A CNN Based Vision-Proprioception Fusion Method for Robust UGV Terrain Classification

Yu Chen, Chirag Rastogi, and William R. Norris, *Member, IEEE*

***Abstract*—The ability for ground vehicles to identify terrain types and characteristics can help provide more accurate localization and information-rich mapping solutions. Previous studies have shown the possibility of classifying terrain types based on proprioceptive sensors that monitor wheel-terrain interactions. However, most methods only work well when very strict motion restrictions are imposed including driving in a straight path with constant speed, making them difficult to be deployed on real-world field robotic missions. To lift this restriction, this letter proposes a fast, compact, and motion-robust, proprioception-based terrain classification method. This method uses common on-board UGV sensors and a 1D Convolutional Neural Network (CNN) model. The accuracy of this model was further improved by fusing it with a vision-based CNN that made classification based on the appearance of terrain. Experimental results indicated the final fusion models were highly robust with strong performance, with over 93% accuracy, under various lighting conditions and motion maneuvers.**

***Index Terms*—Deep learning, field robots, machine learning, robot sensing systems, sensor fusion.**

I. Introduction

THE development of precision agriculture has been greatly enhanced due to advances in sensors, robotics, and artificial intelligence. For instance, automatic inspection robots can make assessments of the soil (terrain) quality of farmland and use that information to optimize the scheduling of farming tasks including plowing, watering, and fertilization. Similarly, scout and surveillance robots can be used to explore the site of interest remotely to identify hazardous terrain that might sink and trap or damage heavy machinery. They can provide reference data for excavation in construction and mining operations. These robotics applications all rely on the robot's ability to perceive and characterize terrain. Additional benefits of terrain classification include the development of terrain-aware traction control and motion planning that minimizes fuel consumption for field robots. The knowledge of site-specific terrain characteristics can improve localization accuracy, provide reliable landmarks, and contribute terrain-related information to a high-definition map.

Previous efforts to address robotic terrain classification (RTC) have explored the use of exteroceptive sensors such as cameras [1], [2], 2D and 3D laser scanners [3], [4], ultrasonic and infrared sensors [5], as well as microphones [6]. In other studies, proprioceptive sensors such as Inertial Measurement Units (IMUs) [7] and acceleration or vibration sensors [8], [9] were used to characterize terrain properties. In terms of the classifier, prior studies have investigated both traditional machine learning techniques (i.e., SVM [10]–[12], kNN [13], and Bayes model [7], etc.) and artificial neural networks (i.e., LSTM [6], RNN [9], CNN [14], FCN [8], etc.), which have led to immense successes in solving RTC problems.

The key to a robust RTC solution is to extract terrain characteristics that are invariant to vehicle motions and environmental factors such as noise and unstable illumination. Vision-based methods are well-studied and relatively more accurate but are also susceptible to influences from environmental illumination [15], and motion blur caused by strong vibrations. Proprioceptive solutions are robust to environmental factors and require only common sensors used on most modern robots. However, the solutions are less accurate and highly dependent on the vehicle's motion. Many previous studies [11]–[14], [16], [17] tested their methods when the UGV moved on a straight path with constant speed, while others did not clearly reveal their testing conditions. These restrictions occurred because the vehicles used in these studies were primarily skid-steer drive, which induced additional slippage and vibration to the system while turning. Also, the vehicle's driving speed is proportional to the driving effort and the frequency of the vibrational response [8], [18]. Resultantly, these motion-dependent interferences cloud the judgment of the proprioception classifiers. To overcome the shortcomings in different sensing modalities, previous studies considered leveraging data from multiple sensor sources. For example, [2] used the visual and texture features gathered by a stereo camera, vibration sensors, and a belly-mounted camera to improve the classification performance between three terrain classes. [11] fused the classification decisions made by an image-based SVM and a vibration-based SVM on a 14-class terrain classification problem. And more recently, [12] combined color and three different proprioceptive features to assess terrain in an agriculture setting. While the aforementioned work demonstrated improved performances using multi-model classifiers, they did

Manuscript received February 24, 2021; accepted July 13, 2021. Date of publication August 4, 2021; date of current version August 20, 2021. This letter was recommended for publication by Associate Editor B. Duncan and Editor P. Pounds upon evaluation of the reviewers' comments. This work was supported in part by NSF NRI-2.0. *(Corresponding author: Yu Chen.)*

Yu Chen is with the Department of Mechanical Science & Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: yuc6@illinois.edu).

Chirag Rastogi is with the Department of Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: chiragr2@illinois.edu).

William R. Norris is with the Department of Industrial & Enterprise Systems Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: wrnorris@illinois.edu).

Digital Object Identifier 10.1109/LRA.2021.3101866

not fully address the problem of motion-dependent interference of the UGV nor quantify the robustness of their models under challenging lighting conditions.

In this article, a fast, compact, and motion-robust proprioception-based classifier using common on-board UGV sensors and a 1D CNN model was developed. Previous work like [7], tackled the motion-dependent interference problem using a hierarchical classifier and performing feature selection on hand-crafted features. In contrast to this approach, the CNN model in this study learns the most effective features from the input data directly. Using this unified pipeline, over 89% accuracy was achieved on data sets recorded under arbitrary and continuous vehicle motions with minimum knowledge in the signal processing and proprioception domains. Furthermore, this study showed that by adopting and fusing an image-classifying CNN module pre-trained on a different data set, the classification accuracy was over 98% under appropriate illumination. The same approach maintained a robust performance of over 93% accuracy under challenging lighting conditions.

The following paper is organized as follows: Section II. provides implementation details of a novel terrain classification approach, including the data set collection and feature generation process, as well as the construction and training of the neural network models. Section III. presents the experimental results, and the performance of the neural network models under realistic and challenging conditions are demonstrated and discussed. Section IV. analyzes the efficacy of the use of the proposed derived features. Section V. concludes the study and provides insights towards future work.

II. Methodology

A. Overview

Our proprioception model uses similar procedures from [12] to generate proprioceptive signal inputs. As opposed to the model in [12], which had no knowledge of the vehicle's motion, wheel encoder readings were fed to the model to serve as motion cues. Also, instead of fully relying on the proprioceptive signals of the vehicle body, this study utilized the signals from the left and right sides of the vehicle to help the classifier reject motion-dependent interference. The approach used in this study retained the input signals in their raw form. So that the neural network model can learn the temporal correspondences between the vehicle's motion and proprioceptive feedback. An ablation study was included to further assess the effectiveness of the proposed derived features and provide future research opportunities. A vision-based classifier was built using a pre-trained module and was later fused with the proprioception-based classifier. This study investigated two different fusion mechanics and demonstrated that visual and proprioceptive signals were complementary. In combining their modalities, the classification performance and robustness can be improved. The model's ability to function across different motion and lighting conditions makes it more suitable for real-world field robotic missions, which often require obstacle avoidance maneuvers. All the data used in this study were collected using a mobile robotic platform and processed offline. Once trained, the classifiers can be deployed on the robot. And they can report a terrain label every second during normal operations (0.2–1 m/s).

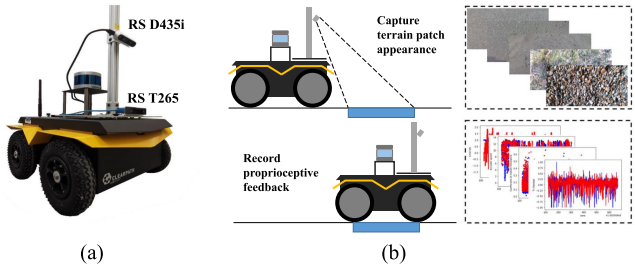

Fig. 1. (a) The Jackal robot platform used for data collection; b) The data collection pipeline.

Fig. 2. Sample images of the different terrain classes used in this study.

B. Data Collection

The Jackal UGV, from Clearpath Robotics, was used as the data collecting platform. This skid-steer four-wheel-drive vehicle, shown in Fig. 1a, comes with an onboard IMU, two DC motors with encoders that measure wheel angular speeds, and current sensors that measure motor current outputs. On each side of the robot, the front wheel and back wheel are jointed with a gearbox and so spin together at the same rate and direction. The IMU provided vehicle attitude measurements in terms of Euler angles, as well as linear acceleration and angular rate of the vehicle body in three Euclidean axes. Camera systems such as the RS D435i and RS T265 were mounted to an aluminum frame attached to the top of the robot platform, as shown in Fig. 1a. While the tracking camera T265, faced forward, the D435i depth camera was positioned and tilted in a way that the camera had a clear visual of the terrain patch. The patch size was $680\ \mathrm{mm} \times 340\ \mathrm{mm}$ with a look ahead distance of 150 mm relative to the chassis of the vehicle. The D435i served as a regular RGB camera for this study and the depth images reconstructed by the D435i were not included in any of the data sets. The T265 camera was used as a Visual-Inertia Odometry (VIO) solution that provided ego-motion estimations of the vehicle.

For this study, six different sensor signals were used: 1) current feedback, 2) wheel encoder readings from each side of the vehicle, 3) 6 DoF VIO measurements from the T265, 4) three-axis linear acceleration, 5) attitude measurement from the IMU, and 6) RGB images taken by the D435i. In addition to the RGB images, all other sensor signals were used as proprioceptive features.

Seven terrain classes were investigated, including asphalt, brick road, grass, gravel, pavement, sand, and coated floors. Fig. 2 shows sample images of these terrain classes.

Training and development data were collected by driving the Jackal robot on each terrain class across different days to account for variance in lighting conditions. Data collection occurred during sunny days from Nov. 2020 to Feb. 2021 in Champaign, Illinois. The average temperatures ranged from 6 to 27 °C. An experienced human operator (one of the authors) was designated for remote control of the Jackal robot during data collection. For each terrain class, multiple independent trials were taken for at least 6 minutes for the *Straight Driving* sessions and at least 10 minutes for the *Remote Control* sessions. The protocols of these sessions were defined as follows:

- *Straight Driving*: the robot was programmed to follow a straight path at the speed of 0.5 m/s and 1 m/s with no reverse driving, stopping, or turning. This session was added to help the neural network understand the proprioceptive baseline for each terrain surface without the interferences induced by aggressive vehicle movements.
- *Remote Control*: the robot arbitrarily drove around the test site to simulate normal robot operations, controlled by a human operator remotely. The operator was instructed to take each path as randomly as possible to reduce the motion bias in the data set. The session conditions included no reverse driving, stopping only if necessary, with a top speed limited to 1 m/s out of safety concerns.

Finally, 600 sec of data from the *Straight Driving* sessions and 1800 sec of data from the *Remote Control* sessions were randomly chosen to form the data set for each terrain class. In total, a data set that contained 7 terrain classes $\times$ (600 s + 1800 s) = 16800 s (4.67 hours) of image-signal data was gathered for training the neural network models. This data set was divided into a training set and a development set with a ratio of [4:1]. The split between the training and development set was a uniform random selection from the shuffled data set. And for both data sets, the number of samples in each class was kept equal to prevent uneven training.

C. Data Processing

1) Proprioceptive Signals: Raw sensor signals were sorted and stored in the form of 1-second data segments as conducted in many previous studies [2], [7]–[11], [13], [17].

Resampling Zero-Order-Hold (ZOH) interpolation and subsampling were used to make sure signals from different sources shared the same sampling frequency (100Hz). After the resampling, one sample of the proprioceptive signal was a $100 \times \mathrm{n}$ vector, where n was the number of signal channels. The use of the ZOH ensured easy transfer when being deployed online.

Data Cleaning All the data segments that contained stopping motions, where vehicle linear velocity $< 0.2\,\mathrm{m/s}$ were removed, as the proprioception-based method was not effective under this condition.

Feature Generation The following proprioceptive features were selected/derived from raw signals: wheel angular speed, motion resistance coefficient, and percentage slip for each side of the vehicle. Also, the linear acceleration of the vehicle body in three Euclidean axes. A total of $2\times(1+1+1)+3 = 9$ proprioceptive features were used as input to the *Proprioception Net*. The details of the feature derivation are provided in Section II.D.

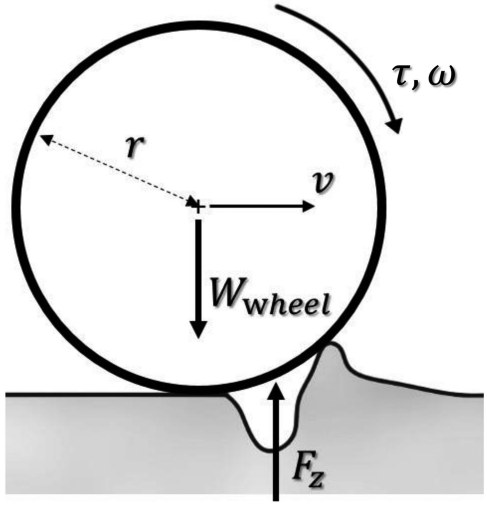

Fig. 3. Motion resistance that arises from wheel-terrain interaction.

2) RGB Images: RGB images were recorded at 25 Hz by the D435i. There were 25 images for each second of proprioceptive signals.

Data Association The approximation was made such that only the first image of a second was used to represent the appearance of the terrain patch that the robot was about to traverse, as shown in Fig. 1b. Since the robot primarily operated with a speed between 0.5 to 1.0 m/s, and each data collection site contained only one terrain class, the assumption was that the fixed frame approximation was sufficient. The correspondence between image and proprioceptive signals did not need to be exact for this application. Ideally, techniques like Simultaneous Localization and Mapping (SLAM) would be applied to provide more accurate image-signal data association to account for misalignments due to differences in speed and steering.

Inverse Perspective Mapping (IPM) IPM was applied to the selected RGB images to transform them into clear and homogeneous bird's-eye view terrain patch visualizations (500×250 pixels). The transformed images were resized to 224×224 pixels to fit the input size of the *Vision Net*.

3) HDF5 File[1]: The processed image-signal pairs were stored in the HDF5 format for fast retrieval. The data pairs were organized by their unique timestamps.

D. Proprioception Net

1) Feature Derivation: Among the proprioceptive features used in this study, the wheel angular speed (rad/s) and the three-axis vehicle linear accelerations (m/s^2) were taken in their raw forms. The derivation and definition of the motion resistance coefficient was adopted from [12]: The motion resistance caused by the deformation of the wheel-terrain interface shifts $\mathrm{F_z}$, the vertical load experienced by the wheel forward with respect to the wheel's geometric center, as shown in Fig. 3. Assuming all the torque generated by the motor is used to overcome the resistance moment, the required driving torque is $\tau = \mathrm{f_r\ r\ F_z}$,

[1]The HDF5 data files used in this study are open-source and available at IEEE DataPort: https://ieee-dataport.org/open-access/jackal-robot-7-class-terrain-dataset-vision-and-proprioception-sensors

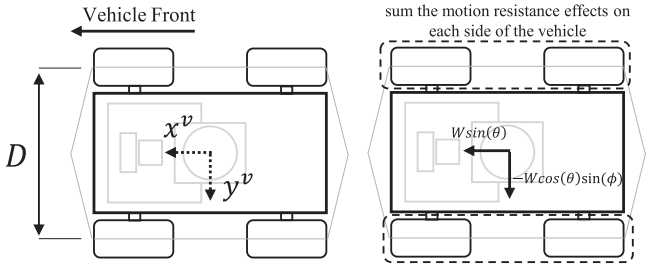

Fig. 4. Free-body diagram of the vehicle system (top-view).

where $\mathrm{f_r}$ is the motion resistance coefficient on a local terrain patch, and r is the radius of the wheel.

From the motor's perspective, the output torque can be roughly estimated by the amount of current, I , drawn by the DC motor: $\tau = \varepsilon\, \mathrm{k_t}\, \mathrm{I}$. ε is the gear ratio and $\mathrm{k_t}$ is the torque constant of the DC motor. Therefore, the motion resistance coefficient can be estimated using current feedback and wheel vertical load:

$$\mathrm{f_r} = \frac{\varepsilon \mathrm{k_t}}{\mathrm{r}} \frac{\mathrm{I}}{\mathrm{F_z}} \tag{1}$$

The wheel vertical load is not a variable that can be directly measured or easily calculated without additional sensors. However, the variable can be approximated using a quasi-static dynamic model, proposed by [12], to neglect complex inertial effects caused by vehicle motion. This approximation is valid as the Jackal robot only operates at a relatively low speed with a maximum of 1 m/s. Applying Newtonian mechanics, the vertical forces in the vehicle coordinate can be expressed. As this study was only concerned with the motion resistance coefficient on each side of the vehicle, as shown in Fig. 4, the vertical forces for the left and right sets of the wheels are:

$$\mathrm{F_{z,l}} = \frac{\mathrm{W}}{2}\cos(\phi)\cos(\theta) - \mathrm{W}\cos(\theta)\sin(\phi)\frac{\mathrm{h}}{\mathrm{D}} \tag{2}$$

$$\mathrm{F_{z,r}} = \frac{\mathrm{W}}{2}\cos(\phi)\cos(\theta) + \mathrm{W}\cos(\theta)\sin(\phi)\frac{\mathrm{h}}{\mathrm{D}} \tag{3}$$

ϕ and θ are the vehicle's roll and pitch angles measured by the onboard IMU. h is the height of the vehicle's center of gravity, and D is the track width. Additional detail on the derivation of (2) and (3) can be found in [12].

The above method is a rough approximation of the motion resistance that the vehicle experiences over a local terrain patch, as the noise in current feedback, the friction loss in the gearboxes, and inertial effects are not accounted for. However, the results showed this estimation was sufficient for use as an indicator of the terrain hardness level.

The percentage slip of the left and right sides of the vehicle are defined as follows:

$$\%\mathrm{slip_l} = 1 - \frac{\omega_{\mathrm{vio,l}}}{\omega_{\mathrm{enc,l}}} \tag{4}$$

$$\%\mathrm{slip_r} = 1 - \frac{\omega_{\mathrm{vio,r}}}{\omega_{\mathrm{enc,r}}} \tag{5}$$

While $\omega_{\mathrm{enc,l}}$ and $\omega_{\mathrm{enc,r}}$ are the left and right wheels' angular speeds measured by the encoders, $\omega_{\mathrm{vio,l}}$ and $\omega_{\mathrm{vio,r}}$ are the left and right wheels' angular speeds calculated from the VIO estimations and skid-steer vehicle kinematics:

$$\omega_{\mathrm{vio,l}} = \left(\mathrm{v_{vio,x}} - \frac{\mathrm{D}}{2}\mathrm{w_{vio,z}}\right) \Big/ \mathrm{r} \tag{6}$$

$$\omega_{\mathrm{vio,r}} = \left(\mathrm{v_{vio,x}} + \frac{\mathrm{D}}{2}\mathrm{w_{vio,z}}\right) \Big/ \mathrm{r} \tag{7}$$

$\mathrm{v_{vio,x}}$ and $\mathrm{w_{vio,z}}$ are the vehicle linear speed on the x-axis and vehicle angular rate on the z-axis, both estimated by the T265.

Layer type	Kernel Size / Stride	Output shape
Conv1D	9*9 / 1	(100, 64)
Max Pooling	2*2 / 2	(50, 64)
Dropout (0.2)	-	(50, 64)
Conv1D	5*5 / 1	(50, 64)
Max Pooling	3*3 / 3	(16, 64)
Dropout (0.4)	-	(16, 64)
Flatten	-	1024
Softmax	-	7

Fig. 5. The *Proprioception Net* model structure.

As opposed to [12], which took the average of the motion resistance coefficient and percentage slip of the whole vehicle to form their proprioceptive features, this study retained the values across the data segment window. To form the desired motion-robustness, the model needed to learn how to distinguish and eliminate motion-dependent interference from noisy proprioceptive features. As a result, the model was given the temporal correspondence between the robot's motions and proprioceptive feedback. This is the reason why this study did not use a Fast Fourier Transform (FFT) when handling acceleration data like many previous studies [8], [10], [11], [13], [17], [18]. Doing so would essentially destroy the underlying temporal information within the data segment.

Finally, at each time step k, a vector $\mathrm{p_k}$ (1×9) that contained nine proprioceptive features $\{\omega_{\mathrm{enc,l}}, \omega_{\mathrm{enc,r}}, \mathrm{accel_x}, \mathrm{accel_y}, \mathrm{accel_z}, \%\mathrm{slip_l}, \%\mathrm{slip_r}, \mathrm{f_{r,l}}, \mathrm{f_{r,r}}\}$ was drawn, where $\mathrm{accel_x}$, $\mathrm{accel_y}$, and $\mathrm{accel_z}$ were the three-axis vehicle linear accelerations. And $\mathrm{f_{r,l}}$, $\mathrm{f_{r,r}}$ were the motion resistance coefficients for the left and right sides of the vehicle. These proprioceptive features informed the model about the vehicle motion, as well as the evenness, slipperiness, and motion resistance of the terrain. Finally, the input to the *Proprioception Net* was a 2D vector p (100×9), where there were 100 time steps in a one-second data segment and with 9 feature channels.

2) Model Building: This study explored using 1D CNN, Multi-branch CNN, CNN with skip-connections, Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), CNN-LSTM [19], and ConvLSTM2D [20] as building blocks of the *Proprioception Net*. For a similar number of parameters with the same order of magnitude, the 1D CNN provided the best results. The best model was a simple two-layer 1D CNN (with 32967 parameters). The detail of the network structure is demonstrated in Fig. 5.

This 1D CNN model performed convolution along the temporal axis, and a Rectified Linear Unit (ReLU) was used at each convolution layer as the activation function. Many previous Natural Language Processing (NLP) studies applied similar practices and achieved great success. Recurrent neural networks like LSTM and GRU were also heavily used in NLP [21]. For this task, while keeping the number of parameters within the same order of magnitude as the 1D CNN model, increasing the

Layer type	Kernel Size / Stride	Output shape
MobileNet v2	-	1280
Dropout (0.25)	-	1280
Dense	-	32
Softmax	-	7

Fig. 6. The *Vision Net* model structure.

number of recurrent units per layer, and the depth of the network did not result in an obvious improvement in accuracy. A similar result was observed in [22] when using LSTM to process haptic signals.

For other hybrid models, the difficulties of structural arrangement and hyper-parameter tuning increased proportionally with their model complexities. Since a simple 1D CNN model achieved a high level of accuracy, the implication is that the features used in this study were very effective for extracting motion-independent terrain characteristics. As a result, there are marginal benefits in using a "deeper" model.

E. Vision Net

To further improve the accuracy of the classification, a *Vision Net* model for capturing the visual characteristics of terrains was constructed. Generally, the training of an image-classifying CNN requires a large amount of data to ensure generalizability and prevent overfitting. To get high performance with a limited amount of data, the *MobileNet v2* module [23] that was pre-trained on ImageNet was adopted. *MobileNet v2*, a compact and efficient CNN architecture developed by Google, was designed for image classification on a device with limited computational power like single-board computers or mobile phones. ImageNet is a large image database that contains multi-millions of image samples.

The weights of this pre-trained CNN module were frozen to reduce the number of trainable parameters (41223) and save training time. Numerous data augmentation techniques were applied, such as horizontal and vertical flipping as well as random rotation, and random brightness on the training data. Using the model shown in Fig. 6, the *Vision Net* achieved over 98% accuracy on both training and development sets.

F. Fusion Net

The hypothesis was made that by fusing the proprioception- and vision-based model, higher levels of performance would be achieved due to their complementarity. To better understand the fusion mechanics, two different fusion schemes were explored, namely the feature map-level and decision-level fusion.

1) Fusion at the Feature Map-Level: First, the Softmax (the last) layer of the trained Proprioception Net was removed. As a result, the output of this module was a 1×1024 feature vector. This module was denoted as a Proprioception CNN. The pre-trained MobileNet v2 module was used and denoted as a Vision CNN. The weights of these two modules were frozen, and they were used as feature extractors to process the image-signal hybrid input. The activations (ReLu) from these two CNN modules were concatenated. The complete pipeline of feature map level fusion can be found in Fig. 7.

To simulate the possible illumination conditions in test time, additional and aggressive data augmentation techniques were applied on images during training, including random channel shift, motion blur, and blackout.

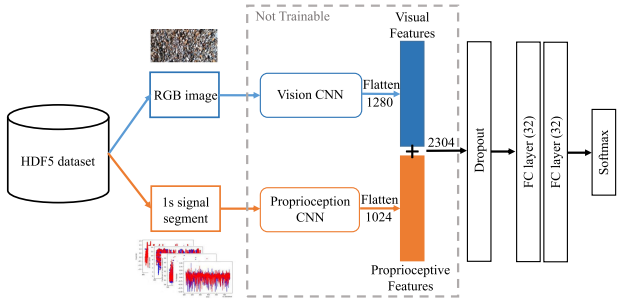

Fig. 7. The feature map-level fusion model pipeline.

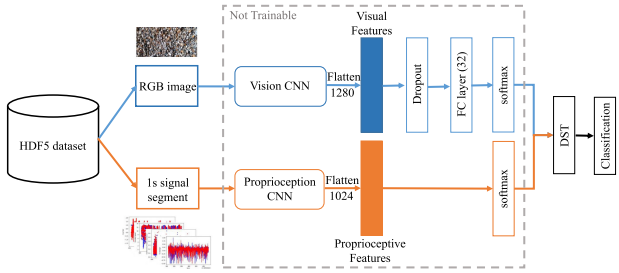

Fig. 8. The decision-level fusion model pipeline.

2) Fusion at the Decision-Level: In this model, the learned weights of the Vision Net and the Proprioception Net were directly transferred. These two networks ran in parallel and made independent predictions (decisions) at their Softmax layers. The prediction outcomes were received by a fusion operator, denoted as a DST, which output the final classification, as shown in Fig. 8.

DST is the acronym for the Dempster–Shafer Theory [24]. It was used in this model to solve the problem of combining multiple belief functions. DST was used in a previous fusion model [25] for classifying sound signals with great results.

In this study, two distinct sets of evidence, appearance and proprioceptive feedback were used to estimate the belief about an event. The belief was the likelihood that a certain terrain type was detected. For this application, Dempster's rule of combination was an appropriate fusion operator. Specifically, the operator was defined as follows:

$$\begin{aligned} \mathrm{m}_{1,2}(\mathrm{A}) &= (\mathrm{m}_1 \oplus \mathrm{m}_2)(\mathrm{A}) \\ &= \frac{1}{1-\mathrm{K}} \sum_{\mathrm{B}\cap\mathrm{C}\,=\,\mathrm{A}\neq\emptyset} \mathrm{m}_1(\mathrm{B})\mathrm{m}_2(\mathrm{C}) \end{aligned} \tag{8}$$

$\mathrm{m}_1(\mathrm{B})$ and $\mathrm{m}_2(\mathrm{C})$ were the mass functions of the *Vision Net* and the *Proprioception Net,* the outputs of Softmax layers. And $\mathrm{m}_{1,2}(\mathrm{A})$ was the jointed mass function that encoded the belief distribution of the class labels, and it satisfied the constraint $\mathrm{m}_{1,2}(\emptyset) = 0$. $\mathrm{K} = \sum_{\mathrm{B}\cap\mathrm{C}\,=\,\emptyset} \mathrm{m}_1(\mathrm{B})\mathrm{m}_2(\mathrm{C})$ indicated the level of conflict between the two mass functions, and it was used for normalizing the mass functions.

Since all the transferred weights were frozen and the DST layer did not contain any parameters, this fusion model did not require training.

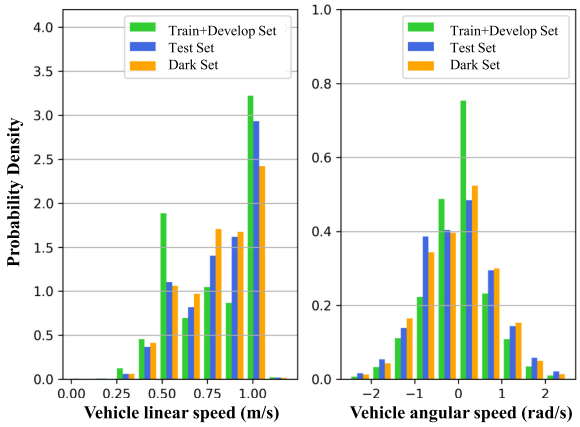

Fig. 9. The probability density histograms of vehicle linear and angular speed for the Training / Development Set; the Test Set; and the Dark Set.

III. Validation Results

1) Data Set: To validate the performance and robustness of the method, two independent sets of testing data were gathered. First, a test set was collected using the same protocol described in section II.B. For all testing data, only the Remote Control sessions were included. This data set, denoted as the Test Set, had 4020 samples (1.1 hours) from seven terrain classes collected under lighting conditions similar to the training set. The other data set, denoted as the Dark Set, had 3912 samples (1.09 hours) and was also collected using the same protocol, only in this case the data were recorded under natural twilight conditions.

Fig. 9 provides a visualization of the vehicle linear and angular speeds collected for each data set. The probability density distributions were similar across all data sets, except that the training and development sets contained more samples at 0.5 m/s and 1 m/s due to the addition of the Straight Driving sessions. This meant the data collected in the Remote Control sessions were sufficiently random (uniform). It can be observed that the human operator tended to have a preferred range of driving and turning speeds when requested to drive the robot with arbitrary motions.

Moreover, as data augmentation techniques such as random brightness were used during training, darker images were not completely unfamiliar to the two *Fusion Net* models. It is possible that the *Fusion Net* models may have been successful with the Dark Set by simply trusting only the Proprioception CNN whenever the brightness of the image was below a certain threshold. To exclude this possible workaround and test how well the models generalized to illumination conditions that were not accounted for during training, the images in the Test Set were augmented under different conditions, and two simulated data sets were created: the Sun Set and the Fog Set. A library called *albumentations* was used to generate realistic overexposed and foggy images. Samples of these test sets can be found in Fig. 10.

Note that the inclusion of the synthetic test sets was not to suggest the models tested had the exact performance under these illumination conditions in the real world. The test results on synthetic sets served as indicators of model robustness outside of the training conditions.

2) Experiment Analysis: Fig. 11 demonstrates the confusion matrices of four models tested under the two testing sets. It was expected that the *Proprioception Net* struggled to tell the differences between asphalt, pavement, and coated floor since they are all relatively flat and solid [8]. Most of the confusion between these three classes occurred when the robot was moving straight, where the terrain characteristics of these classes were difficult to distinguish. Additionally, if the robot turned (skidded) on asphalt or pavement, the robot induced strong vibrations due to high friction. The difference between asphalt and pavement was more subtle in the eyes of the *Proprioception Net*. The darker lighting condition in the Dark Set did not hinder the accuracy of the *Proprioception Net* since it did not rely on visual information. However, the *Vision Net* failed with the Dark Set and tended to guess all input samples to be either pavement or a coated floor. This follows from the general knowledge that image CNNs are sensitive to lighting conditions. The two *Fusion Net* models had higher levels of overall accuracy (up to 99.38%) on the Test Set and the Dark Set as shown in Fig. 11. Leveraging two complementary modalities, the feature map-level fusion model demonstrated strong performance on the two testing sets. The decision-level fusion model was able

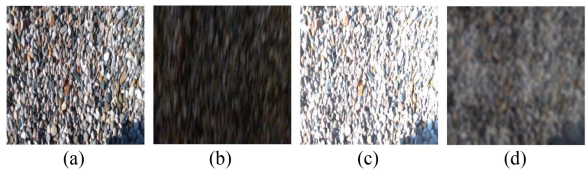

Fig. 10. Sample images of gravel in a) the Test Set, b) the Dark Set, c) the Sun Set, and d) the Fog Set.

Confusion Matrix for Proprioception Net tested under Test Set (overall: 90.18%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	640	0	0	6	0	1	9
GRS	3	374	0	10	0	55	0
GRL	0	8	500	1	4	0	0
PMT	128	12	0	433	0	25	31
SND	0	0	3	1	600	0	0
BRK	0	23	7	1	0	526	0
CFL	63	2	0	2	1	0	551

Confusion Matrix for Proprioception Net tested under Dark Set (overall: 92.78%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	494	5	0	6	0	32	1
GRS	0	498	7	1	0	5	0
GRL	0	0	529	0	0	0	0
PMT	97	0	0	436	0	1	35
SND	0	0	1	0	565	0	4
BRK	8	23	5	0	0	528	0
CFL	32	1	2	20	2	1	573

Confusion Matrix for Vision Net tested under Test Set (overall: 99.76%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	656	0	0	0	0	0	0
GRS	0	442	0	0	0	0	0
GRL	0	0	512	0	1	0	0
PMT	0	0	0	623	6	0	0
SND	0	0	0	0	604	0	0
BRK	0	2	0	0	0	554	1
CFL	0	0	0	0	0	0	619

Confusion Matrix for Vision Net tested under Dark Set (overall: 33.33%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	0	0	0	537	0	0	1
GRS	0	103	0	0	0	0	408
GRL	7	271	19	2	3	0	227
PMT	0	0	0	347	0	0	222
SND	0	0	0	225	1	0	344
BRK	0	0	0	1	0	541	22
CFL	0	0	0	300	0	0	331

Confusion Matrix for feature map-level Fusion Net tested under Test Set (overall: 99.38%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	656	0	0	0	0	0	0
GRS	0	442	0	0	0	0	0
GRL	0	0	512	1	0	0	0
PMT	23	0	0	604	0	2	0
SND	0	0	0	1	603	0	0
BRK	0	0	0	0	0	557	0
CFL	0	0	0	0	0	0	619

Confusion Matrix for feature map-level Fusion Net tested under Dark Set (overall: 93.81%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	**432**	**7**	**0**	**44**	**0**	**44**	**11**
GRS	**0**	**507**	**0**	**1**	**0**	**1**	**2**
GRL	**0**	**2**	**526**	**0**	**1**	**0**	**0**
PMT	**23**	**0**	**0**	**476**	**0**	**0**	**70**
SND	**0**	**0**	**0**	**0**	**567**	**0**	**3**
BRK	**2**	**0**	**0**	**0**	**0**	**562**	**0**
CFL	**5**	**0**	**0**	**23**	**3**	**1**	**599**

Confusion Matrix for decision-level Fusion Net tested under Test Set (overall: 98.95%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	655	0	0	1	0	0	0
GRS	0	442	0	0	0	0	0
GRL	0	0	512	1	0	0	0
PMT	24	2	0	586	0	16	1
SND	0	0	0	0	604	0	0
BRK	0	0	0	0	0	557	0
CFL	0	0	0	1	0	0	618

Confusion Matrix for decision-level Fusion Net tested under Dark Set (overall: 93.68%)

	ASP	GRS	GRL	PMT	SND	BRK	CFL
ASP	413	0	0	91	0	19	15
GRS	0	508	2	1	0	0	0
GRL	0	2	526	0	1	0	0
PMT	28	0	0	499	0	0	42
SND	0	0	0	0	562	0	8
BRK	0	0	0	0	0	564	0
CFL	2	0	0	36	0	1	592

ASP	GRS	GRL	PMT	SND	BRK	CFL
Asphlat	Grass	Gravel	Pavement	Sand	Brick	Coated Floor

Fig. 11. The confusion matrix of Orange: the *Proprioception Net;* Blue: the *Vision Net;* Green: the *Fusion Net* at feature map-level; and Yellow: the *Fusion Net* at decision-level, tested under the Test Set and the Dark Set.

	Test Set	Dark Set	Sun Set	Fog Set	Avg.
Proprioception Net	90.18%	92.78%	90.18%	90.18%	90.83%
Vision Net	**99.76%**	33.33%	88.70%	72.73%	73.63%
Feature map Fusion	99.38%	**93.81%**	95.55%	**96.33%**	96.27%
Decision Fusion	98.95%	93.68%	**96.73%**	96.22%	**96.40%**
C. Weiss et al. PSVM [11]	74.39%	70.51%	74.39%	74.39%	73.42%
C. Weiss et al. VSVM [11]	62.13%	22.66%	24.23%	37.17%	36.55%
C. Weiss et al. FSVM [11]	79.80%	52.11%	52.59%	64.72%	62.31%
G. Reina et al. PSVM [12]	58.52%	55.52%	58.52%	58.52%	57.77%
G. Reina et al. VSVM [12]	84.52%	17.40%	41.46%	60.81%	51.05%
G. Reina et al. FSVM [12]	90.25%	26.52%	48.18%	72.22%	59.29%

Fig. 12. The classification accuracy comparison table, where Blue: models from this study; Orange: [11]; Green: [12]. "PSVM", "VSVM", and "FSVM" stand for proprioception SVM, vision SVM, and fusion SVM respectively. The term accuracy refers to weighted overall accuracy, and the **bold** numbers are the highest accuracy in each column.

to strategically shift its belief between the two CNN models and achieved a similar level of performance. Some samples which were indistinguishable by the *Proprioception Net* and the *Vision Net* were correctly classified with both fusion models. For the Dark Set, even when almost no visual information was available for classification, both fusion models maintained a higher level of performance compared to the *Proprioception Net* and the *Vision Net*. These results confirm the earlier hypothesis that by fusing the proprioception- and vision-based models, higher levels of performance can be achieved due to their complementarity.

This complementarity also implies the addition of a vision-based model may be used to extend the classification range of this pipeline to slower vehicle motions ($<$0.2 m/s). As mentioned earlier, the *Proprioception Net* was not effective when the vehicle's linear velocity was too small, as not enough distinguishable proprioceptive excitations were generated under this condition. However, as the vehicle motion gets closer to a complete stop, a vision-based model is expected to perform better with the absence of velocity-induced motion blur under appropriate illumination conditions. One can force the pipeline to only use the *Vision Net* when the vehicle's linear velocity is below a certain threshold via a simple "if" statement.

Fig. 12 provides an accuracy comparison between the models developed in this study as well as the models from [11] and [12]. The methods in [11] and [12] were recreated with the best effort and trained using the same training set. The test results of [11] and [12] are presented as baselines. As the Sun Set and the Fog Set were synthesized using the Test Set, the test accuracies for all the proprioceptive-based models were the same as the Test Set. Fig. 12 demonstrates that the fusion models developed in this study have consistently high performance (over 93%) even in illumination conditions that were not accounted for during training, namely the Sun Set and the Fog Set. It can also be observed that even though [11] and [12] did improve the classification accuracy in the Test Set by fusing the SVM models, this trend did not generalize well to other test sets. In some cases, the accuracy of the proprioception SVM was much higher (up to a 29% difference) than the fusion SVM, which implies the fusion mechanics used in [11] and [12] could not effectively shift their beliefs between the proprioception and the vision models according to the illumination conditions.

Fig. 13 provides information on the inference time and storage size of the models in this study. The *Proprioception Net* was very fast and compact, while the other models required over 6 ms to process a one-second data pair. As the collection of proprioceptive feedback required a whole second, comparatively, a millisecond-level inference time indicated a classification rate near 1 Hz is possible once the models are deployed online. Moreover, storage sizes for all the models can easily fit in the memory of a single-board computer like a Raspberry Pi or an NVidia TX2.

	Inference Time	Storage Size
Proprioception Net	~1 ms	0.8 Mb
Vision Net	6.83 ms	12.1 Mb
Feature map Fusion	6.84 ms	12.8 Mb
Decision Fusion	6.95 ms	12.4 Mb

Fig. 13. The Time profiling and storage comparison table; tests were performed using an AMD Ryzen7 4800H Processor (2.9 GHz), no GPU was used.

	Develop Set	Test Set	Dark Set
Proprioception Net	93.21%	89.54%	92.90%
Raw Proprioception Net	91.49%	88.86%	92.54%

Fig. 14. The classification accuracy comparison table for the ablation study.

IV. Ablation Study

An ablation study was conducted to help understand the efficacy of the use of the derived features mentioned in Section II.D: Instead of using the features adopted from [12], a Proprioceptive CNN model, denoted as a *Raw Proprioception Net*, that takes the raw proprioceptive sensor signals as inputs was built and compared to the *Proprioception Net*. To ensure a fair comparison, both networks were trained under the same conditions. Two pairs of training and development sets were generated using the same random seed such that they had the same data frames as their counterparts. One pair of the training and development sets used in the ablation study was formed following the same procedures described in Section II.C, while the other excluded the feature derivation procedure and packed the proprioceptive signals untreated (i.e., $\omega_{\mathrm{enc,l}}$, $\omega_{\mathrm{enc,r}}$, $\mathrm{accel_x}$, $\mathrm{accel_y}$, $\mathrm{accel_z}$, $\mathrm{v_{vio,x}}$, $\mathrm{w_{vio,z}}$, $\mathrm{I_l}$, $\mathrm{I_r}$, ϕ, θ; where $\mathrm{I_l}$ and $\mathrm{I_r}$ were the current feedback from the left and right DC motors). Note that in (1, 2, 3), current feedback $\mathrm{I_l}$ and $\mathrm{I_r}$, and vehicle attitude ϕ and θ were used to compute the motion resistance coefficients $\mathrm{f_{r,l}}$ and $\mathrm{f_{r,r}}$. Therefore, by omitting these computations, the input space of the Proprioceptive CNN increased from 100×9 to 100×11. The network structure of the *Raw Proprioception Net* was the same as the *Proprioception Net* except for a larger input layer to accommodate for the change in input space. As a result, the total number of trainable parameters was marginally larger (35527 parameters in total).

Fig. 14 shows the accuracy comparison between the *Raw Proprioception Net* and the *Proprioception Net*. As mentioned earlier, the training and development sets used in the ablation study were newly generated. The data composition was slightly different from before due to the random selection procedure described in Section II.B, which caused accuracy fluctuations for the *Proprioception Net* (less than $\pm$0.64% compared to Fig. 12) in both the Test Set and the Dark Set.

As shown in Fig. 14, the performance margin of using the proposed derived features was small – about 1.72% in the development set. The accuracy of the *Raw Proprioception Net* in the Test Set and Dark Set also suggested the 1D CNN structure

given in Fig. 5 can be trained to process raw proprioceptive signals directly and achieve a similar level of accuracy as the one using the proposed derived proprioceptive features. This ablation study implied that the proposed data processing pipeline can be further simplified in a follow-up study.

V. Conclusion and Future Work

This study successfully developed a fast, lightweight, and motion-robust proprioception-based terrain classification method using a CNN model and signals from common on-board UGV sensors. The strong performance (over 89.54%) and the robustness of this method were demonstrated by testing it under data sets that contained arbitrary vehicle motions. Furthermore, it was shown that the proprioception and vision-based models were complementary. By fusing the two models, a higher level of accuracy (up to 99.38%) was observed in both the feature map-level and decision-level fusion models. Four distinct lighting conditions were used to validate the generalizability of the fusion models. The validation results (over 93.68% accuracy) showed the fusion models in this study can strategically cope with different environmental illumination without human interference and achieve significantly higher accuracy than the baseline methods. The decision-level fusion model achieved the highest average accuracy (96.40%) over the four test sets. Compared to the feature map-level fusion, the decision-level fusion was more stable under different test conditions and did not require further training. It is worth noting that the *Proprioception Net* and *Vision Net* were independently trained before their integration into the decision-level fusion model as frozen layers. This suggested the association between the proprioceptive signals and the image was not critical to the success of the decision-level fusion in this study. Additionally, the time profiling indicated the online deployment of the models in this study was possible. Lastly, an ablation study showed the proposed data processing pipeline can be further simplified by removing the use of all manually derived features. Doing so permitted the proposed method to be less model-dependent, as the knowledge of the vehicle specifications, dynamic, and kinematic models were no longer required.

In the future, more human operators should be recruited to enrich the variety of driving motion in the data sets, or a program can be developed to automate the data collection process. Data augmentation on proprioceptive signals should be performed to further improve the accuracy. A larger range of vehicle motions and data collection sites should be included in the test sets to further validate the generalizability of the models. Feature importance analysis would need to be conducted to reduce the number of required signals and the size of the model. More accurate data association techniques like SLAM should be applied for online deployment. And other fusion mechanics should be investigated so that the correlations between proprioceptive signals and images can be effectively utilized. Moreover, as this method is vehicle-specific, the portability and scalability of the method should be addressed in future studies.

References

[1] R. Marani, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Auton. Robots*, vol. 18, no. 1, pp. 81–102, 2005.

[2] I. Halatci, C. A. Brooks, and K. Iagnemma, "Terrain classification and classifier fusion for planetary exploration rovers," in *Proc. IEEE Aerosp. Conf.*, 2007.

[3] J. C. Andersen, M. R. Blas, O. Ravn, N. A. Andersen, and M. Blanke, "Traversable terrain classification for outdoor autonomous robots using single 2D laser scans," *Integr. Comput. Aided. Eng.*, vol. 13, no. 3, pp. 223–232, 2006.

[4] K. M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard, "Improving robot navigation in structured outdoor environments by identifying vegetation from laser data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, 2009, pp. 1217–1222.

[5] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *J. F. Robot.*, vol. 23, no. 2, pp. 103–122, 2006.

[6] A. Valada and W. Burgard, "Deep spatiotemporal models for robust proprioceptive terrain classification," *Int. J. Rob. Res.*, vol. 36, no. 13–14, pp. 1521–1539, 2017.

[7] D. Tick, T. Rahman, C. Busso, and N. Gans, "Indoor robotic terrain classification via angular velocity based hierarchical classifier selection," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3594–3600.

[8] C. Bai, J. Guo, and H. Zheng, "Three-dimensional vibration-based terrain classification for mobile robots," *IEEE Access*, vol. 7, pp. 63485–63492, 2019.

[9] S. Otte, C. Weiss, T. Scherer, and A. Zell, "Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2016, vol. 2016-June, pp. 5603–5608.

[10] C. Weiss, H. Fröhlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2006, pp. 4429–4434.

[11] C. Weiss, H. Tamimi, and A. Zell, "A combination of vision- and vibration-based terrain classification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, 2008, pp. 2204–2209.

[12] G. Reina, A. Milella, and R. Galati, "Terrain assessment for precision agriculture using vehicle dynamic modelling," *Biosyst. Eng.*, vol. 162, pp. 124–139, 2017.

[13] E. Coyle and E. G. Collins, "A comparison of classifier performance for vibration-based terrain classification," in *Proc. Army Sci. Conf.*, 2008, pp. 1–4.

[14] R. Gonzalez and K. Iagnemma, "Deep terrame chanics: Terrain classification and slip estimation for ground robots via deep learning," 2018, arXiv.

[15] C. Spiteri, S. Al-Milli, Y. Gao, and A. Sarrionandia De León, "Real-time visual sinkage detection for planetary rovers," *Rob. Auton. Syst.*, vol. 72, pp. 307–317, 2015.

[16] G. Reina and R. Galati, "Slip-based terrain estimation with a skid-steer vehicle," *Veh. Syst. Dyn.*, vol. 54, no. 10, pp. 1384–1404, 2016.

[17] E. Coyle *et al.*, "Vibration-based terrain classification for electric powered wheelchairs," in *Proc. 4th IASTED Int. Conf. Telehealth Assist. Technol. Telehealth/AT 2008*, 2008, no. January 2014, pp. 139–144.

[18] E. G. Collins and E. J. Coyle, "Vibration-based terrain classification using surface profile input frequency responses," *Proc. - IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3276–3283.

[19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long short-term memory, fully connected deep neural networks," in *Proc. ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, vol. 2015-August, pp. 4580–4584.

[20] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, 2015, pp. 802–810.

[21] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, arXiv.

[22] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," in *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, 2016, pp. 536–543.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018, arXiv, pp. 4510–4520.

[24] K. Sentz and S. Ferson, "Combination of evidence in Dempster- Shafer Theory," *Contract*, no. April, p. 96, 2002.

[25] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors (Switzerland)*, vol. 19, no. 7, pp. 1–15, 2019.