Multi-Resolution CFAR and CNNs for xView3

Andrew Follmann and Brian Josey

April 1, 2022

Abstract

xView3 was an international challenge hosted by the United States Department of Defense's Defense Innovation Unit in partnership with Global Fishing Watch to supply synthetic aperture radar (SAR) data to the public and promote research in computer vision to combat illegal, unreported, and unregulated (IUU) fishing. IUU fishing poses a major ecological, economic, and geopolitical threat to global fisheries and the communities that depend on them. In this whitepaper, we detail our strategy for detecting and classifying fishing vessels engaged in IUU fishing. We developed a novel implementation of the constant false alarm rate algorithm that passes over the SAR scenes at different image resolutions and detects objects with improved $F1$-accuracy and speed. We then passed the images to a VGG16-style convolution neural network (CNN) to classify ships into four labels. Finally, a shallow CNN predicts vessel length. With this approach, we were able to identify and classify fishing vessels with a high degree of accuracy and achieved seventh place among verified solution.

1 Introduction

The xView challenges are a series of international computer vision competitions organized by the United States Department of Defense's Defense Innovation Unit (DIU) in conjunction with partner organizations [1]. The primary goal of these challenges is to source, clean, and make available large sets of satellite and remote sensing data to the public and advance research while solving urgent problems. Past challenges included classifying objects in commercial satellite images (xView1 [2]) and determining the degree of damage buildings sustained after natural disasters (xView2 [3]). For xView3, DIU partnered with Global Fishing Watch [4] to identify the best computer vision algorithms to combat illegal, unreported, and unregulated (IUU) fishing. IUU fishing poses a threat to geopolitical stability, food supplies, and marine ecosystems. It is estimated that as much as 20 % of fish caught are unreported, and IUU fishing contributes between US$10 billion to US$23 billion in lost economic activity annually, primarily due to fishery depletion and related environmental harm [5].

Our task was to identify ships from a series of images, classify them into one of four categories, and estimate the length of the fishing vessels [6]. The data used for this project were composed of synthetic aperture radar (SAR) scenes depicting different locations on the Earth's surface. Data was captured by the European Space Agency's (ESA) Sentinel-1 satellite constellation [7]. SAR is an all-weather, all-conditions imaging technique that images the Earth's surface by measuring the backscattering of microwave light (Sentinel-1: $f = 5.405$ GHz, $\lambda = 5.55$ cm) emitted by a satellite [8]. It creates a "synthetic" aperture by taking multiple images along its orbit of the same surface locations and combines them into one image with greater resolution. It takes two simultaneous images by emitting light polarized vertically (parallel to the flightpath) and measuring the returning light along the vertical and horizontal directions. The two polarization bands represent different

types of surface scattering. The vertical-to-vertical (VV) polarization is sensitive to surface roughness, smooth surfaces like water result in lower signal intensity [8]. The vertical-to-horizontal (VH) polarization correlates to volume scattering, objects with multiple scattering loci such as trees or buildings give a greater signal intensity [8].

The DIU supplied a data set composed of 750 scenes taken at various locations in the Atlantic Ocean and Mediterranean Sea. Each scene was composed of seven data sets taken simultaneously: two high-resolution SAR images, VV and VH, and five supplemental images. The SAR images were approximately 1 gigapixel each, 25,000 pixel × 50,000 pixel, where each pixel represents the intensity of backscattered light on a logarithmic scale and covers a 10 m × 10 m area. The five supplemental images covered the same locations but at a 200 m × 200 m resolution. The first three images include data about the wind including the speed (m/s), direction (degrees east from true north), and quality (0 to 3 integer scale). The fourth image included the altitude and bathymetery, seafloor depth, which is used to create the final image: a mask that categorizes each pixel as land, sea, or ice. DIU corrected each scene for the Earth's curvature and measurement geometry before releasing it to the public and no further corrections were needed. Details about how the images were collected and processed are given in [6].

DIU organized the public data into 550 training scenes, 50 validation scenes, and 150 test scenes, totaling to approximately 1.7 TB. A combination of automatic computer processing and human experts labeled objects in the scenes and assigned confidences scores of low, medium, or high to them. Images in the test set were not labeled, but a key was provided to compare our results against. Finally, an unspecified number of scenes were withheld for model scoring. Details of how the images were labeled and assigned to each category are given in [6].

Entries were required to be submitted in Docker containers [9] and work through the withheld data set in under fifteen minutes on a computer with a GPU, 8 CPU cores, and 60 GB RAM. Models were scored in five categories: maritime object detection, close-to-shore object detection, vessel classification, and vessel length estimation. A composite score, M_R, was calculated as

$$M_R = F1_D \times \frac{1 + F1_S + F1_V + F1_F + PE_L}{5}, \tag{1}$$

where $F1_i$ is the $F1$-score for criterion i: $F1_D$ is for object detection, $F1_S$ is close-to-shore detection, $F1_V$ is vessel classification, and $F1_F$ is fishing vessel classification. PE_L is the aggregate percent error metric used for the vessel length estimate

$$PE_L = 1 - \min\left(\frac{1}{N}\sum_{n=1}^{N}\frac{|\hat{l}_n - l_n|}{l_n}, 1\right), \tag{2}$$

where l_n and $\hat{l}_n$ denote the true and estimated length, respectively, of object n and N is the total number of vessels. Entries were then ranked by M_R and prizes awarded accordingly.

2 Methods

2.1 Preprocessing

To focus our search on the relevant data, we selected SAR data in the range of −50 dB to 20 dB and bathymetry from −500 m to 0 m and disregarded the remainder. We rescaled the intensity and seafloor depth to a linear range from 0 to 1. To load and resize the images, we used the Python packages CV2 [10] and rasterio [11], using an area-based resampling for SAR data and bilinear resampling for all other channels.

2.2 Object Detection

We adapted the constant false alarm rate (CFAR) algorithm [12, 13] used for object detection in radar data by adding a multi-resolution component, which we dubbed multi-resolution CFAR (MR-CFAR). Our algorithm improved upon CFAR by passing over the data at different resolutions with each successive pass looking only at anomalous pixels, those with a high relative intensity raised by a previous pass. This adaptation reduced the search space of a full-resolution CFAR search and had a lower false positive rate. By exploiting the spatial nature of vessels in the data, we collected many high signal pixels that represented a single object in the high-resolution images. We then down sampled to a lower resolution and aggregated these pixels into a super-pixel that maintained the high relative signal but required less processing.

2.2.1 CFAR

The CFAR algorithm [12, 13] is used for object detection in radar images where it exploits the spatial nature of the targets to identify them in the face of noise an interference. Because radar is an active sensing system, objects backscatter the source beam strongly, which is detected as signals at a higher intensity than the surroundings. CFAR compares a given observation, a pixel in our case, against the surrounding data. If the selected pixel has a higher intensity than the neighboring pixels it is recognized as an object.

As with the search for unidentified maritime objects (SUMO) algorithm commonly used to identify ships from SAR images [14], we used pixel-wise OR logic across the VV and VH channels to raise anomalies from either channels. These pixels were then passed to a cell averaging CFAR algorithm [15] that identified a mean and standard deviation for the intensity among the neighboring pixels, which we used as a test statistic for flagging against a specified threshold.

One complication we encountered was the spatial nature of the objects. To compare a pixel against its background and not other pixels that represent the same object, we obscured the immediate neighborhood of the observation pixel. We specified a number of "guard" cells surrounding the observation pixel that formed a mask and an external "window" beyond the mask. We computed a mean and standard deviation for the pixels in the window and compared it to the observation pixel with a normalized Z-score. If the Z-score was above a specified threshold, it was labeled as an object.

2.2.2 Multi-Resolution CFAR

We adapted CFAR to be faster and more accurate, by serially applying the algorithm at different resolutions and with different guard sizes, window sizes, and Z-score thresholds, see Figure 1. We performed a first pass on a down sampled, low-resolution SAR image, using the land/ice/water mask to exclusively look at the water. We used bilinear resampling to ensure we included all shoreline data. We found the optimal performance occurred at 15 % down sampling level after also looking at the 10 % and 20 % levels. We used 1-cell guard and a 3-cell window, and a Z-score threshold of 3. Subsequently, with the objects detected after the low-resolution pass, we repeated the search at a medium resolution, of 50 %. We conduct the same CFAR procedure, with a 3-cell guard and a 7-cell window, and Z-score threshold of 3.5. With the objects detected at the medium resolution, we then ran two final passes at full resolution, with two related CFAR passes with different guard sizes, one with a guard in the vertical direction of 15 cells, a horizontal guard of 7 cells, and a horizontal and vertical window of 15 cells. The second run was with the guards reversed, a vertical guard of 7 cells and a horizontal guard of 15 cells, and the same 15 cell window. Both methods had a Z-score threshold of 5. This implementation allowed the CFAR method to highlight the ships

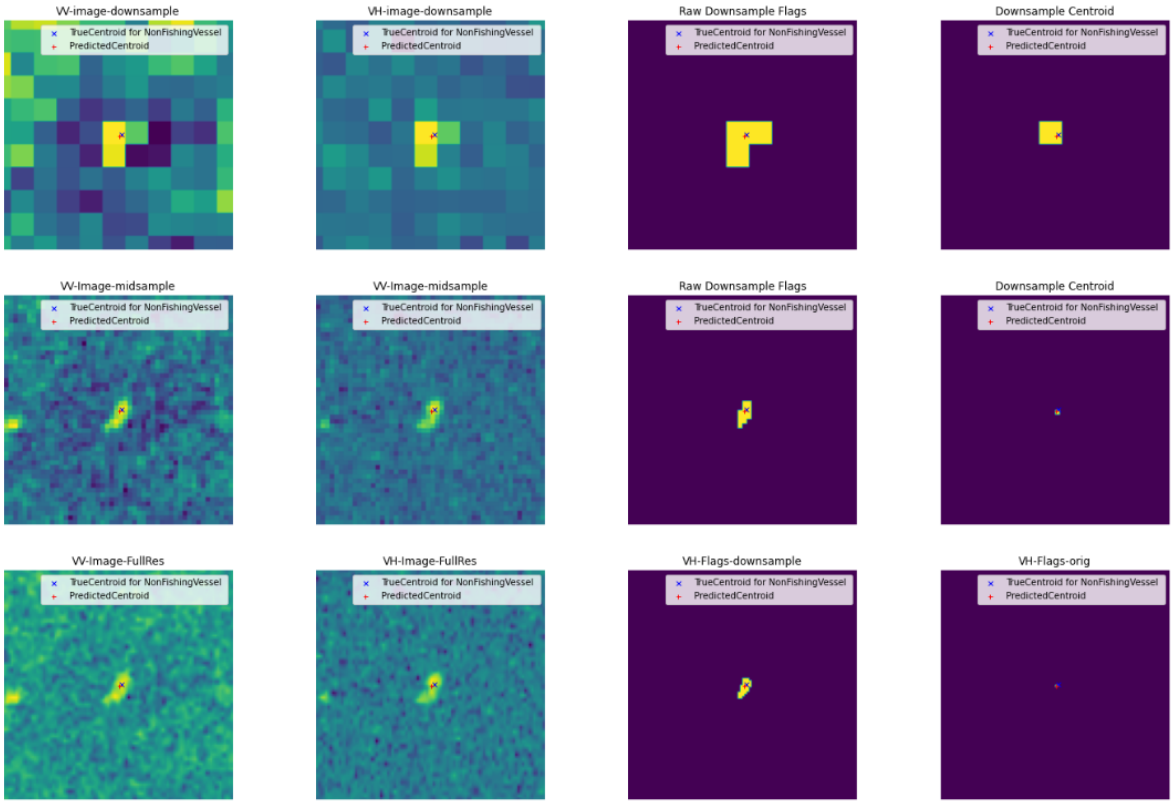

Figure 1: **Example of intermediate MR-CFAR outputs.** Sample images of SAR data and MR-CFAR outputs at different resolutions for the same object. Each row represents different resolutions from low (10 %) at the top, to medium (50 %) in the middle, and full resolution (100 %) at the bottom. The columns represent from left to right, the VV polarization, VH polarization, CFAR identified anomalous pixels, and CFAR anomalous pixel centroids.

that belonged to a series of ships or extended ships with long tails in the horizontal or vertical directions. The two CFAR passes were combined bitwise, which surfaced any ships with a high relative intensity in any direction.

Once we identified the object pixels, we found the centroids and removed any duplicates identified as being too near to one another, those within a 10 pixels window. Dropping the duplicates maintained the centroid that had more associated pixels and removed the one with fewer pixels. We then pass a stack of full resolution, 64 pixel $\times$ 64 pixel, two-channel images (VV and VH) centered on the centroid to the object classification model.

2.3 Object Classification

We selected the VGG16 algorithm [16] as the base of our image classification model. This model is often used as a benchmark classification algorithm because it is open source and highly accurate, having achieved 93 % accuracy on the ImageNet data set, which contains over 14 million images in 1,000 classes [17]. We modified the built-in VGG16 model from the Keras library to accept the 64 pixel $\times$ 64 pixel images outputted by the MR-CFAR algorithm in two channels: VV and VH. Supplemental information about wind and bathymetry were not used for training. Images were classified into the four classes specified by DIU [6]: "non-objects" for naturally occurring objects

like reefs and sea rocks, "non-vessels" for human-made objects like lighthouses and oil rigs, "non-fishing vessels" for ships engaged in activities other than fishing, and "fishing vessels" for any ship engaged in fishing.

Our model had the same architecture as the baseline VGG16 model described in [16], but had fewer trainable parameters, 39,904,516, due to the decreased input image size and reduced number of classes. The model trained using sparse categorical cross entropy as the loss function and accuracy as the training metric. We used stochastic gradient descent (SGD) as the optimizer. Although SGD is known to converge to a solution more slowly during training than other optimizers, such as Adam [18–20], we selected it because we found that optimizers like Adam would often get stuck at a local maximum accuracy and miss the global maximum without careful tuning of the learning rates. This decision led to an increased training time, but a greater accuracy, precision, and recall.

Training was performed on the MR-CFAR output images with a 9 to 1 training-validation split. We specified custom training and validation batches due to a substantial discrepancy in data quality between the prescribed sets. To create this split, we shuffled the scene-level folders containing the data and randomly assigned each to either set. We created a custom data loader using Keras' ImageDataGenerator object [21] to rescale, rotate, shear, flip, and shift the available training images to increase the number of images available for training. We specified maximum ranges of 90 degrees for rotations, 20 pixels for horizontal and vertical shifts, and 0.2 for shear and zoom. Additionally, we also flipped images along the horizontal and vertical axes.

The object classification model was trained in two batches. The first batch consisted of the medium and high confidence predictions from the xView3 data providers, as well as 100 explicitly negative images per scene. These negative images were identified as pixels that were more than 40 pixels from an object identified with low, medium, or high confidence. The second batch consisted of the first batch plus additional examples that were false positives from the MR-CFAR object detection. This second batch was more representative of what the model would need to predict on during test, and the addition of false positives greatly increased the model performance. Training was performed using Amazon Web Service's SageMaker hardware [22]. Each training run took approximately 12 hours to complete on a P1 instance, which runs a V100 GPU with 64 GB of RAM.

2.4 Length Estimation

To estimate the length of the fishing vessels, we created an 8-layer CNN that classified the images identified as fishing vessels and assigned them a single real value output. The model was composed of three two-dimensional convolution layers with max pooling and a dropout rate of 0.2, followed by a flattening layer, and two dense layers with rectified linear unit activation. The final layer was a dense layer that returned a fitted linear number that was converted from pixels to length in meters. The model was trained once on images labeled with medium or high confidence by the data providers as fishing vessels. The model used the Adam optimizer and mean squared error for the loss function and model metric.

3 Results

Our model improved over the baseline model provided by DIU significantly. The final scores of our model, the baseline model provided by DIU [23], and the top scoring model [24] are summarized in Table 1. Our model had an overall score that was more than twice that of the baseline, $M_R = 0.19$ *vs.* 0.42, but fell short of the final winner, $M_R = 0.60$. The increase in our score was primarily driven by improvements in the object detection and object and fishing vessel classifications, as

well as the addition of the length estimation. The smallest increase occurred in the close-to-shore detection category, $F1_S = 0.12$ *vs.* 0.15.

	M_R	$F1_D$	$F1_S$	$F1_V$	$F1_F$	PE_L
Baseline	**0.19**	0.43	0.12	0.71	0.4	0.0
Top Score	**0.60**	0.75	0.52	0.95	0.83	0.69
TRSS	**0.42**	0.61	0.15	0.92	0.75	0.62

Table 1: **Comparison of model results.** The overall score, M_R, for the reference model supplied by DIU (Baseline), highest scoring model (Top Score), and our model (TRSS). M_R is determined by Equation 1 and depends on the $F1$-scores for the object detection ($F1_D$), close-to-shore ($F1_S$), object classification ($F1_V$), and fishing vessel classification ($F1_F$) and the percent error metric for the vessel length (PE_L) given by Equation 2.

4 Discussion

While the model performed well, the final score was more than twice that of the reference model and achieved seventh place in the verified solutions, there is room for improvement. The model's performance was weakest in object detection, particularly close-to-shore. Building a successful object detection algorithm was the most difficult task, and we propose some improvements below. We also believe that we could reduce size of the object classification model to speed up predictions and improve performance.

4.1 Object Detection

While our application of MR-CFAR to object detection was more successful than the reference model, its performance was not as strong as the top scoring models, and our close-to-shore detection only slightly improved upon the reference model. We found that there were many false negatives in the MR-CFAR output. They were introduced in the first, lowest resolution CFAR pass. At the 15 % resolution level, a single pixel represents an area approximately 80 m $\times$ 80 m. In the close-to-shore scenario, a ship may be between 50 meters and 200 meters from the shore. To observe a vessel in that location, a circular window around it will then include the shore, which has a greater intensity of backscattered light than water. To correct this issue, we propose that the algorithm could be altered to pass over the low-resolution images multiple times with different guard-window patterns that exclude neighboring pixels in certain direction when calculating the test statistic. As the shoreline is generally locally linear, censoring one direction ought to censor the shoreline and reduce related noise. This strategy could improve the close-to-shore detection dramatically and the vessel detection generally.

4.2 Object Classification

Our implementation of the VGG16 algorithm proved adept at classifying objects into the four classes. Compared to the baseline model, which used a Faster R-CNN algorithm implemented in PyTorch [23], our model had a significantly improved $F1$-scores for object classification and fishing vessel classification that approached the final score of the first-place finisher. The high values for $F1_V$ and $F1_F$ indicate that the VGG16 model, which is normally used on red-green-blue (RGB) images, can be easily adapted to the two-channel SAR data to provide robust, accurate

results. Furthermore, our shuffling of the provided training and validation data, the addition of false positives, and the image data generator all contributed to the model's success.

While our classification model was successful, it had two features that slowed training significantly: there were many training parameters, and the SGD optimizer is slow. The standard implementation of the VGG16 algorithm requires over 138 million trainable parameters to classify 224 pixel $\times$ 224 pixel RGB images into one thousand classes [16]. While our model reduced the number of trainable parameters to approximately 34 million, this is still a significant number that requires ample training time. Furthermore, the stochastic nature of the SGD optimizer was favorable as it allowed us to find the global minimum for our model, but it also increased the training time. It is possible that a smaller model with fewer parameters and an optimizer like Adam, would give similar results, but it would require us to tune the hyperparameters like the learning rate to find the best model, which we did not have time to fully explore.

4.3 Length Estimation

The length estimation model also performed well. While the baseline model did not include a length estimation algorithm that we can compare our results to, our model performed close to the top-scoring model, $PE_L = 0.62$ *vs.* 0.69. This strong performance indicates that a CNN can be adapted to SAR data for both image classification and length estimation, even if some additional fine-tuning of the models are needed.

5 Conclusions

The identification and classification of IUU fishing vessels is crucial for the ecological health of global fisheries and the economic prosperity and political stability of the communities that depend on them. We proposed a method to identify and classify IUU fishing vessels from SAR data that improved significantly over the baseline model. Our MR-CFAR algorithm was able to detect objects with a greater accuracy in the open ocean but fell short in the close-to-shore category. However, we proposed a method to improve the algorithm. Our object classification and length estimation CNNs were highly accurate, performing at a level comparable to the top scoring entry. With some work, we could improve our method to similar performance levels.

6 Acknowledgments

We would like to thank those that advised or supported this project, especially Ian Coffman, Chris Chang, Berk Ekmekci, Eleanor Hagerman, Blake Howald, Matt Machado, Dustin Martin, Ben Ohno, Valeria Rozenbaum, Zach Seid, Chris Smith, and Spencer Torene. This research would not have been possible without the support of Thomson Reuters Special Services, LLC.

References

[1] *XView Computer Vision Challenge Series*. https://www.diu.mil/ai-xview-challenge. Accessed: 2022-03-31.

[2] *DIUx xView 2018 Detection Challenge: Objects in Context in Overhead Imagery*. http://xviewdataset.org/. Accessed: 2022-03-31.

[3] *Computer Vision for Building Damage Assessment: Using Satellite Imagery of Natural Disasters*. https://xview2.org/. Accessed: 2022-03-31.

[4] *Global Fishing Watch*. https://globalfishingwatch.org. Accessed: 2022-03-31.

[5] David J. Agnew et al. "Estimating the Worldwide Extent of Illegal Fishing". In: *PLOS ONE* 4.2 (Feb. 2009), pp. 1–8. DOI: 10.1371/journal.pone.0004570.

[6] *xView3 Challenge: Detecting Illegal, Unreported, and Unregulated Fishing Vessels*. https://iuu.xview.us. Accessed: 2022-03-31.

[7] *Sentinel-1*. https://sentinel.esa.int/web/sentinel/missions/sentinel-1. Accessed: 2022-03-31.

[8] Franz Meyer. "Spaceborne Synthetic Aperture Radar: Principles, Data Access, and Basic Processing Techniques". In: *SAR Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation*. Ed. by Africa Ixmucane Flores-Anderson et al. National Space Science and Technology Center, 2019.

[9] *Docker*. https://www.docker.com. Accessed: 2022-03-31.

[10] *OpenCV: Open Source Computer Vision Library*. OpenCV, 2015. URL: https://github.com/opencv/opencv.

[11] Sean Gillies et al. *Rasterio: geospatial raster I/O for Python programmers*. Mapbox, 2013. URL: https://github.com/rasterio/rasterio.

[12] Maurizio di Bisceglie and Carmela Galdi. "CFAR detection of extended objects in high-resolution SAR images". In: *IEEE Transactions on Geoscience and Remote Sensing* 43.4 (2005), pp. 833–843. DOI: 10.1109/TGRS.2004.843190.

[13] Khalid El-Darymli et al. "Target detection in synthetic aperture radar imagery: a state-of-the-art survey". In: *Journal of Applied Remote Sensing* 7.1 (Mar. 18, 2013), pp. 071598–071598. DOI: 10.1117/1.jrs.7.071598.

[14] Harm Greidanus et al. "The SUMO Ship Detector Algorithm for Satellite Radar Images". In: *Remote Sensing* 9.3 (2017). ISSN: 2072-4292. DOI: 10.3390/rs9030246.

[15] L.M. Novak and S.R. Hesse. "On the performance of order-statistics CFAR detectors". In: *[1991] Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems Computers*. 1991, 835–840 vol.2. DOI: 10.1109/ACSSC.1991.186564.

[16] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556.

[17] *ImageNet: Large Scale Visual Recognition Challenge 2014*. https://image-net.org/challenges/LSVRC/2014/results. Accessed: 2022-04-01.

[18] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980.

[19] Moritz Hardt, Benjamin Recht, and Yoram Singer. *Train faster, generalize better: Stability of stochastic gradient descent*. 2015. DOI: 10.48550/ARXIV.1509.01240.

[20] Dami Choi et al. *On Empirical Comparisons of Optimizers for Deep Learning*. 2019. DOI: 10.48550/ARXIV.1910.05446.

[21] *ImageDataGenerator*. https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator. Accessed: 2022-03-31.

[22] *Amazon SageMaker Documentation*. https://docs.aws.amazon.com/sagemaker/index.html. Accessed: 2022-03-31.

[23] Timat Cambrio. *xview3-reference*. Defense Innovation Unit, 2021. URL: https://github.com/DIUx-xView/xview3-reference.

[24] Eugene Khvedchenya. *Winning Solution for xView3 Challenge*. Defense Innovation Unit, 2022. URL: https://github.com/DIUx-xView/xView3_first_place.