[Team 80] Proj-C1: Terrain Identification from Time Series Data

Sukruthi Modem (smodem), Srihitha Reddy Kaalam (skaalam), Muthya Ambati (mambati)

I. METHODOLOGY

It is natural for humans to acquire the ability to walk efficiently, stably, adapt to the environment and be strong. However, people with lower limb amputations lose this capability, and they often rely on prosthetic devices to regain basic walking functionality. To improve the comfort and safety of those with lower-limb robotic prosthetics, incorporating context awareness is useful [1]. This study aims to create a terrain identification system that utilizes IMU data from the lower limb to enhance the functionality of a prosthetic leg. Although visual and inertial sensors are usually utilized, the study seeks to investigate the feasibility of terrain identification without visual data. By obtaining such information, the control of a robotized prosthetic leg can be adjusted to adapt to changes in its surroundings.

The dataset consists of IMU data collected from a sensor on the leg of the participant at several sessions from 6 different subjects. The annotations about the terrain type are obtained from a synchronized data stream which serve as labels. A total of 1345061-time stamps is used for training the model and it is observed that the dataset is highly skewed towards solid ground of class 0 and few data with label-going down the stairs of class 1. The distribution is as shown in Fig.1.

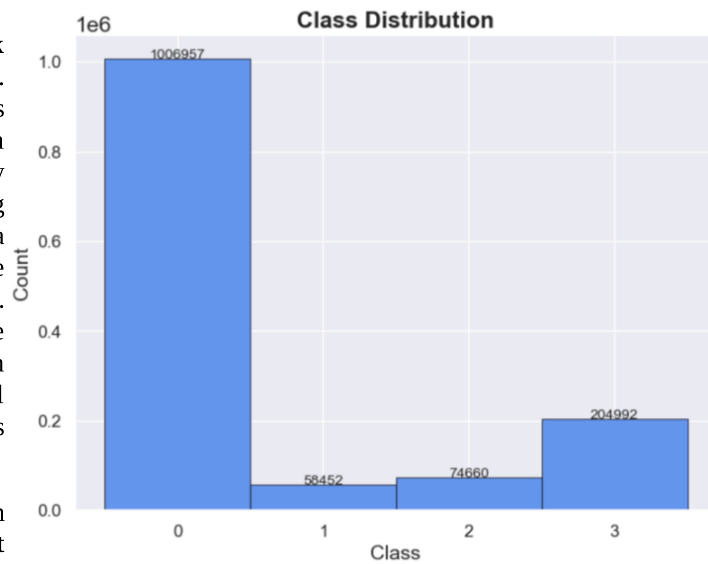

Fig. 1. Class Histogram

The model developed is a convolutional neural network and it consists of one Conv1D layer with 128 filters and a kernel size of 2. The ReLU activation function [2] is used to introduce non-linearity into the model, followed by a max-pooling layer. Two dropout layers and two dense layers are added with softmax activation function which is used to convert the outputs into probabilities, so that the model can make predictions with the highest probability representing the most likely class as determined by the model. The *kernel_regularizer* is used to reduce the overfitting of the model by adding a penalty term to the loss function.

II. MODEL TRAINING AND SELECTION

To balance the class distribution, random under-sampling is performed on the input data and for the model architecture, a Convolutional Neural Network (CNN) is used for the classification task by training the model until no further loss or accuracy improvements could be obtained. The hyperparameters tuning is performed using a grid search with three parameters: dropout rate, L1 regularization value, and L2 regularization value.

A. *Model Training*

The CNN model is trained using the Keras API with TensorFlow backend. The data set is split into training and validation sets in the ratio 3:1. A grid search is implemented to find the optimal combination of hyperparameters that maximizes the validation accuracy.

It is run on a validation set consisting of 25% of the total data, and the best performing combination of hyperparameters is selected based on the mean validation accuracy over the last 5 epochs of training. The dropout rate [3] is varied, L1 regularization [4], and L2 regularization [5] to create a total of 48 possible combinations of hyperparameters. The hyperparameters of the best model were dropout rate of 0.01, L1 regularization of 0.0156, and L2 regularization of 0.0156.

The model is then trained using these hyperparameters for 35 epochs on the training set, with a learning rate of 0.1 and Adam optimizer with default settings. The validation set is used to monitor the training process and prevent overfitting. We used early stopping to terminate the training process if the validation accuracy did not improve for 10 epochs.

The model achieved a validation accuracy of 0.9128 and is saved for later use in testing.

B. *Model Selection*

For the project, sensor data from wearable devices is used to capture human motion. Although the data is not strictly image data, it can be considered as a series of images with multiple channels such as the X, Y, and Z accelerometer and gyroscope readings at each time step.

Considering the spatial nature of the sensor data, a CNN is selected for the classification task. CNNs enable the model to learn relevant features from the raw sensor data automatically without the need for manual feature engineering. Furthermore, CNNs have been demonstrated to be effective for a wide range of classification tasks, including those that involve time series data. Adam optimizer [6] is used as it learns the weights adaptively leading to fast and efficient performance of the models. For the classification tasks, cross entropy loss function [7] is used over others because it penalized incorrect predictions more heavily than other loss functions like mean squared error (MSE). The cross-entropy loss is calculated as follows:

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i)\,, for\ n\ classes$$

Where t_i is the truth label and p_i is the softmax probability for the i^{th} class.

The optimized model is achieved with a dropout rate of 0.01, L2 regularization strength of 0.0039, and a learning rate of 0.1. This model achieved an average validation accuracy of 83.5% over the last 5 epochs of training. Consequently, the model was trained with these hyperparameters for 35 epochs, resulting in a validation accuracy of 87.2%.

Overall, the results suggest that the best model captured important features in the data and generalized well to the validation set. However, further experimentation with hyperparameters and model architectures could potentially lead to even better performance.

Fig. 4. Model Accuracy

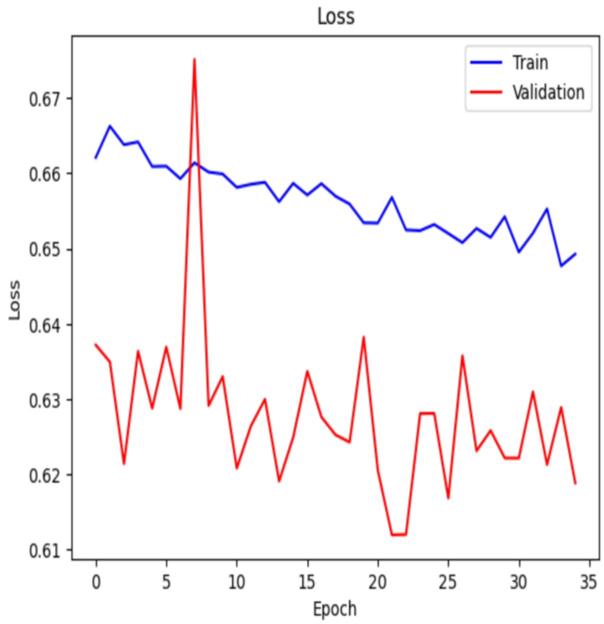

Fig. 4. Model Loss

III. Evaluation

Based on the confusion matrix shown in TABLE I, it can be observed that the model had some difficulty in correctly identifying the samples of the first and fourth classes. Class 0 has a relatively low recall score of 0.63, indicating that the model failed to correctly identify 37% of the samples from this class. Class 3, on the other hand, has a low precision score of 0.62, indicating that the model incorrectly identified 38% of the samples from this class.

TABLE I
CONFUSION MATRIX

Actual \ Predicted	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	1115	104	68	496
Actual 1	98	1608	28	109
Actual 2	47	41	1697	33
Actual 3	689	114	33	1026

To better evaluate the performance of this model, we examined the precision, recall, and the F1 scores of each model classification type. As the class is imbalanced (Fig 1) F1 score is an important evaluation metric for our problem statement. Tables II and I show how well the model performed for each class.

Looking at the overall model evaluation metrics, the model achieved an accuracy score of 0.75, indicating that it correctly predicted the class labels for 75% of the samples. The weighted average of the precision, recall and f1-score is 0.75, indicating that the model's performance is consistent across all the classes. However, the macro-average f1-score of 0.74 indicates that the model's performance varies across the classes, with some classes having better performance than others. Overall, the model's performance is decent but there is certainly room for improvement.

TABLE II

	precision	recall	f1-score	support
0.0	0.57	0.63	0.60	1783
1.0	0.86	0.87	0.87	1843
2.0	0.93	0.93	0.93	1818
3.0	0.62	0.55	0.58	1862
accuracy			0.75	7306
macro avg	0.74	0.75	0.74	7306
weighted avg	0.75	0.75	0.74	7306

IV. Reference

[1] B. Zhong, R. L. d. Silva, M. Li, H. Huang and E. Lobaton, "Environmental Context Prediction for Lower Limb Prostheses With Uncertainty Quantification," in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 2, pp. 458-470, April 2021, doi: 10.1109/TASE.2020.2993399.

[2] Agarap, A.F., 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[3] Sanjar, K., Rehman, A., Paul, A. and JeongHong, K., 2020, December. Weight dropout for preventing neural networks from overfitting. In *2020 8th International Conference on Orange Technology (ICOT)* (pp. 1-4). IEEE.

[4] Murugan, P. and Durairaj, S., 2017. Regularization and optimization strategies in deep convolutional neural network. *arXiv preprint arXiv:1712.04711*.

[5] Shi, G., Zhang, J., Li, H. and Wang, C., 2019. Enhance the performance of deep neural networks via L2 regularization on the input of activations. *Neural Processing Letters*, *50*, pp.57-75.

[6] Bera, S. and Shrivastava, V.K., 2020. Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing*, *41*(7), pp.2664-2683.

[7] Gordon-Rodriguez, E., Loaiza-Ganem, G., Pleiss, G. and Cunningham, J.P., 2020. Uses and abuses of the cross-entropy loss: Case studies in modern deep learning