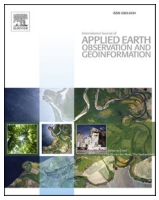
International Journal of Applied Earth Observations and Geoinformation 97 (2021) 102279



Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag





JellyNet: The convolutional neural network jellyfish bloom detector

Ben Mcilwaine, Mónica Rivas Casado *

School of Water, Energy and Environment, Cranfield University, College Road, Cranfield MK43 0AL, United Kingdom

ARTICLE INFO

Keywords:
Jellyfish bloom
Unmanned aerial vehicle
Machine learning
Convolution neural network
Remote sensing
Deep learning

ABSTRACT

Coastal industries face disruption on a global scale due to the threat of large blooms of jellyfish. They can decimate coastal fisheries and clog the water intake systems of desalination and nuclear power plants. This can lead to losses of revenue and power output. This paper presents JellyNet: a convolutional neural network (CNN) jellyfish bloom detection model trained on high resolution remote sensing imagery collected by unmanned aerial vehicles (UAVs). JellyNet provides the detection capability for an early (6–8 h) bloom warning system. 1539 images were collected from flights at 2 locations: Croabh Haven, UK and Pruth Bay, Canada. The training/test dataset was manually labelled, and split into two classes: 'Bloom present' and 'No bloom present'. 500 × 500 pixel images were used to increase fine-grained pattern detection of the jellyfish blooms. Model testing was completed using a 75/25% training/test split with hyperparameters selected prior to model training using a held-out validation dataset. Transfer learning using VGG-16 architecture, and a jellyfish bloom specific binary classifier surpassed an accuracy of 90%. Test model performance peaked at 97.5% accuracy. This paper exhibits the first example of a high resolution, multi-sensor jellyfish bloom detection capability, with integrated robustness from two oceans to tackle real world detection challenges.

1. Introduction

Coastal operators are fighting a relentless battle against both jellyfish and submerged aquatic vegetation (Flynn and Chapra, 2014; Purcell, 2005). Large accumulations of biomass can lead to blockages, disruptions and damages to mechanical water intakes but can also decimate coastal fisheries (Purcell, 2005; Småge et al., 2017). Jellyfish can cause severe disruption through the clogging of filtration systems (Koo et al., 2017). It has been shown that jellyfish can cause annual damages of up to US$ 205 million in Korea, and up to US$ 2 million per day at a nuclear power station in Ontario, USA (Kim et al., 2016; Nuclear Energy Institute, 2015). Jellyfish form blooms that exist in a wide range of environmental conditions (Lucas, 2001). These conditions are becoming increasingly common due to global climate change and reductions of top-down predator controls (Purcell et al., 2007; Daskalov et al., 2007). Despite the rigorous descriptions and reporting by both Hamner and Dawson (2009) and Graham et al. (2001), there still appears to be an absence of a formal jellyfish bloom definition with respect to density. An aggregation of mature *Aurelia* spp. jellyfish medusae consisting of a minimum of one individual per m^3 of water would certainly cause major disruption to coastal operators, especially if totalling upwards of 1,000 total mature individuals. For comparative reasons, this paper will use this definition of a jellyfish bloom. Other definitions based on biological principles may be developed in the future. Here, we have focused on criteria defined by operator and industry experience.

Currently there are no fully functional warning systems in place for an impending jellyfish bloom. If the challenges facing jellyfish bloom detection in coastal environments are to be addressed, the development of a suitable imaging and monitoring system is, naturally, a necessity. For a remote sensing system to be practically useful, a warning of 6–8 h prior to a jellyfish bloom arrival has been suggested (EDF, personal comment). This time frame would be an improvement on current systems but also crucially allow disruption mitigation teams to maximise asset protection measures (Takizawa, 2005). To date, remote sensing techniques have proven the most likely candidate for fulfilling the temporal conditions of an early warning detection system (Mcilwaine et al., 2019). Only a small number of attempts have been successful in remotely sensing jellyfish blooms; unpredictability of presence and high transparency in water being two key challenges to their detection. Satellites, light-aircraft and unmanned aerial vehicles (UAVs) have all been used to varying success to remotely sense jellyfish blooms (Schaub et al., 2018; Barrado et al., 2014; Houghton et al., 2006; Becking et al., 2015). Several studies have focused on the detection of jellyfish blooms using imaging satellites. However, satellites are not an appropriate jellyfish

* Corresponding author.
E-mail address: m.rivas-casado@cranfield.ac.uk (M. Rivas Casado).

https://doi.org/10.1016/j.jag.2020.102279
Received 7 July 2020; Received in revised form 17 October 2020; Accepted 30 November 2020
Available online 21 January 2021
0303-2434/© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

monitoring platform due to the time taken to orbit and access data, their relatively low resolution, and the imaging challenges they face with respect to atmospheric water vapour. In the context of an early warning system, the temporal delay of access to satellite imagery renders it impractical to provide 6-8 h warning, although this may change shortly due to a newly expected satellite constellation (Capella Space, 2020). UAVs can provide frequent monitoring and swift deployment within a known area, leading to rapid access to data (Turner et al., 2016); characteristics lacking from both light-aircraft and satellite based systems. These properties are critical for an early warning detection system to function effectively. These qualities could provide the basic framework that could deliver enough time to act in response to bloom presence. UAVs have already been successful in imaging jellyfish blooms (Schaub et al., 2018) and are currently the best platform to provide images for an early warning detection system. This is due to a combination of ease of access to technology and it's deployment, low relative cost and very high resolution data. Another advantage is being able to fly below clouds and other atmospheric weather conditions that can prevent reliable satellite based remote sensing.

Once imagery has been gathered, the next consideration for an early warning detection system is how it approaches image analysis. The work by Schaub et al. (2018) at the University of British Columbia showcases clustering analysis methods that are novel with respect to jellyfish blooms, however there is a requirement for lengthy post-image processing. If we are to move towards an early detection system that can provide both short term and consistent warning of an impending jellyfish bloom, minimised post-image processing is essential. The natural demand of a task of this complexity is for the data analysis to be as automated as possible. This has been shown in other areas of study to be workable and a highly efficient approach (Ross et al., 2006; Boltze et al., 2019; Feng et al., 2019).

Neural networks have become recognised for their strong performance in automated image classification and feature recognition tasks, especially convolutional neural networks (CNNs) (Rawat and Wang, 2017). The increase in usage of deep learning models, including CNNs, is one of the fastest growing areas of machine learning and is frequently benchmarked by ImageNet database performance (Rawat and Wang, 2017). On top of the exceptionally high performance in image classification tasks (Rawat and Wang, 2017), CNNs have other key advantages. Their ability to generalise to data they have not yet been exposed to is one of their most important properties in the context of an early warning system (Rawat and Wang, 2017). They are also computationally efficient and can be updated after their initial training, are relatively lightweight, and are easy to deploy (Asif et al., 2018). These properties would be advantageous for an early warning system that aims to provide 6–8 h warning of an incoming jellyfish bloom. Having the ability to deploy the trained model on inexpensive electronic devices also keeps the potential of on-board UAV data processing a realistic future capability, as exhibited by Koo et al. (2017).

The most recent work by Koo et al. (2017) and Kim and Myung (2018) exhibit fantastic proofs of concept for using variants of CNNs as a detection method for aggregations of jellyfish, however their focus on using low resolution video sensors lends their work more towards controlled management strategies and away from large scale, higher area coverage early warning systems. If an early warning system could combine the benefits of higher altitude UAV remote sensing (circa 100 m) with the power of CNN image analysis, the reality of a fully comprehensive jellyfish bloom early warning system draws much closer. Once trained, a jellyfish bloom detecting CNN could be deployed on a UAV detection platform and be used for bloom detection in almost any desired environment. The work by Kim et al. (2015) provides the foundations of such a system and demonstrates the core system fundamentals through the detection of individual jellyfish. A CNN was selected due to the ease of deployment, computational efficiency, and the ability to generalise and predict unseen raw imagery with high accuracy; thus providing the framework of a reliable predictive jellyfish bloom detection system.

The aim of this study is to develop the detection capability of a reliable UAV-based jellyfish bloom early warning system. This will be achieved through the following objectives:

1. Develop a robust jellyfish bloom recognition tool through two key steps:
   (i) Production of CNN model architecture with adaption to jellyfish data.
   (ii) Testing of model performance (desired test accuracy >90%).
2. Maximise the transferability of the recognition tool through provision of the aforementioned environmental conditions, and other common non-environmental image artefacts.
3. Discuss the effectiveness and practicalities surrounding a jellyfish bloom detection system.

2. Materials and methods

2.1. Study area selection

The study area consists of two locations: the first is located at Craobh Haven (Argyll and Bute, UK (Fig. 1)) and the second at Pruth Bay (British Columbia, Canada (Fig. 1b)). Craobh Haven is a purpose-built holiday resort village with an easily accessible marina. The location was selected due to the presence of a recurrent *Aurelia aurita* bloom and a broad range of coastal features that are commonplace around coastal industries. Pruth Bay data were collected and provided by the University of British Columbia (Schaub et al., 2018), with the location selected due to the presence of a large *Aurelia* spp. bloom at the entrance to the bay (Schaub et al., 2018). This bloom likely consisted of *A. aurita* and *A. labiata* and potentially a further undescribed species *Aurelia* sp. (Lawley et al., 2020).

2.2. Data collection

One UAV flight was conducted in May 2018 at Craobh Haven under sunny conditions (Fig. 1). A fixed-wing Intel Sirius Pro UAV using a Sony Alpha 6300 RGB camera was used for the Craobh Haven flight. Flight mission planning was conducted using Intel's advanced planning software, MAVinci desktop (MAVinici, St. Leon-Rot, Germany); providing a pre-determined flight path that was optimised for the local area to maximise survey coverage. The flight was completed at an average altitude of 100 m producing a ground sampling distance (GSD) of 1.5 cm for the collected images. The camera used a 23.5 x 15.6 mm complimentary metal–oxide–semiconductor (CMOS) sensor with an ISO range of 100–25,600 and a maximum resolution of 24 MP. Imagery overlap was conducted laterally at 75% to aid the photogrammetric stitching of the site location orthoimage (Fig. 1). Images were visually assessed for quality assurance reasons. A small portion (<5%) were discarded if found to follow specific criteria: excessive blurring, excessive surface glare, and incorrect auto exposure leading to no identifiable content within the image.

15 Linear transects were completed during the Pruth Bay data collection (September 2016 (Schaub et al., 2018)). A rotary quad-blade DJI Phantom 3 Professional (Dà-Jiāng Innovations, Nanshan District, China) UAV using a DJI 12 MP RGB sensor was deployed to capture aerial images of the *Aurelia* spp. bloom. The native DJI GO flying application was used in order to maintain accuracy of intended transects. Images from the first three transects were unable to be used due to poor image quality from a fogged sensor lens. Flight transects that produced usable images had an average flight altitude of 112 m, and a GSD of 5.2 cm (Schaub et al., 2018). The camera used a 6.17 × 4.55 mm CMOS sensor with an ISO range of 100–1,600 and a maximum resolution of 12.4 MP.

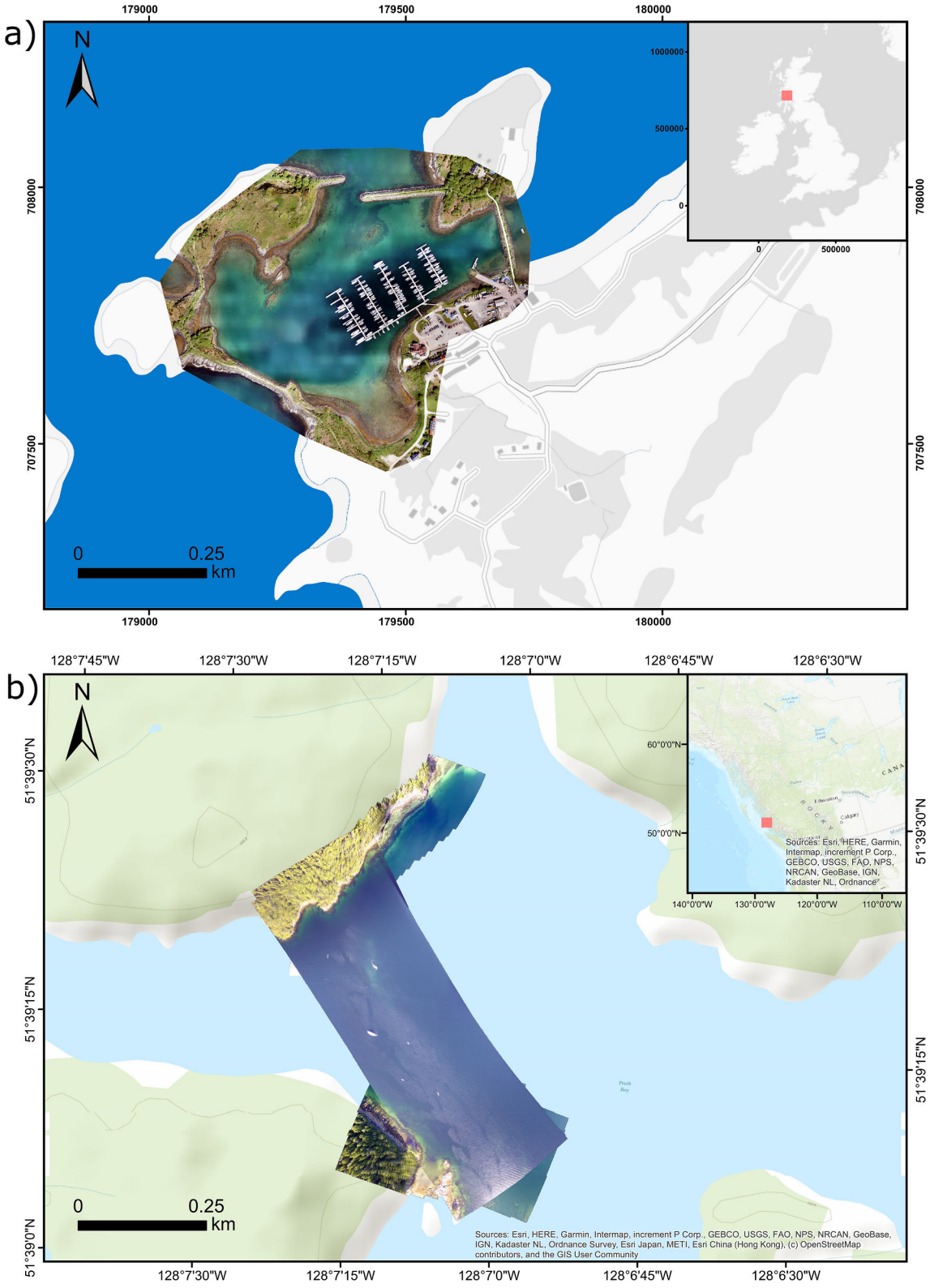


Fig. 1. Hybrid maps for both sampling locations with orthoimages of the areas embedded: (**a**) Craobh Haven, blue denotes unsurveyed marine area. Coordinates used from the British National Grid system. *Contains OS data © Crown copyright and database.* (**b**) Pruth Bay, axes use standard latitude and longitude. *Contains OpenStreetMap contributors data.*

2.3. Imagery pre-processing

The model was passed 4,000 500 × 500 px image chips belonging to four data groups (Table 1). Data were split across these four groups with a 50/50 model class split within each group (Fig. 2). Input imagery balance was maintained across each of these groups and their respective model classes, to minimise training bias as much as practically possible (Lee et al., 2016; Marmanis et al., 2016).

Table 1
The number of images that were included in their respective data collection source and training/test sets. The split of images across both model classes (Bloom present & No bloom present) was 50/50.

Source	Data group	Number of images
Craobh Haven	Training	2,400
	Test	800
Pruth Bay	Training	600
	Test	200

A total of 1,539 usable images were collected (Craobh Haven = 1,038, Pruth Bay = 501) and cropped into 4,000 500 × 500 px chips to train and test the model (Fig. 3). Image chips were extracted using the digital image cropping tool Greenshot (Braun et al., 2020) in combination with the image viewer IrfanView (Skiljan, 2019) to aid speed of processing the chips from the original raw images. 500 × 500 px chips were used to increase fine-grained pattern detection of blooms, and to bring input resolution into line with more recent, efficient models (Tan and Le, 2019; Huang et al., 2018). However, unlike these modern developments, the model input has a native resolution of 500 x 500 px and is not upscaled, eliminating any information degradation due to pixel interpolation (Pandey et al., 2018). Once cropped to 500 × 500 px, the chips were organised into each of the four respective groups (Table 1) and assessed for quality assurance reasons. Chips were manually sorted and labelled as "Bloom present" and "No bloom present", utilising the same techniques as French et al. (2018). The final 4,000 inputs were then entered into the model within their Training and Test datasets, and their respective model classes (50/50 class split).

2.4. Data analysis

2.4.1. Model training and architecture

A binary classification convolutional neural network (CNN) was trained with Python 3.6.9 (Van Rossum and Drake, 2009) using the high-level application programming interface (API) Keras-GPU 2.2.4 (Chollet, 2019), with tensorflow-GPU 1.13.1 as the backend (Abadi et al., 2016). Model training was completed with a Dell Precision Tower 5810 with an Intel Xeon E5-1650 v4 CPU, NVIDIA GeForce GTX 1080 GPU and 64 GB of RAM. Total model training time took 4 h and 48 min to complete.

Training aimed to reduce the model loss function value against training data as each step was processed. Each update to the model, per epoch, was assessed by the independent test dataset which the model was measured against. Model performance was indicated and measured through increases in accuracy of the model against the test dataset. Improvements were achieved through the optimiser minimising the model loss function through alterations of the weight vector values.

The 4,000 image chips were divided into a 3,000/1,000 (training/test) split, with this proportion (75/25%) remaining constant across both model classes: "Bloom present" and "No bloom present". Within the 3,000 training chips, 2,400 were sourced from the higher GSD imagery from Craobh Haven, and 600 from Pruth Bay (Table 1). Within the test data, the same location proportion was maintained with 800 sourced from Craobh Haven, and 200 from Pruth Bay. Test data were set aside to represent the same distribution as training data in order to minimise test bias.

Transfer learning was conducted using VGG-16 (Simonyan and Zisserman, 2015) due to accessible coding support whilst simultaneously maintaining extremely high performance and model simplicity. Having been originally trained on the ImageNet dataset (Russakovsky et al., 2015), VGG-16 has already been exposed to 1,000 image classes and 1.2 million images making it a perfect candidate for passing on it's feature recognition abilities. Transfer learning is crucial to achieve greater than 90% test accuracy for the comparatively small (4000 inputs) jellyfish dataset (Ng et al., 2015). VGG-16 is a CNN that was first formally presented in 2015 (Simonyan and Zisserman, 2015) after competing in, and winning, the 2014 ImageNet challenge. It is a deep network composed of a sequence of processing layers that take a 224 × 224 RGB image input. It is formed from 16 convolutional (conv2d) and five spatial pooling (maxpooling2d) layers followed by 3 fully connected layers with the last of these (a soft-max layer) performing the 1000 class ImageNet classification. Each hidden layer of VGG-16 uses ReLU for rectification of non-linearity (Simonyan and Zisserman, 2015). The VGG-16 model input layer was modified for jellyfish bloom detection to receive a larger 500 × 500 px RGB image input (Fig. 4) with default kernel and max pooling sizes used. A bespoke fully connected classifier was applied on top of the convolutional base consisting of the following layers: flatten, dense (with ReLU) and dense (with a sigmoid activation). VGG-16's final soft-max activation layer was replaced with a sigmoid activation layer as suggested by Francois Chollet for binary classification tasks (Chollet, 2017).

Bottleneck features (data only passed through all five convolutional blocks (Fig. 4)) were established, then exported. These bottleneck features were re-run with a bespoke fully connected jellyfish model (batch size = 50, epochs = 50, optimiser = RMSProp, loss = binary cross-entropy) applied on top to take full advantage of the feature recognition capabilities held within the bottleneck features. The feature extraction process was completed in this order for computational efficiency reasons. An adaptive optimiser (RMSProp) was used to achieve faster convergence during this initial model training (Dauphin et al., 2015; Roy et al., 2017; Qiu et al., 2018). The bespoke fully connected model, as shown within Fig. 4, will hereafter be referred to as the 'jellyfish top model'.

Hyperparameters for final training were selected using a held-out dataset from the same flights. These images were not included during the training or testing of the final model. This was conducted to maintain established deep learning practices (Chollet, 2017) and to ensure that the model had no access to information from the test dataset. Considered hyperparameters were as follows: data augmentation, number of epochs and learning rate.

Final model training was initiated using the pre-trained weights produced from initial training of the jellyfish top model. The weights used to initiate this training were pre-trained to allow transfer learning to occur and prevent major overfitting. The first four convolutional blocks were frozen from final training, and only included the last convolutional block and jellyfish top model (Fig. 4). The earlier blocks dealt with more general feature recognition. These earlier blocks were therefore not re-trained, with final training focusing on the more jellyfish-bloom-specific detection capabilities. The entropic capacity of the model demanded that caution was applied to avoid overfitting: total parameters = 44,206,401, total trainable parameters = 36,571,137. Data augmentation allowed batching of randomised variations of each of the 3,000 training chips (batch size 16). This turned the 3,000 chip training dataset into an artificially inflated 48,000 sized training set with integrated variations of chip shearing, zoom range and axis flipping. Data augmentation and dropout were used to help prevent overfitting and improve generalisation. Training data augmentation consisted of a combination of randomised variations of the following parameters: zoom range = 25%, shearing range = 25%, vertical flips, horizontal flips.

The loss optimiser selected was stochastic gradient decent (SGD) with adaptive optimisers avoided to ensure small updates to the model and not destroy pre-trained cognition (batch size = 16, epochs = 50, optimiser = SGD, learning rate = 1×10^{-4}, momentum = 0.9, loss = binary crossentropy). A very small learning rate was selected for the same rationale. Model performance was tracked using Keras' compile command and selecting the metric output as "accuracy" (Ketkar, 2017), resulting in an output of every epoch's training data "loss" and "accuracy" values. This process provided the data for analysis of the training process from start to finish across all epochs. Test loss and accuracy were also monitored in the same way as mentioned above for training, but

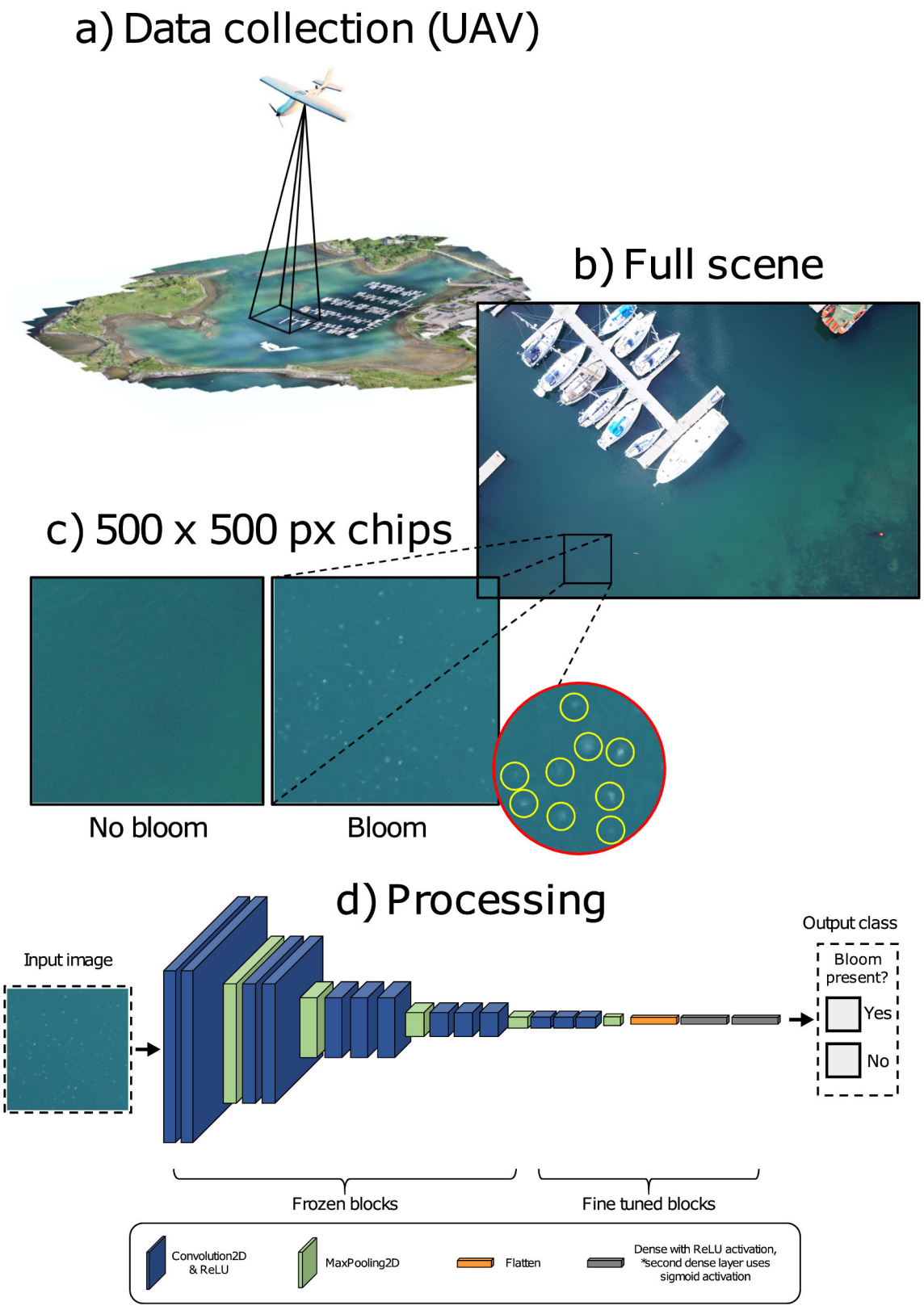


Fig. 2. Overall research flow chart summarising the methodological processes: (**a**) Aerial data collection using unmanned aerial vehicles; (**b**) Example of a full size raw image collected from the Craobh Haven sampling location; (**c**) Example of a 500 × 500 pixel chip from each model class; (**d**) Indicative summary of passing a chip to the model, the training layers used in the model, and prediction output options.

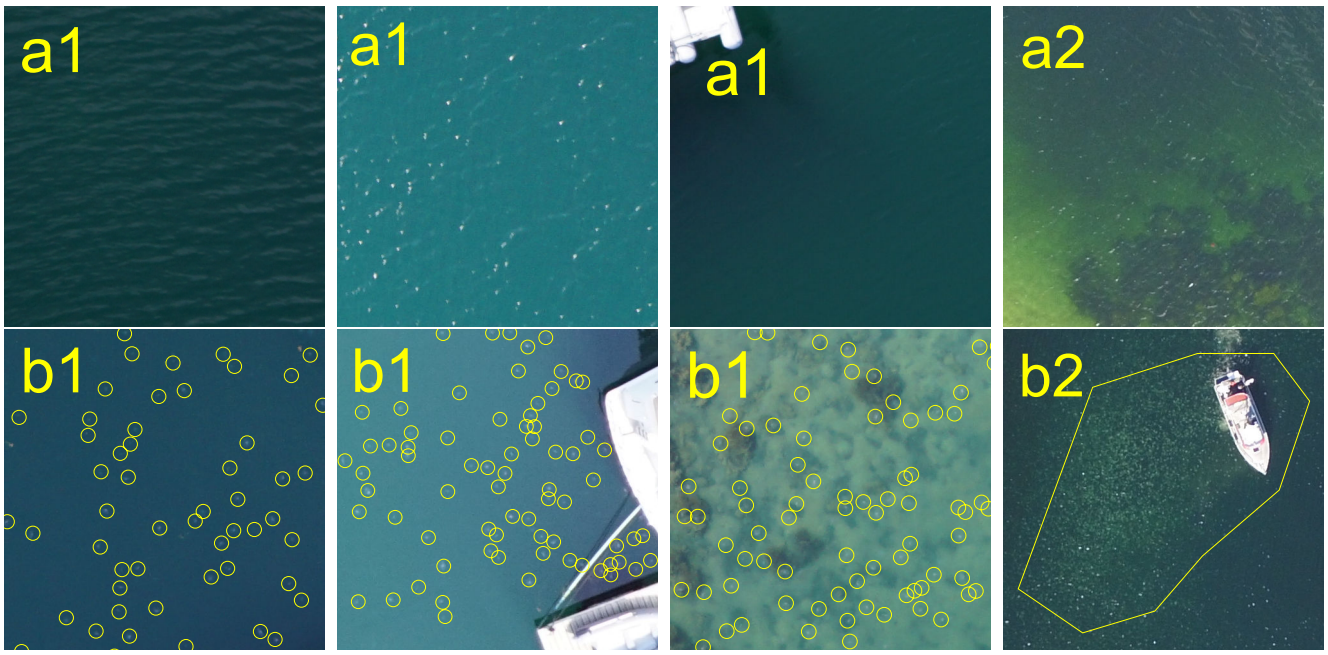


Fig. 3. Example chips (500 × 500 pixels) provided to the model: (**a1**) = No bloom present, Craobh Haven; (**a2**) = No bloom present, Pruth Bay; (**b1**) = Bloom present, Craobh Haven; (**b2**) = Bloom present, Pruth Bay. Jellyfish are manually highlighted by yellow boundaries. Ground sampling distance: Craobh Haven = 1.5 cm, Pruth Bay = 5.2 cm (mean). White artefacts seen in the second and fourth images are glare from sunlight. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

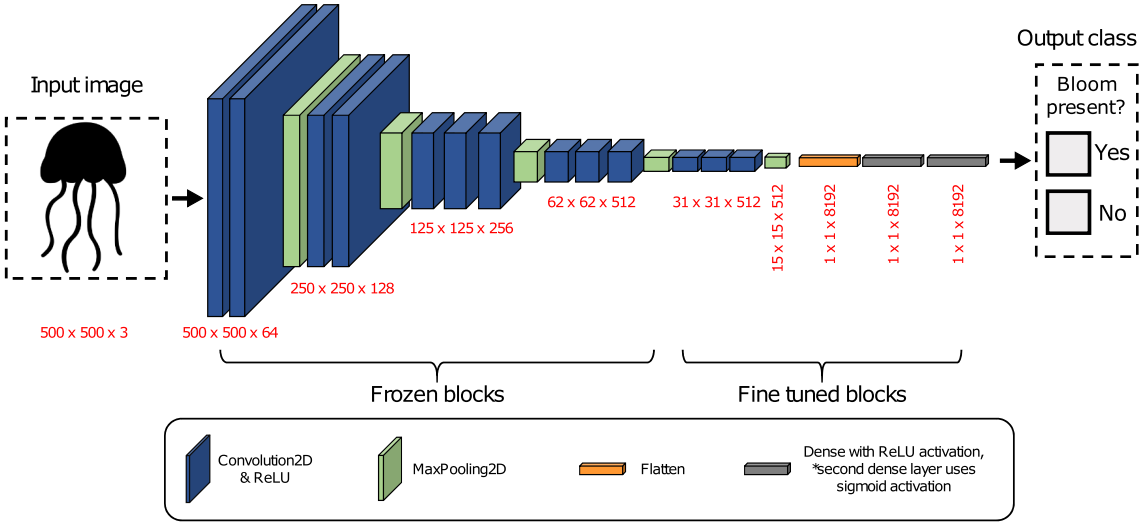


Fig. 4. Model architecture and layers used. An indicative representation of a 500 × 500 pixel chip's journey from input to prediction. Layer sizes stated in red. ReLU = rectified linear unit.

with increases in test accuracy used to trigger model checkpoint saving to output the full model for that specific epoch. Performance graphs were produced on R 3.4.3 (R Core Team, 2017) with the package "ggplot2" (Wickham, 2009).

3. Results

3.1. Model performance

Performance metrics from the initial passing of data (not final training) through VGG-16 and the jellyfish top model were as follows: Epoch 1/50, test acc = 50%; Epoch 50/50, test acc = 81.5%. Final training epochs consisted of 187 steps of batches (batch size = 16) allowing a full pass of all available training data (1,500 images for each model class). As the model is updated to reduce loss against training data, test loss can be seen to closely follow indicating an absence of major overfitting (Fig. 5a).

An assessment of model accuracy (Fig. 5b) supports the findings from the previous assessment based on model loss. Again, the model is not suffering from overfitting; test accuracy is increasing across epochs, alongside training accuracy, and not remaining unimproved whilst training accuracy increases. Variation in test accuracy during convergence is to be expected whilst the optimisation function is hunting for the global minima.

Model performance peaked at epoch 39 with a test accuracy of 97.46%, with the lowest performing epoch having an accuracy of 73.07% at epoch 17 (Table 2). This performance trough coincides with a dramatic increase in model loss, which would be expected, and was

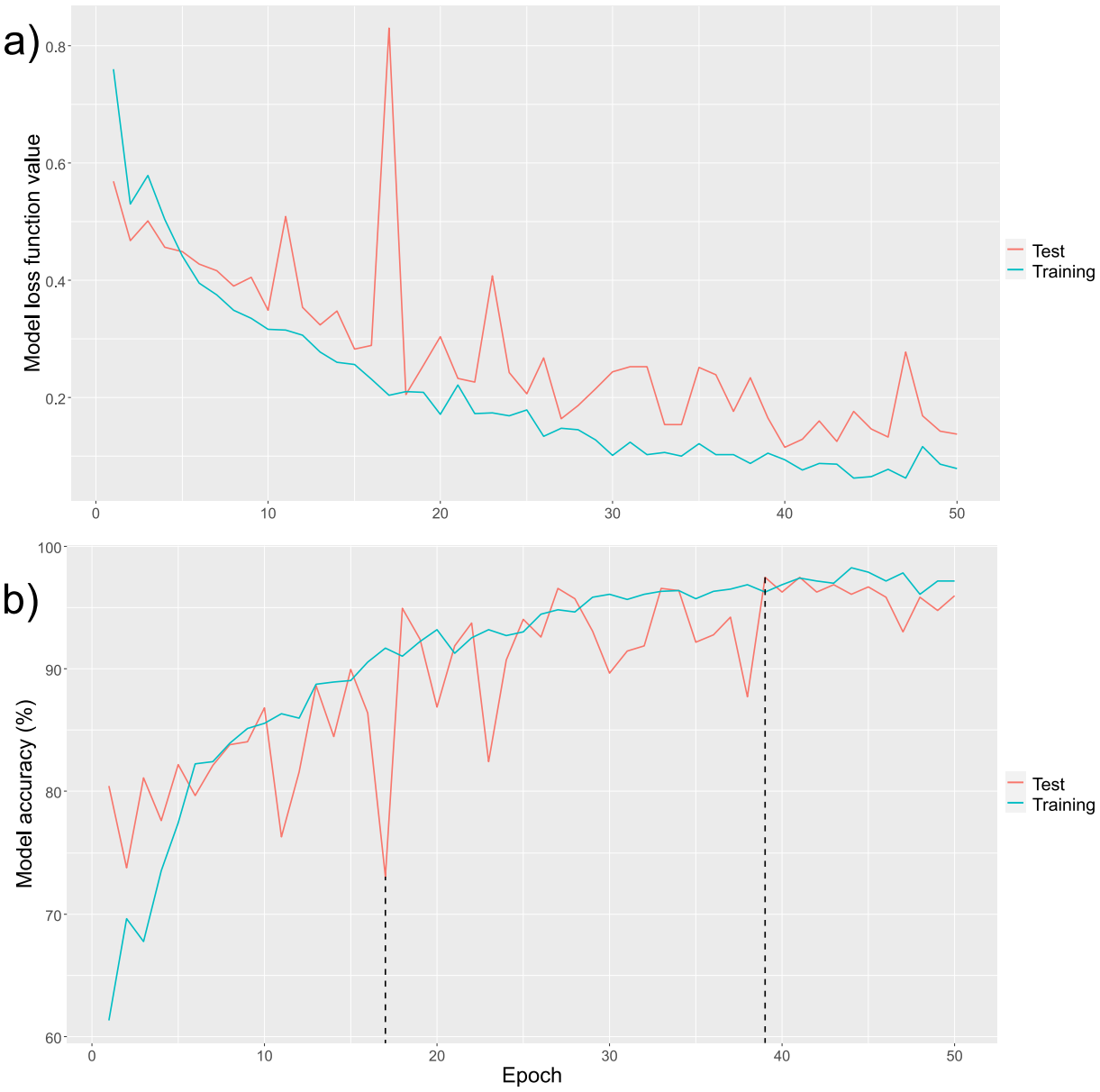


Fig. 5. Model metrics for both training and test datasets: (**a**) Reductions in model loss; (**b**) Increases in model accuracy. Lowest and highest performing epochs (17 & 39) indicated with dashed lines.

likely corrected due to the high momentum hyperparameter (0.9). Model performance appears to level off from epoch 39 onwards with minimal decline. Improved models were automatically saved and exported to ensure the best performing models were accessible post-training. Desired test accuracy was achieved by epoch 18 with only a further two epochs improving model performance. The highest performing epoch (39) has a training loss that is marginally lower than test loss, whilst it also has a training accuracy marginally less than test accuracy. It is logical that training loss is less than test loss, as the model optimiser attempts to reduce training loss, not test loss (Fig. 5a). Likely reasons for the marginally elevated test accuracy are that both dropout regularisation and data augmentation take place during model training. Both of which do not take place during model testing.

The performance of epoch 39 is suggestive of a model that is a good fit and neither overfitting (high training accuracy, low test accuracy (Cogswell et al., 2016)) or underfitting (low training accuracy, low test accuracy (Ali et al., 2019)). Both training and test accuracy appear to converge appropriately across all 50 epochs. This suggests that the number of epochs was appropriate with regards to model hyperparameters (Fig. 5b). The model prediction script is designed to express a worded statement of bloom presence or absence, but also provides a likelihood value of which class that input image belongs to (Fig. 6).

Checking visual predictions made by the model on the test dataset allowed a qualitative assessment of classification performance. Notice how despite the presence of extensive unwanted sunshine glare in the fourth column (Fig. 6), the model successfully predicts the presence/absence of a jellyfish bloom. Due to the similarity in appearance of jellyfish to sunshine glare, this result is a strong indicator of model robustness to image artefacts from environmental conditions. Other commonly found coastal features such as decking and boats (Fig. 6, final column) does not interfere with prediction performance. With peak classification test performance of 97.46% (Table 2), this is not unexpected but is nonetheless valuable to visually confirm the model's classification performance. The presence of buoys (Fig. 6, first column) was found to not negatively affect classification performance. This is of particular interest as marine buoys can be located in various coastal and offshore locations with the potential to prevent accurate bloom

Table 2
Final training model performance for epochs that improved test accuracy (total epochs = 50). Epochs that achieved desired model performance are highlighted in bold, highest performing epoch in red. Bottom three rows (grey) display subsequent epochs that did not improve model test accuracy.

Epoch	Training loss	Test loss	Training accuracy	Test accuracy
1	0.7602	0.5688	0.6133	0.8044
3	0.579	0.5013	0.6778	0.811
5	0.4418	0.449	0.7744	0.8222
8	0.3493	0.3896	0.8392	0.8384
9	0.3345	0.4048	0.8516	0.8404
10	0.3163	0.3491	0.8553	0.8679
13	0.2777	0.3238	0.8874	0.8862
15	0.2565	0.2827	0.8904	0.8994
18	**0.2104**	**0.205**	**0.9101**	**0.9492**
27	**0.148**	**0.1633**	**0.9479**	**0.9654**
39	**0.1053**	**0.1655**	**0.9626**	**0.9746**
40	0.0937	0.115	0.9686	0.9624
45	0.0651	0.1464	0.9786	0.9665
50	0.0785	0.1371	0.9716	0.9593

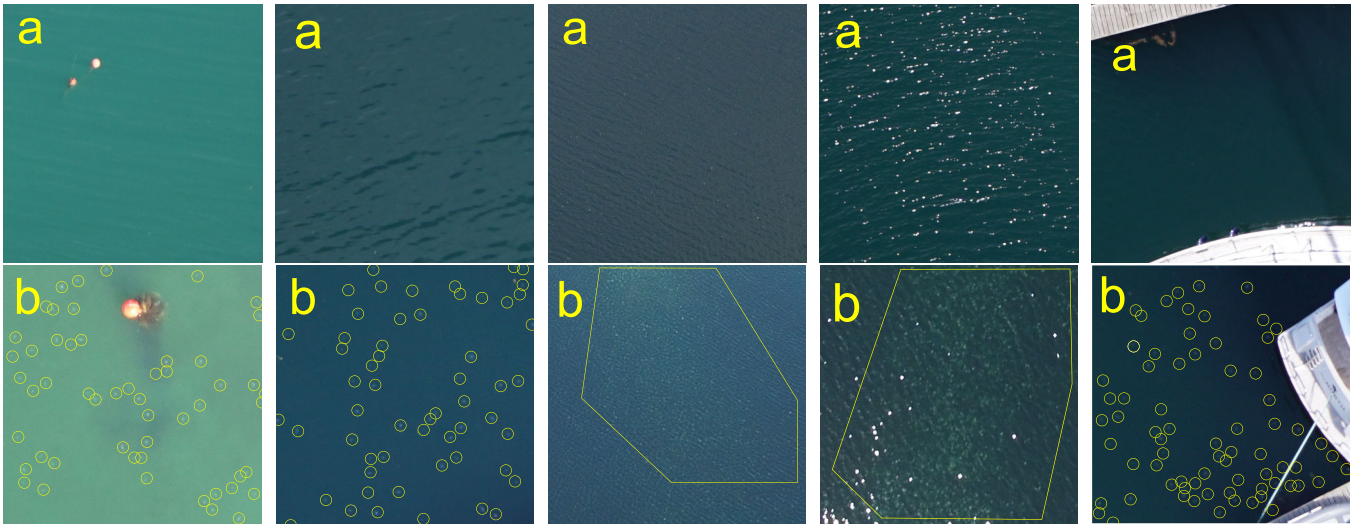


Fig. 6. Examples of classified images (500 × 500 pixels) after model prediction: (**a**) Prediction = No bloom present; (**b**) prediction = Bloom present. Jellyfish are manually highlighted by yellow boundaries. Ground sampling distance = various (1.5 cm or 5.2 cm). White artefacts seen in the fourth column of images are glare from sunlight. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

classification.

4. Discussion

The main restriction on data availability for model training was due to the nature of jellyfish blooms themselves. Their highly unpredictable occurrence, ironically the main driver for developing an early warning system, makes them difficult to locate. Despite this, enough images were captured (1,502). This was enough to process a meaningful amount of data that could be used to train the model (4,000 chips and 48,000 after augmentation); by far the most amount of data collected for remote sensing of jellyfish blooms and building on the work by Kim et al. (2016) and Schaub et al. (2018). However, the risk of overfitting had to be carefully avoided due to the relatively low quantity of data in machine learning terms (Mishkin et al., 2017). Batch size was limited to 16 due to RAM limitations of the computing system. The large chip size of 500 × 500 px made the process far more computationally intensive compared to more commonly used smaller chip sizes (Rawat and Wang, 2017;

Mishkin et al., 2017). The combination of augmentation with dropout, where the network is forced to train to more robust image features by reducing complex neuron co-adaptions (Krizhevsky et al., 2017), was highly successful. Despite being computationally intensive, this process was essential to prevent overfitting to the training data, and to improve the model's ability to generalise (Krizhevsky et al., 2017). Inclusion of the Pruth Bay data prevented training the model to a very specific UAV flight platform, imaging sensor, and location. Inclusion allowed the model to be trained to both Canadian and British coastal water conditions, multiple ground sampling distances, two different sensor resolutions, and quite conceivably different *Aurelia* species as well (Lawley et al., 2020). It is common for remote sensing neural networks to split data into two main datasets (Training/Testing) (Li et al., 2016; Hu et al., 2015; Fu et al., 2017; Ammour et al., 2017) with a further held-out dataset for hyperparameter selection (Chollet, 2017). This process has been continued and referred to as "Training" and "Test", whilst the hyperparameter selection process equates to "Validation" to maintain independence of the Test dataset and to prevent "information leaks" (Chollet, 2017).

Jellyfish are difficult to detect (Koo et al., 2017), but compared to most machine learning problems they have very low intra-class variation (Wei et al., 2015). Maintaining a balanced dataset across all four data groups (Table 1) was a key consideration during processing of training chips. Poor performing models can often be traced to biases across training classes (Kim et al., 2019). This was given particular attention when providing the model examples of various environmental conditions. Both model classes were checked to have near equal numbers of wave topping, solar glare, and water colouration examples. If this careful balance was not considered, the model would have likely begun to incorrectly associate particular environmental conditions with one of the model classes more than the other – especially solar glare which can appear very similar to jellyfish within RGB imagery. This can be noted in (Fig. 6) where classification performance was not impacted by presence of these aforementioned environmental conditions. The importance of keeping this balance across model classes is well documented and the model results support this (Lee et al., 2016; Tsinalis et al., 2016; Buda et al., 2018; Hensman and Masko, 2015). The proportion of non-environmental related artefacts was also preserved: whole boats, parts of boat, pontoons, rocks, buoys, and coastal vegetation were considered during chip processing (Fig. 7). By default, the model has inherent robustness to varied levels of bloom density. This is due to the natural variation of jellyfish present in the "Bloom present" class chips provided during training. This is the same for water colouration but across both model classes, as seen in Figs. 3, 6 and 7.

The presence and inclusion of the non-environmental image artefacts (Fig. 7) undoubtedly constructively contributed to the model's high test performance. Despite only showing a small sample of the non-environmental artefacts included during training, it can be qualitatively noted how these images would help improve model performance. It is particularly clear when visually assessing the first two columns (Fig. 7); examples of inputs where jellyfish blooms would impossibly exist. By introducing this source of intentional bias in the training data, the model can very effectively ascertain where blooms could never exist. The power of this technique emphasises the importance of high quality training data and how damaging errors could be if data are incorrectly labelled. The only source of this intentional bias in the data were a minority of chips within the "No bloom present" model class. A small number of chips contained information in which jellyfish blooms could never exist, such as chips exclusively containing terrestrial shoreline and coastal rocks (Fig. 7). We anecdotally attribute a significant portion of the model's high performance to the balance within the data. Without this balance, however powerful the model architecture was, it would have struggled to perform and effectively been an early performance bottleneck (Lee et al., 2016; Menardi and Torelli, 2014; Vluymans, 2019).

The initial passing of data through VGG-16 with the bespoke jellyfish top model was successful in raising test accuracy to 81.5% after 50 epochs. Unlike the final training stage, an adaptive optimiser (RMSProp) was used in order to accelerate training (Dauphin et al., 2015; Roy et al., 2017; Qiu et al., 2018) and encourage large magnitude updates to the model. Prior to final training, it was necessary to initially pass the data through the model and establish weights for the jellyfish top model. Final training would not have been effective if conducted on a fully connected model with randomly initialised weights; large updates to the model would un-do previously learned features within the convolutional base (Xiao et al., 2014). This is a critical step in the application of



Fig. 7. Examples of chips (500 × 500 pixels) showing non-environmental artefacts that were passed to the model during training to improve model robustness. Jellyfish are manually highlighted by yellow boundaries. Ground sampling distance = various (1.5 cm or 5.2 cm).

transfer learning, particularly when utilising large base models. The smaller the training dataset, the more emphasised this phenomenon can be. This was not an issue for this jellyfish bloom detection model, but nevertheless is worth highlighting.

To attain global success, the system must be robust enough to deal with real world environmental conditions (Carrio et al., 2017). It is imperative that a robust detection system considers the following: ocean wave topping, solar glare, differences in water colouration, miscellaneous surface objects such as boats and other common coastal features, and differences in imaging sensor resolution and ground sampling distance (GSD). If these considerations are not addressed, the reliability of the system would be questionable. Kim et al. (2015) showed that producing a model with imagery from a small and localised area can produce "reasonable" results within a controlled environment, however it would struggle to perform when exposed to varied environmental conditions. A jellyfish bloom detecting CNN could be updated with further species down the line, and accommodate more regional genera demands (Kim et al., 2016). It is also vital that any developed detection system is refined and deployed alongside an optimum survey strategy (Freeman et al., 2019). If not, despite having a reliable detection capability, the overall system performance could bottleneck prematurely. It is pertinent that an optimum survey strategy utilises the knowledge gained from a bloom presence/absence detection capability in an appropriate manner. The survey strategy and detection capability could be viewed as a theoretical symbiotic relationship within a future overall system.

The final training process was highly successful at increasing model test accuracy through iterative reductions of training loss. Despite iterations after epoch 39 not improving the model, performance by that stage in the training process was well above the objective of 90% test accuracy (Table 2). The use of a non-adaptive optimiser, in the form of stochastic gradient descent with momentum, was key to providing steady and progressively improved model updates. Combined with a very slow learning rate, both model performance and the ability to generalise have been found to improve in comparison to adaptive optimisers (Keskar and Socher, 2017; Wilson et al., 2017). The test performance of the model supports these findings. The model converged from around epoch 35 (Fig. 5b), with performance remaining relatively consistent until the final epoch. If the intention was to improve model performance even further, an investigation into using weight decay regularisation and more aggressive dropout and augmentation would be the first recommendation. However, in realistic terms – attempts to improve on the current model performance of 97.46% would not be cost-effective. Variations of less than 1% in test accuracy are more likely due to stochastic effects with respect to the size and composition of the test dataset. Increasing the size of the test dataset would be the most sensible starting point before attempting any changes to the model.

The final model performance of 97.46% was above expectations, and highlights the strength of using transfer learning to enhance performance (Shin et al., 2016). Through the use of very high resolution 500 × 500 px chips, jellyfish blooms were still detectable even when at lower densities (Fig. 6). This increased fine-grained detection came at a drastic computing cost during training. It is for this reason that the majority of CNN models use much lower resolution input chips but we feel was a decision very much worth taking. Despite the model being trained on 500 × 500 px chips, this does not mean the model requires images of this exact size to process for prediction. The model actively upscales smaller images by nearest neighbour interpolation and will crop around the centroid pixel if larger (Chollet, 2015). This allows images of any size to be passed through the model for prediction classification if desired. The ability to pass any aerial image through the model is a great advantage and increases the flexibility of the model's potential deployment.

With a ground sampling distance of 1.5 cm for the Craobh Haven data, and 5.2 cm for Pruth Bay, a 500 × 500 px chip will have a ground coverage of 7.5 m^2 and 26 m^2 respectively. Both have a practically useful coverage and combined with an appropriately efficient survey technique (Yanmaz, 2012), have great potential to be the framework for the first fully functional jellyfish bloom detection system. It would always be advantageous to use a sensor that increases resolution, however the model results categorically show that this is not a pre-requisite for successful bloom detection. A point of diminishing returns will occur with a trade-off occurring between increased cost and weight of sensor, and final model performance. The performance of our bloom detection model shows that resolution of the sensor used (Sony Alpha 6300 RGB) was certainly not limiting bloom detection performance. The highest performing model is 315.6 MB in size which by recent micro-computing standards is easily manageable to store and process, such as the intel NUC i7 small form factor PC (Koo et al., 2017). By being able to cope with a broad range of both environmental and non-environmental imagery artefacts, the robustness of the model should allow practical deployment to real world survey conditions. The final stage of moving towards a fully functional early warning detection system would be to incorporate the model into a semi-automated UAV deployment system. However, considerations should be given around nighttime deployment in such a system. Currently, detection would not be possible during darkness due to the use of a passive sensor (RGB).

Many great modern technological innovations garner inspiration from nature. Comparisons by Suarez and Murphy (2011) exhibit how varied a survey approach can be when drawing on animal foraging strategies for inspiration. Within the context of a nuclear power station that by nature is immobile, the survey strategy can, and should, be optimised for that scenario (Pitre et al., 2012). The most critical consideration should be the adaption of the knowledge provided by the bloom detection model output combined with the survey style. With the provision of a binary output from the model, the survey strategy should react and adapt to the most current information provided. For example, if a bloom is detected using a standard parallel sweep survey route, should the survey strategy continue or adapt and change to a circular radial search pattern? Poor survey strategy can dramatically decrease the overall performance of an early warning system and could negate a high performing detection component. There is currently no available literature on UAV specific survey optimisation over aquatic environments; this is identified as a key area of future work. UAVs do not have the same functionality and coverage capability of light-aircraft, boats, and satellites, for which survey strategy is well researched (Hill, 2003; Rianto et al., 1999; Razi and Karatas, 2016). It is imperative for an early warning detection system to ensure a high performing detection capability is combined with an efficient and optimised survey plan (Xu et al., 2011).

This work delivers the image processing ability for detecting *Aurelia* spp. jellyfish blooms in the marine environment. With regard to nuclear power stations and desalination plants, prior warning of an incoming bloom can allow controlled reduction of the water intake volume and aim to prevent more permanent disruption. Fish farms, for example, would be given time to install temporary guarding systems for their fish pens - similar to how stinger nets are deployed at beaches (Lucas et al., 2014). It is hoped that by using a model robust enough to cope with variable environmental conditions and non-biological coastal objects, the model constructively contributes to addressing the global jellyfish bloom marine ingress problem (Purcell et al., 2007). Although *Aurelia* spp. blooms are one of the most widespread and disruptive genera of all jellyfish (Purcell et al., 2007; Hamner and Dawson, 2009), they are not the only group of species that cause huge disruption to coastal industries (Montgomery et al., 2016; Purcell et al., 2013; Kim et al., 2012; Zhang et al., 2012). Despite the model not being trained on any other genera of jellyfish, it would likely to detect them with practically useful levels of performance (Yosinski et al., 2014; Wang et al., 2018; Chi et al., 2017; Reyes et al., 2015). However, it would be prudent to apply caution when deploying the model on other problem jellyfish species. Unlike other forms of networks, CNNs can be easily re-trained (Zhou et al., 2017); future integration of further jellyfish species classes into the model is advised. This would be especially beneficial for migration pattern studies, species habitat monitoring and ground truthing of bloom

prediction research.

5. Conclusion

This paper exhibits the world's first UAV based high resolution, multi-sensor jellyfish bloom detection capability including integrated robustness from two oceans to tackle real world detection challenges. Final model performance (97.46% test accuracy) was well above the desired performance of 90%. Potential overfitting of the model appeared to be handled exceptionally well with data augmentation and dropout regularisation, supporting the recent work of large models using relatively small datasets. Research into optimal survey routes for UAVs over aquatic environments is a highlighted gap in knowledge and LiDAR sensors may also be a promising avenue for investigation into nighttime detection of jellyfish blooms. The addition of further jellyfish species for model training is recommended, as well as increasing the amount of training data with images from new locations, sensors and environmental conditions. With global changes occurring at a rate faster than has ever been observed, the pursuit of novel machine learning detection techniques provide hope for addressing both current and future challenges. The mitigation of issues relating to global change will become a possibility, with improvements in efficiency and revenue - irrespective of coastal industry - a likely result.

Author Contributions

Ben Mcilwaine: Conceptualisation; Methodology; Software; Validation; Data Analysis; Investigation; Data curation; Writing - Original draft preparation; Writing - Review & editing; Visualisation.

Monica Rivas-Casado: Conceptualisation; Data curation; Writing - Review & editing; Supervision; Project Administration; Funding Acquisition.

Funding

This research was funded by the Engineering and Physical Sciences Research Council (EPSRC), The Smith Institute and EDF Energy under EPSRC Industrial Case Studentship Voucher 586 Number 16000001.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Our greatest thanks go to Jessica Schaub and her colleagues at the University of British Columbia for providing the raw images from their Pruth Bay dataset, their collaboration is most appreciated. We would also like to thank the Engineering and Physical Sciences Research Council (EPSRC), The Smith Institute and EDF Energy for funding this project. We would like to thank the reviewers for their helpful comments and constructive criticism. The manuscript became stronger thanks to their detailed contribution. The underlying data are confidential and cannot be shared due to a permenant embargo.

References

Zhou, Y., Song, S., Cheung, N.M., 2017. On classification of distorted images with deep convolutional neural networks. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Institute of Electrical and Electronics Engineers Inc. IEEE, pp. 1213–1217.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, pp. 265–283. https://tensorflow.org.

Ali, L., Rahman, A., Khan, A., Zhou, M., Javeed, A., Khan, J.A., 2019. An Automated Diagnostic System for Heart Disease Prediction Based on $\chi 2$ Statistical Model and Optimally Configured Deep Neural Network. IEEE Access 7, 34938–34945 http://www.ieee.org/publications_standards/publications/rights/index.html.

Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., Zuair, M., 2017. Deep Learning Approach for Car Detection in UAV Imagery. Remote Sens. 9 (4), 312 http://www.mdpi.com/2072-4292/9/4/312.

Asif, U., Tang, J., Harrer, S., 2018. GraspNet: An efficient convolutional neural network for real-time grasp detection for low-powered devices. In: IJCAI International Joint Conference on Artificial Intelligence. Vol. 2018-July. pp. 4875–4882.

Barrado, C., Fuentes, J.A., Salamí, E., Royo, P., Olariaga, A.D., López, J., Fuentes, V.L., Gili, J.M., Pastor, E., 2014. Jellyfish monitoring on coastlines using remote piloted aircraft. IOP Conf. Ser.: Earth Environ. Sci. 17 (1), 012195 http://stacks.iop.org/1755-1315/17/i=1/a=012195?key=crossref.3bec8734109c44f5d3c2b7da2b8b0f15.

Becking, L.E., de Leeuw, C., Vogler, C., 2015. Newly discovered jellyfish lakes in Misool, Raja Ampat, Papua, Indonesia. Marine Biodiver. 45 (4), 597–598 http://link.springer.com/10.1007/s12526-014-0268-6.

Boltze, J., Ferrara, F., Hainsworth, A.H., Bridges, L.R., Zille, M., Lobsien, D., Barthel, H., McLeod, D.D., Gräßer, F., Pietsch, S., Schatzl, A.-K., Dreyer, A.Y., Nitzsche, B., 12 2019. Lesional and perilesional tissue characterization by automated image processing in a novel gyrencephalic animal model of peracute intracerebral hemorrhage. J. Cerebral Blood Flow Metabol.: Off. J. Int. Soc. Cerebral Blood Flow Metabol., vol. 39 (12), 2521–2535. http://www.ncbi.nlm.nih.gov/pubmed/30239258 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6893983.

Braun, T., Klingen, J., Krom, R., 2020. Greenshot. URL https://getgreenshot.org/.

Buda, M., Maki, A., Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks 106, 249–259.

Capella Space, 2020. Home2 - Capella Space. URL https://www.capellaspace.com/.

Carrio, A., Sampedro Pérez, C., Rodríguez Ramos, A., Campoy, P., 2017. A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles. J. Sensors 2017, 1–13.

Chi, J., Walia, E., Babyn, P., Wang, J., Groot, G., Eramian, M., 2017. Thyroid Nodule Classification in Ultrasound Images by Fine-Tuning Deep Convolutional Neural Network. J. Digit. Imaging 30 (4), 477–486.

Chollet, F., 2015. Image Preprocessing - Keras Documentation. URL https://keras.io/preprocessing/image/.

Chollet, F., 2017. Deep Learning with Python. No. 1. Manning Publications Co. http://www.ncbi.nlm.nih.gov/pubmed/20608803.

Chollet, F., 2019. Keras-gpu. URL https://github.com/fchollet/keras.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D., 2016. Reducing overfitting in deep networks by decorrelating representations. In: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. International Conference on Learning Representations, ICLR.

Daskalov, G.M., Grishin, A.N., Rodionov, S., Mihneva, V., 2007. Trophic cascades triggered by overfishing reveal possible mechanisms of ecosystem regime shifts. Proc. Natl. Acad. Sci. USA 104 (25), 10518–10523.

Dauphin, Y.N., De Vries, H., Bengio, Y., 2015. Equilibrated adaptive learning rates for non-convex optimization. In: Advances in Neural Information Processing Systems. Vol. 2015-Janua. pp. 1504–1512.

Feng, C., Yu, D., Liang, Y., Guo, D., Wang, Q., Cui, X., 2019. Assessment of influence of image processing on fully automatic UAV photogrammetry. Int. Arch. Photogram. https://doi.org/10.5194/isprs-archives-XLII-2-W13-269-2019.

Flynn, K., Chapra, S., 2014. Remote Sensing of Submerged Aquatic Vegetation in a Shallow Non-Turbid River Using an Unmanned Aerial Vehicle. Remote Sens. 6 (12), 12815–12836 http://www.mdpi.com/2072-4292/6/12/12815.

Freeman, M., Vernon, C., Berrett, B., Hastings, N., Derricott, J., Pace, J., Horne, B., Hammond, J., Janson, J., Chiabrando, F., Hedengren, J., Franke, K., 2019. Sequential earthquake damage assessment incorporating optimized suav remote sensing at pescara del tronto. Geosciences (Switzerland) 9 (8), 332 https://www.mdpi.com/2076-3263/9/8/332.

French, G., Mackiewicz, M., Fisher, M., Challiss, M., Knight, P., Robinson, B., Bloomfield, A., 2 2018. JellyMonitor: Automated detection of jellyfish in sonar images using neural networks. In: International Conference on Signal Processing Proceedings, ICSP. Vol. 2018-August. Institute of Electrical and Electronics Engineers Inc., pp. 406–412. https://ieeexplore.ieee.org/document/8652268/.

Fu, G., Liu, C., Zhou, R., Sun, T., Zhang, Q., 2017. Classification for High Resolution Remote Sensing Imagery Using a Fully Convolutional Network. Remote Sens. 9 (5), 498 http://www.mdpi.com/2072-4292/9/5/498.

Graham, W.M., Pagès, F., Hamner, W.M., 2001. A physical context for gelatinous zooplankton aggregations: A review. Hydrobiologia. 451, 199–212 http://link.springer.com/10.1023/A:1011876004427.

Hamner, W., Dawson, M., 2009. A review and synthesis on the systematics and evolution of jellyfish blooms: Advantageous aggregations and adaptive assemblages. Hydrobiologia 616 (1), 161–191.

Hensman, P., Masko, D., 2015. The Impact of Imbalanced Training Data for Convolutional Neural Networks. Ph.D. thesis. KTH Royal Institute of Technology.

Hill, R.R., 2003. Agent models II: search theory, agent-based simulation, and U-boats in the Bay of Biscay. Ballistic Impact Modeling View project Simulation Methodology View project. In: Proceedings of the 35th Winter Simulation Conference: Driving Innovation. DBLB, New Orleans, Louisiana, USA. URL https://www.researchgate.net/publication/221528594.

Houghton, J., Doyle, T., Davenport, J., Hays, G., 2006. Developing a simple, rapid method for identifying and monitoring jellyfish aggregations from the air. Mar. Ecol. Prog. Ser. 314, 159–170 http://www.int-res.com/abstracts/meps/v314/p159-170/.

Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. Remote Sens. 7 (11), 14680–14707 http://www.mdpi.com/2072-4292/7/11/14680.

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, M.X., Chen, D., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., Chen, Z., 2018. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. ArXiv. URL http://arxiv.org/abs/1811.06965.

Keskar, N.S., Socher, R., 2017. Improving Generalization Performance by Switching from Adam to SGD. ArXiv. URL http://arxiv.org/abs/1712.07628.

Ketkar, N., 2017. Introduction to Keras. In: Deep Learning with Python, 1st Edition. Apress, Berkeley, CA, pp. 97–111. Ch. Introduction.

Kim, B., Kim, H., Kim, K., Kim, S., Kim, J., 2019. Learning not to learn: Training deep neural networks with biased data. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2019-June. pp. 9004–9012.

Kim, D.H., Seo, J.N., Yoon, W.D., Suh, Y.S., 2012. Estimating the economic damage caused by jellyfish to fisheries in Korea. Fish. Sci. 78 (5), 1147–1152 http://link.springer.com/10.1007/s12562-012-0533-1.

Kim, H., Kim, D., Jung, Sungwook, Koo, Jungmo, Shin, J.-U., Myung, H., 2015. Development of a UAV-type jellyfish monitoring system using deep learning. In: 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, pp. 495–497.

Kim, H., Koo, J., Kim, D., Jung, S., Shin, J.U., Lee, S., Myung, H., 2016. Image-based monitoring of Jellyfish using deep learning architecture. IEEE Sens. J. 16 (8), 2215–2216.

Kim, K., Myung, H., 2018. Autoencoder-Combined Generative Adversarial Networks for Synthetic Image Data Generation and Detection of Jellyfish Swarm. IEEE Access 6, 54207–54214.

Koo, J., Jung, S., Myung, H., 2017. A jellyfish distribution management system using an unmanned aerial vehicle and unmanned surface vehicles. In: 2017 IEEE Underwater Technology (UT). IEEE, pp. 1–5. http://ieeexplore.ieee.org/document/7890324/.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90 http://dl.acm.org/citation.cfm?doid=3098997.3065386.

Lawley, J.W., Gamero-Mora, E., Maronna, M.M., Chiaverano, L.M., Stampar, S.N., Collins, A.G., Morandini, A.C., 2020. Morphological overlap hinders diagnosability: recognizing cryptic diversity in Aurelia (Cnidaria, Scyphozoa) - Unpublished results.

Lee, H., Park, M., Kim, J., 8 2016. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: Proceedings - International Conference on Image Processing, ICIP. Vol. 2016-August. IEEE Computer Society, pp. 3713–3717. http://ieeexplore.ieee.org/document/7533053/.

Li, W., Fu, H., Yu, L., Cracknell, A., 2016. Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images. Remote Sensing 9 (1), 22 http://www.mdpi.com/2072-4292/9/1/22.

Lucas, C.H., 2001. Reproduction and life history strategies of the common jellyfish, Aurelia aurita, in relation to its ambient environment. Hydrobiologia 451 (1/3), 229–246 http://link.springer.com/10.1023/A:1011836326717.

Lucas, C.H., Gelcich, S., Uye, S.I., 2014. Living with jellyfish: Management and adaptation strategies. In: Jellyfish Blooms. Vol. 9789400770. Springer, pp. 129–150. URL https://www.researchgate.net/publication/258356918.

Marmanis, D., Datcu, M., Esch, T., Stilla, U., 2016. Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks. IEEE Geosci. Remote Sens. Lett. 13 (1) http://www.ieee.org/publications_standards/publications/rights/index.html.

Mcilwaine, B., Rivas Casado, M., Leinster, P., 2019. Using 1st Derivative Reflectance Signatures within a Remote Sensing Framework to Identify Macroalgae in Marine Environments. Remote Sensing 11 (6), 704 https://www.mdpi.com/2072-4292/11/6/704.

Menardi, G., Torelli, N., 2014. Training and assessing classification rules with imbalanced data. Data Min. Knowl. Disc. 28 (1), 92–122 http://link.springer.com/10.1007/s10618-012-0295-5.

Mishkin, D., Sergievskiy, N., Matas, J., 2017. Systematic evaluation of convolution neural network advances on the Imagenet. Comput. Vis. Image Underst. 161, 11–19 https://linkinghub.elsevier.com/retrieve/pii/S1077314217300814.

Montgomery, L., Seys, J., Mees, J., 2016. To Pee, or Not to Pee: A Review on Envenomation and Treatment in European Jellyfish Species. Marine Drugs 14 (7), 127 http://www.mdpi.com/1660-3397/14/7/127.

Ng, H.W., Nguyen, V.D., Vonikakis, V., Winkler, S., 2015. Deep learning for emotion recognition on small datasets using transfer learning. In: ICMI 2015 - Proceedings of the 2015 ACM International Conference on Multimodal Interaction. Association for Computing Machinery Inc, pp. 443–449.

Nuclear Energy Institute, 2015. Economic Impacts of The R.E. Ginna Nuclear Power Plant An Analysis by the Nuclear Energy Institute. Tech. rep., Nuclear energy Institute. URL www.nei.org.

Pandey, R.K., Maiya, S.R., Ramakrishnan, A.G., 10 2018. A new approach for upscaling document images for improving their quality. In: 2017 14th IEEE India Council International Conference, INDICON 2017. Institute of Electrical and Electronics Engineers Inc., pp. 1–6. https://ieeexplore.ieee.org/document/8487796/.

Pitre, R.R., Li, X.R., Delbalzo, R., 2012. UAV route planning for joint search and track missionsan information-value approach. IEEE Trans. Aerosp. Electron. Syst. 48 (3), 2551–2565 http://ieeexplore.ieee.org/document/6237608/.

Purcell, J., Uye, S., Lo, W., 2007. Anthropogenic causes of jellyfish blooms and their direct consequences for humans: a review. Mar. Ecol. Prog. Ser. 350, 153–174 http://www.int-res.com/abstracts/meps/v350/p153-174/.

Purcell, J.E., 2005. Climate effects on formation of jellyfish and ctenophore blooms: a review. J. Mar. Biol. Assoc. U. K. 85 (3), 461–476 http://www.journals.cambridge.org/abstract_S0025315405011409.

Purcell, J.E., Baxter, E.J., Fuentes, V.L., 2013. Jellyfish as products and problems of aquaculture. In: Advances in Aquaculture Hatchery Technology. Elsevier Ltd., pp. 404–430

Qiu, J.X., Yoon, H.J., Srivastava, K., Watson, T.P., Blair Christian, J., Ramanathan, A., Wu, X.C., Fearn, P.A., Tourassi, G.D., 2018. Scalable deep text comprehension for Cancer surveillance on high-performance computing. BMC Bioinform. 19 (S18), 488 https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2511-9.

R Core Team, 2017. R: A language and environment for statistical computing.

Rawat, W., Wang, Z., 9 2017. Deep convolutional neural networks for image classification: A comprehensive review.

Razi, N., Karatas, M., 2016. A multi-objective model for locating search and rescue boats. Eur. J. Oper. Res. 254 (1), 279–293.

Reyes, A.K., Caicedo, J.C., Camargo, J.E., 2015. Fine-tuning deep convolutional networks for plant recognition. In: CEUR Workshop Proceedings. Vol. 1391.

Rianto, Y., Kondo, S., Kim, T., 1999. Detection of roads from satellite image using the optimal search. Proceedings - International Conference on Image Analysis and Processing, ICIAP 1999, 804–809 https://www.researchgate.net/publication/3820923.

Ross, N.E., Pritchard, C.J., Rubin, D.M., Dusé, A.G., 2006. Automated image processing method for the diagnosis and classification of malaria on thin blood smears. Med. Biol. Eng. Comput. 44 (5), 427–436.

Roy, S., Das, N., Kundu, M., Nasipuri, M., 2017. Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. Pattern Recogn. Lett. 90, 15–21.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vision 115 (3), 211–252.

Schaub, J., Hunt, B., Pakhomov, E., Holmes, K., Lu, Y., Quayle, L., 2018. Using unmanned aerial vehicles (UAVs) to measure jellyfish aggregations. Mar. Ecol. Prog. Ser. 591, 29–36 http://www.int-res.com/abstracts/meps/v591/p29-36/.

Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Trans. Med. Imaging 35 (5), 1285–1298.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. University of Oxford.

Skiljan, I., 2019. IrfanView. URL https://www.irfanview.com/.

Småge, S.B., Brevik, y. J., Frisch, K., Watanabe, K., Duesund, H., Nylund, A., 11 2017. Concurrent jellyfish blooms and tenacibaculosis outbreaks in Northern Norwegian Atlantic salmon (Salmo salar) farms. PLoS ONE 12 (11).

Suarez, J., Murphy, R., 2011. A survey of animal foraging for directed, persistent search by rescue robotics. In: 9th IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 314–320. https://www.researchgate.net/publication/236160246.

Takizawa, M., 2005. Countermeasures for jellyfish attacks at Kashiwazaki Kariwa [Japan] Nuclear Power Station. URL http://agris.fao.org/agris-search/search.do?recordID=JP2006001912.

Tan, M., Le, Q.V., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In: 36th International Conference on Machine Learning, ICML 2019. Vol. 2019-June. pp. 10691–10700.

Tsinalis, O., Matthews, P.M., Guo, Y., Zafeiriou, S., 2016. Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks. ArXiv. URL http://arxiv.org/abs/1610.01683.

Turner, I.L., Harley, M.D., Drummond, C.D., 2016. UAVs for coastal surveying. Coast. Eng. 114, 19–24.

Van Rossum, G., Drake, F., 2009. Python 3 Reference Manual. URL https://www.python.org/.

Vluymans, S., 2019. Learning from imbalanced data. In: Studies in Computational Intelligence. Vol. 807. Springer Verlag, pp. 81–110. http://link.springer.com/10.1007/978-3-030-04663-7_4.

Wang, G., Li, W., Zuluaga, M.A., Pratt, R., Patel, P.A., Aertsen, M., Doel, T., David, A.L., Deprest, J., Ourselin, S., Vercauteren, T., 2018. Interactive Medical Image Segmentation Using Deep Learning with Image-Specific Fine Tuning. IEEE Trans. Med. Imaging 37 (7), 1562–1573.

Wei, D., Zhou, B., Torrabla, A., Freeman, W., 2015. Understanding Intra-Class Knowledge Inside CNN. ArXiv. URL http://arxiv.org/abs/1507.02379.

Wickham, H., 2009. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag, New York http://ggplot2.org.

Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B., 2017. The marginal value of adaptive gradient methods in machine learning. In: Advances in Neural Information Processing Systems. Vol. 2017-Decem. pp. 4149–4159.

Xiao, T., Zhang, J., Yang, K., Peng, Y., Zhang, Z., 2014. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In: MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia. Association for Computing Machinery Inc, pp. 177–186.

Xu, A., Viriyasuthee, C., Rekleitis, I., 2011. Optimal complete terrain coverage using an unmanned aerial vehicle. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 2513–2519.

Yanmaz, E., 2012. Connectivity versus area coverage in unmanned aerial vehicle networks. In: IEEE International Conference on Communications, pp. 719–723.
Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. Vol. 4. pp. 3320–3328.
Zhang, F., Sun, S., Jin, X., Li, C., 2012. Associations of large jellyfish distributions with temperature and salinity in the Yellow Sea and East China Sea. In: Jellyfish Blooms IV. Springer, Netherlands, pp. 81–96.



https://dspace.org/

JellyNet: The convolutional neural network jellyfish bloom detector

Mcilwaine, Ben; Rivas Casado, Monica

2021-01-21
Elsevier
https://doi.org/10.1016/j.jag.2020.102279
Mcilwaine B, Rivas Casado M. (2021) JellyNet: The convolutional neural network jellyfish bloom detector. International Journal of Applied Earth Observation and Geoinformation, Volume 97, May 2021, Article number 102279

Downloaded from a DSpace repository. https://dspace.org/