

Adaptive Gait Rehabilitation using Hybrid RL Algorithms

I. Introduction:

Adaptive gait rehabilitation is a critical field in medical research, aiming to improve mobility and quality of life for individuals with lower-limb disabilities. Traditionally, exoskeletons and other assistive devices have been used to support gait rehabilitation, but they require manual adjustments and lack adaptability. Reinforcement Learning (RL) has emerged as a promising approach to develop intelligent control systems for gait rehabilitation, allowing for real-time adjustments based on feedback from the environment.

This report explains a hybrid approach combining Convolutional Neural Networks (CNN) with Proximal Policy Optimization (PPO) and integration of genetic algorithms (Algae optimization). The goal is to develop a robust RL framework capable of learning and adapting to individual patient needs, ultimately providing better gait rehabilitation outcomes.

II. Methodology:

i. Data Loading and Preprocessing:

We have acquired the data from OU-Biometrics dataset (OU-ISIR). There are total 3 datasets namely TrademillDatasetA, TrademillDatasetB and TrademillDatasetD from which only B and D datasets were utilized during this project. Each dataset has its own significance in the label prediction. Input images are converted to tensors and normalized for consistent input to the neural network. By applying the fusion of PPO reinforcement learning method with algae optimization, the data is trained to perform different types of gaits pattern identifications. Below is the description of each dataset:

Dataset A: This dataset determines the speed feature based on the stair image dataset collected from 34 different subjects.

Dataset B: This dataset demonstrates the correlation of clothing with the gait pattern sequence. This dataset was used to experiment PPO+Algae with CNN as it has more samples. We have made custom labels narrowing down the samples to train on gait patterns.

Dataset D: This dataset is used to analyze the NAC values which will help us checking for abnormalities or fluctuations in gait patterns. In case of this dataset, gait fluctuations are measured by Normalized AutoCorrelation (NAC) for the temporal axis of size-normalized gait silhouette images. By doing this, the recursive nature of the gait aspect is tackled and hence the model training would improve.



Fig – Dataset B input images

ii. Convolutional Neural Network (CNN):

Our project utilizes CNN as the underlying architecture for learning the feature maps from image data. The model consists of:

- *Convolutional Layers:* Both dataset B and D are trained upon a 3 layered with ReLU activation and batch normalization, followed by a max-pooling layer (2,2) to reduce spatial dimensions.
- *Fully Connected Layers:* A sequence of dense layers, including a dropout layer for regularization, and a final output layer corresponding to the number of actions.
- *Output Layer:* A final linear layer to produce outputs for number of actions to the reinforcement agent which are 4 in case of dataset B and 2 for the dataset D.

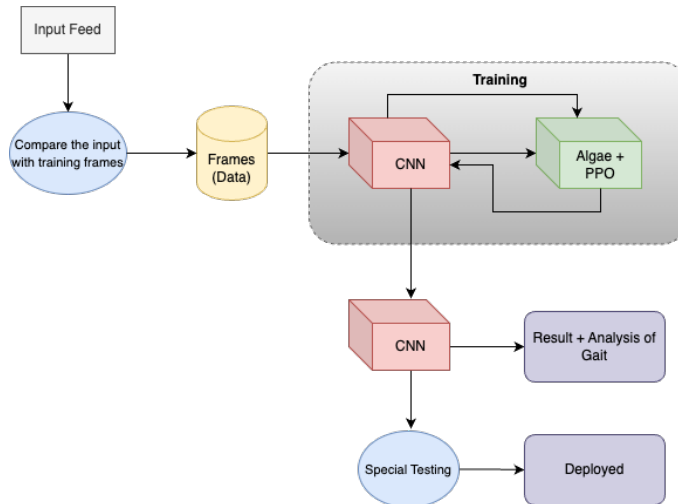


Fig – Proposed project methodology

iii. Proximal Policy Optimization (PPO) + Algae Optimization:

Proximal Policy Optimization (PPO) is a reinforcement learning technique that balances exploration and exploitation during training. This method uses a clipping mechanism to stabilize policy updates and minimize large changes. Algae optimization is inspired by genetic algorithms, using crossover and mutation operations to evolve the weights of the neural networks. This evolutionary approach helps maintain a diverse population of models, promoting robustness and adaptability.

In our project the AlgaePPO class is designed to integrate PPO with Algae-based operations, creating a framework for adaptive training with the below specifications for the methods.

- *Population of Policies:* Multiple CNN models are initialized with random weights and trained concurrently.
- *Optimizers and Learning Rate Schedulers:* Adam optimizers are used to update model parameters with default 0.001 learning rate in case of dataset D and for dataset B there are learning rate schedulers to adjust learning rates over time. Batch size of 256 is utilized in case of dataset D for training. We have utilized dropout in case of dataset B with 0.5 as the p value.
- *Loss Function:* CrossEntropyLoss is used to compute the loss between predicted outputs and actual labels.

iv. Training and Testing:

While training the Algae and PPO fusion on MNSIT dataset during the checkpoint phase we learned after running the model over 100 iterations, we got the average accuracy of 14.5% at a very initial level of experimenting with the selected best set of parameters for the previous neural network by the Algae's method. From the results it was identified that the average fitness is increasing which hints the effectiveness of the using the PPO in fusion with the CNN for training the model.

Hence proceeding ahead with OU-ISIR Treadmill dataset the training process involved below specifications:

- *Training Episodes:* We have run 50 episodes in case of dataset B model for the model to learn and adapt and 10 episodes in case of dataset D since the dataset B has higher volume of data the model takes more time to extract the important features from the data.
- *Evaluation and Weight Update:* After each episode, the models are evaluated for fitness, and their weights are updated based on their performance like loss, optimizers and evolution parameters. The code iterates over the training data loader, calculating the loss for each batch, performing backpropagation, and updating the model weights.
- *Crossover and Mutation:* The crossover method performs crossover between two parent models' weights to create a child model's weights. The mutate method introduces random mutations to the weights of a model and it is adjusted based on specific episodic frequency. Every 5 episodes, crossover and mutation operations are applied to introduce variability and evolve the population for the dataset B.

In case of dataset D, the number of CNN models used is 2 for analyzing the NAC patterns with the gait data.

III. Experimental Results:

The experiment involved training the population of models for 100 episodes. The following outcomes were observed:

DATASET B

In our dataset B we have mainly 4 categories ("40_Percent", "70_Percent", "80_Percent", "100_Percent") with corresponding labels (0, 1, 2, 3) representing the various classes for the dataset. Firstly, the method runs three different models on each episode and fetches the results, after every 5 episodes the evolve() method implements a simple crossover mechanism, selecting two parent models to generate child weights through a combination of their parameters. The mutation process introduces variability, with a 10% mutation probability, to prevent overfitting and maintain robustness.

Loss (Training phase):

```
Population evolved
Model 1 Episode 47 Loss: 2.7010516598820686 Accuracy: 99.92492492492492
Model 2 Episode 47 Loss: 3.3378382777726756e-06 Accuracy: 100.0
Model 3 Episode 47 Loss: 0.29772123694419683 Accuracy: 100.0
Model 1 Episode 48 Loss: 158.58965957391774 Accuracy: 99.54954954954955
Model 2 Episode 48 Loss: 23.985185623168945 Accuracy: 99.92492492492492
Model 3 Episode 48 Loss: 39.928361892700195 Accuracy: 99.84984984984985
Model 1 Episode 49 Loss: 103.61955262720585 Accuracy: 99.69969969969969
Model 2 Episode 49 Loss: 0.03169925451765998 Accuracy: 100.0
Model 3 Episode 49 Loss: 189.97468674182892 Accuracy: 99.62462462462463
Model 1 Episode 50 Loss: 24.980698691913858 Accuracy: 99.84984984984985
Model 2 Episode 50 Loss: 1.864819049835205 Accuracy: 99.84984984984985
Model 3 Episode 50 Loss: 11.509237289428711 Accuracy: 99.92492492492492
```

From the above results from the model, we can observe that the training loss is still fluctuating but however the algae method helps to tackle this by evolving a better model. The low values of losses explain the robust and effective gait pattern segmentation.

Test Accuracy:

The training results show a steady increase in accuracy over 50 episodes, with the evolution process contributing to improved performance. Early episodes demonstrate moderate accuracy, while later episodes consistently achieve high accuracy (above 99%).

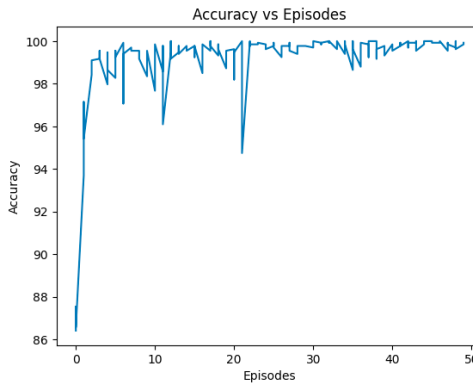


Fig – Accuracy vs Episodes for dataset D

The evaluation process on a test dataset indicates high accuracy across the models, with most achieving near perfect or perfect accuracy on the test set. This outcome suggests effective training and robust model performance.

DATASET D

While measuring the NAC values, the model returned the optimal shift value of 69 and maximum NAC value near to 0.975. The optimal shift value represents the number of frames where the Normalized AutoCorrelation (NAC) between gait silhouettes is maximized. An optimal shift of 69 frames suggests that the gait cycle repeats approximately every 69 frames, given the conditions of your analysis. This shift is the most consistent period over which the gait patterns exhibit the least variation (or the highest similarity) when compared across different periods within the gait sequence.

The NAC value near 1 (like 0.9748) indicates a very high degree of similarity and consistency in the gait pattern at the optimal shift. A high NAC value implies that the subject's gait is stable and consistent over time, which is often a sign of a healthy and regular gait. This could be particularly useful in medical settings where deviations from such patterns may indicate health issues.

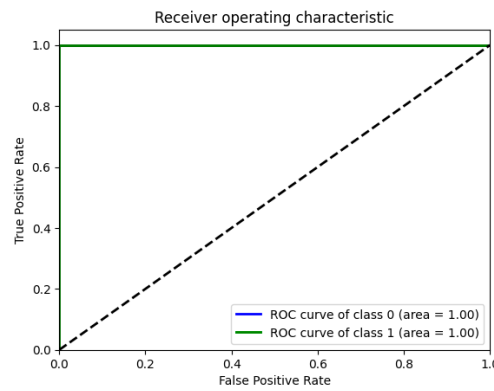


Fig – ROC plot for 2 output classes

Loss (Training phase): Our model training loss decreases significantly from epoch 1 to epoch 10, indicating effective model learning and convergence. The loss decreases from 1.2054 to 0.0045, suggesting that the model has learned to minimize the error during training.

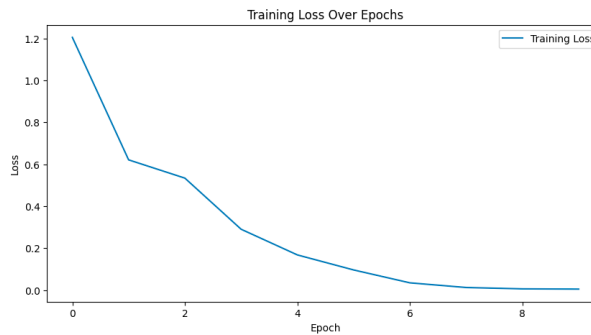


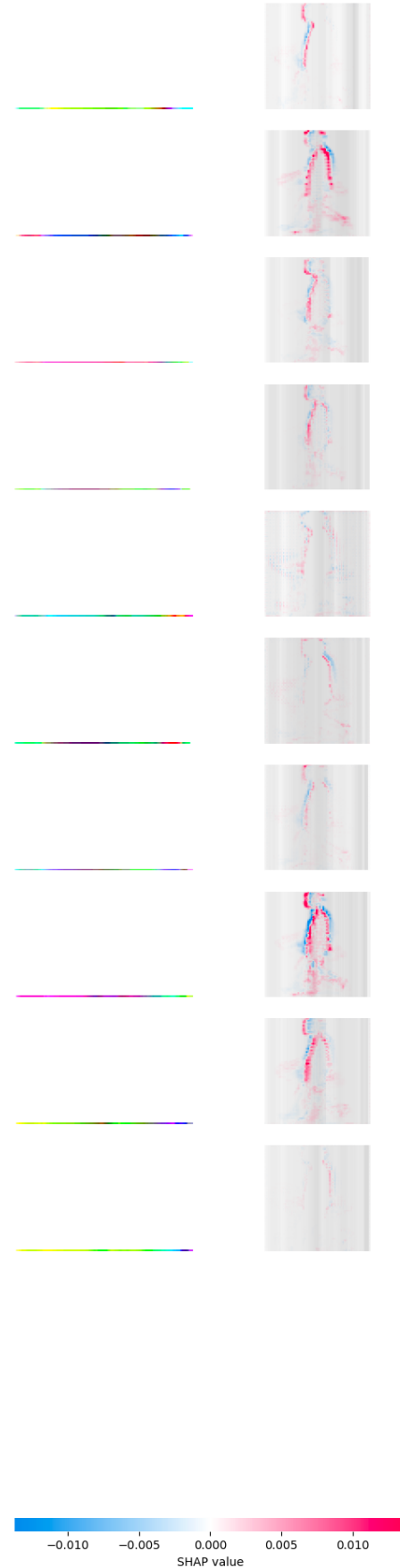
Fig – Training loss for dataset D

Test Accuracy: The test accuracy is improving consistently over the epochs, starting from 0.3505 in epoch 1 and reaching 1.0000 in epochs 6, 7, 8, and 10. This pattern indicates that the model has successfully learned from the training data and is generalizing well to the test data.

Validation and Overfitting: The high (~ 1) test accuracy raise concerns about overfitting. A sharp increase in accuracy could mean the model has memorized the training data rather than generalizing patterns. It is crucial to validate the model with additional unseen data to ensure robustness.

We have also done SHAP analysis (Shapley Additive exPlanations) to decipher the influence of different regions within gait image data on the predictions made by a convolutional neural network (CNN). The color-coded SHAP values, where red indicates positive influence and blue indicates negative, allowed for a detailed inspection of how specific segments of the gait cycle—such as heel strike, toe-off, and mid-stance—affect the model's decision-making process.

Through this visualization on the right, we can conclude that it can identify key patterns and anomalies in walking styles that are critical for diagnosing health conditions and evaluating rehabilitation progress. This insight is invaluable as it directs attention to specific features of the gait that are most relevant, thereby guiding further data collection, feature engineering, and refinement of the CNN architecture.



IV. References:

1. https://git-disl.github.io/GTDLBench/datasets/mnist_datasets/
2. Y. Makihara, H. Mannami, A. Tsuji, M.A. Hossain, K. Sugiura, A. Mori, and Y. Yagi, ``The OU-ISIR Gait Database Comprising the Treadmill Dataset," IPSJ Trans. on Computer Vision and Applications, Vol. 4, pp. 53-62, Apr. 2012.
3. Reward-Adaptive Reinforcement Learning: Dynamic Policy Gradient Optimization for Bipedal Locomotion
4. A Data-Driven Reinforcement Learning Solution Framework for Optimal and Adaptive Personalization of a Hip Exoskeleton
5. Static Standing Balance with Musculoskeletal Models Using PPO With Reward Shaping
6. A Novel Deep Reinforcement Learning Based Framework for Gait Adjustment
7. Challenges with reinforcement learning in prosthesis

V. Contributions:

Team Member	Project Part	Contribution (%)
Gautam Galada	Research on Gait patterns, PPO, Algae, NAC estimation, Data Understanding, Code base pipelining.	33.33
Henish Viradiya	Implementation of PPO + Algae fusion method and performed hyperparameter tuning. Built data loaders.	33.33
Siddharth Haveliwala	Building of CNN and configuring Algae + PPO fusion method. Research documentation.	33.33