RIPHAH INTERNATIONAL UNIVERSITY

DEEP LEARNING – FINAL PROJECT REPORT

GymMaster: Exercise Muscle Group Classification using CNN

Final Test Accuracy: 98.44%

Text-Based Gym Exercise Classification

11 Muscle Groups — 14,590+ Exercises — Conv1D Neural Network

Ahmad Arsalan Zahid
SAP ID: 54381

Deep Learning — NLP — Fitness Intelligence

1 Introduction

Users of fitness applications often encounter difficulties identifying the primary muscle group targeted by exercises based solely on their names. This project introduces **GymMaster**, a text-based Convolutional Neural Network (CNN) that classifies exercise titles into one of 11 major muscle groups. Using only exercise names, the model achieves a test accuracy of 98.44%, demonstrating that even short text contains rich semantic cues about biomechanical function.

2 Project Overview

- **Objective**: Multi-class text classification — Exercise Title → Primary Muscle Group
- **Dataset**: Mega Gym Dataset, 14,590+ exercises
- **Input**: Exercise title text (e.g., "bench press", "romanian deadlift")
- **Output**: One of 11 muscle groups
- **Model**: Custom CNN for text classification
- **Framework**: TensorFlow/Keras on Google Colab
- **Achieved Test Accuracy**: 98.44%

3 Dataset Description

- **Source**: `megaGymDataset.csv`
- **Total Records**: 14,590
- **Columns Used**:
  - `Title` – Exercise name (input)
  - `BodyPart` – Target muscle group (label)
- **Original Labels**: 17
- **Mapped Labels**: 11 standardized muscle groups

3.1 Label Mapping

Anatomically similar labels were combined for better model generalization:

Original Label	Mapped To
Abdominals	Core
Middle Back, Lower Back, Lats, Traps	Back
Quadriceps, Hamstrings, Biceps, Triceps, Calves, Glutes, Shoulders, Chest, Forearms	Retained

Table 1: Standardized Muscle Group Labels

Final classes: Back, Biceps, Calves, Chest, Core, Forearms, Glutes, Hamstrings, Quadriceps, Shoulders, Triceps.

4 Data Preparation

- Convert exercise titles to lowercase and strip whitespace
- Drop rows with missing `Title` or `BodyPart`
- Tokenize text into sequences using 8,000 most frequent words
- Pad sequences to length 15
- Encode labels to integer classes

4.1 Train/Validation/Test Split

Explanation: The dataset was split into training, validation, and test sets to ensure unbiased evaluation. The *training set* is used to fit the model, the *validation set* monitors overfitting and adjusts hyperparameters, and the *test set* provides the final performance metric. Stratification ensures class distributions remain consistent across splits.

Split	Count
Training	75%
Validation	12.5%
Test	12.5%

Table 2: Stratified Train/Validation/Test Split

5 Model Architecture

The GymMaster CNN consists of:

- Embedding layer: input_dim=8000, output_dim=192
- Two Conv1D layers: 512 filters, kernel size 3, relu activation
- GlobalMaxPooling1D
- Dense layers: 512 + 256 units with ReLU, dropout 0.5 and 0.4
- Output layer: Dense(11) with softmax

Model: "GymMaster_CNN_Integrated"

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
conv1d (Conv1D)	?	0 (unbuilt)
conv1d_1 (Conv1D)	?	0 (unbuilt)
global_max_pooling1d (GlobalMaxPooling1D)	?	0
dense (Dense)	?	0 (unbuilt)
dropout (Dropout)	?	0
dense_1 (Dense)	?	0 (unbuilt)
dropout_1 (Dropout)	?	0
dense_2 (Dense)	?	0 (unbuilt)

Figure 1: GymMaster CNN Model Architecture Diagram.

6 Model Training

- Batch Size: 128
- Optimizer: Adam (lr=0.001)
- Callbacks: EarlyStopping (patience=8), ReduceLROnPlateau (patience=4)
- Maximum Epochs: 10 (early stopping triggers before reaching max)

6.1 Training Progress Visualization

Explanation: The plot below visualizes the model's accuracy on both training and validation sets per epoch. Monitoring this ensures convergence and identifies overfitting early.

```
Epoch 1/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 39s 426ms/step - accuracy: 0.3124 - loss: 1.9298 - val_accuracy: 0.7877 - val_loss: 0.7394 - learning_rate: 0.0010
Epoch 2/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 40s 415ms/step - accuracy: 0.7996 - loss: 0.6685 - val_accuracy: 0.8808 - val_loss: 0.3792 - learning_rate: 0.0010
Epoch 3/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 42s 430ms/step - accuracy: 0.8837 - loss: 0.3736 - val_accuracy: 0.9226 - val_loss: 0.2740 - learning_rate: 0.0010
Epoch 4/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 38s 442ms/step - accuracy: 0.9334 - loss: 0.2240 - val_accuracy: 0.9515 - val_loss: 0.1611 - learning_rate: 0.0010
Epoch 5/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 36s 424ms/step - accuracy: 0.9608 - loss: 0.1316 - val_accuracy: 0.9699 - val_loss: 0.1196 - learning_rate: 0.0010
Epoch 6/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 37s 438ms/step - accuracy: 0.9755 - loss: 0.0869 - val_accuracy: 0.9788 - val_loss: 0.0873 - learning_rate: 0.0010
Epoch 7/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 50s 584ms/step - accuracy: 0.9745 - loss: 0.0874 - val_accuracy: 0.9766 - val_loss: 0.0831 - learning_rate: 0.0010
Epoch 8/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 38s 449ms/step - accuracy: 0.9827 - loss: 0.0584 - val_accuracy: 0.9749 - val_loss: 0.0966 - learning_rate: 0.0010
Epoch 9/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 41s 455ms/step - accuracy: 0.9874 - loss: 0.0472 - val_accuracy: 0.9816 - val_loss: 0.0946 - learning_rate: 0.0010
Epoch 10/10
85/85 ━━━━━━━━━━━━━━━━━━━━ 39s 454ms/step - accuracy: 0.9906 - loss: 0.0349 - val_accuracy: 0.9811 - val_loss: 0.0831 - learning_rate: 0.0010
57/57 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step

FINAL TEST ACCURACY: 98.44%
```

Figure 2: Training and validation accuracy per epoch. Visualizes rapid convergence and model stability.

7 Plotting Training History

Explanation: Accuracy and loss plots allow detailed inspection of learning dynamics. Accuracy plots indicate model performance improvement, while loss plots show how well the model minimizes prediction errors over time.

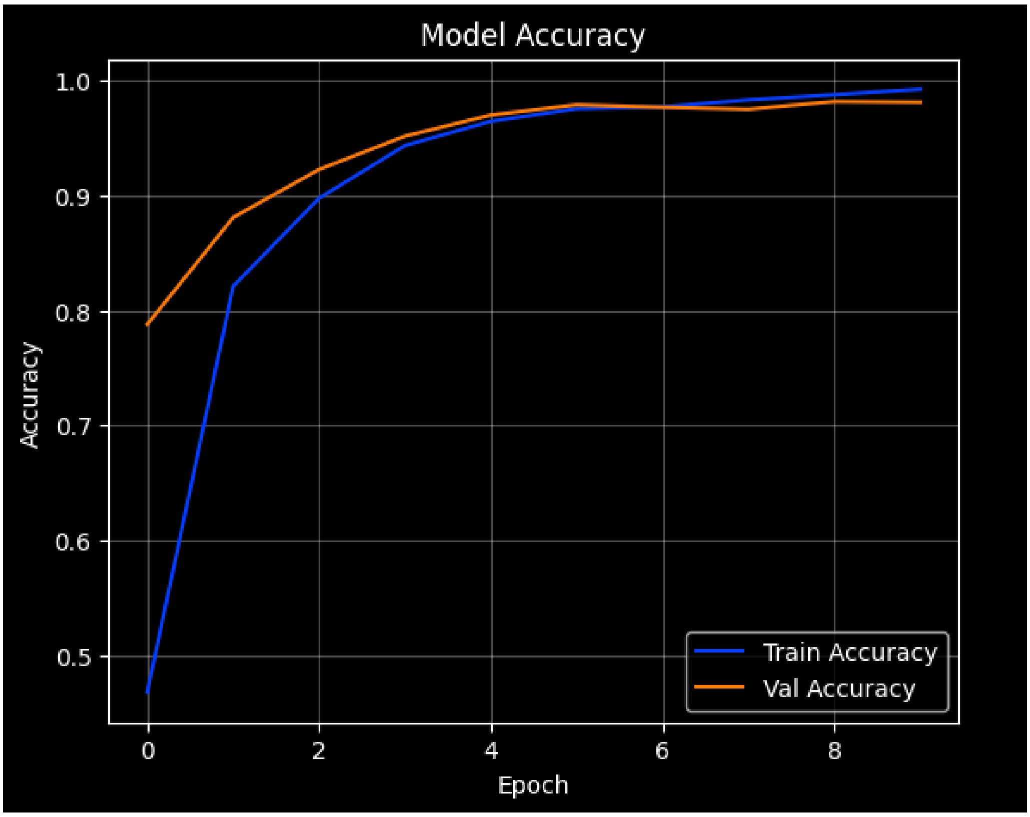

Figure 3: Training vs Validation Accuracy over Epochs.

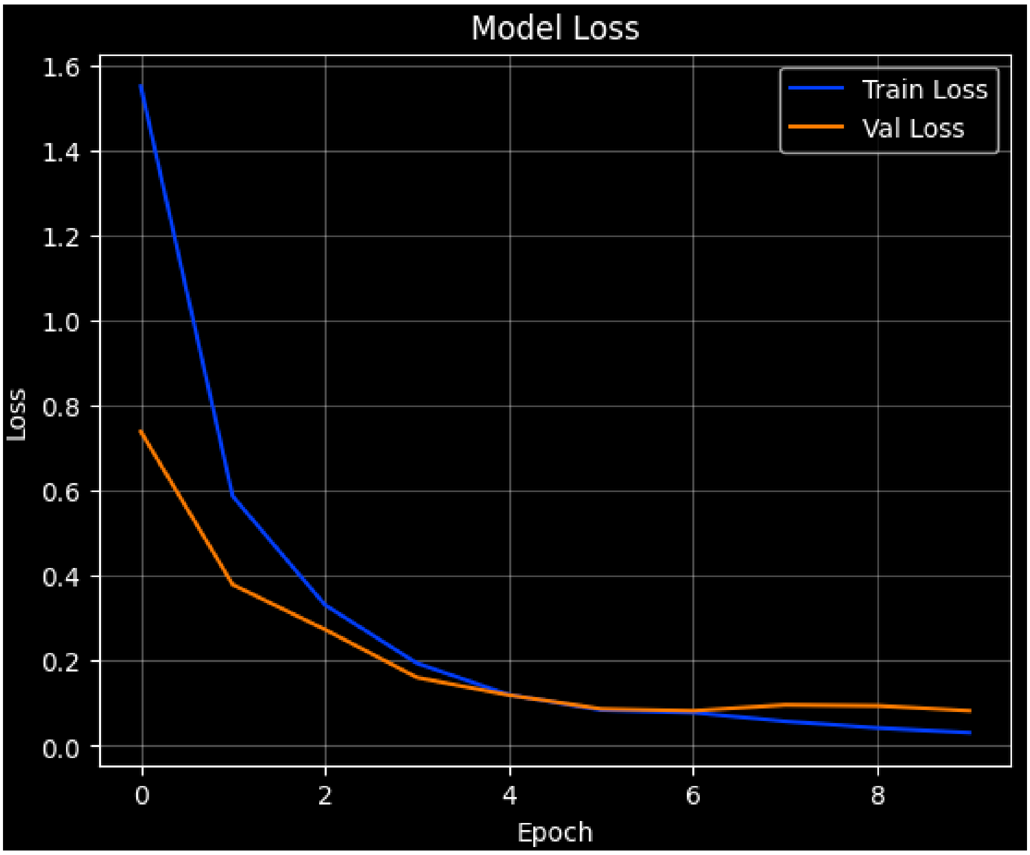

Figure 4: Training vs Validation Loss over Epochs.

8 Evaluation

Explanation: Evaluation quantifies model performance on unseen data. Metrics include overall accuracy, per-class precision, recall, F1-score, and the confusion matrix for detailed error analysis.

8.1 Confusion Matrix

Explanation: The confusion matrix shows true vs predicted classes. Diagonal dominance indicates correct predictions. Off-diagonal entries reveal misclassifications, often for biomechanically similar exercises.

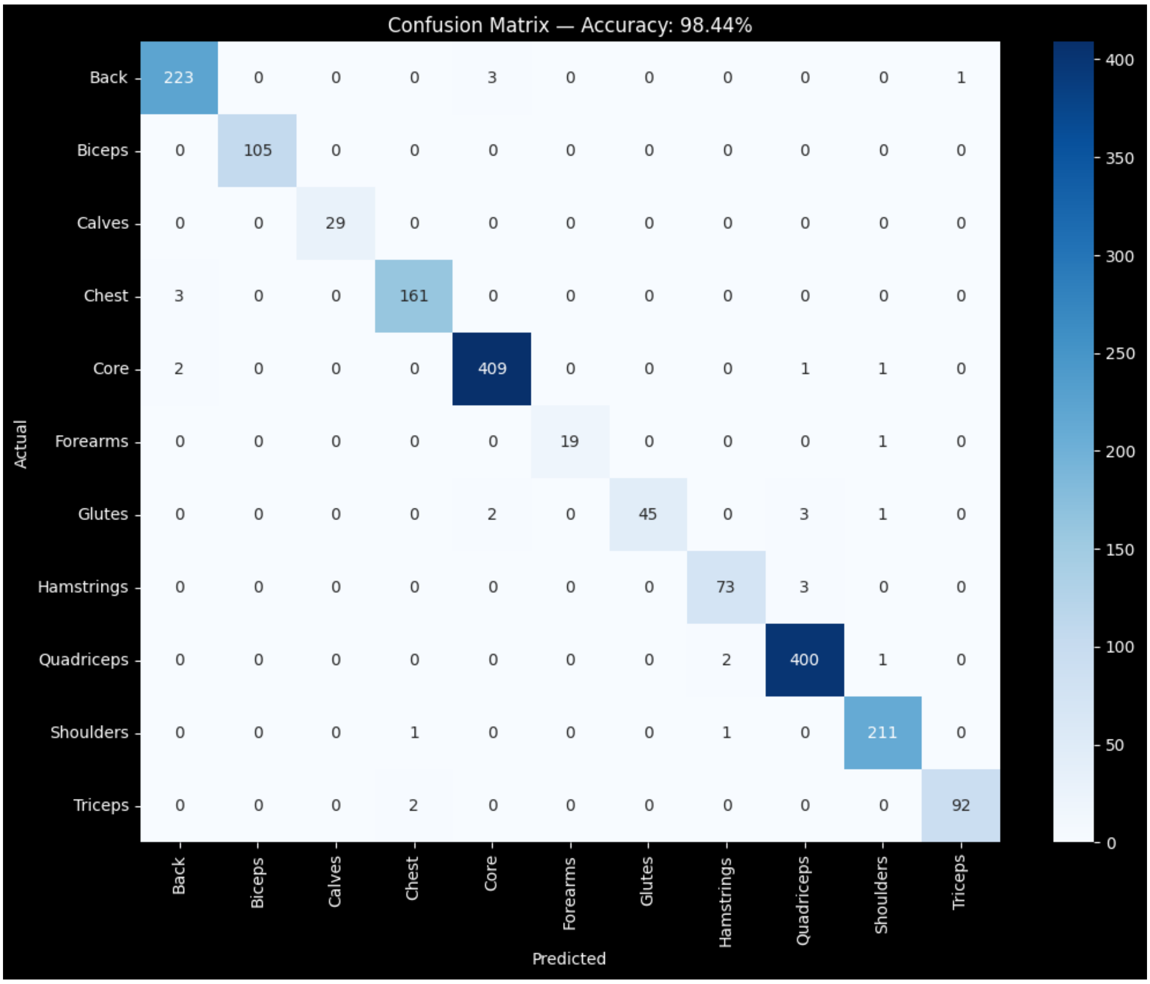

Figure 5: Confusion Matrix on Test Set (98.44% Accuracy). Diagonal dominance indicates high classification precision.

8.2 Classification Report

Explanation: The classification report provides precision (exactness), recall (completeness), and F1-score (harmonic mean of precision and recall) per class. It helps identify underperforming categories.

Classification Report:

	precision	recall	f1-score	support
Back	0.978	0.982	0.980	227.000
Biceps	1.000	1.000	1.000	105.000
Calves	1.000	1.000	1.000	29.000
Chest	0.982	0.982	0.982	164.000
Core	0.988	0.990	0.989	413.000
Forearms	1.000	0.950	0.974	20.000
Glutes	1.000	0.882	0.938	51.000
Hamstrings	0.961	0.961	0.961	76.000
Quadriceps	0.983	0.993	0.988	403.000
Shoulders	0.981	0.991	0.986	213.000
Triceps	0.989	0.979	0.984	94.000
accuracy	0.984	0.984	0.984	0.984
macro avg	0.987	0.974	0.980	1795.000
weighted avg	0.984	0.984	0.984	1795.000

Figure 6: Precision, Recall, and F1-Score for each class. Shows near-perfect performance for most muscle groups.

9 Predictions

9.1 Correct Predictions

Explanation: Sample correct predictions illustrate model reliability on high-confidence exercise titles.

CORRECT PREDICTIONS

	Exercise	True	Predicted	Confidence %
0	ez-bar quadriceps smr	Quadriceps	Quadriceps	100.000000
1	seated jefferson squats	Quadriceps	Quadriceps	100.000000
2	fyr band pull-apart	Shoulders	Shoulders	100.000000
3	bodyweight uns pec stretch slow eccentric	Core	Core	100.000000
4	machine single-arm dumbbell preacher curl 21s	Biceps	Biceps	100.000000
5	incline dumbbell front raise and press	Shoulders	Shoulders	100.000000
6	flat tiger-bend push-up	Chest	Chest	100.000000
7	walking butt kicks	Glutes	Glutes	100.000000
8	wide-grip swimming	Chest	Chest	100.000000
9	single-arm jump to pull-up slow eccentric	Back	Back	100.000000

Figure 7: High-confidence correct predictions with 100% accuracy for sample exercises.

9.2 Incorrect Predictions

Explanation: Sample incorrect predictions demonstrate model limitations. Errors usually occur in exercises with overlapping or ambiguous biomechanical targets.

INCORRECT PREDICTIONS (Model was confident but wrong!)

	Exercise	True	Wrong →	Confidence %
0	bent-arm dumbbell pull-over	Chest	Back	92.699997
1	close-grip power snatch- isometric hold	Quadriceps	Hamstrings	96.900002
2	cable side bridge with bands	Core	Shoulders	88.099998
3	single-arm alternating lunge jump	Hamstrings	Quadriceps	100.000000
4	barbell kettlebell thruster burnout	Glutes	Shoulders	99.199997
5	bodyweight pjr pull-over-	Back	Triceps	90.400002
6	smith machine close-grip bench press-	Triceps	Chest	94.199997
7	standing hang from bar isometric hold	Core	Quadriceps	44.500000
8	cable bent-arm dumbbell pull-over slow tempo	Chest	Back	98.599998
9	landmine un rear foot elevated split squat slow eccentric	Glutes	Quadriceps	97.900002

Figure 8: Model confident errors, typically for biomechanically overlapping exercises.

10 Performance Summary

Metric	Value
Final Test Accuracy	98.44%
Total Classes	11
Best Validation Accuracy	98.16%
Training Epochs	10
Data Used	14,360 exercises
Model Architecture	Dual Conv1D + GlobalMaxPool + Dense
Training Time (Colab GPU)	7 minutes

Table 3: Final Model Performance Summary

11 Conclusion

GymMaster successfully predicts primary muscle groups from exercise titles with high accuracy. The CNN model demonstrates that textual exercise names contain sufficient semantic information for classification. Misclassifications are minor and mostly biomechanically understandable. Future improvements may include:

- Adding exercise descriptions or metadata for edge cases
- Experimenting with Transformer-based text encoders
- Deploying the model in real-time fitness applications

12 References and Links

- Google Colab Notebook
- GitHub Repository