precision, recall, and F1 score provide insights into the strengths and weaknesses of each machine learning model, facilitating a detailed discussion of their performance.

4.1 Phase 1 Results

Table 2: Performance of Random Forest Classifier

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	**1.00**	**0.91**	**0.95**
Goddess Pose	**0.95**	**0.80**	**0.87**
Plank Pose	**0.88**	**0.99**	**0.93**
Tree Pose	**0.93**	**0.98**	**0.95**
Warrior2 Pose	**0.87**	**0.91**	**0.89**

Table 3: Performance of Multilayer Perceptron

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	**1.00**	**0.97**	**0.98**
Goddess Pose	**0.86**	**0.63**	**0.73**

Plank Pose	**0.93**	**0.99**	**0.96**
Tree Pose	**0.72**	**0.98**	**0.83**
Warrior2 Pose	**0.85**	**0.80**	**0.83**

Table 4: Performance of Support Vector Machine

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	**1.00**	**0.98**	**0.99**
Goddess Pose	**0.90**	**0.71**	**0.80**
Plank Pose	**0.92**	**0.99**	**0.95**
Tree Pose	**0.95**	**0.98**	**0.97**
Warrior2 Pose	**0.85**	**0.91**	**0.88**

Table 5: Performance of XGBoost

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	**1.00**	**0.98**	**0.99**
Goddess Pose	**0.93**	**0.82**	**0.87**
Plank Pose	**0.92**	**1.00**	**0.96**
Tree Pose	**0.98**	**0.98**	**0.98**
Warrior2 Pose	**0.90**	**0.91**	**0.90**

Table 6: Accuracy for all ML models

Model	Random Forest Classifier	Multilayer Perceptron	Support Vector Machine	XGBoost
Accuracy	**0.92**	**0.88**	**0.92**	**0.94**

Phase 1 Results Discussion

In Phase 1 Results, we assessed four machine learning models for yoga pose classification: Random Forest Classifier, Multilayer Perceptron (MLP), Support Vector Machine (SVM), and XGBoost, using accuracy, precision, recall, and F1-score.

1. **Random Forest**: 91.93% accuracy, strong precision, slightly lower recall for 'goddess,' robust F1-scores.
2. **MLP**: 87.72% accuracy, high precision for 'down dog' and 'plank,' fair F1-scores due to 'goddess' recall.
3. **SVM**: 91.93% accuracy, excellent precision, lower 'goddess' recall, strong F1-scores for 'down dog' and 'plank.'
4. **XGBoost**: Top performer, 94.04% accuracy, strong precision for 'down dog,' 'plank,' and 'tree,' high F1 scores.

In conclusion, XGBoost is the most promising model, with high accuracy, balanced precision-recall, and significant real-world application potential. Further research and optimization are necessary.

4.2 Phase 2 Results

Table 7: Performance of Random Forest Classifier

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	1.00	0.91	0.95
Goddess Pose	0.93	0.82	0.87
Plank Pose	0.91	1.00	0.95
Tree Pose	0.95	0.98	0.97
Warrior2 Pose	0.88	0.92	0.90

Table 8: Performance of Multilayer Perceptron

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	0.97	0.98	0.97
Goddess Pose	0.60	0.84	0.70
Plank Pose	0.89	0.90	0.89
Tree Pose	0.83	0.58	0.68
Warrior2 Pose	0.83	0.73	0.77

Table 9: Performance of Support Vector Machine

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	0.98	1.00	0.99
Goddess Pose	0.79	0.76	0.77

Plank Pose	**0.96**	**0.96**	**0.96**
Tree Pose	**0.91**	**0.95**	**0.93**
Warrior2 Pose	**0.83**	**0.82**	**0.82**

Table 10: Performance of XGBoost

Yoga Pose	Precision	Recall	F1 Score
Down dog Pose	**1.00**	**0.98**	**0.99**
Goddess Pose	**0.98**	**0.88**	**0.92**
Plank Pose	**0.93**	**1.00**	**0.97**
Tree Pose	**0.95**	**0.98**	**0.97**
Warrior2 Pose	**0.94**	**0.94**	**0.94**

Table 11: Accuracy for all ML models

Model	Random Forest Classifier	Multilayer Perceptron	Support Vector Machine	XGBoost
Accuracy	**0.93**	**0.82**	**0.90**	**0.96**

Phase 2 Results Discussion

In Phase 2, assessing models using both coordinates and angles data, intriguing performance variations emerged.

1. **Random Forest Classifier:** Achieved 92.98% accuracy, maintained high precision for 'down dog' and 'plank,' resulting in robust F1-scores, especially enhanced by angle data.
2. **Multilayer Perceptron (MLP):** Experienced an accuracy drop to 81.75%, struggled with angles, leading to lower recall rates, especially for 'tree' and 'warrior2,' necessitating further fine-tuning.
3. **Support Vector Machine (SVM):** Balanced accuracy at 89.82%, solid precision and recall, notably for 'down dog' and 'plank,' even with angle data.
4. **XGBoost:** Impressed with 95.79% accuracy, strong precision, and robust recall for various poses, effectively utilizing both coordinates and angles data.

Performance variations arose from model characteristics and data nature. Ensemble models excelled due to their strength in handling complex features, managing non-linearity, and resisting overfitting, enabling effective use of angle information. MLP's struggle with angles points to the need for architecture and hyperparameter optimization to leverage this data.

In summary, the Random Forest Classifier adapted well, MLP requires further optimization, SVM demonstrated robustness, and XGBoost excelled. The inclusion of angle data enhances pose classification potential, promising more accurate real-world applications. Further research should refine MLP's configuration and explore broader use cases for these models.

Results from the XBoost Model

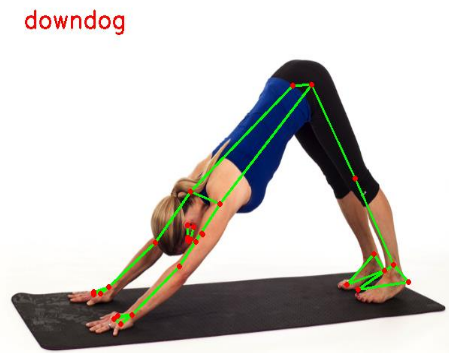
Figure 7: Goddess Pose

Figure 8: Tree Pose

Figure 9: Warrior2 Pose

Figure 10: Plank Pose

Figure 11: Down dog Pose



5. Future Works

There are several methods available to enhance results and improve the efficiency of the models. Below are some of these approaches

5.1 Normalization process using torso size

Normalizing pose data using torso size can be advantageous in certain situations, especially in pose estimation tasks where you want to make the pose data scale-invariant or reduce the impact of variations in the subject's size or distance from the camera. Here are some advantages of normalizing using torso size:

1. **Scale-Invariance:** This approach ensures that pose data becomes scale-invariant, facilitating comparisons across subjects of varying sizes and subject-to-camera distances.
2. **Reduced Sensitivity to Distance**: It mitigates the sensitivity of pose estimation to the subject's distance from the camera, resulting in more robust pose data.
3. **Improved Generalization:** Normalizing based on torso size enhances model generalization, preventing overfitting to a specific subject's size and shape. The focus shifts to relative body part positions rather than absolute distances.
4. **Consistency:** For comparing poses or tracking changes over time, torso size normalization provides more consistent and meaningful results.

5.3 Normalization to create embedding

1. This approach involves more complex normalization and embedding steps.
2. It may provide a more abstract representation of the pose, which could be beneficial for certain machine learning tasks.
3. It might be suitable when you want to feed the data into a neural network or other machine learning models.

5.2 Angle Normalization

In our ongoing research, we aim to enhance angle data consistency for yoga pose classification. We've calculated six angles from pose landmarks via a human pose estimation model. However, these angles can vary for the same yoga pose, potentially impacting classification accuracy.

To address this, we will implement angle normalization techniques (e.g., Min-Max Scaling, Z-Score Normalization, and Circular Statistics). These methods standardize angle ranges across instances of the same yoga pose, ensuring uniformity. Angle normalization will improve model performance, ensuring consistent and reliable yoga pose classification results, a vital contribution to our research's success.

6. Conclusion

In summary, the performance differences among Random Forest Classifier, XGBoost, and Multilayer Perceptron (MLP) in yoga pose classification were notably influenced by the inclusion of 6 angles as features. Random Forest Classifier and XGBoost outperformed MLP in accuracy when these angles were integrated.

Key factors contributing to Random Forest Classifier and XGBoost' s superior performance with added angles include:

1. **Feature Engineering and Interpretability**: Both models excel in handling interpretable features, and the included angles convey meaningful relationships, aiding these models in leveraging the added information effectively.
2. **Non-Linearity and Complexity**: Deep learning models like MLP require more complexity and data to capture non-linear relationships effectively. If data patterns are relatively simple, ensemble methods like Random Forest and XGBoost can excel.

3. **Overfitting and Hyperparameter Tuning**: Deep learning models, such as MLP, are more prone to overfitting, especially with small or unregularized datasets. In contrast, ensemble methods are less susceptible to overfitting and often require less extensive hyperparameter tuning to perform well.
4. **Data Distribution and Complexity**: Model performance is significantly influenced by data distribution and problem complexity. Decision tree-based models (Random Forest) may be better suited for certain data distributions, while gradient boosting (XGBoost) may excel in others.
5. **Data Inconsistency**: High variation in angle ranges for the same yoga pose, such as the angle at the elbow ranging from 50 to 175 for the goddess pose, can introduce challenges in model generalization. Angle normalization techniques should be considered to address this issue effectively.

Figure 12: In the first image the angle at the elbow is around 40 degrees and for the second the angle is over 175 degrees


In optimizing the MLP model's performance, key steps involve increasing complexity, fine-tuning hyperparameters, exploring activation functions, securing sufficient training data, and utilizing techniques like early stopping and regularization to combat overfitting.

These results emphasize the importance of selecting the right model for the specific task and underlie the need for extensive experimentation and evaluation. Model architecture should align with dataset characteristics and the desired balance between interpretability, complexity, and generalization.

7. Acknowledgment

We extend our heartfelt gratitude to Assistant Professor Anitha M for her unwavering support and invaluable guidance throughout the course of this research. Her expertise and encouragement played a pivotal role in exploring various dimensions of this study.

Additionally, we would like to express our gratitude to the professors of the Computer Science Department at Dayanand Sagar College of Engineering. Their generous support and facilitation of access to department computer labs were instrumental in our exploration of various research topics.

8. Authors' Biography

- P. Charith, a student at Dayanand Sagar College of Engineering, specializes in AI/ML, Data science, and computer vision, holding a degree in computer science from the same institution. Their research primarily focuses on AI/ML applications.

- Prajwal Kumar B R, a student at Dayanand Sagar College of Engineering, specializes in AI/ML and computer vision, holding a degree in computer science from the same institution. Their research primarily focuses on AI/ML applications.
- Ms. Anitha M, an Assistant Professor at Dayanand Sagar College of Engineering. Holding an MTech degree, Ms. Anitha boasts over 12 years of experience in the field. Their expertise encompasses Computer Vision, Computer Graphics, and Data Science. Dr. Anitha actively contributes to research in these areas and is a distinguished member of ISTE.

9. References

1. Vivek Anand Thoutam, Anugrah Srivastava, Tapas Badal, Vipul Kumar Mishra, G. R. Sinha, Aditi Sakalle, Harshit Bhardwaj, Manish Raj, "Yoga Pose Estimation and Feedback Generation Using Deep Learning", Computational Intelligence and Neuroscience, Hindawi vol. 2022, Article ID 4311350, 12 pages, 2022
2. Steven Chen, Richard R. Yang, “Pose Trainer: Correcting Exercise Posture using Pose Estimation”, arXiv:2006.11718v1 [cs.CV] 21 Jun 2020
3. Utkarsh Bahukhandi, Dr. Shikha Gupta, “Yoga pose detection and classification using machine learning techniques ”, IRJMETS, Volume:03/Issue:12/December-2021
4. Rohit Srivatsa, Niharika D, P Charith, Prajwal Kumar B R, “Human pose estimation in fitness tracking and guidance”, IRJMETS, Volume:04/Issue:01/January-2022
5. Ashish Ohri, Shashank Agrawal, Garima S. Chaudhary, “On-device Realtime Pose Estimation & Correction ” , International Journal of Advances in Engineering and Management (IJAEM) Volume 3, Issue 7 July 2021, pp: 1691-1696
6. M. Z. Uddin, N. D. Thang, and T. S. Kim, “Human activity recognition via 3-D joint angle features and hidden Markov models,” in Proceedings of the 2010 IEEE International Conference on Image Processing, pp. 713–716, IEEE, Hong Kong, China, 2010 September.
7. Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, Matthias Grundmann, “MediaPipe Hands: On-device Real-time Hand Tracking”, Google Research,arXiv:2006.10214v1 [cs.CV] 18 June 2020.
8. Y. Agrawal, Y. Shah, and A. Sharma, "Implementation of Machine Learning Technique for Identification of Yoga Pose," 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), 2020, pp. 40-43, DOI: 10.1109/CSNT48778.2020.9115758
9. Sen Qiao, Yilin Wang, and Jian Li, “Real-Time Human Gesture Grading Based on OpenPose”, 2017 10th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISPBMEI 2017)
10. Ardra Anilkumar, Athulya K.T., Sarath Sajan, Sreeja K.A., “Pose Estimated Yoga Monitoring System”, 2nd International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS 2021).
11. Chhaihuoy Long, Eunhye Jo, Yunyoung Nam, “Development of a yoga posture coaching system using an interactive display based on transfer learning”, The Journal of Supercomputing Accepted: 6 September 2021 © The Author(s), under exclusive license to Springer Science+Business Media, LLC, part of Springer Nature 2021.

12. Prof. Rupal More, Pallavi Late, Yukta Taral, Nilima Ahire, Bhagyashri, "Live Yoga Pose Classification Using Image Processing and LR Algorithm", 2023 IJCRT | Volume 11, Issue 5 May 2023
13. Khushi Sidana, "Real time yoga pose detection using deep learning: a review". International Journal of Engineering Applied Sciences and Technology, 2022 Vol. 7, Issue 7, ISSN No. 2455-2143, Pages 61-65