ROB 590: Martian Terrain Classification: An Auditory Approach

Dhyey Manish Rajani
Robotics
University of Michigan Ann Arbor
Ann Arbor, USA
drajani@umich.edu

I. Introduction

The Perseverance Rover successfully landed on the red planet on February 28, 2021, and has been given the mission-critical task of looking for evidence of ancient life and gathering specimens of rock and regolith (unconsolidated and heterogeneous rock and soil deposits covering solid rock) for potential return to Earth. This robotic exploration vehicle must optimize its planned trajectory to safely and efficiently navigate a multitude of challenging terrains while operating in uncharted geographical territories. Depending on the nature of the landscape, the rover faces many ha as it travels autonomously across various terrains. A vision-based mapping system that enables accurate terrain identification must be integrated with the rover's navigation stack.

The technique of self-supervised learning allows the autonomous labelling of training data by exploiting the interrelationships between various modalities of input signals either from proprioceptive sensors (detect terrain characteristics through robot interaction with environment) and/or from exteroceptive sensors (detect the terrain well-before in advance hence avoiding the robot's direct interaction with the terrain). Furthermore, the unsupervised aspect in audio classification eliminates the need to manually label audio samples. We advance towards long-term learning for visual terrain classification by leveraging the fact that the distribution of terrain sounds does not depend on the visual appearance of the terrain. This enables us to employ our trained audio terrain classification model in previously unseen visual perceptual conditions to automatically label patches of terrain in images, in a completely self-supervised manner. The visual classification model can then be fine-tuned on the new training samples by leveraging transfer learning to adapt to the new appearance conditions. We, therefore, attempt to leverage variegated data types to autonomously classify the weakly labeled data generated by [1] from the video/image feed rather than training a data-hungry neural network architecture.

In this study, we aim to develop an unsupervised audio-based terrain identification and labelling system that will self-supervise a vision-based image segmentation network for pixel-wise semantic segmentation of images [1] in a multi-modal terrain classification system. The audio classifier learns from the audio feed collected from the rover-terrain interaction by correlating the local audio features (in current sound clip) with the global features (throughout the full audio recording). This classifier once trained can further be used for visual segmentation of terrain elements which the robot traverses. We have constricted our focus within this project towards audio processing and unsupervised & self-supervised audio based terrain classification due to the unavailability of the martian terrain ground truth image dataset as of now. We plan to develop a robust audio terrain classification system corresponding to its image counterparts so that the image synthesis and segmentation module can be easily connected to the audio classifier module for self-supervision and data labelling, upon its availability. We also test our results on the [1] paper's original dataset, by using this dataset as a baseline or ground truth. We manually scrap the data as required by our unsupervised and/or self-supervised algorithm to show the validity of our audio-based terrain classification framework.

II. Related Work

The unavailability of labelled data for visual segmentation and classification necessitates the use of self-supervised (or in some cases unsupervised) learning in the design of autonomous navigation system. In recent years, substantial research has been propagated on the self-supervised terrain classification front and its applicability for assessing terrain attributes for robotic systems.

For instance, an obstacle and path identification method for outdoor fields is presented by [2] and is solely based on stereo-camera aided self-supervised learning. Another motivating work was shown by Konolige et al. [3] in long-term terrain categorization by leveraging a deep belief network on the Learning Applied to Ground Robots robotic platform. They primarily used labels from a short-range stereo-vision segmentation model to train a deep belief network on landscape images to distinguish between several geographic features like surface topography & contours, ground texture & dissections, vicinal obstacles etc. They also produced pixel-based texture metrics for each pixel using the robotic platform.

These methods have only been tested on limited terrain categories and incorporation of proprioceptive sensors can enhance accuracy. Hence, to alleviate this limitation a multi-sensor system has been implemented by [4], which utilized a vision-based sensor for terrain categorization in the robot's

frontal view-field and a vibration sensor (proprioceptive sensor) to identify the current terrain traversed by the rover. For assessing the terrain mobility for mobile robots, [5] developed a learning-free method which identifies the correlations between image scans (exteroceptive) and acceleration inputs (proprioceptive) by contextual causal analysis.

Most of the work done so far focuses on employing a proprioception-aided classiifers like vibration sensors. Usually terrain misclassification in these cases can be attributed to mechanical imperfections in the test setup, making these kind of self-supervised methods brittle to use. The advent of deep learning-based techniques for environment segmentation have recently been explored. One of the formulation by Hirose et al. [6] proposed a semi-supervised deep learning solution to image traversability assessment. They use Generative Adversarial Networks to derive probabilistic decisions to determine the terrain safety for robot traversal. Barnes et al. [7] employs vehicular trajectory for self-supervised categorization within urban settings to determine drivable zones. Trajectory coordinates are used as a data labels to mark pixels in a scene. Due to a lack of supporting semantic knowledge, their technique is constricted to local scenes. To tackle this limitation, [8] provided a self-supervised methodology to predict future outcomes of a short-range proximity sensor by analyzing the current outputs of a long-range sensor, hence effectively training a Convolutional Neural Network (CNN) to classify obstacles in camera images.

While recent research has shown progress in self-supervised terrain classification, they lack 2 basic needs to enable autonomous navigation. First, the current systems either fail to successfully generalize to other classes [7] and second, they tend to use brittle proprioceptive modalities instead of exteroceptive ones for unsupervised labelling. Motivated by these drawbacks we try to perfect the audio (exteroceptive) side of the self-supervised learning, by creating an unsupervised audio terrain classifier which can also be used in self-supervised frameworks directly. Resultantly, we depict how audio terrain categorization system can be used in an unsupervised manner to obtain accurate results for classification.

III. Technical Approach

A. Data and its extraction

The martian terrain audio obtained from the Perseverance rover's traversal in the Jezero crater is 16 minutes of raw mixture of various terrain sounds having a sampling rate of 48000 Hz. Our assumption is that the martian audio is supposedly comprised of 4 terrains viz. Big rock, Bedrock, Sand and Soil [10]. The audio data we get from Zürn et al.'s [1] dataset has a sampling rate of 44100 Hz and has mixed audio clips between 12 to 16 minutes, for 14 different trajectories comprising of 2 to 5 terrains viz. Gravel, Asphalt, Grass, Cobblestone and Parking Lot.

The audio stream that we receive either from the perseverance rover or from Zürn et al.'s [1] dataset doesn't have clearly demarcated sections which signify the presence of different terrains in the corresponding audio clips, which makes it challenging to train a neural network for classification tasks. So we had to manually scrap the dataset from Zürn et al.'s [1] database so that we can train and test the validity of baseline and source separation approaches(discussed ahead). To do so, we leveraged time-synchronized audio-visual data in Zürn et al.'s [1] dataset to create terrain specific audio segregated dataset. In order to ensure absolute ground truth we did not use any probabilistic methods this dataset creation.

B. Audio Preprocessing

The various terrain interaction audio clips, used here, are of different duration; in order to avoid making the system too data specific and less generalizable, we transform the audio clips by data augmentation to equal lengths. [11]

We tested with a multitude of data augmentation techniques like spectrogram inversion, frequency masking, low & high pass filtering, etc. but many of these techniques tend to change the audio completely in long sequences, consequentially interchanging similar audio clips. Hence, to maintain the audio discreetness we select following data augmentation techniques [12]:

- Time Stretching: involves changing the duration of an audio clip while maintaining its pitch and tempo, which is done by randomly expanding/compressing waveforms.
- Pitch Shifting: involves changing the pitch of an audio clip while maintaining the tempo. This is achieved by either increasing or decreasing the frequency of the audio waveform randomly. [11]
- Noise Injection: involves adding random noise to the audio signal to simulate different recording conditions or to increase the model's adaptability to noisy environments.
- Resampling: involves changing the sampling rate of the audio signal, which is useful in simulating different recording conditions.

We then use these aforementioned filters randomly to equalize the duration all the terrain audio clips to that of the largest terrain audio clip amongst all. Now in order to make the terrain specific audio clips ready for the feature extraction phase in the neural network we need to make the audio signal discrete in nature. According to Zürn et al.'s [1] dataset each video frame is time-synchronised with 0.5 secs. of audio, so we need to split each terrain's audio clip into numerous audio bits, each having a duration of 0.5 secs. (2.95 secs. in case of martian audio). This is primarily done for providing input for triplet sampling.

We first divide the audio clips into smaller audio bits, and then we use Mel-Frequency Cepstral Coefficients (MFCCs) to transform each bit into a low-dimensional equivalent [13]. Since raw audio signals from terrains are frequently high-dimensional and nosiy in nature, it is difficult to extract beneficial details from the waveform. The identification of seasonal variations in the unprocessed audio signal is a challenging for feature extraction. Hence to simplify this MFCCs provide a more condensed representation of the audio signal by using the Mel scale which correlates the supposed sensed frequency of the original tone in its low-dimension to its actual measured frequency in order to overcome the fundamental challenges

in processing noisy audio data for pattern recognition. Low frequency sounds are more easier for humans to differentiate based on slight pitch changes rather than high frequency sounds. This scale's incorporation makes the algorithm feature representation akin to the human percetion of sound. This aspect of biomimicking is the main motivation for selecting MFCC. The steps of MFCC conversion are:

- Pre-emphasis: Apply a filter to the audio signal to amplify high-frequency components.
- Framing: Then the signal is divided into short overlapping frames and a Hamming window function is applied to each frame to avoid spectral leakage.
- Short-Time Fourier Transform (STFT): Then the Fourier transform is computed for each windowed frame to obtain a time-frequency representation of the signal.
- Mel-frequency Filterbank: Then we a set of bandpass filters is applied to group the STFT coefficients into a smaller number of frequency bands. and then subsequently the logarithm of the filterbank energies is taken to compress the dynamic range of the signal, following this a Direct Cosine Transform is applied to the log-filterbank energies to obtain a set of cepstral coefficients. The first 12-13 cepstral coefficients are retained as the final set of MFCC features which represent the most significant spectral information.

C. Baseline

In this section we will describe the fundamental architecture used to validate our approaches. Since, we don't have separated martian audio data needed for training neural network, we use the separated audio data we get from Zürn et al.'s [1] dataset. Now different trajectories have different terrains or number of classes, we have trained our baseline on trajectory no. 1 (which has 4 classes), trajectory no. 2 (which has 2 classes) and trajectory no. 12 (which has 5 classes). Let us consider trajectory no. 12 for demonstration, which comprises of all 5 classes of Zürn et al.'s [1] dataset as described above in the data section. So we now have labelled audio segments for different terrains, generated from the original mixed audio.

We will be using Siamese Neural network with Triplet loss as our core neural net architecture for the audio classification task. The choice of this specific architecture is inspired from Zürn et al.'s [1] paper. Since the standard Siamese neural network with contrastive loss cannot handle unbalanced classes and suffers from 'dead' embedding problem [13] we use triplet loss (which is one possible justification for using this architecture from our experience). Triplet loss involves selecting three data points: an anchor, a positive, and a negative example. The anchor and positive example belong to same class, while the negative example is dissimilar to the anchor. The triplet loss function computes the distance of anchor from positive and negative samples. The goal is to minimize the anchor-positive distance such it is relatively less than anchor-negative one. Now For some distance on the embedding space d, the loss of a triplet (anchor(a), positive(p) , negative(n)) is given by:

$$L = \max(d(a,p) - d(a,n) + margin, 0) \tag{1}$$

This equation is used as a cost function which the network tries to minimize, pushing d(a,p) to 0 and d (a,n) to be greater than d(a,p) + margin . As soon as 'n' becomes an "easy negative", the loss becomes zero.

In order to make the neural network work, we need to feed it the inputs by creating triplets. Now triplets can usually be classified into three types, which are as follows:

- easy triplets: triplets which have a loss of zero, because positive closer to anchor than the negative sample.

$$d(A,P) + margin < d(A,N) \tag{2}$$

- semi-hard triplets: triplets where the negative sample is not closer to the anchor sample than the positive ones, but which still have positive loss:

$$d(A,P) < d(A,N) < d(A,P) + margin \tag{3}$$

- hard triplets: triplets where the negative is closer to the anchor than the positive, i.e.

$$d(A,N) < d(A,P) \tag{4}$$

In the paper by Zürn et al. [1] a random approach for triplet formation is taken, wherein we pick any two samples from one terrain's set of audio bits and one sample from some other terrain. So, basically easy triplets category is chosen with an assumption that two consecutive audio bits have high probability of belonging to same terrain when evaluating on real-world dataset. Now the problem with this approach is that it fails to separate two very similar sounding terrains from each other, so we need to keep feeding lots of data specifically regarding those two or either one of those terrains to the neural network. This is primarily the reason we observed during manual data scraping in Zürn et al.'s [1] dataset that the Gravel and Grass had high duration in all of the 14 trajectories as compared to their similar counterparts Asphalt and Parking Lot respectively, which have similar auditory features. This is a great method to subtly shift the weights so that the neural net can learn from abundance of the one feature and can focus on the specific characteristics of the counterpart.

But in case of difficult terrains we have also come up with a solution of using hard triplets built from the random sampling strategy as explained above. The use of each of the three types of triplets, mentioned earlier, depends upon situations where the negative is compared to the anchor and positive. If our loss function is 0 for easy triplets and high loss for hard triplets, then either our model will not learn much. This will give any mislabelled data too much weight. Therefore, we choose a strategy that mixes easy, hard and maybe semi-hard triplets to a certain extent. Based upon the idea from [9] paper, respecting the class distribution, we chose to sample a relatively large number of random sample of triplets (using the standard random sampling strategy) and then we pick 25

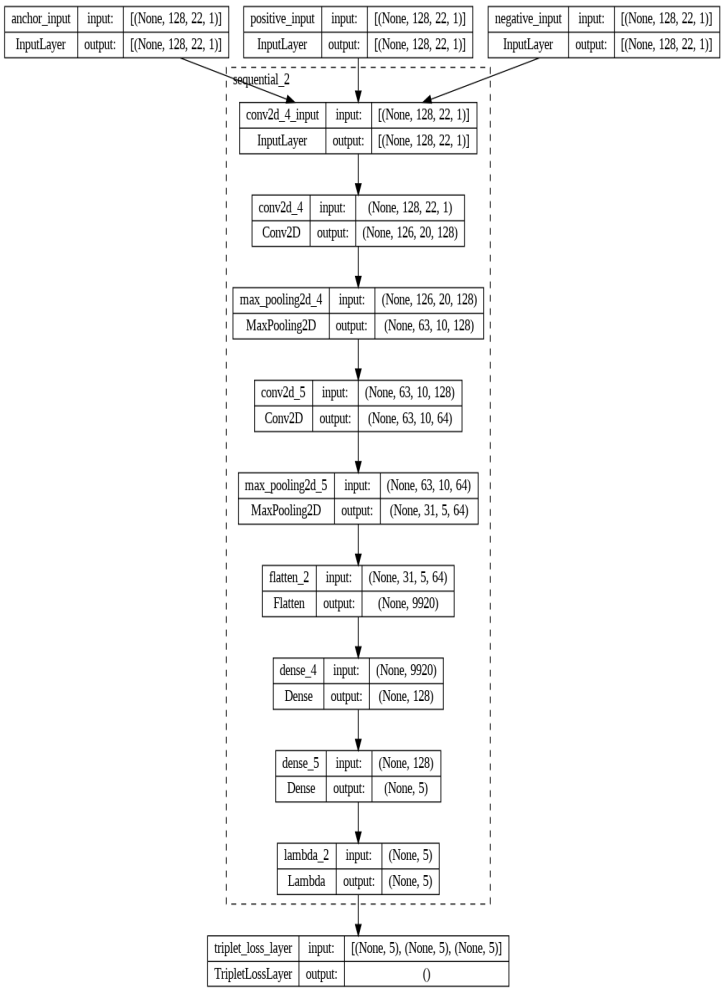

Fig. 1: Proposed Siamese Neural Network with Triplet Loss.

to 40 percent of those large number sampled random triplets. Subsequently, we sort that big pool of random triplets we sampled earlier based on the triplets which have anchor-negative closer distance less than anchor-positive distance. Now out of this sorted pool we choose 60 to 75 percent of this sorted random triplets as hard triplets. This helps us to indirectly control the training process by regulating the weightage of classes in triplets and consequently increasing the quality of triplets without infusing lot of data to shift internal weights. Using this hard sampling method also provides comparable or in some cases better results than those by random sampling method used in Zürn et al.'s [1] formulation.

Once these triplets are formed we pass them through the Siamese neural network which minimizes the loss functions to learn the feature similarity between the inputs. The network which we are using here has triplet loss, hence instead of calculating triplet loss we added the triplet loss function as the last layer of the neural network architecture to directly give us with train and validation loss during the training phase. The neural network which we have finalized as of now is shown in "Fig. 1". This overall architecture ("Fig. 2") is subject to change and fine-tuning is to get more accurate results.

D. Blind Source Separation

In order to develop a self-supervised audio-visual framework better than [1], we must focus on perfecting the unsupervised audio segregation [12]. Since, in the case of martian audio we lack the availability of tagged images corresponding to audio clip, we cannot possibly leverage the vision-based feature entities at this point of time. Hence, we try to perfect the audio classification using unsupervised source separation so that time-synchronized audio labels of the image would be sufficient to ascertain the relation of the corresponding tagged audio bit with respect to the terrain audio clip, leading to correct terrain labelling in the image.

The martian audio categorization problem here is a Blind Source separation problem, where from mixed audio clip we need to isolate the sources [11]. Independent component analysis (ICA) [11] and Non-negative Matrix Factorization (NMF) [12] are two methods which can dela with this problem.

In ICA, the goal is to find a set of independent components (ICs) y(t) that are statistically independent and correspond to the original audio sources s(t) [11]. The ICs can be obtained by applying a linear transformation W to the mixed signal x(t):

$$y(t) = Wx(t) \tag{5}$$

where W is an M x N matrix of weights that maximizes the statistical independence between the ICs.

To estimate the weight matrix W, ICA uses a cost function that measures the non-Gaussianity of the ICs. One commonly used cost function is the negentropy, which is defined as:

$$J(y) = H(y_{gauss}) - H(y) \tag{6}$$

where H(y) is the differential entropy of the IC y and H(ygauss) is the entropy of a Gaussian random variable with the same covariance matrix as y. The goal of ICA is to find the weight matrix W that maximizes the negentropy J(y) i.e. we want to find the W that makes the ICs as non-Gaussian as possible. Maximizing J(y) is equivalent to minimizing the joint entropy of the ICs. Hence, more independent the ICs are, lower their joint entropy will be.To optimize J(y) with respect to W, we use FastICA as the iterative algorithm that updates the weight matrix W at each iteration using a fixed-point iteration method.

Whereas, NMF is a technique for decomposing a non-negative matrix (or audio signal) X into two non-negative matrices V and H:

$$X = V * H \tag{7}$$

where X is M x N non-negative matrix, V is an M x K non-negative matrix (which corresponds to the basis functions), H is a K x N non-negative matrix(where H corresponds to the activation coefficients that define the contribution of each basis function to each source signal), and K is a parameter that determines the number of basis functions. The goal of NMF is to find the non-negative matrices V and H that minimize the reconstruction error between X and VH, subject to the non-negativity constraints. This can be written as the following optimization problem:

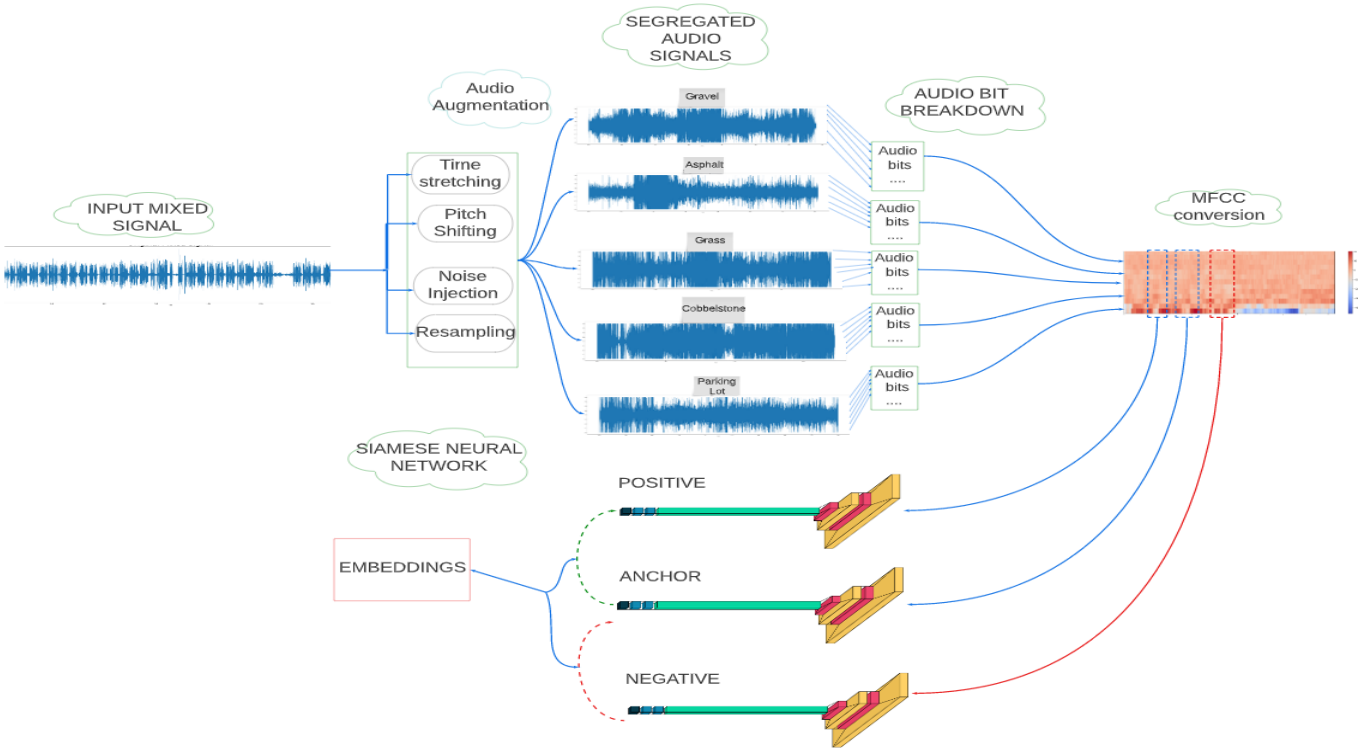

Fig. 2: Proposed Baseline formulation

$$min\|X - VH\|^2 \tag{8}$$

where $\|.\|^2$ denotes the squared Euclidean norm; subject to V and H greater than or equal to zero.

We leverage the Multiplicative Update (MU) ((6) and (7)) and gradient descent algorithm for problem optimization.

$$V_{ij} \leftarrow V_{ij} \frac{(XH^T)_{ij}}{(VHH^T)_{ij}} \tag{9}$$

$$H_{ij} \leftarrow H_{ij} \frac{(V^T X)_{ij}}{(V^T VH)_{ij}} \tag{10}$$

Using the above two update rules we iteratively update the basis matrix V and the activation matrix H until the reconstruction error is minimized. This means we will try to find the V and H that minimizes the distance between the given data matrix X and the estimated one. Once the matrices V and H are estimated, the original matrix X can be approximated by multiplying V and H, as done in (5).

The main problem with using ICA is its assumption that the observed signal is a linear combination of independent sources that correspond to the original sources. Second, the ICA has difficulty separating overlapping frequency spectra. In our experiment when we try ICA source separation on martian audio data its wasn't able to distinguish between big rock and bedrock effectively; also in case of Zürn et al.'s [1] dataset ICA wasn't able to distinguish between gravel, asphalt and grass leading to improper MFCC coefficients and misclassification on the testing dataset. In such cases NMF is inherently beneficial because it can model each source as a linear combination of basis vectors, allowing for more flexible modeling of complex spectra. Also, after training we feels that NMF is better able to handle non-stationary sources because it can adapt its basis vectors over time to better capture the changing statistics of the sources.

Therefore when evaluating the audio classification framework with blind source separation techniques the architecture as illustrated in "Fig. 2" remains the same for the most part but the starting part where the original mixed audio gets separated into distinct terrain audio clips by manual labelling and audio data augmentation is replaced by a NMF module which takes in original audio to output equal length of audio clip for each terrain, eliminating the need for data augmentation and audio duration equalization.

E. Direct comparison and Clustering

To formulate an unsupervised audio classification module, we also consider clustering techniques. Here we consider just the 16 min. martian terrain audio as given data. Before clustering we try to experiment with our own unsupervised segregation method. Here, we divide the martian audio in small bits of 2.95 seconds we sequentially iterate and create numerous triplets among these tiny bits. Then we compute a weighted sum similarity score in time-frequency domain within every set of triplets formed to determine the anchor, positive and negative bit within every triplet. Subsequently, we collect all the anchor, positive and negative audio bits from

the entire dataset by augmenting all the triplets. This results in data segregation, but only when there are 2 types of terrains. However, for more number of terrains this classifier fails and employs partially greedy approach for terrain classification. Hence, this motivates to use clustering methods.

Upon analysing numerous resources and clustering techniques we found out that K-means clustering and spectral clustering are better suited for the job of audio data classification, primarily due to applicability of elbow method; their general consideration of radial & non-radial cluster distribution and good results on toy datasets. But, on testing on martian and Zürn et al.'s [1] dataset we rule out spectral clustering since it was unable distinguish between similar sounding terrains like bedrock and big rock. Instead, k-means clustering was able to identify and segregate the corresponding terrains accurately.

IV. Experiments & Results

A. Results from Baseline

The baseline is created to serve as a standard for comparison of other unsupervised approaches. Here, we have experimented on the Zürn et al.'s [1] dataset since this dataset had time-synchronized audio-visual data, which was scraped to create baseline dataset. Out of the 14 trajectories in Zürn et al.'s [1] dataset we have generated results for trajectory nos. 1,2 and 12 with random and hard triplet sampling, but for demonstration purposes, here in this section, we show the results of 12th trajectory with random triplet sampling only.

Trajectory 12: The results for Trajectory 12 (testing data) (having 5 terrain/classes as mentioned in Section III (A)) are given by Fig. 3 & 4 and TABLE I & II.(Here we have represented the terrains as corresponding classes 0,1,2,3,4 etc. for the ease of understanding)

Distance	Class 0	Class 1	Class 2	Class 3	Class 4
from class 0	5.11e-14	0.086	0.007	0.043	0.005
from class 1	0.086	4.17e-14	0.118	0.121	0.085
from class 2	0.007	0.118	4.73e-14	0.024	0.008
from class 3	0.043	0.121	0.024	1.168e-13	0.026
from class 4	0.005	0.085	0.008	0.026	3.17e-14

TABLE I: Class inter-distance for Trajectory 12 before random batch-based training

Distance	Class 0	Class 1	Class 2	Class 3	Class 4
from class 0	2.15e-12	1.296	2.491	0.454	2.189
from class 1	1.306	4.850e-12	0.648	1.716	0.231
from class 2	2.491	0.648	1.25e-12	2.603	0.527
from class 3	0.454	1.716	2.603	4.783e-12	2.341
from class 4	2.189	0.231	0.527	2.341	3.47e-12

TABLE II: Class inter-distance for Trajectory 12 after random batch-based training

The validation metrics used here are: Receiver Operating Characteristic (ROC) curve and Inter-class Distance. These metrics are described briefly as follows:

ROC CURVE: Standard classifier's performance is assessed by the class with best prediction score, resulting in a confusion matrix with F1 metrics/recall. Since the model generates embeddings for distance calculation we cannot use that method here. The distance should be "low"/"high" for audio bits of same/different classes respectively. Hence, a threshold is needed to determine if the estimated distance is similar or not. Too low of this threshold implies high precision & many false negatives; whereas, too high will generate too many false positives. This is a ROC curve problem, where metrics are thresholded Area Under Curve (AUC) and False Positive Rate (FPR) (soft hyperparameters). We choose FPR to be under 1e-2 and based upon this we will approximate sensitivity of our algorithm. Hence, the validation process will take the test audio bits and evaluate their distance against one another to compute AUC.

INTER-CLASS DISTANCE: This metric is standardized to look at during training, which tells us how far apart are the the embeddings from each of the terrain (class) are from each other. But here this is just to check that the network is converging smoothly for all the classes. Even this is validated on the test dataset.

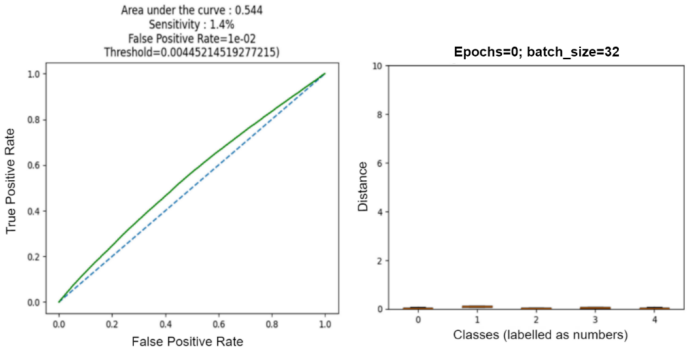

Fig. 3: ROC curve (left) and Class inter-distance plot (right) generated from untrained network (BEFORE)

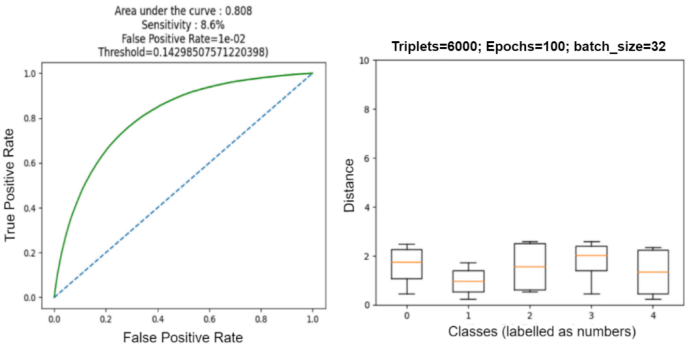

Fig. 4: ROC curve (left) and Class inter-distance plot (right) generated from trained network (AFTER)

B. Results from Blind source separation method

The blind source separation method used here is first implemented on the Martian audio data and then again on trajectory nos. 1,2 and 12 of Zürn et al.'s [1] dataset, out of

which we will show the results of trajectory no. 12 only (here with random triplet sampling and in appendix with hard triplets sampling)

Martian Audio: The results for martian audio (testing data) (having 4 terrain/classes as mentioned in Section III (A)) are given by Fig. 5 and TABLE III & IV. Here, we directly show the change of embedding inter-distance among various classes after training. The sensitivity obtained here was 25% with an area under curve = 0.975 sq. units.

Distance	Class 0	Class 1	Class 2	Class 3
from class 0	4.98e-13	0.001	0.018	0.001
from class 1	0.001	2.21e-12	0.013	0.005
from class 2	0.018	0.013	1.36e-13	0.025
from class 3	0.001	0.005	0.024	3.64e-14

TABLE III: Class Inter-distance for Martian Audio before random batch-based training using NMF

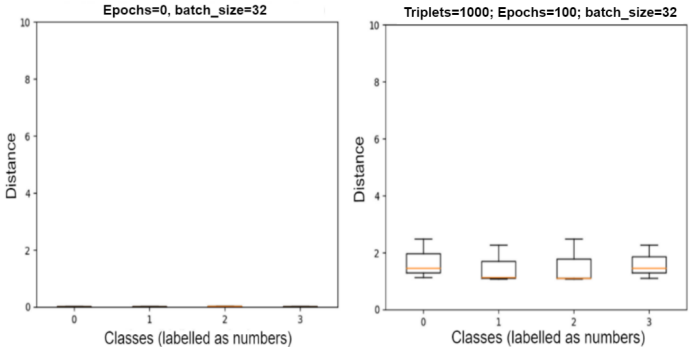

Fig. 5: Class inter-distance (Left; before training) and Class inter-distance plot (Right; after training) for Martian Audio

Distance	Class 0	Class 1	Class 2	Class 3
from class 0	3.05e-13	1.141	2.473	1.461
from class 1	1.142	3.197e-14	1.075	2.276
from class 2	2.473	1.075	1.190e-13	1.107
from class 3	1.461	2.276	1.107	9.11e-13

TABLE IV: Class Inter-distance for Martian Audio after random batch-based training using NMF

Trajectory 12: The results for Trajectory 12 (testing data) based on Blind source separation (NMF method) (having 5 terrain/classes as mentioned in Section III (A)) are given by Fig. 8 and TABLE VII & VI.

Distance	Class 0	Class 1	Class 2	Class 3	Class 4
from class 0	6.28e-7	1.978e-1	0.175	2.103e-2	0.029
from class 1	2.81e-2	3.51e-13	0.255	0.941	3.503e-1
from class 2	0.232	0.352	3.18e-14	0.357	0.696
from class 3	1.026e-1	0.141	1.896e-2	2.63e-13	0.850
from class 4	0.293	3.53e-1	3.711	4.85e-1	2.888e-14

TABLE V: Class Inter-distance for Trajectory 12 before random batch-based training using NMF

C. Results from Clustering technique

We applied k-means clustering to the martian audio to classify the audio based on 4 types of terrains. Prior to the application of clustering technique, we first divided the continuous audio signal into small discrete audio bits/chunks of 2.95 seconds each. Thereafter, we applied elbow method to determine possible number of clusters. We use two versions of elbow cluster determination: First, distortion-based elbow, determined by averaging the euclidean distance squares between each cluster's cluster centers. Second, inertia-based elbow, determined by the sum of all of samples' squared distances from the nearest cluster center. The plots pertaining to both are shown in Fig. 6.

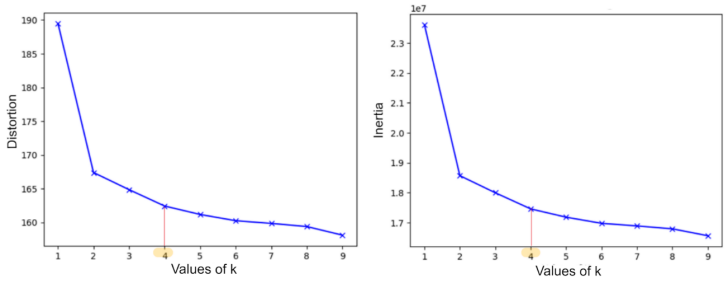

Fig. 6: Elbow methods

Now in Fig. 6 we observe that after k=4 the graph starts to linearize and become uniform in nature, hence providing us with the proof of presence of 4 classes/terrains. The overall clustering results can be see in Fig. 7, where we observe that the martian audio data is segregated into 4 terrains.

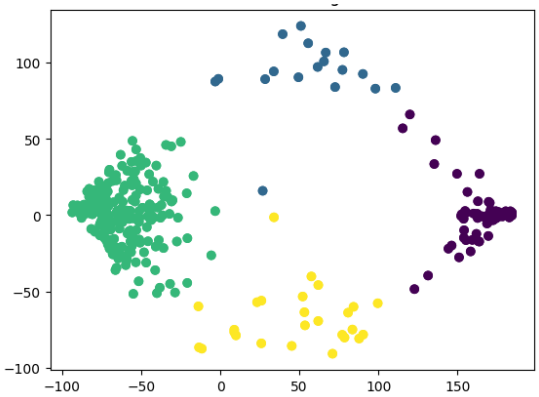

Fig. 7: k-means clustering output on martian audio data

V. Discussion

The results of Trajectory 12 derived from baseline implementation depicts that the proposed audio preprocessing pipeline and the siamese neural network have improved the overall classification performance.

This can be seen by the ROC curve and inter-distance between the embeddings of various classes from each other before and after training the neural network. The area under the ROC curve is directly proportion to the model robustness

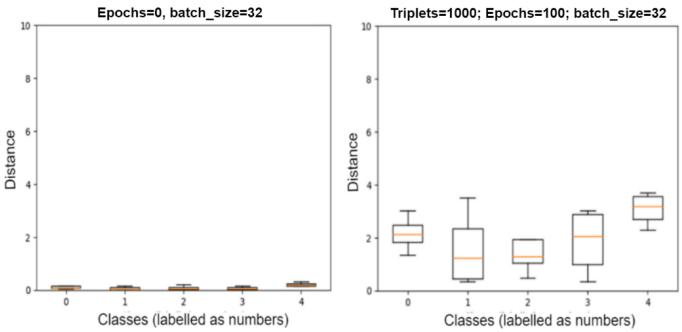

Fig. 8: Class inter-distance (Left; before training) and Class inter-distance plot (Right; after training) for Trajectory 12

to false positives. We can also see that due to slight increase in AUC the model's sensitivity increased by 7%.

In order to quantitatively justify our hypothesis we can also see that the class inter-distance metrics have increased for every class.

Distance	Class 0	Class 1	Class 2	Class 3	Class 4
from class 0	3.28e-14	1.978	1.342	3.027	2.293
from class 1	1.978	3.51e-13	0.475	0.341	3.503
from class 2	2.342	0.475	3.18e-14	1.227	3.780
from class 3	3.027	0.340	1.267	1.62e-13	2.980
from class 4	2.293	3.503	3.696	2.850	2.88e-14

TABLE VI: Class Inter-distance for Trajectory 12 after random batch-based training using NMF

The inter-distance between the embeddings of each class initially being close to 0 suggests that model hardly can distinguish between various terrain sounds. After reducing dimensionality with proposed audio processing pipeline and neural net training the class inter-distances increased greatly, suggesting our model's ability to distinguish various terrains.

The results using our Blind source separation technique are almost comparable or sometimes even better than those of baseline. The proof of which can be seen by increase of inter-distances between the class embeddings for both Martian and the Trajectory 12 audio of Zürn et al.'s [1] dataset. We also validated the model trained on Trajectory 12 with random audio clips from other trajectories of the Zürn et al.'s [1] dataset and found that the model was able to predict terrains correctly almost every time. We also assessed our model's performance on hard sampling strategy that we proposed in this study (results in the appendix), and found out that hard triplets gave comparable results when compared to random triplets.

Zürn et al. [1] used a self-supervised audio-visual framework for terrain classification, whereas we tried to translate their self-supervised approach into unsupervised audio terrain segregation. This is done due to the unavailability of suitable image dataset to aid self-supervised learning. We tend to obtain better results than Zürn et al. [1] method, which is clearly signified by the class inter-distance plot. The curve and neural net hyperparameters can be fine-tuned based on image data availability for better results.

VI. Conclusion and Future Work

In this study, we developed a robust unsupervised audio-based terrain segregation pipeline, which is a core component of the self-supervised terrain classification system. We first described the dataset and established a generalized audio preprocessing & feature extraction pipeline. Subsequently, the baseline was explained in detail which illustrated the effectiveness of the neural network architecture that we used throughout this study and also served as our ground truth to validate the Blind source audio separation results. We also discussed about the random and hard audio triplet sampling methodology. At last we experimented with clustering techniques and shortlisted k-means clustering. Hence, based on overall study we conclude that the use of Blind source separation methods on minimal training provides results comparable to standard clustering techniques, but if trained more can outperform them.

In future, we intend to validate our approach on the martian terrain image dataset to complete the predetermined self-supervised learning architecture.

References

[1] Zürn, Jannik, Wolfram Burgard, and Abhinav Valada. "Self-supervised visual terrain classification from unsupervised acoustic feature learning." IEEE Transactions on Robotics 37.2 (2020): 466-481.

[2] Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, and Yann LeCun. Learning long-range vision for autonomous off-road driving. Journal of Field Robotics, 26(2):120–144, 2009.

[3] Kurt Konolige, Motilal Agrawal, Robert C Bolles, and Aravind Sundaresan. Mapping, navigation, and learning for off-road traversal. Journal of Field Robotics, 26(1):88–113, 2009.

[4] Brooks, Christopher A., and Karl Iagnemma. "Self-supervised terrain classification for planetary surface exploration rovers." Journal of Field Robotics 29.3 (2012): 445-468.

[5] Mohammed Abdessamad Bekhti, Yuichi Kobayashi, and Kazuki Matsumura. Terrain traversability analysis using multi-sensor data correlation by a mobile robot. In 2014 IEEE/SICE International Symposium on System Integration, pages 615–620. IEEE, 2014.

[6] Noriaki Hirose, Amir Sadeghian, and Silvio Savarese. Gonet: A semi-supervised approach for traversability estimation. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3044–3051. IEEE, 2018.

[7] Dan Barnes, Will Maddern, and Ingmar Posner. Find your own way: Weakly-supervised segmentation of path proposals. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 203–210. IEEE, 2017.

[8] Mirko Nava, Jerome Guzzi, R Omar Chavez-Garcia, Luca M Gambardella, and Alessandro Giusti. Learning long-range perception using self-supervision from short-range sensors and odometry. IEEE Robotics and Automation Letters, 4(2):1279–1286, 2019.

[9] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[10] Swan, R. Michael, et al. "Ai4mars: A dataset for terrain-aware autonomous driving on mars." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[11] Naik, Ganesh R., ed. Independent component analysis for audio and biosignal applications. BoD–Books on Demand, 2012.

[12] Wang, Yu-Xiong, and Yu-Jin Zhang. "Nonnegative matrix factorization: A comprehensive review." IEEE Transactions on knowledge and data engineering 25.6 (2012): 1336-1353.

[13] Wei Zheng, Le Yang, Michael Buck, SENSE: Siamese neural network for sequence embedding and alignment-free comparison, Bioinformatics, Volume 35, Issue 11, June 2019, Pages 1820–1828.

VII. Appendix

Here we will show the results of trajectory 12 (5 classes) of Zürn et al. [1] when we sample and train the neural network by employing hard triplet batching using our Blind source separation technique.

A. Results on Blind Source Separation using hard triplets

Distance	Class 0	Class 1	Class 2	Class 3
from class 0	1.85e-5	0.175	3.175	2.103
from class 1	0.175	7.71e-13	1.255	2.941
from class 2	2.283	1.965	2.18e-14	0.357
from class 3	1.950	2.273	1.896	2.63e-13

TABLE VII: Class Inter-distance for Trajectory 12 after training using hard triplet batching

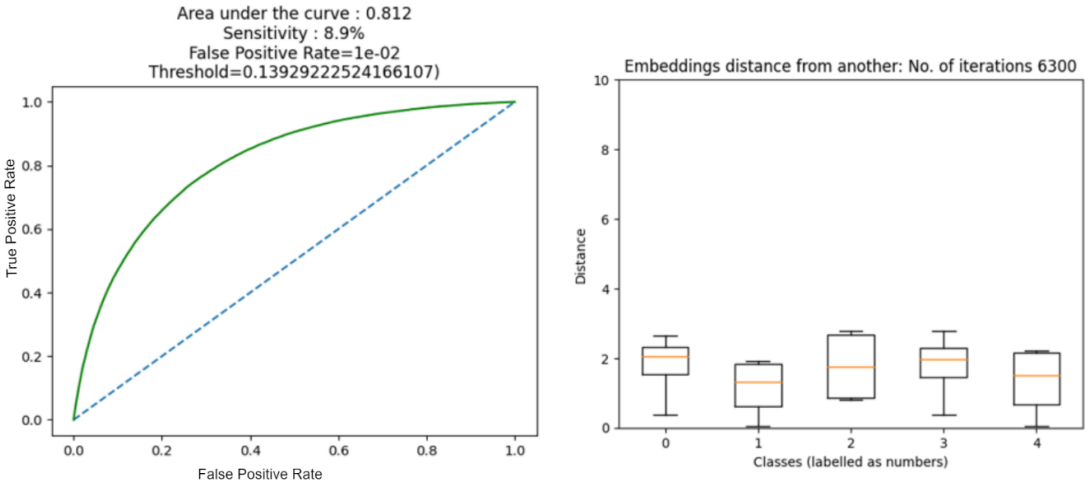

Fig. 9: ROC curve (left) and Class inter-distance (right) for Trajectory 12 after training using hard triplet batching

We can see quantitatively by comparing the inter-distance characteristics of Trajectory 12 when trained with random triplet sampling on Blind source separation data and when trained with hard triplet batching on Blind source separation data, that we usually get comparable and are after some epochs far more superior results when we employ hard triplet batching.

This provides a befitting testimony that our ideation of hard triplet batching provides both qualitative and quantitative results when experimenting on large terrain datasets.