Yogivision: Combining Simple and Complex Features for Yoga Pose Classification

Kirthi Shanbhag
University of California, Berkeley
kirthi_shanbhag@berkeley.edu

Skylar Wang
University of California, Berkeley
skylarmwang@berkeley.edu

Omar Zu'bi
University of California, Berkeley
ozubi@berkeley.edu

Abstract

In this project, we develop a custom image classification system using a combination of simple and complex features. Our dataset includes fifteen classes and a total of 3,644 images, collected using the Bing Image Search API and supplemented by an original Kaggle yoga-pose dataset. After applying preprocessing steps such as augmentation and resizing, we extract multiple types of features: Histogram of Oriented Gradients (HOG), contour-based shape descriptors, MediaPipe pose landmarks, and deep embeddings from a pretrained ResNet50 model. We train Support Vector Machines (SVM) and Logistic Regression classifiers on these features, using a validation set for hyperparameter tuning and PCA to reduce dimensionality where needed. Our best-performing model is Logistic Regression using a combination of MediaPipe and ResNet50 features compressed with PCA, which achieves a test accuracy of 96.44% with a training time of 16.201s and an inference time of 0.02 ms per sample. The most efficient model, an SVM trained only on MediaPipe features, reaches 89.12% accuracy while training in just 0.22 seconds and an inference time of 0.15 ms per sample. Our results show that ResNet50 embeddings achieve the highest overall performance, while MediaPipe features are the most computationally efficient. Contour-based features perform poorly when used in isolation but provide substantial benefits when combined with more complex features. Notably, the top four performing models all use Logistic Regression, which exhibits significantly longer training times than SVM models with the same feature sets, but achieves faster inference, with all models operating at under 0.1 ms per sample.

1 Introduction

Yoga pose classification is a challenging image recognition problem because the same pose can look very different across people due to variations in body shape, flexibility, skill level, and alignment, while different poses can look very similar with only subtle joint-angle differences separating them. This challenge is further complicated by variability in image data, such as camera angle, lighting, and image type (e.g., photographs versus drawings), along with inconsistent pose annotations, making it difficult for models to distinguish pose classification from natural human variation. Our dataset contains 3,644 images across 15 yoga poses, providing sufficient diversity for effective training and evaluation. Our goal is to compare simple features with complex features and evaluate how different combinations of them affect classification accuracy and computational efficiency. We implement four feature types: HOG, contour features, MediaPipe landmarks, and ResNet50 embeddings, and train both SVM and Logistic Regression classifiers. We measure accuracy, training time, and inference time to understand the trade-offs between performance and speed.

2 Dataset

To support robust pose classification, we constructed a diverse dataset combining curated samples from the Kaggle Yoga Pose Dataset (Kalluri, 2018) with additional images retrieved using the Bing Image Search API. The final dataset spans 15 yoga pose classes and contains a total of 3,644 images. Because certain poses appear more frequently than others, we apply stratified splitting to

preserve class balance across the training, validation, and test sets. The dataset includes:

- Total images: 3,644
- Number of classes: 15
- Classes: *bow, bridge, child, cow, dancer, downdog, gate, goddess, lotus, plank, plow, trianglevariation, tree, triangle, warrior2*

Figure 1: Sample images

2.1 Data Preprocessing

The images are loaded with OpenCV, converted from BGR to RGB, and resized as needed for selected feature types.

- HOG features extracted from images resized to 256x256
- ResNet features extracted from images resized to 224×224
- Contour and MediaPipe features extracted at the original image resolution

Horizontal flipping is used as a data augmentation technique during training to increase variability and reduce overfitting, effectively doubling the training set size while leaving the validation and test sets unchanged.

3 Feature Extraction

3.1 Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG) is a widely used descriptor that captures gradient orientation patterns within an image. Since yoga poses involve different limb arrangements and body outlines, edge information is a natural starting point. HOG converts each image into a histogram describing the frequency of local gradient directions, allowing the classifier to pick up differences in posture and silhouette. We computed HOG using 12 orientations, 8×8 cells, and 2×2 blocks. Each image produces a 46,128-dimensional feature vector. Figure 2 shows an example HOG visualization, where the overall structure of the pose remains recognizable. This gave us confidence that HOG would help separate classes with distinct body outlines.

Figure 2: Example images of HOG

3.2 Contour Features

Contour features summarize the shape and structural layout of an image by identifying continuous object boundaries. In the context of yoga pose classification, many poses differ primarily in the spatial arrangement of limbs relative to the torso, making shape-based features particularly informative. Contour descriptors capture these geometric differences without relying on color or texture information. Using Canny edge detection followed by OpenCV contour extraction, we identify all external contours in the image and retain only the largest contour, which typically corresponds to the subject's body. In many cases, this process effectively suppresses background information and isolates the pose silhouette; however, in some images, background elements such as text or high-contrast artifacts remain, which can reduce feature quality. From the selected contour, we compute a set of simple geometric measurements:

- Contour area
- Perimeter
- Aspect ratio

- Extent
- Solidity
- Centroid (cx, cy)

This creates a 7-dimensional shape descriptor for each image. Figure 3 demonstrates an example contour map. Although contour-based features are limited when used in isolation, their low dimensionality and computational efficiency make them well-suited for fast models and for complementing more expressive feature representations.

Figure 3: Contour overlays on sample images.

3.3 MediaPipe Pose Landmarks

To introduce a more semantic understanding of body structure, we used Google's MediaPipe pose estimation model to extract 3D landmark coordinates for key body joints. These pose embeddings capture higher-level information such as joint angles, limb positions, and overall posture, making them well suited for pose recognition tasks. Because MediaPipe outputs a standardized skeleton for each image, these features are generally robust to background clutter and lighting variation. Figure 4 shows an example of the detected keypoints and their connection overlaid with the original images. MediaPipe detects 33 body landmarks. For each landmark, we extract (x, y, z, visibility), giving 132 values. We add 8 joint-distance features, producing a 140-dimensional feature vector per image. If detection fails, a zero vector is used. This strategy avoids discarding samples while ensuring consistent input dimensionality across the dataset. In our experiments, MediaPipe features offered very fast inference and reasonable accuracy, making them a strong choice for efficiency-focused models. However, we did encounter a limitation: a small fraction of images (approximately 14%) failed to produce valid keypoint detections, which reduced consistency across the dataset.

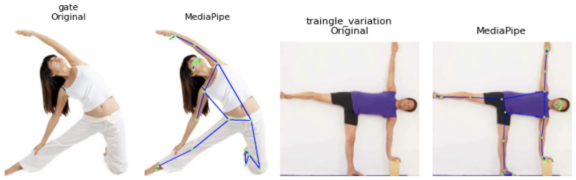

Figure 4: MediaPipe pose landmark overlays.

3.4 ResNet50 Feature Embeddings

To incorporate deeper semantic information, we extract feature embeddings from a pretrained ResNet50 convolutional neural network. Each image is resized to 224×224 and preprocessed using the standard ImageNet normalization before being passed through the network. We use a ResNet50 model with ImageNet weights and remove the final classification layer, extracting features from the global average pooling layer of the network. The resulting embeddings are 2,048-dimensional vectors that encode rich visual information related to texture, shape, and global structure. Although these features are not explicitly pose-specific, they capture complex patterns learned from large-scale image datasets. In our experiments, ResNet50 embeddings were the most discriminative features in the entire pipeline, substantially improving classification accuracy both when used alone and when combined with other features.

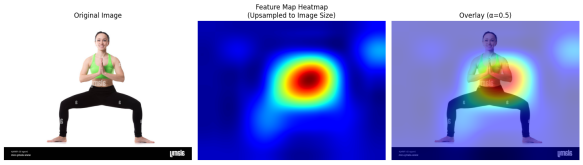

Figure 5: Example ResNet feature activation.

3.5 PCA and T-SNE Analysis

We used PCA and t-SNE to explore the structure of the data set for each feature. The purpose was to understand what kind of information each feature captures, how well the classes separate visually, and whether any features appear more informative before training the classifiers.

3.5.1 Principal Component Analysis (PCA)

To better understand the structure and separability of the different feature representations used in the yoga-pose classification task, Principal Component Analysis (PCA) was applied to each feature set. The first two principal components

were visualized for the HOG, Contour, MediaPipe, and ResNet. Although PCA reduces the high-dimensionality, the visualizations still provide useful insights into how well each feature type captures class-discriminative patterns in Figure 6. We saw that there was substantial overlap between many of the classes in the PCA projections; however, the MediaPipe feature spaces showed some visible clustering patterns, even though these clusters were still partially overlapped.

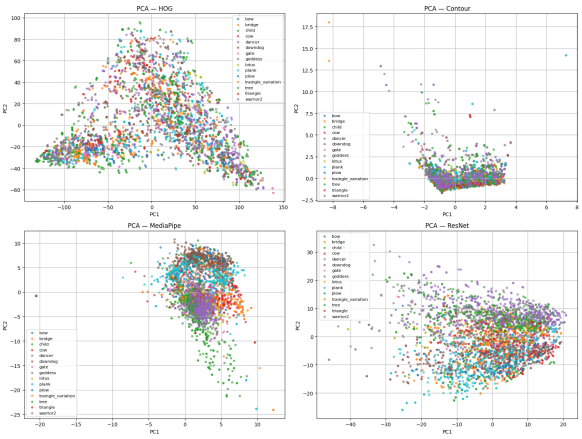

Figure 6: Visualization of the dataset on the first two principal components, revealing class structure and relative spacing between groups.

To further investigate the behavior and discriminative capacity of each feature representation, we analyzed their variance structure and class-wise distribution patterns using Principal Component Analysis (PCA). Figure 7 illustrates how much cumulative variance is captured as the number of principal components increases, providing insight into the intrinsic dimensionality of each feature type. Complementing this, Figure 8 compares feature distributions across classes, revealing how different representations capture inter-class variability. Together, these visualizations help characterize the strengths and limitations of HOG, Contour, MediaPipe, and ResNet before they are used for downstream classification.

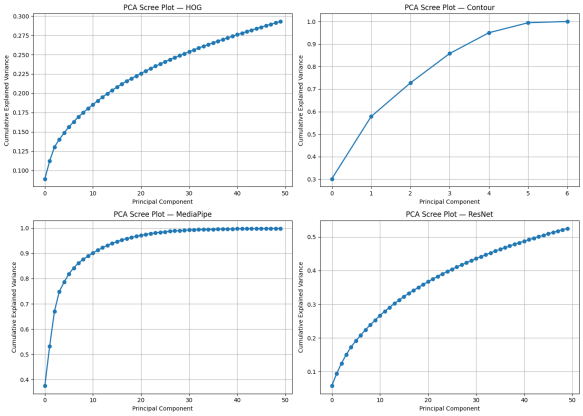

Figure 7: Explained variance as a function of the number of principal components.

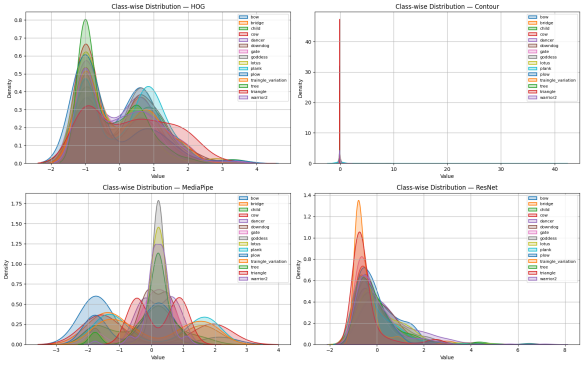

Figure 8: Class-wise comparison of feature distributions across selected dimensions, highlighting variability and inter-class differences.

From Figure 7, HOG features show a gradual, nearly linear increase in cumulative variance across 50+ components, indicating that information is distributed across many dimensions and capturing subtle variations in edges and poses. Contour features are highly compact, with just six components explaining almost all variance, reflecting strong correlations and a low-dimensional structure. MediaPipe features capture most variance rapidly, with about 90% explained by the first ten components, characteristic of skeletal-based representations where key pose variations are concentrated in a few dimensions. ResNet features accumulate variance more gradually, requiring around 50 components to reach 52%, suggesting diverse, distributed representations across the feature space.

From Figure 8, HOG exhibits a bimodal distribution with two peaks, suggesting partial clustering but significant class overlap. Contour features

are sharply peaked near zero with long tails, indicating sparse, similar contours across poses. MediaPipe displays multiple distinct peaks, reflecting clear separability and identification of pose archetypes. ResNet shows a smooth, unimodal distribution, creating a continuous representation space with moderate class distinction. Overall, MediaPipe appears to offer strong standalone classification potential due to its clear separation, while contour features provide limited discriminative power and may be better suited as complementary cues rather than a primary representation.

3.5.2 T-distributed Stochastic Neighbor Embedding (t-SNE) Visualization

To evaluate how well different feature extraction methods capture the structure of our yoga-pose dataset, we performed t-Distributed Stochastic Neighbor Embedding (t-SNE) on four individual feature sets—HOG, Contour, MediaPipe Pose Landmarks, and ResNet. Figure 9 shows a 2D t-SNE representation of the images, colored by their pose class.

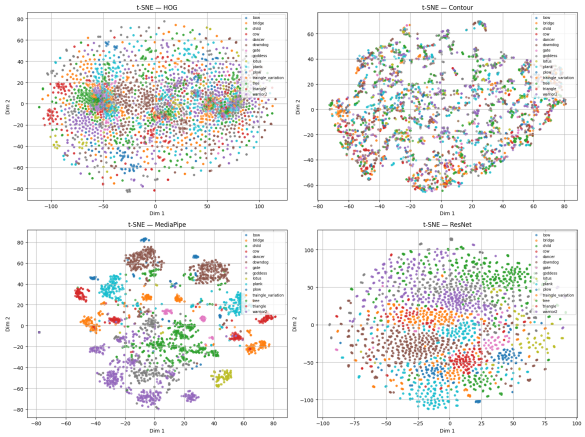

Figure 9: Two-dimensional t-SNE projection of the dataset, illustrating the overall structure of the learned feature space.

HOG features produce a diffused, highly intermixed distribution with no meaningful clustering, which is expected given their sensitivity to background gradients and limited ability to encode full-body geometry. Contour-based features show only a modest improvement, forming loose clouds that still overlap heavily across classes; while contours capture global outline information, many poses share similar silhouettes, and contour quality varies significantly with background noise. In contrast, the MediaPipe keypoint features form more coherent and interpretable patterns. Several poses cluster naturally because the skeletal joint coordinates directly encode human body configuration while ignoring irrelevant image details. This indicates that geometric pose structure is a much stronger signal for this task than raw pixel-based edges or boundaries. ResNet deep features also show clustering, producing somewhat tighter groups and smoother manifolds. However, because ResNet is trained on ImageNet and not specifically on human poses, the embeddings still mix classes based on clothing, background, or texture rather than pose alone.

3.6 Feature Sets

To understand how different types of information work together, we tested every feature combination derived from the four feature types used in this project. This allowed us to compare the performance of single-feature models against multi-feature models and to determine whether certain features complement each other when combined. We evaluated the following feature sets:

Single-feature models:

- HOG
- Contour
- MediaPipe
- ResNet

Two-feature combinations:

- HOG + Contour
- HOG + MediaPipe
- HOG + ResNet
- Contour + MediaPipe
- Contour + ResNet
- MediaPipe + ResNet

Three-feature combinations:

- HOG + Contour + MediaPipe
- HOG + Contour + ResNet
- HOG + MediaPipe + ResNet
- Contour + MediaPipe + ResNet

Four-feature combination:

- All features combined (HOG + Contour + MediaPipe + ResNet)

The largest combination contains more than 48,000 dimensions before PCA. Because of this, PCA was applied to the higher-dimensional combinations to reduce computation time and prevent overfitting.

4 Classification & Results

In this section, we present the performance of the Logistic Regression and SVM classifiers across all different combinations of our simple and complex features: (1) Histogram of Oriented Gradients (HOG), (2) contour-based features, (3) MediaPipe keypoints, and (4) ResNet embeddings. Following standard practice, we divided the dataset into training, validation, and test sets using a two-stage stratified split to preserve the class distribution. First, we allocated 70% of the data to a temporary training+validation pool and 30% to a held-out test set. From the 70%, we then assigned 80% to training (56% of the total data) and 20% to validation (14% of total). This resulted in a final distribution of:

- Training set: 56%
- Validation set: 14%
- Test set: 30%

4.1 Model Performance

Table 1 summarizes validation accuracy, test accuracy, training time, and inference time for all combinations of features and classifiers.

Table 1: Summary of the performance metrics for all evaluated models.

Classifier	Features	Val Acc	Test Acc	Train Time (s)	Infer Time (ms/sample)	Rank
LogReg_PCA	MediaPipe + ResNet	0.97	0.96	16.20	0.02	1
LogReg_PCA	Contour + ResNet	0.96	0.96	23.39	0.03	2
LogReg_PCA	Contour + MediaPipe + ResNet	0.96	0.96	10.47	0.02	2
LogReg_PCA	ResNet	0.96	0.96	23.57	0.08	3
SVM_RBF_PCA	MediaPipe + ResNet	0.96	0.96	1.01	0.40	4
SVM_RBF_PCA	ResNet	0.96	0.95	0.66	0.35	5

Across all feature types, ResNet embeddings provided the strongest performance, showing up in all of the top 12 performing models. The best performing model was Logistic Regression with ResNet and MediaPipe features, achieving a 96.44% test accuracy. The strong performance of Logistic Regression with PCA on MediaPipe + ResNet features suggests that the pose space is largely linearly separable when high-level semantic cues (ResNet) are complemented by explicit skeletal structure (MediaPipe). PCA further reduces redundancy and noise, improving generalization and computational efficiency. The most efficient model in terms of computational cost was SVM with only MediaPipe features. While its test accuracy was lower (0.8912), it trained in just 0.225 seconds and required only 0.1537 ms per sample for inference, making it ideal for scenarios where speed and resource constraints are critical. HOG features provided reasonable performance, whereas contour-only features consistently performed poorly when used in isolation, confirming their limited suitability for fine-grained pose recognition. However, when paired with ResNet features, contour descriptors proved highly effective, with two of the top three performing models leveraging a Contour–ResNet combination. Combining all features does not guarantee improved performance, as several combined-feature models rank lower than simpler ResNet-based configurations, indicating diminishing returns. The degradation observed when using "all features" indicates possible feature interference or overfitting, where heterogeneous descriptors introduce conflicting signals rather than complementary information.

4.2 Classification Report

To evaluate the per-class performance of the models, we computed precision, recall, F1 score, and support for each yoga pose class using the test set. The evaluations focused on the highest accuracy model (LogReg with MediaPipe + ResNet features) and the most efficient model (SVM with MediaPipe features), as identified in Table 1. Per-class metrics for the highest accuracy model are summarized in Table 2, which highlights the precision, recall, and F1 score for each pose class.

Table 2: Per-class metrics for the highest accuracy model (LogReg - MediaPipe + ResNet).

Class	Precision	Recall	F1 Score	Support
bow	1.00	1.00	1.00	26
bridge	0.97	1.00	0.99	34
child	0.97	0.97	0.97	37
cow	0.98	0.95	0.96	56
dancer	0.93	1.00	0.96	38
downdog	0.98	0.96	0.97	177
gate	1.00	1.00	1.00	22
goddess	0.97	0.96	0.97	118
lotus	0.95	1.00	0.97	35
plank	0.95	0.97	0.96	126
plow	0.97	0.97	0.97	31
triangle_variation	0.95	1.00	0.97	77
tree	0.98	0.99	0.98	150
triangle	0.93	0.84	0.88	31
warrior2	0.96	0.95	0.96	136

The highest-accuracy model (LogReg with MediaPipe + ResNet) achieves consistently strong performance across almost all classes, with precision, recall, and F1-scores mostly in the 0.96–1.00 range. Several poses, such as bow, gate, and bridge are classified nearly perfectly, indicating that their visual or pose features are highly dis-

tinctive. Even for classes with larger support like downdog, tree, plank, and warrior2, the model maintains balanced precision and recall, suggesting robustness to class frequency. The main weakness appears in the triangle class, where recall drops to 0.84 and F1 to 0.88, implying confusion with visually similar poses (e.g., triangle-variation). Overall, this model demonstrates strong generalization and minimal class imbalance effects.

The detailed performance of the highest accuracy model can be further visualized through its confusion matrix and ROC curves, which provide insights into how well the model discriminates between different pose classes.

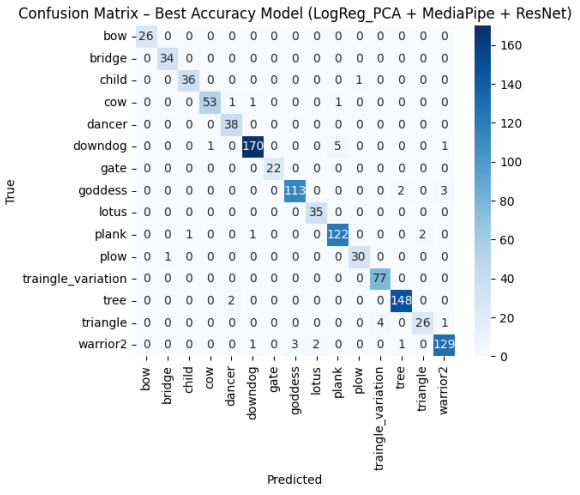

Figure 10: Confusion matrix for the highest accuracy model (LogReg - MediaPipe + ResNet).

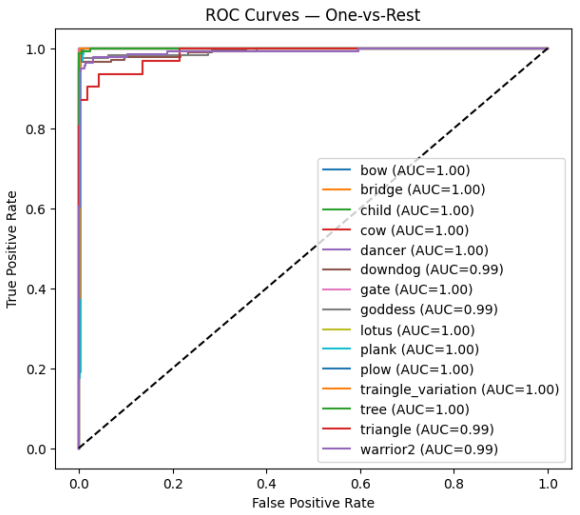

Figure 11: ROC curves for the highest accuracy model (LogReg - MediaPipe + ResNet).

For the highest-accuracy model, the one-vs-rest ROC curves are almost perfectly aligned with the top-left corner, with AUC values at or extremely close to 1.00 for nearly all classes. This indicates excellent separability between each pose and the rest, meaning the model can reliably distinguish even visually similar classes. The corresponding confusion matrix shows a strong diagonal dominance, with very few off-diagonal entries. Misclassifications are rare and mostly occur between closely related poses, such as triangle and triangle-variation, or occasionally between downdog and structurally similar standing poses. Importantly, high-support classes (e.g., downdog, tree, warrior2) maintain near-perfect classification, confirming the model's robustness and balanced performance across the dataset.

Per-class metrics for the most efficient model are shown in Table 3. This table highlights the trade-offs of using a smaller feature set, showing slightly lower recall for some classes while maintaining high precision.

Table 3: Per-class metrics for the most efficient model (SVM_RBF_PCA + MediaPipe).

Class	Precision	Recall	F1 Score	Support
bow	1.00	0.54	0.70	26
bridge	1.00	0.85	0.92	34
child	0.88	0.62	0.73	37
cow	1.00	0.79	0.88	56
dancer	0.90	0.92	0.91	38
downdog	0.63	0.97	0.77	177
gate	0.94	0.68	0.79	22
goddess	0.99	0.86	0.92	118
lotus	1.00	0.83	0.91	35
plank	0.93	0.89	0.91	126
plow	0.94	0.52	0.67	31
traingle_variation	0.91	0.91	0.91	77
tree	0.95	0.92	0.94	150
triangle	0.87	0.84	0.85	31
warrior2	0.95	0.93	0.94	136

The most efficient model (SVM RBF with PCA + MediaPipe) shows noticeably lower and more variable performance, particularly in recall. While precision remains relatively high for many classes (often above 0.90), recall drops substantially for poses such as bow (0.54), plow (0.52), gate (0.68), and child (0.62), resulting in reduced F1-scores. This indicates that the model is more conservative, correctly labeling predictions when confident but failing to capture many true instances. Performance is comparatively better for high-support or structurally distinctive poses like tree, triangle-variation, and warrior2, where F1-scores approach those of the high-accuracy model.

The confusion matrix and ROC curves for the most efficient model (Figures 12 and 13) highlight class-wise performance and the trade-offs from just using the MediaPipe feature.

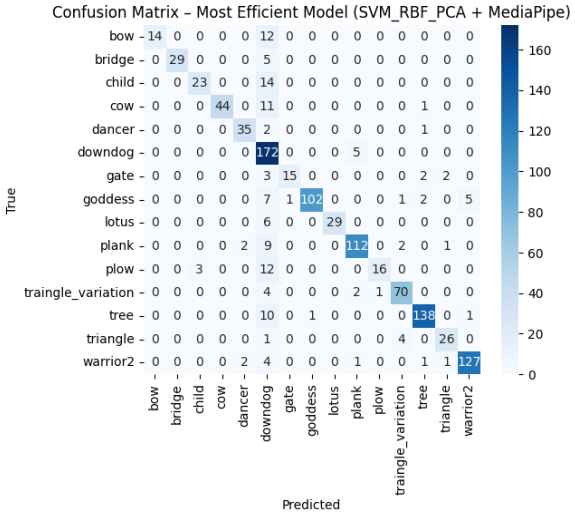

Figure 12: Confusion matrix for the most efficient model (SVM - MediaPipe).

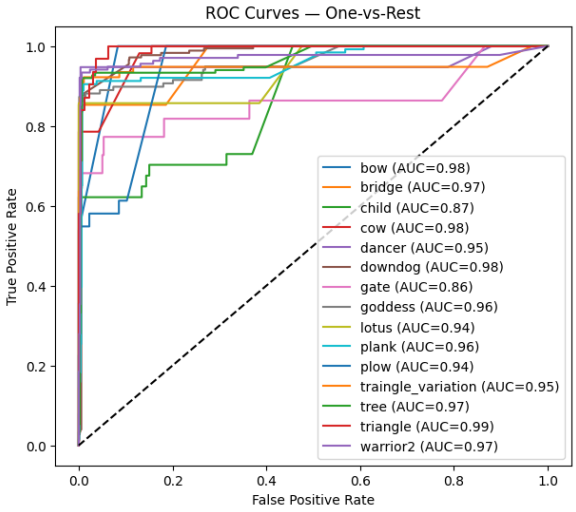

Figure 13: ROC curves for the most efficient model (SVM - MediaPipe).

The most efficient model exhibits noticeably weaker ROC behavior for several classes, with AUC values dropping into the mid-to-high 0.80s for poses like child, gate, and tree. While many classes still achieve strong AUCs above 0.95, the flatter ROC curves for weaker classes indicate reduced discriminative power and greater overlap in feature space. This is reflected clearly in the confusion matrix, which shows increased off-diagonal entries and systematic confusion across multiple classes. Poses such as bow, child, plow, and gate are frequently misclassified, often as more dominant or visually similar poses, leading to the lower recall observed earlier. Overall, the ROC and confusion matrix analyses confirm that the efficient model sacrifices class-level separability and consistency for computational efficiency, whereas the accuracy-focused model achieves near-ideal discrimination and minimal confusion across all yoga poses.

Overall, these classification reports, along with the confusion matrices and ROC curves, confirm that the combination of ResNet embeddings and MediaPipe landmarks provides a robust representation for fine-grained pose recognition, achieving high accuracy and balanced performance across the majority of classes, while MediaPipe alone enables faster inference at the cost of slightly reduced per-class performance.

4.3 Efficiency

To evaluate the practical usability of each feature–classifier combination, we compared both training time and inference time per image, since real-world pose classification systems often operate under resource and latency constraints.

4.3.1 Training Efficiency

Models built on low-dimensional features, particularly contour and MediaPipe, trained very quickly across both classifiers. Among all evaluated configurations, SVM with MediaPipe features was the fastest to train, completing training in approximately 0.22 seconds while achieving a test accuracy of 0.89. SVM-based models generally exhibited shorter training times than their Logistic Regression counterparts, especially when applied to high-dimensional feature sets. In contrast, Logistic Regression models using ResNet or combined feature representations required substantially longer training times, with some configurations exceeding 45 seconds, reflecting the higher computational cost associated with optimizing large feature spaces even after PCA.

4.3.2 Inference Efficiency

Inference efficiency shows a clear distinction between classifiers and feature configurations. Logistic Regression models consistently achieved extremely low inference latency, typically in the range of 0.00–0.08 ms per sample when using

compact or CNN-based feature representations such as MediaPipe and ResNet, making them well-suited for real-time or latency-critical deployment. However, inference time increased noticeably for Logistic Regression when high-dimensional handcrafted features were included, particularly HOG-based combinations, where latency exceeded 1 ms per sample in the worst case. In contrast, SVM models exhibited substantially higher inference costs overall. Per-sample inference times ranged from approximately 0.13 ms for low-dimensional feature sets (e.g., MediaPipe or Contour + MediaPipe) to nearly 1 ms for HOG-based and multi-feature configurations. Although SVM models generally require shorter training times, their inference latency scaled poorly with feature dimensionality due to the computational overhead of kernel evaluations and the dependence on multiple support vectors at test time. Overall, these results highlight a clear trade-off between training efficiency and inference speed, with Logistic Regression offering superior inference efficiency and SVMs favoring faster model training.

4.3.3 Accuracy vs. Efficiency Trade-off

Accuracy and efficiency exhibit a clear trade-off across feature representations and classifiers. ResNet-based features consistently delivered the strongest classification performance, particularly when combined with MediaPipe or contour features, indicating that deep visual embeddings complement pose-based representations effectively. The best-performing configuration—Logistic Regression with MediaPipe and ResNet features—achieved a test accuracy of 96.44% while maintaining extremely low inference latency (approximately 0.02 ms per sample), albeit with a comparatively longer training time. In contrast, MediaPipe-only models provided the most favorable efficiency profile, offering very fast training and inference with only a modest reduction in accuracy. This makes them especially suitable for lightweight or resource-constrained deployment scenarios. Overall, the results underscore a fundamental trade-off: maximizing accuracy benefits from richer, high-dimensional feature representations, whereas real-time and embedded applications are better served by compact, pose-based features such as MediaPipe.

4.4 Generalizability

A major goal throughout our project was ensuring the models could generalize well, that is, perform reliably on new, unseen images rather than just memorizing patterns from the training set. To support this, we utilized dimensionality reduction and hyperparameter tuning. Principal Component Analysis (PCA) was applied selectively depending on the feature dimensionality, reducing redundancy and noise while retaining the most informative components. For classification, we tuned key parameters such as the regularization strength C, kernel scale γ for SVMs, and the number of PCA components, using a grid search over candidate values. Table 4 reports the best-performing hyperparameter configuration for each classifier–feature combination, ranked by test accuracy, along with the corresponding validation and test accuracies and inference times.

Table 4: Best Hyperparameter Configuration per Classifier, Ranked by Test Accuracy

Classifier	Features	PCA	C	Gamma	Val Acc	Test Acc	Infer (ms)
LogReg_PCA	MediaPipe + ResNet	100	2.0	–	0.9667	0.9644	0.02
LogReg_PCA	Contour + ResNet	100	0.5	–	0.9647	0.9580	0.03
LogReg_PCA	Contour + MediaPipe + ResNet	100	0.5	–	0.9647	0.9580	0.02
LogReg_PCA	ResNet	100	1.0	–	0.9569	0.9561	0.08
SVM_RBF_PCA	MediaPipe + ResNet	100	1	scale	0.9647	0.9552	0.40
SVM_RBF_PCA	ResNet	100	1	scale	0.9608	0.9525	0.35
SVM_RBF_PCA	Contour + ResNet	100	1	scale	0.9667	0.9516	0.45
SVM_RBF_PCA	Contour + MediaPipe + ResNet	50	5	scale	0.9647	0.9461	0.30
SVM_RBF_PCA	All Features	100	5	scale	0.9255	0.9132	0.74
SVM_RBF_PCA	HOG + MediaPipe + ResNet	100	5	scale	0.9255	0.9122	0.94
SVM_RBF_PCA	HOG + Contour + ResNet	100	5	scale	0.9255	0.9086	0.59
SVM_RBF_PCA	HOG + ResNet	100	5	scale	0.9255	0.9086	0.81
SVM_RBF_PCA	HOG + Contour	50	5	scale	0.9000	0.8940	0.73
SVM_RBF_PCA	HOG + MediaPipe	50	5	scale	0.9039	0.8940	0.77
SVM_RBF_PCA	MediaPipe	30	5	0.01	0.8941	0.8912	0.15
LogReg_PCA	MediaPipe	30	2.0	–	0.8451	0.8464	0.00
LogReg_PCA	Contour + MediaPipe	30	0.5	–	0.8353	0.8492	0.01
LogReg_PCA	HOG + MediaPipe + ResNet	100	2.0	–	0.8922	0.8693	0.41
LogReg_PCA	All Features	100	2.0	–	0.8902	0.8693	0.36
LogReg_PCA	HOG + Contour + ResNet	100	0.5	–	0.8824	0.8528	0.21
LogReg_PCA	HOG + ResNet	100	0.5	–	0.8804	0.8537	0.34
LogReg_PCA	HOG + MediaPipe	100	0.5	–	0.7980	0.7861	0.49
LogReg_PCA	HOG + Contour	100	0.5	–	0.7824	0.7761	0.28
LogReg_PCA	HOG	100	0.5	–	0.7824	0.7751	1.46
SVM_RBF_PCA	Contour	–	5	scale	0.3333	0.3218	0.27
LogReg_PCA	Contour	–	0.5	–	0.2137	0.2358	0.00

Models that incorporate ResNet-based features dominate the top of the ranking, regardless of classifier type, indicating that deep visual embeddings provide feature representations that transfer well from training to unseen test data. In particular, Logistic Regression and SVM models using MediaPipe + ResNet or Contour + ResNet achieve the highest test accuracies, with minimal degradation from validation accuracy. This small validation–test gap suggests strong generalization and limited overfitting, despite the high dimensionality of the underlying features. Logistic Regression models, especially when paired with ResNet features, show notably strong generalizability. Their consistently high test accuracies and close alignment with validation perfor-

mance indicate that linear decision boundaries are sufficient when the feature space is well structured. This implies that the discriminative power is largely captured by the feature extractor rather than the classifier itself, and that regularization via PCA and the C parameter is effective in controlling overfitting. Moreover, the stability of Logistic Regression performance across multiple ResNet-based feature combinations further supports its robustness to feature variation. In contrast, SVM models exhibit slightly larger drops between validation and test accuracy for more complex, multi-feature combinations (e.g., HOG + MediaPipe + ResNet or all features). This suggests that while SVMs can fit the training distribution well, their nonlinear decision boundaries may become sensitive to redundant or noisy features, reducing generalization when feature dimensionality increases. The strongest SVM generalization is observed when ResNet features are used alone or combined with a small number of complementary features, reinforcing the importance of feature compactness. At the lower end of the table, models relying on simple features alone (HOG or contour) generalize poorly, with substantial gaps between validation and test accuracy and overall low performance. This indicates limited representational capacity and poor transferability to unseen samples, likely due to sensitivity to pose variation, viewpoint changes, and intra-class diversity. Even when combined with stronger features, these handcrafted descriptors rarely improve generalization and sometimes degrade it, highlighting the risk of feature redundancy.

5 Conclusion

This project evaluated different feature representations and classifiers for yoga pose classification, focusing on accuracy, efficiency, and generalization. We compared simple features (HOG and contour), pose-based features (MediaPipe), and deep features (ResNet50) using SVM and Logistic Regression models. The results show that feature choice has a larger impact on performance than the choice of classifier.

The best overall model was Logistic Regression with MediaPipe and ResNet features, which achieved a 96.44% test accuracy with very low inference time (about 0.02 ms per sample). This confirms that ResNet embeddings are the most discriminative features, especially when combined with pose information from MediaPipe. These features generalize well to unseen data and allow simple classifiers to perform effectively. SVM with MediaPipe provided the best balance between accuracy and efficiency. It trained in 0.22s, had near-zero inference latency, and still achieved strong accuracy, making it well-suited for real-time or resource-limited applications. Logistic Regression models consistently had faster inference times than SVM models, often below 0.1 ms per sample, but required longer training times. SVM models generally trained faster, especially on high-dimensional features, but had higher inference costs due to kernel evaluations. Combining all features did not always improve performance. In several cases, simpler models using only ResNet or ResNet combined with MediaPipe performed better than models using all feature types, indicating diminishing returns from feature fusion.

Overall, this work shows that well-designed classical machine learning pipelines, when paired with strong feature extractors, can achieve high accuracy and efficient inference without training end-to-end deep networks. High-accuracy systems benefit most from deep features, while real-time systems benefit from compact pose-based features such as MediaPipe.

Table 5: Summary of the performance metrics for all evaluated models.

Classifier	Features	Val Acc	Test Acc	Train Time (s)	Infer Time (ms/sample)	Rank
LogReg_PCA	MediaPipe + ResNet	0.97	0.96	16.20	0.02	1
LogReg_PCA	Contour + ResNet	0.96	0.96	23.39	0.03	2
LogReg_PCA	Contour + MediaPipe + ResNet	0.96	0.96	10.47	0.02	2
LogReg_PCA	ResNet	0.96	0.96	23.57	0.08	3
SVM_RBF_PCA	MediaPipe + ResNet	0.96	0.96	1.01	0.40	4
SVM_RBF_PCA	ResNet	0.96	0.95	0.66	0.35	5
SVM_RBF_PCA	Contour + ResNet	0.97	0.95	0.69	0.45	6
SVM_RBF_PCA	Contour + MediaPipe + ResNet	0.96	0.95	0.50	0.30	7
SVM_RBF_PCA	All Features	0.93	0.91	8.22	0.74	8
SVM_RBF_PCA	HOG + MediaPipe + ResNet	0.93	0.91	10.24	0.94	9
SVM_RBF_PCA	HOG + ResNet	0.93	0.91	6.23	0.81	10
SVM_RBF_PCA	HOG + Contour + ResNet	0.93	0.91	4.97	0.59	10
SVM_RBF_PCA	HOG	0.90	0.89	6.63	0.71	11
SVM_RBF_PCA	Contour + MediaPipe	0.90	0.89	0.30	0.13	11
SVM_RBF_PCA	HOG + MediaPipe	0.90	0.89	8.65	0.77	12
SVM_RBF_PCA	HOG + Contour + MediaPipe	0.90	0.89	7.97	0.95	12
SVM_RBF_PCA	HOG + Contour	0.90	0.89	5.13	0.73	12
SVM_RBF_PCA	MediaPipe	0.89	0.89	0.22	0.15	13
LogReg_PCA	HOG + MediaPipe + ResNet	0.89	0.87	45.61	0.41	14
LogReg_PCA	All Features	0.89	0.87	46.80	0.36	14
LogReg_PCA	HOG + ResNet	0.88	0.85	34.24	0.34	15
LogReg_PCA	HOG + Contour + ResNet	0.88	0.85	30.07	0.21	16
LogReg_PCA	Contour + MediaPipe	0.84	0.85	3.00	0.01	17
LogReg_PCA	MediaPipe	0.85	0.85	4.97	0.00	18
LogReg_PCA	HOG + Contour + MediaPipe	0.80	0.79	16.13	0.42	19
LogReg_PCA	HOG + MediaPipe	0.80	0.79	16.10	0.49	20
LogReg_PCA	HOG + Contour	0.78	0.78	23.66	0.28	21
LogReg_PCA	HOG	0.78	0.78	24.31	1.46	22
SVM_RBF_PCA	Contour	0.33	0.32	0.39	0.27	23
LogReg_PCA	Contour	0.21	0.24	0.84	0.00	24

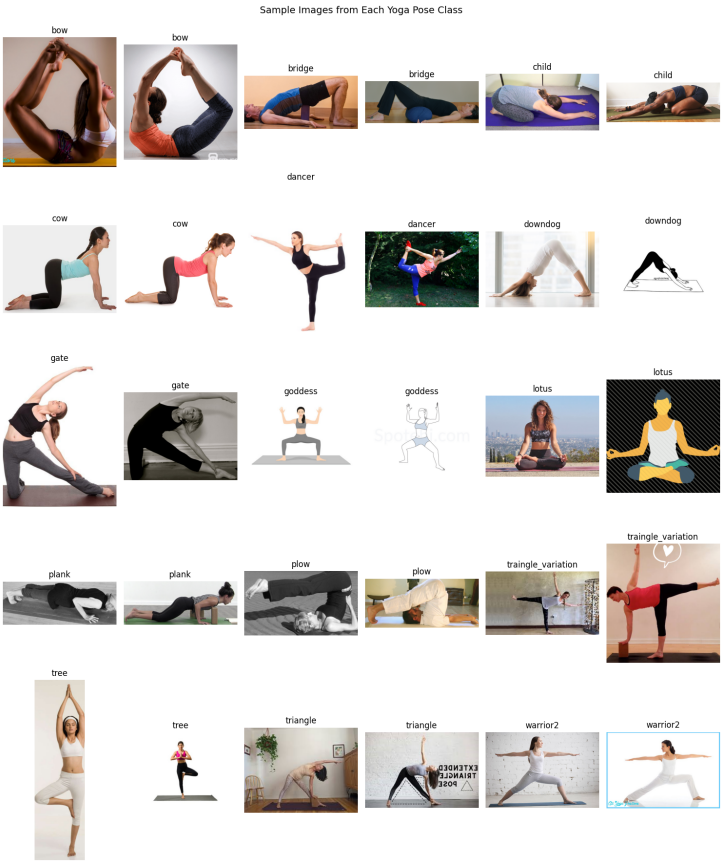

Figure 14: Sample images of all classes.

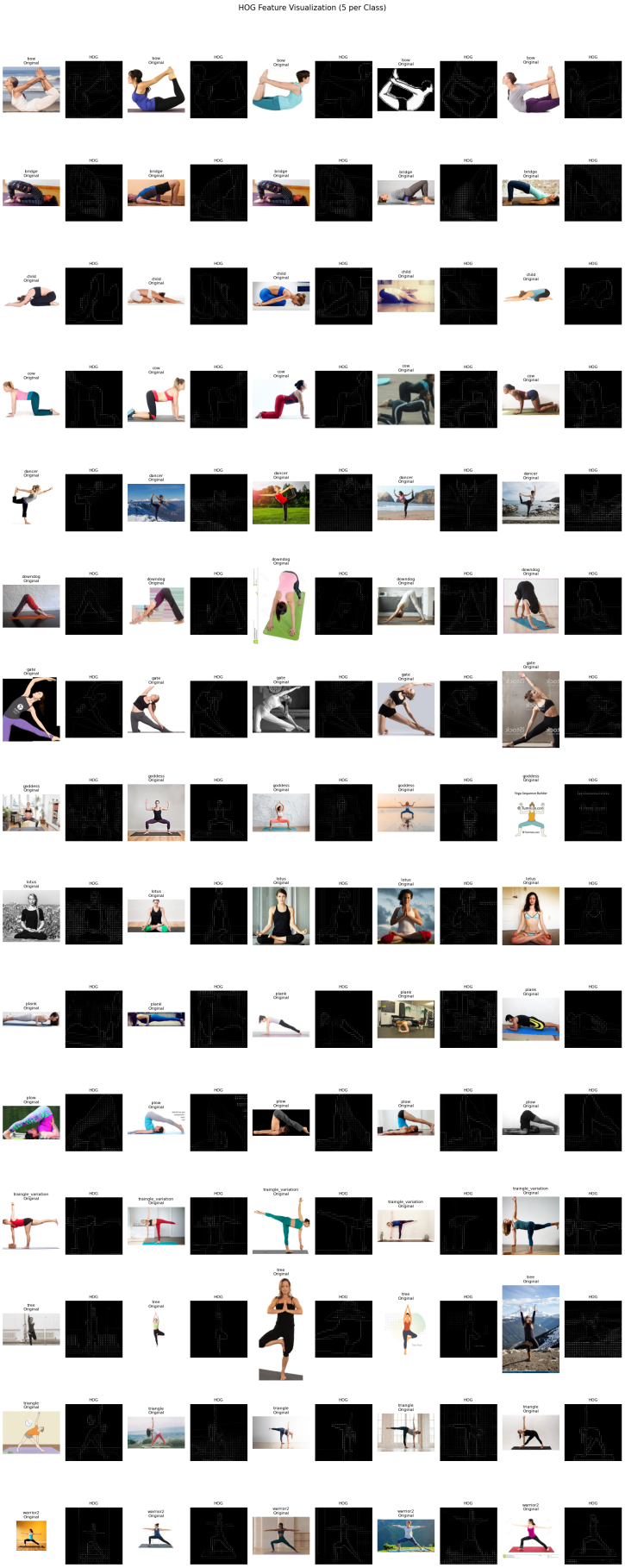

Figure 15: Sample images and feature representation using HOG features.

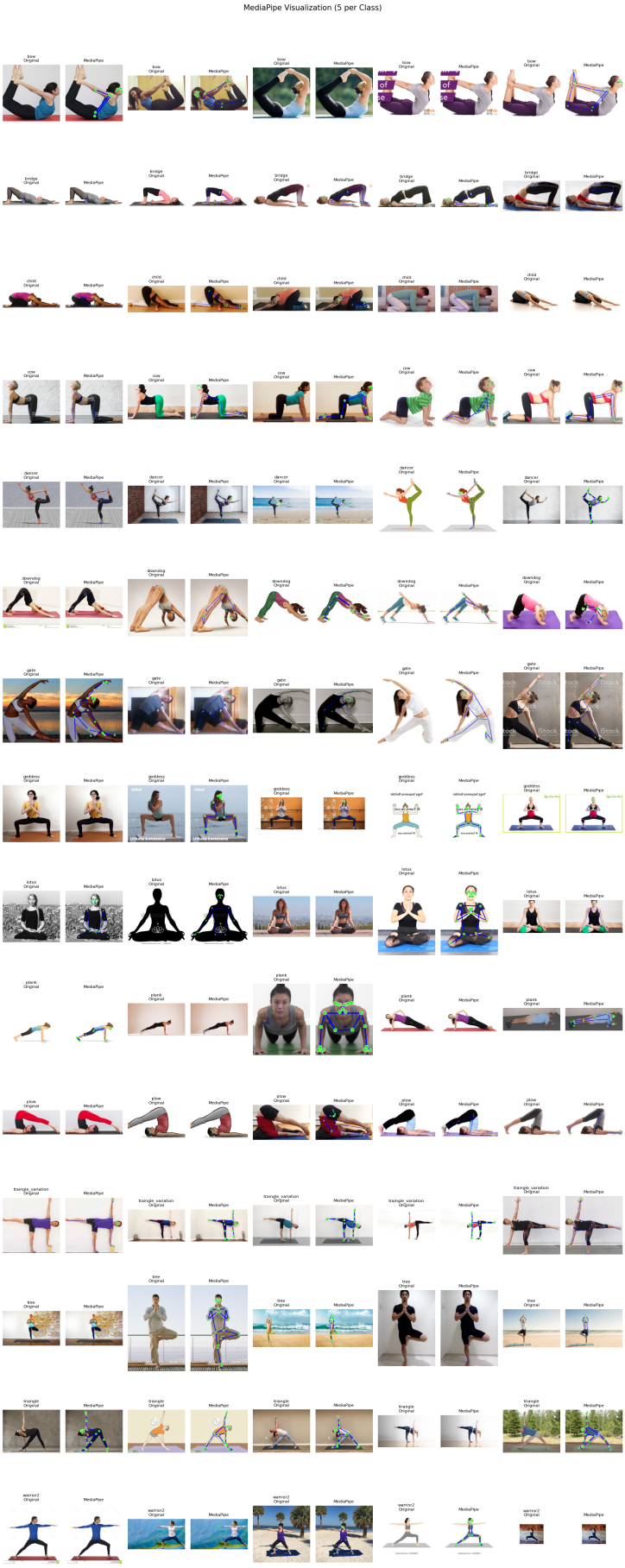

Figure 18: Sample images and feature representation using MediaPipe landmarks.

Figure 16: Sample images and feature representation using Contour features.

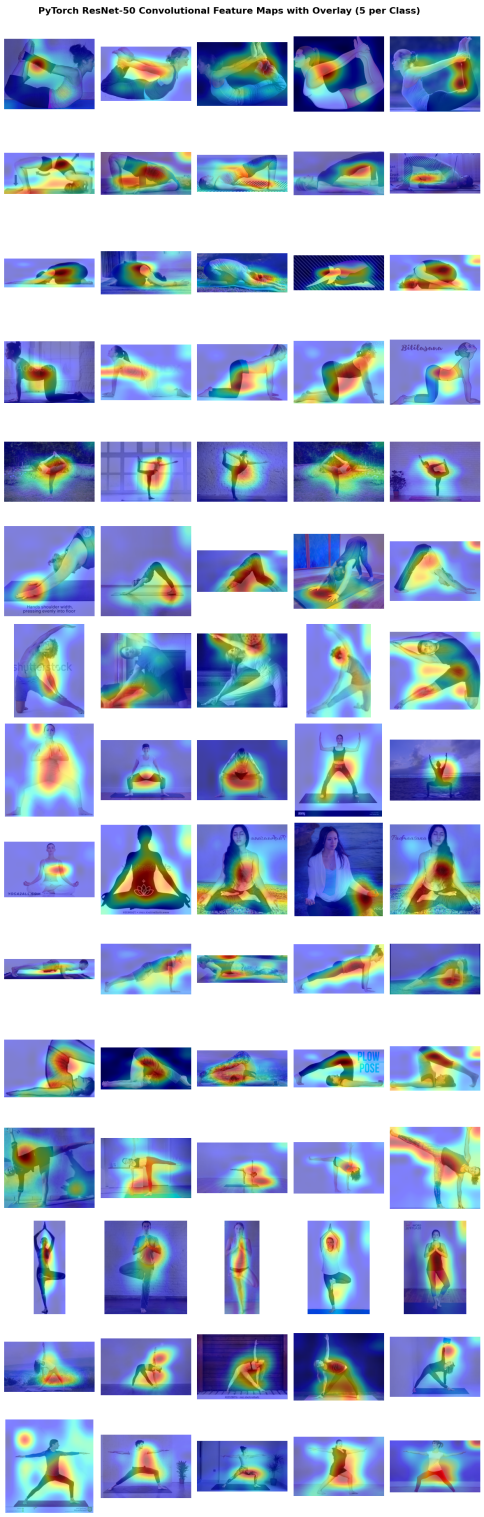

Figure 17: Sample images and feature representation using ResNet embeddings.

Table 6: Team Member Contributions

Name	**Contributions**
Omar	ML pipeline flow, LaTeX reporting, simple/complex feature extraction, and report write-up
Skylar	Classification and training, hyperparameter tuning, and efficiency vs. accuracy reporting , and report write-up
Kirthi	Training data cleaning, augmentation, simple and complex feature extraction, and report write-up