Unsupervised and Semi-supervised Novelty Detection using Variational Autoencoders in Opportunistic Science Missions

Lorenzo Sintini
lorenzo.sintini@gmail.com
Lars Kunze
lars@robots.ox.ac.uk

Oxford Robotics Institute
Department of Engineering Science
University of Oxford
Oxford, UK

Abstract

Scientific opportunities are missed in planetary explorations due to the lack of communication and/or long-time communication delays between rovers and ground stations. By enabling rovers to autonomously detect and explore targets the overall scientific outcome of extraterrestrial missions can be increased.

In this paper, we have designed, developed, and evaluated unsupervised as well as semi-supervised approaches to novelty detection based on Variational Autoencoders (VAE). Our VAE model was trained on typical data from previous missions and tested to infer the novelty of scientific targets. In an ablation study, we investigate the effectiveness of different types of loss functions. We compare losses based on reconstruction errors, losses obtained from the VAE's latent space as well as a combination of both. In our experiments, we have evaluated both unsupervised and semi-supervised approaches on datasets obtained from NASA's Mars Curiosity rover. Results show that our VAE-based approaches are not only robust but also comparable, or better, than the state-of-the-art.

1 Introduction

Extraterrestrial surface missions are essential for scientific discoveries on other planets. In general, these missions are planned well in advance: scientists identify places of interest and engineers ensure that a robot can safely reach and investigate them. However, while executing a mission plan, a robot might have an opportunity to make unplanned scientific discoveries without compromising any of its mission goals [7]. By deviating from its original mission plan and autonomously exploring the surroundings, a robot could identify and probe novel, scientifically interesting targets. Hence, autonomous robots that perceive their environment, interpret what they have seen, and act upon their analysis have a great potential to increase the overall scientific return of expensive surface missions. —

In this paper, we focus on the problem of novelty detection in camera data for the identification of potential targets in opportunistic science missions. We define novelty detection as the task to identify unknown test data which is to some degree different from typical training data. Our approach is based on Variational Autoencoders (VAE) [4] which are not able to reconstruct image data of novel scenes without errors. By interpreting these errors, VAE-based models can provide important information about the novelty of scenes. In this work, we consider this information in unsupervised as well as semi-supervised approaches which are aimed for scenarios in which annotated data is either not available or very limited. In experiments, we evaluate the effectiveness of these approaches and compare them to state-of-the-art methods. To this end, this paper makes the following contributions:

- a set of unsupervised and semi-supervised novelty detection methods based on Variational Autoencoders (VAE);
- an ablation study of VAE-based loss functions (based on reconstruction losses, latent space losses, and their combination) for the task of novelty detection; and
- a state-of-the-art comparison of novelty detection methods based on a multi-spectral camera dataset acquired by NASA's Curiosity rover on Mars.

The remainder of the paper is structured as follows. First, we briefly discuss related work in Section 2. Second, we describe our approach to novelty detection in Section 3. Finally, in Section 4, we present the Mastcam dataset of NASA's Curiosity rover, our ablation study, as well as a comparison with state-of-the-art methods, before we conclude in Section 5.

2 Related Work

Novelty detection is an important task in many application domains including IT security, medical diagnostics, industrial monitoring, video surveillance, and opportunistic science. A review of different approaches to novelty detection is given by Pimentel et al. [9].

Some of the traditional methods use One-Class Support Vector Machine (OCSVM) [12] and Support Vector Data Description (SVDD) [10]. Both methods are frequently used, but do not scale up well with high dimensional data, such as images.

More recently, complex solutions involving deep neural networks have been used to detect and analyse scenes. An approach to anomaly detection based on Autoencoders (AE) was developed by Zong et al. [17]. Convolutional AE (CAE) have also been used successfully for novelty detection in the context of extraterrestrial surface mission as shown by Kerner et al. [3]. In this work, we present a similar approach based on a Variational Autoencoder (VAE) [4]. The VAE provides us with additional probabilistic information of the data and the latent space. This additional knowledge can be used to formulate more loss functions, which can be stochastic rather than deterministic. Other papers ([1] and [16]) have also studied the use of VAEs in the field of anomaly detection. These works however have only used them in an unsupervised model on standard dataset such as MNIST and KDD99. Our work, instead, develops both unsupervised and semi-supervised models and tests them on significantly more challenging real-world multi-spectral image data (6-channels) from Mars. Moreover, this papers provides an extensive ablation study on many different losses to determine their performances, while those papers used a single loss.

Several works proposed novelty detectors based on Generative Adversarial Networks (GAN) [5, 8, 11]. Work by [5] showed that a multi-class discriminator trained with a generator (based on a mixture of nominal and novel data distributions) is optimal.

Vasilev et al. [15] employed VAEs for novelty detection and defined a range of detectors. In their study, they evaluated these detectors based on different information metrics. Similarly, we consider novelty detectors in both latent space and the original feature space.

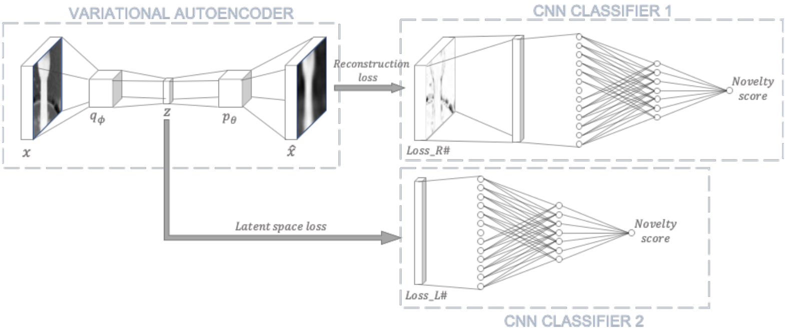

Figure 1: **Novelty Detection Approach.** First, a Variational Autoencoder, which was trained on typical images, encodes and reconstructs a given test image. Second, reconstruction losses as well as latent space losses are computed. Based on normalised losses we calculate a novelty score in our unsupervised method. In our semi-supervised method, we have trained convolutional neural networks (CNN) to classify images (losses) based on a small number of training examples. Thereby, we can significantly improve the overall detection accuracy.

3 Novelty Detection

Our approach to novelty detection in images is based on a convolutional VAE (Figure 1). The aim of a VAE is to reconstruct a given image through an encoder and a decoder network. In this work, we consider a variety of losses that result from the encoder (in latent space) and the decoder (reconstruction losses in image space). We present an unsupervised, threshold-based approach to classify novel images based on one-dimensional losses, and a semi-supervised, CNN-based approach that classifies images based on high-dimensional losses.

3.1 Variational Autoencoder-based Model

A VAE takes an image x as input, encodes it into a latent vector z and tries to output a reconstructed image $\hat{x}$ from it. This is shown in the top left part of Figure 1.

The VAE encoder (q_ϕ) architecture consists of 4 convolutional layers followed by 2 fully connected layers. The kernel sizes of the convolutional layers are $(32 \times 7 \times 7)$, $(128 \times 5 \times 5)$, $(64 \times 3 \times 3)$, $(6 \times 3 \times 3)$, all with single strides and 'same padding'. These are all followed by a batch-normalization layer, a dropout layer with dropout rate of 0.25 and a MaxPool layer with stride 2 (only for the middle two layers). The two fully connected layers both have a size twice as big as z, in order to compute both the mean and the variance for each dimension. The latent space was selected to be a vector of length 768, compared to the original image dimensionality of $64 \times 64 \times 6 = 24,576$. The decoder ($p_\theta$) was built symmetrically and has the same architecture.

The VAE was trained using the ELBO loss, shown in Equation 1, where KL corresponds to the Kullback–Leibler divergence between two probability distributions.

$$L_{VAE} = -ELBO = -E_{z \sim q_\phi}[\log P(x|z)] + KL(q_\phi(z|x)||P(z)) \tag{1}$$

The ELBO loss is made up of two terms, each of which guarantees that two requirements are met during training. The first one is that the image should be reconstructed accurately, the second one is that the latent space values should have a distribution similar to a normal distribution $\mathcal{N}(0,1)$. This leads to the idea that different loss functions can be formulated.

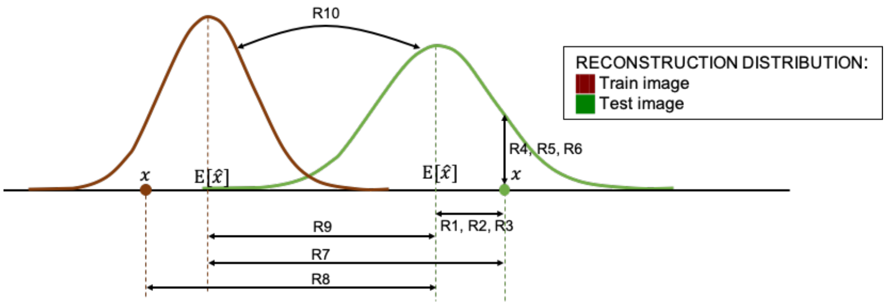

Figure 2: Visual explanation of reconstruction losses.

$Loss_{R1}$	Pixel-wise square error	$\sum(x_{test} - E[p_\theta(x\|E[q_\phi(z\|x_{test})])])^2$
$Loss_{R2}$	Encoder stochastic pixel-wise square error (average)	$\operatorname*{mean}_{z_i \sim q_\phi} \sum(x_{test} - E[p_\theta(x\|z_i)])^2$
$Loss_{R3}$	Encoder stochastic pixel-wise square error (minimum)	$\min_{z_i \sim q_\phi} \sum(x_{test} - E[p_\theta(x\|z_i)])^2$
$Loss_{R4}$	Reconstruction probability	$-\sum \log p_\theta(x_{test}\|E[q_\phi(z\|x_{test})])$
$Loss_{R5}$	Encoder stochastic reconstruction probability (average)	$\operatorname*{mean}_{z_i \sim q_\phi} -\sum \log p_\theta(x_{test}\|z_i)$
$Loss_{R6}$	Encoder stochastic reconstruction probability (minimum)	$\min_{z_i \sim q_\phi} -\sum \log p_\theta(x_{test}\|z_i)$
$Loss_{R7}$	Closest training image pixel-wise square loss (train reconstruction)	$\min_{y \in Y} \sum(x_{test} - E[p_\theta(x\|E[q_\phi(z\|y)])])^2$
$Loss_{R8}$	Closest training image pixel-wise square loss (test reconstruction)	$\min_{y \in Y} \sum(E[p_\theta(x\|E[q_\phi(z\|x_{test})])] - y)^2$
$Loss_{R9}$	Closest training image pixel-wise square loss (both reconstruction)	$\min_{y \in Y} \sum(E[p_\theta(x\|E[q_\phi(z\|x_{test})])] - E[p_\theta(x\|E[q_\phi(z\|y)])])^2$
$Loss_{R10}$	Bhattacharyya distance between reconstruction probabilities	$\min_{y \in Y} \sum D_B(p_\theta(x\|E[q_\phi(z\|x_{test})]), p_\theta(x\|E[q_\phi(z\|y)]))$

Table 1: Reconstruction losses (R1–R10).

3.2 Loss Functions

While the model was trained using the ELBO loss, at test time we use a variety of different losses (similar as in [15]): reconstruction losses, latent space losses, and mixed losses.

Reconstruction Losses (R). Table 1 lists several one-dimensional reconstruction losses and Figure 2 gives a visual explanation of how they are computed.

$Loss_{R1}$ is the total pixel-wise difference between original and reconstructed image. $Loss_{R4}$ is the sum of the probability density function of the decoded distribution for each pixel, evaluated at the test image pixels values. This relates to the probability that the original image comes from the probability distribution computed. In the above losses the reconstruction distribution is computed from the expected value of the latent vector. We can, however, sample latent vectors from its probability distribution and obtain different losses each time. $Loss_{R2}$, $Loss_{R3}$, $Loss_{R5}$ and $Loss_{R6}$ are the mean or minimum of these losses obtained.

The last four losses are based on the density of the data distribution. $Loss_{R7}$ is similar to $Loss_{R1}$. The test image however is not subtracted to its own reconstruction but rather to the reconstruction of whichever training image is closest to it. Y represents the training dataset

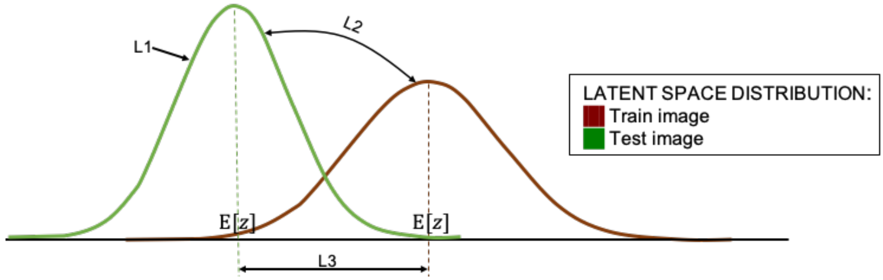

Figure 3: Visual explanation of latent space losses.

$Loss_{L1}$	Kullback–Leibler divergence	$\sum KL(q_\phi(z\|x_{test})\|\|\mathcal{N}(0,1))$
$Loss_{L2}$	Bhattacharyya distance between distributions	$\min_{y\in Y}\sum D_B(q_\phi(z\|x_{test}),q_\phi(z\|y))$
$Loss_{L3}$	Euclidean distance between means of the distributions	$\min_{y\in Y}\sum(E[q_\phi(z\|x_{test})]-E[q_\phi(z\|y)])^2$
$Loss_{L4}$	Density latent space	$\frac{1}{\|Y\|}\sum_{y\in Y} q_\phi(E[q_\phi(z\|x_{test})]\|y)$

Table 2: Latent space losses (L1–L4).

and y are the images in it. This ensures that if a novel image is composed of very simple patterns which the VAE might have learnt to reconstruct well, it will still have a high loss value because it is very different from all other images in the training set. Alternatively the reconstruction of the test image could be compared to the reconstruction of the train images ($Loss_{R9}$) or to the original train image itself ($Loss_{R8}$). $Loss_{R10}$ computes the Bhattacharyya distance D_B between the decoded probability distribution for the test image and a train image. The loss of the closest distributions is picked.

Latent Space Losses (L). Table 2 and Figure 3 list and depict various latent space losses. $Loss_{L1}$ is the sum of the Kullback–Leibler divergences between the encoded probability distribution of each pixel and a normal distribution. $Loss_{L2}$ and $Loss_{L3}$ measure the difference between the latent space distribution of the test image, and the closest training image. The first does so using the Bhattacharyya distance while the second calculates the Euclidean distance between the means. $Loss_{L4}$ represents a latent space density based loss. It is a measure of how close the latent vector of a test image is to the the average latent space of the images in the training set.

Mixed Losses (M). Lastly, we can use combinations of reconstruction and latent space losses. For example the ELBO loss from Equation 1. Table 3 shows three variations of the ELBO loss. $Loss_{M1}$ is based on the expected latent space, while losses $M2$ and $M3$ use various samples of it and compute the average and minimum of losses obtained from them.

$Loss_{M1}$	ELBO	$-\log p_\theta(x_{test}\|E[q_\phi(z\|x_{test})])+KL(q_\phi(z\|x_{test})\|\|\mathcal{N}(0,1))$
$Loss_{M2}$	Encoder stochastic ELBO (average)	$\operatorname{mean}_{z_i\sim q_\phi} -\log p_\theta(x_{test}\|z_i)+KL(q_\phi(z\|x_{test})\|\|\mathcal{N}(0,1))$
$Loss_{M3}$	Encoder stochastic ELBO (minimum)	$\min_{z_i\sim q_\phi} -\log p_\theta(x_{test}\|z_i)+KL(q_\phi(z\|x_{test})\|\|\mathcal{N}(0,1))$

Table 3: Mixed losses (M1–M3).

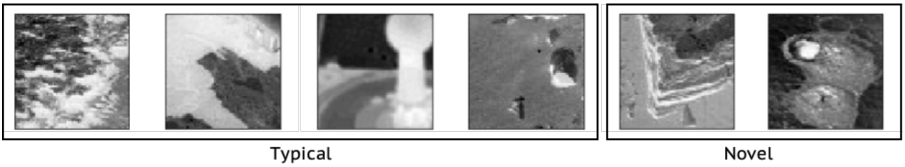

Figure 4: Typical and novel example images obtained by NASA's Curiosity rover.

3.3 CNN-based Classifiers

The functions in Table 1, 2, and 3 all yield one-dimensional losses since we sum over each pixel or latent vector dimension. Removing this summation we obtain a multi-dimensional loss of shape 64 × 64 × 6 for reconstruction losses and 768 × 1 for latent space losses. We refer to these as error-maps and error-vectors respectively.

We now introduce four CNN-based classifiers, each for different types of inputs. These have been evaluated in experiments described in Section 4.2 and 4.3.

Classifier 1 is built for error-maps (top right of Figure 1). The input is passed through two convolutional layers with kernel sizes (32 × 5 × 5) and (64 × 5 × 5), each followed by a MaxPool layer with double stride. After flattening, two dense layers follow: the first has an output of 512 (Relu activation function), the second has a single output (sigmoid activation).

Classifier 2 is built for error-vectors (bottom right of Figure 1). Having a vectorised input of length 768, the model is simply composed of three fully connected layers with 800, 300 and 1 neurons. The last layer has a sigmoid activation function, the others a Relu one.

Classifier 3 is built for a mixture of all four latent space error-vectors, which are merged into a matrix of shape 768 × 4. The classifier then consists of three convolutional layers and three fully connected layers. The convolutional layers have kernels of sizes (64 × 1 × 1), (16 × 1 × 1) and (1 × 1 × 1). The fully connected layers have outputs of 800, 300 and 1.

Classifier 4 is built for the mixed losses of Table 3, made up of an error-map and an error-vector. These two losses cannot be easily merged due to the different shapes. Hence the model runs the error-maps through the same convolutional layers of Classifier 1, after which it is flattened and concatenated to the error-vector. This long intermediate layer is then followed by a hidden layer with 500 neurons before reaching the output layer.

4 Experiments

4.1 Datasets

In this work, we have used the *Mars novelty detection Mastcam labeled dataset*[1], which consists of sub-sampled images (64 × 64 pixels) of the Mars Science Laboratory (MSL) Analyst's Notebook [14] which were obtained by NASA's Curiosity rover. The Mastcam is a multi-spectral camera and captures images at different wavelengths giving each image six channels. Two versions of the dataset are available, which we will refer to as DS1 and DS2.

DS1 is made up of a typical set with 98,800 images and a novel set with 332. Example images are shown in Figure 4. DS2 consists of 9,302 typical training images, 426 typical test images and 430 novel test images. In DS2, novel images are also split into eight classes: DRT spot, Dump pile, Broken rock, Drill hole, Meteorite, Vein, Flat rock, and Bedrock.

[1] DOI 10.5281/zenodo.1486195

Reconstruction losses				
LOSS	WACC	TP (FN)	TN (FP)	AUC
R1	71.1	25 (7)	64 (36)	76.0
R2	70.6	25 (7)	63 (37)	75.0
R3	70.0	24 (8)	65 (35)	75.1
R4	71.1	25 (7)	64 (36)	76.0
R5	70.6	25 (7)	63 (37)	75.1
R6	70.7	27 (5)	57 (43)	75.3
R7	69.9	23 (9)	68 (32)	75.5
R8	62.5	24 (8)	50 (50)	66.2
R9	**76.6**	25 (7)	75 (25)	**87.3**
R10	53.9	15 (17)	61 (39)	49.7

Latent space losses				
LOSS	WACC	TP (FN)	TN (FP)	AUC
L1	65.6	18 (14)	75 (25)	62.9
L2	64.9	22 (10)	61 (39)	61.7
L3	72.3	29 (3)	54 (46)	79.1
L4	61.3	29 (3)	32 (68)	61.8

Mixed Losses				
LOSS	WACC	TP (FN)	TN (FP)	AUC
M1	72.1	25 (7)	66 (34)	76.6
M2	72	24 (8)	69 (31)	75.6
M3	70.5	24 (8)	66 (34)	75.5

Table 4: Experimental results for unsupervised method on dataset DS1.

4.2 Ablation Study of Loss Functions

To study different types of loss functions, images of DS1 were classified using an unsupervised method (one-dimensional loss) and a semi-supervised method (high-dimensional loss). As in [3], 132 images were randomly selected for testing (100 typical, 32 novel).

Unsupervised method. To classify the images as novel or typical, we used different discrimination thresholds for each one-dimensional loss. In Figure 5, we show resulting ROC curves for selected losses. The loss threshold used to categorise novel from typical images was the one that maximised the weighted accuracy (WACC), which is preferred over a simple accuracy (ACC) since it is unbiased to class imbalances:

$$WACC = 0.5\left(\frac{TP}{TP+FP}+\frac{TN}{TN+FN}\right).$$

Complete results are shown in Table 4. The best performance in terms of WACC and AUC (area under the ROC curve) was achieved by loss R9. Reconstruction losses not based on the training data density (R1–R6) all achieved very similar results, which were slightly improved by the addition of latent space information in the mixed losses (M1–M3). This latent space information when used on its own, however, yielded worse results. Finally, the density-based reconstruction losses (R7–R10) generally produced worse accuracies except for $Loss_{R9}$, which achieved a weighted accuracy of 76.6% and an area under the curve of 87.3%, better than any other losses.

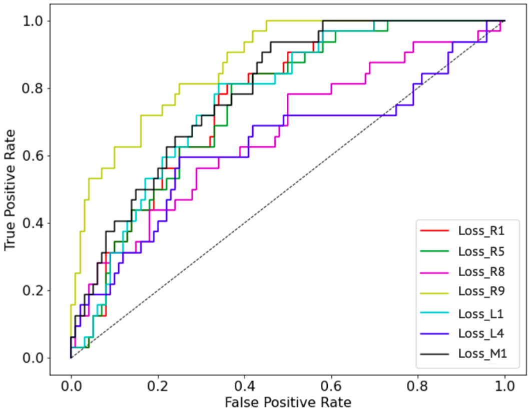

Figure 5: ROC curves for selected losses.

Semi-supervised method. For the semi-supervised method we used the high-dimensional version of each loss. This time the losses of each training image need to be computed, in order to train the CNNs. For the losses based on the typical training dataset, we can split the latter into two equal subsets, and use one to compute the losses of the other, which will then be used to train the classifier. A downside of these losses however is the much larger time

LOSS	WACC	TP	FP	TN	FN	AUC	WACC (unsupervised)	AUC (unsupervised)
Classifier 1								
R1	83.4	22	2	98	10	88.5	71.1	76.0
R2	81.9	23	8	92	9	86.9	70.6	75.0
R3	78,8	20	5	95	12	82.6	70.0	75.1
R4	91.8	29	7	93	3	**96.5**	71.1	76.0
R5	90.8	28	6	94	4	94.3	70.6	75.1
R6	89.2	27	6	94	5	93.6	70.7	75.3
Classifier 2								
L1	66.2	12	5	95	20	68.9	65.6	62.9
L2	81.2	28	25	75	4	89.6	64.9	61.7
L3	80.6	25	17	83	7	84.4	72.3	79.1
L4	67.8	20	27	73	12	70.4	61.3	61.8
Classifier 3								
L1-L4	90.8	29	9	91	3	94.7	–	–
Classifier 4								
R4,L1	**92.9**	30	8	92	2	95.9	–	–

Table 5: Experimental results for semi-supervised method on dataset DS1.

complexity, since an image has to be compared to all training images. The code run-time grows from $O(n)$ to $O(n*m)$, where n is the number of losses to be computed and m is the number of typical training data to compare it to. A space rover such as the Mars 2020 Rover only operates at 200 MHz speed with 0.25MB of RAM [6] (1/10th and 1/8th of an iPhone 8 [13]), this method could quickly become too computationally expensive. Moreover these losses would require the storage of all the training data, definitely not possible in the 2 GB of flash memory [6]. For these reasons the reconstruction losses R7–R10 were omitted.

When training the classifiers there is a large class imbalance between the 98,700 typical images and the 300 novel images. We used Random Over-Sampling (ROS) in conjunction with a cost-sensitive training loss function to solve this bias problem. A smaller dataset was created by combining all the 300 novel images to 3,000 typical ones (hence reducing the dataset class imbalance ration to just 1:10). This smaller dataset was used to train the model, after which the typical images were replaced by the following 3,000 ones to obtain a new dataset used to train the model again. This process was repeated until all 98,700 typical images were used, at which point the typical dataset was shuffled and a new epoch started. During training, the class imbalance ratio of 10:1 was fixed using a Weighted Cross-Entropy (WCE) loss function:

$$L_{WCE} = -\sum y_i * \log(\hat{y}) * W + (1 - y_i) * \log(1 - \hat{y}) \tag{2}$$

whereby y is the true class (0,1), $\hat{y}$ is the classifier output and W is the penalty for the positive class (here $W = 10$). This scales the importance of correctly classifying a class and the assigns penalties corresponding to it based on the relative size of the class.

Table 5 shows the results obtained from each of the classifiers. Each individual loss performs better using the semi-supervised method. While the performance of the latent space based losses is inferior to the reconstruction ones, this is increased significantly when they are combined together in Classifier 3. The best overall solution however turns out to be a mixture of both kinds of losses. This is proven to be the case by combining $Loss_{R4}$ and $Loss_{L1}$ in Classifier 4.

Model	ACC	WACC	TP (FN)	TN (FP)	AUC	Precision	Recall
VAE Classifier 4 (Worst)	0.886	0.893	29 (3)	88 (12)	0.90	0.71	0.91
VAE Classifier 4 (Mean)	0.907	0.908	29 (3)	91 (9)	0.93	0.77	0.91
VAE Classifier 4 (Best)	0.955	**0.949**	30 (2)	96 (4)	0.96	0.88	**0.94**
CAE (Semi-supervised) [3]	**0.962**	0.933	28 (4)	99 (1)	**0.98**	**0.97**	0.88

Table 6: Comparison of our VAE approach with CAE [3]. To demonstrate the robustness of our classifier, we report the best, worst, and average results on 20 randomised test sets.

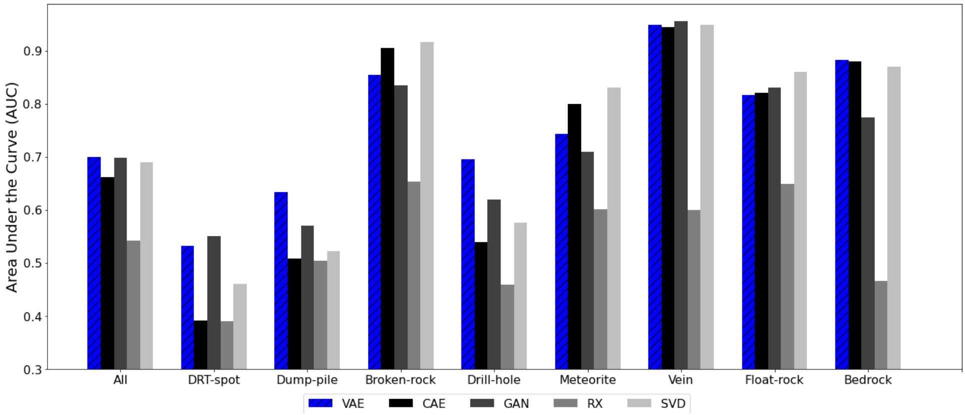

Figure 6: Comparison of our unsupervised method (VAE) with state-of-the-art on DS2.

4.3 State-of-the-Art Comparisons

Dataset 1: In [3], a CAE was used in a semi-supervised method to detect novel images from DS1. A comparison of those results with our semi-supervised Classifier 4 can be used to assess the performance of our model. All the results discussed previously, however, were obtained by randomly selecting 132 test images from the dataset: this means that the results will vary based on the images selected. To fairly assess the performance and robustness of Classifier 4 we have repeated the tests 20 times with differently randomised test sets.

Table 6 shows our best, worst, and average results and compares them to [3]. Our model achieves better WACC and recall values, while under-performs in the ACC and precision. This is because our model detects more novel images (TP=30), but produces more false positives (FP=4). However in this situation, due to the importance of novel targets, we could argue that detecting 2 more novelties at the expense of 3 more error is preferred.

Dataset 2: In [2], a variety of different unsupervised methods for novelty detection (CAE, GAN, RX Detector, SVD) were evaluated on DS2. Here we compare our best-performing unsupervised model (VAE + $Loss_{R9}$) to those methods. Figure 6 shows that, overall, our VAE approach outperforms the Autoencoder (CAE) and is on par with GAN and SVD.

When using our best-performing semi-supervised model (VAE + Classifier 4) on DS2 we significantly improve the accuracy, especially for the poor performing classes. To adopt this approach the novel dataset was split into two sets for training and testing of equal sizes. Table 7 shows how AUC values from the VAE increase when comparing our both methods.

	All	DRT spot	Dump pile	Broken rock	Drill hole	Meteorite	Vein	Float rock	Bedrock
Unsupervised model	0.70	0.53	0.63	0.85	0.70	0.74	0.95	0.82	0.88
Semi-supervised model	0.90	0.94	0.89	0.85	0.95	0.87	0.87	0.82	0.85

Table 7: Comparison of our unsupervised and semi-supervised methods on dataset DS2.

5 Conclusion

In this work, we have shown the benefits of using a VAE over a convolutional AE (CAE) for novelty detection: the probabilistic nature of the model gives the ability to formulate different types of loss functions and consider useful information from the latent space. Our unsupervised, threshold-based approach was very effective and outperformed other methods. In the context of opportunistic science missions, unsupervised approaches are very important considering the absence of labelled training data.

However, if a limited amount of labelled data is available, we have shown how a semi-supervised approach can significantly improve the performance using high-dimensional VAE losses. Best results were obtained when reconstruction and latent space losses were combined. Our approach achieved results comparable to [3], being able to detect more novel images at the expense of a few additional false positives.

Overall, the methods described yielded good results in repeated experiments on multiple datasets, which demonstrates their robustness. This also gives confidence about the possibility of creating autonomous robot systems that can deviate from their intended plan if targets of interests (e.g. novel objects) are detected.

Acknowledgements

The work has been supported by the European Framework Programme Horizon 2020 under grant agreement No 821988 (ADE).

We acknowledge the support of the NVIDIA Corporation with the donation of Titan Xp and Titan V GPUs used for this research.

References

[1] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. In *Special Lecture on IE, 2:1–18*, 2015.

[2] H. R. Kerner, K. L. Wagstaff, B. Bue, D. Wellington, S. Jacob, J. Bell III, and H. Ben Amor. Comparison of Novelty Detection Methods for Multispectral Images from the Mastcam Instrument onboard Mars Science Laboratory. In *4th Planetary Data Workshop*, volume 2151, page 7115, June 2019.

[3] H. R. Kerner, D. F. Wellington, K. L. Wagstaff, J. F. Bell, C. Kwan, and H. Ben Amor. Novelty Detection for Multispectral Images with Application to Planetary Exploration. In *Proc. of Innovative Applications of Artificial Intelligence (IAAI/AAAI)*, 2019. doi: 10.5281/zenodo.1486196.

[4] D. P. Kingma and M. Welling. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] M. Kliger and S. Fleishman. Novelty Detection with GAN. *CoRR*, abs/1802.10560, 2018.

[6] NASA. The Mars 2020 Rover's Brains, accessed 30 March 2020. <https://mars.nasa.gov/mars2020/spacecraft/rover/brains/>.

[7] J. Ocón, F. Colmenero, J. Estremera, K. Buckley, M. Alonso, E. Heredia, J. Garcia, A. Coles, A. Coles, M. Martinez Munoz, E. Savas, F. Pommerening, T. Keller, S. Karachalios, M. Woods, I. Dragomir, S. Bensalem, P. Dissaux, and A. Schach. The ERGO Framework and its Use in Planetary/Orbital Scenarios. In *69th International Astronautical Congress (IAC)*, Bremen, Germany, October, 1–5 2018.

[8] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: One-Class Novelty Detection Using GANs With Constrained Latent Representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. Review: A review of novelty detection. *Signal Process.*, 99:215–249, June 2014. ISSN 0165-1684. doi: 10.1016/j.sigpro.2013.12.026.

[10] L. Ruff, R. Vandermeulen, N. Görnitz, A. Binder, E. Müller, and M. Kloft. Deep Support Vector Data Description for Unsupervised and Semi-Supervised Anomaly Detection. In *ICML 2019 Workshop on Uncertainty & Robustness in Deep Learning*, Long Beach, California, USA, June 2019.

[11] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially Learned One-class Classifier for Novelty Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[12] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylort, and J. Platt. Support Vector Method For Novelty Detection. In *Advances in Neural Information Processing Systems 12*, Denver, Colorado, USA, November, 29 – December, 4 1999.

[13] Device Specification. Apple iPhone 8 CPU, accessed 30 March 2020. <https://www.devicespecifications.com/en/model-cpu/d85c45ac>.

[14] T. C. Stein, R. E. Arvidson, S. J. Van Bommel, K. L. Wagstaff, and F. Zhou. MSL Analyst's Notebook: Curiosity APXS Concentration Data Integration and Mars Target Encyclopedia and Interface Updates. In *Lunar and Planetary Science Conference*, volume 50, 2019.

[15] A. Vasilev, V. Golkov, M. Meissner, I. Lipp, E. Sgarlata, V. Tomassini, D. Jones, and D. Cremers. q-Space Novelty Detection with Variational Autoencoders. *arXiv preprint arXiv:1806.02997*, 2018.

[16] R. Yao, C. Liu, L. Zhang, and P. Peng. Unsupervised anomaly detection using variational auto-encoder based feature extraction. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7, 2019.

[17] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *International Conference on Learning Representations*, February 2018.