Link to published version:
https://doi.org/10.1016/j.jterra.2020.12.002

Recurrent and Convolutional Neural Networks for Deep Terrain Classification by Autonomous Robots

Fabio Vulpi[1, 2], Annalisa Milella[2], Roberto Marani[2], Giulio Reina[1*]

[1]Department of Mechanics, Mathematics & Management, Polytechnic of Bari, Via Orabona 4, 70125, Bari, Italy
[2]Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, via G. Amendola 122/O, 70126 Bari, Italy.
**Corresponding author: G. Reina, giulio.reina@poliba.it*

Abstract
The future challenge for field robots is to increase the level of autonomy towards long distance (>1 km) and duration (>1 h) applications. One of the key technologies is the ability to accurately estimate the properties of the traversed terrain to optimize onboard control strategies and energy efficient path-planning, ensuring safety and avoiding possible immobilization conditions that would lead to mission failure. Two main hypotheses are put forward in this research. The first hypothesis is that terrain can be effectively detected by relying exclusively on the measurement of quantities that pertain to the robot-ground interaction, i.e., on proprioceptive signals. Therefore, no visual or depth information is required. Then, artificial deep neural networks can provide an accurate and robust solution to the classification problem of different terrain types. Under these hypotheses, sensory signals are classified as time series directly by a Recurrent Neural Network or by a Convolutional Neural Network in the form of higher-level features or spectrograms resulting from additional processing. In both cases, results obtained from real experiments show comparable or better performance when contrasted with standard Support Vector Machine with the additional advantage of not requiring an a priori definition of the feature space.

Keywords: Autonomous robots, vehicle-terrain interaction, terrain classification, deep-learning

I. Introduction

Future generations of mobile robots will be required to explore areas, which present highly challenging mobility conditions. In order to fulfill long distance and duration missions, it will be important to be able to understand the type of traversed surface, so that adequate control and planning strategies can be implemented and areas of high risk can be properly negotiated or avoided. Throughout this paper, we generally refer to terrain classification as the task of recognizing from among a list of known possible types the one crossed (or to be crossed) based on the analysis of the sensory data available onboard the robot [1]. This problem is common to different application areas. Notable examples can be found in robotics for surface planetary exploration as described in [2], [3], [4], and [5], and precision agriculture for which terrain identification plays a key role in the fulfillment of numerous tasks such as seeding, ploughing, fertilizing or controlled traffic [6]- [7].

Early research relied on forward imaging sensing and used limited learning [8]. The visual appearance of distant terrain has been used in [9] also combining distance measurements generated by stereovision [10]- [11], radar [12] and lidar [13].

However, observation of a given terrain from a distance does not provide any information about its mechanical properties that directly impact on vehicle mobility. It is known that off-road traversability largely depends on the interaction between the robot and the terrain [14]. Dynamic ill-effects including wheel sinkage, slippage and rolling resistance are the result of this complex interplay. For example, the ground can be considered drivable based on the geometric elevation map. Yet, the robot can incur in serious risks if this terrain offers low traction properties due to high slippage and consequent lack of progression, as explained in [15].

Therefore, recently, methods that use proprioceptive sensing have been also proposed for terrain classification [16], [17], [18], [19] and [20]. The envisaged idea behind this approach is that, from a mechanical perspective, terrain category can be identified using wheels as tactile sensors that generate signals modulated by the vehicle-terrain interaction. Hence, proprioceptive data contain much information, which can be useful to characterize the terrain type.

In addition, learning-based approaches have been introduced in order to make intelligent autonomous robots adaptive to the site-specific environment [21], [22], [23], [24]. More difficult the environment, less likely expert rule-based or heuristic strategies perform well. This is the case for natural terrains that entail many challenges including variability in surface and lighting conditions, lack of structure, no prior information, and in which learning approaches may fit better. Therefore, information pertaining to wheel-terrain interaction can be extracted and, then, a mapping between proprioceptive data and the corresponding surface can be created. A learning approach fits well to this application as: i) the large number of parameters involved make a physics-based terrain model rather complex, ii) the mapping from proprioceptive input to a given terrain's mechanical properties is an extremely complicated function, for which a closed analytical form is very difficult to obtain and one possible solution is to observe the phenomenon and learn about it through training examples, iii) adaptability of the vehicle's behavior can be promoted through learning.

Following recent research trends, this paper tackles the terrain classification problem for rough terrain vehicles relying on proprioceptive sensing and deep learning. Two types of network are discussed here: the Recurrent Neural Network (RNN) and the spectrogram-based Convolutional Neural Network (CNN). RNNs can be further engineered with different structures. This work focuses on Long Short-Term Memory recurrent neural network (LSTM) and Convolutional Long Short-Term Memory recurrent neural network (C-LSTM). The performance of LSTM, C-LSTM and CNN are evaluated in comparison with the well-known state-of-the-art machine learning classifier Support Vector Machine (SVM), showing similar or improved results even with relatively similar terrains. Our hypothesis is that self-learnt features from deep learning may include temporal information from the data that are not captured by the manually designed features used by SVM.

Section II provides a survey on related recent literature, highlighting the novel contribution of this research. Materials and methods used for field validation of the proposed system are detailed in Section III. Classification strategies drawing on deep-learning theory are explained in Section IV, providing insights on practical implementation issues. Section V presents experimental results along with an analysis of the system performance and the impact of different system design parameters. Section VI wraps up the proposed system and lessons learnt.

II. Related Work

Robotic mobility takes advantage of terrain classification by predicting interaction with the soil when optimizing controls or planning paths. The interesting problem of terrain characterization was addressed by terramechanics in [25] where a particle filter algorithm is coupled with regression analysis leading to optimal terramechanics parameters predictions of single wheel experimental measures in laboratory conditions. Cohesion, angle of internal friction and shear modulus, among other descriptive features are defined with a mathematical model to characterize the terrain and estimated trough Bayesian techniques from measurable quantities such as drawbar pull, wheel input torque and sinkage.

In [6] the average values of motion resistance and slippage alongside root mean square and standard deviation of vertical acceleration were combined in a single four-element vector used as input for a proprioceptive support vector machine-based classifier. Results presented in [6] highlight the complementarity of proprioceptive and exteroceptive data especially to distinguish terrains with similar colors.

Many feature extraction algorithms used for proprioceptive-based terrain classification use Fast Fourier Transform (FFT), Discrete Fourier Transform (DFT) or Power Spectral Density (PSD). A single-wheel testbed provided with a one-axes vibration sensor normal to the ground was used in [16] to collect data that are associated trough FFT to feature vectors fed to an SVM model for terrain classification. The proprioceptive classifier proposed in [16] was also used to provide training examples to a visual terrain classifier following a self-supervised approach [29].

Optimal results and clear methods presented in [16] inspired researchers as in [5], [26] and [27] where the three axes vibrations are first transformed with FFT and then concatenated. The feature vector containing FFT results is then used to construct an SVM model in [27] and a multilayer perception deep neural network in [5].

Reference [28] integrated the information provided by a 3-axes accelerometer, a single axis vibration sensor and one microphone through a multiclassifier combination principle. Different machine learning methods were investigated for terrain classification in [28], namely k-Nearest Neighbors (kNN), Naïve Bayes (NB), SVM and Random Forest (RF), and different feature extraction algorithms are also tested such as Modified Mel-Frequency Cepstral Coefficient (MMFCC), FFT, Zero Crossing Rate (ZCR), Short Time Energy (STE), entropy, spectral centroid, spectral roll-off and spectral flux. Results presented in [28] suggest SVM is better suited for online terrain classification compared to the other tested algorithms. Accuracy values for SVM in [28] are between 80% and 90% for signals collected at relatively low speed (0.4 m/s) depending on the number of features selected.

According to [5], [26], [27] and [28] rover's speed has a significant influence on proprioceptive-based terrain classification accuracy. Lower travelling speed corresponds to lower accuracy values because the signal to noise ratio is lower and significant frequency features contained in signals become undetectable by machine learning algorithms. Reaching good classification performance at low speed values is of great importance for proprioceptive-based terrain classification models, since it might be dangerous to travel at high speed on a terrain that is possibly unknown and that the rover is trying to identify.

Among deep learning approaches, Long Short-Term Memory recurrent neural networks (LSTM) have been shown to be effective for prediction and classification of time sequences [30]. These particular units of the recurrent network are capable of retraining information for both short and long term and are reasonably the best choice for applications like speech recognition as in [31], [32].

Various architectures of RNNs are tested in [4] for the construction of a visual terrain classification model outperforming standard visual approaches in dealing with issues such as illumination changes or motion blur. A road surface classification model based on LSTM was also proposed in [34] where 14 sensors output sequences are used together to recognize flat road, sinusoidal road, potholes and bumps showing optimal results. Vehicle-terrain interaction sound was used in [17] to train and test a deep spatiotemporal terrain classification model. The rover used in [17] is therefore equipped with a shotgun microphone and LSTM is integrated with a convoluted neural network to recognize nine different types of terrain taking also into account adverse acoustic conditions. Spectrograms of acoustic signals are computed and a sequence of features is derived from them to be further classified by an LSTM. Results of [17] highlight the importance of temporal dynamics for the terrain-classification task.

Research in [3] compared the performance of several machine learning algorithms, including Support Vector Machine (SVM), Multilayer Perceptron (MLP), Deep Neural Networks (DNN) and Convolutional Neural Network (CNN), for both terrain estimation and slip detection. The paper investigates both vision-based and proprioceptive-based classification with filtered and unfiltered data. The image dataset was provided by the NASA's Planetary Data System and the University of Almería's Fitorobot, while the proprioceptive data were collected from an MIT single wheel test-bed equipped with a torque sensor, an IMU and a displacement sensor.

The absolute value of the wheel torque together with variance of pitch, longitudinal and vertical accelerations were used to compose the four-element feature vector defined by the authors' expert knowledge and train all models for comparison. Reference [3] highlights the advantages of deep learning algorithms for being able to reach good performance with raw data without necessarily requiring an expert feature extraction.

Researchers in [20] trained a feed-forward Neural Network (NN) for each one of 15 different sensors (3-axes gyros and accelerometers, two motor current sensors and two voltage sensors, one ultrasonic and one infrared range sensors, one microphone and one wheel encoder). The authors compared the performance of NNs for different sensor modalities for DFT-based classification of five different terrain types (gravel, grass, sand, pavement, and dirt). Results of [20] show that certain sensors are better suited for identifying certain types of terrain and suggest that better performance can be achieved by combining multiple sensor modalities.

A recent body of research has been devoted to the classification of terrains in the context of legged robots [35]. In [36], torque measurements taken form sensors attached on robot legs were passed through RNN and LSTM to capture temporal data. Classification of terrain class (high-friction, low-friction, deformable, granular) was discussed in [37] for the SAIL-R legged robot using an SVM with 39 hand designed features extracted from the ground reaction forces and motor speed. Finally, an unsupervised learning based on the Pitman-Yor process was also presented in [38].

The main contributions of this research to the terrain classification problem for off-road autonomous wheeled robots refer to

- Reliance on proprioceptive signals that we assume to bring distinctive traits of the terrain as they directly generate from the physical vehicle-ground interaction. Therefore, training a classifier on these signals allows one to identify directly the impact of different terrain types on the vehicle mobility. While proprioceptive sensing has been already proposed generally using one single sensor, here different sensor modalities, e.g. wheel velocities and torques, and inertial measurements, are combined into a single model to achieve robust classification even for terrains with similar properties.
- Use of deep learning to solve the terrain classification problem. While standard machine learning algorithms have been already demonstrated, this research discusses in detail the adoption of deep learning-based solutions. Three different embodiments of deep learning classifiers are presented, namely, Convolutional Neural Network (CNN), Long Short-Term Memory recurrent neural network (LSTM), and Long Short-Term Convolutional recurrent neural network (C-LSTM). While the two RNNs accept as input directly the time series of the signals, CNN operates through signal spectrogram. Therefore, a direct comparison between two different approaches that use either raw signal or their spectral content is performed.
- A CNN-based terrain classifier is presented using a novel approach where heterogenous input data are assembled in a multi-dimensional spectrogram. Previous attempts in this direction have generally dealt with a single-channel spectrogram, as for example in [17] or [20]. In contrast, the proposed approach can be extended to any sensor combination without limitation on the number of measurements and their sampling frequency. By combining different sensor modalities, classification performance is significantly improved.
- Parametric analysis of the proposed deep learning classifiers to evaluate the influence of the design parameters including the temporal "listening" window.
- Direct comparison between the proposed deep terrain classifiers with the existing benchmark, e.g. SVM. Such a comparison is seldom discussed in the literature. We expect that the self-learned features found via deep learning may capture the temporal patterns from the data, which are not expressed by the hand-designed feature space adopted by SVM.

III. Materials and Methods

III.A. System architecture and setup

The proposed terrain classifiers are tested in the field using the experimental testbed Husky that is shown in Figure 1. Husky has a length of 0.7 m and a width of 0.5 m, and its proprioceptive sensor suite is composed of encoders to measure wheel angular velocity, electrical current sensors that provide an indirect measurement of tire torque, and an XSENS MTi-300 inertial sensor module that tracks linear accelerations and angular rates. The sensor suite is completed by an exteroceptive counterpart that includes a stereo-camera, an outdoor laser rangefinder and a GPS that, however, have not been logged for this specific research.

Therefore, two groups of measurements can be logged by the robot during operations: the IMU data sampled at 50 Hz and the wheel service data that we refer to as the PRO data sampled at 15 Hz. The IMU data contain the measurements of the 3-axes gyroscope and accelerometer inertial unit. The PRO data instead consist of the electric driving currents of the left and right motors and the left and right wheel velocities derived from the incremental encoder readings.

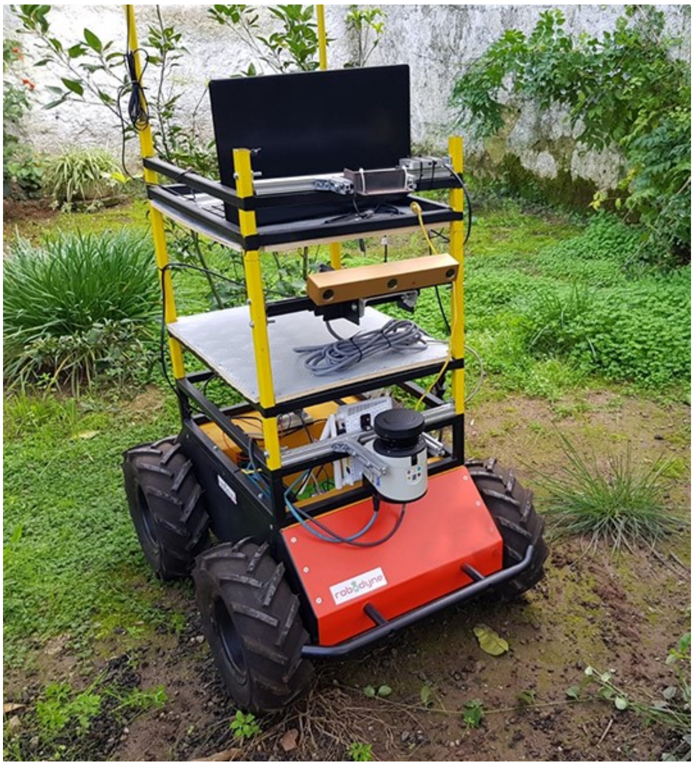

Figure 1: The experimental test bed used for the system validation

III.B. Dataset description

Sensory data are gathered as Husky traverses four different types of terrain: concrete, dirt road, unploughed, and ploughed, showed in Table 1. Extraction of sample S_i from collected sensor's signals is performed after a partitioning procedure of the data to ensure generality of the models and avoid overfitting, according to the pipeline explained in Figure 2. A partition window (PW) of 5 s is selected and consecutive clips of length PW are extracted from each signal and used randomly for testing (green in Figure 2) and training (orange in Figure 2) sets following a k-fold cross validation. To augment the number of available data, clips of length PW are then further windowed using a moving window of MW seconds and a stride of ST seconds. In section V.A different values of MW are tested and the impact on model's performance is investigated. The stride ST is instead adjusted for each terrain type to have a comparable number of samples for each terrain and avoid biasing model's prediction towards those terrains that would present more samples than others in the dataset. Table 1 contains in the third column the total number of windowed 5 s long recordings available for each terrain type.

Table 1:Terrain types and corresponding windowed data

Terrain Type	Terrain sample image	Number of 5-s-long windows
Concrete		24
Dirt Road		16
Ploughed		60
Unploughed		56

The generality of the presented model performance is achieved through *k*-fold ($k = 5$) cross validation. Each fold contains 80% of the available windowed data in the training set whereas the remaining 20% is used as a testing set. For each partition, the training set is used to train the classifiers that are afterwards tested on the corresponding testing set. For each fold, performance metrics, including accuracy, specificity, precision and F1-score are computed on the respective testing set.

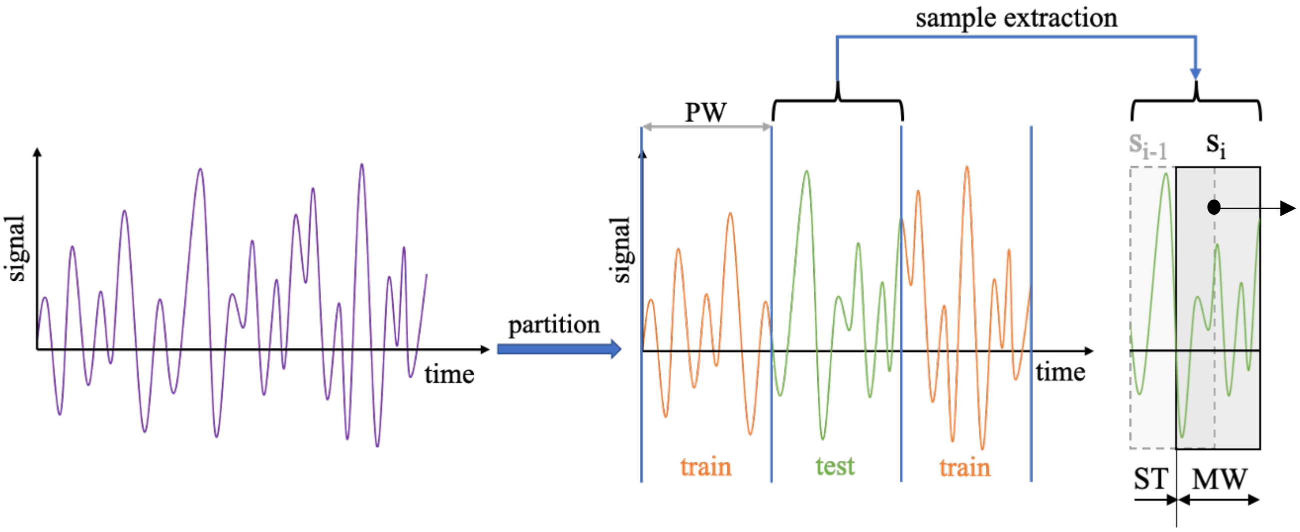

Figure 2: Pipeline of signal partition and sample S_i extraction, with PW the time window used for partitioning, MW the time window used for augmentation and ST the corresponding stride.

IV. Classification algorithms

This section provides a brief overview of the classification algorithms investigated in this research. Properties and implementation issues of each solution are discussed. In the following paragraphs, Nc refers to the number of channels that correspond to the available sensors ($Nc = 10$ in this study). PRO data and IMU data are sampled, respectively, at 15 Hz and 50 Hz. Therefore, down-sampling is performed for the training of RNNs, whereas padding is used for CNN. Four statistical moments of signals (mean, standard deviation, skewness and kurtosis) are computed and appended to form the input feature vector of SVM, instead.

IV.A. Convolutional Neural Network

The input to the Multichannel Spectrogram-based Convolutional Neural Network model (CNN) is defined as an image of Nc channels. Considering that any signal in time $s(t)$ for $t \in [t_0\,, t_1]$ sampled with a sampling frequency of sf can be expressed as a N-tuple $x \in \mathbb{R}^N$ with $N = \lfloor(t_1 - \, t_0) \cdot sf\rfloor$, with $\lfloor\cdot\rfloor$ indicating the extraction of the integer part of a number. Equation (1) computes the discrete Fourier transform $\hat{x} \in \mathbb{C}^N$ and equation (2) defines the matrix operator $\bar{\mathrm{F}}$.

$$\hat{x} = \frac{1}{N}\bar{F}(x) \tag{1}$$

$$\bar{F} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{j2\pi\frac{1}{N}} & e^{j2\pi\frac{2}{N}} & \cdots & e^{j2\pi\frac{N-1}{N}} \\ 1 & e^{j2\pi\frac{2}{N}} & e^{j2\pi\frac{4}{N}} & \cdots & e^{j2\pi\frac{2(N-1)}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j2\pi\frac{N-1}{N}} & e^{j2\pi\frac{2(N-1)}{N}} & \cdots & e^{j2\pi\frac{(N-1)^2}{N}} \end{bmatrix} \tag{2}$$

The discrete Fourier transform $\hat{x}$ contains the complex coefficients associated with the double-sided spectrum of N frequencies in the range $[0\,, sf]$.

Each sample is represented by a group of Nc signals, $s_c(t)$ for $t \in [t_0\,, t_0 + MW]$ and $c = 1, 2, \ldots, Nc$. Each signal $s_c(t)$ is windowed with windows of length wL seconds and overlapping wO seconds resulting in signals $s_c^w(t)$ for $t \in [t_0^w\,, t_1^w]$ and $w = 1, 2, \ldots,$Nwind. Equations (3) define the ranges of time $[t_0^w\,, t_1^w]$ while equation (4) describes computation for the number of windows Nwind.

$$\begin{cases} t_0^w = t_0 + (w-1)(wL - wO) \\ \quad t_1^w = t_0^w + wL \end{cases} \tag{3}$$

$$Nwind = \left\lfloor \frac{\lfloor MW \cdot sf\rfloor - \lfloor wL \cdot sf\rfloor}{\lfloor (wL - wO) \cdot sf\rfloor} \right\rfloor \tag{4}$$

Considering $\mathrm{x}_c^w \in \mathbb{R}^{\lfloor \mathrm{wL \cdot sf}\rfloor}$ the vector associated with signal $s_c^w(t)$, equation (1) allows for the computation of the discrete Fourier transform $\hat{\mathrm{x}}_c^w \in \mathbb{C}^{\lfloor \mathrm{wL \cdot sf}\rfloor}$. Each sample is here transformed in Nwind Fourier transform for every available channel.

Given symmetry with respect to the Nyquist frequency $\frac{sf}{2}$ of the double-sided magnitude spectrum $|\hat{\mathrm{x}}_c^w| \in \mathbb{R}^{\lfloor \mathrm{wL} \cdot sf\rfloor}$, equations (5) define the single-sided magnitude spectrum $\mathrm{M}_c^w \in \mathbb{R}^{Nfreq}$ with frequencies in the range $\left[0\,, \frac{sf}{2}\right]$ and equation (6) computes the number of frequencies $Nfreq$.

$$
\begin{aligned}
&if\ \lfloor wL\cdot sf\rfloor\ is\ even \begin{cases} (M_c^w)_f = |\hat{x}_c^w|_f & \forall\, f=\{1,Nfreq\} \\ (M_c^w)_f = 2|\hat{x}_c^w|_f & \forall\, f=\{2,3,\dots,Nfreq-1\} \end{cases} \\
&if\ \lfloor wL\cdot sf\rfloor\ is\ odd \begin{cases} (M_c^w)_f = |\hat{x}_c^w|_f & \forall\, f=\{1\} \\ (M_c^w)_f = 2|\hat{x}_c^w|_f & \forall\, f=\{2,3,\dots,Nfreq\} \end{cases}
\end{aligned} \tag{5}
$$

$$
Nfreq = \left\lceil \frac{\lfloor wL\cdot sf\rfloor + 1}{2} \right\rceil \tag{6}
$$

where the notation $\lceil\cdot\rceil$ indicates the nearest greatest integer of a number. Furthermore, equation (6) expresses a direct proportion between the sampling frequency sf and the number of frequencies $Nfreq$ for a certain window length wL. Therefore, magnitude spectra M_c^w corresponding to IMU channels providing data at $sf = 50$ Hz have more elements than those associated with PRO data sampled at $sf = 15$ Hz. Each element of the magnitude spectra M_c^w corresponding to PRO data is repeated as many times as needed to match dimensions of magnitude spectra derived from channels providing data at faster frequency, in this case IMU. Considering $Nfreq$ as computed with equation (6) for $sf = 50$ Hz, each sample represented by a group of Nc signals of MW seconds is transformed into Nc spectrograms with dimension $Nwind \times Nfreq$, and subsequentially rearranged in a single multichannel spectrogram $S \in \mathbb{R}^{Nwind \times Nfreq \times Nc}$ defined by equation (7)

$$
S_{w,f,c} = (M_c^w)_f\ \forall\, f = \{1,Nfreq\} \tag{7}
$$

The number of channels Nc corresponds to the number of magnitude spectrograms contained in a single CNN input sample. Figure 3 shows the magnitude spectra associated with four different channels, two derived from IMU data (first row) and two from PRO data (second row). To underline the different scales of channels, two color-bars are presented on the right-hand side of Figure 3, showing the values of acceleration (in $m{\cdot}s^{-2}$) and current intensity (in A).

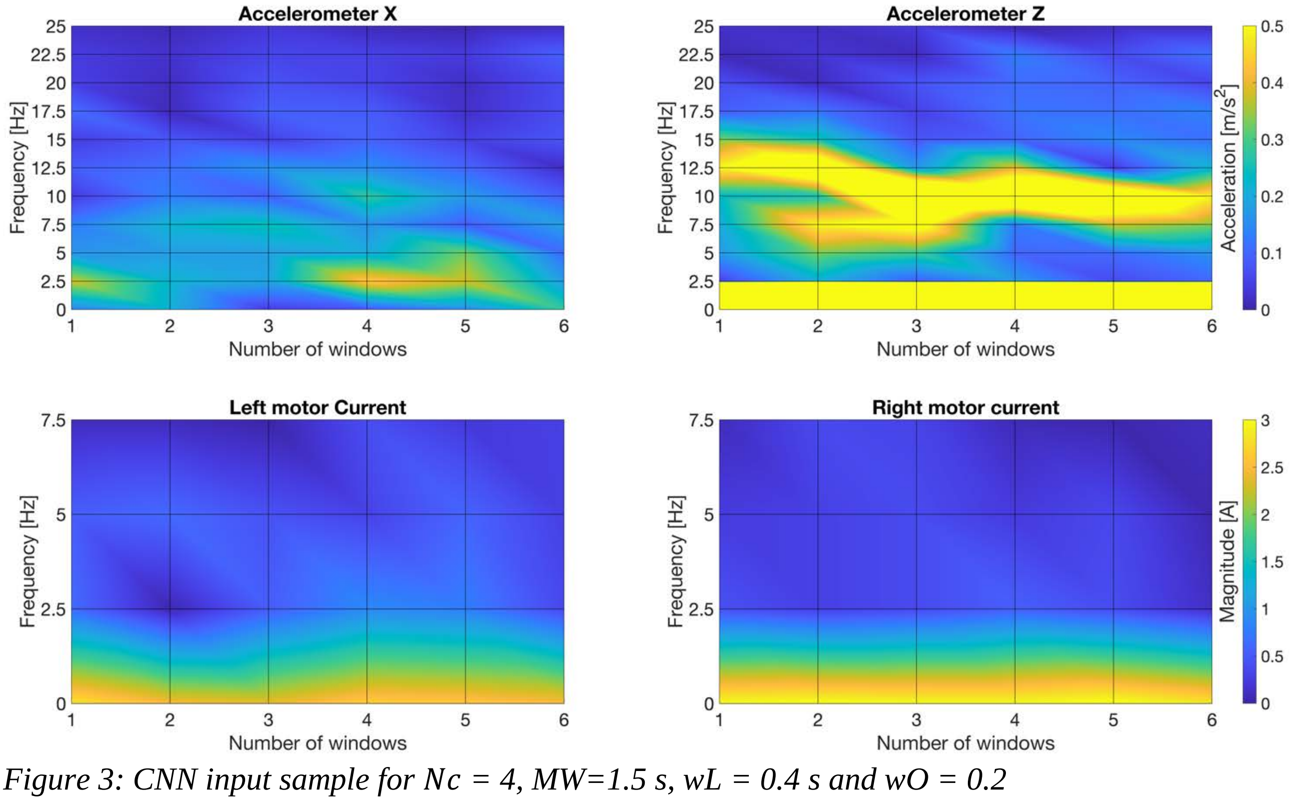

Figure 3: CNN input sample for Nc = 4, MW=1.5 s, wL = 0.4 s and wO = 0.2

The structure of the CNN model is pictorially represented in Figure 4.

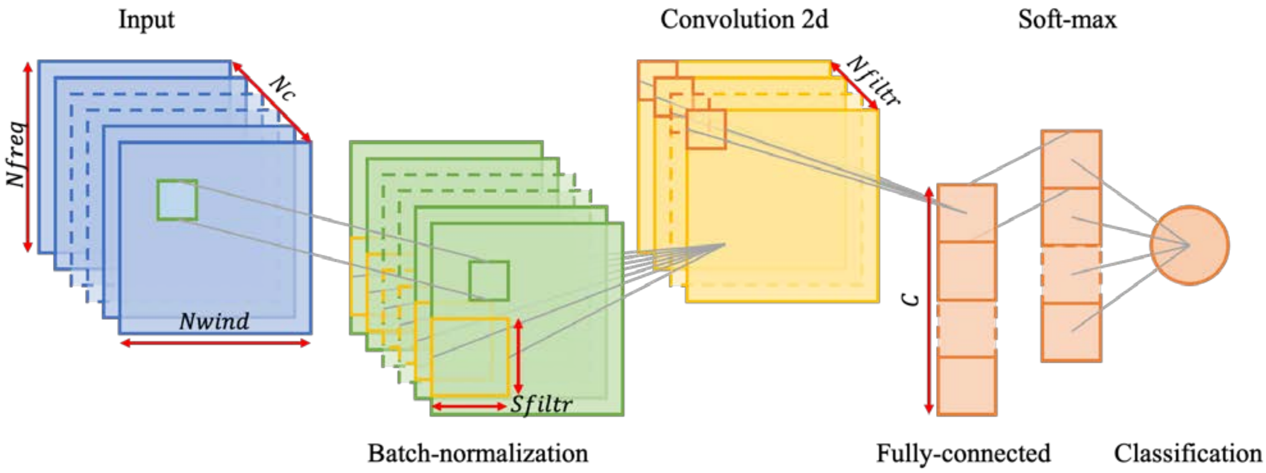

Figure 4: Convolutional Neural Network Model Structure. In this study: the number of channels is Nc=10, the size of 2D convolution filter is Sfiltr=3, the number of applied filters is Nfiltr=5, and the number of terrain classes is C=4.

The input layer is followed by the batch normalization layer that normalizes the spectrogram in each channel relatively to the training batch. The normalized multichannel spectrogram is convolved by the following two-dimensional convolution layer with same padding and a filter size of $Sfiltr \in \mathbb{N}^2$. Each filter combines all channels into a single object trying to find through the training process the combination of channels that better represents the terrain. The convolution 2D layer has two main parameters: the size of the filter to be applied ($Sfiltr$) and the number of filters ($Nfiltr$). In this context, the filter size $Sfiltr$ controls both the frequency and time span in which magnitudes across the channels are combined to form values representative for the terrain. $Nfiltr$ sets, instead, the number of filters that allow the terrain signature to be adequately mapped between control input (e.g., electric motor currents) and sensory output (e.g., the wheel speeds and IMUs measurements). Different values were tested for both $Sfiltr$ and $Nfiltr$ and eventually the best results were empirically found using 5 different squared filters of size 3. The results of the convolution process are passed to the fully-connected layer and the soft-max layer, to become the final class value as the output of the classification layer.

The discussed network structure has been designed to be simple and fast performing, following a classic image classification structure. The proposed algorithm to fuse different sensory output can integrate an arbitrary large set of signals promoting adaptability to eventual enlargement of onboard sensors. A padding procedure is also proposed to integrate signals provided at different frequencies. Convolution process is performed across time and frequency domain to better capture time-dependent aspects of the rover-terrain interaction. The results obtained concatenating spectrograms in the discussed multichannel object suggest good capability to capture differences and characterizing aspects of different terrains.

IV.B. Recurrent Neural Network

Considering Nc as the number of channels and Nh as the number of hidden units, $X_t \in \mathbb{R}^{Nc}$ and $Y_t \in \mathbb{R}^{Nh}$ are respectively the input and output vectors at time instant t of a Recurrent Neural Network (RNN). The architecture of an RNN illustrated in Figure 5 consists of a sequence of recurrent units (RUs). Each RU has the same internal structure and outputs a hidden layer vector of size $(Nh \times 1)$ represented by h_t together with the output vector. The hidden layer vector h_t is then given as input to the following RU together with the input vector X_{t+1}. With sf being the sampling frequency, X is a sequence of $\lfloor T \cdot sf \rfloor$ input vectors meanwhile Y is the corresponding sequence of output vectors. An RNN has two different operating modes, sequence to sequence and sequence to label. The sequence to sequence mode associates the sequence of input vectors to a sequence of output vectors, whereas the sequence to label mode returns the last output vector of the sequence as representative of the entire input sequence. The last output of the sequence Y, namely Y_T is in fact influenced by all the previous inputs and therefore the most representative for classifying the entire sequence. Given the number of classes C, the vector Y_T is passed to a fully-connected layer with a weight matrix of dimension $(C \times Nh)$ and bias vector of size $(C \times 1)$. The fully-connected layer is further connected with a soft-max layer and a classification layer after that, constituting the classic classification structure.

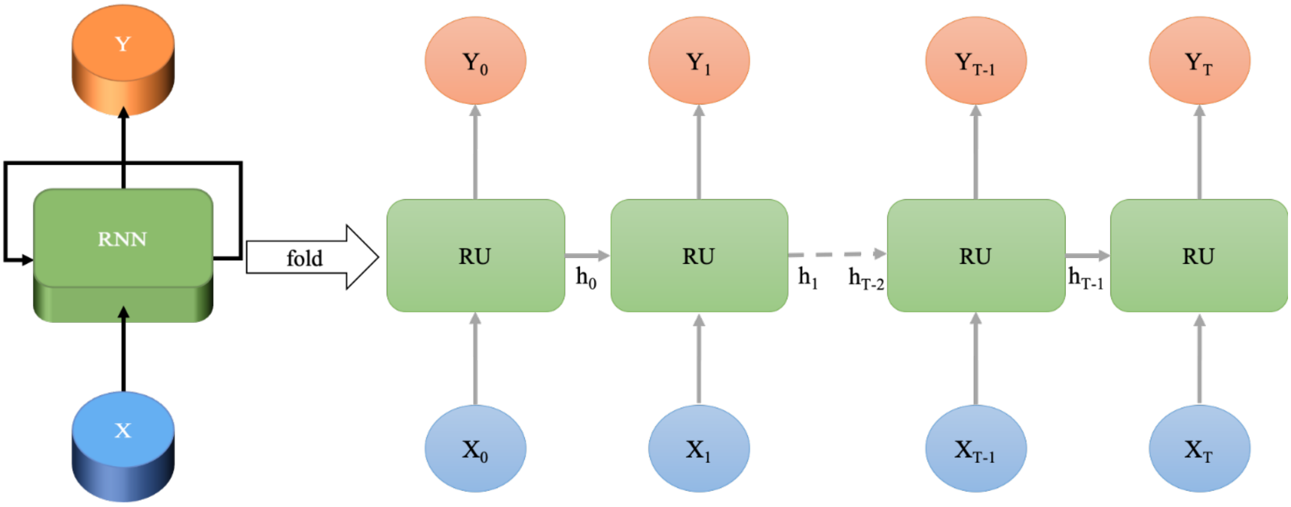

Figure 5: Recurrent Neural Network structure. In this study $X_t \in \mathbb{R}^{10}$, $Y_t \in \mathbb{R}^{15}$

The internal structure of the repeating unit (RU) of a simple RNN is showed in Figure 6 on the left representing the mathematical relationship between the vectors X_T, Y_T *and* h_T, given:

- W_h and W_y weight matrixes of size $(Nh \times Nh)$
- W_x the weight matrix of size $(Nh \times Nc)$
- b_h and b_y bias vectors of size $(Nh \times 1)$

The values retained by these weights and biases are computed during the training process searching for the optimum point of the cost function. The activation function *tanh* is the hyperbolic tangent and constitutes part of the classical recurrent unit of a simple RNN capable of solving simple sequence classification problems. These types of recurrent networks suffer from what is known in literature as problems of vanishing or exploding gradient due to the fact they are not capable of propagating information through time except for the previous time instant.

IV.C. Neural classification with memory

Long Short-Term Memory RNNs (LSTM) solve vanishing gradients problem by changing the structure of the recurrent unit, using the much more complex structure shown in Figure 6 on the right. It has been proven [30], [31] that LSTM networks are capable of blocking, forgetting and retaining information through the block gate $\tilde{C}_t$, the forget gate f_t, the input gate i_t and the memory state C_t, all vectors of dimension $(Nh \times 1)$.

The hidden layer vector together with output and input vectors have the same meaning and size as explained before and the size of weights and biases can be derived to make coherent the following operations:

- $\otimes$ row-by-column matrix product
- $\oplus$ sum between vectors
- $\odot$ Hadamard elementwise product between vectors.

The activation function σ represents the sigmoid function.

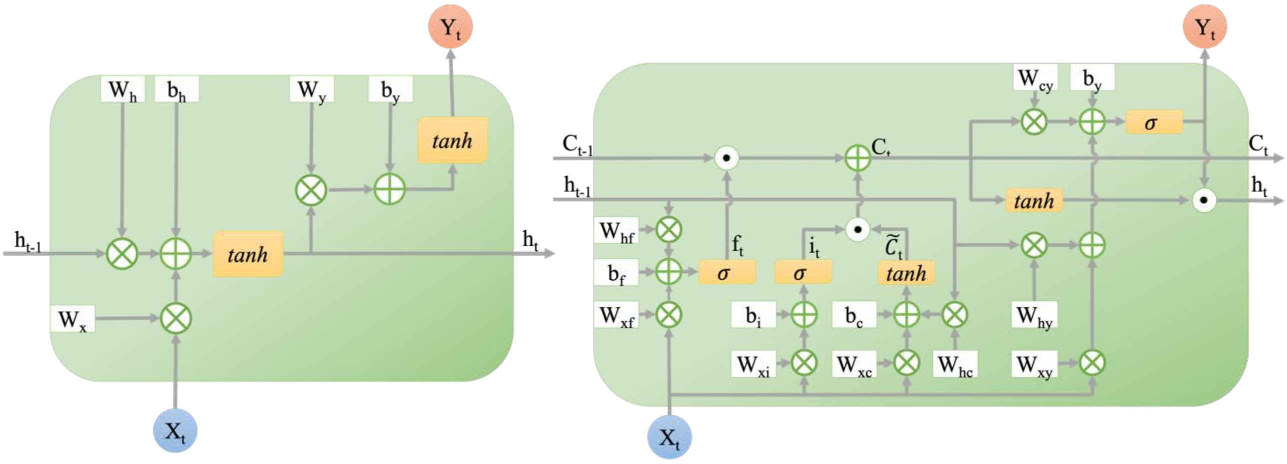

Figure 6: Internal structure of the Recurrent Unit of a simple RNN (left) and a LSTM (right)

The structure of the LSTM model developed for our study and depicted in Figure 7 has a classic Sequence to Single class value recurrent structure. The sequence input layer takes a sequence of feature vectors and passes the data to the LSTM layer whose output is the last vector of the hidden units. The LSTM layer output is then passed to a fully-connected layer with dimension C. The last combination of the two layers soft-max and classification, output the predicted single-value class associated with the sequence of feature vectors. Different number of hidden units, Nh, have been tested, and eventually increasing Nh beyond 15 did not result in any significant improvement.

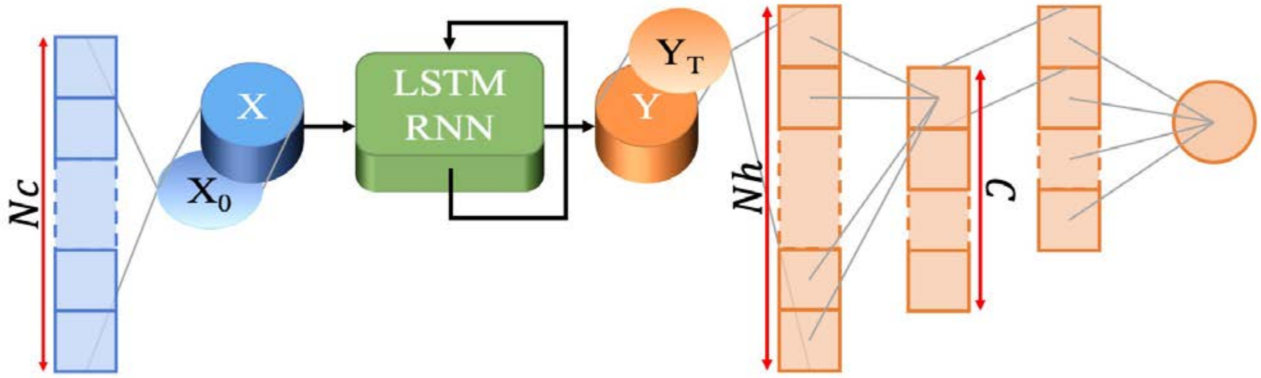

Figure 7: Long Short-Term Memory Model Structure. In this study: the number of channels is Nc=10, the number of hidden units is Nh=15, the number of terrain classes is C=4

IV.D. Convolutional Neural classification with memory

The Convoluted LSTM network (C-LSTM) takes advantage of the convolution process among features for terrain classification. The structure of this model showed in Figure 8 is like the previously explained except for the first part. The sequence input layer is in fact followed by a sequence folding layer that considers every feature vector from the sequence and passes it to a two-dimensional convolutional layer that performs the convolution process, similarly to the procedure described for the CNN model. The convolution results are normalized by the batch normalization layer and handed over to the rectified linear unit (ReLU) layer. The sequence unfolding layer then collects the results of these operations performed on every element of the feature vector sequence and together with the flatten layer they pass this new sequence to the LSTM layer. Specifically, the flatten layer reduces the extra dimensionality produced by the convolutional layer.

The following layers of the net structure are identical to the LSTM model structure. This convolution process allows the C-LSTM model to autonomously search relationships between features thus sensor outputs values for classification purposes. Five different filters have been trained and the filter size

$Sfiltr$ is set equal to the channels vector length (Nc=10) in order to let the net autonomously select a specific subset of inputs to be related and associated in a new channel.

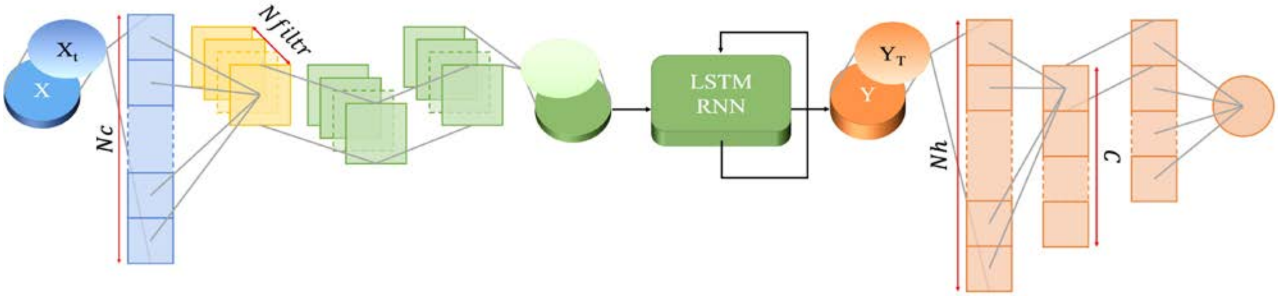

Figure 8: Convoluted Long Short-Term Memory Model Structure. In this study: the number of channels is Nc=10, the number of filers is Nfiltr=5, the number of hidden units is Nh=15 and the number of terrain classes is C=4

IV.E. Benchmarking

The results of the three proposed architectures can be compared with the well-established Support Vector Machine (SVM). An SVM is a powerful supervised machine learning methodology largely implemented for facing both regression and classification problems. SVM classifies the data attempting to separate them with hyperplanes during training and retains for testing only few samples, called support vectors. The input of an SVM model is a feature vector where the sensory data are generally composed or elaborated with expert knowledge. The SVM model used in this research has a polynomial kernel of fourth-order and "one vs one" coding. It may be further refined and tuned to optimize its performance. However, this is out of the scope of the present research and the interested readers are referred to specific Literature, e.g., [39].

SVM input vector is computed simultaneously from IMU data and PRO data without down-sampling. Mean, standard deviation, skewness and kurtosis are computed from samples of length MW for every sensory output. The sample composed of Nc sequences of measurements is therefore collapsed in a single vector of $4 \cdot Nc$ elements describing the distribution of every single sequence up to the fourth moment.

V. Results and discussion

In this Section, the performance of the three deep terrain classifiers is quantitatively evaluated using real data acquired in a commercial vineyard. First, the impact of the moving window (MW) size is investigated with a variation range from 0.5 s to 2 s every 0.1 s corresponding to patches from 25 cm up to 1 m. Then, standard classification metrics are calculated including accuracy, sensitivity, precision, and F1-scores. Normalized confusion matrixes are presented for performance evaluation corresponding to the MW that ensures the best results. Finally, computational burden and memory occupancy of all tested models are discussed to assess the feasibility for online implementation.

V.A. Parametric analysis

The accuracy among 5-fold partitions is plotted in Figure 9 for each proposed model and moving window between the range 0.5 s and 2 s. The x-axis of Figure 9 contains the sample lengths expressed in centimeter. Each sample length (SL) corresponds to the odometry distance traversed by the robot at the constant speed of 0.5 m/s for a specific moving window (MW) period.

The larger the window the better the accuracy. This can be explained when considering that increasing the MW size results in a larger informative content injected in the classifier [34]. CNN model leads to better accuracy than all other models at every MW. The mean accuracy of the CNN model is already above 80% for a sampling window of 0.5 s that corresponds to approximately only 25 cm of traversed terrain, and it sets to 90% for sampling windows larger than 1.2 s (i.e., 60 cm). Compared to CNN, SVM model correctly classifies at least 5% less of the available samples, starting with accuracy values lower than 75% for MW=0.5 s and reaching more than 80% for MW>0.9 s. For SL equal to 85 cm, CNN shows the highest accuracy of 92.8% that is 10.5% better than SVM. LSTM and C-LSTM models present similar accuracy values both always lower than SVM's. The similarity between LSTM and C-LSTM results suggests that combining instant feature values into a sequence of filtered data does not lead to better performance. For SL=85 cm, the LSTM and C-LSTM models correctly perform 16% and 14.2% worse than the CNN model, respectively. C-LSTM Accuracy values contained in Figure 9 are a good way to understand the influence of MW on each model performance. Increasing values of MW generally lead to better 5-fold model's accuracy for small values of MW whereas, for MW>1.2 s each model's accuracy fluctuates around a certain value. Mean values and standard deviation for each model accuracy corresponding to MW$\in [1.3\ s, 2\ s]$ (or SL$\in [65\ cm, 100\ cm]$) are contained in Table 2.

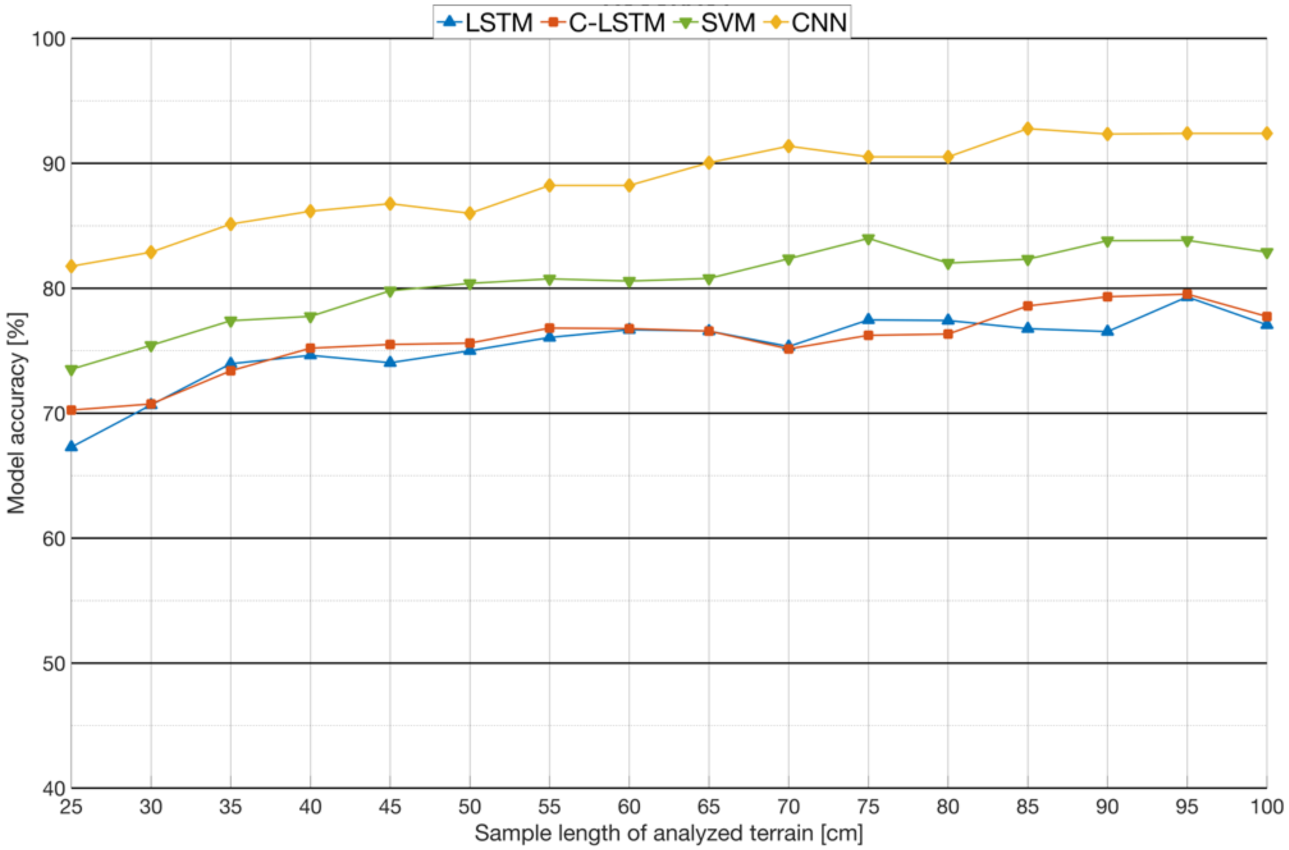

Figure 9: Accuracy of respectively LSTM (blue △), C-LSTM (red □), SVM (green ▽) and CNN (yellow ◊) models for each sample length of terrain tested corresponding to different SL values

Table 2: Mean and standard deviation for accuracy values corresponding to the MW range $[1.3\ s, 2\ s]$ *(i.e. [65 cm, 1 m])*

	LSTM	**C-LSTM**	**SVM**	**CNN**
Mean Accuracy	77.1 %	77.4%	82.8%	91.5%
Accuracy standard deviation	1.13%	1.61%	1.10%	1.08%

Standard deviations and mean accuracy values in Table 2 suggest that not only the CNN model is more accurate than the others, but it is also more stable and less subjected to variations of MW for values greater than 1.2 s (or SL>65 cm).

The distribution of sensitivity and precision is presented in Figure 10 and Figure 11. All models are perfectly able to recognize plowed terrain, independently of the MW and present difficulties in singling out concrete terrain and unploughed. Precision and sensitivity can be combined in terms of F1-score as shown in Figure 12. Sensitivity and Precision values are well balanced for all the models and all the classes; therefore, class specific performance can be evaluated analyzing F1-score in Figure 12. In particular, the CNN model outperforms SVM in classifying concrete samples, but show approximately the same F1-score values for dirt road. CNN model also presents better F1-scores for unploughed terrain where the recurrent models present difficulties. Dirt road F1-scores corresponding to recurrent models indicate better performance for the C-LSTM than LSTM.

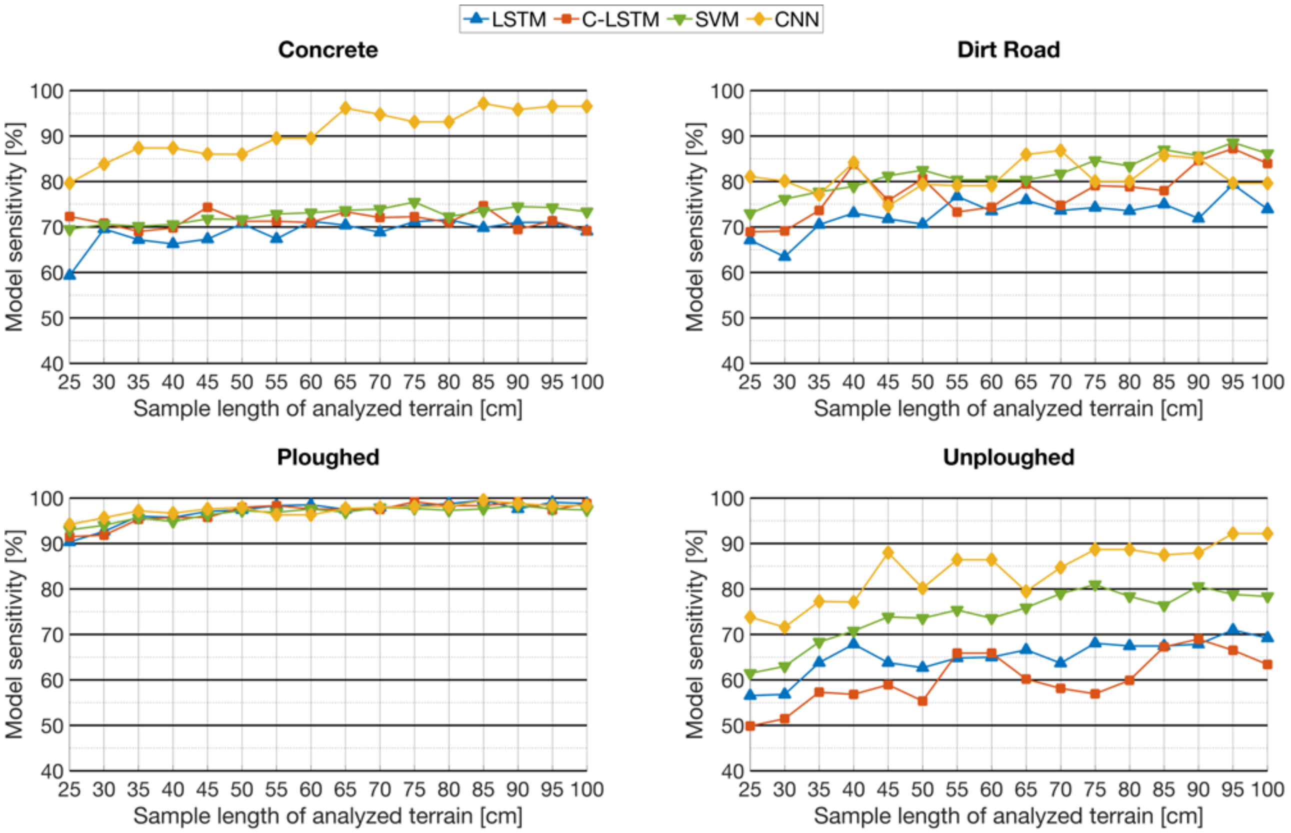

Figure 10: Sensitivity of the LSTM, C-LSTM, CNN and SVM models for each MW tested

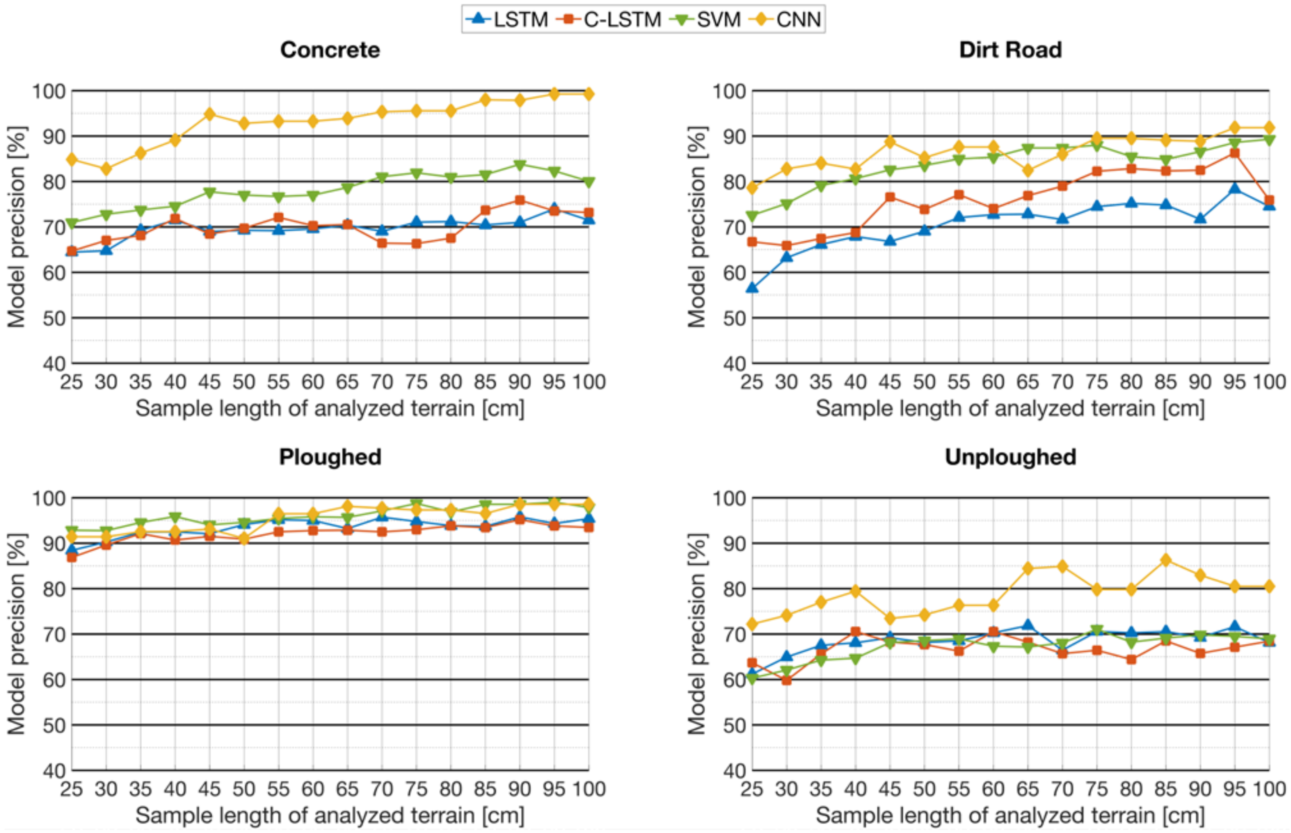

Figure 11: Precision of respectively LSTM, C-LSTM, CNN and SVM models for each MW tested

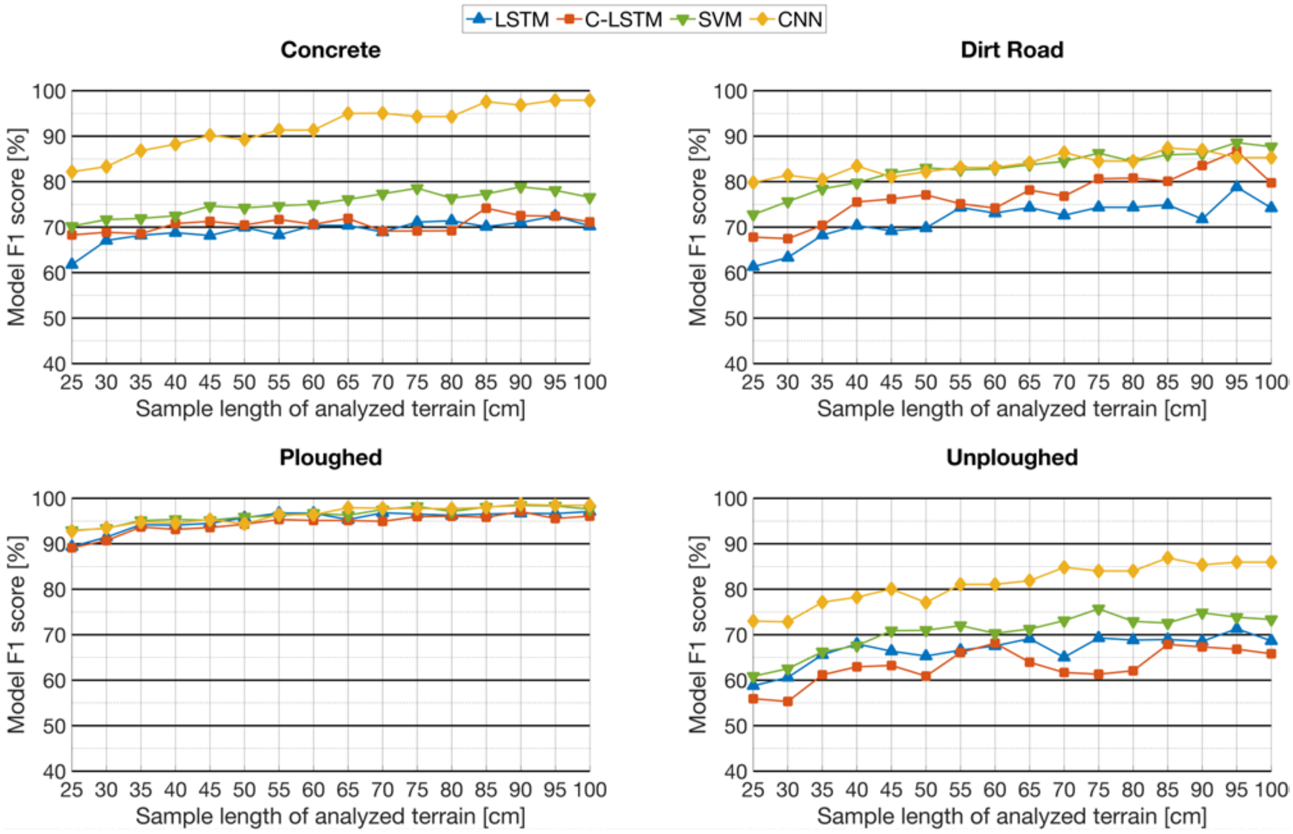

Figure 12: F1 score of respectively LSTM, C-LSTM, CNN and SVM models for each MW tested

As seen from Figure 9, the best accuracy is obtained by the CNN model for a value of the MW size of 1.7 s (or SL=85 cm), which is therefore adopted as the preferred design choice in the classifier implementation.

V.B. Performance evaluation

The normalized confusion matrixes corresponding to the 5-folds results are presented for all models in Figure 13 for a value of MW=1.7 s, that led best results and is therefore used for further considerations. Analysis of normalized confusion matrixes outlines what classes are mistaken for each other. Diagonal elements of normalized confusion matrixes contain the sensitivity value of the corresponding class. As expected, the CNN presents better overall precision and sensitivity values for all classes, showing some difficulties only in sorting out dirt-road samples and unploughed ones. Concrete, dirt road and unploughed terrain are three similar compact terrains, LSTM, C-LSTM and SVM models present difficulties to sort them out from proprioceptive data. Instead, the CNN model proved to be particularly good in recognizing concrete samples, presenting higher values for both precision and sensitivity of this class.

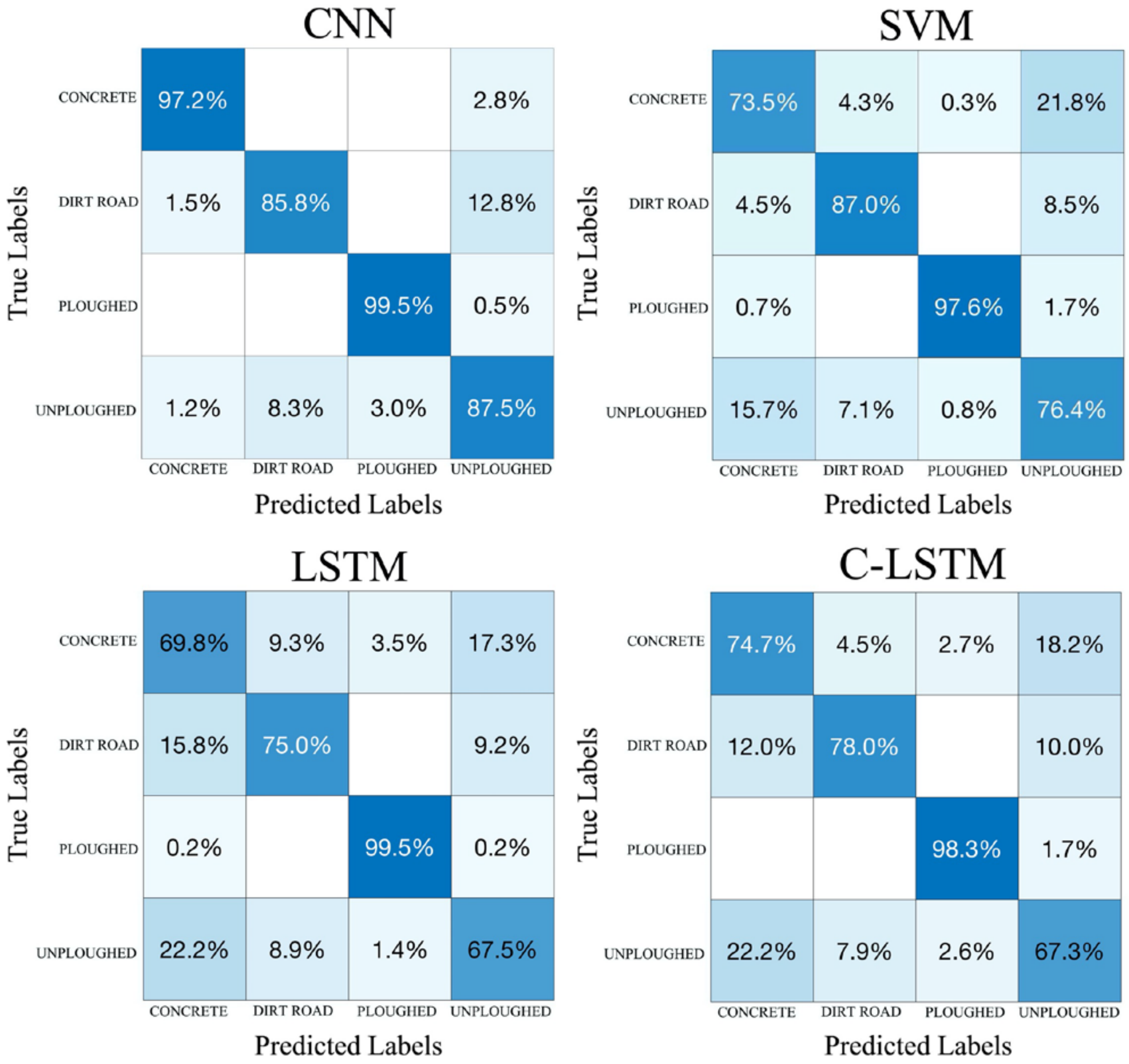

Figure 13:Normalized Confusion Matrixes for all models and MW = 1.7 (SL=85 cm)

V.C. Computational burden

The online implementation of a proprioceptive-based terrain classification model in the onboard processing unit of an autonomous robot requires low memory usage and fast performing. The computational burden of all trained models is therefore analyzed. Results are shown in Table 3 and they were estimated using a CPU intel i-9 working at 2.4 GHz. The overall classification time for a single observation calculated as an average value over the testing dataset is reported for all models in the third row of the table along with the corresponding average k-fold storage space (fourth row).

The time required for the analysis of a given observation can be divided into a feature generation time followed by a classification time. In this respect, RNNs offer an advantage since they do not require any feature generation stage beside being able to give predictions for every instant of acquired raw data. It should be also noted that SVM entails a preliminary stage for the feature space design that can absorb a significant amount of time and it is subject to a large extent on the user expertise.

Although, this “supervised” preparatory stage is overlooked in this analysis, it represents a clear limitation of SVM when compared to CNN.

SVM model needs less time to predict the terrain type but weights more on the memory. Memory space occupied by SVM model depends on how much support vectors are retained to classify a dataset and varies with types of features extracted from signals. Highly non-linear problems correspond to a vast amount of support vectors to be retained hence occupying larger space. On the contrary, storage space occupied by RNNs depend only on the engineered structure, simple recurrent structures result therefore in lighter models for resolving highly non-linear classification problems. It is noteworthy also that classification time required by RNNs is highly dependent on sequence length. Longer sequences require more calculations to be classified due to the recurrent structure of the model. Memory space occupied by CNN model depends on the input size of the multichannel spectrogram that is related to sequence length as well.

Figure 14 shows relationship between sample length (SL) and required classification time for neural networks models. From SL=30 cm to SL=1 m the samples are quite representative of terrain and increasing their dimension results in more computation required for classification. As it can be seen from the slope of curves in Figure 14 this effect is more significant for RNNs than it is for CNN. Increasing SL from 30 cm to 1 m results in 0.083 ms more required by C-LSTM, 0.088 ms more for LSTM and 0.0329 ms more for CNN. Larger values of MW result in higher computation time required also for feature generation. Build the feature vector based on statistical moments for SVM requires at least 0.095 ms for MW=0.5 s and 0.157 ms for MW=2 s. Assembling the multichannel spectrograms computed with FFT requires the largest amount of time for CNN classification, ranging from 0.665 ms for MW=0.5 s and 2.6 ms for MW=2 s.

Table 3: Model’s computation burden for the classification of a single observation. For all models it is assumed MW=1.7 s (i.e., SL=85 cm)

	SVM	CNN	LSTM	C-LSTM
a: Feature generation	0.140 ms	2.157 ms	0 ms	0 ms
b: Classification	0.013 ms	0.104 ms	0.228 ms	0.277 ms
a+b: Total computation time	0.153 ms	2.261 ms	0.228 ms	0.277 ms
Storage space	0.671 MB	0.025 MB	0.020 MB	0.314 MB

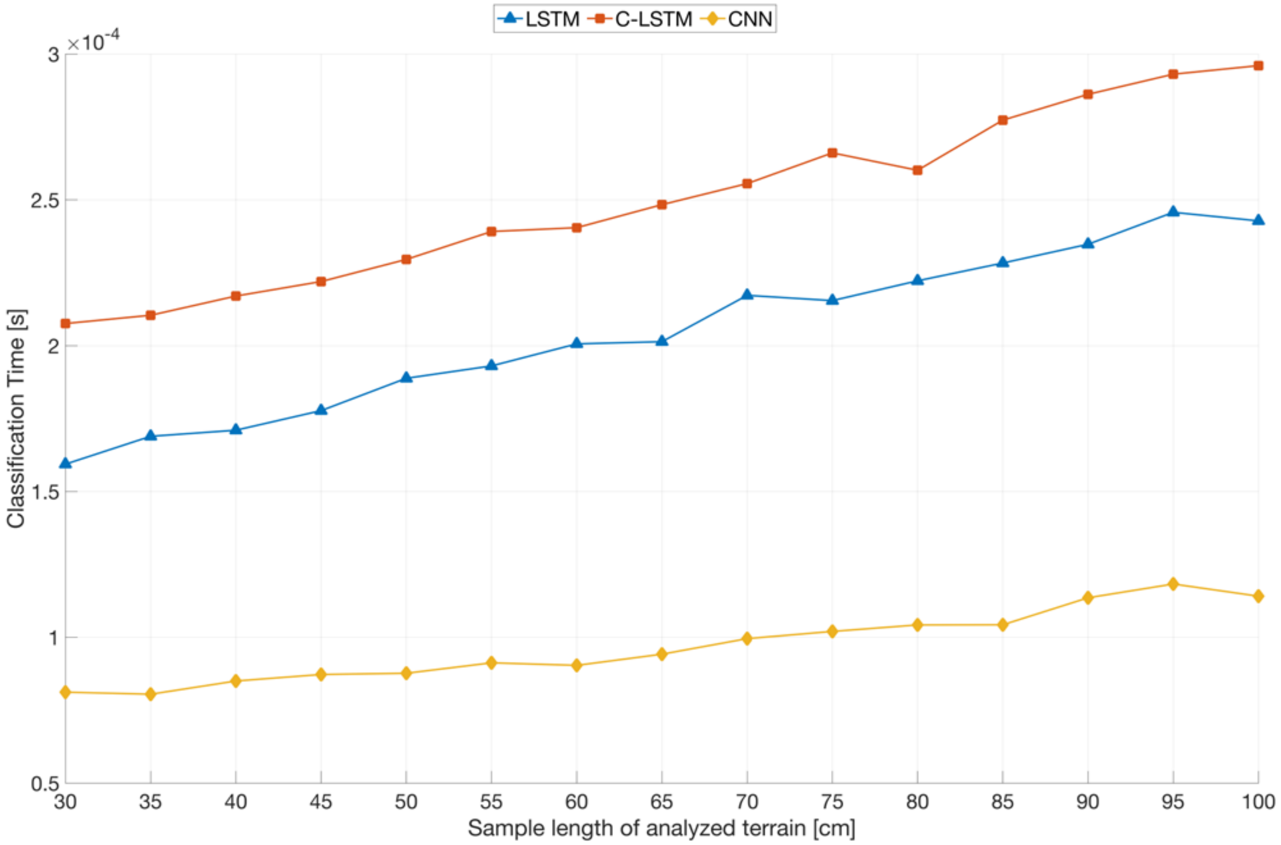

Figure 14: Average classification time as a function of the sample length LSTM (blue Δ), C-LSTM (red □) and CNN (yellow ◊)

VI. Conclusion

This research investigated the use of three deep learning strategies to tackle the terrain estimation problem using proprioceptive data. Ten sensory measurements were used to feed models that predict the traversed terrain after a sufficient listening time. Sensor's channels efficient combination plays a key role in the terrain estimation task. Best performance was obtained using FFT to compute spectrograms of available signals for every terrain patch and assemble them in a multichannel image within a convolutional network. The CNN model correctly classified up to 92.8% of the observations, 10.5% more than standard SVM concatenating statistical moments, 16% more than LSTM and 14.2% more than C-LSTM. In the existing Literature, higher performance for SVM were reached through extraction of terramechanics parameters often requiring expensive sensors like load-cells for forces and torques on wheels or unavailable ones like GPS for wheel slippage. Nevertheless, SVM proved to be faster than all tested models although RNNs can give real time estimate of traversed terrain with lower accuracy. The largest accuracy was achieved for 1.7 s of sensor's recordings or approximately 85 cm of traversed terrain. Required computations for prediction took the CNN model 2.26 ms on average making it the slowest of tested models but still suited for online prediction, especially because the use of a dedicated GPU or low-level programming would boost computation times. The proposed proprioceptive-based CNN along with the feature extraction and organization proved advantageous under a variety of aspects. Besides being more accurate than other deep models while being fast enough for online terrain implementation, it can be fed with unfiltered data, hardly classifiable with SVM [3].

Future developments will deal with the study of scalability and portability to other robots of different size, weights, and locomotion systems. Strategies for self-supervised learning will be also pursued where the vehicle starts its operations with no a priori knowledge of the models that are built progressively during motion.

Software repository

The codes and data used for this research are publicly available at this link (provisional): https://github.com/Ph0bi0/ATLAS_JT

Acknowledgements

The financial support of the projects: Autonomous DEcision making in very long traverses (ADE), H2020 (Grant No. 821988), and Agricultural inTeroperabiLity and Analysis System (ATLAS), H2020 (Grant No. 857125), is gratefully acknowledged.

VII. Bibliografia

[1] S. Wang, Road Terrain Classification Technology for Autonomous Vehicle, Singapore: Springer, 2019.

[2] R. Gonzalez, C. Samuel and D. Apostolopoulos, "Characterization of machine learning algorithms for slippage estimation in planetary exploration rovers," *Journal of Terramechanics,* vol. 82, pp. 23-34, 2019.

[3] R. Gonzalez and K. Iagnemma, "DeepTerramechanics: Terrain Classification and Slip Estimation for Ground Robots via Deep Learning, arXiv:1806.07379v1," 12 June 2018. [Online]. Available: https://arxiv.org/abs/1806.07379v1. [Accessed 3 June 2020].

[4] S. Otte, S. Laible, R. Hanten, M. Liwicki and A. Zell, "Robust Visual Terrain Classification with Recurrent Neural Networks," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning,* Bruges (Belgium), 22-24 April 2015.

[5] C. Bai, J. Guo, L. Guo and J. Song, "Deep Multi-Layer Perception Based Terrain Classification for Planetary Exploration Rovers," *Sensors,* vol. 19, no. 14: 3102, 2019.

[6] G. Reina, M. Annalisa and R. Galati, "Terrain assessment for precision agriculture using vehicle dynamic modelling," *Biosystems Engineering,* vol. 162, pp. 124-139, 2017.

[7] E. G. F. Narváez, A. Escolà, J.R.Rosell-Polo, M. Torres-Torriti e F. A. Cheein, «Terrain classification using ToF sensors for the enhancement of agricultural machinery traversability,» *Journal of Terramechanics,* vol. 76, pp. 1-13, 2018.

[8] S. Goldberg, M. Maimone e L. Matthies, «Stereo-vision and rover navigation software for planetary exploration,» in *IEEE Areospace Conference,* Big Sky, Montana.

[9] G. Reina e A. Milella, «Toward autonomous agriculture: automatic ground detection using trinocular stereovision,» *Sensors,* vol. 12, n. 9, pp. 12405-12423, 2012.

[10] P. Ross, A. English, D. Ball, B. Upcroft e P. Corke, «Online novelty-based visual obstacle detection for field robotics,» in *IEEE International Conference on Robotics and Automation (ICRA),* 2015.

[11] G. Reina, A. Leanza e A. Milella, «Mind the ground: A power spectral density-based estimator for all-terrain rovers,» *Measurement,* vol. 151, 2020.

[12] A. Milella, G. Reina e J. Underwood, «A Self-learning Framework for Statistical Ground Classification using Radar and Monocular Vision,» *Journal of Filed Robotics,* vol. 32, n. 1, 2015.

[13] D. Stavens e S. Thrun, «A self-supervised terrain roughness estimator for off-road autonomous driving,» in *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006.

[14] M. Bekker, Introduction to Terrain-Vehicle Systems, Ann Arbor: University of Michigan Press, 1969.

[15] A. Angelova, L. Matthies, D. Helmick e P. Perona, «Learning and Prediction of Slip,» *Journal of Field Robotics,* vol. 24, n. 3, pp. 205-231, 2006.

[16] C. Brooks and K. Iagnemma, "Vibration-based Terrain Classification for Planetary Exploration Rovers," *IEEE Transactions on Robotics,* vol. 21, no. 6, p. 1185–1191, 2005.

[17] A. Valada and W. Burgard, "Deep Spatiotemporal Models for Robust Proprioceptive Terrain Classification," *The International Journal of Robotics Research,* vol. 36, no. 13-14, p. 1521–1539, 2017.

[18] J. Libby and A. J. Stentz, "Using Sound to Classify Vehicle-Terrain Interactions in Outdoor Environments," in *2012 IEEE International Conference on Robotics and Automation,* Saint Paul, MN, 2012.

[19] P. Giguere and G. Dudek, "Clustering Sensor Data for Autonomous Terrain Identification using Time-Dependency," *Autonomous Robots,* vol. 26, pp. 171-186, 2009.

[20] L. Ojeda, J. Borenstein, G. Witus and R. Karlsen, "Terrain Characterization and Classification with a Mobile Robot," *Journal of Field Robotics,* vol. 23, pp. 103-122, 2006.

[21] P. Bellutta, R. Manduchi, L. Matthies, K. Owens and A. Rankin, "Terrain Perception for DEMO III," *IEEE Intelligent Vehicles Symposium,* pp. 326-331, 2000.

[22] R. Gonzalez, A. Rituerto and J. Guerrero, "Improving Robot Mobility by Combining Downward-Looking and Frontal Cameras," *Robotics,* vol. 5, pp. 25-44, 2016.

[23] R. Manduchi, A. Castano, A. Talukder and L. Matthies, "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation," *Autonomous Robots,* vol. 18, pp. 81-102, 2005.

[24] J. Martinez-Gomez, A. Fernandez-Cabellero, I. Garcia-Varea, L. Rodriguez and C. Romero-Gonzalez, "A Taxonomy of Vision Systems for Ground Mobile Robots," *International Journal of Advanced Robotic Systems,* vol. 11, pp. 1-26, 2014.

[25] Shamrao, C. Padmanabhan, S. Gupta and A. Mylswamy, "Estimation of terramechanics parameters of wheel-soil interaction model using particle filtering," *Journal of Terramechanics,* no. 79, pp. 79-95, 2018.

[26] C. Weiss, H. Frohlich and A. Zell, "Vibration-based terrain classification using support vector machines," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Beijing, China, October 2006..

[27] C. Weiss, M. Stark and A. Zell, "SVMs for Vibration-Based Terrain Classification," in *Autonome Mobile Systeme 2007. Informatik aktuell*, Heidelberg, Springer, Berlin, 2007.

[28] K. Zhao, M. Dong and L. Gu, "A New Terrain Classification Framework Using Proprioceptive Sensors for Mobile Robots," *Hindawi,* vol. 2017, no. 3938502, p. 14, 2017.

[29] C. A. Brooks e K. Iagnemma, «Self-supervised terrain classification for planetary surface exploration rovers,» *Journal of Field Robotics,* vol. 29, n. 3, pp. 445-468, 2012.

[30] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, p. 1735– 1780, 1997.

[31] A. Graves, A.-r. Mohamed and G. Hinton, "Speech Recocnition With Deep Recurrent Neural Networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 2013.

[32] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "Wavenet: a generative model for raw audio, arXiv:1609.03499v2," 19 September 2016. [Online]. Available: https://arxiv.org/abs/1609.03499v2. [Accessed 3 June 2020].

[33] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks, arXiv:1409.3215v3," 14 December 2014. [Online]. Available: https://arxiv.org/abs/1409.3215v3. [Accessed 3 June 2020].

[34] J. Park, K. Min, H. Kim, W. Lee, G. Cho and K. Huh, "Road Surface Classification Using a Deep Ensemble Network with Sensor Feature Selection," *Sensors (Basel),* vol. 18, no. 12, 2018.

[35] K. S. a. M. R. M. Hoffmann, «"The Effect of Motor Action and Different Sensory Modalities on Terrain Classification in a Quadruped Robot Running with Multiple Gaits,» *Robotics and Autonomous Systems,* vol. 62, n. 12, p. 1790–1798, 2014.

[36] M. B. L. W. M. H. a. K. W. J. Bednarek, «What am i touching? learning to classify terrain,» in *International Conference on Robotics and Automation (ICRA)*, 2019.

[37] S. M. H. D. a. S. C. F. Karim, «Lstm fully convolutional networks for time series classification,» *IEEE Access*, vol. 6, pp. 1662-1669.

[38] K. W. P. G. a. B. C. P. Dallaire, «Learning terrain Types with the Pitman-Yor Process Mixtures of Gaussians for a Legged Robots,» in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[39] T. Eitrich and B. Lang, "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *Journal of Computational and Applied Mathematics,* vol. 196, no. 2, pp. 425-436, 2006.

[40] G. Reina, A. Milella, R. Rouveure, M. Nielsen, R. Worst e M. R. Blas, «Ambient awareness for agricultural robotic vehicles,» *Biosystems engineering*, vol. 146, pp. 114-132, 2016.

[41] R. Marani, A. Milella, A. Petitti and G. Reina, "Deep learning-based image segmentation for grape bunch detection," in *Precision agriculture '19*, John V. Stafford, 2019, pp. 791-797.

[42] A. Milella, R. Marani, A. Petitti and G. Reina, "In-field high throughput grapevine phenotyping with a consumer-grade depth camera," *Computers and Electronics in Agriculture,* vol. 156, pp. 293-306, 2019.

[43] T. M. H. A. S. S. A. S. a. M. R. C. X. A. Wu, «Tactile sensing and terrain-based gait control for small legged robots,» *IEEE Transactions on Robotics*, 2019.