Activity Recognition for Locomotion and Transportation Dataset Using Deep Learning

Chan Naseeb, Bilal Al-Saeedi
IBM

Abstract

- This challenge poses a unique opportunity to work on a broad, real-life dataset to classify transport-related activities in a user and location-independent way.
- We focused our experiments on some recent state-of-the-art deep learning architectures such as CNN, Resnet, and InceptionTime.
- Results: we were able to achieve a 79% **F1 score** on the validation dataset using the InceptionTime architecture.
- The objective of this poster is to present the technical description of the Machine Learning pipelines, the algorithms used, and the results achieved.

Data Processing Pipelines

A considerable amount of time was spent on data preprocessing. At the beginning, we performed the following data preparation steps:

- Data were resampled from 100Hz to 50Hz. We experimented with multiple sampling rates, we achieved the best results with a 50Hz sampling rate.
- Data were normalized and standardized either manually or automatically inside the models using batch normalization.
- An Imputation transformer was used for completing missing values.

After that, we performed the following feature engineering:

- 8 manual features were calculated. 5 of them are the magnitude of of the acceleration, gravity, gyroscope, linear acceleration, and magnetometer data. The other 3 features are the Euler angles which is calculated from the orientation data.
- Additionally, we used the tsfel library to autogenerate some statistical, spectral, and temporal features. .

Continued...

Finally, we performed feature selection by running a random search against multiple configurations to understand the usefulness of certain features combinations. By doing this, we found the following observations:

- Orientation, pressure, and gravity raw sensor data were not so useful but rather caused overfitting.
- Acceleration and linear acceleration data were the most influencing features in the results.
- The extra 8 manually calculated features contributed positively to the model performance.

Experiments

We focused our experiments on various deep learning architectures that are popular for solving time series classification problems such as CNN, Resnet, and InceptionTime. We summarize below the results:

- We ran multiple experiments using various convolutional neural network (CNN) architectures by tuning different hyper-parameters such as 1) number of Conv layers, 2) the number of filters in each Conv layer, and 3) the number of neurons in the hidden Dense layer. The results were not so promising due to the difficulties in the datasets.
- Using Residual Networks (ResNets), we ran around 20 experiments using of multiple variants of ResNets by changing network depth, the number of filters, and kernel sizes. The best model of this type achieved an F1 validation accuracy of 67 %. We observed an increase in the accuracy with the increase of the network depth.
- Finally, we experimented with the InceptionTime architecture which is a recent architecture specialized in timeseries classification tasks. We have performed around 260 experiments by playing with the network depth, the number of filters, and the kernel sizes. We also observed that more layers improves accuracy.

Final Selected Model

The final model used to predict the test dataset is based on the InceptionTime architecture. Figure 1 shows an overview of the complete process including the model architecture. The model consists of a total of 3 inception blocks or ensembles each contains 3 inception modules and a final dense layer with a softmax activation.

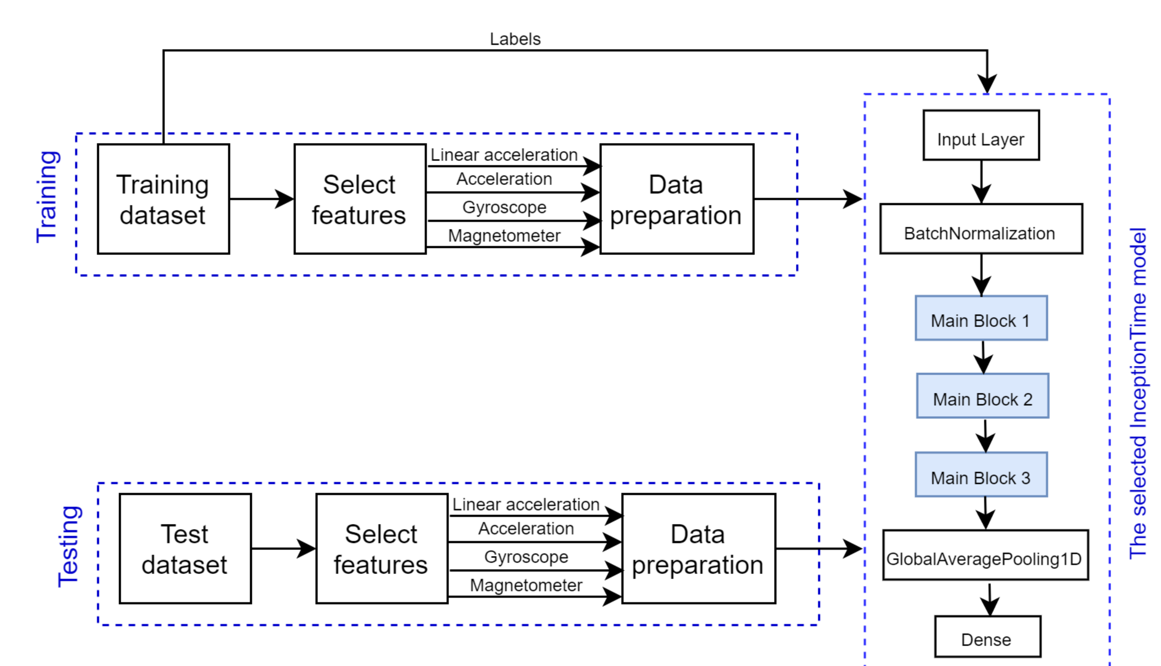

Figure 1: An overview of the complete process including data preprocessing, training, and testing phases

With this model, We were able to achieve the results shown in Table 1. It is also shown from the results that the residual skip connections helped in achieving better accuracy with deeper layers.

Accuracy	0.7935
Precision	0.8304
Recall	0.7615
F1 score	0.7939

Table 1: The final model results against validation dataset

The results also showed that the model extracted features aren't enough to distinguish between similar activities such as recognizing a user being in a train or being in a subway, as shown in Figure 2.

Continued...

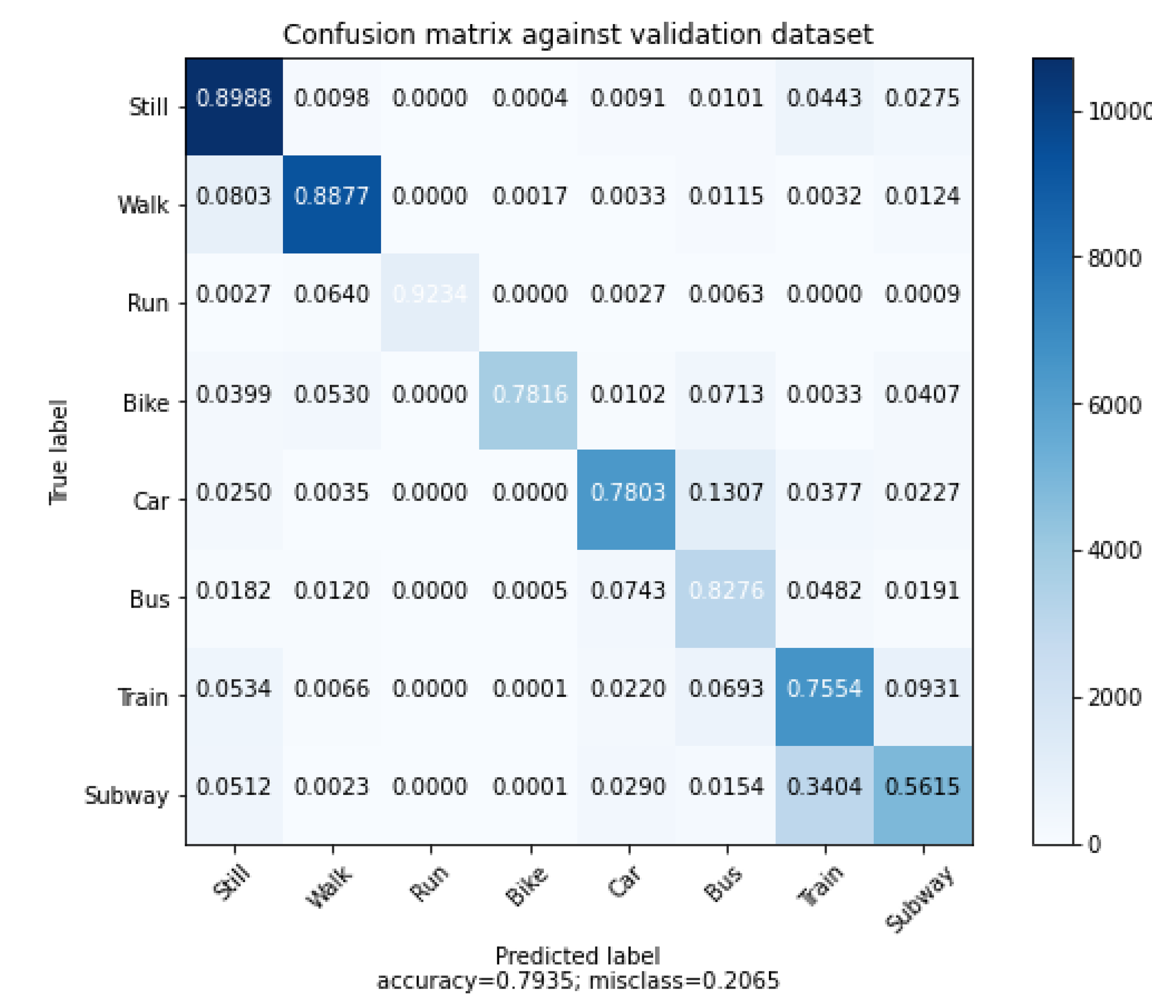

Figure 2: The confusion matrix of the final model according to the validation dataset results

Conclusion

Recognizing locomotion activities in a user and location independent manner is very challenging. Our approach was to focus on:

- Some state-of-the-art deep learning models which proved to show promising results in the activity recognition and times series classification problems.
- We mostly experimented with CNN, ResNets and InceptionTime architectures by trying multiple hyperparameters such as the network depths, the number of neurons of dense layers, convolutional filter numbers and kernel sizes.
- The best model is based on the InceptionTime architecture and achieved 79% F1 score on the validation set. This model generalized well regardless of the user or the smartphone location.
- It was shown also that data preparation, and preprocessing is a critical part to increase the model performance.