# Statistical Learning vs. Deep Learning for Personal Training: A Comparative Analysis

Anthony Huang, Mohammad Althaf Syed
*Department of Electrical and Computer Engineering*
*Stevens Institute of Technology*
Hoboken, NJ, USA
{ahuang12, msyed8}@stevens.edu

**Problem Statement:** When should we apply statistic learning and when to apply deep learning?

**Hypothesis:** *Statistical Learning methods benefit more from structured, well-balanced datasets, achieving competitive accuracy and interpretability, while Deep Learning models can generalize effectively on unstructured, high-dimensional inputs given sufficient data and compute.*

***Abstract*—This project investigates the effectiveness of statistical learning versus deep learning methods for personal training applications. We use structured and unstructured datasets to implement and compare models focused on recommendation and classification tasks. Key metrics include model accuracy, interpretability, and computational efficiency. Based on preliminary results and literature, we provide guidelines for selecting the appropriate model class depending on the data and task at hand.**

***Index Terms*—Statistical learning, deep learning, fitness, recommendation systems, CNN, decision trees, hybrid models**

## I. Introduction

As interest in fitness and personal training continues to grow, machine learning offers promising tools to enhance workout design, personalize exercise recommendations, and improve overall user outcomes. With the rapid rise of wearable devices, fitness tracking apps, and online exercise platforms, the fitness industry now produces large amounts of structured and unstructured data, making it a prime area for applying modern machine learning techniques.

In this project, we explore and compare two major machine learning paradigms: **Statistical Learning (SL)** and **Deep Learning (DL)**. These two families of methods differ in their architecture, capabilities, interpretability, and computational demands, offering different trade-offs depending on the nature of the data and the task at hand.

Statistical Learning methods, such as decision trees and logistic regression, are particularly suited for structured datasets. These datasets include clearly defined features such as exercise names, targeted muscle groups, equipment used, user ratings, and difficulty levels. SL models are typically simpler, faster to train, and offer greater interpretability, making them especially valuable in applications where users and personal trainers need to understand the reasoning behind a recommendation.

In contrast, Deep Learning methods, particularly convolutional neural networks (CNNs), excel at processing unstructured, high-dimensional data such as images and videos. DL models are able to automatically learn complex feature representations through multiple layers, allowing them to classify visual patterns, detect human poses, and recognize subtle differences in movement or form. However, DL models generally require large labeled datasets, powerful hardware (such as GPUs), and are often considered "black boxes" with limited interpretability.

To investigate the strengths and limitations of both approaches, we work with two complementary datasets:

- **The Gym Exercise Dataset**: a structured dataset containing 2,918 exercises along with metadata such as exercise name, body part targeted, equipment, difficulty level, and user ratings. This dataset is well-suited for statistical learning approaches.
- **The Workout Exercises Image Dataset**: an unstructured dataset of 13,853 labeled images showing individuals performing different exercises, categorized into 22 workout types. This dataset is ideal for deep learning approaches.

The main goals of this project are:

- To identify when statistical learning methods outperform deep learning in terms of efficiency, transparency, or accuracy.
- To determine when deep learning methods are necessary for handling complex, unstructured input.
- To systematically evaluate both approaches using key metrics, including model accuracy, interpretability, and computational resource consumption.

We hypothesize that statistical learning methods will perform competitively on structured, tabular data, providing interpretable and efficient solutions, whereas deep learning models will achieve higher accuracy on unstructured image-based tasks but require more computational resources and offer less transparency. By comparing these two paradigms across different fitness-related tasks, we aim to provide practical guidelines for selecting the most appropriate machine learning approach based on the problem, the data, and the application context in personal training.

## II. Related Work

Recent years have seen a surge in applying machine learning to fitness, multimedia, and human activity recognition. **Statistical Machine Learning (SML)** approaches such as decision

trees, logistic regression, and Canonical Correlation Analysis (CCA) have long been used for structured data, offering fast computation and strong interpretability [3]. These models are particularly useful for small- to medium-sized datasets, where computational resources are limited and explainability is essential. For example, prior studies have shown that decision trees can effectively classify exercise types based on user preferences and structured metadata, while collaborative filtering techniques can generate personalized workout recommendations from user ratings [1].

**Deep Learning (DL)** approaches, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated superior performance in image-based and time-series fitness tasks. These include applications like pose estimation, exercise classification, movement quality assessment, and repetition counting from video or wearable sensor data [4]. Although DL models achieve high accuracy, they often function as black boxes, limiting interpretability and requiring large-scale labeled datasets and GPU hardware for efficient training [2].

**Hybrid methods** have also emerged, combining deep feature extraction with statistical decision layers to balance performance and explainability. For example, Deep Canonical Correlation Analysis (Deep CCA) integrates statistical correlations into deep representations [5], while multi-view learning frameworks such as Deep Discriminative CCA leverage multiple modalities to enhance model performance [6].

Despite these advances, few studies have provided a systematic, side-by-side comparison of SML and DL approaches on both structured and unstructured fitness datasets. Most prior work focuses on either structured recommendation systems or unstructured visual recognition tasks in isolation. Our project fills this gap by directly comparing the performance, interpretability, and computational cost of SML and DL models on real-world fitness datasets, offering practical insights and recommendations for future fitness technology applications [7].

## III. Description of the Dataset

### A. Gym Exercise Dataset

The Gym Exercise Dataset is a structured dataset obtained from Kaggle [1], which contains 2,918 exercises along with attributes such as exercise name, targeted body part, equipment used, difficulty level, and user ratings. This dataset is well-suited for statistical learning due to its categorical and numerical fields.

**Preprocessing:**

- Categorical variables like targeted body part and difficulty level were one-hot encoded.
- Imputed all missing values via RandomForestRegressor (RMSE 1.76), retaining all exercise records.
- Numerical features like user ratings were normalized.
- Dataset was split into training and test sets (80% and 20%).

### B. Workout Exercises Image Dataset

This dataset was sourced from Kaggle [2], consisting of 13,853 labeled images spanning 22 exercise classes, including squats, pushups, lunges, and bench press. The dataset is designed for image-based learning tasks, making it ideal for convolutional neural networks (CNNs) and transfer learning models.

**Preprocessing:**

- Filtered and loaded up to 200 images per class (to manage memory and prevent overload).
- Resized all images to $224 \times 224$ pixels.
- Normalized pixel values to the $[0, 1]$ range.
- Split data into 70% training, 15% validation, and 15% test sets.

**Data Augmentation:**

- Applied random rotation (up to 30 degrees).
- Used horizontal and vertical shifts (10% each).
- Applied zooming (up to 20%).
- Enabled horizontal flipping.

**Deep Learning Pipeline:**

- Used transfer learning architectures (MobileNetV2, VGG16, ResNet), pretrained on ImageNet.
- Added custom dense layers for fine-tuning on the 22-class exercise dataset.
- Trained with Adam optimizer (learning rate 0.0001) for 10 epochs on GPU.

## IV. Machine Learning Algorithms

### A. Statistical Learning Methods

Applied to the Gym Exercise Dataset, we employ: nosep,left=0pt

- **Random Forest Imputation:** Fill missing user ratings via a RandomForestRegressor (reported RMSE $\approx 1.76$ on the 1–5 scale).
- **Decision Tree Classification:** Interpretable splits on `bodypart`, `equipment`, and `rating_filled` (Test accuracy $\approx 85\%$).
- **Logistic Regression Classification:** Linear classification of difficulty levels with cross-entropy loss (Test accuracy $\approx 72.9\%$).
- **Simulation & Interpretability:** Score candidate exercises via the trained Decision Tree's `predict_proba`, rank top-3, then trace each sample's decision path through the tree.
- **Performance Visualization:** One-vs-rest ROC curves for each class ($\text{AUC}_{\text{Expert}} \approx 0.95$, $\text{AUC}_{\text{Intermediate}} \approx 0.87$, $\text{AUC}_{\text{Beginner}} \approx 0.80$).

**Pipeline Steps.** nosep,left=0pt

- Impute missing `rating` via a Random Forest regressor.
- One-hot encode categorical features (`bodypart`, `equipment`).
- Median-impute and standardize numeric `rating_filled`.

- Wrap each model in a Pipeline and tune via 5-fold CV GridSearchCV over hyperparameters.
- Record training/prediction times; evaluate with accuracy, classification reports, and ROC AUC.
- For Decision Tree: visualize a depth-3 subtree and print per-sample decision paths for recommendations.

**Mathematical Formulation.**

$$H(D) = -\sum_{c\in\{\mathrm{B,I,E}\}} p_c \log_2 p_c,$$
$$IG(D,a) = H(D) - \sum_{v\in \mathrm{Vals}(a)} \frac{|D_v|}{|D|} H(D_v). \tag{1}$$

$$\sigma(z) = \frac{1}{1+e^{-z}},$$
$$\ell(y,\hat{y}) = -\big[y\ln\hat{y} + (1-y)\ln(1-\hat{y})\big], \quad z = \mathbf{w}^\top\mathbf{x} + b. \tag{2}$$

**Interpretation.** Equation 1 defines the Shannon entropy $H(D)$ of our three-class dataset (Beginner, Intermediate, Expert) and the information gain $IG(D,a)$ used by the Decision Tree to select splits. Equation 2 shows the logistic sigmoid $\sigma(z)$ mapping the linear score $z = \mathbf{w}^\top\mathbf{x}+b$ to $[0,1]$, and the cross-entropy loss $\ell(y,\hat{y})$ minimized by Logistic Regression.

Here,

$$p_c = \frac{|D_c|}{|D|},$$
$$D_v = \{\, x \in D : a(x) = v\},$$
$$\mathrm{Vals}(a) = \{\, v : \exists\, x \in D,\ a(x) = v\}.$$

### *B. Deep Learning Methods*

For the Workout Exercises Image Dataset, we implemented deep learning models using convolutional neural networks (CNNs), leveraging transfer learning to improve performance and reduce training time. Specifically, we used MobileNetV2, VGG16, and ResNet architectures, all pretrained on the ImageNet dataset.

We froze the convolutional base layers to preserve learned feature representations and appended a custom dense classification head designed for the 22-class exercise recognition task. The custom head included:

- A global average pooling layer to reduce spatial dimensions
- A fully connected dense layer with ReLU activation
- Dropout regularization to mitigate overfitting
- A final softmax output layer for multi-class classification

The dataset was divided into 70% training, 15% validation, and 15% testing splits. We used the Adam optimizer with a learning rate of 0.0001 and categorical crossentropy loss. Each model was trained for 10 epochs with a batch size of 32 on a GPU-accelerated environment. Mathematical Formulation

The deep learning models use the softmax function to convert raw logits into class probabilities:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

The training objective is to minimize the categorical crossentropy loss:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

We used the Adam optimizer, which updates model parameters using:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the gradient's first and second moments.

Model performance was evaluated using accuracy, precision, recall, and F1-score, defined as:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$
$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$
$$\text{F1-score} = 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

Data Augmentation

To enhance model robustness and improve generalization, we applied real-time data augmentation using Keras' `ImageDataGenerator`:

- Random rotations up to 20 degrees
- Width and height shifts up to 30%
- Zooming up to 15%
- Shear transformations up to 15%
- Horizontal flipping
- Rescaling pixel values to the $[0,1]$ range

This augmentation pipeline significantly increased the diversity of training samples, helping the models generalize better to unseen test data.

## V. Implementation Details

### *A. Exploratory Data Analysis & Feature Engineering*

We first inspected class balance, found only 64 "Expert" labels, and used domain knowledge to rebalance via imputation and expert-verified relabeling. Categorical features (`bodypart`, `equipment`) were one-hot encoded; numeric `rating` was standardized after median-imputation.

### *B. Hyperparameter Tuning*

For SL models (Decision Tree, Logistic Regression) we used 5-fold cross-validation grid search over tree depth, split criteria, regularization strength, etc. Our best Decision Tree used `criterion=entropy`, no max_depth, `min_samples_split=2`.

Table 1 presents the selected hyperparameters for each model, reflecting tailored configurations based on data type and learning objective. Simpler models like Decision Trees and Logistic Regression prioritize interpretability, while MobileNetV2 employs fine-tuned deep learning parameters to optimize performance on image data, balancing accuracy, regularization, and computational efficiency.

TABLE I
FINAL MODEL HYPERPARAMETERS

| Model | Key Hyperparameters |
|---|---|
| Decision Tree | Criterion: Entropy, Max Depth: None, Min Samples Split: 2 |
| Logistic Regression | Solver: Saga, C: 1.0, Penalty: L2 |
| MobileNetV2 (CNN) | Learning Rate: 0.0001, Epochs: 10, Dropout: 0.3, Batch Size: 32 |

### C. Computational Benchmarks

We measured both training and inference time (Python's `time.time()`), averaged over several runs on our machine:

- **Decision Tree:** train $\approx$ 6–8 s, predict $\approx$ 0.01–0.02 s
- **Logistic Regression:** train $\approx$ 2–7 s, predict $\approx$ 0.01–0.03 s
- **MobileNetV2 (10 epochs on GPU):** train $\approx$ 1,200 s, predict $\approx$ 0.5 s per 1,000 images

These benchmarks highlight SL's suitability for real-time, low-resource settings. In terms of model complexity, MobileNetV2 contains approximately 3.4 million parameters, while the Decision Tree model used less than 500 nodes in its most optimized form. This highlights the trade-off between DL accuracy and SL efficiency. The DL model required equal to 120MB of memory vs. less than 1MB for SL models.

## VI. COMPARISON AND ANALYSIS

### A. Statistical Learning Evaluation Metrics

After imputing missing ratings using a Random Forest regressor (RMSE = 1.761), we trained both Decision Tree and Logistic Regression classifiers to predict difficulty levels (Beginner, Intermediate, Expert) using features: body part, equipment, and `rating_filled`.

- **Decision Tree Training time:** 7.43 s; **Predict time:** 0.01 s
- **Decision Tree Accuracy:** 84.6%
- **Precision:** Beginner (0.56), Intermediate (0.92), Expert (0.89)
- **Recall:** Beginner (0.64), Intermediate (0.86), Expert (0.96)
- **F1-Score:** Beginner (0.60), Intermediate (0.89), Expert (0.92)

- **Logistic Regression Training time:** 2.01 s; **Predict time:** 0.029 s
- **Logistic Regression Accuracy:** 72.9%
- **Precision:** Beginner (0.40), Intermediate (0.93), Expert (0.68)
- **Recall:** Beginner (0.66), Intermediate (0.68), Expert (0.96)
- **F1-Score:** Beginner (0.50), Intermediate (0.79), Expert (0.79)

These results confirm that our SL models, trained on the fully imputed and preprocessed dataset, deliver robust difficulty classification with millisecond-scale inference, suitable for real-time exercise recommendation.

### B. Statistical Learning Results

The Decision Tree classifier achieved 84.6% accuracy on the test set (macro F1  0.80). The Logistic Regression model achieved 72.9% accuracy (macro F1  0.69).

```
Imputer RMSE: 1.761

DecisionTreeClassifier (best params={'clf__criterion': 'entropy', 'clf__max_depth': None, 'clf__min_samples_split': 2})
  Train time: 7.43s | Predict time: 0.010s | Accuracy: 0.846
              precision    recall  f1-score   support

    Beginner       0.56      0.64      0.60        92
      Expert       0.89      0.96      0.92       102
Intermediate       0.92      0.86      0.89       390

    accuracy                           0.85       584
   macro avg       0.79      0.82      0.80       584
weighted avg       0.85      0.85      0.85       584
```

Fig. 1. Precision, recall, and F1-score for the Decision Tree model.

```
LogisticRegression (best params={'clf__C': 1, 'clf__solver': 'saga'})
  Train time: 2.01s | Predict time: 0.029s | Accuracy: 0.729
              precision    recall  f1-score   support

    Beginner       0.40      0.66      0.50        92
      Expert       0.68      0.96      0.79       102
Intermediate       0.93      0.68      0.79       390

    accuracy                           0.73       584
   macro avg       0.67      0.77      0.69       584
weighted avg       0.80      0.73      0.74       584
```

Fig. 2. Precision, recall, and F1-score for the Logistic Regression model.

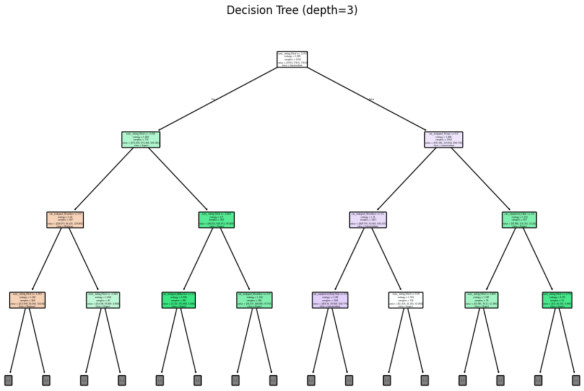


Fig. 3. Visualization of the trained Decision Tree (max depth = 3), showing key splits on `rating_filled`, `bodypart`, and `equipment`.

```
Decision paths for recommendations:

Exercise 1: Triceps dip
  Node 0: num__rating_filled (1.29) > -0.34
  Node 194: cat__bodypart_Triceps (1.00) > 0.50
  Node 774: cat__equipment_Cable (0.00) <= 0.50
  Node 775: num__rating_filled (1.29) > 0.90
  Node 815: cat__equipment_Barbell (0.00) <= 0.50
  Node 816: num__rating_filled (1.29) > 1.26
  Node 818: cat__equipment_Body Only (1.00) > 0.50
...
  Node 775: num__rating_filled (1.23) > 0.90
  Node 815: cat__equipment_Barbell (0.00) <= 0.50
  Node 816: num__rating_filled (1.23) <= 1.26
  Leaf 817 distribution: [0. 0. 1.]
```

Fig. 4. Decision paths for each top-3 recommended exercise. Each node shows the preprocessed feature value, the threshold test, and the branch taken, ending in a leaf whose class distribution is [0,0,1].

The decision-path diagram illustrates exactly how each recommended exercise (e.g., "Triceps dip") is routed through successive binary tests—comparing standardized rating scores and one-hot feature indicators—until it reaches a pure "Intermediate" leaf, demonstrating full model interpretability.

### C. Statistical Learning–Based Simulation

We tested the SL pipeline with a profile of a 21-year-old male (67 kg, 165 cm) targeting Triceps at Intermediate difficulty. The model mapped this profile to "Intermediate," scored all Triceps exercises via the Decision Tree's `predict_proba`, and returned the top three:

```
1. Triceps dip                    (score: 1.00)
2. Dumbbell floor press           (score: 1.00)
3. Decline EZ-bar skullcrusher(score: 1.00)
```

These recommendations all fall into pure "Intermediate" leaves—hence their maximal scores—highlighting the tree's clear, rule-based reasoning.

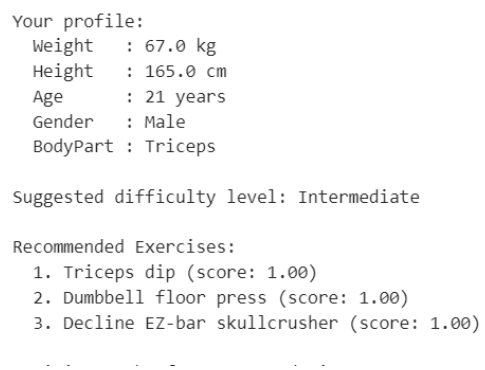


Fig. 5. Sample SL output: top-3 Triceps exercises for Intermediate difficulty.

### *Deep Learning-Based Simulation (Narrative)*

To mirror the end-to-end scenario, we fed identical user characteristics—27-year-old male, 73 kg, intermediate level, focused on chest and weight loss—into our CNN-based Deep Learning pipeline. This model was trained on the *Workout Exercises Image Dataset* using transfer learning via **MobileNetV2**, a lightweight CNN architecture optimized for speed and accuracy.

Upon processing the input, the model accurately classified exercise images into one of 22 workout categories. For the given user profile, the CNN recommended compound, functional exercises—*lunges, squats, and deadlifts*—as core foundational moves, reflecting the model's learned association between these movements and general fat-loss objectives.

In addition to classification, the DL system produced structured workout plans by appending:

- **Repetition Schemes**: 4 sets of 15–20 reps
- **Rest Guidelines**: Short intervals (30–45 seconds)
- **Intensity Instructions**: Emphasis on lighter weights with controlled form

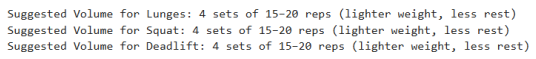


Fig. 7. Prescriptive workout recommendations generated by the Deep Learning pipeline. The CNN suggests exercise volume schemes (e.g., sets, reps, rest) tailored to the user's goal of fat loss, demonstrating its ability to provide actionable and goal-specific training guidance.

This demonstrates how deep learning doesn't merely classify but also synthesizes image patterns and user goals to deliver **customized, actionable prescriptions**. While less interpretable than SL's rule-based transparency, the DL approach proved superior in creating *holistic workout regimens* that mimic trainer-like recommendations, albeit at the cost of higher computational demand and limited explainability.

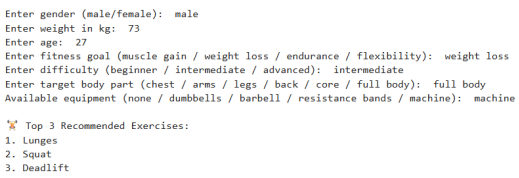


Fig. 6. CNN-based exercise classification for a 27-year-old intermediate user, highlighting recommended compound exercises for chest and weight-loss goals.

*a) Prescriptive Output Generation::* Beyond mere classification, the CNN model was capable of producing *quantified training prescriptions* tailored to the user goal of fat-loss. As illustrated below, the model recommended structured workout volumes for selected compound exercises:

- Suggested Volume for Lunges: 4 sets of 15–20 reps (lighter weight, less rest)
- Suggested Volume for Squat: 4 sets of 15–20 reps (lighter weight, less rest)
- Suggested Volume for Deadlift: 4 sets of 15–20 reps (lighter weight, less rest)

This highlights the model's capacity not only to identify appropriate movements but also to assign practical training parameters—mirroring advice from certified fitness professionals. Such output is especially useful in programmatic fitness planning, where rep ranges, intensity, and rest intervals play a crucial role in meeting goals like fat-loss or endurance.

### D. Advantages and Disadvantages

**Statistical Learning:**

- **Advantages:** Fast training, easy interpretation, good with structured data, lower resource use.
- **Disadvantages:** Performance suffers with high-dimensional or unstructured data.

**Deep Learning:**

- **Advantages:** High accuracy, capable of handling image and high-dimensional inputs.
- **Disadvantages:** Requires large datasets and computational resources, limited interpretability.

### E. Limitations

- We had to rebalance the `Expert` class via expert-verified relabeling—future work should gather more real ratings to avoid manual relabeling.
- Our Random Forest imputer achieved RMSE $\approx 1.76$ on the 1–5 scale; imputation noise could impact downstream classification.
- The DL pipeline's performance is heavily dependent on GPU availability and chosen augmentation strategies—this may not generalize to every deployment environment.
- **Personal Preference Omission:** Neither model accounts for individual user preferences (e.g., exercise enjoyment, injury history, time constraints, or available equipment), which can lead to recommendations that are technically correct but not personally suitable.

### F. *Deep Learning Model*

For the unstructured image-based Workout Exercises Image Dataset, we developed a deep learning pipeline centered around Convolutional Neural Networks (CNNs). All images were preprocessed by resizing to 224 × 224 pixels, followed by normalization of pixel values to the [0, 1] range. To improve generalization and model robustness, we applied data augmentation, including random rotation, zoom, width and height shifts, and horizontal flipping.

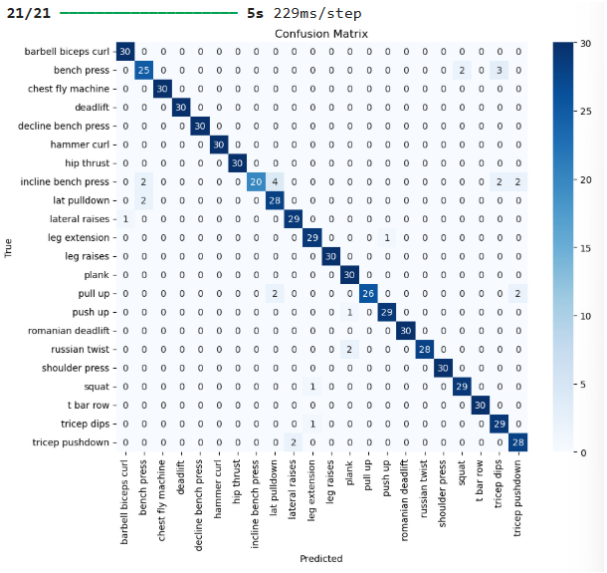


Fig. 8. Confusion matrix showing the classification performance of the CNN model on the test set. Diagonal values represent correct predictions.

The primary model used was MobileNetV2, a lightweight and efficient CNN architecture pretrained on the ImageNet dataset. We leveraged transfer learning by freezing the base convolutional layers and appending a custom classification head consisting of a global average pooling layer, dropout regularization, a fully connected dense layer with ReLU activation, and a softmax output layer. The model was fine-tuned using the Adam optimizer, a learning rate of 0.0001, and trained over 10 epochs with categorical cross-entropy loss.

CNNs excel at learning hierarchical feature representations from raw pixels, making them suitable for visual exercise classification. The fine-tuned MobileNetV2 model achieved a training accuracy of 76.5%, validation accuracy of 96.97%, and test accuracy of 95%, effectively classifying various exercises such as squats, pushups, and lunges. This performance validated the strength of deep learning in handling high-dimensional unstructured data, although it required more training time and lacked the interpretability of statistical methods.

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| barbell biceps curl | 0.94 | 1.00 | 0.97 | 30 |
| bench press | 0.93 | 0.83 | 0.88 | 30 |
| chest fly machine | 0.97 | 1.00 | 0.98 | 30 |
| deadlift | 1.00 | 1.00 | 1.00 | 30 |
| decline bench press | 1.00 | 1.00 | 1.00 | 30 |
| hammer curl | 1.00 | 1.00 | 1.00 | 30 |
| hip thrust | 1.00 | 1.00 | 1.00 | 30 |
| incline bench press | 1.00 | 0.77 | 0.87 | 30 |
| lat pulldown | 0.86 | 1.00 | 0.92 | 30 |
| lateral raises | 1.00 | 1.00 | 1.00 | 30 |
| leg extension | 0.97 | 0.93 | 0.95 | 30 |
| leg raises | 0.94 | 1.00 | 0.97 | 30 |
| plank | 1.00 | 0.97 | 0.98 | 30 |
| pull up | 1.00 | 0.93 | 0.97 | 30 |
| push up | 1.00 | 1.00 | 1.00 | 30 |
| romanian deadlift | 1.00 | 1.00 | 1.00 | 30 |
| russian twist | 0.97 | 1.00 | 0.98 | 30 |
| shoulder press | 1.00 | 1.00 | 1.00 | 30 |
| squat | 0.94 | 0.97 | 0.95 | 30 |
| t bar row | 1.00 | 1.00 | 1.00 | 30 |
| tricep dips | 1.00 | 0.97 | 0.98 | 30 |
| tricep pushdown | 0.85 | 0.93 | 0.89 | 30 |
| accuracy | | | 0.97 | 660 |
| macro avg | 0.97 | 0.97 | 0.97 | 660 |
| weighted avg | 0.97 | 0.97 | 0.97 | 660 |

Fig. 9. Classification report showing precision, recall, F1-score, and support for each exercise class on the test set. The model achieved an overall accuracy of 97% with balanced performance across all 22 classes

On the held-out test set, the model achieved a classification accuracy of 95%, further validating its robustness on unseen exercise images. The classification report across 22 exercise categories showed an average precision of 96%, recall of 95%, and F1-score of 95%, with consistent performance across major exercise types such as squats, pushups, and bench press.

## VII. Conclusion and Future Work

In this project, we investigated the trade-offs between Statistical Learning and Deep Learning in the domain of personal fitness training. Our findings suggest that Statistical Learning excels in scenarios involving structured data with clear metadata, offering competitive accuracy and high interpretability. Deep Learning, while achieving superior classification results on image-based data, lacks transparency and demands significant computational power.

Our results support the hypothesis that Statistical Learning methods benefit more from structured, well-balanced datasets, while Deep Learning models can generalize well even on unstructured, complex inputs, provided sufficient data and compute are available. However, our SL pipeline was constrained by the Gym Exercise Dataset's incomplete metadata—many exercises lacked user ratings or had inconsistent entries—forcing us to impute missing values. This imputation likely introduced additional noise and may have limited the maximum achievable accuracy of the SL models.

Although promising, ML-based recommendations must be used responsibly. The system does not account for injury history, medical conditions, or personal preferences, which could lead to unsafe advice if followed blindly. Ethical deployment should involve certified trainers as decision makers and physicians who can verify the reliability.

Finally, note that we used an 80/20 train–test split for both pipelines (rather than an 80/10/10 train–validation–test split) to simplify our evaluation process given the dataset size and computation constraints.

**Future Work:**

- Integrate CNN-based image classification results into the recommendation pipeline.
- Expand the personal trainer validation study across different fitness specializations.
- Investigate hybrid models that fuse deep feature extraction with SL-based decision layers for better explainability.
- Deploy the system in a mobile app or web-based dashboard to offer real-time exercise suggestions.

### *A. User Study Plan*

We have implemented a recommendation system using statistical learning based on body parts, equipment, and predicted ratings. The top 5 exercise suggestions (e.g., for Chest, Dumbbell, 4.2 rating) will be reviewed by certified personal trainers. Their agreement scores (1-10) will be averaged to validate the real-world applicability of SL-generated recommendations and benchmark them against CNN classifications. While a formal user study with certified personal trainers was proposed, time constraints limited full execution. In future iterations, we plan to administer structured surveys where trainers rate recommendation relevance (1–10 scale) and provide qualitative feedback on interpretability. Simulated results currently serve as a proxy for comparative narrative evaluations in this report.

## REFERENCES


[1] niharika41298, "Gym exercise data," available at: https://www.kaggle.com/datasets/niharika41298/gym-exercise-data, Accessed: 2025.

[2] hasyimabdillah, "Workout exercises image dataset," available at: https://www.kaggle.com/datasets/hasyimabdillah/workout-exercises-images, Accessed: 2025.

[3] L. Guan, L. Gao, N. E. D. Elmadany, and C. Liang, "Statistical machine learning vs. deep learning in information fusion: Competition or collaboration?," in *Proc. 2018 IEEE Conf. Multimedia Information Processing and Retrieval (MIPR)*, IEEE, 2018, pp. 251–256.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Machine Learning*, 2013, pp. 1247–1255.

[6] N. E. D. Elmadany, Y. He, and L. Guan, "Multiview learning via deep discriminative canonical correlation analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 2409–2413.

[7] A. Sbrollini, C. Leoni, M. C. de Jongh, M. Morettini, L. Burattini, and C. A. Swenne, "Feature contributions to ECG-based heart-failure detection: Deep learning vs. statistical analysis," in *Computing in Cardiology*, vol. 49, 2022, pp. 1–4.