# Design & Prototyping of a Minaturised Forwards Looking Imager using Deep Learning for Responsive Onboard Operations

**Steve Greenland [(1)], Murray Ireland [(1)], Chisato Kobayashi [(1)], Peter Mendham [(2)], Mark Post [(3)], David White [(4)]**

*[(1)] Craft Prospect Ltd, Tontine Building, Glasgow, G1 5ES, United Kingdom,*
*+44 (0) 75 88 888 146, steve.greenland@craftprospect.com*
*[(2)] Bright Ascension Ltd, Edinburgh, United Kingdom.*
*[(3)] University of Strathclyde,Glasgow, United Kingdom*
*[(4)] University College London, London, United Kingdom*

## ABSTRACT

This work presents the design and prototyping work of a miniaturised camera system with integrated 'deep learning' neural network capabilities, developed within a framework for implementing autonomous data processing onboard small and nanosatellites. The framework targets low-resource algorithms developed in other sectors including autonomous vehicles and commercial machine learning. For proof of concept, the system has been initially trained for real time cloud detection and classification, looking 1 min ahead of the satellite to enable responsive decision making for Earth observation and telecommunication applications. The design has been miniaturised and modularised to allow accommodation on small and nanosatellite systems. Flight representative and heritage components have been selected for prototyping. Compatibility of the autonomy framework with ECSS and CCSDS standards and existing off-the-shelf flight software was evaluated. A simulator to facilitate end to end testing of the system has been developed using existing data sets as input, incorporating distortions to test robustness. Results show that a competitive low power < 2 W system can be delivered, with the chain < 5 seconds from capture to input into the onboard planning and with timing consistent with continuous real time decision-making.

## 1 INTRODUCTION

With progress in miniaturised high performance sensing capabilities, and despite technology improvements in data rates, small and nanosatellite systems will remain downlink limited; able to capture more data than can be returned to the ground cost-effectively in traditional raw or near-raw forms. The current state of the art is the reported as 120 Mbps X-band downlink achieved by Planet on their satellites using an extensive ground network sized for ~5 TB of downlink per day [1], whilst the incoming generation of multi-, hyper- spectral and high resolution CubeSat imagers are capable of generating data rates in excess of 100 Mbps [2]. This presents a challenge both to the ground station infrastructure, solved to some extent by shared outsourced facilities, and technological enhancements in downlink bandwidth. Growth in the market for new applications like video capture [3] will further push downlink demands, while the embedding of existing ground-based image processing algorithms into onboard systems is non-trivial especially in limited resource nanosatellites, necessitating new approaches. Pushing the boundaries of these systems beyond Earth orbit present additional challenges around relay availability and bandwidth, and delay-tolerance, leading to more autonomous approaches. This paper presents part of a larger work into onboard data autonomy, part funded by the Centre for Earth Observation and Instrumentation on behalf of the UK Space Agency, performed in a consortium led by Craft Prospect Ltd.

## 2 FORWARDS LOOKING IMAGER INSTRUMENT CONCEPT

The Forwards Looking Imager (FLI) instrument provides analysis of the upcoming environment of the satellite to allow real-time decision making. This can take a number of forms: it may be the prioritisation of sensing targets from a list, repositioning and slew manoeuvres to capture targets of opportunity, or reassignment of targeting tasks among a constellation. Figure 1 shows the system architecture of the FLI instrument.

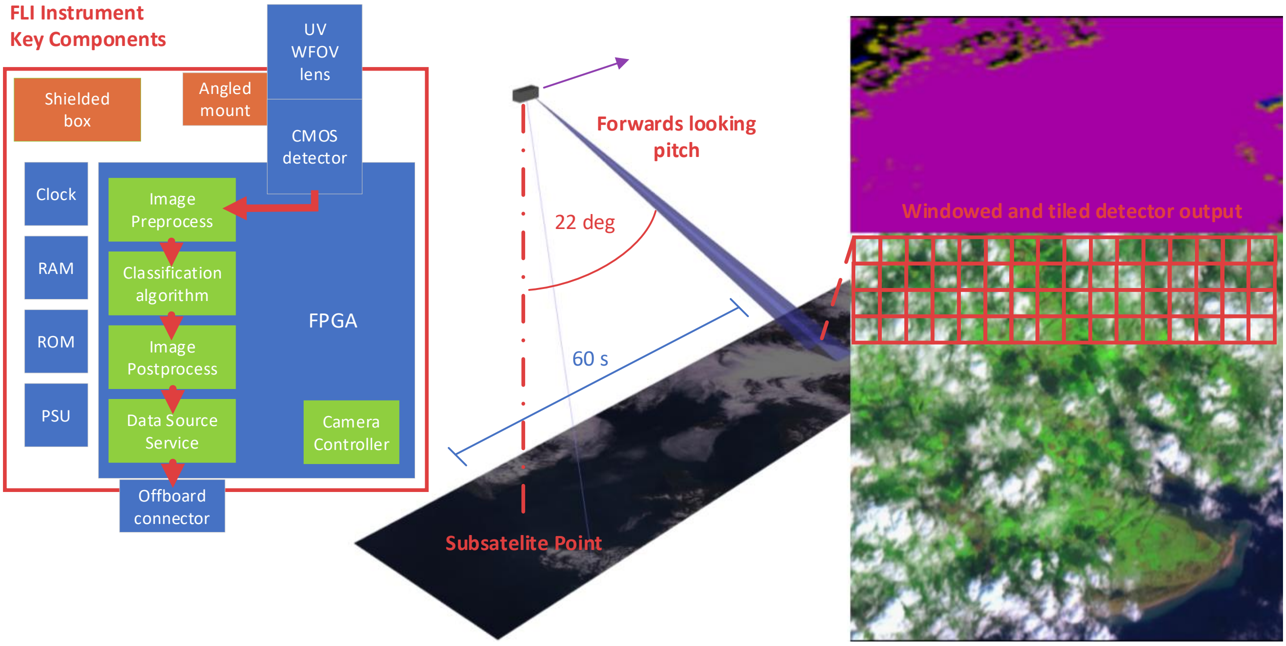

**Figure 1; System architecture of the Forwards Looking Imager concept**

Within this field, the cloud detection case represents a near-term opportunity for responsive Earth observation imaging, and an opportunity to apply the framework described in previous work [4] and summarised here in Sec. 3. The operations concept considered is real-time detection of clouds using forwards looking wide field of view imager to allow a second near-Nadir pointing payload to be targeted at cloud gaps, such as for high resolution Earth imaging or to enable optical laser communications. In all avoidance scenarios, a rapid reacquisition of the payload is necessary, with the assumption that the overhead is significantly less than the active operations. In subsequent passes retasking the satellite or satellites within the same network can fill the gaps, provided information on previous acquisitions can be efficiently exchanged.

The FLI is designed to provide imagery across a swath sized for the application and dependent upon the range of viable targets. Two design points have been considered in development: (a) a wide swath, with a design point defined maximise both access to targets across a wide elevation and response time to subsatellite point, and (b) a moderate swath, with the design point driven by a factor of CubeSat imaging state of the art, taken as the PlanetScope instrument [5], in order to meet a minimum response time. These design points imply differing levels of control authority, resolution, and responsiveness for a targeting payload.

(a) Taking a design point altitude of 400 km and ignoring oblateness effects, the swath to 30 deg elevation is 1480 km. Based on a camera with a detector width of 2000 pixels, each pixel will equate to around 750 m. With a pitched forwards looking angle of 55 deg the satellite response time after imaging would be up to 2 min, however this would be at the expense of distortion the imagery due to the viewing angle at elevations close to 30 deg.

(b) Taking a design point of the FLI operating over 10x the swath of the current state of art for a CubeSat imaging system at 4 m GSD, would give a 40 m per pixel system, over a 2000 pixel width driving a total swath of 80 km. With a desired advanced time of 60 s in this case, a shallower pitched forwards angle of 22 deg leads to significantly improved viewing factor and distortion at elevations above 60 deg.

## 3 AUTONOMY FRAMEWORK

It is recognised that the FLI instrument represents an enabling technology for autonomy, but must exist within a wider framework of hardware and software elements to realise the goal of responsive onboard operations. Previous work has been undertaken to characterise this framework by extension of existing ECSS and CCSDS standards [4]. The CCSDS Mission Operations Service Standard (MOSS) [6] was identified as a basis for adding the autonomy building blocks, using a Service Oriented Architecture (SOA) to facilitate transition from more common ‘monolithic’ architectures to provide a service-driven networked system for new applications. This allowed the definition of a high level autonomy framework visualised in Figure 2 and referenced against existing off-the-shelf space software, GenerationOne [7] produced by Bright Ascension Ltd.

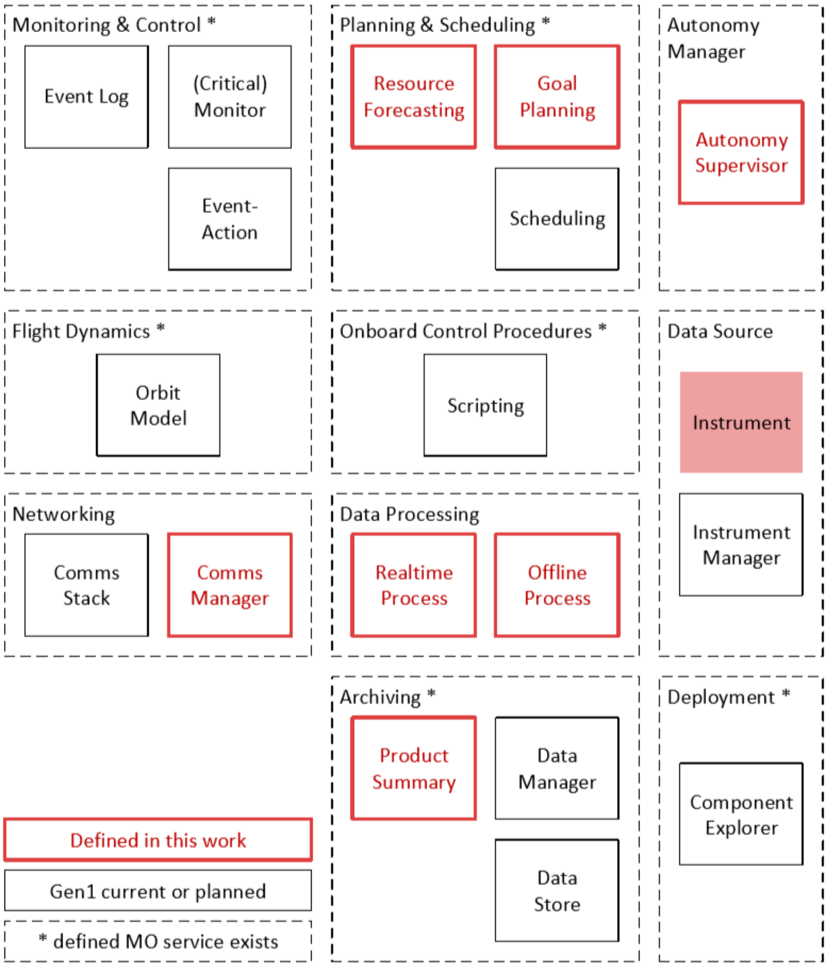

**Figure 2; Autonomy framework architecture, defined with respect to MOSS, identifying new services, and functional groupings based on existing standard off-the-shelf software for CubeSats**

MOSS enables a modular approach to the operational system design through the identification of components which interact through open and published service interfaces. In reviewing the suitability of MOSS for the operational concept anticipated here, additional classes of services and element were identified.

- The Communications Management Service permits the management of communications functions, controlling aspects such as link utilisation and data routing.
- The Data Source Service captures the ability to generate data, either from the original source, in the case of an instrument, or as an output of a data processing chain. Data can either be sourced for real-time applications ("pushed" to other onboard functions) or archived for later use.
- The Data Processing Service provides the ability to query and configure data processing chains which can be used with a data source or archived data.

Within the framework shown in Fig. 2, the FLI represents elements within the Data Source and Processing Services, which might be expected to sit within an implementation such as is described in Fig. 3 to enable an autonomous imaging system for a next generation Earth observation nanosatellite.

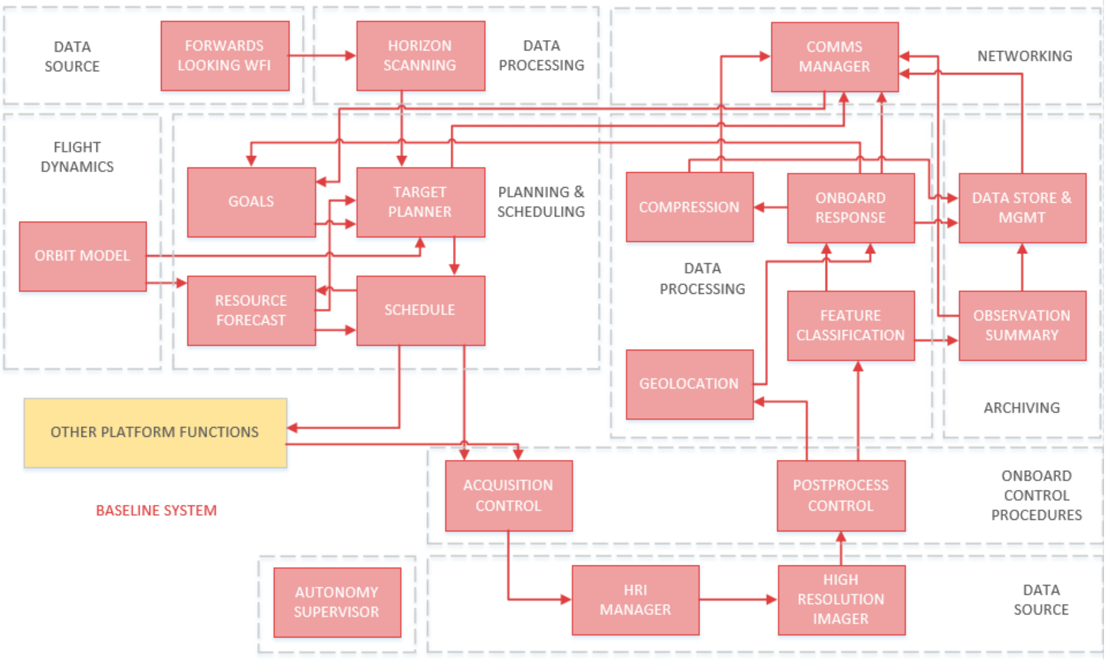

**Figure 3; Autonomy framework implementation within the FLI instrument operations concepts**

## 4 CLOUD DETECTION ALGORITHM

Within the implementation architecture proposed in Fig. 3, a survey of potential algorithms for each of the functional blocks was undertaken. For the FLI system, an evaluation of three different algorithms were brought forward for further study, outlined in Table 1. Using available open source implementations of code initially, data from LandSat-7 images was processed to compare anticipated processing speeds and implementation challenges for these techniques. Of these, both TextureCam and VGG19 were taken forward, to demonstrate two different toolchains for the target hardware device: a Zynq Z-7020. Whilst the TextureCam C code was well maintained and compact enough for implementation and acceleration on the Zynq, a number of flavours of Convolutional Neural

Network algorithms were then explored to assess (a) the acceleration possible to meet a minimum performance threshold and (b) the ease of transfer from a high level language into an FPGA.

**Table 1; candidate detection algorithms for the FLI instrument**

| Algorithm class | Reference implementation | Advantages | Disadvantages |
|---|---|---|---|
| Rule-based | Fmask | Well known and understood<br>Deterministic | Requires specific wavelength inputs including infrared<br>Require a catalogue of georectified reference images which is not available in real time<br>Computationally expensive |
| Random forest | TextureCam | Well known and understood<br>Rapid inference | Adequate pre-processing of data set might be challenging onboard<br>Requires good feature engineering |
| Deep learning | VGG19 (CNN) | Rapid inference | Requires training data set<br>Training computationally expensive |

Deep learning CNNs consist of a series of multiple layers of different types, each tailored for a given function. The convolution layers will extract features from their inputs (whether the original input layer or else deriving higher-level features from those extracted by previous layers). Each convolution layer contains a set of kernels, that is clusters of processing units that are applied across their input (convolved, analogous to passing a filter across an image) using the same weights at each point. By repeatedly applying the same kernels across many points in the image, the space of weights that must be learnt is greatly reduced, improving scalability. Rectified Linear Unit layers (ReLU) are usually applied after a convolution layer; they output the maximum of their inputs, and improve the efficiency of training. Pooling layers improve scalability, by downsampling and thus reducing the dimensionality of the input to subsequent layers. Fully-connected layers subsequently learn non-linear functions of the features learnt by previous layers.

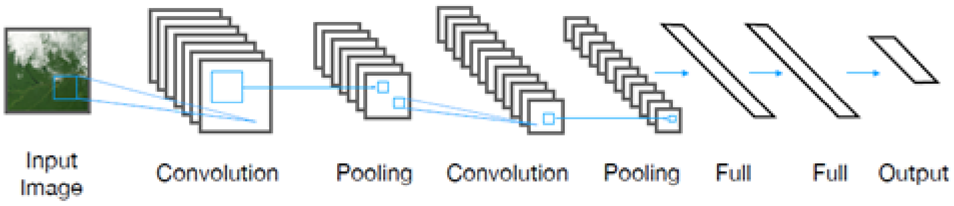

**Figure 4; an example of a simplified convolutional neural network**

## 5 IMPLEMENTATION

A prototype of the end to end system was next developed, using the cloud detection case for a system-in-the-loop test bench to verify the onboarding process and as the first step for qualifying new algorithms in flight ready hardware. Whilst the goal of the prototype system (see Sec. 6.2) was to act as a proof of concept focussed on the cloud detection case; the bench was also designed to be reconfigurable to other use cases. To facilitate rapid deployment of new algorithms into a LEO environment, a number of additional strategies were identified during implementation,

- target Zynq FPGA was selected with known flight heritage systems, Zynq Z7020

- use of tools to enable transfer from high level languages associated with deep learning like Keras, Caffe and Python into accelerated code using high level synthesis and overlays
- adaption of existing open source libraries for image processing
- transfer learning of pre-trained neural networks for application to the Earth observation case
- application of techniques to discretise the convolutional neural network

A common limitation of machine learning in general is the availability and suitability of training data sets. Although LandSat data was used for the input into the simulation, the deep learning algorithm itself was trained on pre-classified data available from Planet, demonstrating the potential for inference from a different data set to that trained. It is noted however, that although distortions were applied to the data set to represent the as viewed profile of the images, the trained data set used was provided after anomalies have been discarded and radiometric calibration applied; future work will acquire or emulate a more representative dataset at the detector source.

### 5.1 Simulation Environment

A modular python-based simulation environment has been developed for the onboard data autonomy program, able to run as a standalone model or to interface with hardware under test. The simulation outputs a visualisation of the orbit and classification algorithm performance, and outputs test data. Further work will interface this simulation to existing off-the-shelf spacecraft software emulators. The simulation feeds the hardware under test with raw image data from LandSat, and receives back a tiled classification. A path finding function was also applied to determine where to focus the primary payload or, where no targets existed for a given period, inactivate the primary payload to conserve power. For initial testing a simple discrimination of cloudy, partial and clear was selected, allowing more ready human visual confirmation of correct function.

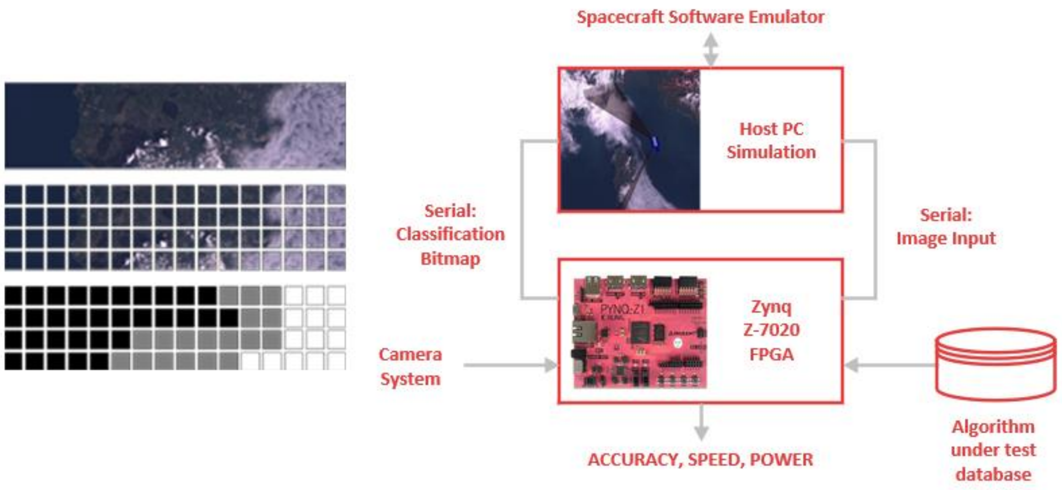

**Figure 5; embedded deep learning system-in-the-loop simulation**

## 6 PROTOTYPING RESULTS

Typical results from the system-in-the-loop prototyping and test run is shown in Fig 6. The test run divided test images of swath 800 x 200 pixels into 64 tiles (16 x 4) for processing by the deep learning algorithm. At the 400 km design point each tile would be just under 2 km in width approximating an area access rate of 250 km2/s.

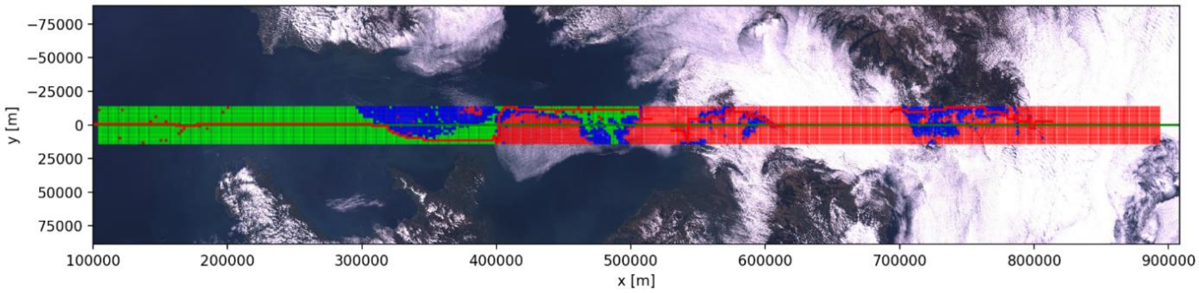

**Figure 6; prototyping results superimposed over the input image within the system-in-the-loop simulation [colours: R: cloudy, G: clear B: partial]**

## 6.1 Performance

There are three key performance metrics for the MLI instrument: accuracy/confidence, power and timing. The current algorithm after training shows a 98% accuracy in correctly classifying the tile against the human observed data set. This provides a baseline for relative performance but was non-optimised and could be improved with the training data set used.

The software-only and accelerated implementations of the deep learning algorithm were compared in terms of power and timing for the accuracy level above. In a test run of 100 images, the time to complete was 5293 s in the former, which was accelerated to 19 s in the latter, an improvement of nearly x300. Although the instantaneous power consumption of the accelerated hardware was higher due to the increased logic under load, the power reduction available is much lower.

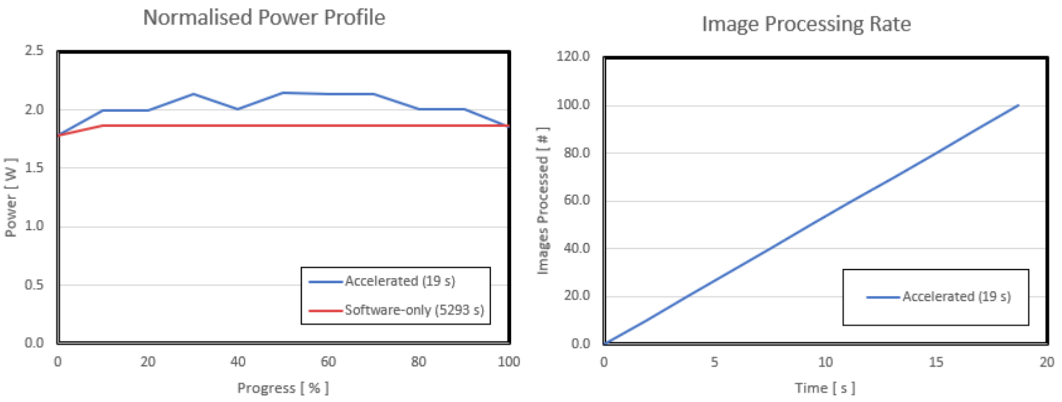

**Figure 7; comparison of power consumption between software-only and accelerated normalised with respect to % completion and image processing**

In a LEO context for design point specified in Sec.2(b), it would be expected that the forwards looking imagery requires processing at an area access rate equal to the swath width multiplied by the ground speed, around 580 km$^2$/s. Maintaining the same tile to pixel ratio to the trial, which was found to be strongly impacted by the training data set available, would require a minimum processing rate of 2.2 images/s. This hard timing constraint is therefore met by the accelerated algorithm which can process 5.3 image/s, allowing real-time operation with a margin of 140%. Taking the inverse would equate to a duty cycling in the current (non-optimised) version of > 60%, leading to a mean power consumption estimate of 0.9 W. It is unlikely that the system would be powered down between samples, rather in maintaining an overall consumption of 2 W and with optimisation, it is plausible a

3x repeated sampling strategy per tile might be used to increase overall confidence. With latencies in the data transfer considered, a response time of less than < 5 s from capture to onboard planning is then expected.

## 6.2 Objective compliance

Based against the objectives specified for prototyping in Table 2, all high-level requirements have been verified for the cloud detection case.

**Table 2; requirement compliance for the high level objectives of the prototyping activity**

| Requirement | Description | Verification | Compliance |
|---|---|---|---|
| Proof-of-concept | Demonstrate autonomy in the system loop simulation | Use imagery from forward camera to enable onboard decision-making and vertical camera pointing | C |
| Cloud detection | Deploy an algorithm for cloud detection | Coordinates of clouds in supplied camera image are returned to satellite OBC | C |
| Integrated unit test | Develop unit test for system-in-the-loop | Images are transmitted from simulation to FPGA and cloud coordinates are returned | C |
| Acceleration | Accelerate algorithm in FPGA programmable logic | Speed and power draw of algorithm running in PL is compared to performance in PS | C |

# 7 CONCLUSION

This work has outlined the design and prototyping of the FLI instrument embedded with a deep learning algorithm for target classification. The work has shown that a cost and power effective system may be implemented using emerging deep learning concepts to enable more responsive onboard operations for Earth observation in real time. A framework for integrating this system into existing off-the-shelf software for CubeSats and aligned to existing standards has been considered.

An initial use case of the system for cloud detection in LEO has been developed. Such a system may be accommodated within a CubeSat form factor for immediate application and in support of more complex operations of high value missions or payloads. For the MLI cloud detection use case, a preliminary design point is explored and tested, demonstrating the feasibility of continuous real-time operations at a 0.9 W power consumption if duty cycled providing an area access rate equivalent to scanning for targets over an 80 km swath and with a 60 s advance notification. Other swath width appear feasible provided appropriately scaled training data sets can be acquired; timing up to 120 s may be feasible noting the distortions in the field of view this implies.

Future work will allow flight demonstration of the core systems within the MLI instrument; this will begin with a planned drone flight in Q2 2018 looking towards a wind turbine use case and which can also serve to demonstrate the rapid application of transfer learning. This, together with acquisition of raw in-orbit imaging from a comparable camera system, will address concerns over image quality in the as-deployed system in comparison to the training data used. Following space ruggedization, an initial MLI protoproduct is planned for the second half of 2018.

## 8 REFERENCES

[1] Henry, C, “Planet Preps Ground Network to Handle 5 Terabytes of Data Per Day,” www.satellitetoday.com, August 2016.

[2] Selva, D, and Krejci, D, “A survey and assessment of the capabilities of CubeSats for Earth observation,” Acta Astronautica, Vol 74, pp50-68, 2012.

[3] Earthi, “Earthi releases its first full-colour video of Earth taken from space,” Press Release, www.earthi.space, April 2018.

[4] Greenland, S., “System Framework for Autonomous Data Processing Onboard Next Generation of Nanosatellite,” Reinventing Space Conference, BIS-RS-2017-5, 2017.

[5] Planet Labs, “Planet Imagery Product Specification,” www.planet.com, October 2016.

[6] CCSDS, “Mission Operations Service Standard,” Greenbook, CCSDS-520.0-G-3

[7] McCrum M, and Mendham P, “Integrating Advanced Payload Data Processing in a Demanding CubeSat Mission,” CalPoly CubeSat Workshop, April 2015.