OPEN-SET SOURCE ATTRIBUTION FOR PANCHROMATIC SATELLITE IMAGERY

E. D. Cannas[†] *S. Baireddy*[⋆] *E. R. Bartusiak*[⋆] *S. K. Yarlagadda*[⋆]
D. Mas Montserrat[⋆] *P. Bestagini*[†] *S. Tubaro*[†] *E. J. Delp*[⋆] [*]

[†]Image and Sound Processing Lab (ISPL), Department of Electronics, Information and Bioengineering,
Politecnico di Milano, Milan, Italy
[⋆]Video and Image Processing Laboratory (VIPER), School of Electrical Engineering,
Purdue University, West Lafayette, IN, USA

ABSTRACT

In the last few years, several companies started offering the possibility of buying different kinds of overhead images acquired by satellites orbiting around the planet. This market is interesting for several customers, from those who simply fancy a shot of their house from space, to those aiming to acquire strategic information on portions of land. Due to the sensitive nature of this data, which can be maliciously altered by anyone, the forensic community has started investigating methodologies to verify overhead imagery authenticity and integrity. Within this context, in this paper we investigate the possibility of using Convolutional Neural Networks (CNNs) to attribute a panchromatic satellite image to the satellite used to acquire it. In our investigation we tackle both closed-set and, adapting Deep Ensemble (DE) and Monte Carlo Dropout (MCD) techniques, open-set image attribution problems.

***Index Terms*—** Image forensics, overhead images, source attribution, open-set classification, deep learning

1. INTRODUCTION

Editing and manipulating digital images is becoming easier over time. For instance, it is possible to use one of the many available and user-friendly software suites (e.g., Adobe PhotoShop, GIMP, etc.) or, alternatively, to rely on automatic services and techniques driven by deep learning and computer vision technologies (e.g., Face2Face, Faceswap, etc.). In order to avoid the widespread use of maliciously manipulated pictures, the multimedia forensics community has put a great effort into developing image authenticity and integrity assessment techniques [1].

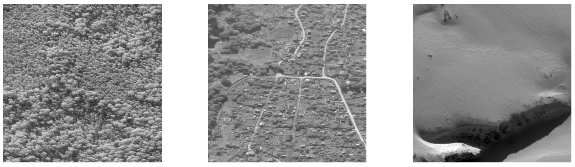
In additional to common photograph manipulation, many image editing tools can be successfully used to forge other kinds of imagery. This is the case of overhead images, i.e., images of the ground acquired by satellites orbiting around the planet. Obtaining this kind of data is nowadays extremely simple. As a matter of fact, it is possible to freely download them from multiple websites [2], or to buy them from specialized companies. As malicious editing of this kind of images has already led to serious consequences [3], developing forensics techniques tailored to overhead image analysis is now a necessity.

It is worth noting that manipulation of overhead images has become a threat only recently due to the current possibility of gathering satellite data. Therefore, state-of-the-art techniques for their forensic analysis are still underdeveloped, and they mostly focus on forgery detection and localization. As an example, the authors of [4] have proposed a method to localize inpainted regions on satellite images. The work in [5] introduces a method to detect and localize overhead image splicing using a Generative Adversarial Network (GAN) paired with a one-class classifier. The authors of [6] have proposed a method to detect forgeries using a conditional GAN. In [7], forgery detection is cast as an anomaly detection problem. Finally, in [8] generative autoregressive models are used to model the image pixel distribution and detect deviations.

In this paper, we consider a different forensic problem that has been broadly studied for natural images but not yet investigated for satellite data to the best of our knowledge: image source attribution. Attributing an image to its source means detecting which acquisition device has been used to capture the image under analysis. For natural images, this problem has been studied at different granularity in the literature [9]: some methods consider the problem of detecting which kind of device was used for the acquisition (e.g., camera vs. scanner); other methods consider the problem of detecting the specific brand or model (e.g., Sony vs. Canon); other methods considered the problem of detecting the specific instance of the device (e.g., this iPhone X vs. that iPhone X).

Our goal is to understand which satellite has been used to acquire a panchromatic image under analysis. More specifically, we consider this attribution problem both in closed-set and open-set. In the first one, we assume that the image may only come from a set of known satellites. In the second one, we assume instead that the image may also come from a satellite that is unknown to the system, yet the system should be able to label the image as such. The proposed method is based on the use of a Convolutional Neural Network (CNN) that acts as classifier. In order to deal with the open-set problem, we exploit the idea behind model uncertainty recently used also in forensic scenarios [10]. Specifically, we adapt and compare two different strategies: Deep Ensemble (DE) [11] and Monte Carlo Dropout (MCD) [12]. The main idea is that it is possible to run multiple attribution tests on a single image under analysis. If the image comes from an unknown satellite, the trained CNN should exhibit an uncertain behavior across its responses. This behavior can be captured and used to reject the image as coming from an unknown satellite. Otherwise, the image is attributed to the originating satellite.

[*]This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.



(a) Forest (b) Urban (c) Snow

Fig. 1. Panchromatic image samples from different regions.

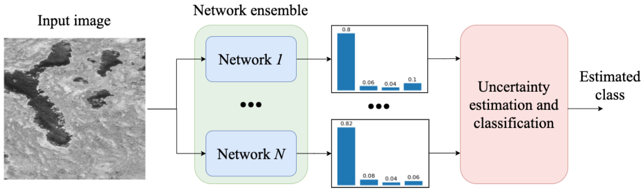


Fig. 2. Complete pipeline of the considered source attribution method. Each CNN in the ensemble returns an M-element vector representing the likelihood for each class. Uncertainty is evaluated based on the coherence of the scores. Attribution is performed based on the estimated uncertainty.

2. SATELLITE ATTRIBUTION

The satellite attribution problem we tackle can be formulated as follows. Given a panchromatic image $\mathbf{I}$ (see Figure 1) and a set of M known satellites, we want to understand if $\mathbf{I}$ has been acquired with one of the M known satellites or not (i.e., open-set). If the answer is positive, we also want to determine which satellite's sensor among the M considered ones has generated it (i.e., closed-set).

In order to do so, we rely on a CNN ensembling method as depicted in Figure 2. Considering N CNNs trained in closed-set over the M known classes, we test the image against all of them, obtaining N attribution scores, and then we evaluate the classification uncertainty of the ensemble based on these responses. If the evaluation of the uncertainty is high, we conclude that the image does not belong to any of the known satellites. Otherwise, we attribute the image according to the closed-set classification response. The following subsections describe the ensembling and uncertainty evaluation techniques.

2.1. Model ensembling

In this work we consider two different methods to obtain an ensemble of N classifiers. One is based on DE [11] and the other one id based on MCD [12]. Both techniques rely on a single network model, which in our case is the EfficientNetB0 of the EfficientNet family [13], which offers a good trade-off in terms of number of parameters and performances and has recently been deployed in the forensics literature consistently [14, 15, 16]. The methods differ in the way the network is trained and scores are obtained.

Deep Ensemble. Let us consider a backbone CNN (EfficientNetB0 in our experiments). Having M known satellites, we train this CNN in closed-set as an M-class classifier. This means that the output of the CNN is an M element vector

$$\mathbf{y} = [y_0,\ y_1, ...,\ y_{M-1}], \tag{1}$$

where the m-th element y_m represents the likelihood of the sample to belong to the m-th satellite. We repeat the training procedure N times, starting each time from a different seed for the random initialization of network weights. This provides us with N different trained CNN instances with slightly different weights due to the optimization process. The ensemble is then composed of these N models, where the n-th model returns the prediction vector $\mathbf{y}_n$. While surely presenting some computational issues, this method is easily scalable and parallelizable at both the training and testing stage.

Monte Carlo Dropout. Let us consider a backbone CNN (an EfficientNetB0 in our experiments). Considering M known satellites, this time we train a single instance of the network as an M-class classifier using dropout [17] before the last fully connected inner-product layer. Also in this scenario, the output of our network is an M element vector $\mathbf{y}$, where each element represents an attribution score for one of the known satellites. To obtain the network ensemble, during testing, we keep dropout active and test the sample image N times. This means that each inference is obtained by randomly turning off a series of neurons in the penultimate layer. This operation can be viewed as processing the image with N different sub-networks, each one contained in the parameter space of the complete trained network, with weights chosen according to the dropout distribution. We obtain N vectors $\mathbf{y}_n$ per tested image.

While the original idea is described as model averaging in [17], the authors of [12] demonstrated how the use of dropout in neural networks actually makes the network approximate Bayesian inference in a deep Gaussian process [12]. This allows us to represent the model uncertainty with a theoretically founded base, while maintaining the available deep learning tools and with very little cost in terms of testing time and no additional costs in training time.

2.2. Uncertainty Analysis

Given a sample image and the N CNN outputs obtained by one of the ensembling techniques, our goal is to estimate the model's uncertainty on that sample. Specifically, we are interested in modeling the epistemic uncertainty, i.e., the uncertainty on the model parameter values resulting from the fact that our training data inevitably captures only a portion of the input domain.

Since both techniques aim at modeling this kind of uncertainty in a deep model, our goal is to exploit this estimation to characterize samples of unknown classes as samples out of the distribution of the training data, i.e., samples for which the confidence of our models in processing them is low. To do so, we slightly modify the uncertainty estimation proposed by the authors of both DE and MCD.

We consider the N M-element outputs $\mathbf{y}_n,\ n \in [0, N-1]$ altogether as samples of a multi-dimensional distribution in $\mathbb{R}^M$, where M is the number of classes considered. We evaluate the entropy $H(\mathbf{y})$ of these N realizations and use it as uncertainty measure. Intuitively, if all N CNNs point to the same class (i.e., low entropy), we tend to trust the classification output (i.e., the image belongs to the satellite pointed by all networks). Conversely, if the N networks provide inconsistent results (i.e., high entropy), we conclude that none of the satellites has acquired the image, which is then attributed to the "unknown" class (i.e., a satellite not in the training set).

3. EXPERIMENTS

3.1. Dataset

In order to test the effectiveness of the investigated pipeline, we created a dataset of satellite images. We collected 8-bit panchromatic images from the DigitalGlobe portal [18] coming from 5 different

Table 1. Classification report on the closed-set scenario for the baseline network trained with all classes of panchromatic images.

All classes	GE01	QB02	WV01	WV02	WV03
Precision	0.772	0.768	0.965	0.901	0.780
Recall	0.710	0.894	0.908	0.814	0.853
F1-score	0.740	0.827	0.936	0.815	0.830
Overall accuracy	0.832				

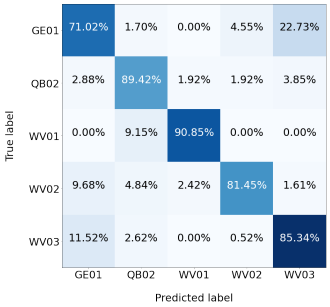


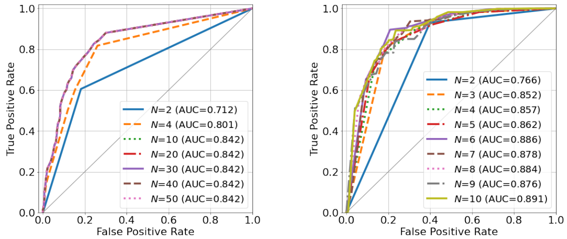
Fig. 3. Confusion matrix for a baseline EfficientNetB0 trained with all satellite classes available in closed set (accuracy of 83%).

satellites: GeoEye (GE01), QuickBird (QB02), WorldView 1, 2 and 3 (WV01-02-03). All images are provided in GeoTIFF format, with no compression, in non-overlapping tiles of 16384x16384 pixels. All tiles are orthorectified, sensor and radiometrically corrected. To avoid a possible bias due to the semantics of the scene represented in the images, we have selected different geographical regions, so that for each satellite we have samples coming from urban, snowy, barren, forest and field areas. Some samples are reported in Figure 1. However, as these images were too large to be processed by our baseline network, we performed a patch extraction procedure on each sample. Specifically, we extracted pixel patches of size 1024×1024, ending up with a dataset of roughly 3000 images equally balanced in number among all satellite classes and in the geographical areas represented.

3.2. Setup

The baseline network we considered in our setup is an EfficientNetB0 that we trained as an M-class classifier. The network is trained from scratch, as most available pre-trained models work on natural images, therefore having a strong mismatch with respect to the panchromatic imagery we are analyzing. For closed-set experiments, we consider $M = 5$ (i.e., all available satellites in the dataset) and we follow a train-validation-test scheme, where we use roughly 50% of samples for training, 25% for validation, and the remaining 25% for testing. For the open-set experiments, we consider $M = 4$ (i.e., we leave one satellite out of the training set) and we add to the test set all images coming from the unknown satellite (i.e., the one left out during training). Open-set experiments are repeated five times following a leave-one-out procedure (i.e., each satellites is considered as unknown in one experiment).

For MCD we adopted a single seed for random initializing the model. For DE, we trained 10 networks using 10 different seeds increasing in value computed with a fixed step of 10. Finally, regarding CNN hyperparameters, all networks have been trained using batches of 5 images and a simple cross-entropy loss for 200 epochs, relying on Adam [19] for optimization. We started with a learning rate of 0.001 reduced on a plateau of the validation loss after 10 consecutive epochs, and early stopping of the training if the validation loss did not improve after 50 consecutive epochs or if the learning rate reached a minimum of 10^{-8}. We used PyTorch [20] as our deep learning framework.



(a) MCD ensembles with $N \in \{2, 4, 10, 20, 40, 50\}$.

(b) DE ensembles with $N \in \{2, \ldots, 10\}$.

Fig. 4. ROC curves obtained using MCD or DE ensembles with different N values. AUC stops increasing at a for $N \geq 10$.

All experiments have been run on a machine equipped with an Intel Xeon E5-2687W-v4 and a NVIDIA Titan V. Training of the baseline networks took 2-6 hours, while the test inference on a single sample by a single network took on average 0.1 seconds. This number must be multiplied accordingly to the ensembling technique adopted (i.e., MCD or DE).

3.3. Evaluation Metrics

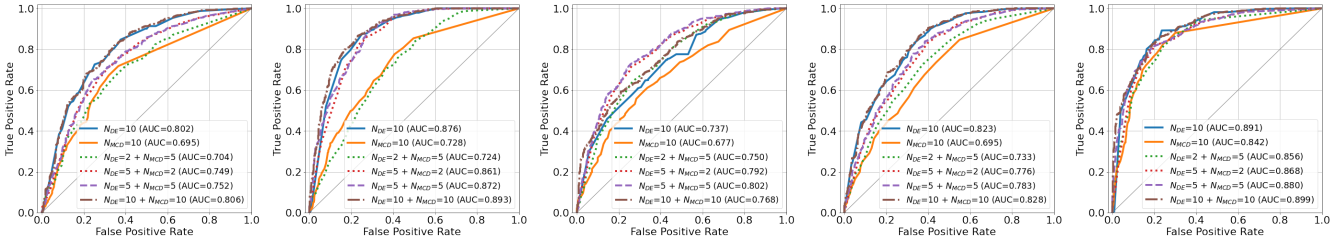
To measure the performance of the investigated solution, in the closed-set scenario we use accuracy, precision, recall, and F1-score. For the open-set scenario, we have to set a threshold on the entropy of the ensemble's output. We therefore decided to use Receiver Operating Characteristic (ROC) curves and the corresponding Area Under the Curve (AUC) to see if it is possible to clearly distinguish samples outside the training distribution based on this threshold. In this scenario, threshold intervals for the ROC curves vary depending on the network ensemble used and are automatically computed by the implementation available in the Scikit-learn library [21].

4. RESULTS

4.1. Closed-set

The first experiment is performed to evaluate whether the selected backbone network can solve the source attribution problem in closed-set. To this purpose, we trained the EfficientNetB0 considering all the available satellites (i.e., $M = 5$). Figure 3 reports the obtained confusion matrix, while Table 1 reports all the achieved metrics. These results show that it is possible to exploit EfficientNetB0 for this task, however also highlighting that the problem is more challenging with respect to classic camera attribution. In particular, we can see that 11.52% of the samples of class GE01 are mistaken for WV03 and 9.68% for WV02, and vice versa 22.73% of WV03 samples are classified as GE01.

The second experiment is aimed at understanding the performance of the baseline network trained on $M = 4$ classes. Despite this seeming redundant, it is important for the open-set evaluation. Indeed, in the open-set setup, we exclude one of the five known satellites from the training set, thus considering the backbone network as



(a) Training without GE01 samples.
(b) Training without QB02 samples.
(c) Training without WV01 samples.
(d) Training without WV02 samples.
(e) Training without WV03 samples.

Fig. 5. Open-set results for the leave-one-class experiments. N_{DE} and N_{MCD} denote N values for DE and MCD whenever they are combined.

Table 2. Classification report on the closed-set scenario considering the worst and best leave-one-class out experiments in terms of overall accuracy.

WV01 samples out	GE01	QB02	WV02	WV03
Precision	0.786	0.872	0.860	0.864
Recall	0.727	0.927	0.888	0.878
F1-score	0.755	0.900	0.874	0.871
Overall accuracy	0.844			
WV03 samples out	GE01	QB02	WV01	WV02
Precision	0.981	0.943	0.941	0.888
Recall	0.898	0.961	0.973	0.945
F1-score	0.938	0.952	0.957	0.916
Overall accuracy	0.940			

an $M = 4$ class classifier. Table 2 reports the classification metrics related to the worst and best, in terms of overall accuracy, leave-one-class out experiments with the baseline network, i.e., a single (no ensemble) EfficientNetB0. While all CNNs reach an accuracy equal or greater than 85%, performances on the same classes of satellite are not consistent across all experiments. The most evident case is the one of satellite GE01, for which the baseline network reaches an F1 score of 0.755 when trained without the WV01 samples, whereas in the other three scenarios achieves results equal or above 0.9.

These numbers suggest that satellite attribution maintains a certain level of difficulty even for a very simple dataset, fact which is more surprising when considering that each sensor model possesses unique features related to design, processing chain and satellite characteristics that should make this task closer to the well-known forensic camera model identification.

4.2. Open-set

For the open-set classification task, our efforts initially focused in determining an appropriate number of networks for both MCD and DE. To this purpose, Figure 4 shows the ROC curves obtained using MCD and DE with a different N in the case EfficientNetB0 is trained over four satellites and WV03 samples are used as unknown class. It is possible to observe that the AUC increases with N, but has diminishing returns after $N = 10$

The next experiment we performed aimed at evaluating DE, MCD and their combination considering all five leave-one-class out scenarios (i.e., four satellites in the training set, and one left out as unknown). Figure 5 shows these open-set classification performances. In addition to the DE and MCD with $N = 10$, we considered two equivalent combinations of 10 networks obtained with 5 and 2 MCD inferences of a DE of 2 and 5 networks respectively, and an ensemble of 25 and 100 networks considering a combination of 5 and 10 MCD inferences with a DE of 5 and 10 networks respectively. In all these experiments we use N_{DE} and N_{MCD} to distinguish the N value used for DE and MCD.

For all 5 classes we obtained high AUC values with every ensemble. DE offered always better performances with respect to MCD, with a minimum AUC of 0.73 for the WV01 satellite open-set classification and a maximum of 0.891 for WV03. Interestingly, the equivalent combination of MCD and DE, while always performing equivalently or better than MCD, did not always reach the performances obtained by the 10 networks DE, with the only exception being the case where WV01 is used as unknown class. Moreover, the simple 10 networks DE almost always performed better than the combination of 25 networks obtained with 5 MCD inferences of the 5 networks DE, and comparably with the final combination of 10 MCD inferences of the 10 networks DE.

Looking also at the better performances obtained by the combination of the 5 networks DE with the 2 MCD inferences with respect to its specular equivalent, these numbers seem to suggest that the sampling of the scores' distribution obtained through DE gives more advantages with respect to MCD. However, the results obtained for the open-set classification of the WV01 samples seem also to point to the fact that a smaller number of networks, better conditioned due to their initial random initialization, might sometimes perform better than a bigger ensemble.

5. CONCLUSIONS

In this paper we investigated the problem of satellite attribution for panchromatic images. We considered this problem both in closed-set and open-set. In closed-set, we assume that the image under analysis may only come from a series of known satellites. In the open-set scenario, we assume that the image under analysis may also come from a different source not known at training time.

In order to cope with the open-set problem, we adapted a well-known Convolutional Neural Network (CNN) to our goal, and we investigated two different strategies: Deep Ensemble (DE) [11] and Monte Carlo Dropout (MCD) [12]. Both strategies slightly perturb a single network to obtain multiple predictions for each image under analysis. These pieces of information from all predictions are then exploited to obtain the final answer. Results on the gathered dataset show the feasibility of the proposed approaches. Future work will be devoted to explore the challenges of a possible large-scale test with a consistent number of satellites, like those of Planet platform [22].

6. REFERENCES

[1] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, pp. 22, 2013.

[2] GIS Geography, *15 Free Satellite Imagery Data Sources*, August 2017 (accessed January 12, 2020), *http://gisgeography.com/free-satellite-imagery-data-list*.

[3] BBC News, *Conspiracy Files: Who shot down MH17?*, April 2016 (accessed January 12, 2020), `http://www.bbc.com/news/magazine-35706048`.

[4] L. Ali, T. Kasetkasem, F. G. Khan, T. Chanwimaluang, and H. Nakahara, "Identification of inpainted satellite images using evalutionary artificial neural network (EANN) and k-nearest neighbor (KNN) algorithm," *IEEE International Conference of Information and Communication Technology for Embedded Systems (ICICTES)*, 2017.

[5] S. K. Yarlagadda, D. Güera, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp, "Satellite image forgery detection and localization using gan and one-class classifier," in *IS&T Electronic Imaging (EI)*, 2018.

[6] E. R. Bartusiak, S. K. Yarlagadda, D. Güera, P. Bestagini, S. Tubaro, F. M. Zhu, and E. J. Delp, "Splicing detection and localization in satellite imagery using conditional GANs," in *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019.

[7] J. Horvath, D. Güera, S. K. Yarlagadda, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp, "Anomaly-based manipulation detection in satellite images," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019.

[8] D. Mas Montserrat, J. Horvath, S. K. Yarlagadda, F. M. Zhu, and E. J. Delp, "Generative autoregressive ensembles for satellite imagery manipulation detection," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020.

[9] M. Kirchner and T. Gloe, "Forensic Camera Model Identification," in *Handbook of Digital Forensics of Multimedia Data and Devices*, pp. 329–374. John Wiley & Sons, Ltd, Chichester, UK, 2015.

[10] B. Lorch, A. Maier, and C. Riess, "Reliable jpeg forensics via model uncertainty," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020.

[11] L. Balaji, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *International Conference on Neural Information Processing Systems (NIPS)*, 2017.

[12] Gal Y. and Ghahramani Z., "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*, 2016.

[13] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.

[14] N. Bonettini, E.D. Cannas, S. Mandelli, L.Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of cnns," in *International Conference on Pattern Recognition (ICPR)*, 2021.

[15] L. Bondi, E.D. Cannas, P. Bestagini, and S. Tubaro, "Training strategies and data augmentations in cnn-based deepfake video detection," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020.

[16] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro, "Training cnns in presence of JPEG compression: Multimedia forensics vs computer vision," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[18] Maxar Technologies, *DigitalGlobe discover portal*, accessed January 12, 2021, *https://discover.digitalglobe.com/*.

[19] D.P. Kingma and J. Ba, "Adam: a method for stochastic optimization. arxiv: 14126980," 2014.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] Planet, *Planet Imagery products*, June 2021 (accessed June 15, 2021, *https://www.planet.com/products/planet-imagery/*.