Terrain Classification Enhanced with Uncertainty for Space Exploration Robots from Proprioceptive Data

Mariela De Lucas Álvarez [*1] **Jichen Guo** [*2] **Raúl Domíguez** [1] **Matias Valdenegro-Toro** [3]

Abstract

Terrain Classification is an essential task in space exploration, where unpredictable environments are difficult to observe using only exteroceptive sensors such as vision. Implementing Neural Network classifiers can have high performance but can be deemed untrustworthy as they lack transparency, which makes them unreliable for taking high-stakes decisions during mission planning. We address this by proposing Neural Networks with Uncertainty Quantification in Terrain Classification. We enable our Neural Networks with Monte Carlo Dropout, DropConnect, and Flipout in time series-capable architectures using only proprioceptive data as input. We use Bayesian Optimization with Hyperband for efficient hyperparameter optimization to find optimal models for trustworthy terrain classification.

1. Introduction

Terrain Classification (TC) is a common research problem in rover exploration, typically addressed through computer vision (Liyanage et al., 2020; Wietrzykowski & Belter, 2014). Given the many challenges that arise when deploying rovers for space exploration, integrating proprioceptive sensors to classify terrain is a practical solution.

We address TC for rovers using only proprioceptive data with the goal of enabling reliable extra-planetary missions. Environmental conditions that cause fluctuations in illumination or atmospherical conditions can impact the reliability of visual input. To mitigate this issue, we propose classifiers that are not reliant on visual input to maintain accuracy and robustness in the event of unforeseen environmental events. Specifically, we aim to use visual-independent or proprioceptive sensors such as Inertial Measurement Units (IMUs) and motor joint states to train a set of Neural Networks (NNs). The collected data contains terrain properties, which represent mobility characterizations from the navigation of the rover.

Neural networks still pose reliability concerns when being deployed in high-stakes applications (Li et al., 2023). We mitigate the issue of NN untrustworthiness by generating networks with Uncertainty Quantification (UQ) layers to Time Series Classification (TSC) architectures such as the Long Short-Term Memory Network (LSTM) (Hochreiter & Schmidhuber, 1997) and the Convolutional Neural Network (CNN) (LeCun et al., 1998). We also enhance state-of-the-art architectures which include the Fully Convolutional Network (FCN), a ResNet and an Attention Encoder (Wang et al., 2017; Serrà et al., 2018).

We use Bayesian Optimization with Hyperband (BOHB) (Falkner et al., 2018) for hyperparameter optimization to generate a set of high-accuracy CNN and LSTM candidates. We then assess UQ and non-UQ networks in terms of predictive accuracy, Expected Calibration Error (ECE) (Guo et al., 2017; Naeini et al., 2015) and use predictive entropy as a metric for UQ to determine the reliability of the classification.

We quantitatively show the advantage of using UQ-NNs for reliably selecting good-performing candidates for a task where vision is traditionally used. We show that our networks are trustworthy for TC in critical settings with data that is difficult to interpret by simple observation.

2. Related Work

Identifying unknown terrain types has always been one fundamental challenge for mobile robot navigation. Existing research on Terrain Classification primarily focuses on visual or exteroceptive sensors to classify terrain by using navigation cameras (Helmick et al., 2009; Rothrock et al., 2016), RGB-D cameras (Kozlowski & Walas, 2018), hyperspectral cameras (Winkens et al., 2017; Li et al., 2020) or LADAR (Vandapel et al., 2004). Although the visual features of different terrains provide important information

[*]Equal contribution [1]German Research Center for Artificial Intelligence [2]AG Robotik, University of Bremen [3]Bernoulli Institute, University of Groningen. Correspondence to: Mariela De Lucas Álvarez <mariela.de_lucas_alvarez@dfki.de>, Jichen Guo <jichen@uni-bremen.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Figure 1. The AsguardIV is a hybrid leg-wheel robot designed at the DFKI to allow navigation in unstructured environments. Its rimless wheels are simpler, more energy-efficient, and more reliable than articulated legs, adapting effectively to obstacles and uneven terrain.

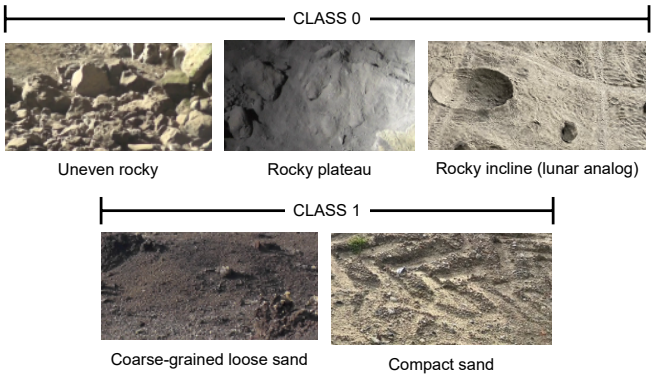


Figure 2. The experimental sites for recording data include terrain that is mostly comprised of sand and rock. Rock is represented as uneven unconsolidated rock and flat rock plateaus landscapes or inclines shown as $Class_0$ in the first three images. Sand is represented present in a loose and compact form shown as $Class_1$ in the last two images. These images serve as visual aids for labeling and are not used to train our models.

to estimate the ground type nearby, some constraints for Terrain Classification with images include lighting and atmospherical conditions, such as inadequate lighting for a hyperspectral camera and clouds of gas, smoke, or sand when using a LADAR or standard cameras.

Conversely, proprioceptive sensors provide information about the internal state of the robot and are therefore less susceptible to external factors that could compromise the acquisition of data. Existing work addresses TC by using proprioceptive sensors such as IMUs and torque sensors with traditional Machine Learning methods showing promising results (Ojeda et al., 2006; Dimastrogiovanni et al., 2020; Vulpi et al., 2020; Dimastrogiovanni et al., 2021; Ugenti et al., 2022).

Among the most successful methods used for TC are LSTMs and CNNs. Despite their robustness, NNs do not give any quantifiable confidence in a classification problem. To address this, some UQ methods have now been integrated into Deep Learning (DL) approaches to make them trustworthy for critical decisions. The most commonly used UQ methodologies are Probabilistic Approximation and Ensemble Learning (Abdar et al., 2021). In our work, we focus on probabilistic approximation specifically Monte Carlo Dropout (Gal & Ghahramani, 2016), DropConnect (Mobiny et al., 2021) and Flipout (Wen et al., 2018). These Bayesian UQ techniques are increasingly being applied to DL to address the lack of quantifiable uncertainty and combine regularization methodologies. Ensembles are common but computationally demanding. We use Bayesian NNs which are scalable, easy to train and produce high-quality uncertainty.

3. Methods

Our experimental platform the AsguardIV, shown in Figure 1, is a hybrid leg-wheel rover designed to allow navigation in unstructured environments. The rimless wheels are less complex, energy efficient and less prone to failures than articulated legs as they can adapt better when traversing obstacles and uneven terrain. This rover has been used to collect an array of data logs from trials executed across various locations with soil signatures representative of a lunar surface. The experimental sites are comprised of compact and loose sand, and rocky areas. For illustration purposes, we show in Figure 2 the soil characteristics and level of traversal difficulty of compact and loose sand and rocky terrain from such sites. These images also aid in the labeling process but are not used to train our networks.

Binary classification for this task provides a clear contrast between two distinct types of terrain that would be found in a lunar landscape. Based on the characteristics of deformability and evenness, we label rock terrains as $Class_0$ for being uneven and undeformable, and sand terrains as $Class_1$, for even and deformable.

Prior to formatting the inputs for the NNs, we remove any *idle* gaps by trimming the streams. Here, *idle* refers to prolonged gaps when the rover is not making progress. We explore two distinct settings for sequence generation to determine the optimal format. The first approach involves employing a sliding window with predefined width and step length, while the second approach entails sequence subsampling using a specified factor. For the sliding window sequence generation, we define a set $(W, S) = \{(100, 25), (400, 100), (1000, 100)\}$ of three pairs with dif-

Table 1. Configuration space of networks. Values shown are ranges for the BO to select.

CNN & LSTM Layers	Filters	Kernels	Max Pooling	LSTM Cells	Batch Size	Dropout %
[1-3]	[16-128]	[4-16]	[2-8]	[8-128]	[16-64]	[0 - 0.5]

ferent window size w and step size s in time steps.

In this approach, we extract consecutive and overlapping subsequences from the original time series by sliding a fixed-length window of size w along the series with a step size of s. During sequence subsampling, we employ three different subsampling factors f from the set $F = \{8, 16, 32\}$. These factors are selected heuristically to strike a balance between the number of samples and sample length. In this approach, we generate sequences by selecting every f-th time step from the original time series.

We evaluate all architectures with three different proprioceptive input configurations, the IMU and joint data independently and the fused IMU-joint data. The IMU array is a 6-feature vector of the accelerometer and the gyroscope in three axes. The joint data is a time-synced 12-feature vector of speed, acceleration, and effort for 4 wheels. The fused data makes an input of an 18-feature vector. The sensors are recorded at 100 Hz and compose a total of approximately 6 hours of data. We have split the data logs into 70% for training and 30% for testing, ensuring that there is no correlation between test and train samples. The training set is further split into $80\% - 20\%$ for training and validation.

We choose NNs for TC based on the work by (Wang et al., 2017). In this work, architectures composed by CNNs achieve state-of-the-art performance for TSC tasks, specifically a deep Residual Network (ResNet), the Fully Convolutional Neural Network (FCN), and an Attention Encoder. In addition, we design and tune our own LSTMs and CNNs.

We use a combined hyperparameter search and tuning using the automatic method Bayesian Optimization on Hyperband (BOHB) (Falkner et al., 2018). This allows us to speed up the tuning of our networks and generate only good-performing CNN and LSTM networks by combining state-of-the-art performance BO (Bergstra et al., 2013; Snoek et al., 2015) with efficient resource allocation of HB (Li et al., 2017). The implementation for BOHB is available at: `https://github.com/automl/HpBandSter`.

We generate three core architectures with BOHB namely CNN, LSTM and CNN-LSTM. The Bayesian optimizer selects the number of blocks, number of filters, and kernel sizes in each block, maximum pooling size for CNNs and number of layers and cells for LSTMs. General settings like batch size and Dropout percentage are also tuned. The parameter ranges are listed in Table 1. We use the TSC networks with the hyperparameters set according to (Wang et al., 2017) and thus are not optimized with BOHB. After this process, we choose a set of candidate networks.

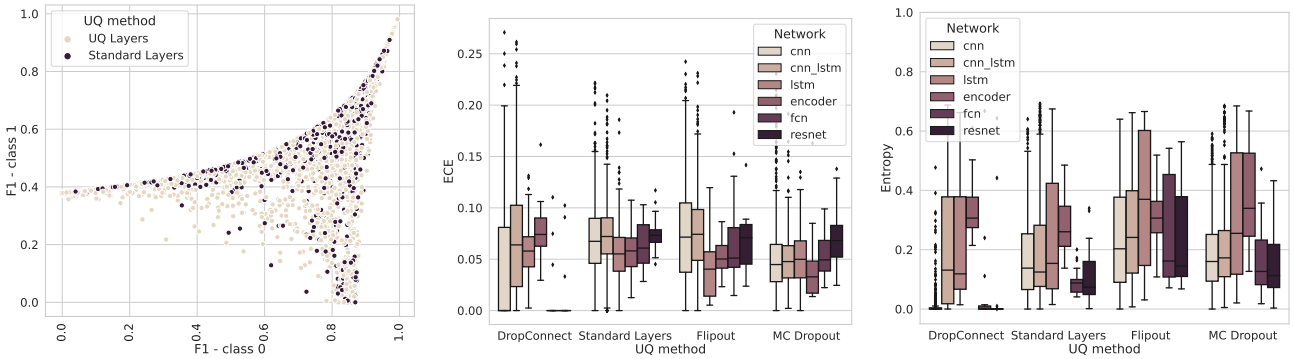
Even with high-performing networks, their reliability is questionable in real-world applications. We address this by using three UQ techniques, namely Monte Carlo Dropout (Gal & Ghahramani, 2016), DropConnect (Mobiny et al., 2021) and Flipout (Wen et al., 2018). Monte Carlo Dropout (MC Dropout) (Gal & Ghahramani, 2016) drops activations at inference time with probability p, while Monte Carlo DropConnect (Mobiny et al., 2021) drops weights at a similar probability at inference time. Both methods produce an approximation to the predictive posterior distribution, which can be reconstructed through sampling.

Flipout (Wen et al., 2018) uses variational inference to approximate the posterior distribution for each weight with a Gaussian distribution, by maximizing the Evidence Lower Bound (ELBO). Flipout in particular is a variation of Bayes by Backprop (Blundell et al., 2015), where different kernel and bias samples are applied to each element in a batch, reducing variance during the training process and improving convergence. To make predictions, we estimate the predictive posterior distribution using $M = 10$ samples with Equation 1,

$$\mathbb{P}(\mathbf{y} \mid \mathbf{x}) \sim M^{-1} \sum_{i}^{M} \mathbb{P}(\mathbf{y} \mid \theta_i, \mathbf{x}) \quad \theta_i \sim \mathbb{P}(\mathbf{w} \mid \mathbf{x}) \tag{1}$$

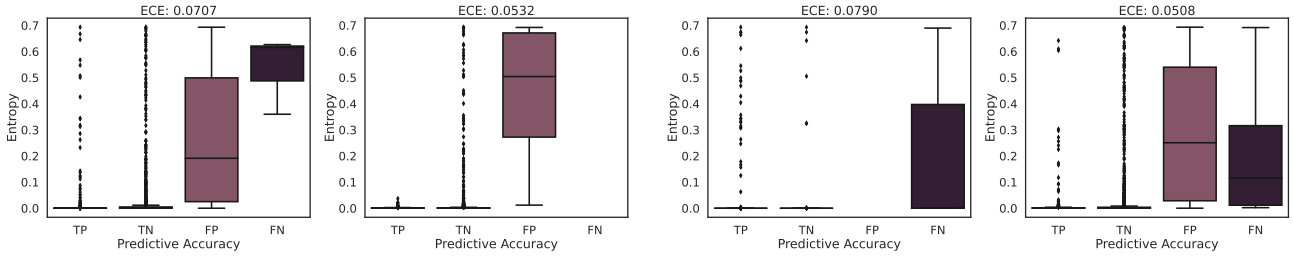
For Flipout, there is weight sampling, while in the case of MC Dropout and MC DropConnect, we only make forward passes $\mathbb{P}(\mathbf{y} \mid \theta, \mathbf{x})$. Monte Carlo Dropout layers are introduced after each 1-D Convolutional layer and before batch normalization in only the first two Convolutional blocks. This also applies if the architecture generated has only one CNN block. An MC Dropout layer is also applied before the classification layer if the architecture has LSTM layers. DropConnect Convolutional layers replace the standard layers in CNN-LSTM and CNN architectures. DropConnect Dense layers only replace the standard Dense classification layers in LSTM architectures. Flipout Dense layers are applied only to the output classification layer in all three architectures.

Similarly, the TSC architectures, ResNet, FCN and Attention Encoder from (Wang et al., 2017), are also enhanced with UQ layers. DropConnect layers substitute the traditional ones in the ResNet only for the convolutional blocks and not the *shortcut* block used for channel expansion. All



(a) F1 score for each class (b) ECE per UQ method (c) Entropy per UQ method

Figure 3. All candidates, both BOHB and TSC. (a) Performance behavior is observed by comparing $F1_{cl_0}$ vs. $F1_{cl_1}$. (b) ECE scores by UQ and architecture. (c) Predictive entropy scores by UQ and architecture.



(a) LSTM - DropConnect (b) CNN-LSTM - MC Dropout (c) CNN - Flipout (d) LSTM - Standard Layer

Figure 4. Predictive accuracy versus entropy from the selected candidates for each UQ method.

layers are replaced for the FCN architecture and the Attention Encoder. MC Dropout and Flipout Dense layers replace all standard Dropout layers and Dense classification output layers respectively.

We assess all UQ networks against their non-UQ counterparts. The assessment criteria that we use for UQ is predictive entropy which measures the uncertainty of the prediction. This is useful to detect classifications from high-performing networks with high uncertainty. This metric is given by Equation 2,

$$H(\mathbb{P}) = -\sum_{c \in C} \mathbb{P}(c) \log \mathbb{P}(c) \tag{2}$$

with $c \in C$ the set of classes, and $\mathbb{P}(c)$ the probability of class c. In addition, we use ECE as a metric to assess the calibration of our candidate models. By quantifying the difference between predicted probabilities and the actual frequencies, ECE provides a measure of confidence and reliability in the predictive outputs of the model. This score is calculated by Equation 3 to calculate the empirical probabilities of all instances that fall into bin i,

$$ECE = \sum_{i=1}^{K} P(i) \cdot |o_i - e_i| \tag{3}$$

where o_i is the true fraction of positive instances and e_i the mean of post-calibrated probabilities.

4. Results and Discussion

We train all the networks with the Nvidia GeForce RTX 3070 and RTX 2070 graphics cards. We use Adam (Kingma & Ba, 2014) as our optimization algorithm with an initial learning rate of 10^{-2}. The maximum budget given to the Bayesian optimizer was 50 epochs, successive halving being executed at the 16_{th} epoch. We conduct a total of 216 BOHB studies, each with different input data, UQ settings, and architecture search configurations. The studies were run for 20 BO iterations resulting in a pool of $6,480$ full-budget candidates. Similarly, the experiments for TSC architectures result in a total of 216 candidates.

We analyze the performance of UQ versus non-UQ networks paying attention to the $F1$ scores for each class, i.e. the harmonic mean of precision and recall. Sequences corresponding to $Class_1$ are underrepresented in our dataset.

Table 2. Performance and hyperparameters of selected networks with UQ methods of MC Dropout (DO), DropConnect (DC), and Flipout (FO) and with Standard Layers (SL).

ID	Study	UQ	Entropy	ECE	$F1_{cl_0}$	$F1_{cl_1}$	$F1_{weighted}$	Acc	Dropout rate	CNN	LSTM
	CNN										
1	jstate-imu-in1000-step100	DO	0.0739	0.0134	0.9726	0.9110	0.9582	0.9581	0.46	$\{f_1:22, f_2:21, k_1:11, k_2:16\}$	
2	imu-in1000-step100	SL	0.0374	0.0556	0.9704	0.9095	0.9562	0.9554	0.21	$\{f_1:26, f_2:19, k_1:6, k_2:7\}$	
3	imu-in400-step100	DO	0.0843	0.0122	0.9787	0.9280	0.9669	0.9672	0.42	$\{f_1:16, k_1:7\}$	
4	imu-in400-step100	DO	0.0451	0.0354	0.9801	0.9342	0.9694	0.9695	0.32	$\{f_1:77, k_1:11\}$	
5	imu-in400-step100	DO	0.0410	0.0493	0.9704	0.9050	0.9551	0.9548	0.34	$\{f_1:77, f_2:22, k_1:14, k_2:8\}$	
6	imu-in1000-step100	DO	0.0707	0.0122	0.9742	0.9171	0.9608	0.9606	0.50	$\{f_1:16, k_1:16\}$	
7	imu-in1000-step100	DO	0.0381	0.0355	0.9701	0.9120	0.9565	0.9554	0.18	$\{f_1:16, k_1:6\}$	
8	imu-in1000-step100	DO	0.0211	0.0621	0.9813	0.9430	0.9724	0.9719	0.11	$\{f_1:52, k_1:13\}$	
9	imu-in1000-step100	DO	0.0405	0.0290	0.9678	0.9071	0.9536	0.9522	0.11	$\{f_1:54, k_1:9\}$	
10	imu-in1000-step100	DO	0.0693	0.0264	0.9723	0.9177	0.9596	0.9586	0.43	$\{f_1:63, f_2:41, k_1:8, k_2:7\}$	
11	imu-in1000-step100	DO	0.0433	0.0238	0.9761	0.9255	0.9643	0.9638	0.15	$\{f_1:47, k_1:10\}$	
12	imu-in1000-step100	DO	0.0381	0.0291	0.9798	0.9375	0.9699	0.9694	0.23	$\{f_1:56, f_2:78, k_1:7, k_2:4\}$	
13	imu-in1000-step100	DO	0.0402	0.0127	0.9800	0.9388	0.9704	0.9698	0.09	$\{f_1:19, k_1:14\}$	
14	imu-in1000-step100	DO	0.0187	0.0494	0.9840	0.9508	0.9763	0.9759	0.02	$\{f_1:29, k_1:11\}$	
15	imu-in1000-step100	DO	0.0323	0.0644	0.9730	0.9181	0.9602	0.9594	0.23	$\{f_1:51, f_2:75, f_3:57, k_1:11, k_2:6, k_3:6\}$	
16	imu-in400-step100	FO	0.0710	0.0256	0.9797	0.9315	0.9685	0.9687	0.40	$\{f_1:97, k_1:15\}$	
17	imu-in400-step100	FO	0.0261	0.0515	0.9761	0.9132	0.9614	0.9625	0.01	$\{f_1:116, k_1:14\}$	
18	**imu-in400-step100**	**FO**	**0.0096**	**0.0790**	**0.9871**	**0.9550**	**0.9796**	**0.9799**	0.17	$\{f_1:16, f_2:63, f_3:102, k_1:6, k_2:5, k_3:11\}$	
19	imu-in1000-step100	FO	0.0413	0.0296	0.9784	0.9336	0.9679	0.9674	0.10	$\{f_1:23, k_1:6\}$	
20	imu-in1000-step100	FO	0.0227	0.0343	0.9843	0.9485	0.9759	0.9759	0.02	$\{f_1:46, f_2:36, f_3:17, k_1:12, k_2:14, k_3:14\}$	
21	imu-in1000-step100	DO	0.0252	0.0369	0.9819	0.9445	0.9731	0.9727	0.17	$\{f_1:107, f_2:28, k_1:12, k_2:5\}$	
	CNN-LSTM										
22	imu-in400-step100	DO	0.0629	0.0235	0.9775	0.9308	0.9666	0.9660	0.06	$\{f_1:64, f_2:62, k_1:12, k_2:9\}$	$\{u_1:8\}$
23	imu-in1000-step100	DO	0.0163	0.0556	0.9670	0.9044	0.9524	0.9509	0.02	$\{f_1:53, k_1:12\}$	$\{u_1:22, u_2:18, u_3:36\}$
24	**imu-in1000-step100**	**DO**	**0.0142**	**0.0532**	**0.9942**	**0.9814**	**0.9912**	**0.9912**	0.04	$\{f_1:26, k_1:9\}$	$\{u_1:67\}$
25	imu-in1000-step100	DO	0.0497	0.0152	0.9705	0.9120	0.9568	0.9558	0.12	$\{f_1:54, k_1:12\}$	$\{u_1:11, u_2:59\}$
26	imu-in1000-step100	FO	0.0417	0.0444	0.9816	0.9362	0.9710	0.9715	0.01	$\{f_1:79, f_2:94, f_3:105, k_1:8, k_2:14, k_3:4\}$	$\{u_1:18, u_2:9\}$
27	imu-in1000-step100	FO	0.0675	0.0466	0.9718	0.9165	0.9588	0.9578	0.06	$\{f_1:94, f_2:27, k_1:5, k_2:5\}$	$\{u_1:65, u_2:10\}$
	LSTM										
28	imu-in400-step100	SL	0.0596	0.0483	0.9760	0.9253	0.9642	0.9637	0.0		$\{u_1:41\}$
29	jstate-imu-in400-step100	DC	0.0347	0.0580	0.9689	0.9091	0.9549	0.9536	0.25		$\{u_1:113, u_2:13\}$
30	imu-in1000-step100	DC	0.0304	0.0531	0.9791	0.9337	0.9685	0.9682	0.25		$\{u_1:93\}$
31	**imu-in400-step100**	**DC**	**0.0361**	**0.0707**	**0.9664**	**0.9032**	**0.9517**	**0.9502**	0.25		$\{u_1:23, u_2:74\}$
32	imu-in400-step100	DC	0.0383	0.0587	0.9690	0.9078	0.9547	0.9537	0.25		$\{u_1:53\}$
33	imu-in400-step100	DO	0.0868	0.0616	0.9714	0.9148	0.9581	0.9571	0.25		$\{u_1:24, u_2:10\}$
34	imu-in400-step100	DO	0.0395	0.0656	0.9675	0.9028	0.9524	0.9513	0.25		$\{u_1:95\}$
35	imu-in400-step100	SL	0.0607	0.0519	0.9708	0.9075	0.9560	0.9556	0.0		$\{u_1:44\}$
36	**imu-in400-step100**	**SL**	**0.0367**	**0.0508**	**0.9701**	**0.9109**	**0.9562**	**0.9552**	0.0		$\{u_1:53\}$
	ResNet										
37	imu-in1000-step100	FO	0.0812	0.0238	0.9683	0.9085	0.9544	0.9530	0.25		

The percentage varies according to each sequence generation setting but it approximately averages to 50%. Hence, we are interested in high $F1$ scores for both classes. The overall performance of all the networks is shown in Figure 3. We disregard deficient models and present the performance of $6,603$ candidates by comparing $F1$ scores for each class, ECE per architecture, and Entropy per architecture.

Figure 3a shows how BOHB ensures that many candidates reach optimal accuracy performance for both UQ-enabled and non-UQ architectures. However, it is clear from Figure 3b that MC Dropout architectures achieve better calibration scores whereas Figure 3c illustrates how DropConnect architectures appear to have even lower uncertainty than other UQ methods. To assess this more closely, we select a set S of candidates that meet the following criteria,

$$S = \begin{cases} \text{Select, if } (F1_{cl_0} \wedge F1_{cl_1}) >= 0.9 \\ \qquad \text{and } H(\mathbb{P}) <= 0.1 \\ \text{Reject, if } (F1_{cl_0} \vee F1_{cl_1}) < 0.9 \\ \qquad \text{or } H(\mathbb{P}) > 0.1 \end{cases} \tag{4}$$

Table 2 shows the 37 selected models with this criteria. We identify the studies by the sensor input and some variation in sequence structure. We then list their Uncertainty Quantification method, performance by predictive entropy, ECE, $F1$ scores for both classes and weighted score, accuracy, and the hyperparameters for the candidates that have been tuned with BOHB.

We observe that models trained only with IMU input significantly outperform models trained with other sensor inputs. Additionally, we find that sequences generated with the sliding window approach achieve better performance compared to subsampled sequences. This suggests that when dealing with distinct mobility signatures, subsampling may result in the loss of crucial information, leading to underperformance.

Establishing a direct impact of the input size on the performance of the model is an interesting task given the number of settings and parameters. Out of a total of 37 selected models, 10-second inputs are required for 22 of them, while the remaining 15 require 4-second inputs. Although longer 10-second sequences may enhance the identification of characteristic terrain features, we observe that the performance of the shorter 4-second sequences is comparable to their 10-second counterparts. These results are important for

evaluating the impact of input size on model performance and suggest that shorter input sequences may be sufficient to achieve robust classification in a planetary exploration scenario where computational advantages are necessary.

We highlight interesting values from the selected studies in bold. From this set, we choose one candidate with low entropy for each UQ method and analyze the performance as shown in Figure 4. Here, we display four important details: ECE, Entropy, and Predictive Accuracy values, and architecture.

We note that each candidate has a good calibration score. While this varies on each application, it is generally accepted that values below 0.1 are considered well-calibrated. In addition, we expect that incorrect predictions, FPs and FNs, should have higher entropy, i.e. be more uncertain, than correct predictions, TPs and TNs. This assumption holds again for the MC Dropout candidate with no misclassifications for $Class_1$. Flipout has no FPs and wide uncertainty for FNs. Standard Layers and DropConnect are comparable in uncertainty performance and surpassed by MC Dropout. We also note that these candidates are all BOHB networks.

Overall, we believe that these results show the usefulness of UQ predictive entropy when used to assess the quality and likelihood of a prediction being correct. For our Terrain Classification problem, NNs without uncertainty are inadequate and only MC Dropout models provide reliable classifications. TSC architectures appear to be less variable in UQ and calibration performance, but this is due to the number of total candidates compared to BOHB.

5. Conclusion and Future Work

We focus on the advantage of UQ in a problem of Terrain Classification for an exploration rover motivated by the need for reliable and trustworthy NNs. We compare a wide range of NN candidates using fast and efficient BOHB optimization and UQ methods. We have also compared our models with UQ-enabled benchmarks: ResNet, FCN and Attention Encoder.

In the context of space missions, models that produce high-confidence outputs with low uncertainty can ensure navigation safety by enabling navigation with hazardous terrain avoidance. Our results demonstrate a clear advantage to integrating UQ into TC. This furthermore provides higher trustworthiness during planetary exploration.

Our future work involves online testing our networks in analogous scenarios to provide insights into the limitations of binary TC and extend the classification to additional terrain types for terrestrial missions. We also aim to employ multiple-objective optimization techniques. By incorporating entropy as an additional performance metric, we seek to generate neural network configurations that optimize both high performance and low entropy simultaneously. This approach will enable us to further refine the training process and achieve more robust and balanced models.

Acknowledgements

We gratefully acknowledge the support and funding provided by projects Insys and Persim funded by the Space Agency of the German Aerospace Center with federal funds of the Federal Ministry of Economic Affairs and Climate Action (BMWK) in accordance with the parliamentary resolution of the German Parliament, grants no. 50RA2036 and 50RA2124.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, dec 2021. ISSN 1566-2535. doi: 10.1016/j.inffus.2021.05.008.

Bergstra, J., Yamins, D., and Cox, D. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. volume 28, 2013.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks, 2015. URL https://arxiv.org/abs/1505.05424.

Dimastrogiovanni, M., Cordes, F., and Reina, G. Terrain estimation for planetary exploration robots. *Applied Sciences (Switzerland)*, 10(17):6044, aug 2020. ISSN 20763417. doi: 10.3390/app10176044. URL https://www.mdpi.com/2076-3417/10/17/6044.

Dimastrogiovanni, M., Cordes, F., and Reina, G. Terrain Sensing for Planetary Rovers. *Mechanisms and Machine Science*, 91:269–277, 2021. ISSN 22110992. URL https://link.springer.com/chapter/10.1007/978-3-030-55807-9_31.

Falkner, S., Klein, A., and Hutter, F. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. In *35th International Conference on Machine Learning, ICML 2018*, volume 4, pp. 2323–2341, 2018. ISBN 9781510867963.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On Calibration of Modern Neural Networks. 2017.

Helmick, D., Angelova, A., and Matthies, L. Terrain adaptive navigation for planetary rovers. *Journal of Field Robotics*, 26(4):391–410, 2009.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. dec 2014. URL http://arxiv.org/abs/1412.6980.

Kozlowski, P. and Walas, K. Deep neural networks for terrain recognition task. In *2018 Baltic URSI Symposium (URSI)*, pp. 283–286. IEEE, 2018.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., and Berkeley, U. Hyperband: Bandit-based Configuration Evaluation for Hyperparameter Optimization. In *International Conference on Learning Representations*, 2017.

Li, L., Zhang, Y., Ren, L., Xiong, Y., and Xie, T. Reliability assurance for deep neural network architectures against numerical defects, 2023.

Li, Y., Wang, J., Gao, T., Sun, Q., Zhang, L., and Tang, M. Adoption of machine learning in intelligent terrain classification of hyperspectral remote sensing images. *Computational Intelligence and Neuroscience*, 2020, 2020.

Liyanage, D. C., Hudjakov, R., and Tamre, M. Hyperspectral imaging methods improve RGB image semantic segmentation of unstructured terrains. *15th International Conference Mechatronic Systems and Materials, MSM 2020*, jul 2020. doi: 10.1109/MSM49833.2020.9201738.

Mobiny, A., Yuan, P., Moulik, S. K., Garg, N., Wu, C. C., and Van Nguyen, H. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, 11(1):1–14, 2021.

Naeini, P., Cooper, G. F., and Hauskrecht, M. Obtaining Well Calibrated Probabilities Using Bayesian Binning. *Proceedings of the 2015 SIAM International Conference on Data Mining.*, 2015.

Ojeda, L., Borenstein, J., Witus, G., and Karlsen, R. Terrain characterization and classification with a mobile robot. *Journal of field robotics*, 23(2):103–122, 2006.

Rothrock, B., Kennedy, R., Cunningham, C., Papon, J., Heverly, M., and Ono, M. Spoc: Deep learning-based terrain classification for mars rover missions. In *AIAA SPACE 2016*, pp. 5539. 2016.

Serrà, J., Pascual, S., and Karatzoglou, A. Towards a universal neural network encoder for time series. In *CCIA*, pp. 120–129, 2018.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M. M. A., Prabhat, and Adams, R. P. Scalable bayesian optimization using deep neural networks. *International Conference on Machine Learning*, 2015.

Ugenti, A., Vulpi, F., Domínguez, R., Cordes, F., Milella, A., and Reina, G. On the role of feature and signal selection for terrain learning in planetary exploration robots. *Journal of Field Robotics*, 39(4):355–370, 2022.

Vandapel, N., Huber, D. F., Kapuria, A., and Hebert, M. Natural terrain classification using 3-d ladar data. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pp. 5117–5122. IEEE, 2004.

Vulpi, F., Milella, A., Cordes, F., Dominguez, R., and Reina, G. Deep Terrain Estimation for Planetary Rovers. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2020.

Wang, Z., Yan, W., and Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

Wietrzykowski, J. and Belter, D. Boosting Support Vector Machines for RGB-D Based Terrain Classification. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 8(3):28–34, jan 2014. ISSN 1897-8649. doi: 10.14313/JAMRIS_3-2014/24. URL https://www.jamris.org/index.php/JAMRIS/article/view/318.

Winkens, C., Sattler, F., and Paulus, D. Hyperspectral Terrain Classification for Ground Vehicles. *VISIGRAPP (5: VISAPP)*, pp. 417–424, 2017. doi: 10.5220/0006275404170424.