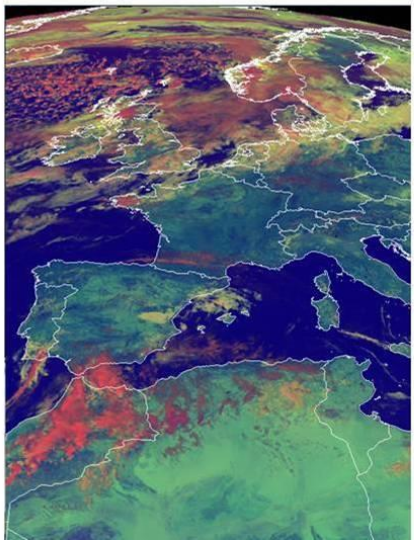
Novelty Or Anomaly Hunter (NOAH)
Executive Summary Report



SSL/10652/DOC/015

Issue 1.0, 17/03/2019

SCISYS UK Limited, Clothier Road, Bristol, BS4 5SS, UK
info@scisys.co.uk | +44 (0)117 916 5165 | www.scisys.co.uk

Intentionally Blank

Project: Novelty Or Anomaly Hunter (NOAH)
Title: Executive Summary Report

Document Control Information	
Contract/ ITT Reference	4000118843/16/NL/LvH
SCISYS Reference	SSL/10652/DOC/015
Issue	1.0
Issue Date	17/03/2019
Customer Reference	ESR
Classification	ESA Unclassified

Role	Name(s)	Signature(s)
Author	Spyros Karachalios	Digitally signed by Spyros Karachalios Date: 2019.05.03 11:01:59 +01'00'
Reviewed	Mark Woods	Digitally signed by Mark Woods Date: 2019.05.03 11:02:29 +01'00'
Authorised for Release	Nancy Phan	Digitally signed by Nancy Phan Date: 2019.05.03 11:03:20 +01'00'

DISTRIBUTION

Copy Number(s)	Recipient
1 (electronic)	Luc Joudrier - ESA
1 (electronic)	SCISYS

ISSUE RECORD

Issue	Issue Date	Sections Affected	Relevant Information
1.0	17/03/2019	All	Initial issue

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 Purpose and scope

This document is the Executive Summary Report of the NOAH activities. It consists of an introduction and a concise summary of the NOAH findings.

1.2 Definitions

1.2.1 Acronyms

Acronym	Definition
AUC	Area Under the Curve
CPU	Central Processing Unit
ESA	European Space Agency
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
NOAH	Novelty Or Anomaly Hunter
NOAH-H	Novelty Or Anomaly Hunter - HiRISE
PFD	Prototype Flight Detector
PR	Precision-Recall
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TET	Training and Evaluation Tool
TPR	True Positive Rate
TRL	Technology Readiness Level

2. SUMMARY REPORT

2.1 Introduction

Within the context of Mars exploration, Novelty Or Anomaly Hunter (NOAH) was designed to research further improvements in novelty detection techniques initially proposed by the Mobile Autonomous Scientist for Terrestrial and Extra-terrestrial Research (MASTER) project.

The following objectives were proposed and completed as part of the NOAH project:

- Develop a generic web-based Dataset Annotation Tool (DAT).
- Create a large crowd-sourced Martian rover image dataset.
- Improve on the previous ESA MASTER project by adding deep learning technology to the algorithmic set.
- Advance the TRL of planetary science autonomy with improved algorithms
- Research the potential of offloading the algorithms on hardware for acceleration.

At the beginning, we started with the development of the Dataset Annotation Tool. The tool is a web-based service that can be run locally on a computer, distributed in a local network across multiple computers and also globally by hosting the application on a cloud service. We then initiated a large crowd-sourcing competition to label a large number of real Martian rover navigation images. The competition used the DAT as the primary tool for annotating the dataset that was used in the NOAH algorithmic framework.

In parallel, algorithmic research was ongoing in order to determine potential ways for improving the performance compared to the MASTER project. A few of alternative saliency detection techniques were researched that used classical machine learning techniques and were added in the NOAH pipeline. Moreover, Deep Learning technology was also researched and integrated into the NOAH pipeline in order to improve the previous algorithmic performance.

Finally, we translated parts of the NOAH pipeline system into flight code implementations in order to advance the TRL of the planetary science autonomy. To further advance the algorithms we researched the potential offloading of the Deep Learning architectures on FPGAs for acceleration and potential use of the system in the upcoming missions.

2.2 Prototype Flight Detector

The Prototype Flight Detector (PFD) delivers a C-implementation of the MASTER algorithms. It does not support training, since this operation costs a lot of energy and therefore it is not expected to be performed on the on-board computer. As in MASTER, the classification is based on kernelized Support Vector Machines (SVM) library. Due to its prototype nature, PFD uses several third-party C++ libraries, these are:

- libSVM to perform image classification
- Piotr Dollar's Toolbox to build feature vector from an image, which is later used by libSVM
- OpenCV for the following operations:
  - Resize images
  - Calculate Discrete Linear Transformation (DCT) and inverse DCT
  - Find contours in a binary map
  - Approximate contours to a polygon
  - Calculate polygon's boundary box

- Load images from paths

PFD uses also math.h header to calculate ceiling and exponential values. It might be possible to avoid these function calls in the future iterations of PFD or provide math operations implemented in newlib library.

Current usage of OpenCV functions comes with few caveats:

- They perform dynamic memory allocation
- Polygon approximation uses recursion
- OpenCV by default enables OpenCL acceleration to many of its algorithms. This means that PFD execution time is comparable to TET-implementation of MASTER algorithms as they all benefit from Graphics Processing Unit (GPU) on modern machines

2.3 Dataset Overview

In order to effectively train the Training and Evaluation Tool (TET) component of the NOAH system an adequately sized volume of training data needs to be provided. For NOAH 5000 annotated images taken from previous Mars missions were classified through crowdsourcing using the Dataset Annotation Tool (DAT) and used for this purpose.

There are six main features that can describe most of the features of interest in the images. Each of the main categories are also divided into related subcategories with additional classifiers for the features if applicable as shown in Table 1. However, it is not always easy, or even possible, to make those decisions and in these cases best guesses are valid.

Table 1: Ontology definition of classes for NOAH

Category	Sub-Category	Classifier 1
Artificial	Foreign object debris	
	Shadows from hardware	
	Spacecraft parts	
	Tracks	
Float Rock	Alteration	Concretions/Nodules
		Crystals
	Magmatic	Dark toned
		Light toned
	Meteorite	
	Sedimentary	Dark toned
		Light toned
Outcrop	Alteration	Bleaching
		Concretions/Nodules
		Veins
	Impact Related	Craters and Ejecta
		Rock Outcrops
	Magmatic	Dark toned
		Light toned
	Sedimentary	Dark toned
		Light toned
Unconsolidated	Drifts	
	Dunes	
	Gravel Beds	Homogeneous
		Structured
Sky		
Don’t know		

2.4 Saliency Evaluation

For our assessments, we adopted the experimental protocol followed in the MASTER Final Report. We plot both the pixelwise ROC and PR curves and calculate the AUC of both curves.

The pixelwise term means that we consider each pixel in terms of counting the True Positive and False Positive numbers instead of the per object saliency evaluation.

By varying the saliency threshold th_s, multiple TPR, FPR, Precision and Recall rates can be obtained for each image. Finally, by keeping track of the achieved average rates over the entries of a given dataset, we compute the mean AUC and identify the best performing algorithm.

For the Attention Mechanism the following algorithms were implemented and tested:

1. Image Signature RGB
2. Global Contrast Saliency
3. Graph Based Manifold Ranking Saliency
4. FasterRCNN Regional Proposal Network

Table 2 shows the performance evaluation of the Attention Mechanism in terms of accuracy and execution time. By comparing the accuracy of the algorithms, we see that the Image Signature RGB has a slight advantage over the rest of the algorithms apart from the Graph Based which is less accurate. In the Saliency Evaluation there is a pattern where the PR AUC values are a lot higher than the ROC AUC values. This can be explained because in the PR case we measure the precision of the system meaning how many of the true positives are actually correct versus the total true positive rate of the system. Because the output of the saliency areas is usually very small but very precise it causes the precision to be very high. On the other hand, in the ROC case, both the false positive rate and the true positive rate can be very low as there can be a lot of false negatives and true negatives. However, since the results are very close to each other when comparing the different algorithms, a decision on whether to use one over another is not clear and depends on what someone might consider as salient pixel in an image or not.

Table 2: Performance evaluation for the Attention Mechanism algorithms

Algorithm	ROC AUC	PR AUC	Total Execution Time	Average Time Per Image
PFD Image Signature RGB	0.619	0.907	5.481s	0.004s
TET Image Signature RGB	0.619	0.907	3.306s	0.002s
TET Global Contrast	0.611	0.896	17m 48s	0.715s
TET Graph Based	0.537	0.878	35m 58s	1.444s
TET Faster RCNN	0.606	0.904	1m 13s	0.049s

In terms of the speed of algorithms, the Image Signature RGB (both TET and PFD implementation) is fast and can produce saliency maps faster than 250 fps (frames per second). The deep learning algorithm FasterRCNN is still comparably fast and performs better than the pure CPU implementations of the Global Contrast and Graph based by producing saliency maps at a rate of 20 fps using the GPU. However, the execution time differs from machine to machine as it is heavily depended on the processing power of the computer.

Figure 2 shows an example of the saliency output of the Attention Mechanism component of the algorithms used in NOAH. Each of the algorithms behaves differently from each other and since the accuracy results are similar it is difficult to chose one over the others. However, by observing

the results we can see that the Image Signature RGB, Global Contrast and Graph Based algorithms that are based on classical computer vision algorithms, they are able to pick up as

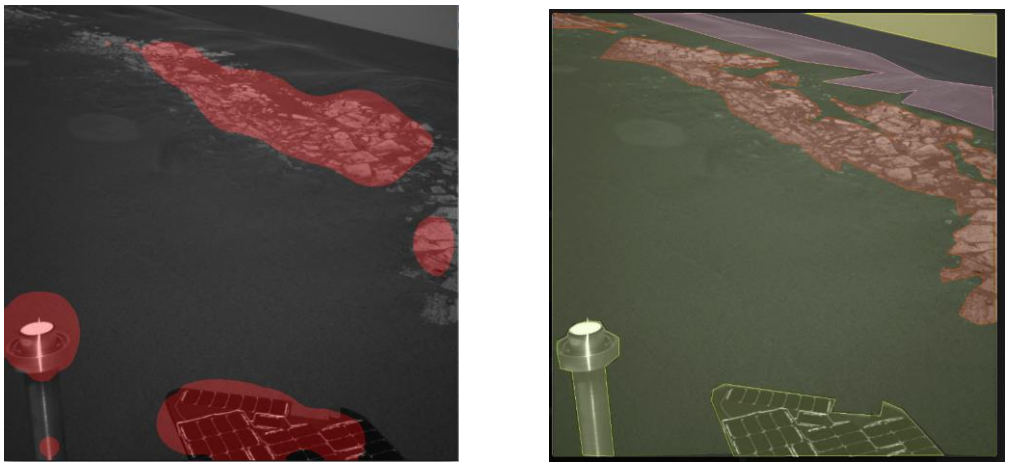


Figure 1: Example of saliency output

Left: TET Image Signature RGB, Right: Ground Truth annotations

salient the most prominent features in the images. On the other hand, the deep learning approach has the ability to find a lot of features that can be salient or not as it can have a very large number of region proposals. The accuracy of the algorithm is limited, however, due to the fact that the algorithm is outputing bounding boxes and not polygons as the annotations, that causes unwanted pixels to be marked as salient and thus reduces the accuracy.

2.5 Classifier Evaluation

In order to evaluate the Classifier for Known Phenomena component we plot, as in the Attention Mechanism case, the ROC and PR curves and we calculate the AUC of the graphs for each of the class. In the classifier evaluation case, we evaluate the algorithms on the object scale as defined by the ground truth bounding boxes.

By varying the classifier confidence threshold th_c, multiple TPR, FPR, Precision and Recall rates can be obtained for each image. Finally, by keeping track of the achieved average rates over the entries of a given dataset, we compute the mean AUC and identify the best performing algorithm.

For the Classifier for Known Phenomena the following algorithms were implemented and tested:

1. Support Vector Machine (SVM) Classifiers
2. FasterRCNN Classification Network

In the figures Figure 2 and Figure 3 we present the ROC and PR curve results of each algorithm by combining the classes into the 2nd level of aggregation

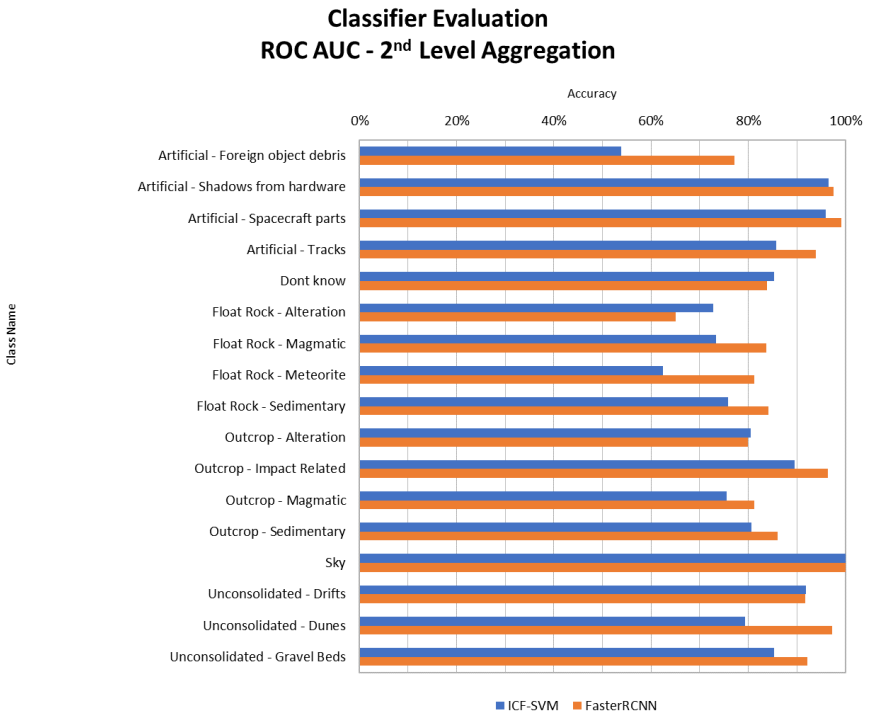


Figure 2: AUC accuracy of the Classifier for Known Phenomena algorithms based on the ROC curves for the combined classes at the 2nd level of aggregation

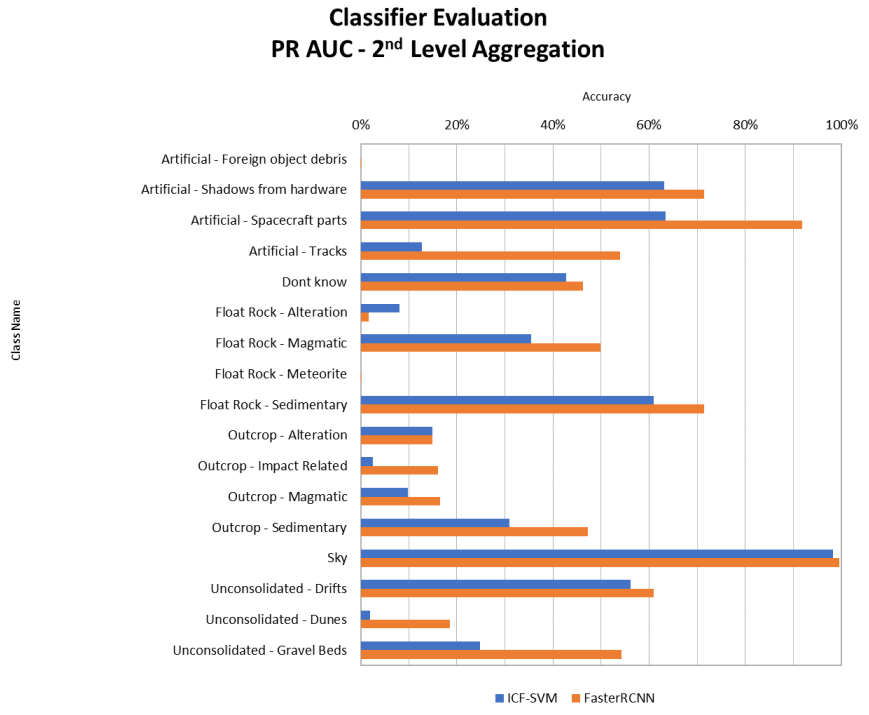


Figure 3: AUC accuracy of the Classifier for Known Phenomena algorithms based on the PR curves for the combined classes at the 2nd level of aggregation

From the results in the previous figures, it is evident that the Faster RCNN deep learning approach outperforms the classical machine learning approach of the ICF-SVM in almost all classes and in every level of aggregation. However, the accuracy in precision of both approaches is very low when using the full set of classes or the sub-categories. That can be explained for various reasons:

- The number of examples in some of the classes is not enough to train a model with high accuracy.
- The distribution of the annotations is not uniform across all classes causing confusion to the classifiers as some classes are over represented in the dataset.
- In the 3rd and 2nd level of aggregation there are not many features in the images to distinguish that level of detail
- The number of total classes might not be suitable for the selected deep network architecture that was used and different approaches with ensemble of networks or shallower or deeper networks might be preferable

Table 3 shows the performance of the two different classification methods (Faster RCNN and ICF-SVM for the TET and PFD implementations) by comparing the time required to train the full set of models for all classes and the execution time of the classification of the images. It is evident that the FasterRCNN implementation benefits from the GPU processing power and performs the classification in around 3FPS while the ICF-SVM which uses only the CPU is around 25 times slower.

Table 3: Time performance of the classification algorithms

Algorithm	Total Training Time	Total Classification Time	Average Time Per Image	Average Time Per Bounding Box
TET Faster RCNN	4h 15m	8m 27s	0.34s	0.008s
TET ICF SVM	487h 25m	3h 37m	8.738s	0.728s
PFD ICF SVM	N/A	16d 8h	15m 48s	2m 15s

Moreover, in the case of the FasterRCNN the classification of the bounding boxes is faster since there is only one deep network that is used for all the classes. On the other hand, the ICF-SVM uses one model per each of the classes that causes a big impact in loading the models and running inference for each model separately. Also, the big difference between the TET and the PFD implementation is that in the first case we are able to leverage multiprocessing across multi CPUs of the system while in the PFD case we are constrained to only one image and one model sequentially.

Finally, another important difference between the two algorithms is the total training time needed to produce the models where again the FasterRCNN leverages the use of a GPU and the training time is around four hours while the ICF-SVM requires a total training time of almost a month even when using the multiple CPUs of the system.

2.6 Novelty Detection Evaluation

The novelty detection is derived by combining the output of the classifier component and the frequency of the classes in the dataset which is an invariant property of the dataset. This means that there is no scientific reason to measure the performance of the novelty component individually but rather the novelty detection as a whole system.

The AUC from the PR curve showed that the system recognises novelties with accuracy of 6% with Recall being at around 65% and Precision at around 7%. This means that the system can indeed identify the novelties we introduced in the dataset but at a cost of an increased number of false positives. An example of the output of the system can be seen in Figure 5.

The example in Figure 5 shows the output of the NOAH system on an input image. The chosen algorithms based on the previously discussed sections are the TET Image signature RGB for the saliency evaluation and the FasterRCNN as the classifier of the regions of interest. The example shows that three regions have been detected on the image and based on the classification output and the novelty detection thresholds, two of the objects have been identified as “Artificial – Spacecraft parts” and have been marked as “Expected” and are not of interest to the novelty system. On the other hand, the third region has been correctly classified as “Outcrop – Sedimentary – Light toned” but has been marked as a classified novelty.

We can see from the classifier output on the right side of Figure 5 that the classifier is confused for the object 2 and three different classes are contributing to the confusion. These are the “Outcrop”, the “Drifts” and the “Dunes”. The novelty identified here is that the classifier has not seen many examples of outcrops with drifts and dunes around it and therefore identified it as a novel outcrop. By comparing the output to the original ground truth in Figure 2 we can see that the confusion is indeed correct because the bounding box that encloses the area identified as “Object 2” indeed contains the identified features of the associated classes.

However, as the example also shows, the precision of the novelty detection system is very low since in this case the “Outcrop” is not novel as it is a known class from the dataset. The low accuracy can be explained for various reasons:

- The final output of the NOAH system accumulates all the errors from the previous stages of the saliency detection and the classification output. In both cases as already discussed the precision is low and when combining them the result is lower than expected.
- The conversion from polygonal annotations to bounding boxes is creating a lot of confusion. The original MASTER pipeline that NOAH was based on is using bounding boxes for the regions of interest and the classification. That as shown above is causing excessive confusion to the system in both the training and the validation stages as the features of interest inside a polygonal annotation are polluted with features that belong to other classes when converted to bounding boxes.

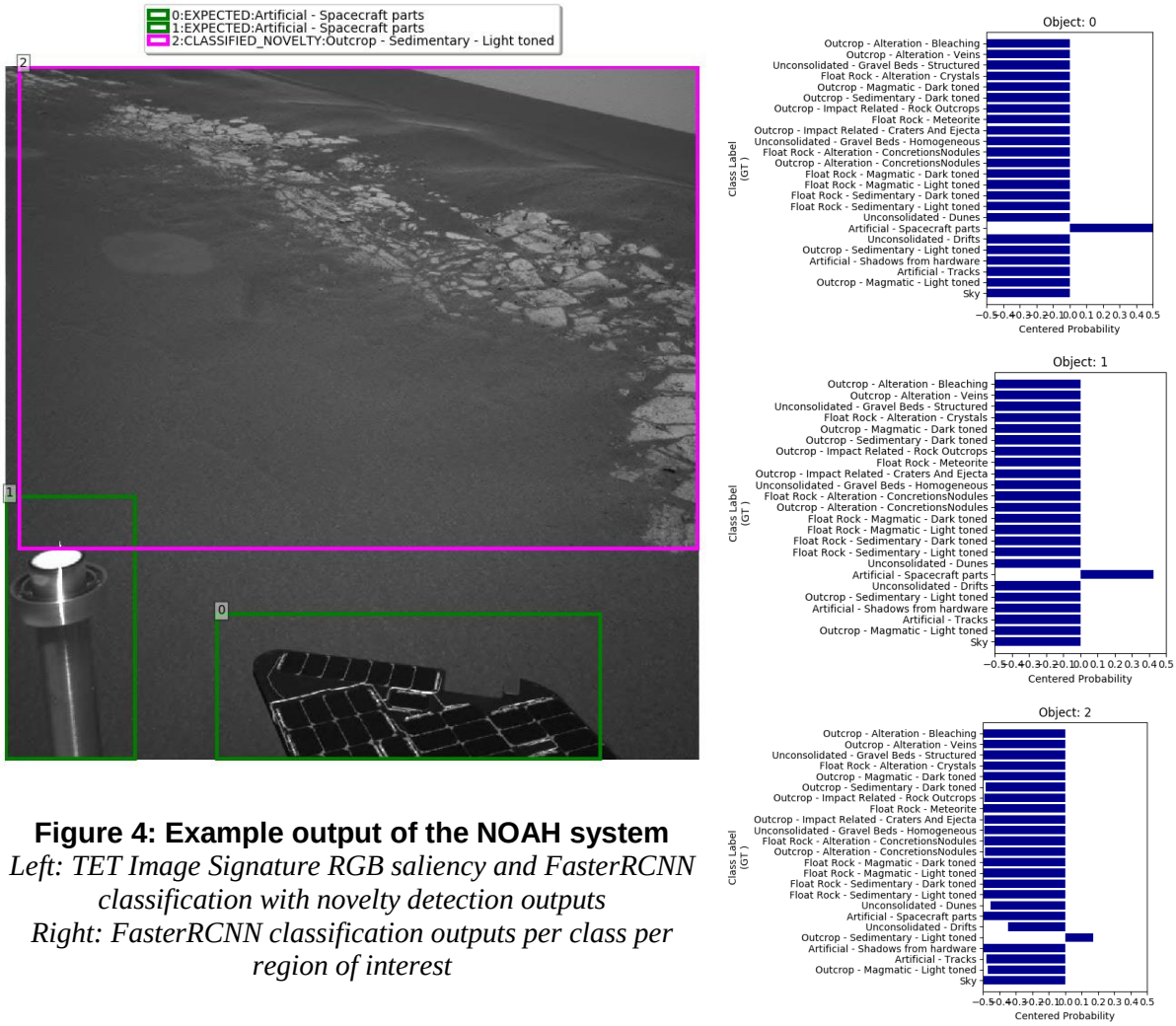


Figure 4: Example output of the NOAH system
Left: TET Image Signature RGB saliency and FasterRCNN classification with novelty detection outputs
Right: FasterRCNN classification outputs per class per region of interest

2.7 Conclusions & Future Development

The Novelty Or Anomaly Hunter (NOAH) system was designed, within the context of Mars exploration, to research further improvements in novelty detection techniques initially proposed by the Mobile Autonomous Scientist for Terrestrial and Extra-terrestrial Research (MASTER) project. In order to achieve increased performance and better evaluation of the system various objectives were proposed and completed as part of the NOAH project.

One of the key objectives of NOAH was to improve on the previous ESA MASTER project by adding also deep learning technology to the algorithmic set. The available options at the start of the project were aligned to the NOAH pipeline and allowed us to easily incorporate a deep learning technique into the framework. The deep learning approach of FasterRCNN was proven to outperform the classical machine learning approaches used in MASTER and NOAH, showing the importance of using such an algorithm. However, the FasterRCNN and in general the MASTER and NOAH pipeline that uses bounding boxes was proven to lack in performance due to the restriction of the bounding boxes. The pipeline of the NOAH can be improved in various ways:

Finally, our research showed that there is a great potential of offloading the Deep Learning architectures of the current NOAH pipeline along with other algorithms onto FPGAs in order to increase speed performance and give us the ability to operate the algorithms on potential flight qualified hardware.