

**INFORMATION FUSION FOR
ROBUST DETECTION OF
SCARCE FEATURES/OBJECTS
IN HIGH RESOLUTION
ELECTRO-OPTICAL SATELLITE IMAGERY**

A Dissertation
presented to
the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by
ALAN BRUCE CANNADAY II
Dr. Curt Davis, Dissertation Supervisor

DEC 2021

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

INFORMATION FUSION FOR
ROBUST DETECTION OF
SCARCE FEATURES/OBJECTS
IN HIGH RESOLUTION
ELECTRO-OPTICAL SATELLITE IMAGERY

presented by Alan Bruce Cannaday II,
a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Curt H. Davis

Dr. Grant J. Scott

Dr. Derek T. Anderson

Dr. Joel L. Andrus

DEDICATION

To my wife, Brooke Cannaday (née McSpadden), and her sacrifice with not only raising the five children we brought with us when we moved to Columbia and two more since we arrived, but also being my biggest support and encouraging me all along the way. Also to my children, for being patient having their father home late or working on weekends because of a homework or paper deadline. To my mother, Francis Cannaday (née Barrett), and my father, Alan Cannaday, whom fostered my engineering abilities from a young age, for teaching me to work, and of whom I seek to honor daily. To Clinton “Grandpa” and Carmen “Grandma” Davis (née Shumway) who counseled and guided me with love. Finally and mostly, I’d like to acknowledge my Father in Heaven for bringing our family to Columbia, “for giving me a PhD to keep me busy”, and putting us on this great adventure.

ACKNOWLEDGMENTS

I'd like to acknowledge Dr. Curt Davis for all the “engineering discussions”, helping me keep perspective, and giving me the latitude to make mistakes. I would like to give acknowledgement to all the hard working people at the University of Missouri Center for Geospatial Intelligence (CGI) who helped with acquiring and maintaining data, expedited data processing, and just kept everything running: especially Dr. Grant Scott, Alex Lasley, Alex Hurt, Ray Chastain, David Huangul, Will Starms, and Lori Thunhorst. I'd like to acknowledge Dr. Derek Anderson, for taking on the extra class load, and the people in the Mizzou INformation and Data FUsion Laboratory (MINDFUL) as well as A. J. Maltenfort from the National Geospatial-Intelligence Agency (NGA). And finally, I wish to acknowledge my former FamilySearch colleague and friend Dr. Patrick Schone for teaching and mentoring me about data engineering and research basics.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
LIST OF TABLES	viii
LIST OF FIGURES	xiv
ABSTRACT	xix
1 INTRODUCTION	1
2 BROAD AREA SEARCH AND DETECTION OF SURFACE-TO-AIR MISSILE SITES USING SPATIAL FUSION OF COMPONENT OBJECT DETECTIONS FROM DEEP NEURAL NETWORKS	5
2.1 INTRODUCTION	6
2.2 SOURCE DATA	7
2.2.1 SAM Site Dataset	7
2.2.2 SE China AOI Tiles	8
2.2.3 Component Objects Dataset	9
2.3 DATA PROCESSING	10
2.3.1 Training Data and DNN Architecture Selection	10
2.3.2 Image Scanning and Spatial Clustering	12
2.3.3 Candidate Tile Generation & Component Object Processing	13
2.3.4 Cluster Score Normalization & Truncation	14
2.3.5 Decision-Theoretic Approach for Optimization	18
2.4 SAM SITE ONLY EXPERIMENT RESULTS	19
2.5 DECISION-LEVEL COMPONENT METRIC FUSION	19
2.5.1 Component Feature Types	21

2.5.2	Decision-Level Fusion Techniques	21
2.5.3	MLP Input Data Normalization	23
2.6	RESULTS & OBSERVATIONS	24
2.6.1	Improved Candidate <i>SAM Site</i> Rankings	24
2.6.2	Improved <i>SAM Site</i> Detection	26
2.7	CONCLUSION AND FUTURE WORK	29
3	DECISION-LEVEL FUSION OF DNN OUTPUTS FOR IMPROVING FEATURE DETECTION PERFORMANCE ON LARGE-SCALE REMOTE SENSING IMAGE DATASETS	31
3.1	INTRODUCTION	31
3.2	SOURCE DATA	33
3.2.1	Multi-Class and Parent Class Datasets	33
3.2.2	Binary Datasets	36
3.2.3	Multi-Scale Construction Site Detection Datasets	38
3.2.4	Candidate Construction Site Datasets	38
3.3	EXPERIMENTS	39
3.3.1	5-fold Cross-Validation Experiments	39
3.3.2	Multi-Scale Construction Site Detection Experiments	40
3.3.3	Component Object Image Scanning and Decision-Level Fusion	40
3.4	RESULTS	44
3.4.1	5-Fold Multi-Class and Parent Class Experimental Results	44
3.4.2	Decision-Level Fusion Results	45
3.5	CONCLUSION AND FUTURE WORK	46
4	EVALUATION OF FUZZY INTEGRAL DATA FUSION METHODS FOR RARE OBJECT DETECTION IN RGB HIGH-RESOLUTION SATELLITE IMAGERY	48

4.1	INTRODUCTION	48
4.2	DATA FUSION TECHNIQUES	50
4.2.1	Fuzzy Integral	51
4.2.2	Fuzzy Measures	52
4.3	EXPERIMENTAL OVERVIEW	59
4.3.1	DNN Model Training	60
4.3.2	Multi-DNN Architecture Fusion	60
4.3.3	Object Background Confirmation	61
4.4	DATA SOURCE	62
4.4.1	xView Scene GSD Correction	62
4.5	MULTI-DNN ARCHITECTURE FUSION EXPERIMENTS	63
4.5.1	Datasets	64
4.5.2	Experimental Design	65
4.5.3	Results	67
4.6	BACKGROUND CONFIRMATION	72
4.6.1	Datasets	75
4.6.2	Results	76
4.7	CONCLUSION AND FUTURE WORK	79
5	EVALUATION OF FUZZY INTEGRAL DATA FUSION METH- ODS FOR RARE OBJECT DETECTION IN 8-BAND MULTI- SPECTRAL HIGH-RESOLUTION SATELLITE IMAGERY . . .	81
5.1	INTRODUCTION	81
5.2	EXPERIMENTAL OVERVIEW	82
5.3	DATASETS	84
5.3.1	Multi-Spectral Band Partitioning	85

5.3.2	Bit Conversion	85
5.3.3	Updated <i>Engineering Vehicle</i> Training Samples	87
5.3.4	Validation	87
5.4	RESULTS	89
5.4.1	5-fold Experiments	89
5.4.2	<i>Engineering Vehicle</i> Detection Experiments	90
5.5	CONCLUSION AND FUTURE WORK	94
6	NEURAL LEARNING BASED BOUNDING-BOX MODEL FU- SION/ENSEMBLING FOR SCARCE OBJECT DETECTION	96
6.1	INTRODUCTION	96
6.2	SOURCE DATA	97
6.2.1	SAM-Focused Dataset	97
6.2.2	80-20 Scene Partition	99
6.2.3	Bounding-Box Detection Class Labels & Datasets	99
6.2.4	Image Scanning Class Labels & Datasets	100
6.2.5	Image Size Reductions for Training and Testing	101
6.3	BOUNDING-BOX OBJECT DETECTORS & IMAGE SCANNING + SPATIAL CLUSTERING EXPERIMENTS	104
6.3.1	Bounding-Box Detection	104
6.3.2	YOLOv5 Model Size Experiments	105
6.3.3	Bounding Boxes for Image Scanning + Spatial Clustering	106
6.3.4	Evaluation Metrics	106
6.3.5	Results	107
6.4	BOUNDING-BOX MODEL FUSION/ENSEMBLING EXPERIMENTS	112

6.4.1	Updated and Reduced Bounding-Box Ensembling Dataset	113
6.4.2	OOB Bounding-Box Model Ensembling Techniques	114
6.4.3	Bounding-Box Pairing for Neural Network Datasets	115
6.4.4	Multi-Layer Perceptrons	116
6.4.5	Pseudo-Cell State (PCS) Long-Short Term Memory (LSTM)	119
6.4.6	Bounding-Box Coordinate Computation	121
6.4.7	Minimize Expected Calibration Error	122
6.4.8	Results	123
6.5	BOUNDING-BOX FUSION/ENSEMBLING FOR THREE DETECTOR INPUTS	131
6.5.1	Additional Model & Extended Training	131
6.5.2	Expanded MLP and PCS-LSTM	131
6.5.3	Results	131
6.6	CONCLUSION AND FUTURE WORK	137
	APPENDIX	140
	A	140
A.1	Sample Counts by Initial xView Objects with Assumed 0.3 meter GSD.	140
A.2	DNN Dataset Augmentations	142
A.3	Detailed Flow Chart for Clustering Counts	144
A.4	DeepNET Complete Object Counts for Experimental Selected Scenes	145
	BIBLIOGRAPHY	149
	VITA	153

LIST OF TABLES

Table		Page
2.1	Summary of Curated Training Data	8
2.2	Summary of <i>SAM Site</i> DNN detector performance from 5-fold cross-validation testing. metrics shown are recall or True Positive Rate (<i>TPR</i>), True Negative Rate (<i>TNR</i>), Average Accuracy (<i>ACC</i>), and Area Under the ROC Curve (<i>AUC</i>).	12
2.3	NASNet 5-fold cross validation results for DNN models of <i>SAM Sites</i> and each component object, including component object DNN models with negative component data. Metrics shown are True Positive Rate (<i>TPR</i>), True Negative Rate (<i>TNR</i>), <i>F1</i> score, and Standard Deviation (<i>SD</i>).	12
2.4	Sample thresholds calculated by DTA.	18
2.5	Spatial clustering results from DNN scanning of the SE China AOI for candidate <i>SAM Sites</i> . Given values are pre-cluster counts over α -cut threshold (F^α), post-cluster counts, and average True Positive (<i>TP</i>) cluster rank.	19
2.6	Average rank of known <i>SAM Sites</i> in SE China AOI from fusing cluster scores from a single component object class with a baseline candidate <i>SAM Site</i> cluster score.	25
2.7	Average rank of known <i>SAM Sites</i> in SE China AOI from fusing cluster scores from all four component object classes with the baseline candidate <i>SAM Site</i> cluster score.	26

2.8	Experiment results with highest <i>F1</i> Scores. The first line after the header (in red) are the results for <i>SAM Site</i> detection without error reduction from spatial fusion of any component feature type(s). The highest <i>F1</i> scores were achieved by fusing multiple components using neural learning techniques (MLP or ANFIS). Also, raw detection counts (pre-clustering) showed the most separability. All top solutions achieved a relative error reduction of greater than 96%. These results would be optimal if error reduction was the primary goal. The error rate includes both false positives and false negatives.	28
2.9	Experiment results with highest <i>F1</i> scores while maintaining a <i>TPR</i> of 100%. The highest <i>F1</i> scores resulted from fusing a feature from all components with a simple MLP. Also, Cluster Count features yielded the top results. All top solutions show a reduction of relative error between 85.2 – 88.5% which is 3X the error rate shown in Table 2.8.	28
3.1	Breakdown of designated xView parent/child class relationships. The parent class name is in bold at the top of each column. The classes in the “none” column have no additional parent classification and we used as a unique class when creating the parent class datasets. Note also that each parent class has a subclass of the same name, not listed under the parent class heading.	34
3.2	Minimum and maximum MPL at 0.5m GSD required for a sample to be included for a given DNN model size and total samples available/used pre/post-capping.	35
3.3	xView sample counts belonging to each overlapping parent class for 0.5m GSD.	36
3.4	Sample counts for binary DNN models. Negative samples were selected randomly from a pool of xView features after excluding xView features in the <i>CS</i> , <i>EV</i> , and <i>TR</i> parent classes to avoid contamination. Negative sample counts are approximately 4X the positive sample count.	38
3.5	Candidate location dataset sample size for <i>CS</i> detection experiments.	39
3.6	Assessment metrics for unique <i>CS</i> samples from xView validation data to determine the best DNN model size to use for final evaluation. Top results are highlighted in green.	40

3.7	Weights used for decision-level fusion using maximum response, raw component responses over a thresholds, and component counts after local spatial clustering. Note that <i>Truck</i> models 64 and 128 were not included due to the fact that decision boundaries could not be achieved through <i>F1</i> -score optimization and were therefore removed from the final decision. This is also true for any cell in the table marked as ‘-’.	43
3.8	Average metrics of multi-class 5-fold experiments.	44
3.9	Average <i>F1</i> scores for binary DNN parent class models.	45
3.10	Assessment metrics for <i>Constructions Sites</i> (CS) detection with baseline 0.5 α -cut and multi-DNN decision-level fusion. <i>F1</i> score, error rate, and error reduction in % points and relative error reduction (red). The top result for locally clustered component counts is highlighted in green.	47
4.1	Details of FM approaches used in the experiments. The Mobius transform was not used beyond the DNN architecture experiments because the unbounded nature created a large number of false positives. Also, neither of the ChiMP approaches were used in the MS experiments because there were training issues that were not resolved within the research time constraints.	54
4.2	Sample sizes for <i>Engineering Vehicles</i> used in DNN architecture fusion experiments.	65
4.3	Scanning stride, local clustering apertures, and proximity radii used to calculate and validate candidate <i>Engineering Vehicle</i> locations.	66
4.4	Results from 5-fold experiments for <i>EV</i> object detection models using the ProxylessNAS, NASNet, and Xception model architectures for 0.3 m and 0.5 m GSD and various model sample sizes.	69
4.5	Comparative metrics for each ProxylessNAS <i>EV</i> model for sizes 32, 64, and 128 and 0.3 m and 0.5 m GSDs.	69
4.6	Validation scanning results for the pooled model sizes for all three DNN architectures. Red shows the scanning results for ProxylessNAS models that are used as a baseline for comparison with the fusion experiments.	70
4.7	Top <i>F1</i> results for multi-DNN architecture fusion experiments. Red indicates comparative baseline. A few extra methods were included for the 0.5 m GSD results.	71

4.8	Top <i>TPR</i> results for DNN architecture fusion experiments. Red shows comparative baseline. Baselines are placed within their ranked position and included with the rest of the results.	72
4.9	Sample counts used for <i>EV</i> BG experiments. Because xView object locations can be located on or slightly off the edge of a provided xView scene, pure black samples might be produced, and, if so, these were removed from the datasets and appear as slight differences in counts between different datasets.	76
4.10	5-fold experiments for <i>EV</i> background models.	76
4.11	Results for BG confirmation using single-sized BG with the same GSD, different GSD, and combining GSDs through OR and AND binary logic. A confidence value of 0.5 was used as the background confirmation threshold.	77
4.12	Top <i>F1</i> score results for BG fusion experiments. Red indicates comparative baseline. Note that non-fused results of 0.5 m GSD were also included in the ranking.	78
4.13	Top <i>TPR</i> results for BG fusion experiments. Red indicates comparative baseline. Note that non-fused results of 0.5 m GSD were also included in the ranking.	79
5.1	Details of FM approaches used in the experiments. The Mobius transform was not used beyond the DNN architecture experiments because the unbounded nature created a large number of false positives. Also, neither ChiMP approach was used in the MS experiments b/c there were training issues that were not resolved within time constraints. Note that this is an expansion from Table 4.1	83
5.2	Sample counts for <i>EV</i> object training samples after GSD correction used in 8-band training.	87
5.3	5-fold <i>EV</i> comparison of the original xView competition imagery and the in-house, bit-converted RGB results.	89
5.4	5-fold <i>EV</i> experiments for research-generated MS band partitioned models.	90
5.5	<i>TPR</i> score results for individual MS band partition experiments both with and without pooling the 256-model size. Red indicates comparative RGB baseline.	91
5.6	Top <i>F1</i> score results for MS-band fusion experiments from models of size 32, 64, and 128. Red indicates baseline result.	92

5.7	Top <i>TPR</i> results for MS-band fusion experiments for models of size 32, 64, and 128. Red indicates baseline result.	92
5.8	Top <i>F1</i> score results for MS partition fusion experiments with 256 models included. Red indicates comparative baseline.	93
5.9	Top <i>TPR</i> score results for MS partition fusion experiments with 256 models included. Red indicates comparative baseline.	94
6.1	Train/test scene counts for 80-20 partition and class densities used to create the partition. Densities are calculated by the number of objects per km ² of valid pixel (no black) scene content.	100
6.2	Class counts for 37-class experiments used in bounding-box detector experiments. An approximate 80-20 partition was achieved on a per-class basis using the partitioning method described in Section 6.2.2. Primary object classes of interest used in the partitioning schema are highlighted in blue.	102
6.3	Class counts for 11-class experiments used in bounding-box detection experiments. An approximate 80-20 partition was achieved on a per class basis using the partitioning method described in Section 6.2.2. Primary object classes of interest used in the partitioning schema are highlighted in blue.	103
6.4	Class counts for 4-class experiments used in bounding-box detection experiments. Same counts apply to separate <i>TEL</i> and <i>LP</i> datasets.	103
6.5	Training sample counts for image scanning + spatial clustering experiments.	104
6.6	Bounding-box experimental results for <i>TELS</i> for different size YOLOv5 models. Best results shaded in blue. Note that the 37-class and 11-class experiments are only finding <i>SAM TELS</i> where the 4-class and <i>TEL</i> experiments are all <i>TELS</i>	109
6.7	Bounding-box experimental results for <i>LP</i> detection for extra-large YOLOv5 models. Top portion is the different <i>LP</i> classes for the 37-class and 11-class experiments. Bottom portion is a comparison of the <i>All LP</i> classes for the 4-class and <i>LP</i> experiments and the weighted sum of the <i>SAM LP</i> classes from the 37 and 11-class experiments. Blue shading indicates the best results for each column. Yellow shading indicates that one of the <i>SAM LP</i> classes from the 37 or 11-class experiments would have been best.	110

6.8	Top results for each type of model for image scanning + spatial clustering for <i>SAM TELs</i> . Integer values in ‘Truth Box Type’ column indicate the size of box created for evaluation as opposed to using the original labeled bounding box. Results are for IoU @ 0.25.	111
6.9	Top results for each type of model for image scanning + spatial clustering for combined <i>SAM LP</i> and <i>SAM LP w/ Revetment</i> classes. Integer values in ‘Truth Box Type’ column indicate the size of box created for evaluation as opposed to using the original labeled bounding box. Results are for IoU @ 0.25.	111
6.10	Updated class counts for 11-class experiments used in bounding-box ensembling experiments. An approximate 80-20 partition was achieved on a per-class basis using the partitioning method described in Section 6.2.2. Primary object classes of interest used in the partitioning schema are shaded in blue.	113
6.11	Table of abbreviations for publicly available bounding-box ensembling methods.	115
6.12	Metric types included in MLP dataset as well as input/output size vector size. Bounding-box coordinates include top left and bottom right coordinates for each model, $[x_{1a}, y_{2a}, x_{2a}, y_{2a}]$ and $[x_{1b}, y_{2b}, x_{2b}, y_{2b}]$	119
6.13	Abbreviations for different bounding-box coordinate computation methods.	123
6.14	Results for top and selected two-detector ensembling.	124
6.15	Top five results for bounding-box ensembling along with the top results for each type of neural-network ensembling. IDX is the ranking index of all methods tested.	125
6.16	Results for bounding-box neural-learning ensembling with backend OOB ensembling. IDX is the ranking index of all methods tested.	127
6.17	Results for top and selected three-detector ensembling.	132
6.18	Results for three bounding-box detector ensembling. IDX is the ranking index of all methods tested.	135
6.19	Results for three bounding-box detector ensembling with backend OOB ensembling. IDX is the ranking index of all methods tested.	136
6.20	Summary of neural-leaning ensembling results compared to ‘Out-Of-the-Box’ ensembling and bounding-box detector outputs for two detector and the expended three detector inputs.	139

LIST OF FIGURES

Figure	Page
2.1 Example <i>SAM Site</i> with smaller-scale <i>Launch Pad</i> and <i>TEL Group</i> component objects.	7
2.2 10 geo-cells in the Southeast China area of interest. Courtesy of [19] .	8
2.3 Samples of <i>SAM Site</i> component objects used in this study.	10
2.4 Center points of DNN inference response fields post scanning (cyan lattice). These lattices were produced by scanning a 256x256 image using detectors of size (left to right): 128x128 pixels, 64x64 pixels, and 32x32 pixels. The yellow square in the top left corner represents the size of the DNN model detector relative to the image.	13
2.5 Distance-decay functions used for calculating local clustering scores. The function $\exp(-d/R)$ (blue) is used as a weight when summing raw detections within distance R . The function $-\exp(-(2R-d)/R)$ (red) is used to calculate an exponential penalty weight for raw detections outside R	17
2.6 True Positive Rate (<i>TPR</i>) or recall, Positive Predictive Value (<i>PPV</i> or precision), and <i>F1</i> score versus the threshold for the cluster count of <i>TEL</i> cluster centers within 150 m of a candidate <i>SAM Site</i> location. In this example, the value 3 was used for the final threshold as shown in Table 2.4.	18
2.7 Processing flow chart for decision-level fusion of multiple component object detections.	20

2.8	Comparison of $F1$ scores produced for candidate <i>SAM Site</i> locations from different fusion techniques. Techniques include individual component threshold from DTA as well as component fusion using an OR gate and MLP. Note that “(CN)” at the end of the feature type label in the key indicates that component negative models were used in the processing. Gaps in scores occur when the MLP was unable to train on a given feature type.	23
2.9	Comparison of True Positive Rate (TPR) produced for candidate <i>SAM Site</i> features from different fusion techniques.	23
2.10	Process flow used for improved ranking of candidate <i>SAM Sites</i>	24
2.11	$F1$ score results for DTA thresholds of original cluster scores, normalized cluster scores, cluster scores with a penalty of -1 and distance-decay penalty with $R = 150$ m.	27
3.1	Example of a <i>Surface-to-Air Missile (SAM) Site</i> (left) and <i>Construction Site</i> (right) with various constituent components highlighted. Note the contrast in the regularly distributed and permanently installed components of the SAM site and the irregular spatial distribution and highly mobile nature of the construction site components. Left image courtesy of the DigitalGlobe Foundation; right image and all other images in this paper are from the xView dataset [45].	32
3.2	Graph representation of proximity counts of xView classes within 150 meters of any other class. The closer classes are to each other, the higher count those classes have within proximity to each other. The proximity of classes belonging to parent classes EV and TR to the CS class (in yellow modularity cluster) informed our selection of the parent classes used for CS constituent object scanning and detection. Node colors indicate modularity clusters.	37
3.3	Overview of processing workflow from image scene to decision-level fusion. From left to right: 1) Broad area scanning (simulated in this study) to identify candidate <i>Construction Site</i> (CS) samples. 2) Generate inference response for candidate CS samples by passing through Construction Site detection model. 3) Scan candidate CS samples with component DNN models of different sizes (e.g. multi-scale). 4a) Pass CS inference response and component inference responses through MLP, or 4b) apply logic tree to CS inference response and extracted per-component features. 5) Final binary fused decision. Note that the final decision is not a combination of the results from the MLP and decision tree and is instead mutually exclusive, i.e. the MLP was used as an alternate approach for comparison with the decision-level fusion results.	42

4.1	Visualization of a Sugeno Integral. Dashed lines go up to the minimum between the input value, δ , and the corresponding g for the commutative subset if X . The black dot is the intersection between the two sets of data. The intersection is closest to the values as π_2 . Therefore the conservative estimation would be $g\{x_{\pi_1}, x_{\pi_2}\}$	52
4.2	Fuzzy measure lattice, g , showing edges used to “move through” the lattice while calculating the fuzzy integral in π index order. Note that the each set in X is order independent, i.e. $g\{x_1, x_2\} = g\{x_2, x_1\}$. . .	53
4.3	The two graphs illustrate the difference between linear and non-linear solutions for the XOR problem. The dotted lines in the left graph show 4 possible linear solutions for decision boundaries, but none of these decision boundaries serve the data well. The graph on the right shows a non-linear solution for the same data, where the same data becomes much more separable.	59
4.4	Scatter plot of the calculated vertical and horizontal GSDs of the xView scenes.	63
4.5	Flowchart for multi-DNN architecture fusion experiments.	65
4.6	Processing flowchart for fusion of multi-DNN architecture scanning outputs.	67
4.7	<i>EV</i> candidate locations (cyan crosses) and proximity radii (red circles) used to validate candidate locations for 32, 64, and 128 pixel (left to right) DNN models. Note that DNN models with size 32 and 128 would have been detected for this <i>EV</i> object as there are detections within the proximity radii for the respective size.	68
4.8	Flow chart showing the process of searching for <i>EV</i> s with the addition of local scene BG confirmation. The initial candidate selection as described in Section 4.5 is shown in green. The pink shows the BG sampling with inference processing and <i>EV</i> BG DNN model response fusion. The BG confirmation options shown in blue include considering BGs from multiple GSDs.	74
4.9	Example of how <i>EV</i> local background (BG) samples are created around an xView <i>EV</i> object. Magenta lines are crop lines for 128x128 pixel samples and cyan lines are crop lines for 64x64 pixel samples. The same process was used for both 0.3 m and 0.5 m GSD datasets. . . .	75
5.1	Flow chart of 8-MS band partitioning for the training and validation datasets.	84

5.2	Illustration of the 8-MS band partitions and conversion from 13+ bit pixel depth to 8-bit pixel depth.	86
5.3	Conversion of 13+ bit MS imagery (red) to the 8-bit xView competition imagery (blue). Note the overlapping blue values. The red is the many-to-one mapping used for bit conversion of each band using a piece-wise linear fit of the actual mapping. These mappings were used to convert the RGB partition and the respective bands of the other partitions.	88
6.1	Sample scenes created by mosaicking several 512x512 pixel images tiles.	98
6.2	Example of bounding-boxes with labels from the DeepNET dataset v2020.12.	101
6.3	Graphical representation of Intersection-over-Union (IoU) for squares and circles (equal sizes) at 25% overlap.	107
6.4	Diagram shows data-flow, processes, and fused bounding-box outputs for comparison of different fusion/ensembling methods.	112
6.5	DNN architectures used in the MLP experiments. Inputs varied depending on dataset type. The clamping activation function was only used when applicable. FCL is an abbreviation for Fully-Connected Layer.	118
6.6	Diagram for a common Long Short Term Memory (LSTM) cell architecture with the addition of the softmax activation function before the output.	121
6.7	Diagram of the Pseudo-Cell State Long-Short Term Memory (PCS-LSTM) neural network architecture. A two-step LSTM with input from two bounding-box detection sources (a & b) utilizing pseudo memory cell states. The first-layer input is a class confidence vector for the respective bounding box. The first-layer pseudo memory cell state is one of four bounding-box metrics: the max bounding-box dimension as a percentage of the image, the max dimension in normalized meter space (i.e. length in meters/128), or the ratio of the bounding-box dimensions in normalized image or meter space (i.e. $\max[\text{dim}]/\min[\text{dim}]$). The second LSTM layer takes the concatenated cell states ($c^{<t>}$) and output ($h^{<t>}$) from the first layer as input. The output from the second layer is then processed through a final fully-connected layer and softmax function to produce a new bounding-box confidence vector.	122

6.8	Candle-stick charts showing the $F1$ score ranges for different types bounding-box coordinate computations after neural-learning ensembling. (a) Results for using data directly from YOLOv5 and D2. (b) Results for YOLOv5 and D2 after ECE calibration. Top and bottom of the boxes are $0.5*\text{StDev}$	126
6.9	Best $F1$ score results for two detector ensemble techniques.	128
6.10	Candle-stick plots showing $F1$ score improvement ranges for different OOB ensembling techniques applied after neural-learning ensembling. Top and bottom of the boxes are $0.5*\text{StDev}$	129
6.11	Best $F1$ score results for two detector ensemble techniques with OOB ensembling on the backend.	130
6.12	Best $F1$ score results for three detector ensemble techniques with OOB ensembling on the backend.	134
6.13	Comparative summary of bounding-box ensembling for two and three detector inputs.	138
A.1	Detail of fusion method 4 described in Chapter 2 From left to right: 1) Passing candidate location through <i>Construction Site (CS)</i> detection model to obtain inference response. 2) Scan candidate <i>CS</i> samples with component DNN models of different sizes (e.g. multi-scale). 3) Local clustering followed by counting the number of cluster centers (cyan crosses) within feature radius (red circles). 4) Apply logic tree to α -cut to <i>CS</i> inference response and cluster counts. 5) Final binary fused decisions.	144

ABSTRACT

We conducted research to develop and test methods to improve the detection of scarce objects in high-resolution electro-optical satellite imagery. We demonstrated improvements through various forms of information and data fusion that included heuristic analyses, fuzzy integrals, and neural learning. Scarce objects of interest included *Surface-to-Air Missile (SAM) Sites*, *SAM Launch Pads (SAM LP)*, *SAM Transporter Erectile Launchers (SAM TELs)*, *Construction Sites*, and *Engineering Vehicles*.

We demonstrated improved detection and reduced error in broad area search for *SAM Sites* in Southeast China by fusing the larger feature with the detection of smaller, multi-scale elements/components (i.e. *SAM TELs & LPs*) within the larger feature using heuristics and neural learning. We expanded these techniques to demonstrate improved detection of *Construction Sites*. We next demonstrated the improved detection of small, scarce objects (i.e. *Engineering Vehicles*) by fusing the outputs from multiple deep CNNs of various architectures and sizes through multi-layer perceptrons and fuzzy integrals. Major improvements were then achieved by leveraging existing CNN models designed for 3-band (RGB) models to train and process 8-band multi-spectral imagery partitioned into a set of three 3-band images and then fusing the results using fuzzy integrals.

We finally demonstrated that bounding-box object detection for *SAM TELs* could be improved by ensembling/fusing bounding-box results from multiple detectors using a novel Pseudo-Cell State Long-Short Term Memory (PCS-LSTM) neural network developed in this research. The PCS-LSTM is a two-layer, non-serial LSTM neural network architecture that utilizes bounding-box context to act as a first-layer quasi/pseudo cell memory.

Chapter 1

INTRODUCTION

Deep Neural Networks (DNN) have shown through extensive experimental validation to deliver outstanding performance for feature/object detection/recognition in a variety of benchmark high-resolution Electro-Optical (EO) remote sensing image datasets. For example, the overall detection accuracy for 34 different feature/object classes, excluding land use/land cover classes, was 98.7% in recent published studies [1]-[3] that utilized four different benchmark datasets [4]-[7]. Methods such as You Only Look Once (YOLO) [8], Region-based Convolutional Neural Network (RCNN) [9], and derivations thereof [10]-[15] have all shown promising results for a variety of object-detection applications in remote sensing imagery.

The demonstrated ability of DNNs to automatically detect a wide variety of man-made objects with very high accuracy has tremendous potential to assist human analysts in labor-intensive visual searches for objects of interest in high-resolution satellite imagery over large areas of the Earth's surface. However, the vast majority of published studies for DNN object detection in remote sensing imagery have focused on development of new deep learning algorithms/methods and/or comparative testing/evaluation of these methods on benchmark datasets (both public and private).

As noted by Xin *et al.* [16], comparatively fewer studies have attempted to apply promising DNN methods to demonstrate efficacy and/or further develop these new

methods via applications to large-scale or broad area remote sensing image datasets, e.g. [17]-[19]. Since “large-scale” or “broad area” are subjective descriptors, here we define these to be applications where the algorithm is applied to validation image datasets, i.e. excluding training data, covering a land area greater than 1,000 km².

Further, even DNN detectors that demonstrate exceptionally high accuracy (e.g. 99%) on benchmark testing datasets will still generate a tremendous number of errors when applied to large-scale/broad area remote-sensing image datasets. For example, a DNN detector with 99% average accuracy, chip size of 128 x 128 pixels, and a chip scan overlap of 50% will generate 88,000 errors when applied to a 0.5 m GSD image dataset covering an area of interest (AOI) of 10,000 km² (e.g. 1° x 1° cell).

Finally, DNNs trained to detect scarce or rare objects typically have much lower detection accuracies (e.g. <80%) such that millions of errors can be generated over modest AOI sizes (e.g. ~1° x 1° cell). If DNN detection results are intended to be reviewed by human analysts in machine-assisted analytic workflows, then large numbers of detection errors can quickly lead to “error fatigue” and a corresponding negative end-user perception of a machine-assisted analytic workflows. Thus, it is important to develop methods to reduce error rates resulting from application of DNN detectors to large-scale remote-sensing image datasets to improve machine-assisted analytic workflows.

The overall objective of this doctoral research is to develop, test, refine, and then combine/integrate a variety of decision-level (i.e. post-detection) fusion methods by aggregating detections in both space and time from a variety of different DNNs to reduce errors, increase confidence, and ultimately improve human analytic performance in large-scale remote sensing image applications (e.g. [19]).

In this research we have developed and demonstrated a variety of fusion techniques through a series of conference papers [20] [21] [22], a journal paper [28], unpublished results, and ongoing research. As part of Chapter 2, we first demonstrate

improved ranking for detections of a large feature by fusing CNN responses of small objects or components found within the larger feature. This was published at IEEE IGARSS 2019 [20] and demonstrated improved discrimination and ranking of *Surface-to-Air Missile (SAM) Site* features in Southeast China using both *Transporter Erector Launcher (TEL)* and *Launch Pad* component object classes.

Chapter 3 is based on results presented in a conference paper at IEEE BIGDATA 2019 [21]. This research extended the approach in Chapter 2 and demonstrated a significant reduction in false detections for world wide *Construction Site* features, using *Engineering Vehicles* and *Trucks* as component objects to confirm/deny candidate *Construction Site* detections. In [21], we also introduced additional component attributes, e.g. component count and maximum response, and more advanced decision-level fusion techniques: decision trees and Multi-Layer Perceptrons (MLPs). The IGARSS 2019 research was then extended and published in an IEEE JSTARS journal paper [28] in 2020 and this constitutes the remainder of Chapter 2. Here we developed two additional concepts from [21] to improve the detection and retrieval of *SAM Site* features. Over 200 experiments with permutations of component object type, component object attribute, and fusion techniques were conducted and the results demonstrated significant error reduction while optimizing the F1-score or True Positive Rate (TPR).

The research presented in Chapter 4 and Chapter 5 is primarily derived from a conference paper presented at IEEE BIGDATA 2020 [22], but also includes additional experiments that were not included because of IEEE conference paper length restrictions. In Chapter 4, we demonstrate that fusing detections from multiple CNN architectures using fuzzy integrals can improve scarce object detection, i.e. *Engineering Vehicles*. We next demonstrate that scarce object detection can be improved by using the surroundings context or background of a candidate *Engineering Vehicle* to confirm/deny the detection. In Chapter 5, we demonstrate further improvements

in the scarce object detection by partitioning 8-band Multi-Spectral (MS) images into three 3-band images that are separately scanned and then the resulting CNN detections are fused using fuzzy integral techniques.

Finally, in Chapter 6 we present results comparing image chip scanning + detection clustering technique with state-of-the-art bounding-box object detection neural architectures. We further test known bounding box ensembling techniques, such as non-maximal suppression [23] [24], non-maximal weighting [25] [26], and weight box fusion [27], against using a multi-layer perceptron with bounding-box metrics used as supplemental features. We also introduce the Pseudo Cell State Long Short Term Memory (PCS-LSTM) as a means of improving confidence vectors using box metrics as pseudo/quasi cell states.

Chapter 2

BROAD AREA SEARCH AND DETECTION OF SURFACE-TO-AIR MISSILE SITES USING SPATIAL FUSION OF COMPONENT OBJECT DETECTIONS FROM DEEP NEURAL NETWORKS

This chapter is taken from work that was originally published at the IEEE Geoscience and Remote Sensing Society Conference in 2019 (IGARSS 2019) [20] in Yokohama, Japan and later expanded for the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS) special issue [28] associated with the IGARSS 2019 Conference. Co-authors for the original conference paper were Curt H. Davis and Grant J. Scott, whereas Blake Ruprecht and Derek T. Anderson were added for the subsequent JSTARS publication.

Further because of the year long gap between the IGARSS 2019 and JSTARS publications, ideas and concepts presented in Chapter 3 that were first published in Dec 2019 at the IEEE BIGDATA 2019 Conference [21] were subsequently utilized in the research presented in JSTARS journal paper and are therefore referenced in this chapter.

2.1 INTRODUCTION

This research builds upon work done by Marcum *et al.* [19] where broad area search and detection of *SAM Sites* (Fig. 2.1) was demonstrated over a 10 geo-cell ($\sim 90,000$ km²) study Area of Interest along the SE coast of China (SE China AOI). Key results from this previous study were:

1. A machine-assisted approach was used to reduce the original AOI search area by $660X$ to only ~ 135 km².
2. The average machine-assisted search time for ~ 2100 candidate *SAM Site* locations was ~ 42 minutes which was $81X$ faster than a traditional human visual search.

While Marcum *et al.* used a single binary DNN detector to locate candidate *SAM Sites*, here we demonstrate how Deep Neural Network (DNN) detections of multiple constitutive or component objects that are part of the larger, more complex, and encompassing feature can be spatially fused to improve the search, detection, and retrieval (ranking) of the larger complex feature. First, scores computed from a spatial clustering algorithm are normalized to a reference space so that they are independent of image resolution and DNN input chip size. Then, multi-scale DNN detections from various component objects are fused to improve the detection and retrieval of DNN detections of the larger complex feature. We demonstrate the utility of this approach for broad area search and detection of *Surface-to-Air Missile (SAM) sites* that have a very low occurrence rate (only 16 sites) over a $\sim 90,000$ km² study area in SE China. The results demonstrate that spatial fusion of multi-scale component-object DNN detections can reduce the detection error rate of *SAM Sites* by $>85\%$ while still maintaining a 100% recall. The novel spatial fusion approach demonstrated here can be easily extended to a wide variety of other challenging object search and detection problems in large-scale remote sensing image datasets.

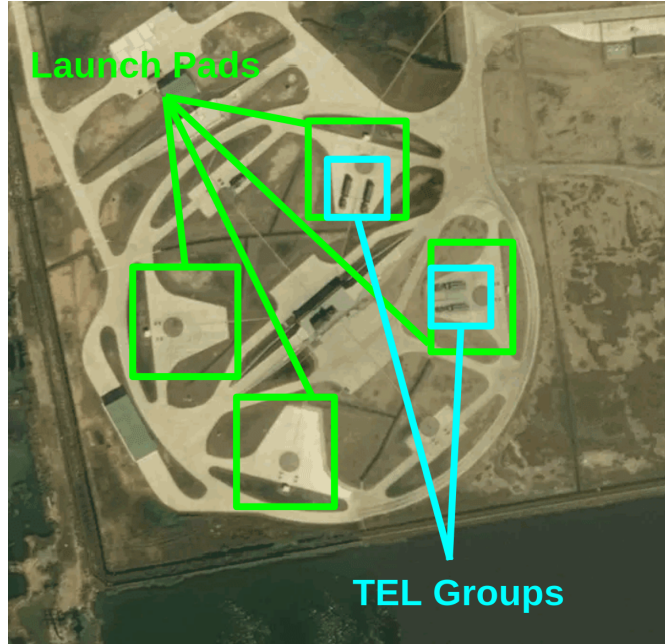


Fig. 2.1: Example *SAM Site* with smaller-scale *Launch Pad* and *TEL Group* component objects.

2.2 SOURCE DATA

2.2.1 SAM Site Dataset

The SE China AOI has 16 known *SAM Sites* which includes 2 newer *SAM Sites* found in the previous study [19]. In addition, there are and 101 other known *SAM Sites* located in China, outside of the SE China AOI (XAOI). Only the 101 XAOI sites were used for DNN training to ensure blind testing against the 16 known *SAM Sites* within the SE China AOI. As in [19], negative training chip samples were selected using a 5-km offset in the four cardinal directions (i.e. N/S/E/W) for each XAOI *SAM Site*. Additionally, while [19] used a 227x227 pixel chip size a nominal Ground Sampling Distance (GSD) of 1.0 m to train a ResNet-101 [29] DNN, here we used a 299x299 pixel chip size (at the same GSD) for DNN training which better matches the typical dimensions of a Chinese *SAM Site*. All image chip training samples were derived from a DigitalGlobe (now Maxar) worldwide satellite image basemap.

Table 2.1: Summary of Curated Training Data

Object Class	<i>SAM Sites</i>	<i>Launch Pads</i>	<i>Missiles</i>	<i>TELS</i>	<i>TEL Groups</i>
TP	101	3910 ¹	1976	2733	1179
TN	404	3696	2624	2272	1054
Combo TP	n/a	9798 ²	as above	as above	as above
Combo TN	n/a	8512	6530	10,078	5762

1: Empty

2: Includes those with co-located *Missiles*, *TELS*, and *TEL Groups*

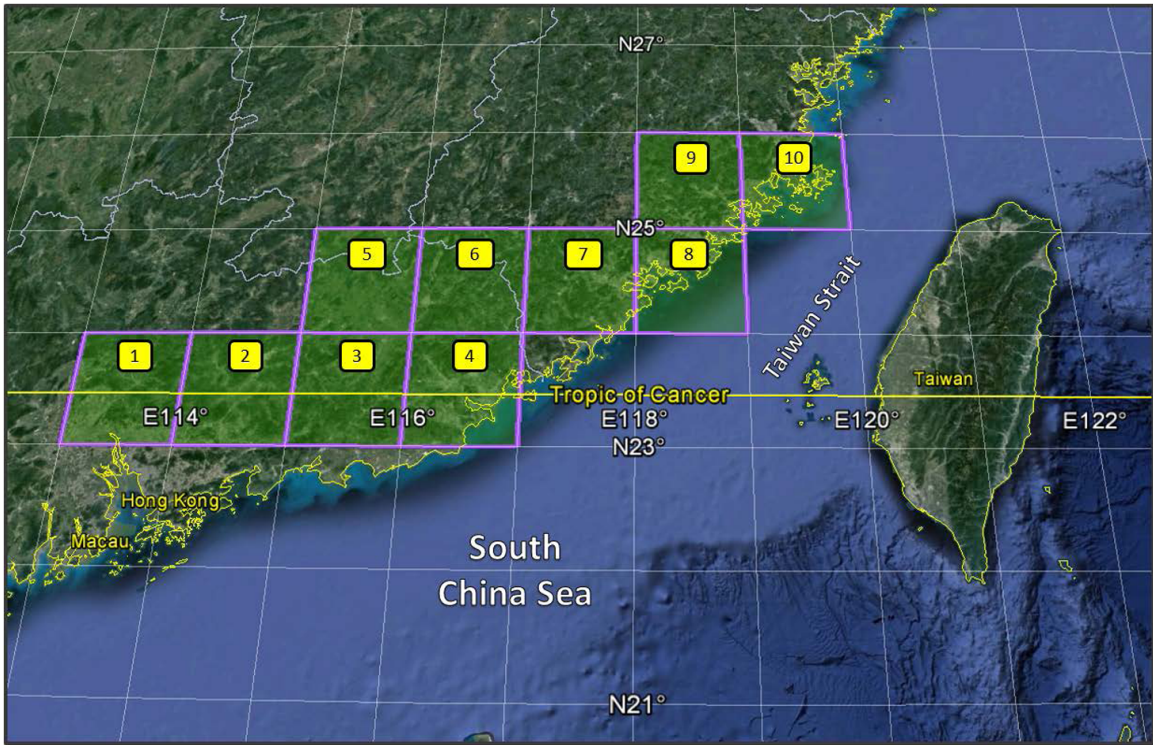


Fig. 2.2: 10 geo-cells in the Southeast China area of interest. Courtesy of [19]

2.2.2 SE China AOI Tiles

As in [19], images used in the broad area search for *SAM Sites* in the SE China AOI were comprised of ~66K 1280x1280 pixel tiles at 1.0 m GSD with 10% overlap between tiles covering almost the entirety of the SE China AOI (Fig 2.2). Some tiles were excluded from image scanning if they were 100% over the water.

2.2.3 Component Objects Dataset

We developed binary DNN detectors for four different *SAM Site* component objects: *Launch Pads*, *Missiles*, *Transporter Erector Launchers (TEs)*, and *TEL Groups* (two or more \sim co-located *TEs*) (Fig. 2.3). Component object binary DNN detectors were trained using curated data at 0.5 m GSD from China *SAM Sites* outside the AOI. We first created negative training samples for each component object using nearby image chips (similar land cover context), but outside the known spatial extent of the *SAM Sites*. This produced an \sim 1:1 ratio of negative to positive component object training samples (Table 2.1).

In addition, we created a second training dataset using all four components objects to train a combined *Launch Pad* detector (empty and non-empty) knowing that the other component objects (e.g *Missiles*, *TEs*, etc.) are generally co-located with *Launch Pads*. We then developed a second set of component object detectors for the *Missile*, *TEL*, and *TEL Group* object classes by combining negative training data from the other component objects and then randomly paring down the data to produce a \sim 4:1 ratio of negative to positive samples (Table 2.1). For the *Missile* component, samples from empty *Launch Pads*, *TEL*, and *TEL Group* and their negatives were added. However, only samples from empty *Launch Pads* and *Missiles* were added to the negatives for *TEs* and *TEL Groups* to reduce confusion between these two components object classes.

Different chip sizes were used for the training samples based on known object sizes. A 128x128 pixel chip size was used for detecting both empty and combined *Launch Pads* and *TEL Groups*. While a 64x64 pixel chip size was used for *Missiles* and *TEs*. Counts for all training data are provided in Table 2.1 and these only include component object samples outside the SE China AOI to ensure subsequent blind image scanning. All image chip DNN training samples were derived from a DigitalGlobe (now Maxar) worldwide satellite image basemap at a nominal GSD of

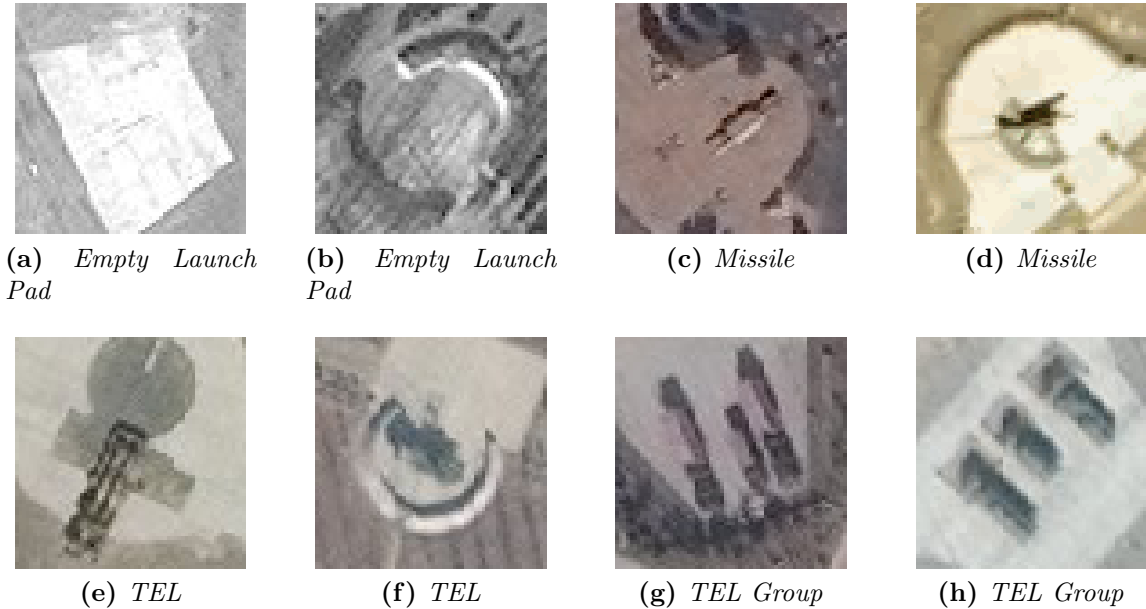


Fig. 2.3: Samples of *SAM Site* component objects used in this study.

0.5 m.

2.3 DATA PROCESSING

2.3.1 Training Data and DNN Architecture Selection

Augmentation strategies from [19] were used to train the *SAM Site* DNN and all component object DNNs to improve detector performance. A 144X augmentation was used for all 5-fold validation experiments which included a vertical flip (2X) and rotating each sample for $5^\circ \in [0^\circ, 360^\circ)$ (72X). While a 9504X augmentation was used for the final *SAM Site* DNN used for scanning the SE China AOI by including a combination of brightness adjustment, contrast adjustment, and image “jitter” (see explanation in Appendix A.2). To save computing time, augmentations were reduced for training the component object DNNs due to the much larger sample sizes. These changes included using RGB samples only, reducing the number of rotations, using a single jitter distance, and removing the contrast augmentation. Most of the final component object DNNs were trained with 648X augmentations, except the combined

Launch Pad DNN used a 216X augmentation.

From [31], a k -fold experiment is when a dataset is split into k partitions where each partition is held back once as a test dataset and the remaining partitions are used to train the DNN model. We used $k = 5$ partitions for this research where 20% of training data was held back and used as a blind test while the remaining 80% was used to train a DNN model. This was repeated k times such that all dataset samples are used once for the blind validation. After training, the held back partition is processed using the trained DNN model, thus providing blind test results. This k -fold cross validation is done to provide an initial assessment of the trained model’s performance on a limited/restricted set of data.

We completed 5-fold cross-validation experiments for several modern DNN architectures to evaluate their performance for *SAM Site* detection. The modern DNN architectures we evaluated were NASNet [30], Xception [32], ProxylessNAS [33], and all seven EfficientNet [34] models. *SAM Site* detection results from these modern DNNs are compared to the ResNet-101 DNN results published in the Marcum et al. [19] study (Table 2.2).

The results in Table 2.2 show that the NASNet DNN outperformed all the other DNNs for *SAM Site* detection as measured by recall or True Positive Rate (*TPR*), Average Accuracy (*ACC*), and Area Under the ROC Curve (*AUC*). In addition to training the *SAM Site* DNN detector, the NASNet DNN was used for training all component DNN detectors used throughout the rest of this study. Training for all the NASNet DNNs utilized transfer learning from ImageNet [35], Adam [36] for optimization, and cross entropy for the objective function.

5-fold cross validation experiments were performed for all training datasets in Table 2.1. The results provided in Table 2.3 show an average *F1* score $> 99\%$ for the baseline dataset and $> 98\%$ for the dataset with component negatives. The decrease in *F1* score for the DNNs with component negatives was anticipated given

Table 2.2: Summary of *SAM Site* DNN detector performance from 5-fold cross-validation testing. metrics shown are recall or True Positive Rate (*TPR*), True Negative Rate (*TNR*), Average Accuracy (*ACC*), and Area Under the ROC Curve (*AUC*).

DNN	TPR (%)	TNR (%)	ACC (%)	AUC (%)
<i>ResNet-101</i>	94.1%	98.8%	96.4%	99.4%
<i>Xception</i>	98.0%	98.3%	98.1%	99.9%
<i>NASNet</i>	99.0%	99.8%	99.4%	99.995%
<i>ProxylessNAS</i>	95.0%	100.0%	97.5%	99.2%
<i>EfficientNet-B4</i> ¹	91.1%	99.8%	95.4%	99.8%

1: Only results from the EfficientNet model with the highest AUC are shown.

Table 2.3: NASNet 5-fold cross validation results for DNN models of *SAM Sites* and each component object, including component object DNN models with negative component data. Metrics shown are True Positive Rate (*TPR*), True Negative Rate (*TNR*), *F1* score, and Standard Deviation (*SD*).

Object Class	TPR (%)	TNR (%)	<i>F1</i> (%)	<i>SD</i>
<i>SAM Site</i>	99.00	99.75	99.39	1.06
<i>Empty LPs</i>	99.80	99.70	99.65	0.2
<i>Combo LPs</i>	99.74	99.74	99.74	0.15
<i>Missiles</i>	99.8	99.46	99.63	0.24
<i>TELS</i>	99.72	99.38	99.55	0.32
<i>TEL Groups</i>	99.41	99.43	99.42	0.21
Including Component Negatives				
<i>Missiles</i>	97.42	99.66	98.52	0.72
<i>TELS</i>	97.51	99.37	98.42	0.5
<i>TEL Groups</i>	96.78	99.6	98.15	1.1

the inclusion of objects in the negative training data that were visually similar to the component object that a given DNN was trained to detect.

2.3.2 Image Scanning and Spatial Clustering

The SE China AOI tiles were individually scanned using a “sliding window” technique (Fig 2.4) with 25% stride (75% overlap) for a window size equal to the model size of 299x299 pixels, i.e. using a stride of 75 pixels. A chip of 299x299 pixels centered about

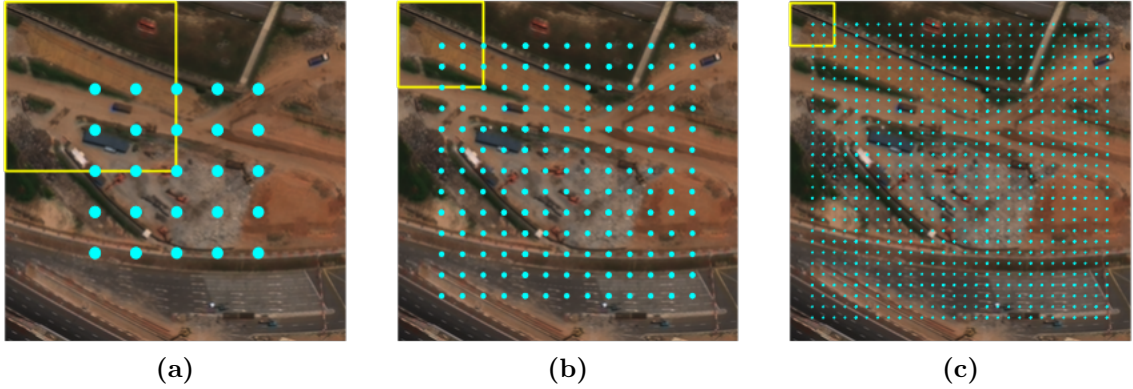


Fig. 2.4: Center points of DNN inference response fields post scanning (cyan lattice). These lattices were produced by scanning a 256x256 image using detectors of size (left to right): 128x128 pixels, 64x64 pixels, and 32x32 pixels. The yellow square in the top left corner represents the size of the DNN model detector relative to the image.

each lattice point is cropped out of the tile to create a total of $\sim 19.7\text{M}$ image chips that were then input to the trained NASNet DNN. This produced a raw detection or inference response field, F , of softmax outputs from the DNN.

Next a thresholding or α -cut of F is performed at $\alpha = 0.9$ to produce F^α , to greatly reduced the size of the detection response field. F^α is then used to produce an amplified spatial detection field, δ , (see Section 2.3.4.1). The δ is used to weight a spatial clustering of F^α to produce mode clusters, F' , using an aperture radius, R (see [19]), of 300 m . Cluster locations were then rank-ordered by summing the scores of all detections within a mode cluster to generate an initial set of “candidate” *SAM Sites*.

2.3.3 Candidate Tile Generation & Component Object Processing

New 1280x1280 pixel tiles at 0.5 m GSD, centered on each candidate *SAM Site*’s cluster location were generated and used for all component object DNN scans. Similar to the F produced for *SAM Sites*, component response fields were also spatially clustered to generate locations and cluster scores for each component object. After applying an

α -cut of $\alpha = 0.99$ to generate distinct cluster locations for a given component object relative to neighboring same class components present at each candidate *SAM Site*, an aperture radius of $R = 32$ m, approximately half the typical distance between *SAM Site* launch pads in China, was used for clustering the post α -cut detections. In this study we simply used *a priori* knowledge for our selection of R . However, we recognize that for other objects and/or applications the appropriate selection of R may need to be incorporated in the technical approach, as it can be sensitive to scanning stride and target object co-location separation (e.g. vehicles parked next to each other).

Likewise, in order to determine thresholds used for the Decision-Theoretic Approach (DTA) described in Section 2.3.5, a training set of 1280×1280 pixel pseudo-candidate tiles at 0.5 m GSD and centered about the known *SAM Sites* outside the SE China AOI were generated along with corresponding offset tile negatives. The same scans and processes performed for candidate tiles within the SE China AOI (described above), were also used for the pseudo-candidate training dataset.

2.3.4 Cluster Score Normalization & Truncation

Cluster scores from one object class to another are not necessarily comparable since they can result from objects with different physical sizes, corresponding R values, and scanning strides. In addition, results generated from image tile scans with different DNN input chip size and/or GSD will have a variable spatial density. Since we wish to spatially fuse, and potentially weight, the output from various component DNN detectors, the cluster scores must be normalized to bring both the *SAM Site* and component detection clusters into a common reference space. Here we use raw detection field density, i.e. the number of raw detections per unit area, as the means to achieve a common reference space prior to spatially fusing the cluster scores from the candidate *SAM Sites* and their associated component detection clusters.

2.3.4.1 Normalization for a Single Detection Location

The amplified spatial detection field, δ , contains an intersected volume, δ_n , for each raw detection, n , in F^α . δ_n is calculated as the weighted sum of scores of each n with its neighboring raw detections, p . The weight is determined by the distance-decay function $s(p) = \exp(-d/R)$, where $d = \text{haversine}(p, n) < R$ and is 0 otherwise. An approximate maximum intersection volume for a single raw detection can be calculated by integrating the truncated distance-decay function around a raw detection location. As mentioned above, R and d are normalized using raw detection field density. Let $R' = R/\text{stride}$ and $d' = d/\text{stride}$, where stride is the image chip's scanning stride distance in meters, so that $s'(p) = \exp(-d'/R')$. Let s represent the height or the dimension of magnitude for F , then the approximate max intersection volume for a single raw detection can then be calculated as:

$$\begin{aligned}
 n_{\text{volume}} &= \pi \cdot (R')^2 \left(\left(\int_{1/e}^1 \log^2(1/s) ds \right) + (1/e) \right) \\
 &= \pi \cdot (R')^2 ((2 - 5/e) + (1/e)) \\
 &= \pi \cdot (R')^2 (2 - 4/e)
 \end{aligned} \tag{2.1}$$

2.3.4.2 Cluster Score Weighting and Truncation

The cluster score, C_{score} , should also be limited for normalization. In the previous algorithm from [19], the number of raw detections in a cluster, C , was virtually unbounded. As a result, raw detections that were a large distance away from the cluster location can potentially contribute to C_{score} . We have often observed that these far away detections are commonly false positives. In order to minimize their impact, it can be beneficial to weight each raw detection within a cluster based on the raw detection distance relative to the cluster center. Alternately, truncation can be implemented by assigning each raw detection within a Haversine distance R a weight of 1 and outside of R a weight of 0. The C_{score} is then a weighted sum of

Procedure 1 Object Detection Cluster Ranking with Normalized Scores and Optional Penalty

Input: α -cut F^α , Mode-Cluster F'
Output: Ranked Clusters C_i , where $C_i < C_{i+1}$
begin
 $i := 0$
 while $F' = \emptyset$ **do**
 $p := pop(F')$
 $C_i := p$ // Init. C_i with chip, p
 $N^\alpha := NN(p, F^\alpha, R)$
 $N := NN(p, F', R)$
 for all $n \in N(p)$ **do**
 $C_i := \{C_i, n\}$ $F'.remove(n)$
 if $n \subset N^\alpha(p)$ **then**
 $n_{weight} = 1$
 else
 $n_{weight} = \text{penalty} // 0$ if no penalty
 end if
 end for
 $C_i.score = (C_{norm})^{-1} \cdot \sum_{n \in C_i} n_{weight} \cdot \delta_n$
 $i++$
 end while
 $\{C_i\} := \text{sort } C_{V_i}$ by score, descending
end

the raw detection inference scores with their respective weights (Procedure 1). In Section 2.3.4.4, we discuss the possibility of applying a negative penalty weight to all raw detections with a Haversine distance greater than R .

2.3.4.3 Approximate Max Cluster Score

The number of raw detections within R can be seen as a Gauss Circle Problem. Thus, the max number of raw detections within the aperture area surrounding a cluster location can be approximated in terms of raw detection field density as:

$$n_{max.p} = \pi \cdot (R')^2 \quad (2.2)$$

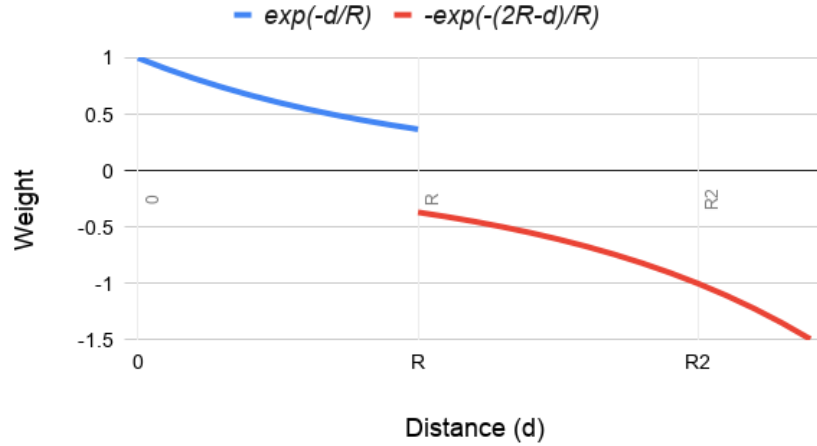


Fig. 2.5: Distance-decay functions used for calculating local clustering scores. The function $\exp(-d/R)$ (blue) is used as a weight when summing raw detections within distance R . The function $-\exp(-(2R-d)/R)$ (red) is used to calculate an exponential penalty weight for raw detections outside R .

Using equations (2.1) and (2.2), a normalizing cluster factor can be calculated as:

$$\begin{aligned}
 C_{norm} &= n_{volume} \cdot n_{max-p} \\
 &= \pi^2 \cdot (R')^4 (2 - 4/e)
 \end{aligned}
 \tag{2.3}$$

2.3.4.4 Over-Detection Penalty

In previous work we have observed FP hotspots, i.e. large numbers of spatially co-occurring false positive detections. In order to mitigate this potential problem, a penalty can be applied when computing C_{score} . As mentioned in Section 2.3.4.2, instead of using a weight of 0 when $d > R$, a negative weight can be applied. We explored two types of penalty assignments. The first used a flat weight of -1. The second is similar to the distance-decay function, however the sign was changed to negative and increases in value exponentially as d increases (Fig. 2.5). The penalty is calculated using the following formula: $s(p) = -\exp(-(2R - d)/R)$.

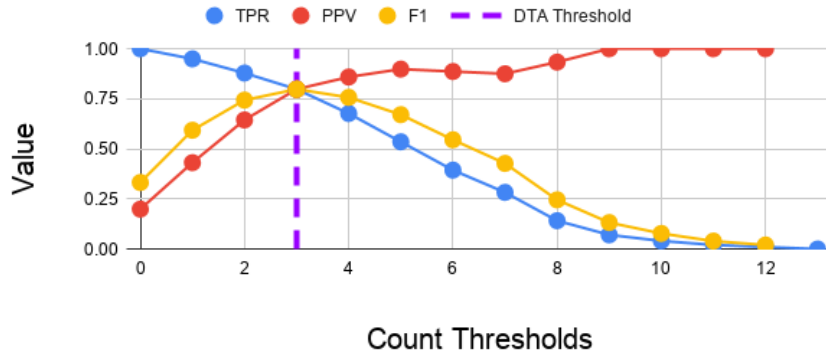


Fig. 2.6: True Positive Rate (TPR) or recall, Positive Predictive Value (PPV or precision), and $F1$ score versus the threshold for the cluster count of TEL cluster centers within 150 m of a candidate SAM Site location. In this example, the value 3 was used for the final threshold as shown in Table 2.4.

Table 2.4: Sample thresholds calculated by DTA.

<i>Feature Type</i>	<i>Empty LPs</i>	<i>Combo LPs</i>	<i>Missiles</i>	<i>TELs</i>	<i>TEL Groups</i>
Cluster Count	2	1	1	3	1
Raw Count	5	4	4	15	1
Raw Max	1.00000	0.99954	1.00000	1.00000	0.58236
Including Component Negatives					
Cluster Count	n/a	n/a	1	1	1
Raw Count	n/a	n/a	2	2	1
Raw Max	n/a	n/a	0.99989	0.98450	0.60315

2.3.5 Decision-Theoretic Approach for Optimization

In order to make discrete decisions, we used Decision-Theoretic Approach (DTA) [37] advocated by Lewis [38] that computes thresholds based on the optimal prediction of a model to obtain the highest expected F -measure. In this study, decision thresholds were selected based on the optimization of the $F1$ score from features extracted from the pseudo-candidate training dataset (see Fig. 2.6). Optimal $F1$ score thresholds were determined through empirical analysis and selected examples are provided in Table 2.4.

Table 2.5: Spatial clustering results from DNN scanning of the SE China AOI for candidate *SAM Sites*. Given values are pre-cluster counts over α -cut threshold (F^α), post-cluster counts, and average True Positive (*TP*) cluster rank.

DNN Architecture & Post-Processing	F^α Count	C Count	AVG <i>TP</i> Cluster Rank
ResNet-101 [19]	93,000	2100	181.9
NASNet	2079	354	62.8
NASNet w/ norm	2079	354	62.8
NASNet w/ norm and penalty	2079	354	62.8

2.4 SAM SITE ONLY EXPERIMENT RESULTS

NASNet significantly outperformed ResNet-101 for scanning the SE China AOI for *SAM Sites* (Table 2.5). This is consistent with the cross-validation results given in Table 2.2. NASNet had $\sim 44X$ fewer *SAM Site* candidate locations after the 0.9 α -cut (Section 2.3.2). Further, while both DNNs correctly located all 16 known *SAM Sites* (e.g. *TPs*) in the SE China AOI, NASNet had $6X$ fewer candidates compared to ResNet-101 while the average *TP* cluster rank (Table 2.5) was also $\sim 3X$ lower.

2.5 DECISION-LEVEL COMPONENT METRIC FUSION

This section describes the feature selection and fusion techniques used to reduce the number of candidate *SAM Sites* that could then be presented for human review in machine-assisted analytic workflows. An overview of the processing flow is provided in Fig. 2.7. Note that most of the fusion techniques, save summing the normalized component DNN responses, were published in the JSTARS paper and not the IGARSS 2019 paper and were borrowed from the research in Chapter 3 which was published at IEEE BIGDATA [21] in December 2019.

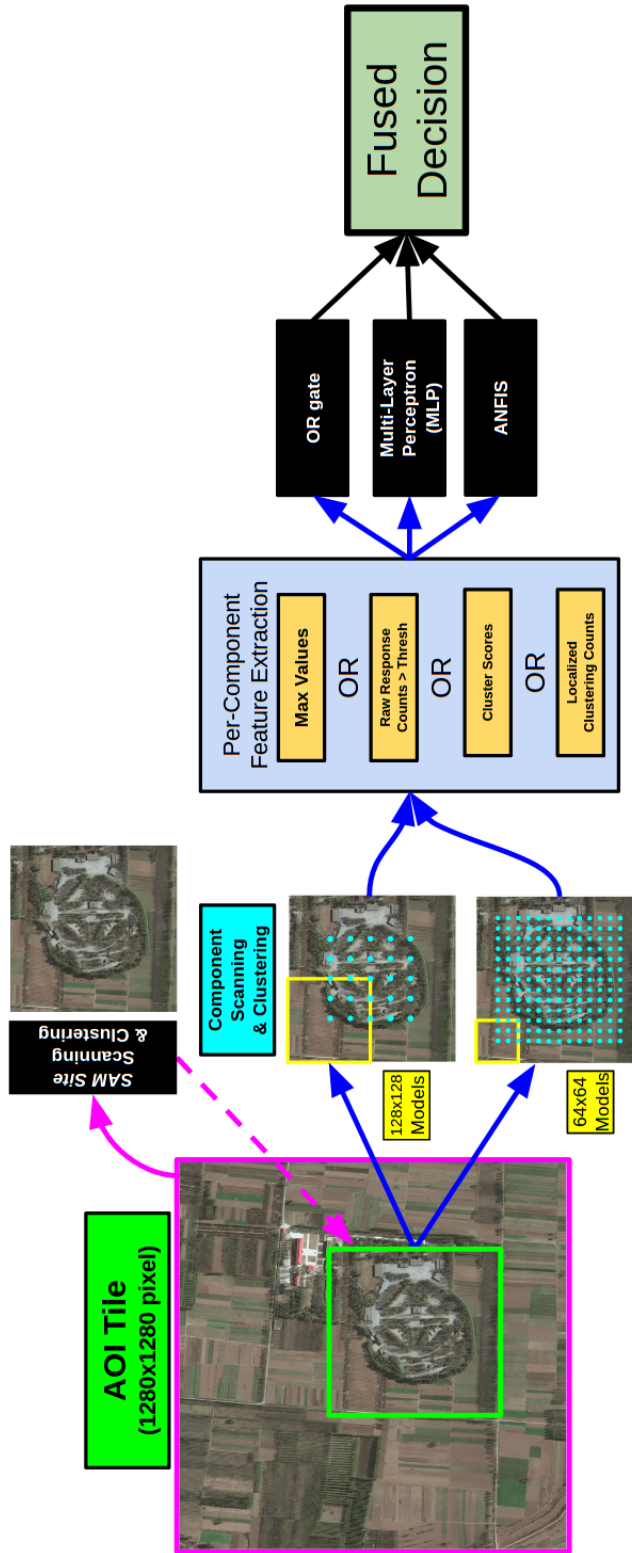


Fig. 2.7: Processing flow chart for decision-level fusion of multiple component object detections.

2.5.1 Component Feature Types

Five different feature types were used in Chapter 3 for decision-level fusion of component objects to improve the final detection of *Construction Sites*. Here we tested feature types that used the DTA optimization of the *F1 score* and represent the first three feature types listed below. In addition, we used the normalized cluster scores from the spatial clustering as an additional feature type. To maintain consistency between techniques employed in this study, only responses (raw or cluster centers) within a 150 m radius of the candidate *SAM Site* location were used. The feature types that were evaluated were:

1. Sum of normalized cluster scores for each component, both unweighted and expert weighted.
2. Maximum raw inference detection response (confidence value) for each component.
3. Count of raw inference detections for each component retained within the reduced field (F^α).
4. Count of clusters produced for each component.

2.5.2 Decision-Level Fusion Techniques

Baseline results for the candidate *SAM Site* locations were first computed using only the spatial cluster outputs of the NASNet *SAM Site* detector. We then tested how each individual component would perform using the various feature types. *SAM Site* cluster scores were excluded because the pseudo-candidate training dataset was NOT generated through scanning and clustering. Consequently, some of the pseudo-candidates would have no cluster within a sufficient radius of the *SAM Site* center location.

Three data fusion techniques were tested:

1. **Decision Tree:** The decision tree simplified is related to Chapter 3 to a digital logic OR gate with the DTA decisions as binary inputs.
2. **Multi-Layer Perceptron (MLP):** A feature vector was created for each candidate *SAM Site* location and used as input for training and validation. The MLP architecture consisted of two fully connected hidden layers of 100 nodes. We also tested normalization and feature bounds before use as input based on the thresholds from DTA optimization (Section 2.3.5).
3. **ANFIS:** A first order Takagi-Sugeno-Kang (TSK) adaptive neuro-fuzzy inference system (ANFIS) [39] [40] [41] was utilized. The goal was to explore a neural encoding and subsequent optimization of expert knowledge input. Specifically, five IF-THEN rules were used whose IF components (aka rule firing strengths) were derived using the expert knowledge from the Decision Tree in 1) above. The consequent (i.e., ELSE) parameters of ANFIS were optimized via back-propagation [39]. The reader can refer to [42] [43] and [44] for an in-depth discussion of the mathematics, optimization, and robust possibilistic clustering-based initialization of ANFIS. Finally, the output decision threshold was chosen through DTA.

The different *Launch Pad* detector types were tested independently and in combination during the fusion step along with the three other component types (i.e. *Missiles*, *TELS*, and *TEL Groups*):

- Empty *Launch Pads* plus three (*Empty LPs+3*)
- Combined *Launch Pads* plus three (*Combo LPs+3*)
- *Empty LPs* and *Combo LPs* plus three (All 5)

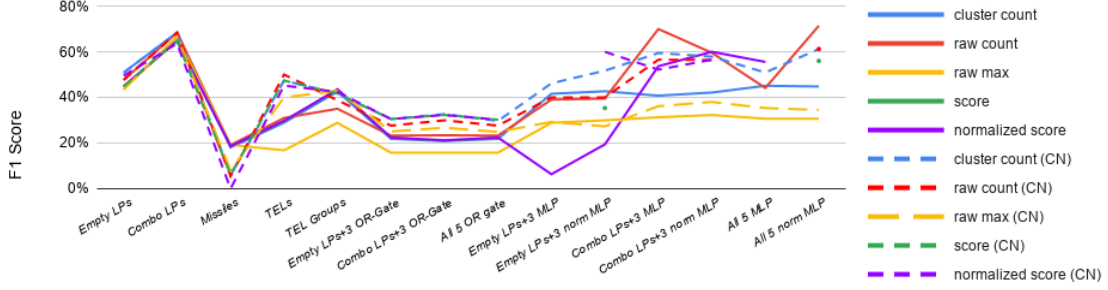


Fig. 2.8: Comparison of $F1$ scores produced for candidate *SAM Site* locations from different fusion techniques. Techniques include individual component threshold from DTA as well as component fusion using an OR gate and MLP. Note that “(CN)” at the end of the feature type label in the key indicates that component negative models were used in the processing. Gaps in scores occur when the MLP was unable to train on a given feature type.

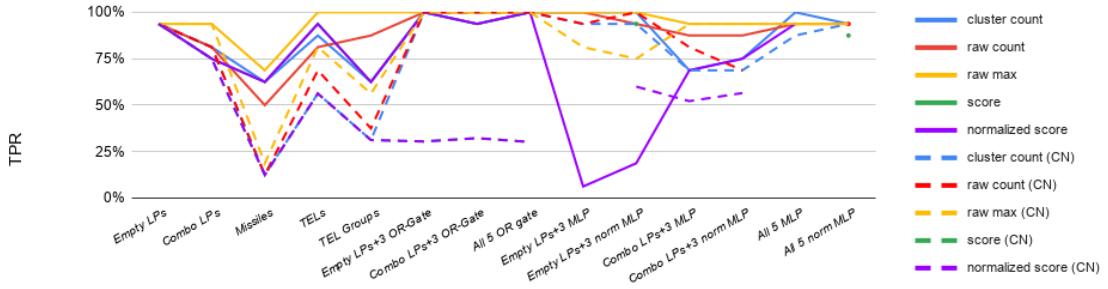


Fig. 2.9: Comparison of True Positive Rate (TPR) produced for candidate *SAM Site* features from different fusion techniques.

2.5.3 MLP Input Data Normalization

We found that the MLPs had some difficulty training with datasets that had larger values, so we used the common practice of linearly scaling and bounding to constrain the data to fall within the range $[-1, 1]$. Let v_i be the vector of values over the entire dataset for component i for a given feature and let t_i be the DTA thresholds computed for component i , then the normalized and bounded vector v'_i can be defined as follows:

$$v'_i = \begin{cases} \text{if } (v_i - t_i)/t_i \in [-1, 1], \text{ then } (v_i - t_i)/t_i \\ \text{if } (v_i - t_i)/t_i < -1, \text{ then } -1 \\ \text{if } (v_i - t_i)/t_i > 1, \text{ then } 1 \end{cases} \quad (2.4)$$

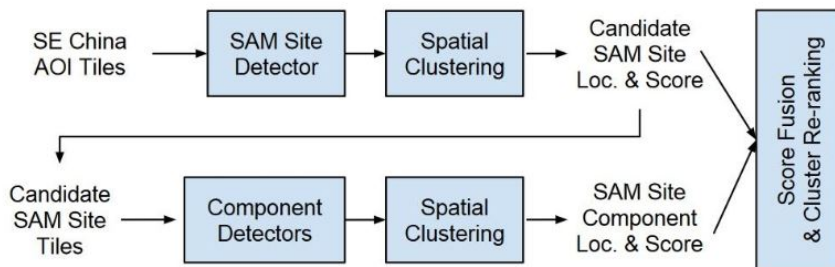


Fig. 2.10: Process flow used for improved ranking of candidate *SAM Sites*.

2.6 RESULTS & OBSERVATIONS

2.6.1 Improved Candidate *SAM Site* Rankings

This section discusses techniques, observations, and results used to re-rank candidate *SAM Sites* for utilization in machine-assisted human analytic workflows. The objective is to utilize the component detection clusters to re-rank the candidate *SAM Sites* such that true *SAM Sites* appear higher in a rank-ordered list relative to a baseline ranking derived only from the candidate *SAM Sites*' cluster scores (Table 2.6). An overview of the processing flow is given in Fig. 2.10.

For re-ranking we are used the summing as described in Section 2.5.1 no additional fusion as described in Section 2.5.2 in completed afterwards. The summation is completed in two ways:

1. Normalized cluster scores for candidate *SAM Sites* and all components found within R are summed using uniform or human expert provided weights (Fig. 2.10).
2. Expert weights were only used when fusing all four components with its corresponding candidate *SAM Site*. The weights were: 4 for *Launch Pads*, 2 for *TEL Groups*, and 1 for *Missiles*, *TEs*, and *SAM Sites*.

The *TEL* detector achieved the most improvement in the average cluster rank of known *SAM Sites* (*TPs*) compared to fusion with any other single component detector (Table 2.6). This, coupled with the *Combo LPs* detector and other component

Table 2.6: Average rank of known *SAM Sites* in SE China AOI from fusing cluster scores from a single component object class with a baseline candidate *SAM Site* cluster score.

	<i>SAM Site</i> Only	with Single Component Fusion				
		<i>Empty LPs</i>	<i>Combo LPs</i>	<i>Missiles</i>	<i>TELEs</i>	<i>TEL Groups</i>
ResNet-101 [19]	139.9	n/a	n/a	n/a	n/a	n/a
NASNet	62.8	36.4	40.8	43.0	28.0	46.1
w/ Norm	62.8	34.3	34.4	43.6	28.1	47.3
w/ Norm & Penalty	62.8	34.0	34.3	43.6	27.9	47.1
Including Component Negatives						
w/ Norm	n/a	n/a	n/a	79.1	28.8	51.6
w/ Norm & Penalty	n/a	n/a	n/a	79.1	28.7	51.48

detectors trained with expert weighting (Section 2.6.1) improved the average cluster rank of known *SAM Sites* (*TPs*) to 15.9 (Table 2.7). This is $\sim 4X$ better than the average rank for *SAM Sites* without spatial fusion of the component object cluster scores.

We observed that the addition of normalization and penalty had no detectable impact on the known *SAM Site* TP average cluster rank. This indicates minimal FP presence and/or uniformly distributed FP noise within the candidate *SAM Site* locations generated by the spatial clustering algorithm.

Component negative models improved the ranking results compared to the *SAM Site* score alone, but not as well as models trained without component negatives. Again, this can be interpreted as ambiguity being introduced to the dataset by essentially asking the detector to ignore the background (i.e. the *Launch Pad*) and focus on the smaller component.

Table 2.7: Average rank of known *SAM Sites* in SE China AOI from fusing cluster scores from all four component object classes with the baseline candidate *SAM Site* cluster score.

	<i>Empty LPs</i>		<i>Combo LPs</i>	
	Unweighted Fusion	Weighted Fusion	Unweighted Fusion	Weighted Fusion
NASNet	26.3	21.4	25.3	22.9
w/ Norm	20.3	22.9	17.9	15.9
w/ Norm & Penalty	19.9	22.5	17.8	16.0
Including Component Negatives				
w/ Norm	24.8	24.9	18.1	16.8
w/ Norm & Penalty	24.1	24.9	18.1	16.8

2.6.2 Improved *SAM Site* Detection

Over 200 different combinations of data feature types, component combinations, and fusion techniques were tested in this study to improve the final detection of candidate *SAM Sites*.

Evaluation of the *F1* score improvements (Table 2.8) shows that decision-level component fusion can reduce the relative error rate by up to 96.75%. It was somewhat surprising that the Raw Count feature generated five out of the top six best results. Although *Combo LPs* were only able to generate an *F1* score of 68.4% using DTA, the neural approaches (MLP and ANFIS) were able to do slightly better using multiple components where the top results fused all 5 components in an MLP to yield an *F1* score of 71.4%. Comparisons of *F1* scores for different feature types and fusion techniques can be found in Fig. 2.8.

However, when performing a broad area search for a very rare object (low geographic occurrence rate), it is often desirable to sacrifice some error reduction in order to achieve a higher TPR. The results in Table 2.9 show that the highest *F1* score was 45.1% while achieving a *TPR* of 100%. Although this *F1* score is less than half of the

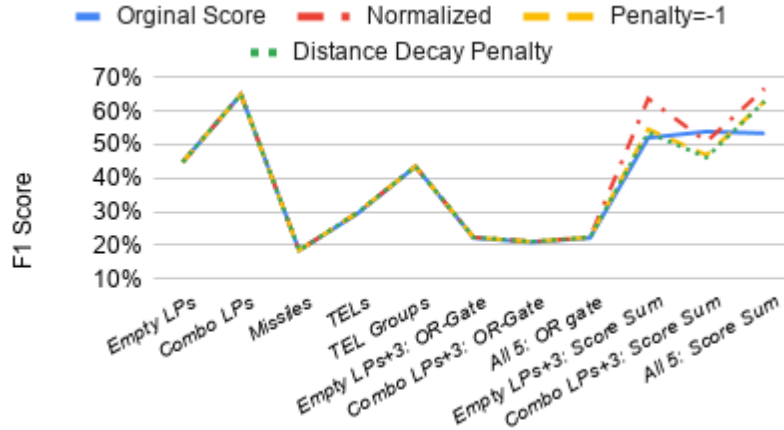


Fig. 2.11: *F1* score results for DTA thresholds of original cluster scores, normalized cluster scores, cluster scores with a penalty of -1 and distance-decay penalty with $R = 150$ m.

maximum in Table 2.8, this technique still achieved a 88.5% relative error reduction compared to the baseline (no component fusion) results for the candidate *SAM Site* locations within the SE China AOI. These scores were produced using Cluster Count features and the *All 5* component combination as inputs to a simple MLP. It is also worth noting that four of the top five scores used the *Empty LPs+3* component combination. Comparisons of TPRs for different feature types and fusion techniques can be found in Fig. 2.9.

It was also observed that cluster score truncation and normalization were able to improve the *F1* scores for DTA when fusing multiple component detectors. However, the introduction of negative score penalty did not improve the score further (Fig 2.11), while introducing expert weighting (described in Section 2.6.1) also showed no improvement for the *F1* scores.

Additionally, in general there was improvement in *F1* scores for models trained with component negatives, however these improvements came at a sacrifice in TPR and only have one appearance in the Tables 2.8 and 2.9. Again, this can be interpreted as ambiguity being introduced to the dataset by essentially asking the detector to ignore the background (i.e. the *Launch Pad*) and focus on the smaller component.

Table 2.8: Experiment results with highest $F1$ Scores. The first line after the header (in red) are the results for *SAM Site* detection without error reduction from spatial fusion of any component feature type(s). The highest $F1$ scores were achieved by fusing multiple components using neural learning techniques (MLP or ANFIS). Also, raw detection counts (pre-clustering) showed the most separability. All top solutions achieved a relative error reduction of greater than 96%. These results would be optimal if error reduction was the primary goal. The error rate includes both false positives and false negatives.

Components	Feature Type	Processing Technique	Component Negatives	TP	FP	TPR (Recall)	PPV (Precision)	$F1$ score	Error / km ² (x10 ⁻³)	Relative Error Reduction
<i>SAM Sites</i>	BASELINE-NO COMPONENTS		16	338	100.00%	4.52%	8.65%	3.080	n/a	
All 5	Raw Counts	MLP	NO	15	11	93.75%	57.69%	71.43%	0.109	96.45%
<i>Combo LPs+3</i>	Raw Counts	ANFIS	NO	13	8	81.25%	61.90%	70.27%	0.100	96.75%
<i>Combo LPs+3</i>	Raw Counts	MLP	NO	14	10	87.50%	58.33%	70.00%	0.109	96.45%
<i>Combo LPs</i>	Cluster Count	DTA	n/a	13	9	81.25%	59.09%	68.42%	0.109	96.45%
<i>Combo LPs</i>	Raw Count	DTA	n/a	13	9	81.25%	59.09%	68.42%	0.109	96.45%
All 5	Raw Count	ANFIS	NO	13	9	81.25%	59.09%	68.42%	0.109	96.45%

Table 2.9: Experiment results with highest $F1$ scores while maintaining a TPR of 100%. The highest $F1$ scores resulted from fusing a feature from all components with a simple MLP. Also, Cluster Count features yielded the top results. All top solutions show a reduction of relative error between 85.2 – 88.5% which is 3X the error rate shown in Table 2.8.

Components	Feature Type	Processing Technique	Component Negatives	TP	FP	TPR (Recall)	PPV (Precision)	$F1$ score	Error/ km ² (x10 ⁻³)	Relative Error Reduction
<i>SAM Sites</i>	BASELINE-NO COMPONENTS		16	338	100%	4.52%	8.65%	3.080	n/a	
All 5	Cluster Count	MLP	NO	16	39	100%	29.09%	45.07%	0.355	88.46%
<i>Empty LPs+3</i>	Cluster Count	MLP (Normalized)	NO	16	43	100%	27.12%	42.67%	0.392	87.28%
<i>Empty LPs+3</i>	Cluster Count	MLP	NO	16	45	100%	26.23%	41.56%	0.410	86.69%
<i>Empty LPs+3</i>	Raw Count	MLP (Normalized)	YES	16	48	100%	25.00%	40.00%	0.437	85.80%
<i>Empty LPs+3</i>	Raw Count	MLP	NO	16	50	100%	24.24%	39.02%	0.456	85.21%

2.7 CONCLUSION AND FUTURE WORK

This study extended the work in [19] where a combination of a DNN image scanning and spatial clustering of the resulting detections was used to perform a machine-assisted broad area search and detection of *SAM Sites* in a SE China AOI of $\sim 90,000$ km².

Here we significantly improved upon this prior study by using multiple DNNs to detect smaller component objects, e.g. *Launch Pads*, *TELS*, etc. belonging to the larger and more complex *SAM Site* feature. Scores computed from an enhanced spatial clustering algorithm were normalized to a reference space so that they were independent of image resolution and DNN input chip size. A variety of techniques were then explored to fuse the DNN detections from the multiple component objects to improve the final detection and retrieval (ranking) of DNN detections of candidate *SAM Sites*. Key results from this effort include:

1. Spatial fusion of DNN detections from multiple component objects using neural learning techniques that maximize the *F1* score reduced an initial set of ~ 350 *SAM Site* detections (Table 2.5) to only ~ 25 candidate *SAM Sites* (Table 2.8).
2. An alternate spatial fusion approach from that used in 1) reduced the overall error rate by $>85\%$ while preserving a 100% TPR (Table 2.9) and also reduced the initial set of detections to $\sim 55-60$ candidate *SAM Sites*.
3. The average rank of 16 known *SAM Sites* (*TPs*) in a list of ~ 350 candidate *SAM Sites* was improved by $\sim 9X$ (Tables 2.6 and 2.7) compared to the previous study [19].

In subsequent chapters we use decision-level fusion and component object confirmation to other challenging object search and detection problems in large-scale remote sensing image datasets and explore how to use more sophisticated fusion

techniques (similar to ANFIS) to maintain TPR while achieving even higher error reduction. Additional future work could include A) investigate data-driven optimization of the component fusion weights and compare performance vs. human-expert provided weights, B) extend this approach to include fusion of multi-temporal DNN detections, and C) extend this approach to include fusion of multi-source DNN detectors applied to high-resolution EO/MS and SAR imagery.

Chapter 3

DECISION-LEVEL FUSION OF DNN OUTPUTS FOR IMPROVING FEATURE DETECTION PERFORMANCE ON LARGE-SCALE REMOTE SENSING IMAGE DATASETS

This research in this chapter was primarily published at the IEEE BIGDATA 2019 Conference [21] in Los Angeles, California. Co-authors included several members of the MU Center for Geospatial Intelligence: Raymond L. Chastain, J. Alex Hurt, Curt H. Davis, and Grant J. Scott along with an outside contributor from NGA: A.J. Malentfort.

3.1 INTRODUCTION

The objective for this research was to develop a method to refine initial “candidate” feature/object detections generated by a single Deep Neural Network (DNN) detector by fusing results from multiple DNNs designed to detect component and/or related objects, often at different resolutions/scales, to improve the detection of the candidate feature/object. This approach can be potentially used in hierarchical and computationally efficient processing strategies to reduce errors and improve machine-assisted analytic performance for large-scale remote-sensing applications.

DNN detections of multiple constitutive or component objects that are part of a larger, more complex, and encompassing feature are spatially fused to improve the

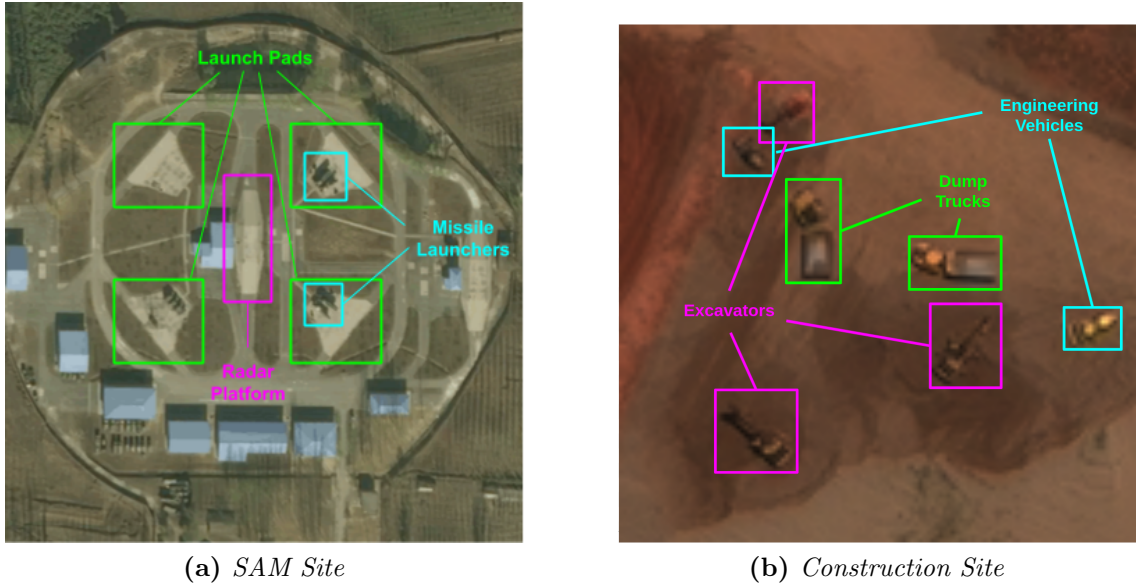


Fig. 3.1: Example of a *Surface-to-Air Missile (SAM) Site* (left) and *Construction Site* (right) with various constituent components highlighted. Note the contrast in the regularly distributed and permanently installed components of the SAM site and the irregular spatial distribution and highly mobile nature of the construction site components. Left image courtesy of the DigitalGlobe Foundation; right image and all other images in this paper are from the xView dataset [45].

detection performance of a larger complex feature. A wide variety of experiments were conducted using the public domain xView dataset [45] to develop and evaluate multiple fusion strategies. The purpose of these experiments was to improve the detection of *Construction Sites* using DNN detections of constitutive/component objects commonly associated with construction activity (e.g. cement mixers, dump trucks, etc). The results demonstrate that spatial fusion of multi-scale component object DNN detections can reduce the total detection error rate of *Construction Sites* by $\sim 30\text{-}40\%$. The best results were obtained when local spatial clustering was used to reduce noise in component vehicle object detections generated of scanning of candidate *Construction Site* locations. This multi-scale spatial fusion approach can be easily extended to improve detection performance in a wide variety of other challenging feature/object search and detection problems in large-scale remote sensing image datasets.

3.2 SOURCE DATA

This study used data from the “DIUx xView 2018 Detection Challenge” [45]. The xView dataset is a series of large image scenes (~ 1 km by 1 km) with sets of feature/object bounding boxes corresponding to one of sixty different class labels. These classes are further grouped into 7 parent or superclasses and 9 classes without a superclass (“None” column in Table 3.1). The xView dataset released in “Train” and “Test” groupings were combined into a larger training dataset for our experiments. The xView “Validation” dataset was then used for blind validation. Labeled image samples of the feature/object classes were created by cropping a desired sample image size from the larger scene centered on each feature’s bounding box.

For computational simplicity and hardware optimization, we trained DNN models using square image samples with image side lengths in powers of 2. After looking at the size of the desired feature/object classes in the xView dataset, we decided to train DNN detectors in square pixel dimensions of 32, 64, 128, and 256. The Maximum Pixel Length (MPL) of the xView-provided bounding box was used to select which samples would be used to train a DNN model for a given input window size. The xView dataset was produced at an assumed nominal GSD of ~ 0.3 m. However for our experiments we re-scaled or re-sampled to 0.5m GSD and adjusted the MPL accordingly for each sample. The re-sampling to 0.5m GSD was done in anticipation of utilizing the methods developed in this study for another independent dataset that had a mean GSD at 0.5 m. However, the application to this additional independent dataset was not part of this initial effort.

3.2.1 Multi-Class and Parent Class Datasets

The xView training data had 581,953 total samples across all classes. This was split into four overlapping partitions of square dimensions: 32, 64, 128, and 256 with pre- and post- augmentation sample counts provided in Table 3.2. These data partitions

Table 3.1: Breakdown of designated xView parent/child class relationships. The parent class name is in bold at the top of each column. The classes in the “none” column have no additional parent classification and we used as a unique class when creating the parent class datasets. Note also that each parent class has a subclass of the same name, not listed under the parent class heading.

Fixed Wing Aircraft	Passenger Vehicle	Truck	Railway Vehicle	Maritime Vessel	Engineering Vehicle	Building	None
<i>Passenger Cargo Plane</i>	<i>Bus</i>	<i>Cargo Truck</i>	<i>Cargo Container Car</i>	<i>Barge</i>	<i>Cement Mixer</i>	<i>Aircraft Hangar</i>	<i>Helipad</i>
<i>Small Aircraft</i>	<i>Small Car</i>	<i>Pickup Truck</i>	<i>Flat Car</i>	<i>Container Ship</i>	<i>Container Crane</i>	<i>Damaged Building</i>	<i>Pylon</i>
		<i>Trailer</i>	<i>Locomotive</i>	<i>Ferry</i>	<i>Crane Truck</i>	<i>Facility</i>	<i>Shipping Container</i>
		<i>Truck Tractor (TT)</i>	<i>Passenger Car</i>	<i>Fishing Vessel</i>	<i>Dump Truck</i>	<i>Hut Tent</i>	<i>Shipping Container Lot</i>
		<i>TT w/ Box Trailer</i>	<i>Tank car</i>	<i>Motorboat</i>	<i>Excavator</i>	<i>Shed</i>	<i>Construction Site</i>
		<i>TT w/ Flatbed Trailer</i>		<i>Oil Tanker</i>	<i>Front Loader</i>		<i>Helicopter</i>
		<i>TT w/ Liquid Tank</i>		<i>Sailboat</i>	<i>Ground Grader</i>		<i>Storage Tank</i>
		<i>Utility Truck</i>		<i>Tugboat</i>	<i>Haul Truck</i>		<i>Vehicle Lot</i>
				<i>Yacht</i>	<i>Mobile Crane</i>		<i>Tower</i>
					<i>Reach Stacker</i>		
					<i>Scrapper Tractor</i>		
					<i>Sraddle Carrier</i>		
					<i>Tower Crane</i>		

Table 3.2: Minimum and maximum MPL at 0.5m GSD required for a sample to be included for a given DNN model size and total samples available/used pre/post-capping.

Overlapping Multi-Class Dataset Used for Training DNN Models with Different Input Size				
Input DNN Model Size	32	64	128	256
Minimum Feature MPL	0	16	32	64
Maximum Feature MPL	32	64	128	∞
Total Available Sample Counts	451,412	276,000	125,908	25,986
Class Count After Thresholding and Reduction	46	55	41	23
Sample Counts After Thresholding and Reduction	254,257	135,994	113,731	25,902

were used in multi-class experiments, i.e. experiments using all 60 classes, and were created using the following criteria:

1. Datasets for training each DNN model used an MPL limited to a maximum length of the model size and a minimum of one quarter the model size (i.e. model size x^n would contain all xView features with MPL within the range $[x^{n-2}, x^n]$). All xView features with MPL smaller than 32 pixels were included in the dataset for DNN models with a 32 input size and all xView features with MPL greater than 256 pixels were included for DNN models with a 256 input size, but cropped at 256x256 pixels.
2. Classes with insufficient percentage (less than 25% of total class count) for a particular size were dropped from the training data for a given DNN model input size (Appendix A.1).
3. Classes with very large sample counts (e.g. *Buildings* and *Small Car*) were reduced to enforce balance. For the multi-class experiments, the sample number was capped at 10,000.

A set of parent class (see Table 3.1) datasets were created and the final sample

Table 3.3: xView sample counts belonging to each overlapping parent class for 0.5m GSD.

Overlapping Parent Class Datasets Used for Training DNN with Different Model Input Size				
Parent Class	32	64	128	256
<i>Building</i>	4254	4038	2886	20,530
<i>Construction Site</i>	0	300	606	619
<i>Engineering Vehicle</i>	3900	2918	530	285
<i>Fixed Wing Aircraft</i>	347	676	573	339
<i>Helicopter</i>	48	66	19	0
<i>Helipad</i>	51	82	64	0
<i>Maritime Vessel</i>	1373	2333	1350	764
<i>Passenger Vehicle</i>	19,258	6177	1545	0
<i>Pylon</i>	100	327	241	0
<i>Railway Vehicle</i>	1784	3543	336	0
<i>Shipping Container</i>	1493	996	0	0
<i>Shipping Container Lot</i>	455	1368	1359	648
<i>Storage Tank</i>	940	903	636	233
<i>Tower</i>	46	66	37	0
<i>Truck</i>	29,533	13,544	2244	0
<i>Vehicle Lot</i>	779	2079	2525	1591

counts provided in Table 3.3. The same criteria were used to create the parent class datasets as the multi-class datasets with the caveat that the classes with large sample counts within a parent class group were reduced to not exceed the total count of the remainder classes. In the case of size 32, the *Building* and *Small Car* classes were both reduced to not exceed the total of the remaining classes, excluding each other. Each class listed under “None” in Table 3.1 was used as a unique class when creating the parent class datasets.

3.2.2 Binary Datasets

To further reduce ambiguity in the multi/parent class detectors (discussed in Section 3.4.1), we created binary DNN models that focused on the *Construction Site* (CS) class and likely components within *CS* features. The average count of each class within 150 meters of any other class was computed and used to inform selection

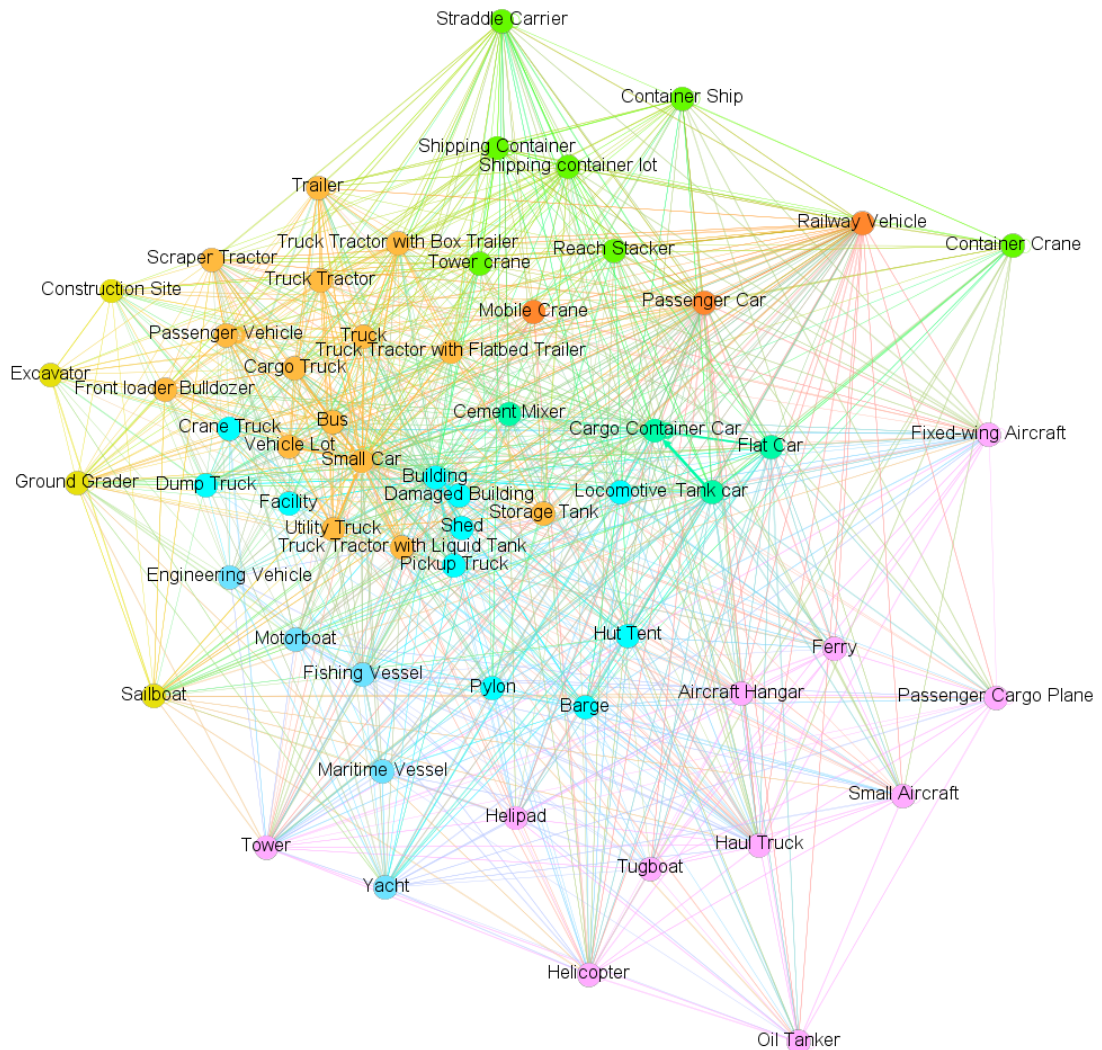


Fig. 3.2: Graph representation of proximity counts of xView classes within 150 meters of any other class. The closer classes are to each other, the higher count those classes have within proximity to each other. The proximity of classes belonging to parent classes *EV* and *TR* to the *CS* class (in yellow modularity cluster) informed our selection of the parent classes used for *CS* constituent object scanning and detection. Node colors indicate modularity clusters.

of the *Engineering Vehicle* (*EV*) and *Truck* (*TR*) parent classes for *CS* constituent component object scanning and detection (Fig. 3.2)

Negative samples were selected randomly from a pool of the remaining samples and were approximately 4 times (4X) the number of positive samples for each class

Table 3.4: Sample counts for binary DNN models. Negative samples were selected randomly from a pool of xView features after excluding xView features in the *CS*, *EV*, and *TR* parent classes to avoid contamination. Negative sample counts are approximately 4X the positive sample count.

Sample Counts for Binary DNN Models				
Input DNN Model Size	32	64	128	256
Construction Site				
Positive Count	-	300	605	615
Negatives Count	-	1200	2420	2475
Engineering Vehicle				
Positive Count	3900	2915	530	285
Negatives Count	15,600	11,670	2125	1140
Truck				
Positive Count	29,533	13,544	2244	-
Negatives Count	57,515	38,199	2244	-

(excluding *CS*, *EV*, and *TR* parent classes to avoid contamination). Sample counts for the training image sets using the binary models are provided in Table 3.4.

3.2.3 Multi-Scale Construction Site Detection Datasets

Three unique *CS* datasets were created from the xView validation dataset to evaluate the performance of *CS* models at different scales that could be used for large area scanning/detection. The first *CS* dataset used an MPL less than 64 pixels, the second between 64 and 128 pixels, and the third used an MPL greater than 128 pixels. Datasets were then created for each MPL range for various DNN model input sizes (i.e. 64, 128, & 256 pixels) thereby creating a total of nine datasets.

3.2.4 Candidate Construction Site Datasets

Based on the results provided later in Section 3.3.2, separate candidate *CS* datasets were created from the xView training and validation datasets. *CS* features with MPL less than or equal to 128 were included in the 128 size *CS* candidate location dataset, otherwise *CS* features >128 were included in the 256 candidate location datasets.

Table 3.5: Candidate location dataset sample size for *CS* detection experiments.

Dataset Type	CS Candidate Size	Positive Count	Negative Count
Training	128	638	2732
Training	256	251	10,004
Validation	128	221	884
Validation	256	162	648

The training datasets were used for development of decision-level fusion methods and the validation datasets were used for final bind testing. Sample counts are provided in Table 3.5.

3.3 EXPERIMENTS

3.3.1 5-fold Cross-Validation Experiments

Multi-fold cross-validation experiments are useful for testing DNN models during development, especially when using datasets with limited sample sizes. We chose to perform 5-fold cross validation as described in Section 2.3.1. For multi-class and parent class datasets, 5-fold validation was performed using NASNet [30] DNN models trained for 1 epoch on an augmented dataset with transfer learning from ImageNet [35] weights. The augmented dataset utilized per-class sample augmentation to balance the sample size to approximately 100,000 post-augmentation samples per class. For example, there were 809 *Excavators* with MPL<16m in the multi-class experiment. Flipping each image produced a 2X multiplier, and rotating every $6^\circ \in [0^\circ, 360^\circ)$ a 60X multiplier, thus producing a 120X multiplier for a total of 97,080 unique samples. 5-fold cross validations were completed similarly for binary datasets but using ProxylessNAS [33] with a 144X augmentation consisting of a vertical flip of the image sample and rotations every $5^\circ \in [0^\circ, 360^\circ)$.

Table 3.6: Assessment metrics for unique *CS* samples from xView validation data to determine the best DNN model size to use for final evaluation. Top results are highlighted in green.

Validation Data Results			
MPL Input Range	DNN Model Size		
	64	128	256
Construction Site Recall			
<64	81.46%	95.36%	90.73%
64<128	84.62%	92.31%	92.31%
>128	62.96%	83.95%	82.10%
Weighted F1			
<64	88.97%	95.92%	92.65%
64<128	89.45%	93.79%	94.38%
>128	75.07%	86.35%	88.81%

3.3.2 Multi-Scale Construction Site Detection Experiments

Broad area scanning using DNNs is computationally expensive and, therefore, if certain model sizes can be dropped during the scanning phase, then the evaluation of DNN model size for object detection is worthwhile. These experiments were conducted on the uniquely partitioned binary image sets described in Section 3.2.4 and evaluated *CS* detection models of size 64, 128, and 256.

A surprising result from this experiment was that a DNN model size of 128 was more accurate for detecting smaller *CS* than a DNN model size of 64 (Table 3.6). This result informed the decision to divide the final xView validation data into 128 & 256 model sizes and drop the 64 model size from the remaining analysis.

3.3.3 Component Object Image Scanning and Decision-Level Fusion

In the experiment described in the previous section, we found that the DNN model with 128 input size was the most effective for *CS* features with MPL less than 256. In addition, we also found that the DNN model with 256 input size was most effective for *CS* features with MPL greater than 256. Thus, the rest of the *CS* experiments

were designed with these results in mind.

Binary *CS* component object detectors were trained to scan the candidate *CS* locations. The binary DNN models were trained using the NASNet architecture utilizing the complete datasets from the 5-fold experiments and using custom augmentation schemes to produce post-augmentation sample sizes of $\sim 6,000,000$. Image jitter, brightness shifting, and contrast adjustment (details found in Appendix A.2) were used in addition to aforementioned flip and rotations.

Each candidate *CS* sample for a designated DNN model input size (128 or 256) was scanned (Section 2.3.2) using a 75% overlap (25% stride) for all binary component detectors with DNN input size equal to or less than the candidate location sample size. The following methods were tested for decision-level fusion (Fig. 3.3) of the DNN inference response fields:

1. An initial threshold or α -cut for the *CS* detector responses was done. Next, the Decision-Theoretic Approach (DTA), as described in Section 2.3.5, was used to find optimal *F1*-score thresholds for the maximum response for each component. If the maximum response was greater than the *F1*-score thresholds for any given component then it was deemed a positive *CS* detection.
2. A “flattened” response field was used as an input vector for a Multi-Layer Perceptron (MLP) while preserving certain characteristics between candidate *CS* samples. The following characteristics were preserved for separate experiments:
 - (a) Response orientation within the sample, by component.
 - (b) Descending order of response magnitude, by component.
3. An initial threshold or α -cut for the *CS* detector responses was done. Next, DTA was used to find optimal *F1*-score thresholds for the number of raw inference detections over a given threshold for each component within a spatial proximity

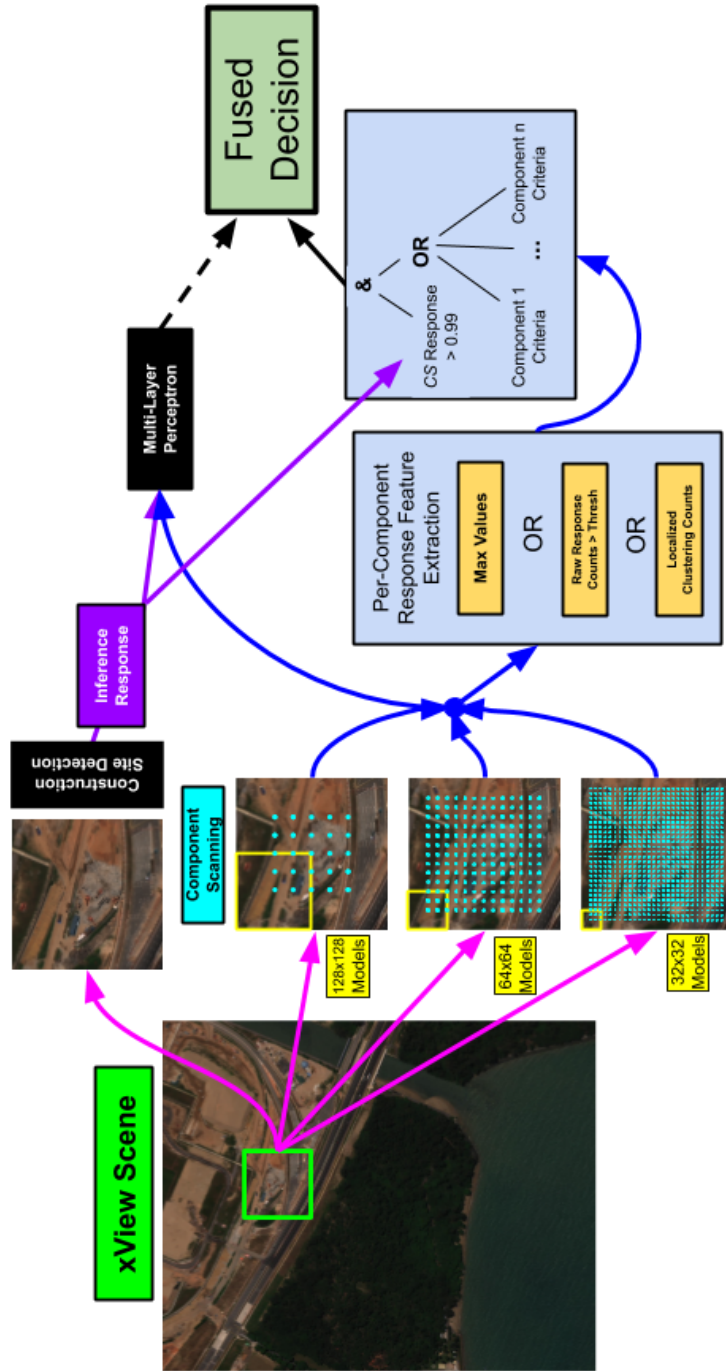


Fig. 3.3: Overview of processing workflow from image scene to decision-level fusion. From left to right: 1) Broad area scanning (simulated in this study) to identify candidate *Construction Site* (CS) samples. 2) Generate inference response for candidate CS samples by passing through *Construction Site* detection model. 3) Scan candidate CS samples with component DNN models of different sizes (e.g. multi-scale). 4a) Pass CS inference response and component inference responses through MLP, or 4b) apply logic tree to CS inference response and extracted per-component features. 5) Final binary fused decision. Note that the final decision is not a combination of the results from the MLP and decision tree and is instead mutually exclusive, i.e. the MLP was used as an alternate approach for comparison with the decision-level fusion results.

Table 3.7: Weights used for decision-level fusion using maximum response, raw component responses over a thresholds, and component counts after local spatial clustering. Note that *Truck* models 64 and 128 were not included due to the fact that decision boundaries could not be achieved through *F1*-score optimization and were therefore removed from the final decision. This is also true for any cell in the table marked as ‘-’.

Processing Method	Pre- α -cut	EV	EV	EV	TR
DNN Model Size		32	64	128	32
Candidate Location Samples w/ 128 Side Length					
Maximum Response Analysis	0.99	0.6	0.5	0.1	-
Raw Component Counts	0.99	3	1	0	-
Locally Clustered Component Counts	0.95	1	0	0	2
Candidate Location Samples w/ 256 Side Length					
Maximum Response Analysis	0.99	0.79	0.5	0.27	-
Raw Component Counts	0.99	-	13	18	-
Locally Clustered Component Counts	0.99	0	0	0	-

to the candidate *CS* sample. If the maximum response was greater than the *F1*-score thresholds for any given component then it was deemed a positive *CS* detection.

4. An initial threshold or α -cut for the *CS* detector responses was done. Next, DTA was used to find optimal *F1*-score thresholds for the counts of cluster centers after local clustering (Section 2.3.2) for each component within a spatial proximity to the candidate *CS* sample. We used α -cuts described in Table 3.7 and maximum radius of 3/8 the model width in meters to allow for differentiation between *EVs* in close proximity (i.e. a 32 pixel model had a maximum radius of 6m for clustering). If the cluster center counts was greater than the above thresholds for any given component then it was deemed a positive detection. A more detailed flow chart can be found in Appendix A.3.

Table 3.8: Average metrics of multi-class 5-fold experiments.

5-Fold Experiments for Multi-Class and Parent Class				
DNN Model Input Size	32	64	128	256
Multi-Class Averages				
Class Count	47	56	43	42
Recall	43.9%	52.0%	56.1%	65.2%
Precision	35.1%	48.0%	55.8%	66.6%
<i>F1</i>	38.1%	49.3%	55.4%	65.5%
Parent Class Averages				
Class Count	15	16	15	8
Recall	62.9%	72.5%	79.2%	89.1%
Precision	50.0%	69.4%	80.6%	86.8%
<i>F1</i>	54.0%	70.7%	79.6%	87.9%

3.4 RESULTS

3.4.1 5-Fold Multi-Class and Parent Class Experimental Results

The average *F1* score for the multi-class 5-fold experiments ranged from 38% -66% (Table 3.8). Overall this shows low reliability for multi-class DNNs with 60 distinctive classes. Decreasing the number of classes through class aggregation has been shown to decrease ambiguity between classes. This is evident in the xView data where the average *F1* scores for the parent class 5-fold results ranged from 50-89% across the four DNN model sizes. Although the results of the 256 size parent class models are promising, the lower accuracy of the smaller model sizes indicates that models with fewer classes would produce more accurate results for scanning.

Based on the results in Table 3.8, we decided to train binary DNN models to further decrease the class ambiguity. 5-fold binary DNN model validations were then completed for parent classes: *CS*, *EV*, and *TR*. These experiments yielded *F1* scores between 87% -95% (Table 3.9) which are more reasonable for use in large-area image scanning.

Table 3.9: Average $F1$ scores for binary DNN parent class models.

5-Fold Experiments for Binary Models				
DNN Model Input Size	32	64	128	256
<i>Construction Site</i> (CS)	-	89.8%	91.7%	94.4%
<i>Engineering Vehicle</i> (EV)	90.4%	92.5%	93.3%	91.5%
<i>Truck</i> (TR)	87.2%	95.0%	93.4%	-

3.4.2 Decision-Level Fusion Results

Baseline results were established using the maximum class response from *CS* inference processing with the DNN model of appropriate size. For binary models, a 50% threshold was used after processing through a normalizing activation function (e.g. softmax). While a 128 DNN model size could be used to scan 256 candidate locations and possibly enhance *CS* detection, we used inference responses of candidate locations that were the same size as the *CS* in the final analysis to solely examine enhancements based on fusion of the component detections.

The experiments also showed that the *TR* binary models were, for the most part, not helpful in final decision making because the class was too common and not specific enough for *Construction Sites*. Therefore, *EV* component scanning results were used almost exclusively for maximum response and counts analyses. We can see that by simply increasing the response threshold from 0.5 to 0.99 (Table 3.10), the relative reduction in error was 16% and 32% for the 128 and 256 candidate location sizes, respectively. Utilization of the maximum response for each component yielded a error relative reduction of 23.5% and 37.8% from the *CS* only detector baseline. Using a simple MLP produced mixed results, but overall did poorly compared to other methods, as did raw component response counts. However, counting cluster centers for a candidate *CS* samples after local spatial clustering produced the greatest relative reduction in error of 28.4% for 128 candidate location size and 40.5% for 256 candidate location size. The optimized $F1$ -score thresholds can be found in Table 3.7.

3.5 CONCLUSION AND FUTURE WORK

This research has demonstrated that decision-level fusion of multiple DNN detections can improve feature detection performance and significantly reduce error rates for irregularly shaped/complex features (e.g. *Construction Sites*) with mobile and ephemeral constitutive components (e.g. *Engineering Vehicles*). While this finding is not surprising, this is something that, as far as we know, has not been demonstrated through controlled experiments such as these.

Further, the best results were obtained when local spatial clustering was used to reduce noise in the multi-scale component scanning detections with the maximum response coming in a close second. This indicates that scanning noise reduction is vital for small object detections in local targeted area scans just as has been demonstrated for larger features in broad area scanning such as in [19], [46], and the initial SAM Site experiments in Chapter 2 ([20]).

Although the use of simple MLPs was not as effective as other fusion methods, we believe that the use of more sophisticated DNNs might be able to improve decisions on both the raw responses and possibly the extracted per-component features described in Section 3.3.3. Likewise, the optimized *F1*-score thresholds used in the final, fused decision could also be replaced with rules from human expert judgement and/or rule-based fuzzy logic. For the latter, the membership functions could be either empirically derived or generated from a trained system such as Adaptive Network based Fuzzy Inference System (ANFIS) [40], [41] which was implemented in Chapter 2 as part of the JSTARS 2020 research (see Section 2.5.2).

Further, the multi-scale spatial fusion approach demonstrated here can be easily extended to improve performance in a wide variety of other challenging feature/object detection problems using large-scale remote sensing image datasets such as the recent work on *SAM Site* broad area search and detection presented in Chapter 2.

Table 3.10: Assessment metrics for *Constructions Sites* (CS) detection with baseline 0.5 α -cut and multi-DNN decision-level fusion. *F1* score, error rate, and error reduction in % points and relative error reduction (red). The top result for locally clustered component counts is highlighted in green.

Processing Method	F1	Error Rate**	Error Difference (Red is Relative Error Reduction)
128 Candidate Location and CS Model Size			
CS Only w/- α -Cut = 0.50	89.58%	21.77%	N/A (baseline)
CS Only w/- α -Cut = 0.99	90.93%	18.28%	-3.49% (-16.0%)
with Component Processing^{^^}			
Maximum Response Analysis	91.48%	16.67%	-5.11% (-23.5%)
MLP w/ Unsorted Confidence Field	90.22%	20.16%	-1.61% (-7.4%)
MLP w/ Sorted Confidence Field	89.27%	22.04%	+0.31% (+1.42%)
Raw Component Counts	86.57%	24.19%	+2.42% (+11.1%)
Locally Clustered Component Counts	91.90%	15.59%	-6.18% (-28.4%)
256 Candidate Location and CS Model Size			
CS Only w/- α -Cut = 0.50	87.79%	25.69%	N/A (baseline)
CS Only w/- α -Cut = 0.99	91.17%	17.36%	-8.33% (-32.4%)
with Component Processing^{^^}			
Maximum Response Analysis	91.81%	15.97%	-9.72% (-37.8%)
MLP w/ Unsorted Confidence Field	76.62%	50.00%	+24.31% (+94.6%)
MLP w/ Sorted Confidence Field	87.30%	27.08%	+1.39% (+5.4%)
Raw Component Counts	87.36%	22.92%	-2.78% (-10.8%)
Locally Clustered Component Counts	92.14%	15.28%	-10.42% (-40.5%)

** Error Rate = (FP + FN) / (TP + FN)

^{^^} from EV_32, EV_64, EV_128, TR_32

Chapter 4

EVALUATION OF FUZZY INTEGRAL DATA FUSION METHODS FOR RARE OBJECT DETECTION IN RGB HIGH-RESOLUTION SATELLITE IMAGERY

The research presented in this chapter is primarily taken from a published paper presented at the IEEE BIGDATA 2020 Conference [22]. Additional information is provided herein on concepts that were unable to be published because of the IEEE conference page constraint. Co-authors included Curt H. Davis and A.J. Malentfort. Also included in this chapter is context background scene confirmation that was also cut from the conference paper due to space constraints.

4.1 INTRODUCTION

The objective of this research is to develop, test, refine, and then combine/integrate decision-level fusion methods to improve machine-assisted analytic workflows by aggregating detections in both space and time from multiple Deep Neural Networks (DNN) to reduce scene and/or object detection error rates.

In this research we developed and tested unique technical approaches that combined results from multiple DNN detectors to improve the detection of *Engineering Vehicles (EV)* superclass in the public domain benchmark xView dataset. These were:

1. Fusion of DNN *EV* object detections from multiple DNN architectures.
2. Fusion of DNN *EV* object detections with additional DNN detectors designed to separately detect local *EV* scene/background context.

The *EV* class was selected because of its very low occurrence, e.g. scarce/rare object, where *EV* samples represent only $\sim 0.8\%$ of the 581,953 sample objects in the xView dataset. Several advanced fusion strategies were explored and tested for each of the three technical approaches listed above. These included the Sugeno and Choquet fuzzy integrals implemented with a variety of different training/learning methods. A non-public xView validation dataset (provided by NGA) was used to independently assess overall performance of the different techniques and fusion strategies.

Results from 1) demonstrated that fusing multi-DNN architectures can achieve a maximum reduction in relative *error rate per square kilometer* (*EpSK*) for *EV* detection by up to $\sim 90\%$ with a corresponding $\sim 19\%$ absolute reduction in the recall or *True Positive Rate* (*TPR*). While a more modest $\sim 17\%$ *EpSK* reduction can be obtained with only a $\sim 4\%$ loss in *TPR*. In each case there was a clear trade-off between optimizing the relative *EpSK* error reduction vs. maintaining the *TPR* performance. But in either case, the reduction in *EpSK* was $\sim 4X$ greater than the loss in *TPR* performance.

Results from 2) demonstrated that fusing local *EV* background/context from a single DNN background detector with the DNN *EV* object detections can achieve a maximum *EpSK* reduction up to $\sim 45\%$ with a corresponding $\sim 9\%$ loss in *TPR*. While fusing background detections from multiple DNN architectures could obtain a $\sim 44\%$ reduction in *EpSK* with a $\sim 15\%$ loss in *TPR*. In total, these results demonstrate that fusing local/scene background with a DNN object detector can achieve significant reduction in *EpSK* with only modest loss in the *TPR*.

4.2 DATA FUSION TECHNIQUES

Multiple data sources can provide more information such that the combination, aggregation, or fusion of these data can result in a more informed decision. Several fusion techniques were tested in our experiments in order to determine which techniques were beneficial for DNN object detection in high resolution commercial satellite images. A given fusion technique was used on the DNN inference outputs produced by scene scanning covered in Section 2.3.2. Most of the fusion methods covered in this chapter used three detection sources ($n = 3$), except for a subset of the background confirmation experiments which utilized six detection sources ($n = 6$).

We first employed two basic fusion techniques. These were the arithmetic mean or average (i.e. $x' = \frac{1}{n} \sum_{i=1}^n x_i$) which equally weights all sources, and arrogance (i.e. $x' = \max([x_1, \dots, x_n])$) which takes the maximum inference response from the input detection sources. A shortcoming of the average is that it has the potential of eliminating good responses from two or more sources based on the poor performance from a single source. Arrogance overcomes this shortcoming but in general can produce more false positive detections.

More advanced fusion techniques try to leverage source-specific performance to improve the fusion outcome. In addition to the simple average and arrogance techniques this research explored the use of fuzzy integrals for multi-source detection fusion. Specifically, the Sugeno (SI) [47] [48] and the Choquet (CI) [49] fuzzy integrals were used in all the experiments. The CI is flexible and general enough that with an appropriately constructed FM lattice, the CI can be used to compute the average, arrogance, softmax, order weighted average (OWA), and many other weighted average approaches.

4.2.1 Fuzzy Integral

The fuzzy integral uses a combination of statistics and logic to try to determine the best final output based on the combined information from a) the initial DNN detection outputs, which are the integral source inputs (δ), and b) the overall confidence in the DNN source models themselves. The source confidences or permuted combination of source confidences are better known as Fuzzy Measures (FM) and are arranged into the FM lattice, g , which is described in more detail in Section 4.2.2. Source permutations, X , for $n = 3$ sources can be denoted thusly: $\{\{\emptyset\}, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}\}$ where each permutation has an associated FM within g .

4.2.1.1 Sugeno Integral

The SI, S_g , computes the conservative intersection between the sorted order, or the π index order of inputs δ_π , and the FM for running combined outputs in g , to obtain the conservative intersection between the two shown in Eq. 1 and Fig. 4.1.

$$S_g = \max(\min(\delta_{\pi_i}, g(A_{\pi_i}))) \text{ where } A_{\pi_i} \subseteq X \text{ s.t } x_{\pi_k} \text{ for all } k \leq i \quad (1)$$

4.2.1.2 Choquet Integral

The output for SI is limited because it only includes the input from the sources (δ) and values of g that are included in the π index order (A_π). The Choquet Integral (CI), C_g , takes a weighted-sum approach to fuse the input values, where the weights are based on the confidence of the sources. The Riemann sum approach to computing the integral is reflected in this approach, as the CI uses the sum of the difference between δ_π as the width of the subintervals and uses $g(A_\pi)$ as the height (Eq. 2).

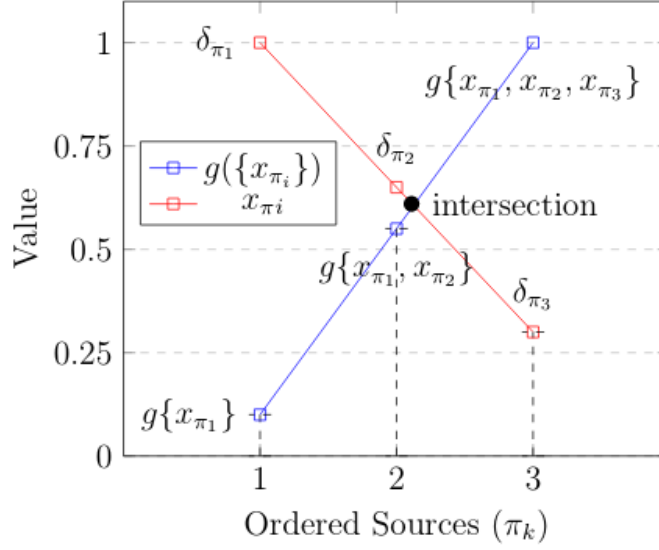


Fig. 4.1: Visualization of a Sugeno Integral. Dashed lines go up to the minimum between the input value, δ , and the corresponding g for the commutative subset if X . The black dot is the intersection between the two sets of data. The intersection is closest to the values as π_2 . Therefore the conservative estimation would be $g\{x_{\pi_1}, x_{\pi_2}\}$.

$$C_g = \sum_{i=1}^n [\delta_{\pi_i} - \delta_{\pi_{i+1}}] \cdot g(A_{\pi_i}) \text{ where } A_{\pi_i} \subseteq X \text{ s.t. } x_{\pi_k} \text{ for all } k \leq i \text{ and } \delta_{\pi_{n+1}} = 0 \quad (2)$$

Consequently, the more statistical approach used in the CI may have the ability to produce higher accuracies than the SI in some cases due to this flexibility.

4.2.2 Fuzzy Measures

As shown above, the effectiveness of the fuzzy integral is completely dependent upon the constitution of g . In general, g consists of the $2^X - 1$ crisp sets of X . When three source inputs are combined there are then eight crisp subsets including the empty and exhaustive sets: $\{\{\emptyset\}, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}\}$. As mentioned above, g is a lattice of running confidences for the individual and the combinations of a given set of inputs (Fig. 4.2).

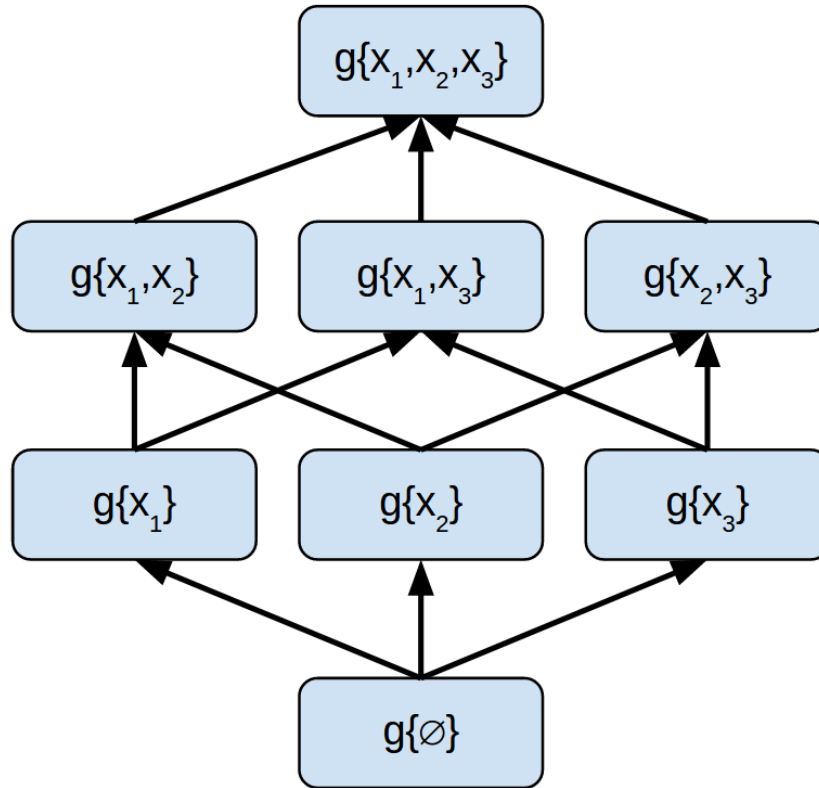


Fig. 4.2: Fuzzy measure lattice, g , showing edges used to “move through” the lattice while calculating the fuzzy integral in π index order. Note that the each set in X is order independent, i.e. $g\{x_1, x_2\} = g\{x_2, x_1\}$.

For most computations of g in this report, the following constraints are enforced to bound and create a bounded monotonicity within g :

1. $g \in [0, 1]$
2. $g(\emptyset) = 0$
3. $g(X) = 1$
4. $g(B) \leq g(A)$ if $B \subseteq A$

Table 4.1: Details of FM approaches used in the experiments. The Mobius transform was not used beyond the DNN architecture experiments because the unbounded nature created a large number of false positives. Also, neither of the ChiMP approaches were used in the MS experiments because there were training issues that were not resolved within the research time constraints.

Experiment	Abbreviation	Multi-DNN Architecture	Background Confirmation -One GSD	Background Confirmation -Two GSD
Number of DNN Input Sources		3	3	6
FM Approach				
Average	AVG	X	X	X
Arrogance	ARR	X	X	X
Quadratic Programming	QP	X	X	X
Sugeno-λ				
$F1$ Score	$F1$	X	X	X
Normalized $F1$ Score	$F1/\#$	X	X	X
Combining Sugeno-λ& QP				
% Sub. (QP & $F1$)	25P $F1$	X	X	X
% Sub. (QP & $F1/\#$)	25P $F1/\#$	X	X	X
Average (QP & $F1$)	AVG $F1$	X	X	X
Average (QP & $F1/\#$)	AVG $F1/\#$	X	X	X
ChiMP				
Mobius Transform	M (other FM)	X		
iChiMP	iChiMP	X	X	X

4.2.2.1 Sugeno- λ FM

The Sugeno- λ approach seeks to normalize the input source confidences or densities, d , of the source models by calculating the constant λ which effectively determines a solution of $\lambda > -1$ for the following equation:

$$(1 + \lambda) = \prod_{i=1}^n (1 + \lambda d_i), \text{ where } n \text{ is the number of sources.} \quad (3)$$

With λ , g is computed using set theory union principles in the following manner:

1. The FM of each source; $[g\{x_1\}, g\{x_2\}, g\{x_3\}] = [d_1, d_2, d_3]$
2. The FM of the union of two sources; $g\{x_i, x_j\} = g_\lambda(d_i \cup d_j) = d_i + d_j - \lambda(d_i d_j)$
3. The union can then be extended for the next step of three sources as follows;

$$g\{x_i, x_j, x_k\} = g_\lambda(g_\lambda(d_i \cup d_j) \cup d_k) \quad (4.1)$$

$$= (d_i + d_j - \lambda(d_i d_j)) + d_k - \lambda((d_i + d_j - \lambda(d_i d_j)) d_k) \quad (4.2)$$

$$= d_i + d_j + d_k - \lambda(d_i d_j + d_i d_k + d_j d_k) + \lambda^2 d_i d_j d_k \quad (4.3)$$

4. The union can be extended in a similar manner as item 3 for each permutation until g is completed.

It should be noted that Eq. 4.3 can be used to construct the results for any permutation of X and this, therefore, demonstrates the communicative property of the Sugeno- λ FM.

The computed $F1$ scores of the concatenated 5-fold results were used as d . Additional FM lattices were computed after normalizing the source densities (i.e $d' = d/n$). These additional experiments are included in the ranking tables for the results of each section of experiments.

4.2.2.2 Quadratic Programming FM

There are also ways of calculating g directly from the data instead of using the densities and computing everything from the overall performance of the 5-fold results. The data-driven approach utilizes every individual response from the blind concatenated 5-fold responses to calculate the confidence of the model based on its collective performance.

The Quadratic Programming (QP) approach uses linear algebra to solve for g directly from the development data. In this case, the data used are the blind concatenated outputs from the 5-fold experiments (see Section 4.5.2.1). The QP approach solves the following linear equation:

$$E = g^t \left(\sum_{k=1}^n A_k A_k^t \right) g - \left(\sum_{k=1}^n (-2) \Gamma A_k^t \right) + \Gamma^2 \quad (4.4)$$

where A is a sparse representation of the differences of δ for each sample in π index order and Γ is a representation of data labels. Where, in the case of $n = 3$ sources, we define A_i as follows:

$$A_i = \begin{bmatrix} \dots \\ \delta_{i,\pi_1} - \delta_{i,\pi_2} \\ \dots \\ \delta_{i,\pi_2} - \delta_{i,\pi_3} \\ \dots \\ \delta_{i,\pi_3} \end{bmatrix} \quad (4.5)$$

π is the descending order index of the δ_i . The positions are filled in correspondence to the index of g such that $\delta_{i,\pi_1} - \delta_{i,\pi_2}$ is in the position of $g(\{x_{\pi_1}\})$, $\delta_{i,\pi_2} - \delta_{i,\pi_3}$ is in the position $g(\{x_{\pi_1}, x_{\pi_2}\})$, and δ_{i,π_3} is in the position of $g(\{x_{\pi_1}, x_{\pi_2}, x_{\pi_3}\})$. All other values are set to 0.

Since g is constant if we let $D = \sum_{k=1}^n A_k A_k^t$ and $Z = \sum_{k=1}^n (-2) \Gamma A_k^t$ and we can

rewrite Eq. 4.4 as:

$$E = g^t Dg - Zg + \Gamma^2 \quad (4.6)$$

Keeping in mind the *FM* constraints outlined in Section 4.2.2, Equation 4.6 can be solved for g .

4.2.2.3 Combining FMs

Although statistically superior to the Sugeno- λ approach, one shortcoming of the quadratic formula approach is that if a certain source permutation in X does not have enough representation from the π index order during computation, then it may have difficulty learning the FM for g . In this research we conducted experiments to combine the macro Sugeno- λ and the micro QP approaches to overcome the shortcomings of each. The first approach implemented a simple average between the two lattices. Since each method falls within the constraint criteria, it can be shown that the simple average preserves the criteria. The following proofs shows that averaging FMs which adhere the FM criteria in Section 4.2.2 also adhere to the criteria:

1. If $g_S \in [0, 1]$ and $g_d \in [0, 1]$ then $g_S + g_d \in [0, 2]$, thus $(g_S + g_d)/2 \in [0, 1]$
2. If $g_S(\emptyset) = 0$ and $g_d(\emptyset) = 0$ then $(g_S(\emptyset) + g_d(\emptyset))/2 = (0 + 0)/2 = 0$
3. If $g_S(X) = 1$ and $g_d(X) = 1$ then $(g_S(X) + g_d(X))/2 = (1 + 1)/2 = 1$
4. If $g_S(\{B\}) \leq g_S(\{A\})$ and $g_d(\{B\}) \leq g_d(\{A\})$ for $\{B\} \subseteq \{A\}$,
then $g_S(\{B\}) + g_d(\{B\}) \leq g_S(\{A\}) + g_d(\{A\})$ for $\{B\} \subseteq \{A\}$

The second method deals with the possible lack of representation in the π index order by measuring how many times a specific FM in d is represented and then replacing the FM from the quadratic formula approach with the FM from the Sugeno- λ approach. For the experiments in this research, substitutions were used for anything

less than 25% representation in the π index. If both g hold to the FM criteria in Section 4.2.2 we know that the criteria 1-3 are preserved in substitution. However, in this effort, we were not able to completely eliminate the instances of violation for FM criteria 4 with this substitution approach.

4.2.2.4 Improved Choquet Integral Multi-layer Perceptron (iChIMP) FM

While the QP approach brings in more information to compute d , g is still bound to a linear model. By learning weights of a neural network using the concatenated 5-fold results as inputs, non-linear solutions can be computed. A simple example of how a non-linear solution can improve classification is shown in Fig. 4.3. Taking from Islam *et. al.* [50], the iChiMP approach computes g by constructing a multi-layer perceptron with an architecture that mimics the CI with the values in g as the computed weights between layers.

4.2.2.5 Mobius Transform FM

The Mobius transform approach takes g computed by some other approach and converts it using the mobius transform. This transform also introduces non-linearity but can also be super-additive and so can violate FM criteria 3 in Section 4.2.2. The Mobius transform is a Choquet integral Multi-layer Perceptron (ChiMP) that is a less sophisticated alternative to iChiMP as discussed in [50]. This technique was added mainly out of curiosity. Mobius transformed FMs were only processed using the CI.

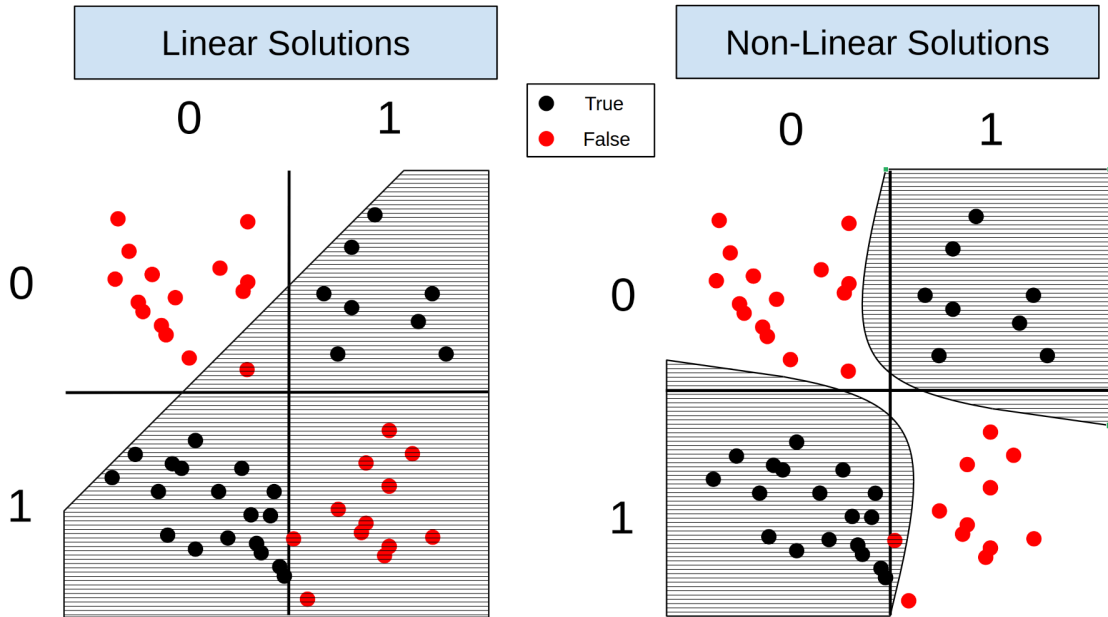


Fig. 4.3: The two graphs illustrate the difference between linear and non-linear solutions for the XOR problem. The dotted lines in the left graph show 4 possible linear solutions for decision boundaries, but none of these decision boundaries serve the data well. The graph on the right shows a non-linear solution for the same data, where the same data becomes much more separable.

4.3 EXPERIMENTAL OVERVIEW

The *EV* class from the public-domain xView benchmark dataset was selected for all the fusion experiments. The *EV* class was selected because of its very low occurrence, e.g. scarce/rare object, where *EV* objects represent only $\sim 0.8\%$ of the 581,953 sample objects in the xView dataset. Further the xView dataset has many other ground vehicles (e.g. *Trucks*, *Railroad Vehicles*, etc.) and other object classes (e.g. *Helicopter*, *Shipping Container*, *Pylon*, etc.) that present many potential challenging “confuser” objects for *EV* detection.

All the results from the data fusion experiments are compared to an *EV* baseline detection result. Baseline *EV* results were generated by processing the 8-bit xView RGB validation scenes using trained ProxylessNAS DNN models for each model size.

Results are compared using the *TPR*, harmonic average of precision and recall (F-score), *EpSK*, and a new metric: the Estimated Observation to Target Ratio (*EOTR*). The *EpSK* was computed by summing the false positives and false negatives (missed *EV* detections) and dividing by the total calculated area of the validation data (287 km²). The *EOTR* is the estimated ratio between the number of raw *EV* detections a human analyst would have to review before being presented with a true positive *EV* detection.

Below we provide an overview for each experiment while the detailed experimental results are provided in their companion sections later in this chapter. Results are also presented using the pooled results for the different input model sizes for a given GSD. Baseline (no fusion) experimental results were generated from pooled results of the ProxylessNAS models and are created for both 0.3 m and 0.5 m GSDs. These were then used to compare the performance of the fusion experiment results relative to the baseline result for the same GSD.

4.3.1 DNN Model Training

DNN detectors were independently trained for each of three partitioned image datasets using transfer learning from the pre-trained RGB model weights. Training for all the DNNs models utilized transfer learning from ImageNet [35], Adam [36] for optimization, and cross entropy for the objective function.

4.3.2 Multi-DNN Architecture Fusion

DNN models from disparate architectures can presumably learn different features for the same dataset since they employ different neural network building blocks. Consequently, it has been observed that fusing the object detections of multiple DNN models can increase recall and reduce false positives compared to any single-architecture approach [1]. In this research we fused the inference results from DNN processing

(as described in Section 4.2.1) of the xView validation scenes using three disparate DNN model architectures: Xception [32], NASNet [30], and ProxylessNAS [33]. The xView validation scenes were then scanned with the three DNNs to generate initial “candidate” *EV* locations that were then used as inputs for the fusion experiments.

4.3.3 Object Background Confirmation

It has been hypothesized that the context provided by the image scene around a localized object can be used to help determine and/or confirm the identity of the object. One approach to do this takes *EV* candidate locations produced by individual DNN models and/or fused *EV* object detections and then uses detected scene backgrounds to determine if the initial *EV* candidate locations should persist. This is done by retrieving background image samples from the validation scenes for each *EV* candidate and processing the background sample through DNN models trained to specifically detect *EV* backgrounds. Note that specific object background scene types (e.g. parking lot, grass, road, construction site, etc.) do not necessarily need to be detected. Thus, a binary scene background detector can be trained to recognize a variety of *EV* backgrounds based on the training data and then used without having to specifically identify or classify different background scene types.

We performed background confirmation experiments that used both single DNN BG detector inputs and fused detection results from the Xception, NASNet, and ProxylessNAS architectures. Experiments also include combining results from 0.3 m and 0.5 m image GSDs by using: a) binary logic for single architecture or fused-architecture results; and b) using a six-source input that aggregates the detections from three DNN object detection architectures and two scene background GSDs as fuzzy integral inputs.

To simplify these experiments only candidates produced by the ProxylessNAS *EV* models were used as baseline *EV* candidate locations. Further, all BG confirmation

fusion experiments used 0.5 m GSD ProxylessNAS *EV* models for producing the baseline *EV* candidate locations.

4.4 DATA SOURCE

The dataset used in this study is from the “DIUx xView 2018 Detection Challenge” [45] and used the same data sets described in Section 3.2.2 for the *EV* experiments. Again object size was based on the xView dataset using bounding box Maximum Pixel Length (MPL) and used to determine which feature/object samples would be used to train a DNN model for a given input window size. The initial xView data had an assumed nominal GSD of 0.3 m which was used to create the *EV* object datasets used in the DNN architecture fusion and background confirmation experiments (Sections 4.5 & 4.6). This assumption was carried forward when re-sampling the scenes to create the 0.5 m GSD datasets used in the same experiments.

However, during the course of this research, it was discovered that the nominal 0.3 m GSD assumption was incorrect when attempting to process the xView validation scenes. Inconsistencies were found in the location metadata for the scanning chip center point locations. Consequently, a corrective approach was implemented to re-sample the validation images to a desired “true” or “correct” GSD.

4.4.1 xView Scene GSD Correction

As mentioned previously, the published xView dataset had an assumed nominal GSD of 0.3 m [45]. However, the image GSD within an xView scene could differ between the N/S and the E/W directions (see Fig. 4.4). Metadata provided by xView included the latitude and longitude coordinates for the scene corners of each scene which we used to calculate the actual scene width and height in meters. These measurements were then divided by the width and height of the image in pixels to produce an image size that would have approximately the correct GSD. After this correction was

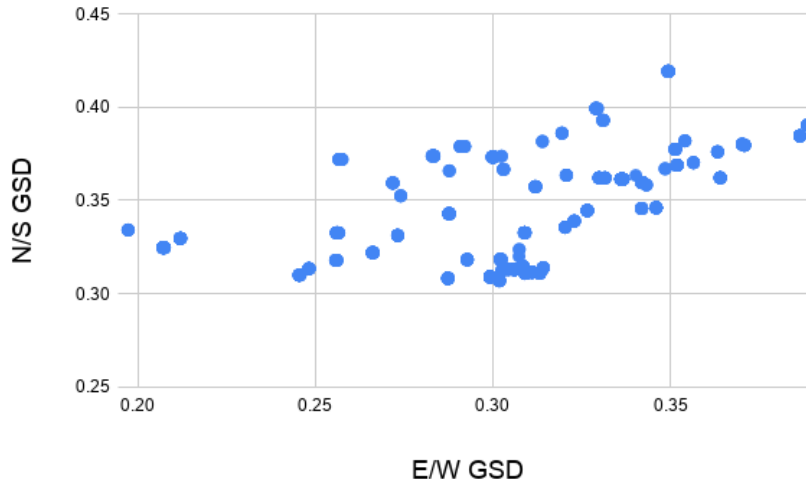


Fig. 4.4: Scatter plot of the calculated vertical and horizontal GSDs of the xView scenes.

applied, each xView scene was resampled to produce both 0.3 m or 0.5 m GSDs for use in the research experiments.

4.5 MULTI-DNN ARCHITECTURE FUSION EXPERIMENTS

Models from disparate DNN architectures can presumably learn different features for the same dataset since they employ different neural network building blocks. In this research we fused inference results from DNN processing of the xView validation scenes using three disparate DNN model architectures: Xception, NASNet, and ProxylessNAS. Each of the DNN models from these architectures were pre-trained on the ImageNet [35] dataset and then batch trained for 1 epoch using the datasets described in Section 4.5.1. Then the xView validation scenes were scanned with the three trained DNNs to generate initial “candidate” *EV* locations used as inputs for the fusion experiments. The scanning results for the individual architectures and fused results were then locally clustered before the evaluation metrics were computed.

These experiments were designed to detect/locate 1,601 *EV*s contained within

281 xView validation image scenes. In order to achieve this, each xView validation scene followed the same basic processing steps (more details are provided in later subsections):

1. GSD correction/resampling to 0.3 m or 0.5 m GSD (see Section 4.4.1).
2. Scanning for each *EV* model size with 25% stride (75% overlap).
3. Fusion of results from the three different DNN architectures from scanning output centerpoints.
4. Local clustering performed on the scanning results to produce cluster centers or *EV* candidate locations after lower-confidence detections below an α -cut.
5. True Positives (*TP*) and False Positives (*FP*) were computed using resulting cluster centers in spatial proximity to known *EV* locations to calculate the *TPR*, precision or Positive Predictive Value (*PPV*), *F1* score, and *EOTR*.

4.5.1 Datasets

As mentioned previously, these experiments used xView images with the assumed 0.3 m GSD (i.e. without GSD correction) and also re-sampled for 0.5 m GSD under the same assumption. Because *EVs* can range from small excavators to large cranes, DNN model inputs used were square samples of size 32, 64, and 128 pixels for both 0.3 m and 0.5 m GSD image scenes. Negative samples for each dataset were created using objects from balanced, non-*EV* object classes. Online sample augmentation (Appendix A.2) was used to increase post sample counts to approximately four million samples. Training sample counts can be found in Table 4.2.

Validation was completed by scanning the 281 xView validation scenes where each scene had ~ 1 square km extent. All validation scenes were corrected to achieve nominal 0.3 m and 0.5 m GSDs (see Section 4.4.1).

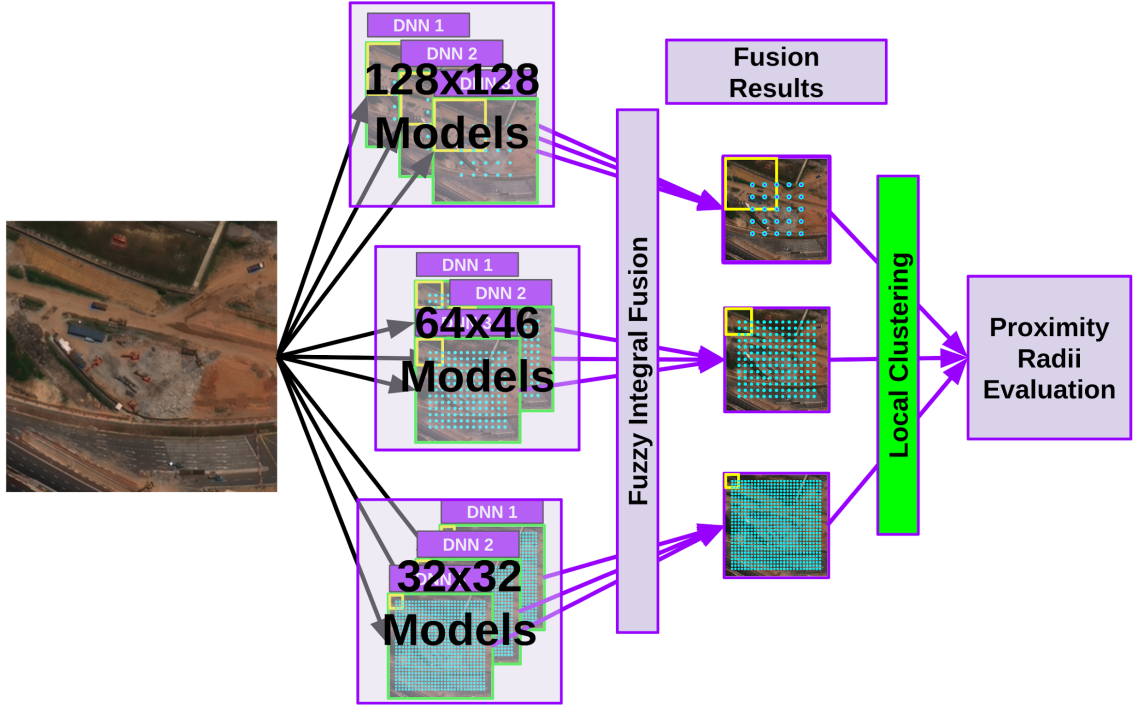


Fig. 4.5: Flowchart for multi-DNN architecture fusion experiments.

Table 4.2: Sample sizes for *Engineering Vehicles* used in DNN architecture fusion experiments.

		Input DNN Model Size		
GSD (m)	Sample Type	32	64	128
0.3	True Positive	2,535	4,101	1,985
0.3	True Negative	10,140	16,404	7,940
0.5	True Positive	3,900	2,918	532
0.5	True Negative	15,600	11,672	2,128

4.5.2 Experimental Design

4.5.2.1 5-fold Cross Validation Experiments

Initial 5-fold cross validations (Section 2.3.1) were completed for each *EV* sample dataset for all experiments. Each fold was trained using flip and 5° rotation augmentations to produce a 144X online augmentation (Appendix A.2). F-scores from the concatenated outputs of the 5-fold experiments were used as source densities (d) for constructing the FM lattice (g). The concatenated 5-fold outputs were also used to

Table 4.3: Scanning stride, local clustering apertures, and proximity radii used to calculate and validate candidate *Engineering Vehicle* locations.

DNN Model Size (pixels)	Stride (meters)	Local Clustering Aperture (meters)	Proximity Radius δ_k (meters)
0.3 m GSD			
32	2.4	3.6	4.8
64	4.8	7.2	9.6
128	9.6	14.4	19.2
0.5 m GSD			
32	4	6	8
64	8	12	16
128	16	24	32

train the data-driven approaches for computing g .

4.5.2.2 Scanning and Local Clustering

As mentioned above, scanning of the xView validation scenes was performed for each DNN architecture with input model sizes of 32, 64, and 128. Scanning and clustering were completed as described in Section 2.3.2 using an α -cut of 0.99 with strides and aperture radii provided in Table 4.3.

4.5.2.3 Data Fusion

Utilization of multiple data sources can provide more information such that aggregation or fusion of these multiple data sources can result in a more informed decision [1]. Experiments fusing the softmax scanning outputs from three different DNN architectures (i.e. Xception, NASNet, and ProxylessNAS) were performed. The fusion techniques and FM lattice types are found in Section 4.2.1 and Table 4.1 respectively.

4.5.2.4 Determining True and False Positive Detections

After clustering a proximity radius for model size k , δ_k , is used to determine if an *EV* has been successfully detected or *TP* detection and, conversely, if a candidate

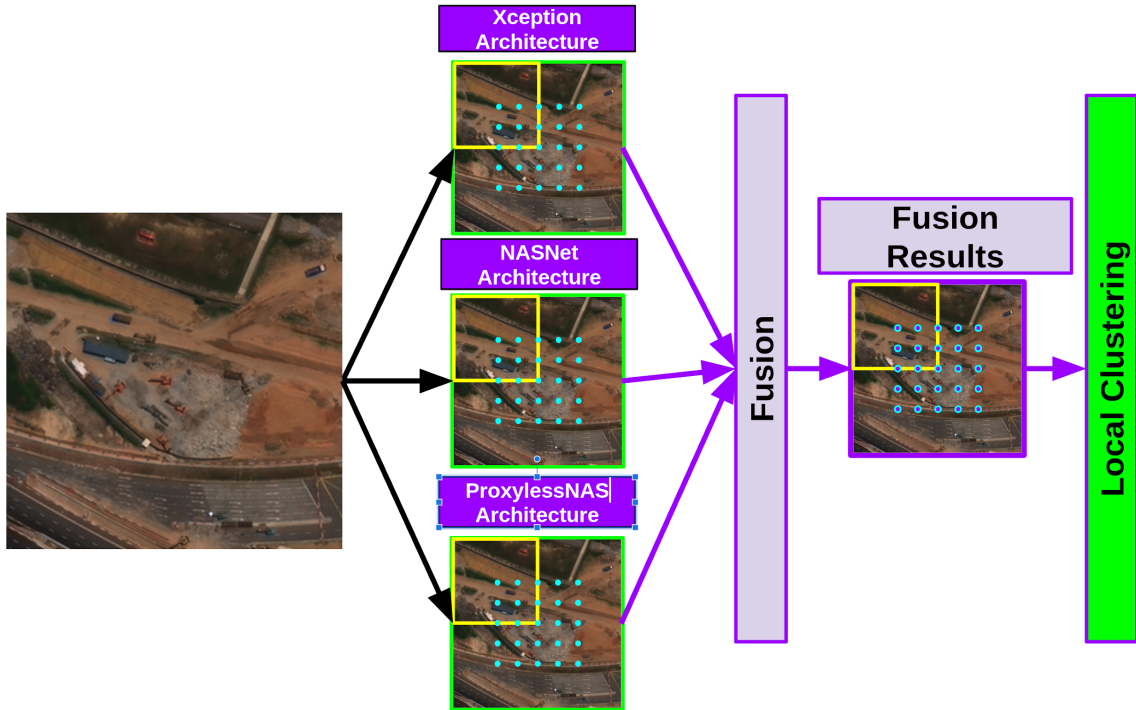


Fig. 4.6: Processing flowchart for fusion of multi-DNN architecture scanning outputs.

location is a *FP*. The proximity radii used in these experiments are provided in Table 4.1. To determine the *TPR*, the distance is computed for each candidate location vs. each of the 1601 known *EV* locations in the xView validation scenes. If the candidate *EV* location is within δ_k , then the known *EV* was found and the *EV* is counted as a *TP*. If a candidate’s minimum distance to a known *EV* is larger than δ_k then it is considered a *FP*. An illustration of this process is shown in Fig. 4.7. Proximity radii are provided in Table 4.1.

4.5.3 Results

4.5.3.1 5-fold Experiments

Results from the 5-fold *EV* experiments in Table 4.4 show *F1* scores between 77-95% for all models. Surprisingly, the Xception models outperformed all the other model architectures in terms of *F1* score for every GSD and input model size. However,



Fig. 4.7: *EV* candidate locations (cyan crosses) and proximity radii (red circles) used to validate candidate locations for 32, 64, and 128 pixel (left to right) DNN models. Note that DNN models with size 32 and 128 would have been detected for this *EV* object as there are detections within the proximity radii for the respective size.

Xception is also the most computationally expensive. So we selected the ProxylessNAS model for its faster training speed because of the large number of models and experiments we conducted in this research. Therefore, ProxylessNAS results were used as the baseline for comparing the performance of fusion methods. The NASNet and Xception models were used as additional architectures for the multi-architecture fusion experiments. The 5-fold *F1* scores for the various model architectures were used as densities to compute the values in the FM lattice.

4.5.3.2 *EV* Detection

This research first looked at how the *EV* models for different GSDs and input size performed against each other. These results only used the trained ProxylessNAS model due to time constraints. The results in Table 4.5 show that the 0.3 m GSD models usually, but not always, had better *TPR* than the 0.5 m GSD models. However, the higher *TPR* comes at the expense of $\sim 3X$ increase in the *EpSK*. By pooling the results from all three model sizes, the *TPR* increases dramatically compared to the results from any single model. The remainder of the experiments are presented in terms of pooled size results and compared to the pooled ProxylessNAS DNN model

Table 4.4: Results from 5-fold experiments for *EV* object detection models using the ProxylessNAS, NASNet, and Xception model architectures for 0.3 m and 0.5 m GSD and various model sample sizes.

Architecture	Size	GSD (m)	<i>TPR</i>	<i>PPV</i>	<i>F1</i>
ProxylessNAS	32	0.3	81.7%	91.6%	86.4%
NASNet	32	0.3	68.5%	89.4%	77.6%
Xception	32	0.3	87.1%	90.8%	88.9%
ProxylessNAS	64	0.3	89.1%	93.4%	91.2%
NASNet	64	0.3	90.8%	92.9%	91.8%
Xception	64	0.3	92.5%	93.3%	92.9%
ProxylessNAS	128	0.3	93.1%	94.7%	93.9%
NASNet	128	0.3	92.7%	94.7%	93.7%
Xception	128	0.3	92.4%	96.1%	94.2%
ProxylessNAS	32	0.5	85.7%	82.4%	84.0%
NASNet	32	0.5	84.2%	87.8%	85.9%
Xception	32	0.5	85.6%	91.0%	88.2%
ProxylessNAS	64	0.5	87.6%	91.2%	89.4%
NASNet	64	0.5	82.2%	91.4%	86.6%
Xception	64	0.5	91.4%	92.0%	91.7%
ProxylessNAS	128	0.5	86.7%	87.0%	86.8%
NASNet	128	0.5	85.8%	90.5%	88.1%
Xception	128	0.5	90.8%	89.7%	90.2%

Table 4.5: Comparative metrics for each ProxylessNAS *EV* model for sizes 32, 64, and 128 and 0.3 m and 0.5 m GSDs.

Model Size	GSD	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	Error/ km ²	<i>EOTR</i>
32	0.3	55.2%	0.8%	1.6%	386	126
64	0.3	67.9%	4.2%	7.9%	88.5	23.9
128	0.3	69.6%	15.8%	25.7%	22.5	6.3
Pooled	0.3	85.8%	1.0%	1.9%	492	104
32	0.5	59.2%	2.6%	5.0%	126	38.5
64	0.5	62.6%	11.3%	19.2%	29.5	8.9
128	0.5	29.6%	28.1%	28.8%	8.2	3.6
Pooled	0.5	79.4%	2.8%	5.3%	157	36.1

results for a given GSD.

The results in Table 4.6 show the comparison of the different results for the different architectures. The ProxylessNAS models had the highest *TPR* and the NASNet models achieved the highest *F1* scores. The Xception models presented the lowest

Table 4.6: Validation scanning results for the pooled model sizes for all three DNN architectures. Red shows the scanning results for ProxylessNAS models that are used as a baseline for comparison with the fusion experiments.

Model Size	GSD	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>EpSK</i>	<i>EOTR</i>
ProxylessNAS	0.3	85.8%	1.0%	1.9%	492	104
Xception	0.3	84.0%	0.7%	1.5%	638	137
NASNet	0.3	75.8%	2.8%	5.4%	148	35.8
ProxylessNAS	0.5	79.4%	2.8%	5.3%	157	36.1
Xception	0.5	77.3%	1.9%	3.7%	224	52.7
NASNet	0.5	72.8%	5.8%	10.8%	67.3	17.2

PPV and *F1* scores which is the opposite of what was seen in the 5-fold experiments. This may be an indication of overfitting in the Xception models.

Previously it was mentioned that the *TPR*, *F1* score, *EpSK*, and *EOTR* metrics would be used to assess the experimental results. After analyzing some of the results, it was found that using the highest *F1* score and lowest *EpSK* and *EOTR* tended to produce the same rankings. So for the rest of this experiments, we present two tables for each set of results: the first highlighting the fusion and FM methods that produce the highest *F1* score and the second highlighting the *TPR* results.

The results in Table 4.7 show that SI fusion achieved the highest *F1* scores. This included SI with Sugeno- λ with normalized densities used to compute the lattices (*F1/3*) and/or combining the QP and Sugeno- λ FM lattices (25P *F1/3*, AVG *F1*, AVG *F1/3*). The *EpSK* was reduced by $\sim 87\%$ for 0.3 m GSD and $\sim 79\%$ for 0.5 m GSD with an $\sim 19\%$ absolute reduction in *TPR*. Fusion using SI produce the top results is counter to our hypothesis when we first introduced the SI and CI (see Section 4.2.1). This may indicate that the *F1* scores from the 5-fold experiments were “overconfident” in predicting the performance of the trained models on the validation scene scanning images. Normalizing these densities seems to have “grounded” these confidences while preserving the relative confidence magnitude between the DNN models. Thus, normalization may represent something closer to the true confidences of the system.

Table 4.7: Top $F1$ results for multi-DNN architecture fusion experiments. Red indicates comparative baseline. A few extra methods were included for the 0.5 m GSD results.

Fusion	FM	GSD (m)	TPR	PPV	$F1$	$EpSK$	$EOTR$
ProxylessNAS		0.3	85.8%	0.96%	1.9%	492	104
SI	25P $F1/3$	0.3	67.0%	7.4%	13.3%	48.6	13.5
SI	AVG $F1/3$	0.3	67.0%	7.4%	13.3%	48.6	13.5
SI	$F1/3$	0.3	67.0%	7.4%	13.3%	48.6	13.5
CI	25P $F1/3$	0.3	69.6%	5.6%	10.3%	67.9	18.0
CI	$F1/3$	0.3	69.5%	5.5%	10.1%	68.9	18.3
ProxylessNAS		0.5	79.4%	2.8%	5.3%	157	36.1
SI	AVG $F1$	0.5	60.3%	13.4%	22.0%	23.9	7.5
SI	AVG $F1/3$	0.5	60.3%	13.4%	22.0%	23.9	7.5
SI	25P $F1/3$	0.5	60.2%	13.4%	21.9%	23.9	7.5
SI	$F1/3$	0.5	60.2%	13.4%	21.9%	23.9	7.5
SI	QP	0.5	60.2%	13.4%	21.9%	23.9	7.5
CI	$F1/3$	0.5	62.8%	10.5%	18.0%	32.0	9.5
SI	$F1$	0.5	63.3%	10.5%	18.0%	32.2	9.5

Table 4.8 presents the fusion results that achieved the best TPR . It is no surprise to see arrogance (ARR) as the top fusion technique even over the ProxylessNAS baseline result, but it also had $\geq 2X$ the $EpSK$ for both GSDs which is a significant drawback of this simple fusion approach. Some of the best results were achieved by the CI with the Mobius-transformed normalized $F1$ score FM ($M(F1/3)$) for 0.3 m GSD. This yielded only a $\sim 4\%$ absolute TPR sacrifice to achieve a 15% error reduction. Also iChiMP was effective in producing a high TPR for the 0.3 m GSD and the Xception models alone also produce a high TPR . Looking beyond the Mobius transform results (not provided in the table), the CI also produced the top TPR for fused results, opposite of what was shown in Table 4.7. This may be a manifestation of the more conservative nature of the SI when selecting the fusion output. Ideally the fusion process would have produced results that increased TPR and significantly reduced the error as shown in Chapter 2. However, this outcome was not present in the results from this comprehensive set of multi-DNN architecture fusion experiments

Table 4.8: Top *TPR* results for DNN architecture fusion experiments. Red shows comparative baseline. Baselines are placed within their ranked position and included with the rest of the results.

Fusion	FM	GSD (m)	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>EpSK</i>	<i>EOTR</i>
ARR	N/A	0.3	91.8%	0.53%	1.1%	959	188
ProxylessNAS		0.3	85.8%	1.0%	1.9%	492	104
CI	iChiMP	0.3	85.6%	1.0%	2.0%	480	101
Xception		0.3	84.0%	0.7%	1.5%	638	137
CI	25P <i>F1</i>	0.3	81.7%	1.0%	2.0%	480	101
CI	M(<i>F1</i> /3)	0.3	81.6%	1.1%	2.2%	407	90.1
CI	M(AVG <i>F1</i> /3)	0.3	81.6%	1.1%	2.1%	419	92.8
ARR	N/A	0.5	86.9%	1.5%	2.9%	323	67.4
ProxylessNAS		0.5	79.4%	2.8%	5.3%	157	36.1
CI	M(AVG <i>F1</i> /3)	0.5	77.5%	2.7%	5.2%	157	37.0
Xception		0.5	77.3%	1.9%	3.7%	224	52.7
CI	M(<i>F1</i> /3)	0.5	77.3%	2.8%	5.3%	154	36.4
CI	M(QP)	0.5	76.9%	3.0%	5.8%	139	33.1

for the challenging *EV* object class.

4.6 BACKGROUND CONFIRMATION

This set of experiments leveraged the local context of a scene background (BG) around the object to confirm the *EV* detections. Adding a layer onto the previous approach, this process takes the *EV* candidate locations produced by the ProxylessNAS models at 0.5 m GSD (Section 4.5.3.2) and processes the BG samples of those locations to see if each should persist as a candidate (Fig. 4.8). BG samples are retrieved from the xView validation scenes for each candidate using the cluster center as a center point for the sample. Experiments included generating results using single DNN architectures and fusing results from Xception, NASNet, and ProxylessNAS architectures. A threshold value of 0.5 was used as a background confidence thresholds (θ in Fig. 4.8) for most of the BG experiments. Experiments also included: i) confirming the 0.3 m GSD scan results with the 0.5 m GSD BG samples and vice versa, ii) combining confirmations results from both GSDs using AND and OR binary logic, and iii) fusing results from all six possible *EV* BG models for a given model size

(i.e. using the responses from three architectures and both GSDs as fuzzy integral inputs).

The BG confirmation steps were added between the local clustering and evaluation steps (i.e. 4a-4d below) described in Section 4.5. Baselines are placed within their ranked position and included with the rest of the results.

1. Vertical and horizontal GSD correction to 0.3 m or 0.5 m GSD (see Section 4.4.1)
2. Scanning for each *EV* model size with 75% overlap (25% stride).
3. Fusion of scanning results from different DNN architectures per scanning field point.
4. Post α -cut, local clustering on scanning results to produce cluster centers.
 - (a) For each cluster center, BG sample(s) are extracted from the validation scene with the appropriate size and GSD.
 - (b) Each BG sample is processed using the appropriate *EV* BG model(s).
 - (c) If applicable, employ fusion methods to combine results. DNN architectures were fused using both SI and CI and FM computed in the approaches noted in Table 4.1.
 - (d) Determine if each candidate location should persist as a candidate. The persistence of a candidate was tested using single GSD BGs and combining BGs of different GSDs using an AND or OR binary logic.
5. True Positives (*TP*) and False Positives (*FP*) were computed using resulting cluster centers compared to known *EV* locations to calculate the *TPR*, *PPV*, *F1* score, *EpSK*, and *EOTR*.

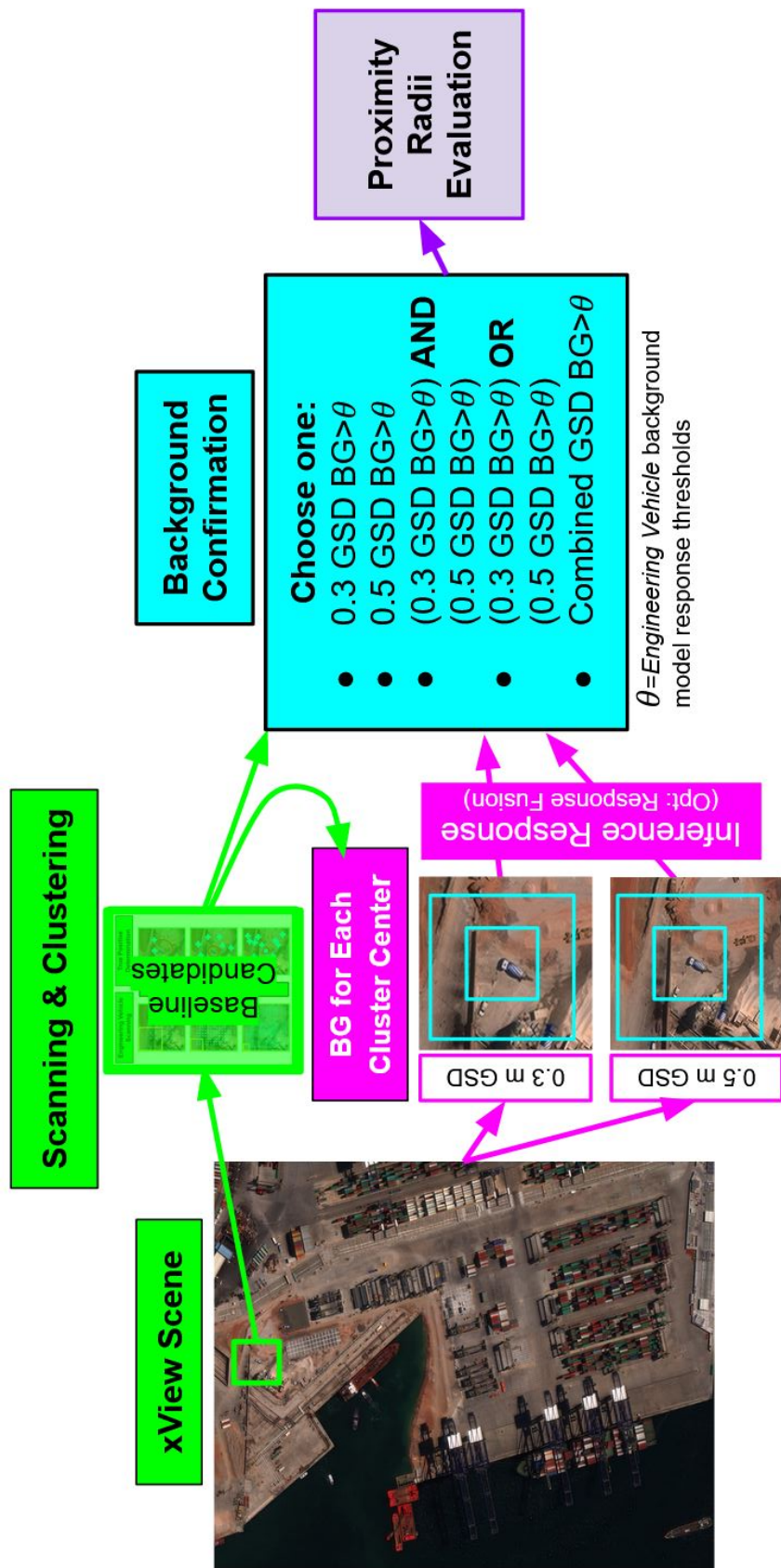


Fig. 4.8: Flow chart showing the process of searching for EVs with the addition of local scene BG confirmation. The initial candidate selection as described in Section 4.5 is shown in green. The pink shows the BG sampling with inference processing and EV BG DNN model response fusion. The BG confirmation options shown in blue include considering BGs from multiple GSDs.

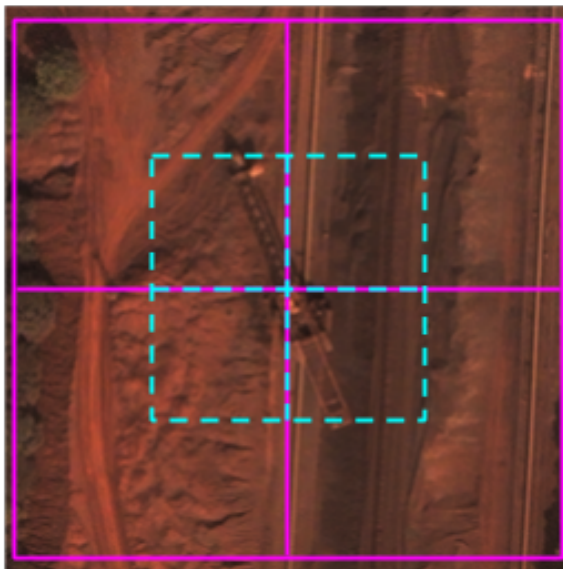


Fig. 4.9: Example of how *EV* local background (BG) samples are created around an xView *EV* object. Magenta lines are crop lines for 128x128 pixel samples and cyan lines are crop lines for 64x64 pixel samples. The same process was used for both 0.3 m and 0.5 m GSD datasets.

4.6.1 Datasets

Positive samples of local scene backgrounds (BGs) around *EV*s were created by extracting four samples (Fig. 4.9) for all *EV* objects from the xView training scenes. A dataset of negative samples was created by random selection of xView feature/object classes that were greater than 100 m away from any *EV* feature and then extracting samples in the same manner as positive samples. Overabundant xView classes (e.g. *Building*, *Small Car*) were reduced to 100,000 samples before the selection of xView features for negative sample objects. Online augmentations (Appendix A.2) using image rotations and flips were used during training to increase the training sample counts by 144X. Datasets were created for both 0.3 m and 0.5 m GSD and for chip sizes 64 and 128. Final sample counts are provided in Table 4.8.

Table 4.9: Sample counts used for *EV* BG experiments. Because xView object locations can be located on or slightly off the edge of a provided xView scene, pure black samples might be produced, and, if so, these were removed from the datasets and appear as slight differences in counts between different datasets.

	Input DNN Model Size			
	64	128	64	128
Sample Type	0.3 m GSD		0.5 m GSD	
True Positive	19,174	19,184	19,182	19,188
True Negative	76,720	76,756	76,726	76,749

4.6.2 Results

4.6.2.1 5-fold Experiments

The 5-fold results for these experiments do not show a clear favorite amongst the model architectures that were tested. The Xception models seem to have higher metrics for the 0.3 m GSD datasets, whereas the NASNet model showed higher metrics for the 0.5 m GSD datasets (Table 4.10). As before, the *F1* scores from the 5-folds were used to compute the FM lattices.

Table 4.10: 5-fold experiments for *EV* background models.

Architecture	Size	<i>TPR</i> <i>PPV</i> <i>F1</i>			<i>TPR</i> <i>PPV</i> <i>F1</i>		
		0.3 m GSD			0.5 m GSD		
ProxylessNAS	64	74.8%	83.0%	78.7%	76.3%	84.1%	80.0%
NASNet	64	76.2%	87.0%	81.3%	78.5%	86.3%	82.2%
Xception	64	76.6%	84.4%	80.3%	77.3%	85.4%	81.2%
ProxylessNAS	128	80.4%	87.4%	83.8%	81.3%	87.3%	84.2%
NASNet	128	82.0%	88.4%	85.1%	83.1%	89.8%	86.3%
Xception	128	82.4%	89.1%	85.6%	83.0%	88.7%	85.8%

4.6.2.2 Single Model BG Confirmation

One thing to remember for the BG confirmation experiments is that these can do no better than the baseline *TPR* since it uses the baseline *EV* candidate locations as a starting point. Thus, this can only be used to reduce false positive error rates. Ideally the process would maintain high *TPR* and *F1* scores while significantly reducing the

Table 4.11: Results for BG confirmation using single-sized BG with the same GSD, different GSD, and combining GSDs through OR and AND binary logic. A confidence value of 0.5 was used as the background confirmation threshold.

<i>EV</i> GSD (m)	BG GSD (m)	BG Size	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>EpSK</i>	<i>EOTR</i>
0.3	ProxylessNAS		85.8%	0.96%	1.9%	492	104
0.3	0.3	64	73.6%	1.8%	3.5%	225	55.4
0.3	0.3	128	69.1%	2.2%	4.2%	175	45.9
0.3	0.5	64	63.0%	2.3%	4.5%	149	42.8
0.3	0.5	128	61.5%	2.3%	4.4%	149	43.7
0.3	OR	64	78.0%	1.6%	3.2%	267	62
0.3	OR	128	74.5%	1.9%	3.6%	220	53.6
0.3	AND	64	59.1%	3.1%	5.8%	107	32.8
0.3	AND	128	56.5%	3.0%	5.7%	104	33.1
0.5	ProxylessNAS		79.4%	2.8%	5.3%	157	36.1
0.5	0.5	64	62.0%	5.3%	9.7%	64.2	18.9
0.5	0.5	128	56.3%	5.0%	9.2%	61.8	19.9
0.5	0.3	64	30.9%	3.2%	5.9%	55.2	30.9
0.5	0.3	128	30.0%	3.3%	6.0%	52.7	30.2
0.5	OR	64	67.3%	4.2%	7.9%	87.3	23.8
0.5	OR	128	60.4%	4.1%	7.7%	80.4	24.2
0.5	AND	64	25.2%	4.8%	8.1%	32.0	20.8
0.5	AND	128	25.6%	4.6%	7.8%	33.6	21.6

EpSK and *EOTR*. Table 4.11 shows that the smallest decrease in *TPR* for both GSDs is for BG models of size 64 that combine the results using OR logic. For 0.3 m GSD this process reduces the *EpSK* by $\sim 45\%$ with a *TPR* sacrifice of only $\sim 9\%$, while for 0.5 m GSD this reduces the *EpSK* by $\sim 44\%$ for a *TPR* sacrifice of $\sim 15\%$.

4.6.2.3 BG Confirmation with Fusion

To simplify the study, all BG fusion was completed using ProxylessNAS *EV* models at 0.5 m GSD from Section 4.5 as a baseline and running BG confirmation on the *EV* candidates produced by those models. Thus, there is no differentiation between baseline GSDs in the tables as in earlier experiments. The results from Table 4.11 for *EV* GSD of 0.5 m were included when computing the top results shown in Table

Table 4.12: Top $F1$ score results for BG fusion experiments. Red indicates comparative baseline. Note that non-fused results of 0.5 m GSD were also included in the ranking.

Fusion	FM	BG GSD (m)	BG Size	TPR	PPV	$F1$	$EpSK$	$EOTR$
ProxylessNAS (0.5 m)		N/A	N/A	79.4%	2.8%	5.3%	157	36.1
CI	AVG $F1$	0.3	128	67.0%	5.9%	10.8%	64.3	17.0
CI	AVG $F1/3$	0.3	128	61.2%	5.6%	10.2%	60.3	18.0
CI	$F1/3$	0.3	128	60.7%	5.5%	10.1%	60.3	18.2
SI	$F1/6$	COM	128	55.2%	5.5%	10.0%	55.2	18.1
SI	AVG $F1/3$	0.3	128	61.6%	5.4%	9.9%	62.6	18.6
SI	QP	0.3	128	61.6%	5.4%	9.9%	62.6	18.6
AVG	N/A	0.3	128	61.3%	5.4%	9.8%	62.7	18.7
CI	$F1/6$	COM	128	57.0%	5.3%	9.7%	59.0	18.8
CI	QP	0.3	128	61.5%	5.2%	9.6%	64.8	19.2

4.12 and 4.13 .

From Table 4.12 it can clearly be seen that the higher-resolution BG samples of 0.3 m GSD achieve a higher accuracy either alone or combined with the 0.5 m GSD. As seen in previous experimental results, fusion techniques using normalized $F1$ scores for densities performed well here. However the top $F1$ score came from combining the QP and Sugeno- λ using the non-normalized $F1$ score which reduced the $EpSK$ by $\sim 59\%$ with a TPR sacrifice of $\sim 16\%$ and a over a 50% $EOTR$ reduction. Also worth noting is the QP approach computing the FM lattice for both the SI and CI and the 0.5 m BG of size 64 which, by most metrics, fared similarly to the top ranked approach. This time there is a bit of ambiguity of whether SI or CI is a more effective approach.

In Table 4.13 it can clearly be seen that the higher-resolution BG samples of 0.3 m GSD achieve a higher accuracy either alone or combined with the 0.5 m GSD. This time using SI seemed to have a slight advantage to maintain TPR compared to CI. However, the top results seem to just persist most of the candidates which would indicate very little advantage. The most interesting results among the top 10 from using the iChiMP FM approach processed through CI. This approach reduced the $EpSK$ by $\sim 25\%$ with a TPR sacrifice of only 2.5% or 97% retention.

Table 4.13: Top *TPR* results for BG fusion experiments. Red indicates comparative baseline. Note that non-fused results of 0.5 m GSD were also included in the ranking.

Fusion	FM	BG GSD (m)	BG Size	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>EpSK</i>	<i>EOTR</i>
ProxylessNAS (0.5 m)		N/A	N/A	79.4%	2.8%	5.3%	157	36.1
ARR	N/A	COM	64	78.9%	2.9%	5.6%	147	34.2
SI	<i>F1</i>	COM	64	78.7%	3.0%	5.7%	146	33.9
SI	AVG	0.3	64	78.6%	3.0%	5.7%	146	33.7
SI	iChiMP	0.3	64	78.6%	3.0%	5.7%	146	33.7
CI	<i>F1</i>	COM	64	78.6%	3.0%	5.7%	144	33.6
CI	<i>F1</i>	0.3	64	78.5%	3.0%	5.8%	143	33.3
ARR	N/A	0.3	64	78.5%	3.0%	5.8%	143	33.4
SI	<i>F1</i>	0.3	64	78.1%	2.9%	5.7%	145	34.0
CI	AVG <i>F1</i>	0.3	64	77.0%	3.2%	6.2%	131	31.0
CI	iChiMP	0.3	64	76.8%	3.3%	6.3%	126	30.1
CI	AVG	COM	64	76.8%	3.5%	6.8%	118	28.3

4.7 CONCLUSION AND FUTURE WORK

In this research we developed and tested two unique technical approaches that combined results from multiple DNN detectors to improve the detection of *EV* in the public domain benchmark xView dataset. These were:

1. Fusion of DNN *EV* object detections from multiple DNN architectures.
2. Fusion of DNN *EV* object detections with additional DNN detectors designed to separately detect local *EV* scene/background context.

Several advanced fusion strategies were explored and tested for both technical approaches listed above. These included the Sugeno and Choquet fuzzy integrals implemented with a variety of different training/learning methods. The non-public xView validation dataset was used to independently assess overall performance of the different techniques and fusion strategies.

The best fusion results from 1) and 2) demonstrated that the relative *EpSK* for *EV* detection could be reduced 45-90% with corresponding absolute reduction in the *TPR* in the range of 9-19%. The multi-DNN architecture fusion in 1) produced the greatest *EpSK* reduction (85%) but also the largest *TPR* loss (19%), while the local

scene/background fusion in 2) had much lower loss in *TPR* (9%) but still produced a very good reduction in *EpSK* (45%).

Future work worth pursuing could include: A) applying these fusion techniques to a variety of other rare object classes, B) exploring more sophisticated ways of combining Sugeno- λ and data-driven fuzzy measure lattices, and C) scaling up these experiments to broad area scanning ($>1,000 \text{ km}^2$) for rare objects under more realistic operational scenarios (e.g. mission-relevant AOIs).

Chapter 5

EVALUATION OF FUZZY INTEGRAL DATA FUSION METHODS FOR RARE OBJECT DETECTION IN 8-BAND MULTI-SPECTRAL HIGH-RESOLUTION SATELLITE IMAGERY

This work in this chapter is primarily taken from a paper published at the IEEE BIG-DATA 2020 Conference [22]. Additional information is provided herein on concepts that were unable to be published because of the IEEE conference page constraints. Co-authored include Curt H. Davis and A.J. Malentfort.

5.1 INTRODUCTION

The objective of this research is to develop, test, refine, and then combine/integrate a variety of decision-level fusion methods to improve machine-assisted analytic workflows by aggregating detections in both space and time from a variety of different DNNs to reduce errors, increase confidence, and improve human analytic performance. The focus of this effort was to develop and test methods to fuse scanning results from multiple DNN detectors to improve rare object detection by fusing object detections from multiple DNNs derived from three partitions of 8-band Multi-Spectral (MS) imagery (e.g. DigitalGlobe WV3 satellite).

Just as fusing different DNN architectures can increase the amount of information for a system to learn, MS imagery may also provide additional information beyond

the standard RGB bands that are widely used for DNN processing of commercial satellite imagery. Consequent, we developed experiments that partitioned 8-band MS imagery used to create the xView dataset into a set of three 3-band images.

The fusion of three 3-band MS images showed an improved result compared to those in Chapter 4. The results demonstrated that recall or *True Positive Rate (TPR)* gains up to 5% could be achieved while at the same time reducing the *Error per Square Kilometer (EpSK)* by $\sim 20-60\%$. The best results were generated using the Choquet Integral (CI) with Sugeno- λ computed fuzzy measures. This shows that the additional information provided in 8-band multispectral imagery can be leveraged using pre-existing, pre-trained 3-band RGB DNN models to achieve significant error reduction while maintaining or even increasing the *TPR*.

5.2 EXPERIMENTAL OVERVIEW

These experiments involved partitioning 8-band MS xView imagery (WorldView-3 satellite) into a set of three 3-band images. Separate scanning results from the individual 3-band partitions are then fused in an attempt to reconstitute the information from the original 8 MS bands. The MS fusion experiments closely follow the DNN architecture fusion experiments presented in Chapter 4. However, the MS band partitioning is added to the process below as Step 0.

0. Scene - bit conversion and MS band partitioning.
1. Vertical and horizontal GSD correction to 0.3 m or 0.5 m GSD (Section 4.4.1)
2. Scanning for each *EV* model size with 75% overlap (25% stride).
3. Fusion of scanning results from the three different band partitions per scanning field point. The MS partition results were fused using both Sugeno Integral (SI) and CI with Fuzzy Measures (FM) computed in the approaches noted in Table 5.1.

Table 5.1: Details of FM approaches used in the experiments. The Mobius transform was not used beyond the DNN architecture experiments because the unbounded nature created a large number of false positives. Also, neither ChiMP approach was used in the MS experiments b/c there were training issues that were not resolved within time constraints. Note that this is an expansion from Table 4.1

Experiment	Abbreviation	Multi-DNN Architecture	Background Confirmation -One/Two GSD	Multi-Spectral
Number of DNN Input Sources		3	3/6	3
FM Approach				
Average	AVG	X	X	X
Arrogance	ARR	X	X	X
Quadratic Programming	QP	X	X	X
Sugeno-λ				
<i>F1</i> Score	F1	X	X	X
Normalized <i>F1</i> Score	<i>F1</i> /#	X	X	X
Combining Sugeno-λ & QP				
% Sub. (QP & <i>F1</i>)	25P <i>F1</i>	X	X	
% Sub. (QP & <i>F1</i> /#)	25P <i>F1</i> /#	X	X	
Average (QP & <i>F1</i>)	AVG <i>F1</i>	X	X	X
Average (QP & <i>F1</i> /#)	AVG <i>F1</i> /#	X	X	X
ChiMP				
Mobius Transform	M (other FM)	X		
iChiMP	iChiMP	X	X	

4. Local clustering was performed on the scanning results to produce cluster centers or *EV* candidate locations after lower-confidence detections below an α -cut.
5. True Positives (*TP*) and False Positives (*FP*) were computed using resulting cluster centers compared to known *EV* locations to calculate the *TPR*, precision or Positive Predictive Value (*PPV*), *F1* score, and Estimated Observation to Target Ratio (*EOTR*).

Models were trained in a similar manner described in Section 4.3.1 with data processing similar to that described in Section 4.5.2 (i.e. cross validation, scanning, clustering, fusion, and determining true and false detection).

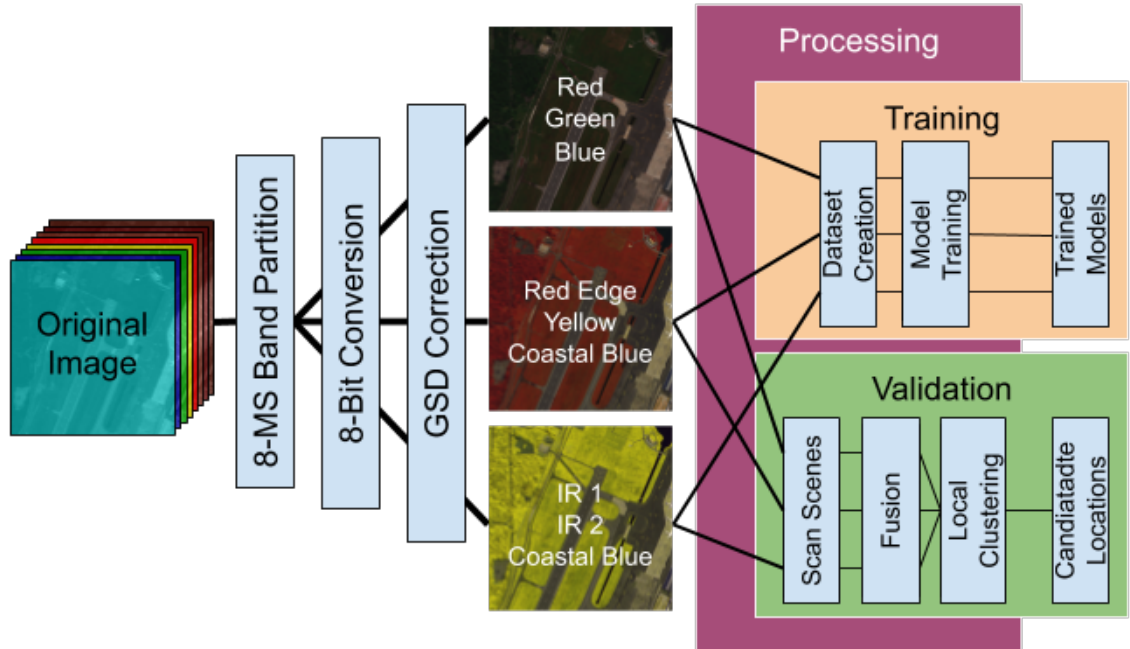


Fig. 5.1: Flow chart of 8-MS band partitioning for the training and validation datasets.

5.3 DATASETS

These experiments partitioned the 8-band MS imagery used to create the xView competition imagery into a set of three 3-band images to leverage commonly used 3-band transfer learning model training approaches. The transfer learning weights used seeds to train all the DNN models came from pre-training on an the ImageNet [35] dataset which is a 3-band RGB imagery with a bit depth of 8 bits. The MS xView dataset has 8 spectral bands with a bit depth >13 . Ideally the imagery would be trained from scratch for an accommodating 8-band model, but this would be very time consuming in both refactoring the DNN architectures as well as for the additional training time. In order to take advantage of the same transfer learning used in the preceding experiments, the MS imagery was partitioned into three 3-band images. After band partitioning, the bit depth was mapped from >13 bits to 8 bits (see below) followed by the GSD correction.

Some previous experiments using solely RGB imagery showed little improvement in results when using *EV* model sizes of 256. However, because MS imagery contains information outside of the RGB band, the 256-model size was reintroduced for these experiments along with the 32, 64, and 128 sizes.

5.3.1 Multi-Spectral Band Partitioning

The 8-band MS images were partitioned into three sets of 3-band images:

1. Red, Green, and Blue (RGB)
2. Edge Red, Yellow, and Coastal Blue (ERYC)
3. Infrared 1, Infrared 2, and Coastal Blue (IR12C)

Mace *et al.* [51] replicated the weights for a pre-trained RGB model to accommodate the additional MS bands for a single DNN network. The replication of the weights by Mace *et al.* [51] informed how we partitioned the 8-band MS images into three sets of 3-band images as shown above. However, in our approach the pre-trained RGB weights were in effect re-used (copied) in each of the three 3-band partitioned DNN models.

5.3.2 Bit Conversion

Fortunately, having both the xView MS imagery and the xView competition imagery on hand provides a good reference for converting from the 13+ bit MS data to the desired 8-bit data. The mapping from 13+ bit to 8-bit pixel values was not many-to-one as expected, but actually produced overlapping pixel ranges for each 8-bit pixel value. A piece-wise linear fit for the mapping was used to create a many-to-one mapping for conversion from 13+ bit to 8 bit imagery. Conversion maps were created for the RGB bands for each scene (Fig. 5.3).

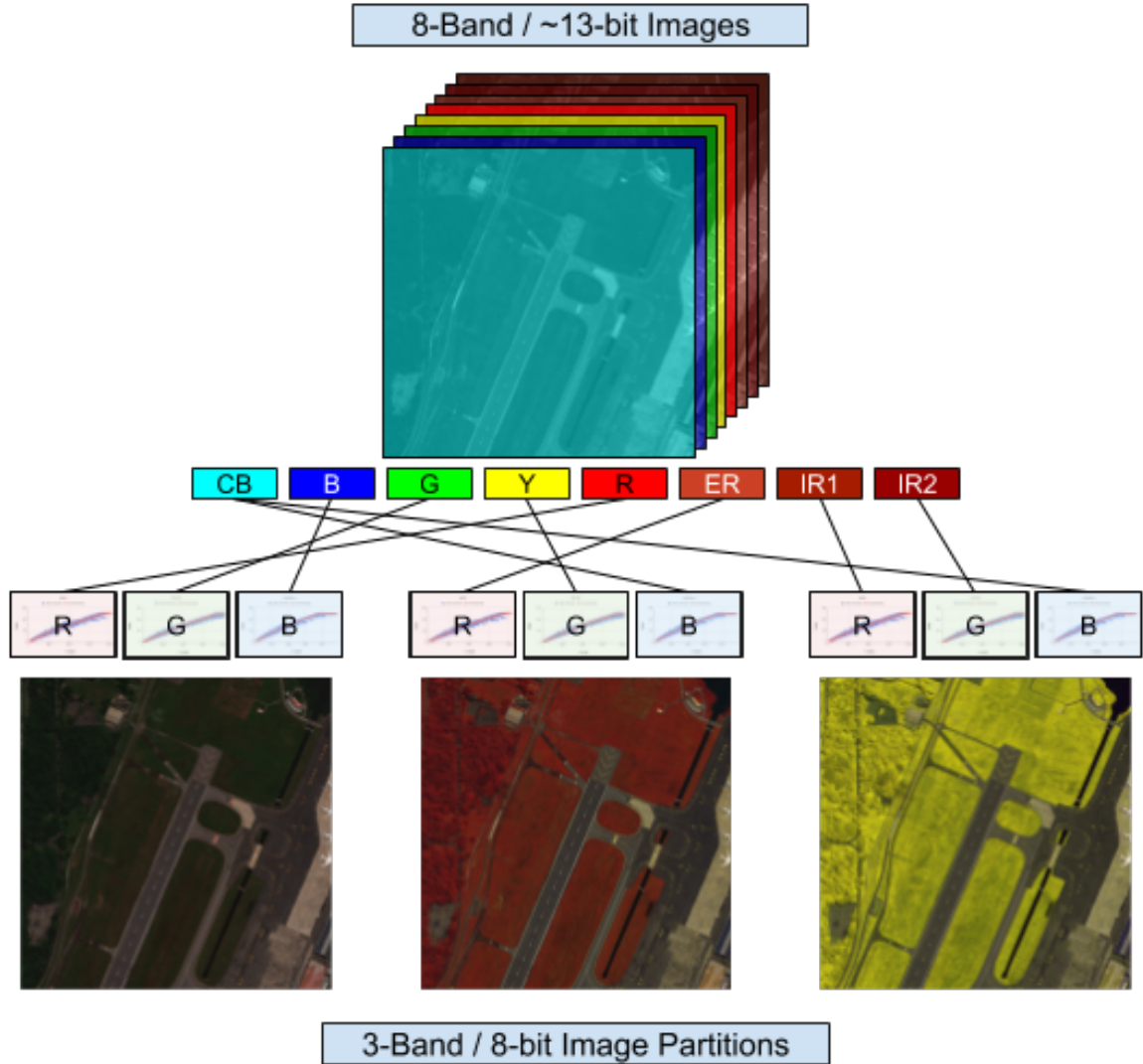


Fig. 5.2: Illustration of the 8-MS band partitions and conversion from 13+ bit pixel depth to 8-bit pixel depth.

We then used the bit conversion from the RGBs band to convert the additional MS bands in each partition. Red, Edge Red, and Infrared 1 were converted using the Red channel conversion, Green, Yellow, Infrared 2 using the Green channel conversion, and Blue, Coastal Blue using the Blue channel conversion as depicted in Fig. 5.2.

Table 5.2: Sample counts for *EV* object training samples after GSD correction used in 8-band training.

Sample Type	GSD (m)	Input DNN Model Size			
		32	64	128	256
True Positive	0.3	1,939	4,087	2,583	700
True Negative	0.3	7,756	16,346	10,339	2,815
True Positive	0.5	3,850	3,458	8,48	416
True Negative	0.5	15,399	13,835	3,399	1,670

5.3.3 Updated *Engineering Vehicle* Training Samples

The *EV* datasets were created similarly to *EV* datasets described in Section 4.5.1. However, these datasets had coinciding samples across the three different 3-band partitions created for the MS experiments. Given the need for scene GSD correction, the opportunity was taken to construct new sample datasets with corrected object sizes based on the re-sampled scenes for corrected GSD (Section 4.4.1). Sample sets were created for each model-size, GSD, and 3-band partition. Online augmentation (Appendix A.2) was also used during training to increase the training samples to approximately six million samples. Table 5.2 provides the sample counts for all the MS experiment datasets.

5.3.4 Validation

Validation was completed on the 281 xView validation scenes with ~ 1 km square extent. All validation scenes were converted to 8 bit pixel depth and the GSDs were corrected to both 0.3 m and 0.5 m.

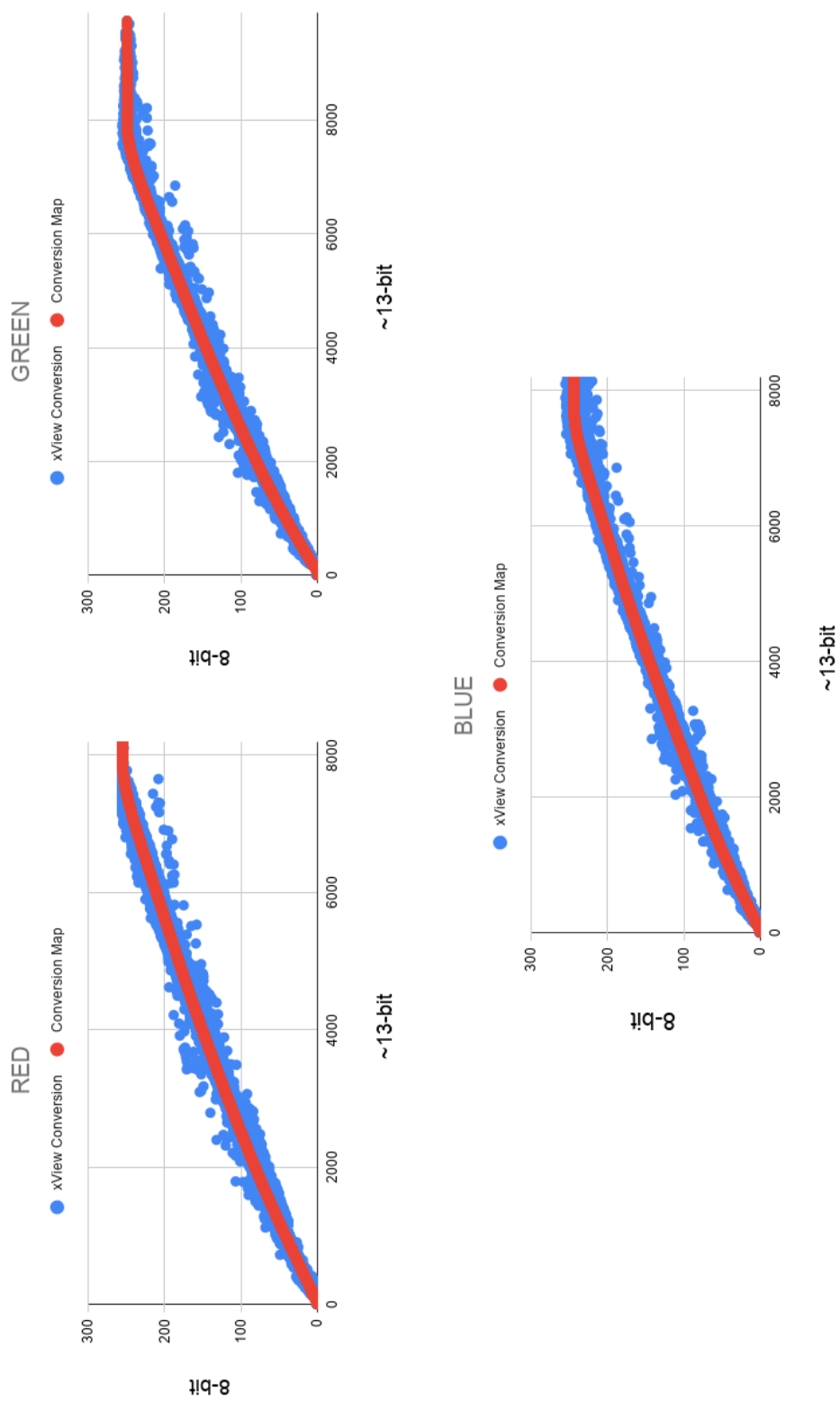


Fig. 5.3: Conversion of 13+ bit MS imagery (red) to the 8-bit xView competition imagery (blue). Note the overlapping blue values. The red is the many-to-one mapping used for bit conversion of each band using a piece-wise linear fit of the actual mapping. These mappings were used to convert the RGB partition and the respective bands of the other partitions.

Table 5.3: 5-fold *EV* comparison of the original xView competition imagery and the in-house, bit-converted RGB results.

Dataset	GSD (m)	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>EpSK</i>	<i>EOTR</i>
Original 8-bit RGB	0.3	85.8%	1.0%	1.9%	492	104
Bit-Converted RGB	0.3	86.0%	1.7%	3.4%	277	58.5
Original 8-bit RGB	0.5	79.4%	2.8%	5.3%	156.7	36.1
Bit-Converted RGB	0.5	92.1%	3.9%	7.6%	125.8	25.4

5.4 RESULTS

The MS experiment analyses did NOT use the same baseline for comparison as the previous experiments. Instead, a new baseline was developed using the ProxylessNAS models for the band-partitioned, bit-converted xView validation scenes. Interestingly the MS baseline results shown in Table 5.3 outperformed the competition data baselines (also Table 5.3) for every metric. This may be a result of the number of online augmentations used in the training, the fact that the MS dataset was corrected to the corrected GSD prior to creating the *EV* datasets, or it may also be an artifact of the different ways the xView scenes were converted to RGB. Although both the xView competition imagery and the research-generated RGB partitioned imagery are derivatives of the 13+ bit multi-spectral imagery produced by the DigitalGlobe WV3 satellite, differences in the conversion processes can introduce unexpected noise which may help or hinder the ability of a DNN to train more robustly.

5.4.1 5-fold Experiments

5-fold results provided in Table 5.4 show that the *F1* scores for *EV* detection varied between 85% and 94% . Other than size 128 models performing slightly better than the other models for a given GSD, there really is not anything that stands out with these results.

Table 5.4: 5-fold *EV* experiments for research-generated MS band partitioned models.

Partition	Size	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>TPR</i>	<i>PPV</i>	<i>F1</i>
		0.3 m			0.5 m		
RGB	32	84.8%	85.9%	85.4%	84.1%	89.4%	86.7%
ERYC	32	83.6%	87.0%	85.3%	84.1%	90.6%	87.2%
IR12C	32	84.2%	85.1%	84.7%	82.5%	90.3%	86.2%
RGB	64	86.1%	94.1%	89.9%	89.1%	92.7%	90.9%
ERYC	64	88.4%	93.2%	90.7%	91.3%	91.6%	91.4%
IR12C	64	88.1%	93.4%	90.7%	90.0%	91.7%	90.8%
RGB	128	92.7%	93.9%	93.3%	90.2%	92.8%	91.5%
ERYC	128	91.9%	94.8%	93.3%	88.0%	94.5%	91.2%
IR12C	128	91.5%	93.5%	92.5%	90.4%	92.3%	91.3%
RGB	256	89.6%	93.6%	91.5%	88.7%	92.2%	90.4%
ERYC	256	89.0%	95.3%	92.0%	90.8%	88.3%	89.5%
IR12C	256	90.1%	91.3%	90.7%	90.1%	88.2%	89.2%

5.4.2 Engineering Vehicle Detection Experiments

Table 5.5 provides the validation scanning results for the individual band partitions. The results show that some of the ERYC and IR12C models yielded higher *TPRs* than the baseline RGB results, but this coincides with *EpSK* increases of 30-65% for ERYC and \sim 130% for IR12C. The RGB partitions did better across the board for *F1*, *EpSK*, and *EOTR*. This is likely due to the fact that the transfer learning used to prime the DNN models was only trained on RGB bands.

Table 5.6 presents the top MS-partition fusion experiments ranked by *F1* score. These are the highest *EV* scanning *F1* scores seen thus far compared to both the multi-DNN and scene BG fusion experiments presented in Chapter 4 (see Tables 4.7 and 4.12). For 0.3 m GSD images, an 85% reduction in *EpSK* is achieved with only a 12% absolute *TPR* sacrifice using SI, while for 0.5 m GSD images SI achieved an 84% *EpSK* reduction with 11% *TPR* sacrifice. Other experiments required a 20-30% absolute loss in *TPR* to achieve the same *EpSK* reduction. Note also that the ranking of the fusion technique results are sorted approximately in the same order for both GSDs (recognizing that the top fusion techniques for each GSD have the same score).

Table 5.5: *TPR* score results for individual MS band partition experiments both with and without pooling the 256-model size. Red indicates comparative RGB baseline.

Partition	GSD (m)	<i>TPR</i>	<i>PPV</i>	<i>F1</i>	<i>EpSK</i>	<i>EOTR</i>
Pooled results from model sizes 32, 64, & 128						
RGB	0.3	88.3%	1.8%	3.5%	275	56.7
ERYC	0.3	92.8%	1.4%	2.8%	366	71.6
IR12C	0.3	93.0%	0.8%	1.6%	646	125
RGB	0.5	92.1%	3.9%	7.5%	126	25.4
ERYC	0.5	93.0%	2.5%	4.9%	203	40.7
IR12C	0.5	90.3%	3.3%	6.3%	150	31.0
Pooled results from model sizes 32, 64, 128, & 256						
RGB	0.3	94.6%	1.8%	3.6%	283	54.6
ERYC	0.3	96.0%	1.4%	2.8%	372	70.4
IR12C	0.3	95.5%	0.8%	1.6%	654	124
RGB	0.5	93.8%	3.9%	7.6%	128	25.4
ERYC	0.5	94.6%	2.5%	4.8%	210	40.0
IR12C	0.5	92.6%	3.2%	6.2%	155	30.7

Interestingly, the 0.5 m GSD results consistently produced higher *TPR* results while still maintaining an *EpSK* that was >50% smaller than the 0.3 m GSD results. This behavior is not necessarily reflected in the 5-fold experiments. We do not have a hypothesis at this time for why the 0.5 m GSD results performed so much better than the 0.3 m GSD results. Consequently, further investigation would be needed to better understand this somewhat counter-intuitive outcome.

The results in Table 5.7 show that by fusing the partitioned MS detections using CI with the Sugeno- λ computed FM lattice, an absolute *TPR* gain of 5.1% is achieved in conjunction with a reduced *EpSK* of $\sim 20\%$ for 0.5 m GSD images. Using the same FM lattice but using SI, the *EpSK* can be reduced by $\sim 62\%$ but with a smaller absolute *TPR* gain of 1.3%. Also, using CI with the Sugeno- λ computed FM lattice an absolute *TPR* gain of 0.3% is obtained with an *EpSK* reduction of $\sim 48\%$ for 0.3 m GSD images. This is the first time a *TPR* gain in conjunction with significant error reduction has been observed in these experiments. This shows that the additional information provided in 8-band multi-spectral imagery can be leveraged using pre-

Table 5.6: Top $F1$ score results for MS-band fusion experiments from models of size 32, 64, and 128. Red indicates baseline result.

Fusion	FM	GSD (m)	TPR	PPV	$F1$	$EpSK$	$EOTR$
RGB		0.3	88.3%	1.8%	3.5%	275	56.7
SI	QP	0.3	76.5%	11.5%	20.0%	34.1	8.7
SI	AVG $F1/3$	0.3	76.4%	11.5%	20.0%	34.2	8.7
SI	AVG $F1$	0.3	76.4%	11.5%	20.0%	34.2	8.7
SI	$F1/3$	0.3	76.4%	11.5%	20.0%	34.2	8.7
CI	$F1/3$	0.3	80.9%	8.9%	16.0%	47.2	11.2
AVG	N/A	0.3	82.1%	8.2%	14.9%	52.2	12.2
CI	AVG $F1/3$	0.3	83.3%	8.1%	14.8%	53.7	12.4
RGB		0.5	92.1%	3.9%	7.5%	126	25.4
SI	AVG $F1/3$	0.5	81.1%	25.2%	38.4%	14.5	4.0
SI	AVG $F1$	0.5	81.1%	25.2%	38.4%	14.5	4.0
SI	QP	0.5	81.1%	25.2%	38.4%	14.5	4.0
SI	$F1/3$	0.5	83.0%	21.6%	34.3%	17.8	4.6
CI	$F1/3$	0.5	83.9%	20.0%	32.3%	19.7	5.0
AVG	N/A	0.5	84.3%	19.4%	31.5%	20.4	5.2
CI	AVG $F1/3$	0.5	85.3%	18.6%	30.6%	21.6	5.4

Table 5.7: Top TPR results for MS-band fusion experiments for models of size 32, 64, and 128. Red indicates baseline result.

Fusion	FM	GSD (m)	TPR	PPV	$F1$	$EpSK$	$EOTR$
ARR		0.3	96.0%	0.6%	1.2%	871	163.5
CI	$F1$	0.3	93.4%	2.3%	4.5%	220	43.2
SI	$F1$	0.3	89.6%	4.6%	8.8%	104	21.7
RGB		0.3	88.3%	1.8%	3.5%	275	56.7
CI	AVG $F1$	0.3	87.3%	6.0%	11.2%	77.0	16.7
CI	QP	0.3	85.1%	7.4%	13.6%	60.3	13.5
ARR		0.5	95.9%	1.4%	2.7%	383	73.1
CI	$F1$	0.5	92.4%	7.3%	13.6%	65.5	13.6
RGB		0.5	92.1%	3.9%	7.5%	126	25.4
CI	AVG $F1$	0.5	88.0%	13.7%	23.8%	31.5	7.3
SI	$F1$	0.5	86.9%	16.4%	27.6%	25.5	6.1
CI	QP	0.5	86.9%	12.7%	22.2%	34.0	7.9

existing, pre-trained 3-band RGB DNN models to achieve significant error reduction while maintaining or even increasing the TPR .

Table 5.8: Top $F1$ score results for MS partition fusion experiments with 256 models included. Red indicates comparative baseline.

Fusion	FM	GSD (m)	TPR	PPV	$F1$	EpSK	$EOTR$
RGB		0.3	94.6%	1.8%	3.6%	283	54.6
SI	QP	0.3	84.4%	11.9%	20.8%	35.7	8.4
SI	$F1/3$	0.3	84.3%	11.9%	20.8%	35.8	8.4
SI	AVG $F1/3$	0.3	84.3%	11.9%	20.8%	35.8	8.4
SI	AVG $F1$	0.3	84.3%	11.9%	20.8%	35.8	8.4
CI	$F1/3$	0.3	88.1%	9.2%	16.7%	49.2	10.9
AVG	N/A	0.3	89.0%	8.5%	15.5%	54.2	11.8
CI	AVG $F1/3$	0.3	90.1%	8.3%	15.3%	55.8	12.0
RGB		0.5	93.8%	3.9%	7.6%	126	25.4
SI	AVG $F1/3$	0.5	84.6%	24.5%	38.0%	15.4	4.1
SI	QP	0.5	84.9%	24.2%	37.7%	15.6	4.1
SI	AVG $F1$	0.5	84.9%	24.2%	37.7%	15.7	4.1
SI	$F1/3$	0.5	86.1%	21.2%	34.0%	18.7	4.7
CI	$F1/3$	0.5	86.9%	19.4%	31.7%	20.9	5.2

Tables 5.8 and 5.9 provide updated results that include 256-model size results in the pooling. These results are consistent with the best results without pooling 256-model size results. Compared to Table 5.6, the top TPR result (0.5 m GSD) in Table 5.8 shows that the addition of the 256 model produced a 2.5% absolute gain in the TPR with essentially no change in the $EpSK$.

The results in Table 5.9 again show that by fusing the partitioned MS detections using CI with the Sugeno- λ computed FM lattice, an absolute TPR gain of 0.3% is achieved in conjunction with a reduced $EpSK$ of $\sim 46\%$ for 0.5 m GSD images. For 0.3 m GSD images an absolute TPR gain of 1.8% is obtained with an $EpSK$ reduction of $\sim 20\%$. Compared to the 0.5 m GSD result in Table 5.7, the addition of the 256 model increased the absolute TPR by 1.1% with only an $EpSK$ increase of 4%.

Table 5.9: Top TPR score results for MS partition fusion experiments with 256 models included. Red indicates comparative baseline.

Fusion	FM	GSD (m)	TPR	PPV	$F1$	$EpSK$	$EOTR$
ARR		0.3	97.8%	0.6%	1.2%	884	163
CI	$F1$	0.3	96.4%	2.3%	4.5%	227	43.2
RGB		0.3	94.6%	1.8%	3.6%	283	54.6
SI	$F1$	0.3	93.8%	4.6%	8.7%	109	21.8
CI	AVG	0.3	93.1%	6.2%	11.5%	79.6	16.2
CI	QP	0.3	91.3%	7.6%	14.0%	62.6	13.2
ARR		0.5	97.3%	1.4%	2.7%	391	73.1
CI	$F1$	0.5	94.1%	7.2%	13.4%	68.0	13.9
RGB		0.5	93.8%	3.9%	7.6%	126	25.4
CI	AVG $F1$	0.5	90.4%	13.3%	23.2%	33.3	7.5
CI	QP	0.5	89.4%	15.8%	26.9%	27.1	6.3
SI	$F1$	0.5	89.3%	12.4%	21.8%	35.7	8.0

5.5 CONCLUSION AND FUTURE WORK

In this research we developed and tested a unique technical approaches that combined results from multiple DNN detectors to improve the detection of *Engineering Vehicles* (EV) in the public domain benchmark xView dataset. We did this by utilizing fusion of DNN *EV* object detections (single DNN architecture) from multiple 3-band multi-spectral (MS) images that were partitioned from the original 8-band MS xView imagery.

Significant improvements over the results from Chapter 4 were achieved by fusing individual detection results from a single DNN architecture applied to three 3-band MS images. The results demonstrated that a 0.3%-5.1% gain in TPR could be achieved while at the same time reducing the $EpSK$ by up to $\sim 60\%$. The best results were generated using the Choquet integral with Sugeno- λ computed fuzzy measures. This shows that the additional information provided in 8-band multi-spectral imagery can be leveraged using pre-existing, pre-trained 3-band RGB DNN models to achieve significant error reduction while maintaining or even increasing the TPR .

Future work worth pursuing could include: A) incorporating multiple DNN architectures into the MS band partition fusion experiments, B) applying these fusion techniques to a variety of other rare object classes, C) exploring more sophisticated ways of combining Sugeno- λ and data-driven fuzzy measure lattices, and D) scaling up these experiments to broad area scanning ($>1,000 \text{ km}^2$) for rare objects under more realistic operational scenarios (e.g. mission-relevant AOIs).

Chapter 6

NEURAL LEARNING BASED BOUNDING-BOX MODEL FUSION/ENSEMBLING FOR SCARCE OBJECT DETECTION

6.1 INTRODUCTION

The overall objective of this research is to develop and test several technical approaches to perform bounding-box object detection for scarce objects. These include DNN image scanning (Section 2.3.2) with spatial clustering, bounding-box object detectors, and model fusion/ensembling methods for multiple bounding-box object detectors. We compare these technical approaches to detect primarily *Surface-to-Air Missile Transporter Erector Launchers* (SAM TELs) and secondarily, *Surface-to-Air Missile Launch Pads* (SAM LPs) using a world wide, in-house curated dataset. Research presented in this chapter includes:

1. Comparing the performance of image scanning + spatial clustering with bounding-box object detectors.
2. Evaluating multiple bounding-box object detectors:
 - (a) Currently available or “out-of-the-box” (OOB) bounding-box model ensembling techniques.

- (b) Bounding-box ensembling using basic Multiple Layer Preceptors (MLPs) to predict bounding-box coordinates and produce improved class confidence vectors.
 - (c) The introduction of a Pseudo-Cell State Long-Short Term Memory (PCS-LSTM) neural network to produce improved class confidence vectors.
3. Extending and demonstrating the MLP and PCS-LSTM architectures for ensembling with three bounding-box detector inputs.

The *SAM TEL* class was selected as the primary class because of its military importance given its widespread use in modern air-defense systems throughout the world. Results from image scanning using state-of-the-art DNN architectures (e.g. ProxylessNAS [33] and EfficientNet [34]) are compared to state-of-the-art bounding-box object detectors: YOLOv5 [52] and Detectron2 [53].

6.2 SOURCE DATA

All source imagery was collected by DigitalGlobe satellites and the labeled object datasets were mainly curated by the Center for Geospatial Intelligence (CGI) at the University of Missouri.

6.2.1 SAM-Focused Dataset

The initial dataset consisted of 12,226 image scenes selected from a collection in the CGI DeepNET [?] database. Scenes typically were comprised of multiple 512×512 image tiles mosaicked together (see Fig. 6.1). The scene selection was determined based on the presence of *SAM TELs*, *Surface-to-Air Missile Sites* (SAM Sites), and/or *SAM LPs* within a given scene. For this preliminary analysis we used DeepNET data v2020.12 which was known to have incomplete and inconsistently labeled data. For this reason and to reduce error, the following heuristic criteria was used in final scene

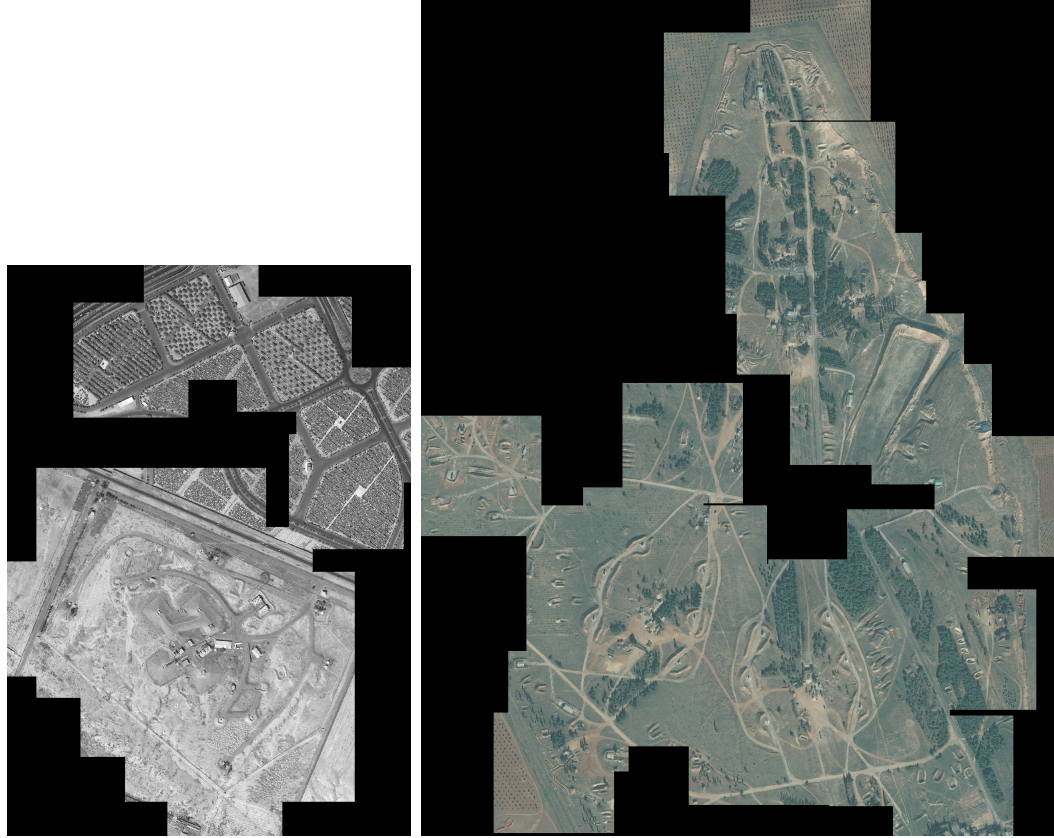


Fig. 6.1: Sample scenes created by mosaicking several 512×512 pixel images tiles.

selection:

- (i) There must be at least the same number of *SAM LPs* as *SAM Sites* while also containing at least one *SAM TEL*. Purpose: *SAM Sites* should contain multiple *SAM LPs*, so counts revealing the opposite (i.e. $SAM LPs < SAM Sites$) would indicate incorrect or incomplete labeling.
- (ii) There must be at least one *SAM TEL* and no *SAM Sites*. Purpose: To include staged or parked *SAM TELs* outside a designated *SAM Site*.
- (iii) There must include at least one *SAM LP* without *SAM Sites* or *SAM TELs*. Purpose: To include empty *SAM LPs* outside a designated *SAM Site*.

6.2.2 80-20 Scene Partition

We divided the scenes collection into an 80-20 split for training and test/validation respectively, in the following manner:

1. Five empty bins were initialized.
2. Sort the scenes using the criterion (i) above with the *SAM TEL* “density” or total number of *SAM TELs* in the overall valid area of a scene (i.e. excluding black or white buffer pixels within the scenes).
3. Each sorted scene is then assigned to the bin of smallest total area in order to create partitions of $\sim 20\%$ of the total valid area and *SAM TELs*.
4. Repeat steps 2 and 3 for scenes using criterion (ii).
5. Repeat steps 2 and 3 for the scenes using criterion (iii), but using *SAM LP* density instead of *SAM TEL* density.
6. Finally, select a random bin from the five bins as the test/validation dataset.

The final training and test/validation scene counts and object densities are provided in Table 6.1.

Because quality enhancements for the CGI DeepNET dataset are ongoing, we implemented heuristic filters to clean-up the object labeling within the scenes. We surmised that any *LP* within a *SAM Site* could be assumed to be a *SAM LP* and any *TEL* within a *SAM Site* or *SAM LP* could be assumed to be a *SAM TEL*. This was done before partitioning.

6.2.3 Bounding-Box Detection Class Labels & Datasets

The DeepNET database has a growing number of increasingly descriptive object class labels. For the scenes used in these experiments there were 127 total classes

Table 6.1: Train/test scene counts for 80-20 partition and class densities used to create the partition. Densities are calculated by the number of objects per km² of valid pixel (no black) scene content.

	Total	Train	Test
Scenes Count	12,226	9,748	2,478
Scenes Percentage	100%	79.9%	20.1%
Scene Area (km ²)	3,816	3,050	766
<i>SAM Site</i> Density	2.2	2.2	2.1
<i>SAM TEL</i> Density	10.5	10.5	10.6
Both <i>SAM LP</i> Density	12.3	12.3	12.2

(complete list can be found in Appendix A.4). We narrowed these classes into a 37-class dataset (Table 6.2) based on class similarity and count. In addition, we further pruned the classes to create an 11-class dataset (Table 6.3) by removing some classes and creating a new *TEL Confuser* superclass (e.g. engineering vehicles, buses, other military vehicles, etc...). This was done to evaluate the performance of bounding-box detectors based on the number of classes. We next created a 4-class dataset (Table 6.4) with an *All TEL* class consisting of *SAM TELs* and non-*SAM TELs*, an *All LP* class consisting of *SAM LP* and any other *LP*, and the *TEL Confuser* class. Finally, we further pruned the 4-class dataset down to one dataset containing just the *All TEL* and *TEL Confuser* classes and another dataset containing just the *All LP* class. Example of bounding-boxes with labels in a scene is provided in Fig. 6.2.

6.2.4 Image Scanning Class Labels & Datasets

Since the best performing bounding-box object detectors for *SAM TELs* were achieved in the 11-class experiments (see Section 6.3.2) we decided to use this class structure for the image scanning + spatial clustering (IS+SC) experiments. However, because the *TEL* and *LP* classes are frequently co-located, we created two datasets to reduce confusion: a) an *LP* dataset with all the *TEL* and *SAM Missile* classes removed and b) a *TEL* dataset with all the *LP* classes removed. In addition, we added a *Background* (BG) class by simply sampling the scenes and throwing out samples that



Fig. 6.2: Example of bounding-boxes with labels from the DeepNET dataset v2020.12.

were either $>25\%$ black/white buffer or overlapped any of the truth bounding boxes for the 11-class dataset objects. The *BG* samples were limited from larger scenes to reduce the chance of over-fitting on the larger scenes. There were $\sim 200,000$ *BG* samples for each sample size. Sample counts can be found in Table 6.5, Per-class online augmentation (Appendix A.2) was used to increase to the sample size of each relevant class to $\sim 4,000,000$ and the *BG* sample size to $\sim 15,000,000$.

6.2.5 Image Size Reductions for Training and Testing

We would have preferred to process each full scene. However, due to GPU memory limitations with YOLOv5, all scenes were partitioned into 512×512 sub-images for processing. Although the original scene mosaicks were produced from 512×512 image

Table 6.2: Class counts for 37-class experiments used in bounding-box detector experiments. An approximate 80-20 partition was achieved on a per-class basis using the partitioning method described in Section 6.2.2. Primary object classes of interest used in the partitioning schema are highlighted in blue.

Class	Total	Train	Test	Train(%)	Test(%)
Scenes	12,226	9,748	2,478	79.7%	20.3%
<i>Aircraft</i>	2,082	1,591	491	76.4%	23.6%
<i>Antenna</i>	3,212	2,555	657	79.5%	20.5%
<i>Anti-Aircraft Artillery</i>	2,734	2,179	555	79.7%	20.3%
<i>Anti-Aircraft Artillery Site</i>	3,626	2,884	742	79.5%	20.5%
<i>Bunker</i>	4,421	3,482	939	78.8%	21.2%
<i>Combat Ground Motor Vehicle</i>	3,803	2,877	926	75.7%	24.3%
<i>Combat Support Vehicle</i>	3,335	2,650	685	79.5%	20.5%
<i>Confuser Vehicle</i>	531	436	95	82.1%	17.9%
<i>Emitter Site</i>	394	313	81	79.4%	20.6%
<i>Emitter Structure</i>	240	191	49	79.6%	20.4%
<i>Field Artillery</i>	963	813	150	84.4%	15.6%
<i>Flat Bed Truck</i>	623	549	74	88.1%	11.9%
<i>Ground Motor Vehicle</i>	25,527	20,293	5,234	79.5%	20.5%
<i>Howitzer</i>	687	625	62	91.0%	9.0%
<i>Infrastructure</i>	819	669	150	81.7%	18.3%
<i>LP w/ Revetment</i>	1,458	1,237	221	84.8%	15.2%
<i>LP</i>	12,803	10,080	2,723	78.7%	21.3%
<i>Military Site</i>	316	262	54	82.9%	17.1%
<i>SAM TEL</i>	40,118	31,965	8,153	79.7%	20.3%
<i>Mound</i>	1,041	823	218	79.1%	20.9%
<i>Multi Ramp Platform</i>	4,774	3,871	903	81.1%	18.9%
<i>Other TEL</i>	1,598	1,189	409	74.4%	25.6%
<i>Other Vehicle</i>	5,204	4,178	1,026	80.3%	19.7%
<i>Parabolic Antenna</i>	686	527	159	76.8%	23.2%
<i>Platform</i>	931	726	205	78.0%	22.0%
<i>Revetment</i>	29,416	23,701	5,715	80.6%	19.4%
<i>Semi Truck</i>	2,030	1,741	289	85.8%	14.2%
<i>Single Ramp Platform</i>	2,088	1,661	427	79.5%	20.5%
<i>SAM Missile</i>	21,715	17,531	4,184	80.7%	19.3%
<i>SAM LP</i>	11,246	9,096	2,150	80.9%	19.1%
<i>SAM LP w/ Revetment</i>	35,668	28,501	7,167	79.9%	20.1%
<i>SAM Site</i>	8,330	6,734	1,596	80.8%	19.2%
<i>SAM Launcher</i>	9,739	7,884	1,855	81.0%	19.0%
<i>Transloader</i>	3,024	2,395	629	79.2%	20.8%
<i>Tripod Mast</i>	529	408	121	77.1%	22.9%
<i>Truck</i>	12,289	9,795	2,494	79.7%	20.3%
<i>Weapon</i>	1,146	908	238	79.2%	20.8%

Table 6.3: Class counts for 11-class experiments used in bounding-box detection experiments. An approximate 80-20 partition was achieved on a per class basis using the partitioning method described in Section 6.2.2. Primary object classes of interest used in the partitioning schema are highlighted in blue.

Class	Total	Train	Test	Train(%)	Test(%)
<i>Combat Support Vehicle</i>	6,359	5,045	1,314	79.3%	20.7%
<i>Confuser Vehicle</i>	4,957	3,862	1,095	77.9%	22.1%
<i>LP w/ Revetment</i>	1,458	1,237	221	84.8%	15.2%
<i>LP</i>	12,803	10,080	2,723	78.7%	21.3%
<i>SAM TEL</i>	40,118	31,965	8,153	79.7%	20.3%
<i>Other TEL</i>	1,598	1,189	409	74.4%	25.6%
<i>SAM Missile</i>	21,715	17,531	4,184	80.7%	19.3%
<i>SAM LP</i>	11,246	9,096	2,150	80.9%	19.1%
<i>SAM LP w/ Revetment</i>	35,668	28,501	7,167	79.9%	20.1%
<i>SAM Launcher</i>	9,739	7,884	1,855	81.0%	19.0%
<i>Weapon</i>	5,530	4,525	1,005	81.8%	18.2%

Table 6.4: Class counts for 4-class experiments used in bounding-box detection experiments. Same counts apply to separate *TEL* and *LP* datasets.

Class	Total	Train	Test	Train(%)	Test(%)
Combat Support Vehicle	6,359	5,045	1,314	79.3%	20.7%
Confuser Vehicle	4,957	3,862	1,095	77.9%	22.1%
<i>LP</i>	61,175	48,914	12,261	80.0%	20.0%
<i>TEL</i>	41,716	33,154	8,562	79.5%	20.5%

tiles, projecting the more recently generated bounding boxes back onto the original images after labeling at the scene level would have created spatial errors in registering the images and the final objects labels. For this purpose, only bounding-boxes that retained a 50% overlap with the scene partition were included with the given partition. These same partitions were used for the IS+SC experiments to facilitate easier comparison and possible fusion.

Table 6.5: Training sample counts for image scanning + spatial clustering experiments.

Class	TEL Dataset	LP Dataset
<i>Combat Support Vehicle</i>	5,045	5,045
<i>Confuser Vehicle</i>	3,862	3,862
<i>LP</i>	10,080	10,080
<i>LP w/ Revetment</i>	1,237	1,237
<i>SAM TELs</i>	31,965	N/A
<i>Other TEL</i>	1,189	1,189
<i>SAM Missile</i>	17,531	N/A
<i>SAM Launcher</i>	7,884	N/A
<i>SAM LP</i>	N/A	9,096
<i>SAM LP w/ Revetment</i>	N/A	28,501
<i>Weapon</i>	4,525	4,525
<i>Background</i> (for 32x32 models training)	199,964	199,964
<i>Background</i> (for 64x64 models training)	189,374	189,374
<i>Background</i> (for 128x128 models training)	213,749	213,749

6.3 BOUNDING-BOX OBJECT DETECTORS & IMAGE SCANNING + SPATIAL CLUSTERING EXPERIMENTS

6.3.1 Bounding-Box Detection

The purpose of a bounding-box object detector is to produce a rectangular box that encapsulates an object within an image. Regions with CNN features (R-CNNs) [9] showed that image objects can be detected using class-independent region identification and subsequent classification using basic CNNs. YOLO [8] takes this a step further and allows the CNN to identify the bounding box along with the class. Various iterations of YOLO [56] [12] [57] [52] improved computational speed and introduced multi-scale detections, more accurate CNNs, anchor boxes, support for regression used in class-specific box learning, and input image augmentation. For the experiments in this section, we used YOLOv5 [52] to generate our bounding-box object detectors. In later sections, we include Facebook’s Detectron2 [53] for a second bounding-box object detection model. The output for each detector returns an anchor point (box

center or top left), the height and width of the bounding box, and the class index with corresponding confidence for each bounding box. In our experiments we used a 25% Intersection-over-Union (IoU) for bounding-box regression. We also tested a small, medium, large, and extra-large model sizes for YOLOv5 for each dataset using MS COCO [54] pre-trained weights along with 300-epoch training cycles.

6.3.2 YOLOv5 Model Size Experiments

There are four model sizes available in the YOLOv5 repository: small (yolov5s), medium (yolov5m), large (yolov5l), and extra-large (yolov5x). We tested all permutations of these models with all the datasets described in Section 6.2.3. The results in Table 6.6 show that the 11-class YOLOv5 XL model produced the best recall for *TELS* with IoUs @ 0.25, 0.5, and 0.75 as well as the best *F1* score for IoU @ 0.5. Although the XL model for 37-class had higher *F1* scores for IoUs @ 0.25 and 0.75, a few % point gain in recall is preferred over the same gain in precision for more practical applications. It should also be noted that the highest precision for each IoU also resulted in a $\sim 10\%$ reduction in recall. Once again, this demonstrated the precision vs. recall trade-off mentioned in previous chapters. The fact that the 11-class XL model produced the best recall at all three IoU values indicates that the bounding-box predictions are more centralized or “tighter” to the truth bounding boxes.

For the results in Table 6.7, the weighted sum of the two *SAM LP* classes was calculated (using the object percentage as weights) to gain a clearer picture of overall model performance and to compare the All LP class of the 4-class and LP models. The results indicate that models with fewer classes performed better for general *LPs*. The same thing was observed in Section 3.4.1. However, we also found that models with more classes produce better results (indicated in yellow) for the *SAM LP w/ Revetments* class which is $\sim 75\%$ of the total *SAM LP* classes. This indicates that a unique visual feature such as a revetment can reduce ambiguity in class detection. Further,

since *SAM TELs* are the primary target in these experiments we decided to move forward with using the 11-class datasets for all experiments comparing bounding-box object detection and IS+SC.

6.3.3 Bounding Boxes for Image Scanning + Spatial Clustering

We trained and tested CNN models with input sizes 32x32, 64x64, and 128x128 for each dataset described in Section 6.2.4 for ProxylessNAS, EfficientNet-B3, and EfficientNet-B7 architectures. Each CNN model was trained with ImageNet [35] pre-trained weights for 1 epoch using augmentation (Section 2.3.1) to produce $\sim 1,000,000$ samples per class. CNNs do not generate bounding boxes from IS+SC because only a centerpoint (e.g. cluster mode) is generated after clustering the CNN centerpoint inputs from the image scanning. Therefore, we generated bounding boxes for testing with each bounding box centered on a cluster mode. We used bounding-box height and widths of 32, 64, or 128 pixels with the size corresponding to the trained model size. However, the areas of these bounding boxes presented an additional challenge. Even if the predicted bounding box was aligned perfectly with the truth bounding box for input size 32, 33% of the truth bounding boxes have an area < 526 pixels which is the minimum to yield a 25% IoU. Therefore, we also produced a truth dataset using the existing bounding-box center but with bounding boxes of height and width equal to the same size as the CNN model to facilitate a more fair comparison.

6.3.4 Evaluation Metrics

A common technique for evaluating bounding-box object detectors uses the Intersection-over-Union (IoU), which is the ratio of the intersection between two bounding boxes over the union of the same (Fig. 6.3). For our preliminary experiments we evalu-

$$\text{IoU}(\text{dashed square}, \text{solid square}) = \frac{\text{Intersection}}{\text{Union}}$$

$$\text{IoU}(\text{dashed circle}, \text{solid circle}) = \frac{\text{Intersection}}{\text{Union}}$$

Fig. 6.3: Graphical representation of Intersection-over-Union (IoU) for squares and circles (equal sizes) at 25% overlap.

ated multiple IoU values to assess detector performance. A detection is counted if a prediction bounding box has an $\text{IoU} >$ a certain threshold vs. the truth bounding boxes. If multiple predictions overlap a truth bounding box then priority is given to the bounding box with the greatest overlap. Also, once a prediction bounding box has been paired with a truth bounding box or other model bounding box it may not be paired again. A True Positive (TP) results if the paired bounding boxes are of the same class, otherwise the prediction is considered a False Positive (FP). All remaining truth bounding boxes after pairing are considered False Negatives (FN). The TP , FP , and FN counts are then used to compute the recall, precision, and $F1$ scores.

6.3.5 Results

Detections results for the *SAM TEL* class for different YOLOv5 model depths and dataset with varying number of classes are presented in Table 6.6. We observed that the best recall was achieved using the 11-class dataset with the XL YOLOv5 model across all three IoU thresholds with the large models ranking second. The 11-class XL model only achieved the best $F1$ score for $\text{IoU} @ 0.5$, but is within one percentage point of the highest for both $\text{IoUs} @ 0.25$ and 0.75 . Table 6.7 shows similar results for the *SAM LP* and *SAM LP w/ Revetment* classes alone, but when these classes are combined, the 4-class and *LP* datasets did better than the post-cluster combined *SAM LP* classes ('COMBO' in the table). This might indicate that using a model

with fewer classes and then running a refining classifier on the backend could perform well for *Launch Pads*, but would need further investigation. However, this would need further investigation. However, considering that three out of four top-ranked models used the 11-class dataset and that *SAM TELs* are the primary target of this effort, we decided to use the 11-class dataset for the remainder of the experiments in this chapter.

The results of *SAM TEL* detection for IS+SC are presented in Table 6.8. The top performing model was EfficientNet-B7 trained on 32×32 samples and using the 32×32 pixel truth bounding box for evaluation. The *F1* score of 19.1% was 3.6% larger than the evaluation using the original truth labels. However, this was not half the *F1* score of 59.4% achieved by YOLOv5 in Table 6.6 and is in fact a relative reduction of 67.8%. Similar, but not quite as dramatic results are found for *SAM LPs* in Table 6.9. Again, the EfficientNet-B7 architecture did very well, but this time using the 64×64 pixel truth bounding boxes. This is expected as *SAM LPs* are much larger than *SAM TELs*. The decrease in *F1* score is only a relative loss of 35.6%. However, given the poor results generated by IS+SC, we did not pursue any further experiments using IS+SC as a bounding-box detector.

Table 6.6: Bounding-box experimental results for *TELs* for different size YOLOv5 models. Best results shaded in blue. Note that the 37-class and 11-class experiments are only finding *SAM TELs* where the 4-class and *TEL* experiments are all *TELs*.

Dataset	Model Size	BB Threshold	Class Label	IoU@0.25			IoU@0.5			IoU@0.75		
				Prec.	Recall	F1 score	Prec.	Recall	F1 score	Prec.	Recall	F1 score
11-class	XL	0.25	<i>SAM TEL</i>	58.5%	60.3%	59.4%	50.9%	52.4%	51.7%	20.1%	20.7%	20.4%
11-class	LARGE	0.25	<i>SAM TEL</i>	58.0%	59.6%	58.8%	50.1%	51.5%	50.8%	19.4%	20.0%	19.7%
37-class	XL	0.25	<i>SAM TEL</i>	62.2%	58.7%	60.4%	53.6%	50.6%	52.1%	21.4%	20.2%	20.8%
11-class	MEDIUM	0.25	<i>SAM TEL</i>	58.8%	57.8%	58.3%	50.3%	49.5%	49.9%	19.1%	18.8%	19.0%
37-class	LARGE	0.25	<i>SAM TEL</i>	62.8%	56.8%	59.7%	54.1%	48.9%	51.4%	21.5%	19.4%	20.4%
<i>TEL</i>	XL	0.25	All <i>TEL</i>	53.8%	56.7%	55.2%	46.8%	49.3%	48.0%	18.2%	19.2%	18.7%
37-class	MEDIUM	0.25	<i>SAM TEL</i>	63.3%	54.5%	58.5%	53.7%	46.3%	49.7%	20.9%	18.0%	19.4%
4-class	XL	0.25	All <i>TEL</i>	60.2%	52.8%	56.2%	51.4%	45.1%	48.0%	20.6%	18.1%	19.2%
<i>TEL</i>	LARGE	0.25	All <i>TEL</i>	56.8%	52.6%	54.6%	49.3%	45.6%	47.4%	19.3%	17.9%	18.6%
11-class	SMALL	0.25	<i>SAM TEL</i>	58.3%	51.4%	54.6%	49.1%	43.4%	46.0%	16.8%	14.8%	15.7%
4-class	LARGE	0.25	All <i>TEL</i>	62.3%	49.8%	55.4%	53.5%	42.8%	47.6%	21.1%	16.9%	18.7%
37-class	SMALL	0.25	<i>SAM TEL</i>	61.7%	48.2%	54.1%	51.7%	40.4%	45.3%	17.6%	13.8%	15.4%
<i>TEL</i>	MEDIUM	0.25	All <i>TEL</i>	56.8%	47.7%	51.9%	49.2%	41.3%	44.9%	18.9%	15.9%	17.3%
4-class	MEDIUM	0.25	All <i>TEL</i>	62.2%	46.4%	53.2%	53.1%	39.7%	45.4%	20.4%	15.2%	17.5%
<i>TEL</i>	SMALL	0.25	All <i>TEL</i>	55.8%	44.1%	49.3%	48.1%	38.1%	42.5%	16.7%	13.2%	14.8%
4-class	SMALL	0.25	All <i>TEL</i>	60.7%	40.6%	48.6%	50.8%	33.9%	40.7%	18.1%	12.1%	14.5%
<i>TEL</i>	XL	0.5	All <i>TEL</i>	84.2%	27.6%	41.6%	77.2%	25.3%	38.1%	35.2%	11.5%	17.4%
11-class	LARGE	0.5	<i>SAM TEL</i>	86.5%	25.2%	39.1%	79.3%	23.1%	35.8%	38.0%	11.1%	17.2%
37-class	XL	0.5	<i>SAM TEL</i>	86.3%	25.2%	39.0%	79.4%	23.1%	35.8%	40.1%	11.7%	18.1%
11-class	XL	0.5	<i>SAM TEL</i>	86.7%	25.0%	38.8%	79.8%	23.0%	35.7%	39.3%	11.4%	17.6%
37-class	LARGE	0.5	<i>SAM TEL</i>	86.9%	23.5%	37.0%	79.8%	21.6%	34.0%	40.5%	11.0%	17.3%
<i>TEL</i>	LARGE	0.5	All <i>TEL</i>	85.7%	22.7%	35.9%	79.2%	21.0%	33.2%	36.3%	9.6%	15.2%
4-class	XL	0.5	All <i>TEL</i>	87.2%	22.6%	35.9%	80.7%	21.0%	33.3%	42.2%	10.9%	17.4%
11-class	MEDIUM	0.5	<i>SAM TEL</i>	86.1%	22.2%	35.3%	78.9%	20.4%	32.4%	38.6%	10.0%	15.8%
4-class	LARGE	0.5	All <i>TEL</i>	89.3%	19.5%	31.9%	82.5%	18.0%	29.6%	43.0%	9.4%	15.4%
37-class	MEDIUM	0.5	<i>SAM TEL</i>	87.4%	19.5%	31.9%	80.6%	18.0%	29.4%	41.4%	9.2%	15.1%
<i>TEL</i>	MEDIUM	0.5	All <i>TEL</i>	83.9%	19.3%	31.4%	76.0%	17.5%	28.5%	34.8%	8.0%	13.0%
4-class	MEDIUM	0.5	All <i>TEL</i>	89.1%	17.5%	29.2%	81.8%	16.0%	26.8%	42.3%	8.3%	13.9%
<i>TEL</i>	SMALL	0.5	All <i>TEL</i>	84.9%	15.2%	25.8%	77.8%	14.0%	23.7%	33.6%	6.0%	10.2%
11-class	SMALL	0.5	<i>SAM TEL</i>	89.4%	14.6%	25.1%	81.2%	13.3%	22.8%	38.3%	6.3%	10.8%
37-class	SMALL	0.5	<i>SAM TEL</i>	88.5%	12.8%	22.4%	81.3%	11.8%	20.5%	36.6%	5.3%	9.3%
4-class	SMALL	0.5	All <i>TEL</i>	90.5%	10.5%	18.7%	84.2%	9.7%	17.4%	38.6%	4.5%	8.0%

Table 6.7: Bounding-box experimental results for *LP* detection for extra-large YOLOv5 models. Top portion is the different *LP* classes for the 37-class and 11-class experiments. Bottom portion is a comparison of the *All LP* classes for the 4-class and *LP* experiments and the weighted sum of the *SAM LP* classes from the 37 and 11-class experiments. Blue shading indicates the best results for each column. Yellow shading indicates that one of the *SAM LP* classes from the 37 or 11-class experiments would have been best.

Dataset	Model Size	BB Threshold	Class Label	IoU@0.25			IoU@0.5			IoU@0.75		
				Prec.	Recall	F1 score	Prec.	Recall	F1 score	Prec.	Recall	F1 score
11-class	XL	0.25	<i>SAM LP</i> w/ Rev.	70.5%	74.2%	72.3%	65.6%	69.1%	67.3%	47.5%	49.9%	48.7%
37-class	XL	0.25	<i>SAM LP</i> w/ Rev.	71.1%	71.3%	71.2%	66.6%	66.9%	66.7%	48.4%	48.6%	48.5%
11-class	XL	0.5	<i>SAM LP</i> w/ Rev.	82.2%	63.7%	71.8%	78.0%	60.5%	68.1%	59.1%	45.8%	51.6%
37-class	XL	0.5	<i>SAM LP</i> w/ Rev.	83.4%	59.0%	69.1%	79.4%	56.1%	65.7%	61.0%	43.1%	50.5%
11-class	XL	0.25	<i>SAM LP</i>	40.5%	29.9%	34.4%	37.1%	27.4%	31.5%	21.3%	15.7%	18.1%
37-class	XL	0.25	<i>SAM LP</i>	42.4%	18.8%	26.1%	39.7%	17.6%	24.4%	25.5%	11.3%	15.7%
11-class	XL	0.5	<i>SAM LP</i>	62.3%	8.9%	15.6%	61.2%	8.8%	15.3%	46.1%	6.6%	11.5%
37-class	XL	0.5	<i>SAM LP</i>	70.7%	6.5%	11.9%	69.6%	6.4%	11.7%	53.3%	4.9%	9.0%
4-class	XL	0.25	<i>All LP</i>	72.1%	78.7%	75.2%	61.8%	67.5%	64.6%	39.1%	42.7%	40.8%
<i>LP</i>	XL	0.25	<i>All LP</i>	72.5%	78.5%	75.4%	62.4%	67.5%	64.9%	40.4%	43.7%	41.9%
<i>LP</i>	XL	0.5	<i>All LP</i>	86.1%	65.9%	74.6%	75.4%	57.6%	65.3%	51.8%	39.7%	44.9%
4-class	XL	0.5	<i>All LP</i>	86.7%	64.3%	73.8%	76.4%	56.6%	65.0%	51.8%	38.4%	44.1%
11-class	XL	0.25	<i>SAM LP</i> (COMBO)	63.6%	63.9%	63.5%	59.1%	59.4%	59.0%	41.4%	42.0%	41.6%
37-class	XL	0.25	<i>SAM LP</i> (COMBO)	64.4%	59.2%	60.8%	60.4%	55.5%	57.0%	43.1%	40.0%	40.9%
11-class	XL	0.5	<i>SAM LP</i> (COMBO)	77.6%	51.1%	58.8%	74.1%	48.5%	55.9%	56.1%	36.8%	42.4%
37-class	XL	0.5	<i>SAM LP</i> (COMBO)	80.5%	46.9%	55.9%	77.1%	44.6%	53.3%	59.2%	34.3%	40.9%

Table 6.8: Top results for each type of model for image scanning + spatial clustering for *SAM TELs*. Integer values in ‘Truth Box Type’ column indicate the size of box created for evaluation as opposed to using the original labeled bounding box. Results are for IoU @ 0.25.

Model	Model Size	Thresh	Truth Box Type	Recall	Prec.	F1
B7	32	0.9	32	26.0%	15.1%	19.1%
B7	64	0.9	64	14.4%	24.2%	18.1%
B3	64	0.5	64	17.3%	18.1%	17.7%
B7	32	0.99	Original	12.3%	20.9%	15.5%
B3	128	0.2	128	8.4%	31.8%	13.3%
B3	32	0.9	Original	15.4%	11.7%	13.3%
B7	128	0.2	128	7.8%	36.5%	12.9%
B3	32	0.6	32	31.9%	8.0%	12.8%
ProxylessNAS	32	0.6	32	11.6%	11.2%	11.4%
ProxylessNAS	64	0.4	64	14.1%	8.2%	10.3%
ProxylessNAS	32	0.6	Original	9.7%	9.4%	9.6%
ProxylessNAS	128	0.2	128	7.1%	9.3%	8.0%
B7	64	0.9	Original	0.7%	1.2%	0.9%
B3	64	0.8	Original	0.6%	1.1%	0.8%
ProxylessNAS	64	0.4	Original	0.6%	0.4%	0.5%
ProxylessNAS	128	0.2	Original	0.0%	0.0%	0.0%
B7	128	0.2	Original	0.0%	0.0%	0.0%
B3	128	0.2	Original	0.0%	0.0%	0.0%

Table 6.9: Top results for each type of model for image scanning + spatial clustering for combined *SAM LP* and *SAM LP w/ Revetment* classes. Integer values in ‘Truth Box Type’ column indicate the size of box created for evaluation as opposed to using the original labeled bounding box. Results are for IoU @ 0.25.

Model	Model Size	Thresh	Truth Box Type	Recall	Prec.	F1
B7	64	0.99	64	56.0%	42.6%	48.4%
B7	128	0.5	128	54.6%	42.4%	47.8%
B3	64	0.99	64	60.7%	37.0%	46.0%
B3	128	0.3	128	51.9%	39.8%	45.1%
B7	64	0.99	Original	52.2%	39.6%	45.1%
B3	64	0.99	Original	56.9%	34.7%	43.1%
B3	32	0.99	32	36.8%	26.4%	30.8%
B7	32	0.99	32	23.5%	41.9%	30.1%
ProxylessNAS	64	0.8	64	33.4%	24.6%	28.3%
ProxylessNAS	128	0.2	128	29.5%	25.4%	27.3%
ProxylessNAS	64	0.8	Original	31.5%	23.3%	26.8%
B3	32	0.99	Original	27.3%	19.6%	22.8%
B7	32	0.99	Original	17.0%	30.2%	21.8%
B7	128	0.7	Original	26.0%	17.4%	20.8%
B3	128	0.3	Original	23.1%	17.8%	20.1%
ProxylessNAS	32	0.7	32	14.2%	16.8%	15.4%
ProxylessNAS	128	0.3	Original	17.2%	10.3%	12.9%
ProxylessNAS	32	0.7	Original	10.2%	12.1%	11.1%

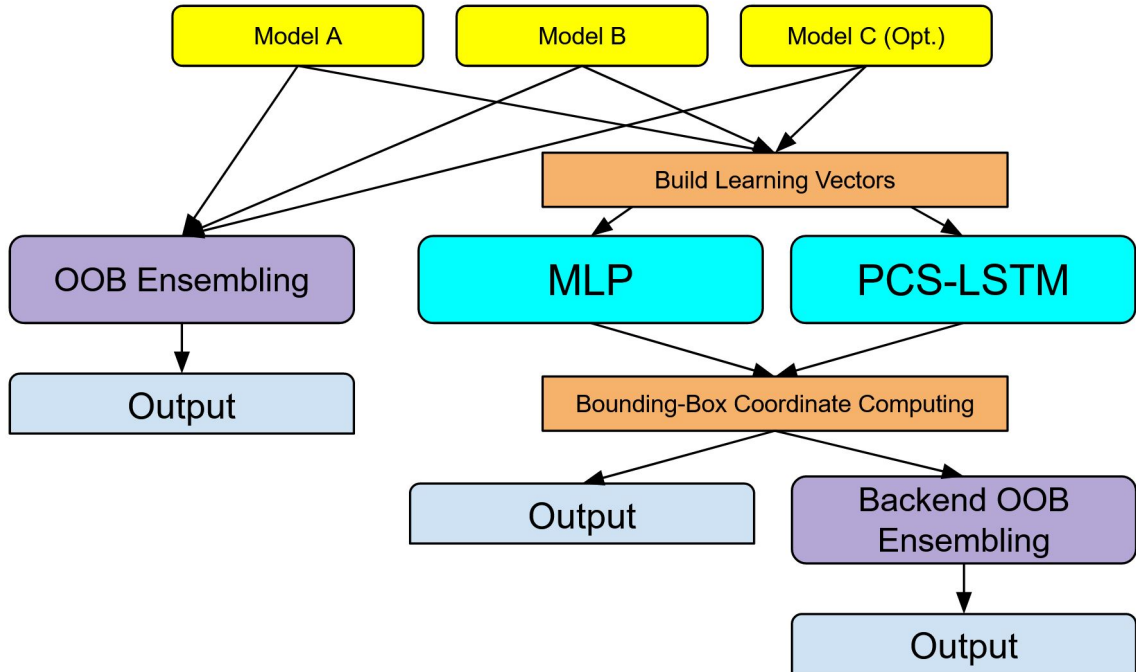


Fig. 6.4: Diagram shows data-flow, processes, and fused bounding-box outputs for comparison of different fusion/ensembling methods.

6.4 BOUNDING-BOX MODEL FUSION/ENSEMBLING EXPERIMENTS

This sections explores both “Out-Of-theBox” (OOB) and neural learning bounding-box model ensembling techniques using, as input, outputs from multiple bounding-box object detectors. We did not continue with CNN IS+SC methods as bounding-box object detectors since the results demonstrated these were much less accurate. In this section we present ensembling results from YOLOv5 and Detectron2 (D2) bounding-box object detectors. We selected a model for our YOLOv5 training that produced the $F1$ score closest to the $F1$ score from our D2 training on the same dataset. We then processed the YOLOv5 and D2 detections using OOB and neural learning techniques. Fig. 6.4 provides an overview of the data flows as well as various fused bounding-box output locations.

Table 6.10: Updated class counts for 11-class experiments used in bounding-box ensembling experiments. An approximate 80-20 partition was achieved on a per-class basis using the partitioning method described in Section 6.2.2. Primary object classes of interest used in the partitioning schema are shaded in blue.

Class	Total	Train	Test	Train(%)	Test (%)
Image Count	61,653	51,610	10,043	83.7%	16.3%
<i>Combat Support Vehicle</i>	7,146	6,149	997	86.0%	14.0%
<i>Confuser Vehicle</i>	14,312	12,117	2,195	84.7%	15.3%
<i>LP w/ Revetment</i>	1,199	984	215	82.1%	17.9%
<i>LP</i>	19,086	16,004	3,082	83.9%	16.1%
<i>SAM TEL</i>	72,706	61,050	11,656	84.0%	16.0%
<i>Other TEL</i>	1,811	1,546	265	85.4%	14.6%
<i>SAM Missile</i>	11,173	10,002	1,171	89.5%	10.5%
<i>SAM LP</i>	8,184	6,862	1,322	83.8%	16.2%
<i>SAM LP w/ Revetment</i>	11,967	9,982	1,985	83.4%	16.6%
<i>Weapon</i>	3,010	2,569	441	85.3%	14.7%
<i>SAM Launcher</i>	2,611	2,313	298	88.6%	11.4%

6.4.1 Updated and Reduced Bounding-Box Ensembling Dataset

An updated dataset, DeepNET data v2021.03, became available because of ongoing data curation, and this was used in the bounding-box model ensembling experiments described in this section. This dataset was corrected in the same manner as the original dataset as described in Section 6.2.1. Additionally, we found that *SAM TELs* tended not to appear individually in scenes. Consequently, we decided to further constrain the scene set (see itemized list in Section 6.2.1) by requiring at least two *SAM TELs* to be present for a scene to be valid. Updated image and class counts are provided in Table 6.10. Note that we are aware the partition is slightly off the desired 80-20 split, and this is because the additional scene constraint criterion was enforced after partitioning.

6.4.2 OOB Bounding-Box Model Ensembling Techniques

Bounding-box model ensembling is the process of combining a set or sets of bounding boxes in order to refine the bounding-box geometry and/or reduce the number of overlapping bounding boxes. bounding boxes are generally selected for ensembling using an IoU $>$ a certain threshold between overlapping bounding boxes (Section 6.3.4). For these experiments we used an IoU threshold = 0.25. A common ensembling technique called Non-Maximum Suppression (NMS) [23] removes or suppresses all bounding boxes within a set of bounding boxes, B , that share a significant IoU with each other and have a confidence less than the maximum confidence. A less aggressive technique called soft-NMS or linear-NMS [24] does not eliminate bounding boxes in B with lower confidences, but instead scales the confidence either using a Gaussian or linear function. Non-Maximum Weighted (NMW) suppression [25] [26] used a more sophisticated approach by reducing the number of potential false positives by utilizing information found in the bounding boxes in B with lower class confidences. This is accomplished by weighting each bounding box in B by the product of the bounding box’s class confidence and the IoU with the bounding box with the highest confidence in B .

More recently, Weighted Box Fusion (WBF) [27] was introduced. This technique simplifies the weightings of B using the confidence of the bounding boxes. In addition [27] also introduced alternatives to computing the final bounding-box confidence that uses the maximum confidence from B (as in NMS and NMW). Suggested techniques include using the average confidence score, the “weighted average for boxes” which allows weighting of scores from individual models, and the “absent model aware weighted average” which scales confidences more positively when more models are present within the IoU boxes. We used the publicly available companion source code for [27] at GitHub repository <https://github.com/ZFTurbo/Weighted-Boxes-Fusion> [58] to calculate the results for the OOB ensembling methods.

Table 6.11: Table of abbreviations for publicly available bounding-box ensembling methods.

Abbreviation	Method
NMS	Non-Maximum Suppression [23].
NMSsoft	Soft Non-Maximum Suppression [24].
NMSlin	Linear Non-Maximum Suppression [24].
NMW	Non-Maximum Weighting [25] [26].
WBFarr	Weighted Box Fusion w/ max confidence of bounding-box set [27].
WBFmean	Weighted Box Fusion w/ mean confidence of bounding-box set [27].
WBFbama	Weighted Box Fusion w/ weighted confidence based on the number of bounding boxes in set and the total number of models [58].
WBFamaa	Weighted Box Fusion w/ weighted confidence based on the number of bounding boxes in set and the total number of models represented by the set of boxes [58].

6.4.3 Bounding-Box Pairing for Neural Network Datasets

Unlike the methods in the previous section that allows B to contain any number of bounding boxes with a significant IoU, neural networks do not respond well to inputs with unknown size. Consequently, we needed to create consistently sized input datasets for neural learning approaches. To do this we uniquely paired bounding boxes from distinct models as described in Procedure 2. The results of the list of bounding-box pairs, P , are then used to create input and truth datasets to train and test the neural networks. Each bounding-box pair is represented as an input vector and corresponding a truth vector. The input and truth vectors consist of the following parts:

1. Input vector ¹
 - (a) If bounding box is present for model then include:
 - i. Bounding-box coordinates
 - ii. Appropriate bounding-box metrics (described in Sections 6.4.4 and 6.4.5)
 - iii. Confidence vector created by filling each appropriate class index with the confidence from the bb and each duplicated bounding box in bb_D .
 - (b) Else:

¹Input and output vector sizes can be found in Table 6.12

- i. Fill in 0s for model values.
- 2. Truth vector
 - (a) If only predicting the confidence vector:
 - i. If truth bounding box is not *NULL*:
 - A. Confidence vector created by “one hot encoding” for the truth class index.
 - ii. Else:
 - A. Create truth vector with “one-hot encoding” with a value in the *BG* class index (i.e. the 12th position for an 11-class model).
 - (b) Else:
 - i. If truth bounding box is not *NULL* then include:
 - A. Bounding-box coordinates
 - B. Appropriate bounding-box metrics (described in Sections 6.4.4 and 6.4.5)
 - C. Confidence vector created by “one hot encoding” for truth class index.
 - ii. Else:
 - A. Fill in 0s for bounding-box coordinate and metric value.
 - B. Create truth vector with “one-hot encoding” with a value in the *BG* class index.

6.4.4 Multi-Layer Perceptrons

In order to develop novel bounding-box fusion/ensembling approaches, we applied the learning capabilities of neural network architectures to improve confidences and predict the final bounding-box coordinates and metrics. First, we tested a Multi-Layer Perceptron (MLP). The MLP architecture consisted of four fully-connected layers of 100 nodes each with ReLU activation between each layer. Because we want to predict both bounding-box coordinates, metrics, and confidence, the output of the MLP’s final layer used two different final-layer activation functions. The bounding-box coordinates and metrics were processed through a clamping function within the range [0-1] and the confidence vector was generated using a softmax normalization. In addition, experiments allowed the system to: i) learn the new confidence vector

Procedure 2 Generate Bounding-Box Pairings for Single Image

Input: Prediction Bounding-Box Sets B^α, B^β , Truth Bounding-Box set T ,
IoU threshold IoU_{thresh}

Output: Bounding-Box Prediction Pairs P

main

$P = \emptyset$

for all bb in B^α and B^β **do**

$U = \operatorname{argmax}([\operatorname{IoU}(bb,t) \text{ for } t \text{ in } T])$

if $\max(U) \geq IoU_{thresh}$ **then**

$bb_{ti} = \operatorname{argmax}(U)$

else

$bb_{ti} = \text{Null}$

end if

end for

for all $bb^\alpha \in B^\alpha$ **do**

$bb_D^\alpha = [bb[x_1, y_1, x_2, y_2] == bb^\alpha[x_1, y_1, x_2, y_2] : bb \in B^\alpha]$ // identify duplicates

$U = [\operatorname{IoU}(bb^\alpha, bb^\beta) \text{ for } bb^\beta \in B^\beta]$

if $\max(U) \geq IoU_{thresh}$ **then**

$i = \operatorname{argmax}([\operatorname{IoU}(bb^\alpha, bb^\beta) \text{ for } bb^\beta \in B^\beta])$

$bb^\beta = B^\beta[i]$

if $bb^\beta == bb_{ti}^\alpha$ **then**

$bb_D^\beta = [bb[x_1, y_1, x_2, y_2] == bb^\beta[x_1, y_1, x_2, y_2] : bb \in B^\beta]$ // identify duplicates

$P.\text{append}(bb^\alpha, bb^\beta)$

$B^\beta.\text{remove}(bb^\beta)$

$B^\beta.\text{remove_all}(bb_D^\beta)$

else

$P.\text{append}(bb^\alpha, \text{NULL})$

end if

else

$P.\text{append}(bb^\alpha, \text{NULL})$

end if

$B^\alpha.\text{remove}(bb^\alpha)$

$B^\alpha.\text{remove_all}(bb_D^\alpha)$

end for

for all $bb^\beta \in B^\beta$ **do**

$bb_D^\beta = [bb[x_1, y_1, x_2, y_2] == bb^\beta[x_1, y_1, x_2, y_2] : bb \in B^\beta]$ // identify duplicates

$P.\text{append}(\text{NULL}, bb^\beta)$

$B^\beta.\text{remove}(bb^\beta);$

$B^\beta.\text{remove_all}(bb_D^\beta)$

end for

end

exclusively from any other metric, and ii) split the network into two independent branches after the initial architecture to allow the bounding-box coordinates and metrics to train in parallel before final activation. Diagrams for the initial MLP

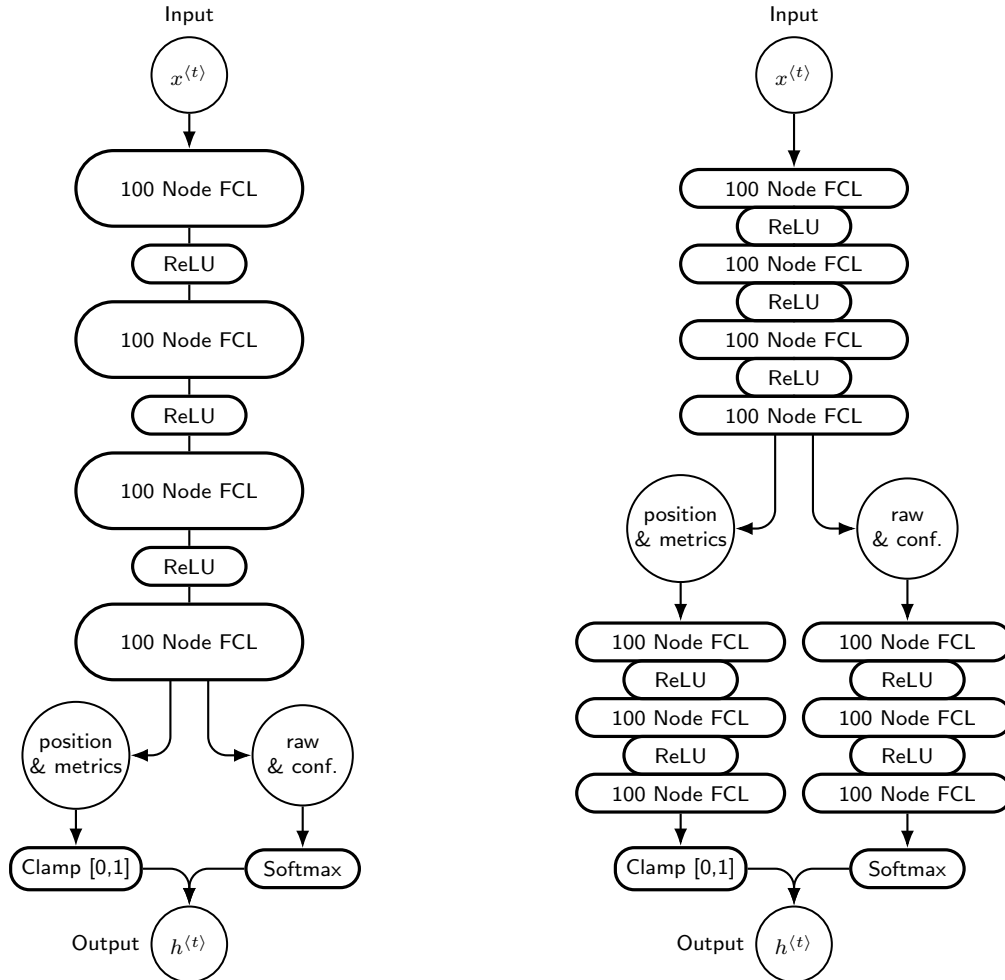


Fig. 6.5: DNN architectures used in the MLP experiments. Inputs varied depending on dataset type. The clamping activation function was only used when applicable. FCL is an abbreviation for Fully-Connected Layer.

architecture and the split architecture are provided in Fig. 6.5.

6.4.4.1 Bounding-Box Metrics and Datatype

All of the datasets used in our experiments were primarily in Normalized Image Space (NImgS). NImgS encodes the bounding-box coordinates as a percentage of the width and height of the overall image. For example, if a 32×64 pixel bounding box was centered in the middle of an 512×512 pixel image then the bounding-box coordinates $[x_1, y_1, x_2, y_2]$ would be $[0.468, .438, 0.531, 0.523]$. We first experimented using only the bounding-box coordinates and confidence vector for each pair, without

Table 6.12: Metric types included in MLP dataset as well as input/output size vector size. Bounding-box coordinates include top left and bottom right coordinates for each model, $[x_{1a}, y_{2a}, x_{2a}, y_{2a}]$ and $[x_{1b}, y_{2b}, x_{2b}, y_{2b}]$.

Dataset	BB Locations	Confidence Metrics	NImgS Metrics	NMtrS Meter	Input Size	Output Size
Confidence Only Prediction						
INFER	X	X			30	12
NImgS	X	X	X		36	12
NMtrS	X	X		X	36	12
KS	X	X	X	X	42	12
Confidence w/ BB Coordinates and Metrics Prediction						
INFER	X	X			30	16
NImgS	X	X	X		36	19
NMtrS	X	X		X	36	19
KS	X	X	X	X	42	22

any additional bounding-box metrics. We next added NImgS bounding-box metrics such as the bounding-box width, bounding-box height, and the ratio of the minimum over the maximum bounding-box dimensions. We next computed the same metrics in Normalized Meter Space (NMtrS). NMtrS uses the scene GSD and pixel width and height to compute the bounding-box size in terms of meters. Because neural networks tend to train better with values between 0 and 1, we divided the metric values by 128. As in the xView data (Section 4.4.1), there may be differences between the vertical and horizontal GSDs. Therefore, a separate bounding-box dimension ratio was also calculated in NMtrS. We experimented using the NImgS and NMtrS metrics independently, as well in combination for what we called the Kitchen Sink (KS) experiments. Table 6.12 shows which metrics were included for the various experiments and the size of the input and output data vectors.

6.4.5 Pseudo-Cell State (PCS) Long-Short Term Memory (LSTM)

One of the advantages of using Long-Short Term Memory (LSTM) cell architectures [59] (Fig. 6.6) is that the predicted output and cell state from a previous iteration

can be used to predict the output for the next set of inputs for serialized input data (e.g. temporal sequences). Since we do not have serial input data, we explored using bounding-box metrics as a quasi or pseudo memory for the initial cell state of LSTM cells to improve bounding-box classification. To accomplish this we augmented the traditional LSTM cell in the following ways:

1. Most LSTM implementation train using cell-state vector that is the same size as the input vector for computational simplicity. However, we narrowed the initial cell state to a single scalar and reduced the batch size to one since we do not have many bounding-box metrics. Increasing the batch size and larger pseudo-cell state vectors could be explored in future work.
2. LSTMs sometimes have a difficult time converging with spatial input datasets such as those used in this research. Therefore, we introduced normalized random noise with $\mu = 0.005$ and $\sigma = 0.005$.
3. We removed the initial hidden component ($h^{<t-1>}$ in Fig. 6.6) because it is generally initialized: i) using a zero vector which would serve no purpose, or ii) randomly which would introduce more noise to the system which was already added to the data directly.
4. We used multiple models where each model had a different bounding-box metric used for the pseudo-cell state. Thus, we used a two-layer LSTM where the first layer processes the confidence vector for each model using its individual bounding-box metric for the pseudo cell state. Next, the cell state and confidence vectors produced by the first layer were used as the input cell state and inputs values for a second-layer LSTM cell. The output from the second-layer was then fed into a final FCL and softmax activation function to produces a final confidence vector (see diagram in Fig. 6.7).

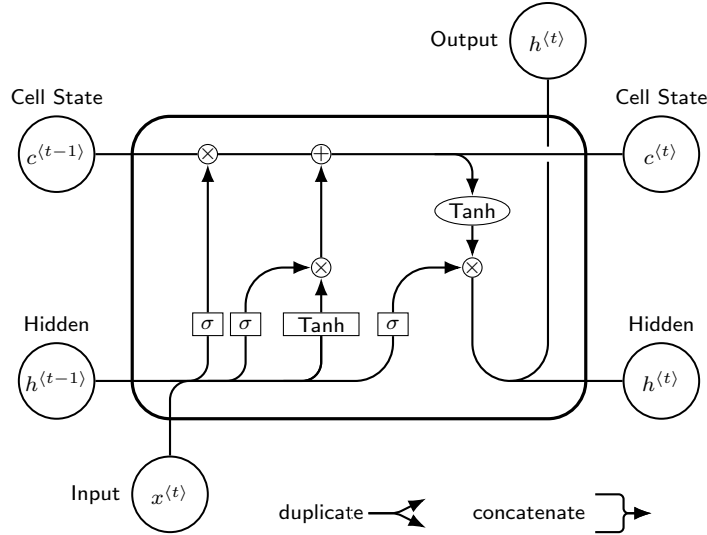


Fig. 6.6: Diagram for a common Long Short Term Memory (LSTM) cell architecture with the addition of the softmax activation function before the output.

We chose not to predict bounding-box metrics with PCS-LSTMs, but believe this could be worth exploring in future research.

6.4.5.1 Bounding-Box Metrics and Datatype

The input vector for each model consists only of the computed confidence vector described in Section 6.4.3. The pseudo cell state used only one of the following bounding-box metrics:

1. The max bounding-box dimension in NImgS.
2. The max bounding-box dimension in NMtrS.
3. The bounding-box dimension ratio in NImgS.
4. The bounding-box dimension ratio in NMtrS.

6.4.6 Bounding-Box Coordinate Computation

Since we are interested in the ability of the MLPs to both improve the confidences as well as predict refined coordinates and bounding-box metrics, we used a variety of methods to predict the final bounding-box outputs. Table 6.13 provides the abbrevi-

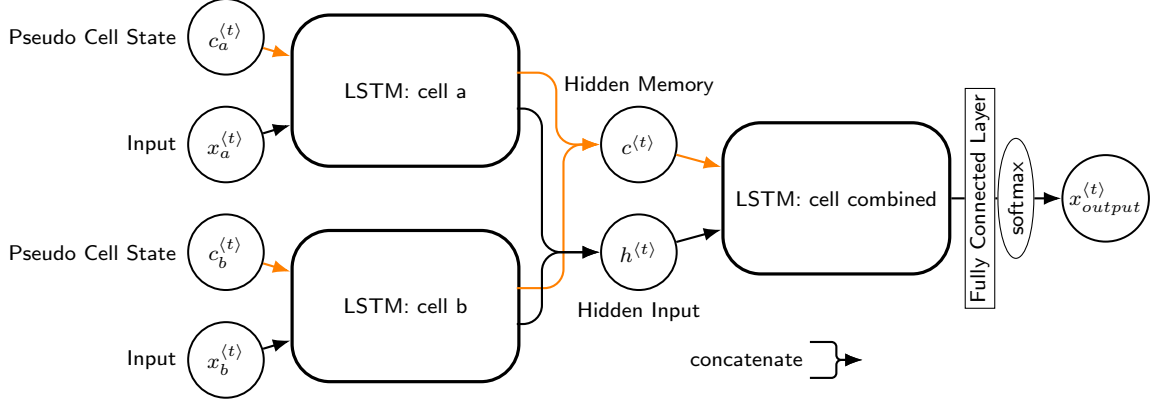


Fig. 6.7: Diagram of the Pseudo-Cell State Long-Short Term Memory (PCS-LSTM) neural network architecture. A two-step LSTM with input from two bounding-box detection sources (a & b) utilizing pseudo memory cell states. The first-layer input is a class confidence vector for the respective bounding box. The first-layer pseudo memory cell state is one of four bounding-box metrics: the max bounding-box dimension as a percentage of the image, the max dimension in normalized meter space (i.e. length in meters/128), or the ratio of the bounding-box dimensions in normalized image or meter space (i.e. $\max[\text{dim}]/\min[\text{dim}]$). The second LSTM layer takes the concatenated cell states ($c^{<t>}$) and output ($h^{<t>}$) from the first layer as input. The output from the second layer is then processed through a final fully-connected layer and softmax function to produce a new bounding-box confidence vector.

ations used in the results section and descriptions for each method of computing the final bounding boxed.

6.4.7 Minimize Expected Calibration Error

We found in initial testing that optimized confidence thresholds used to achieve the highest $F1$ score for different object-detection models could be drastically different. Therefore, we ran experiments to determine if calibrating the models confidences could improve the ensembling results. A frequently used calibration method is the Expected Calibration Error (ECE) [60]. The ECE is a measurement used to compute the difference between the expected accuracy and expected confidence:

$$ECE = \sum_{i=1}^K P(i) \cdot |o_i - r_i|,$$

Table 6.13: Abbreviations for different bounding-box coordinate computation methods.

Abbreviation	Method
arr	The bounding-box coordinates for the bounding box in the pair with highest total confidence.
mean	Mean of the bounding-box coordinates in the pair.
wbf	Weighted bounding-box fusion of the bounding-box coordinates using total confidence as weight.
NNout	Bounding-box coordinates produced by the neural network.
imgWHarr	NN-computed width & height in NImgS using the arrogance of the center coordinates of the paired bounding boxes.
imgWHmean	NN-computed width & height in NImgS using the mean of the center of the paired bounding boxes.
imgWHwbf	NN-computed width & height in NImgS using the wbf of the center coordinates of the paired bounding boxes.
mtrWHarr	NN-computed width & height in NMtrS converted to NImgS using the arrogance of the center coordinates of the paired bounding boxes.
mtrWHmean	NN-computed width & height in NMtrS converted to NImgS using the mean of the center coordinates of the paired bounding boxes.
mtrWHwbf	NN-computed width & height in NMtrS converted to NImgS using the wbf of the center coordinates of the paired bounding boxes.
...TEL	Methods in this table used to compute bounding boxes, but with <i>SAM_TEL</i> only confidences and class index.

where o_i is the precision in bin i , e_i is the mean of confidences for the instances in bin i , and $P(i)$ is the recall of all instances that fall into bin i . To calibrate, we first computed the precision for a set confidence thresholds using an IoU @ 0.25. Next we computed a “calibration curve” by interpolating between the original confidences and the precision. This curve was then used to map the original bounding-box confidence to the new ECE-calibrated confidence.

6.4.8 Results

Results for OOB ensembling of the YOLOv5 and D2 models direct outputs are given in Table 6.14. We see that the OOB ensembling improves upon the individual $F1$ detector results by $\sim 4\%$. In addition, the NMW and NMS ensembling techniques seem to do best for both the individual detector results and ECE calibrated results,

Table 6.14: Results for top and selected two-detector ensembling.

Method§	ECE	OOB Ens. Type	Prec.	Recall	F1	Thresh
OOB	X	NMW	69.6%	81.5%	75.1%	0.731
OOB	X	NMS	69.4%	81.3%	74.9%	0.365
OOB		WBFmax	70.3%	80.2%	74.9%	0.443
OOB		NMW	73.5%	76.3%	74.9%	0.429
OOB		NMS	73.1%	76.6%	74.8%	0.220
OOB		WBFamaa	73.2%	76.3%	74.8%	0.239
OOB	X	WBFbama	70.8%	78.6%	74.5%	0.362
OOB		WBFbama	71.7%	77.2%	74.3%	0.215
OOB		WBFavg	73.5%	75.2%	74.3%	0.273
OOB	X	WBFamaa	69.8%	77.9%	73.6%	0.482
OOB	X	WBFmax	65.6%	83.5%	73.5%	0.727
OOB	X	WBFavg	67.2%	77.8%	72.1%	0.640
D2		n/a	66.2%	77.6%	71.5%	0.659
D2	X	n/a	67.0%	76.5%	71.4%	0.669
Ycoco		a/a	68.1%	73.9%	70.9%	0.372
OOB		NMSsoft	62.7%	81.1%	70.7%	0.384
OOB		NMSlin	60.7%	8.1%	70.7%	0.384
Ycoco		n/a	65.7%	76.5%	70.7%	0.726
OOB	X	NMSlin	6.9%	59.9%	64.0%	0.425
OOB	X	NMSsoft	68.7%	59.8%	63.9%	0.424

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Ycoco’:YOLOv5 w/ COCO pre-training, ‘YxView’:YOLOv5 w/ xView pre-training.

although the top nine results are within a single percentage point of each other. It is also interesting that NMS and NMW seem to work the best even though they utilize conflicting strategies. Specifically, NMS is the most forgetful of overlapping bounding boxes while NMW utilizes as much possible information from overlapping bounding boxes. The impact of ECE calibration on the various methods was inconclusive.

Table 6.15 shows that the top five results after completing neural-learning ensembling are still achieved by OOB techniques. In fact the only neural ensembling to improve upon the single detectors was PCS-LSTM which rank 27 places above D2. Notably WBF bounding-box coordinate computation yielded the best results for every type of neural architecture. The candle-stick plots in Fig. 6.8 re-enforce this

Table 6.15: Top five results for bounding-box ensembling along with the top results for each type of neural-network ensembling. IDX is the ranking index of all methods tested.

IDX	Method§	ECE	Input Data	Box Comp.	TEL Only	OOB Type	Prec.	Recall	F1	Thresh
1	OOB	X	Orig.			NMW	69.6%	81.5%	75.1%	0.731
2	OOB	X	Orig.			NMS	69.4%	81.3%	74.9%	0.365
3	OOB		Orig.			WBFmax	70.3%	80.2%	74.9%	0.443
4	OOB		Orig.			NMW	73.5%	76.3%	74.9%	0.429
5	OOB		Orig.			NMS	73.1%	76.6%	74.8%	0.220
Top Results for Other Ensembling Methods										
10	PCS-LSTM		NMtrS Max	wbf	X		73.6%	73.8%	73.7%	0.640
37	D2		Orig.				66.2%	77.6%	71.5%	0.659
39	Ycoco		Orig.				68.1%	73.9%	70.9%	0.372
43	MLP		NImgS	wbf			69.7%	71.4%	70.5%	0.728
49	MLP CO		INFER	wbf	X		68.0%	72.5%	70.2%	0.751
73	MLP split		NImgS	wbf	X		70.5%	68.5%	69.5%	0.746
157	PCS-LSTM	X	NImgS ratio	wbf			71.0%	60.3%	65.2%	0.718
175	MLP CO	X	INFER	wbf	X		66.2%	62.6%	64.4%	0.853
181	MLP	X	INFER	wbf			69.2%	60.1%	64.3%	0.834
205	MLP split	X	INFER	wbf	X		68.2%	60.4%	64.1%	0.843

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Ycoco’:YOLOv5 w/ COCO pre-training, ‘YxView’:YOLOv5 w/ xView pre-training.

observation. We can clearly see that arrogance, mean, and WBF were the best and most consistent choices for bounding-box coordinate computation for both the original data and ECE calibration. It is also notable that WBF yielded the best results for ALL of the neural ensembling techniques. Figure 6.9 provides a bar chart showing the best *F1* results for each neural architecture. The original data does better than ECE-calibrated data besides, of course, the OOB ensembling techniques.

As mentioned previously the input restraints for the MLPs and PCS-LSTM neural learning methods prevents them from being able to ensemble more than two bounding boxes at a time. This, therefore, gives the OOB methods an advantage since they are able to accommodate more than two input bounding boxes. Consequently, the results from the neural learning approaches have the potential for improvement using OOB ensembling methods on the backend since there are likely additional overlapping bounding boxes in the neural ensembling outputs. Therefore, we tested each type of

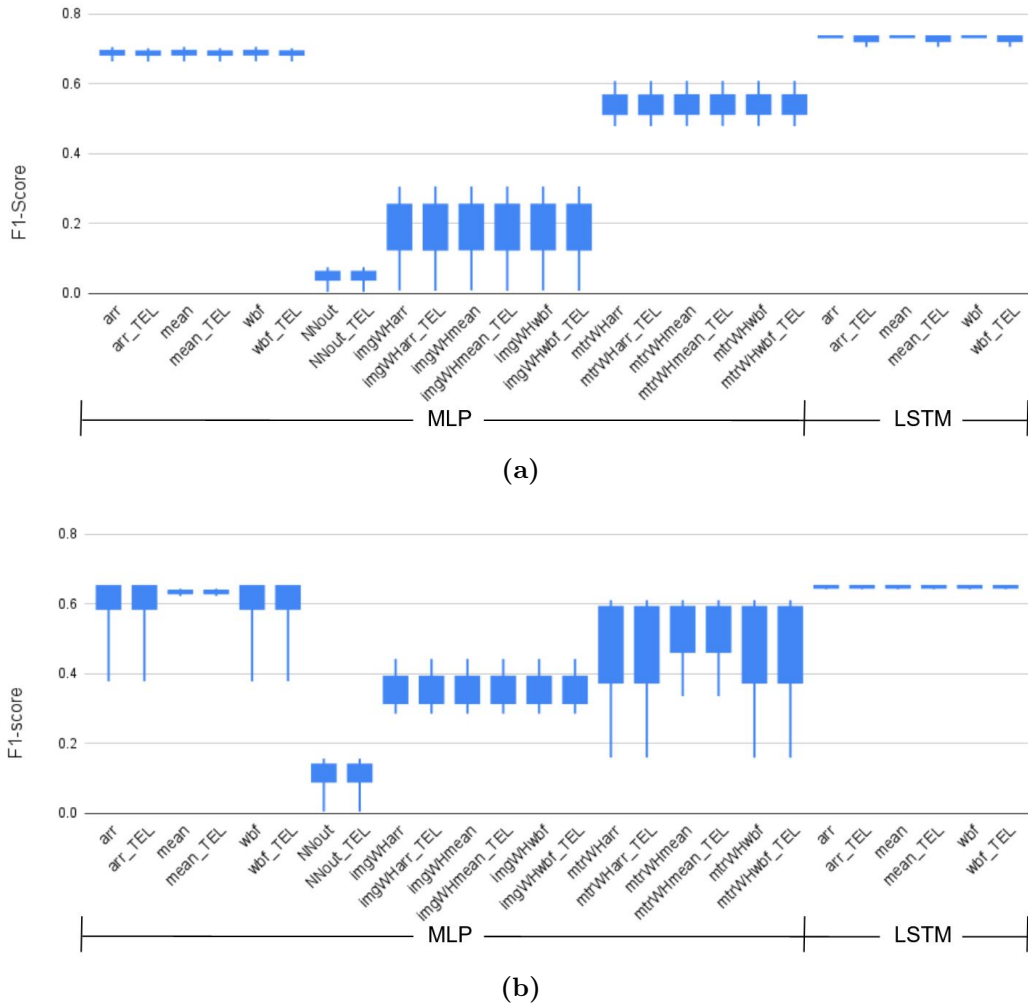


Fig. 6.8: Candle-stick charts showing the $F1$ score ranges for different types bounding-box coordinate computations after neural-learning ensembling. (a) Results for using data directly from YOLOv5 and D2. (b) Results for YOLOv5 and D2 after ECE calibration. Top and bottom of the boxes are $0.5 \cdot \text{StDev}$.

OOB ensembling for the neural ensembling outputs. The results of these experiments are presented in Table 6.16. The results show that backend OOB ensembling was able to elevate both the PCS-LSTM models and the MLP, split MLP, and confidence prediction MLP for ECE calibrated data results (i.e. all neural learning ensembling methods) above the initial OOB ensembling. In almost every case the top results from backend OOB ensembling was achieved using non-maximum weighting, which was the same as for the baseline OOB ensembling. This is reinforced by the candle-stick

Table 6.16: Results for bounding-box neural-learning ensembling with backend OOB ensembling. IDX is the ranking index of all methods tested.

IDX	Method§	ECE	Input Data	Box Comp.	TEL Only	OOB Type	Prec.	Recall	F1	Thresh
1	PCS-LSTM	X	NMtrS max	arr	X	NMS	70.3%	82.6%	75.9%	0.609
2	PCS-LSTM	X	MImgS ratio	arr		NMW	70.4%	82.3%	75.9%	0.606
3	PCS-LSTM	X	NMtrS ratio	arr		NMW	71.3%	81.2%	75.9%	0.597
4	PCS-LSTM	X	NMtrS ratio	arr	X	NMW	70.4%	82.4%	75.9%	0.608
5	PCS-LSTM	X	NMtrS ratio	mean	X	NMS	70.3%	82.5%	75.9%	0.609
Top Results for Other Ensembling Methods										
28	MLP	X	KS	wbf		NMW	69.7%	82.6%	75.6%	0.718
49	PCS-LSTM		NImgS ratio	arr	X	NMW	71.0%	80.7%	75.5%	0.570
51	MLP split	X	KS	wbf		NMW	70.9%	80.7%	75.5%	0.729
104	MLP CO	X	INFER	wbf		NMW	69.4%	82.0%	75.2%	0.697
186	OOB	X	Orig.			NMW	69.6%	81.5%	75.1%	0.731
225	OOB		Orig.			WBFmax	70.3%	80.2%	74.9%	0.443
423	MLP CO		KS	arr	X	WBFmax	70.1%	78.3%	74.0%	0.644
433	MLP		NImgS	arr	X	NMW	70.9%	77.2%	73.9%	0.578
461	MLP split		KS	arr	X	NMW	68.4%	80.0%	73.7%	0.672
1091	D2		Orig.				66.2%	77.6%	71.5%	0.659
1124	Ycoco		Orig.				68.1%	73.9%	70.9%	0.372

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Ycoco’:YOLOv5 w/ COCO pre-training.

plot in Fig. 6.12. Here we see that each ensembling technique yielded a maximum improvement $> 20\%$. Also, NMS, NMW, and WB ensembling techniques (using maximum confidence in B) were more consistent at producing improved results.

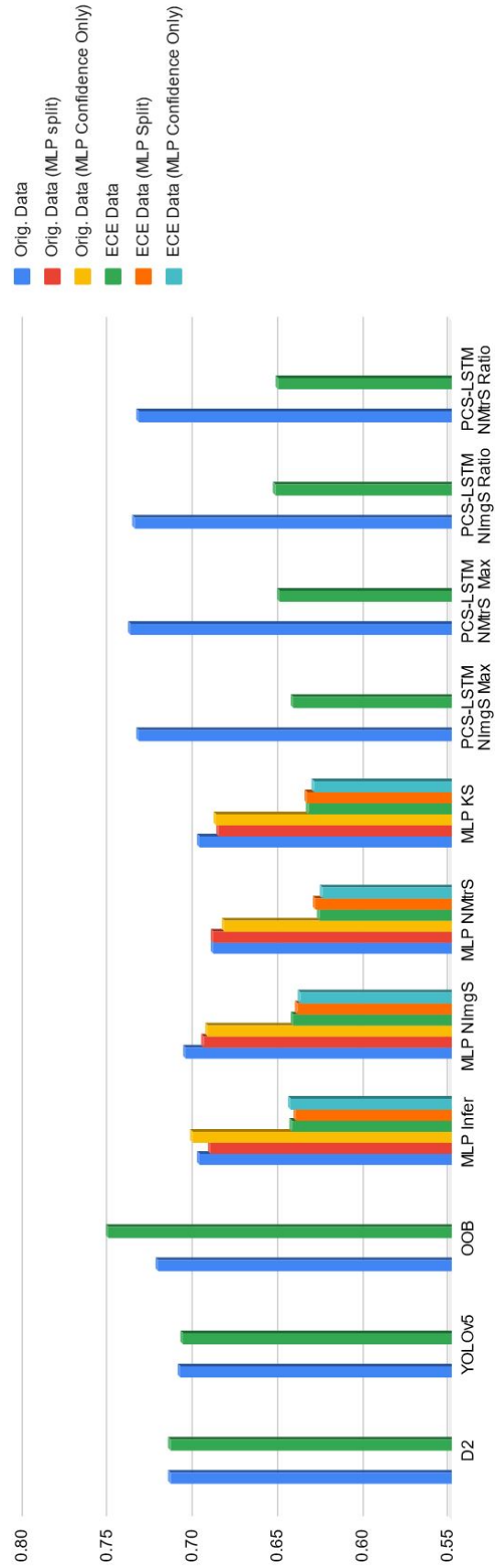


Fig. 6.9: Best $F1$ score results for two detector ensemble techniques.

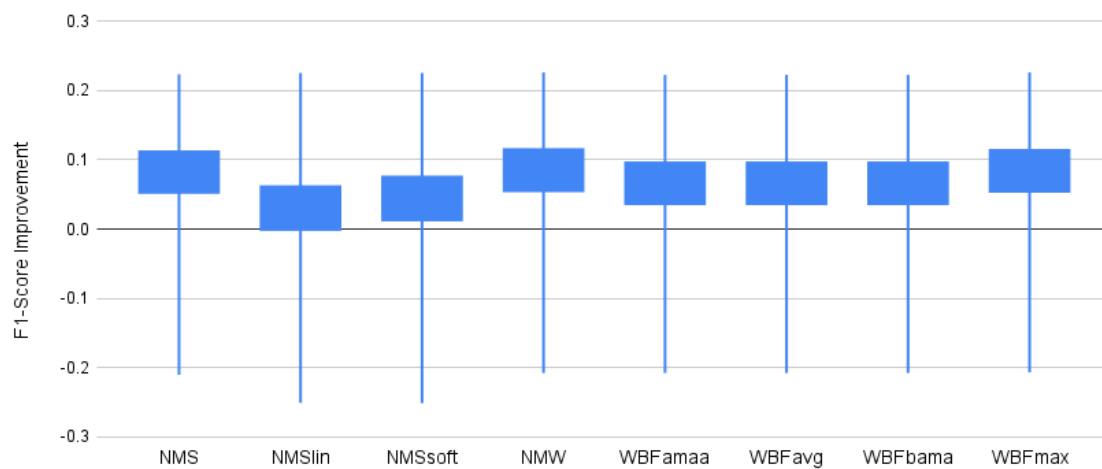


Fig. 6.10: Candle-stick plots showing $F1$ score improvement ranges for different OOB ensembling techniques applied after neural-learning ensembling. Top and bottom of the boxes are $0.5 \cdot \text{StDev}$.

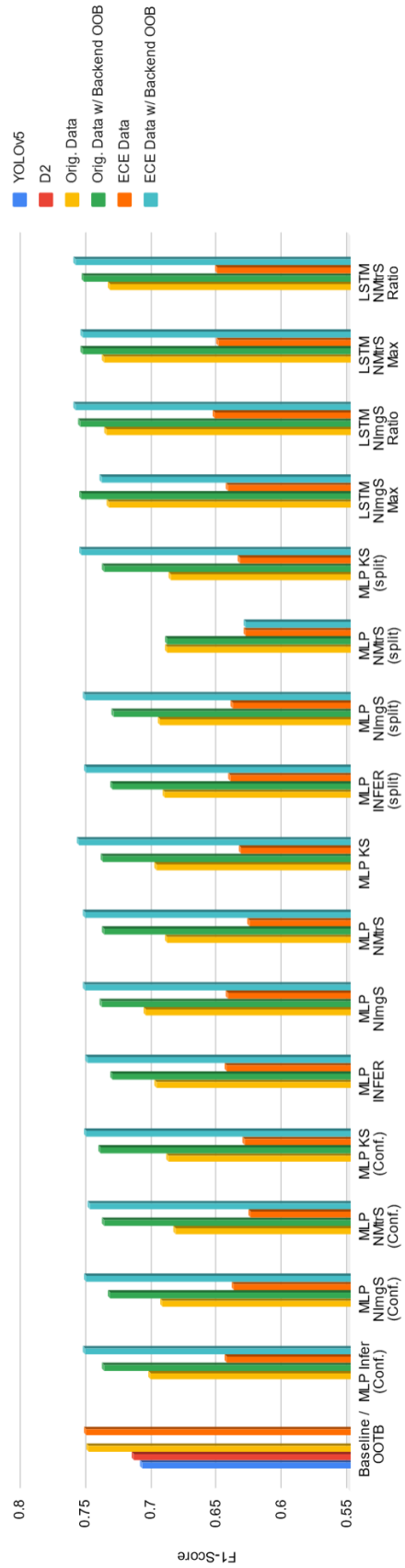


Fig. 6.11: Best $F1$ score results for two detector ensemble techniques with OOB ensembling on the backend.

6.5 BOUNDING-BOX FUSION/ENSEMBLING FOR THREE DETECTOR INPUTS

6.5.1 Additional Model & Extended Training

In general, more information can be gleaned using additional information sources. Consequently, we introduced a third detector to test the extensibility of the MLP and PCS-LSTM neural learning methods. To do this we started with the COCO pre-trained weights and added additional 150 epochs of training using the complete xView [45] dataset. We then used these pre-trained xView weights and trained the YOLOv5 detector using the reduced dataset described in Section 6.4.1 for an additional 150 epochs. We also trained a detector using the weights produced by training the 11-class YOLOv5x model in Section 6.3.1 for an additional 100 epochs on the complete DeepNET v2021.3.2 dataset and then 150 epochs on the pruned DeepNET v2021.3.2 dataset. The initial training of D2 detector already showed a “leveling out” of the loss curve; thereby indicating that little improvement could be obtained from further training. Consequently, no additional training was performed for the D2 detector.

6.5.2 Expanded MLP and PCS-LSTM

The MLP described in Section 6.4.4 was adapted to accommodate a larger input vector with little additional changes. For the PCS-LSTM an additional model cell was added to the first level to modify it compared to the PCS-LSTM from Section 6.4.5.

6.5.3 Results

The baseline and initial OOB ensembling results are provided in Table 6.17. We see that once again OOB ensembling improved over the original model outputs, but only by a couple of percentage points for the $F1$ score. This time the WBF ensembling did the best for the three input detectors, whereas NMS and NMW ensembling performed

Table 6.17: Results for top and selected three-detector ensembling.

Method§	ECE	OOB Type	Prec.	Recall	F1	Thresh
OOB	X	WBF _{amaa}	75.5%	85.2%	80.1%	0.567
OOB		WBF _{max}	77.5%	82.7%	80.0%	0.362
OOB		WBF _{amaa}	76.7%	83.0%	79.8%	0.245
OOB		WBF _{bama}	76.2%	83.5%	79.7%	0.224
OOB	X	NMW	77.1%	81.9%	79.4%	0.855
OOB	X	NMS	77.1%	81.8%	79.4%	0.285
OOB		WBF _{avg}	76.5%	82.5%	79.4%	0.280
OOB: Y _{coco} -D2	X	NMS	75.8%	83.3%	79.4%	0.428
OOB	X	WBF _{bama}	70.6%	89.5%	78.9%	0.559
OOB	X	WBF _{max}	71.2%	88.2%	78.8%	0.850
Y _{coco}			76.0%	81.4%	78.6%	0.370
OOB		NMW	77.7%	79.5%	78.5%	0.462
OOB		NMS	77.5%	79.4%	78.4%	0.154
OOB: Y _{coco} -D2		NMW	78.0%	78.1%	78.1%	0.415
OOB	X	WBF _{avg}	77.7%	77.8%	77.7%	0.564
Y _{coco}	X		79.6%	75.2%	77.4%	0.856
Y _{xView}			71.4%	79.3%	75.2%	0.335
Y _{xView}	X		70.4%	80.6%	75.1%	0.824
D2			66.2%	77.6%	71.5%	0.659
D2	X		67.0%	76.5%	71.4%	0.669
OOB		NMS _{lin}	61.3%	82.1%	70.2%	0.265
OOB		NMS _{soft}	61.3%	82.0%	70.1%	0.265
OOB	X	NMS _{lin}	75.3%	49.2%	59.5%	0.294
OOB	X	NMS _{soft}	70.4%	48.9%	57.7%	0.307

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Y_{coco}’:YOLOv5 w/ COCO pre-training, ‘Y_{xView}’:YOLOv5 w/ xView pre-training.

the best when there were only two input detectors. As before, it was inconclusive if ECE calibration helped improve upon the results.

Unlike the two-detector ensembling given in Section 6.4.8, we see from Table 6.19 that the three-detector PCS-LSTM was able to improve by a percentage point compared to the initial OOB ensembling. Though with these new detectors the PSC-LSTM for YOLOv5 with COCO pre-trained weights and D2 were able to match the initial OOB ensembling with the three detector inputs. Also, the bounding-box ratio

in NImgS was the most beneficial pseudo-cell state in training the PCS-LSTM. In addition, most of the neural ensembling techniques responded favorably to arrogant bounding-box coordinate computation.

As before, utilization of the backend OOB ensembling produced additional improvement. These results are presented in Table 6.19 and show a 2.5% improvement in $F1$ over the neural ensembling output from PCS-LSTMs using the maximum box dimension in NMtrS or MImgS. Once again, NMS and NMW yielded the best results for the backend OOB ensembling. Arrogant bounding-box coordinate computation also dominated the best results. The bar graph in Fig. 6.12 shows that ECE calibrated had a tendency to produce lower $F1$ scores after backend OOB ensembling, which was very similar to the two-detector ensembling results.

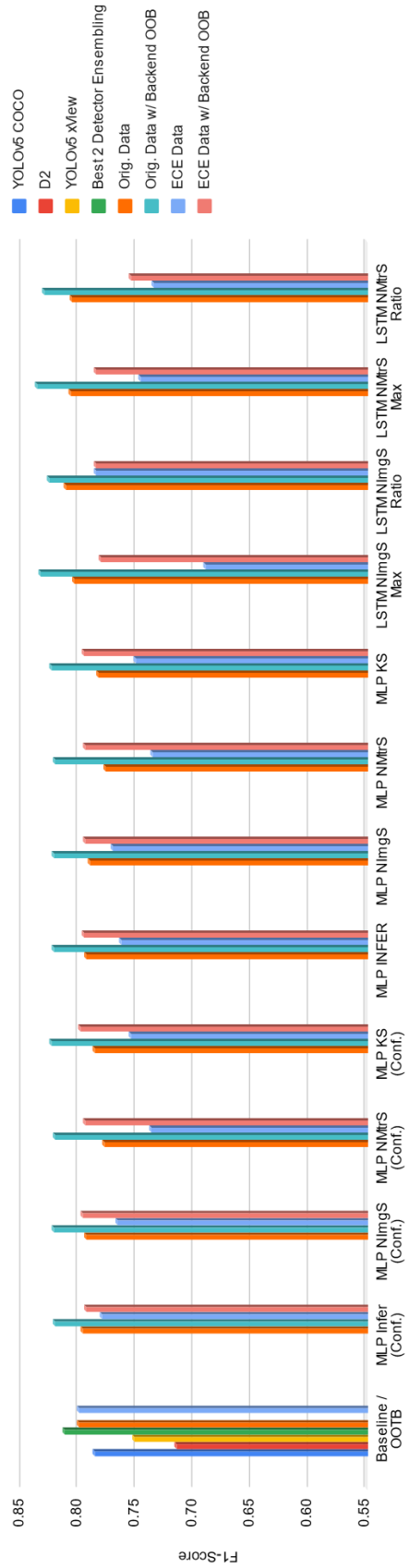


Fig. 6.12: Best $F1$ score results for three detector ensemble techniques with OOB ensembling on the backend.

Table 6.18: Results for three bounding-box detector ensembling. IDX is the ranking index of all methods tested.

IDX	Method§	Data Source	ECE	Input Data	Box Comp.	TEL Only	OOB Type	Prec.	Recall	F1 score	Thresh
1	PCS-LSTM	3 Detectors		NImgS ratio	arr			80.3%	82.0%	81.1%	0.548
2	PCS-LSTM	3 Detectors		NImgS ratio	mean			80.2%	82.0%	81.1%	0.548
3	PCS-LSTM	3 Detectors		NImgS ratio	wbf			80.2%	82.0%	81.1%	0.548
4	PCS-LSTM	3 Detectors		NImgS ratio	arr	TEL		80.2%	82.0%	81.1%	0.549
5	PCS-LSTM	3 Detectors		NImgS ratio	mean	TEL		80.2%	82.0%	81.1%	0.549

Top Results for Other Ensembling Methods											
25	PCS-LSTM	Ycoco-D2		NMtrS max	arr	TEL		79.9%	80.3%	80.1%	0.693
35	OOB	3 Detectors	X				WBFamaa	75.5%	85.2%	80.1%	0.567
38	OOB	3 Detectors					WBFmax	77.5%	82.7%	80.0%	0.362
41	MLP CO	3 Detectors		INFER	arr			80.0%	79.3%	79.6%	0.609
59	MLP	3 Detectors		INFER	mean			79.6%	79.2%	79.4%	0.603
64	OOB	Ycoco-D2	X				NMS	75.8%	83.3%	79.4%	0.428
88	Ycoco	Original						76.0%	81.4%	78.6%	0.370
92	PCS-LSTM	3 Detectors	X	NImgS ratio	arr	TEL		77.0%	80.1%	78.5%	0.697
98	OOB	Ycoco-YxView					NMW	72.5%	85.5%	78.5%	0.362
107	OOB	Ycoco-D2					NMW	78.0%	78.1%	78.1%	0.415
112	MLP CO	3 Detectors	X	INFER	arr	TEL		75.3%	80.8%	78.0%	0.790
136	MLP	3 Detectors	X	MImgS	arr	TEL		75.1%	79.3%	77.1%	0.791
144	OOB	Ycoco-D2					WBFamaa	76.4%	77.6%	77.0%	0.213
165	MLP	Ycoco-D2		INFER	arr			74.3%	76.8%	75.5%	0.816
172	MLP CO	Ycoco-D2		INFER	arr			73.9%	77.1%	75.5%	0.806
190	YxView	Original						71.4%	79.3%	75.2%	0.335
248	D2	Original						66.2%	77.6%	71.5%	0.659
263	MLP	Ycoco-D2	X	MImgS	arr	TEL		73.6%	65.0%	69.1%	0.885
283	MLP CO	Ycoco-D2	X	MImgS	arr			70.2%	67.8%	69.0%	0.955
298	PCS-LSTM	Ycoco-D2	X	NMtrS max	arr			75.1%	63.5%	68.8%	0.799

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Ycoco’:YOLOv5 w/ COCO pre-training, ‘YxView’:YOLOv5 w/ xView pre-training.

Table 6.19: Results for three bounding-box detector ensembling with backend OOB ensembling. IDX is the ranking index of all methods tested.

IDX	Method§	Data Source	ECE	Input Data	Box Comp.	TEL Only	OOB Type	Prec.	Recall	F1	Thresh
1	PCS-LSTM	3 Detectors		NMtrS max	arr		NMS	79.0%	88.8%	83.6%	0.313
2	PCS-LSTM	3 Detectors		NMtrS max	arr		NMW	79.0%	88.7%	83.6%	0.313
3	PCS-LSTM	3 Detectors		NMtrS max	arr	X	NMS	79.1%	88.1%	83.4%	0.449
4	PCS-LSTM	3 Detectors		NImgS max	arr		NMS	78.8%	88.5%	83.4%	0.334
5	PCS-LSTM	3 Detectors		NImgS max	arr		NMW	78.7%	88.4%	83.3%	0.334
Top Results for Other Ensembling Methods											
75	MLP	3 Detectors		KS	arr		NMW	77.8%	87.7%	82.4%	0.516
76	MLP CO	3 Detectors		KS	arr		NMW	79.3%	85.8%	82.4%	0.492
568	OOB	3 Detectors	X				WBFamaa	75.5%	85.2%	80.1%	0.567
579	OOB	3 Detectors					WBFmax	77.5%	82.7%	80.0%	0.362
589	MLP CO	3 Detectors	X	KS	arr		NMW	76.7%	83.3%	79.9%	0.652
645	MLP	3 Detectors	X	INFER	arr		NMS	77.2%	82.2%	79.6%	0.635
747	OOB	3 Detectors	X				NMS	75.8%	83.3%	79.4%	0.428
1055	Ycoco	Original						76.0%	81.4%	78.6%	0.370
1085	PCS-LSTM	3 Detectors	X	NImgS ratio	arr	X		77.0%	80.1%	78.5%	0.697
1894	YxView	Original						71.4%	79.3%	75.2%	0.335
2417	D2	Original						66.2%	77.6%	71.5%	0.659

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Ycoco’:YOLOv5 w/ COCO pre-training, ‘YxView’:YOLOv5 w/ xView pre-training.

6.6 CONCLUSION AND FUTURE WORK

In this research we developed and tested several technical approaches to compare DNN detections from image scanning (Section 2.3.2) + spatial clustering with bounding-box object detectors. We then developed and tested a variety of fusion/ensembling methods that used multiple bounding-box object detectors to improve the detection of *Surface-to-Air Missile Transporter Erector Launchers* (SAM TELs).

We showed that image scanning + spatial clustering significantly underperformed bounding-box detectors because the bounding boxes have to be heuristically inferred using the chip size from the DNN model. We then developed and tested a novel Pseudo Cell State LSTM (PCS-LSTM) that at least matched the $F1$ score the ‘Out-Of-the-Box’ (OOB) bounding-box ensembling techniques using two bounding-box object detector inputs. By using three bounding-box object detector inputs the PCS-LSTM produced a 1% gain in $F1$ compared to the OOB ensembling. We further showed that by adding backend OOB ensembling on top of the PCS-LSTM the $F1$ improvement increased to 3.5% compared to OOB ensembling alone and overall this was a 5% improvement relative to any single bounding-box object detector (see summary Table 6.20 and Fig. 6.13).

Future research in this area should explore: 1) developing methods to allow the MLP to actually learn the bounding-box coordinates, 2) extending the PCS-LSTM to receive an n -length vector as a pseudo cell state so that we can included multiple bounding-box metrics, 3) developing methods to allow the LSTM to learn bounding-box coordinates, and 4) exploring fuzzy learning techniques for bounding-box model ensembling.

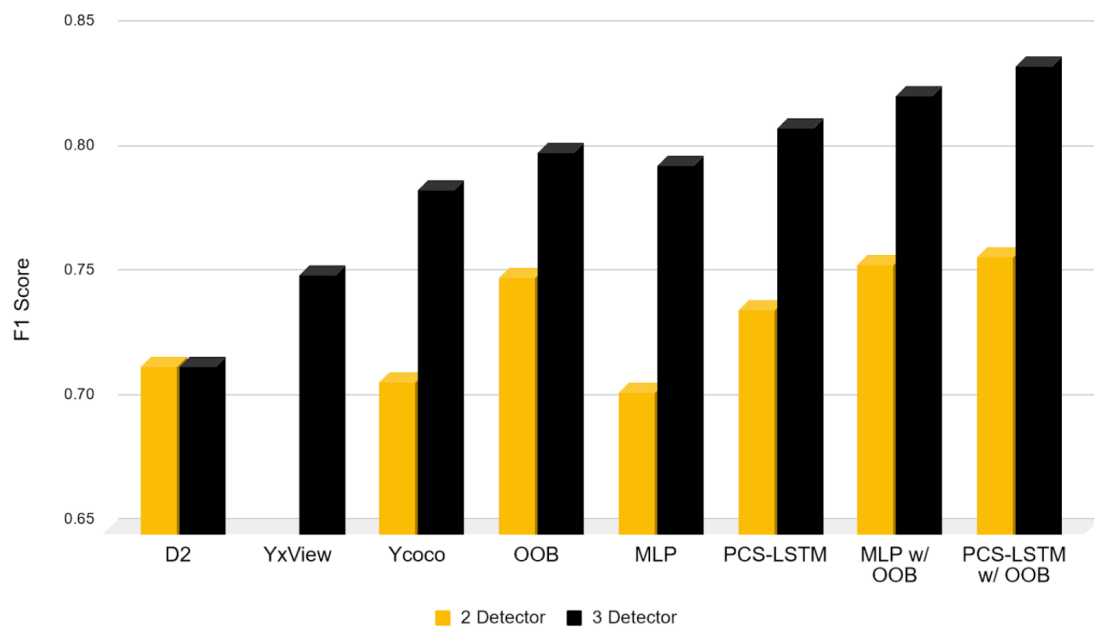


Fig. 6.13: Comparative summary of bounding-box ensembling for two and three detector inputs.

Table 6.20: Summary of neural-learning ensembling results compared to ‘Out-Of-the-Box’ ensembling and bounding-box detector outputs for two detector and the expanded three detector inputs.

Method§	F1	Processing Notes
Two Detector Ensembling Results		
PCS-LSTM w/ OOB	75.9%	Used ECE calibrated data, NMtrS max pseudo memory, arrogant bounding-box coordinate computation, <i>SAM TEL</i> only labels, and non-maximum suppression backend OOB.
MLP w/ OOB	75.6%	Used ECE calibrated data, KS input, weight box fusion bounding-box coordinate computation, and non-maximum weighting backend OOB.
OOB	75.1%	Used ECE calibrated data and non-maximum weighting OOB
PSC-LSTM	73.8%	Used NMtrS max pseudo memory, weight box fusion bounding-box coordinate computation, and <i>TEL</i> only confidences.
D2	71.5%	n/a
MLP	70.5%	Used NImgS bounding-box metrics and non-maximum weighting OOB.
Ycoco	70.9%	n/a
Three Detector Ensembling Results		
PCS-LSTM w/ OOB	83.6%	Used NMtrS max pseudo memory, arrogant bounding-box coordinate computation, and non-maximum weighting backend OOB
MLP w/ OOB	82.4%	Used KS bounding-box metrics, arrogant bounding-box coordinate computation, and non-maximum weighting backend OOB.
PCS-LSTM	81.1%	Used NImgS ratio pseudo memory and arrogant bounding-box coordinate computation.
OOB	80.1%	Used ECE calibrated data and weighted box fusion with ‘absent model aware average’ confidence.
MLP	79.6%	Used bounding-box coordinates and confidence vector as input and mean bonding-box computation.
Ycoco	78.6%	n/a
YxView	75.2%	n/a
D2	71.5%	n/a

§Definitions; ‘MLP CO’:MLP with confidence-only predictions, ‘Ycoco’:YOLOv5 w/ COCO pre-training, ‘YxView’:YOLOv5 w/ xView pre-training.

Appendix A

A.1 Sample Counts by Initial xView Objects with Assumed 0.3 meter GSD.

Overlapping One-vs-All Dataset Used for Training						
Input Chip Size		32	64	128	256	Total Features**
Min Feature Size		0	16	32	64	
Max Feature Size		32	64	128	inf	
Available Feature Counts		451412	276000	125908	25986	581953
After Thresholding and Reduction Counts		252525	135505	111945	24960	
Image Count Per Class						
(classes too small to be included for feature size are in red, using 25% cutoff)						
Parent Class	Label	32	64	128	256	Total Features**
<i>Building</i>	<i>Aircraft Hangar</i>	3	68	143	106	174
<i>Maritime Vessel</i>	<i>Barge</i>	22	93	125	75	168
<i>Building</i>	<i>Building</i>	187167*	239895*	111898*	20550	302226
<i>Passenger Vehicle</i>	<i>Bus</i>	6701	3136	109	0	6810
<i>Railway Vehicle</i>	<i>Cargo Container Car</i>	1584	1653	203	1	1787
<i>Truck</i>	<i>Cargo Truck</i>	5605	2138	216	2	5821
<i>Engineering Vehicle</i>	<i>Cement Mixer</i>	286	170	0	0	286
<i>Construction Site</i>	<i>Construction Site</i>	76	301	607	625	934
<i>Engineering Vehicle</i>	<i>Container Crane</i>	22	46	103	97	146
<i>Maritime Vessel</i>	<i>Container Ship</i>	1	36	174	222	258
<i>Engineering Vehicle</i>	<i>Crane Truck</i>	155	155	16	2	171
<i>Building</i>	<i>Damaged Building</i>	475	847	533	98	1021
<i>Engineering Vehicle</i>	<i>Dump Truck</i>	1330	768	9	0	1339
<i>Engineering Vehicle</i>	<i>Engineering Vehicle</i>	163	120	40	8	203
<i>Engineering Vehicle</i>	<i>Excavator</i>	809	638	15	0	824
<i>Building</i>	<i>Facility</i>	51	324	593	453	781

<i>Maritime Vessel</i>	<i>Ferry</i>	55	125	108	53	180
<i>Maritime Vessel</i>	<i>Fishing Vessel</i>	532	510	165	56	703
<i>Fixed Wing Aircraft</i>	<i>Fixed-wing Aircraft</i>	40	64	32	6	72
<i>Railway Vehicle</i>	<i>Flat Car</i>	79	121	44	1	123
<i>Engineering Vehicle</i>	<i>Front loader Bulldozer</i>	619	268	5	0	624
<i>Engineering Vehicle</i>	<i>Ground Grader</i>	82	52	1	0	83
<i>Engineering Vehicle</i>	<i>Haul Truck</i>	214	316	107	0	321
<i>Helicopter</i>	<i>Helicopter</i>	48	66	20	0	68
<i>Helipad</i>	<i>Helipad</i>	51	82	65	10	116
<i>Building</i>	<i>Hut Tent</i>	640	253	58	17	701
<i>Railway Vehicle</i>	<i>Locomotive</i>	27	109	87	4	114
<i>Maritime Vessel</i>	<i>Maritime Vessel</i>	260	278	325	197	620
<i>Engineering Vehicle</i>	<i>Mobile Crane</i>	106	220	189	76	306
<i>Maritime Vessel</i>	<i>Motorboat</i>	1392	554	45	0	1437
<i>Maritime Vessel</i>	<i>Oil Tanker</i>	0	0	21	59	59
<i>Railway Vehicle</i>	<i>Passenger Car</i>	152	1536	1388	3	1540
<i>Fixed Wing Aircraft</i>	<i>Passenger Cargo Plane</i>	35	271	547	349	620
<i>Passenger Vehicle</i>	<i>Passenger Vehicle</i>	2928	45	0	0	2928
<i>Truck</i>	<i>Pickup Truck</i>	1094	30	0	0	1094
<i>Pylon</i>	<i>Pylon</i>	100	328	244	14	345
<i>Railway Vehicle</i>	<i>Railway Vehicle</i>	12	14	5	1	17
<i>Engineering Vehicle</i>	<i>Reach Stacker</i>	59	65	9	0	68
<i>Maritime Vessel</i>	<i>Sailboat</i>	648	273	35	1	683
<i>Engineering Vehicle</i>	<i>Scraper Tractor</i>	77	49	0	0	77
<i>Building</i>	<i>Shed</i>	1012	531	138	87	1166
<i>Shipping Container</i>	<i>Shipping Container</i>	1493	998	55	2	1548
<i>Shipping Container Lot</i>	<i>Shipping container lot</i>	455	1369	1374	657	2055
<i>Fixed Wing Aircraft</i>	<i>Small Aircraft</i>	307	341	43	5	350
<i>Passenger Vehicle</i>	<i>Small Car</i>	209535*	1533	68	19	209606
<i>Storage Tank</i>	<i>Storage Tank</i>	940	909	645	238	1588
<i>Engineering Vehicle</i>	<i>Straddle Carrier</i>	10	22	42	32	54
<i>Railway Vehicle</i>	<i>Tank car</i>	109	112	9	0	118
<i>Tower</i>	<i>Tower</i>	46	66	37	6	84
<i>Engineering Vehicle</i>	<i>Tower crane</i>	14	46	96	88	135
<i>Truck</i>	<i>Trailer</i>	3791	2215	225	24	4023
<i>Truck</i>	<i>Truck</i>	11561	4509	425	0	11986
<i>Truck</i>	<i>Truck Tractor (TT)</i>	816	358	33	0	849
<i>Truck</i>	<i>TT w/ Box Trailer</i>	2278	3210	1249	1	3527
<i>Truck</i>	<i>TT w/ Flatbed Trailer</i>	691	713	190	0	881

<i>Truck</i>	<i>TT w/ Liquid Tank</i>	126	128	19	0	145
<i>Maritime Vessel</i>	<i>Tugboat</i>	40	169	166	37	206
<i>Truck</i>	<i>Utility Truck</i>	3571	328	0	0	3571
<i>Vehicle Lot</i>	<i>Vehicle Lot</i>	779	2080	2533	1630	3792
<i>Maritime Vessel</i>	<i>Yacht</i>	138	346	277	74	421

* truncated to 100,00 samples

** Total feature count is smaller than sum of chip size counts b/c of overlap used in selecting training data for a given chip size

A.2 DNN Dataset Augmentations

Descriptions of online augmentations use in training. Dataset multiplier in red.





Original 128x128 chip sample of <i>TEL group</i>	
Vertical flip example. Horizontal flip is similar along the y axis. Note, using both flips is the same as a 180° rotation. Multiplier of $2X$	
Image rotation example as -15° with black buffer. Used as $step(^\circ) \in [0^\circ, 360^\circ)$. Multiplier of $\text{ceiling}(step/360)X$	
Image jitter example. Jitter is the shifting of pixel values within the image about the image center (yellow dot). Two mode of jitter are allowed. Mode 4 shifts the center of the image exclusively up, down, left, and right (cyan dots), where mode 8 also allows for vertical and horizontal shifting together (magenta dots). A black buffer was used again. Multiplier of $5X$ for mode 4 and $9X$ for mode 8.	

Image contrast example. This image shows an contrast increase by a factor of $1.3X$. This has a multiplying factor of $(1 + n)X$ where n is the number of input values.



Image brightness example. This image shows an brightness increase by a factor of $1.3X$. This has a multiplying factor of $(1 + n)X$ where n is the number of input values.



A.3 Detailed Flow Chart for Clustering Counts

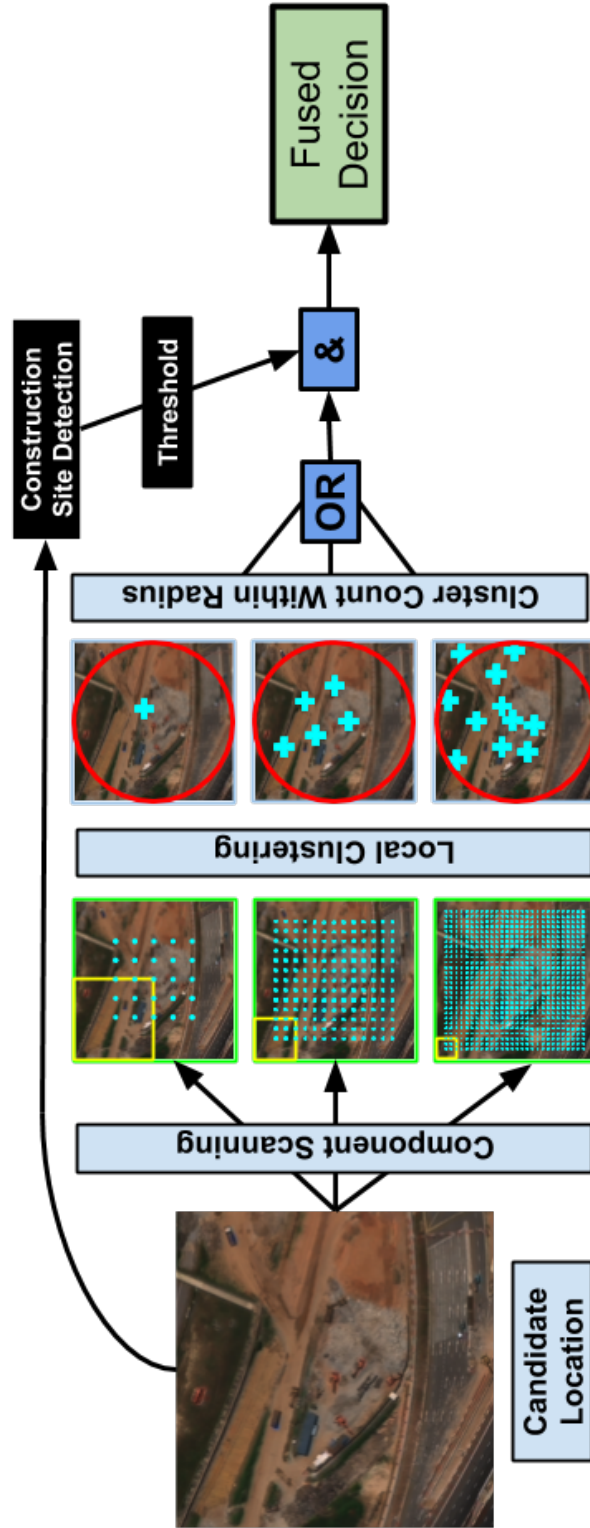


Fig. A.1: Detail of fusion method 4 described in Chapter 2 From left to right: 1) Passing candidate location through *Construction Site (CS)* detection model to obtain inference response. 2) Scan candidate *CS* samples with component DNN models of different sizes (e.g. multi-scale). 3) Local clustering followed by counting the number of cluster centers (cyan crosses) within feature radius (red circles). 4) Apply logic tree to α -cut to *CS* inference response and cluster counts. 5) Final binary fused decisions.

A.4 DeepNET Complete Object Counts for Experimental Selected Scenes

The table contains the breakdown of how classes were partitioned and used for the different experiments in Chapter 6. Some abbreviation have been made for that it would fit on the page: *SAM*=Surface-to-Air Missile, *SSM*=Surface-to-Surface Missile, *TEL*=Transporter Erector Launcher, *LP*=Launch Pad, *AAA*=Ant-Aircraft Artillery, *CSV*=Combat Support Vehicle, *CV*=Confuser Vehicle.

Class	37-Class	11-class	4-Class	TEL Dataset	LP Dataset	Instance Count
<i>SAM LP w/ Revetment</i>	<i>SAM LP w/ Revetment</i>	<i>SAM LP w/ Revetment</i>	<i>All LP</i>	Remove	<i>All LP</i>	35668
<i>SAM TEL with Canister</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	Remove	30123
<i>Revetment</i>	<i>Revetment</i>	Remove	Remove	Remove	Remove	29416
<i>Ground Motor Vehicle</i>	<i>Ground Motor Vehicle</i>	Remove	Remove	Remove	Remove	25527
<i>SAM Missile LP</i>	<i>SAM Missile All LP</i>	<i>SAM Missile All LP</i>	Remove	Remove	Remove	21715
<i>Truck</i>	<i>Truck</i>	Remove	Remove	Remove	Remove	12289
<i>SAM LP</i>	<i>SAM LP</i>	<i>SAM LP</i>	<i>All LP</i>	Remove	<i>All LP</i>	11246
<i>SAM TEL</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	Remove	9995
<i>SAM Launcher</i>	<i>SAM Launcher</i>	<i>SAM Launcher</i>	Remove	Remove	Remove	8811
<i>Multi Ramp Platform</i>	<i>Multi Ramp Platform</i>	Remove	Remove	Remove	Remove	4774
<i>Radial SAM Launcher Site</i>	<i>SAM Launcher Site</i>	Remove	Remove	Remove	Remove	4644
<i>Bunker</i>	<i>Bunker</i>	Remove	Remove	Remove	Remove	4421
<i>AAA Site</i>	<i>AAA Site</i>	Remove	Remove	Remove	Remove	3626
<i>SAM Launcher Site</i>	<i>SAM Launcher Site</i>	Remove	Remove	Remove	Remove	3131
<i>Transloader</i>	<i>Transloader</i>	<i>CSV</i>	Remove	Remove	Remove	3024
<i>Combat Ground Motor Vehicle</i>	<i>Combat Ground Motor Vehicle</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	3020
<i>Antenna</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	2770
<i>AAA</i>	<i>AAA</i>	<i>Weapon</i>	Remove	Remove	Remove	2734
<i>Automobile</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	2370
<i>Mobile Radar-Communications Vehicle</i>	<i>CSV</i>	<i>CSV</i>	<i>CSV</i>	Remove	Remove	2333
<i>Single Ramp Platform</i>	<i>Single Ramp Platform</i>	Remove	Remove	Remove	Remove	2088
<i>Semi Truck</i>	<i>Semi Truck</i>	Remove	Remove	Remove	Remove	2030
<i>LP w/ Revetment</i>	<i>LP w/ Revetment</i>	<i>LP w/ Revetment</i>	<i>All LP</i>	Remove	<i>All LP</i>	1386
<i>Ground Vehicle</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	1115
<i>Mound</i>	<i>Mound</i>	Remove	Remove	Remove	Remove	1041
<i>Field Artillery Platform</i>	<i>Field Artillery Platform</i>	<i>Weapon</i>	Remove	Remove	Remove	963
<i>Fixed SAM Launcher</i>	<i>SAM Launcher</i>	<i>SAM Launcher</i>	Remove	Remove	Remove	931
<i>TEL</i>	<i>Other SAM TEL</i>	<i>Other TEL</i>	<i>All TEL</i>	<i>All TEL</i>	Remove	850
<i>Ground Motor Passenger Vehicle</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	799
<i>Fighter Airplane</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	749
<i>Aircraft Hangar</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	748
<i>Radar Vehicle</i>	<i>CSV</i>	<i>CSV</i>	<i>CSV</i>	Remove	Remove	693
<i>Howitzer</i>	<i>Howitzer</i>	<i>Weapon</i>	Remove	Remove	Remove	687
<i>Parabolic Antenna</i>	<i>Parabolic Antenna</i>	Remove	Remove	Remove	Remove	686
<i>Flat Bed Truck</i>	<i>Flat Bed Truck</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	623
<i>Directional SAM Launcher Site</i>	<i>SAM Launcher Site</i>	Remove	Remove	Remove	Remove	555

<i>Tripod Mast</i>	<i>Tripod Mast</i>	Remove	Remove	Remove	Remove	529
<i>Tank</i>	<i>Combat Ground Motor Vehicle</i>	CV	CV	CV	Remove	412
<i>Emitter Site</i>	<i>Emitter Site</i>	Remove	Remove	Remove	Remove	394
<i>TEL with Canister</i>	<i>Other TEL</i>	<i>Other TEL</i>	<i>All TEL</i>	<i>All TEL</i>	Remove	358
<i>Anti-Aircraft Machine Gun</i>	<i>Weapon</i>	<i>Weapon</i>	Remove	Remove	Remove	286
<i>Prepared Position</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	272
<i>SSM TEL</i>	<i>Other SAM TEL</i>	<i>Other SAM TEL</i>	<i>SAM TEL</i>	<i>SAM TEL</i>	Remove	253
<i>Tower</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	243
<i>Emitter Structure</i>	<i>Emitter Structure</i>	Remove	Remove	Remove	Remove	240
<i>Aircraft</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	222
<i>Communications Vehicle</i>	<i>CSV</i>	<i>CSV</i>	<i>CSV</i>	Remove	Remove	218
<i>Dump Truck</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	212
<i>Mobile Projectile Launcher</i>	<i>Combat Ground Motor Vehicle</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	187
<i>Rectangular Antenna</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	181
<i>Projectile</i>	<i>Weapon</i>	<i>Weapon</i>	Remove	Remove	Remove	176
<i>Radome</i>	Remove	Remove	Remove	Remove	Remove	175
<i>Vehicle</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	174
<i>Cargo Automobile</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	171
<i>Missile Launcher</i>	<i>Weapon</i>	<i>Weapon</i>	Remove	Remove	Remove	167
<i>Field Artillery Site</i>	<i>Military Site</i>	Remove	Remove	Remove	Remove	160
<i>Projectile Launcher</i>	<i>Weapon</i>	<i>Weapon</i>	Remove	Remove	Remove	150
<i>Personal Motorized Vehicle</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	148
<i>Helicopter</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	141
<i>Projectile Launcher Site</i>	<i>Military Site</i>	Remove	Remove	Remove	Remove	141
<i>Mast</i>	Remove	Remove	Remove	Remove	Remove	140
<i>Pole</i>	Remove	Remove	Remove	Remove	Remove	138
<i>SSM TEL with Canister</i>	<i>Other TEL</i>	<i>Other TEL</i>	<i>All TEL</i>	<i>All TEL</i>	Remove	137
<i>Bus</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	133
<i>Circular Dish Antenna</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	132
<i>Bongo Truck</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	132
<i>Armored Personnel Carrier</i>	<i>Combat Ground Motor Vehicle</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	130
<i>Van</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	125
<i>Helipad</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	124
<i>Airplane</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	99
<i>Pickup Truck</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	95
<i>SSM LP</i>	<i>All LP</i>	<i>All LP</i>	<i>All LP</i>	Remove	<i>All LP</i>	92
<i>Tanker Truck</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	74
<i>SSM LP w/ Revetment</i>	<i>LP w/ Revetment</i>	<i>LP w/ Revetment</i>	<i>All LP</i>	Remove	<i>All LP</i>	72
<i>Air Traffic Control Tower</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	68
<i>Sport Utility Vehicle</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	62
<i>CSV</i>	<i>CSV</i>	<i>CSV</i>	<i>CSV</i>	Remove	Remove	58
<i>Multiple Rocket Launcher System</i>	<i>Weapon</i>	<i>Weapon</i>	Remove	Remove	Remove	57
<i>Bomber Airplane</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	57
<i>Mobile Construction Equipment</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	57
<i>Building</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	55
<i>Mortar</i>	<i>Weapon</i>	<i>Weapon</i>	Remove	Remove	Remove	54

<i>Infantry Fighting Vehicle</i>	<i>Combat Ground Motor Vehicle</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	54
<i>Installation Structure</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	47
<i>Array of Antennas</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	46
<i>Towed Multiple Rocket Launcher System</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	44
<i>Mobile Rocket Launcher</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	40
<i>Self-Propelled Howitzer</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	39
<i>Weapon</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	39
<i>Rectangular Array of Antennas</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	35
<i>Train Car</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	33
<i>Engineering Support Vehicle</i>	<i>CSV</i>	<i>CSV</i>	<i>CSV</i>	Remove	Remove	33
<i>Self-Propelled Artillery</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	29
<i>Self-Propelled AAA</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	24
<i>Self-Propelled Anti-Air Machine Gun</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	22
<i>Airliner</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	22
<i>Precision-Guided Missile</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	18
<i>Planar Phased Array</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	16
<i>All-Terrain Vehicle</i>	<i>Other Vehicle</i>	Remove	Remove	Remove	Remove	13
<i>Circular Array of Antennas</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	12
<i>Linear Array of Antennas</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	12
<i>Command and Control Aircraft</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	10
<i>Emergency Vehicle</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	10
<i>Propeller Bomber Aircraft</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	8
<i>Yagi Antenna</i>	<i>Antenna</i>	Remove	Remove	Remove	Remove	8
<i>Power Substation</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	8
<i>Weapon Site</i>	<i>Military Site</i>	Remove	Remove	Remove	Remove	8
<i>Cargo Airplane</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	7
<i>Aircraft Carrier</i>	<i>Military Site</i>	Remove	Remove	Remove	Remove	7
<i>Combatant Service Ship</i>	Remove	Remove	Remove	Remove	Remove	7
<i>Jet Bomber Aircraft</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	5
<i>Excavation Crane</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	5
<i>Rotary Wing Manned Aircraft</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	4
<i>Crane</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	4
<i>Combat Airplane</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	3
<i>Unmanned Aerial Vehicle</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	3
<i>Infrastructure</i>	<i>Infrastructure</i>	Remove	Remove	Remove	Remove	2
<i>Rail Transport Vehicle</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	2
<i>Rocket Launcher</i>	<i>Weapon</i>	Weapon	Remove	Remove	Remove	1
<i>Delta Wing Bomber</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	1
<i>Passenger Airplane</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	1

<i>Surveillance Aircraft</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	1
<i>Variable-Sweep Wing Bomber</i>	<i>Aircraft</i>	Remove	Remove	Remove	Remove	1
<i>Communications Ship</i>	Remove	Remove	Remove	Remove	Remove	1
<i>Locomotive</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	<i>CV</i>	Remove	1

Bibliography

- [1] G. J. Scott, K. C. Hagan, R. A. Marcum, J. A. Hurt, D. T. Anderson, and C. H. Davis, "Enhanced fusion of deep neural networks for classification of benchmark high-resolution image datasets," *IEEE Geoscience & Remote Sensing Letters*, Vol. 15, No. 9, pp. 1451-1455, 2018, DOI: 10.1109/LGRS.2018.2839092.
- [2] J. A. Hurt, G. J. Scott, D. T. Anderson, C. H. Davis, "Benchmark meta-dataset of high-resolution remote sensing imagery for training robust deep learning models in machine-assisted visual analytics," *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 9-11 October, 2018.
- [3] G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, and C.H. Davis (2017), "Training deep convolutional neural networks for land cover classification of high-resolution imagery," *IEEE Geoscience & Remote Sensing Letters*, Vol. 14, No. 4, pp. 549-553, DOI: 10.1109/LGRS.2017. 2657778.
- [4] G. J. Scott , R. A. Marcum, C. H. Davis and T. W. Nivin, "Fusion of deep convolutional neural networks for land cover classification of high-resolution imagery," *IEEE Geoscience & Remote Sensing Letters*, Vol. 14, No. 9, 2017, pp. 1638-1642, DOI: 10.1109/LGRS. 2017.2722988.
- [5] Y. Yang and S. Newsam, "Bag-of-visual words and spatial extensions for land-use classification," *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2010, pp. 270-279.
- [6] G. Sheng, W. Yang, T. Xu, and H. Sun, "High-resolution satellite scene classification using a sparse coding based multiple feature combination," *International Journal of Remote Sensing*, Vol. 33, No. 8, 2012, pp. 2395-2412.
- [7] G. Cheng, J. Han, and X. Lu., "Remote sensing image scene classification: benchmark and state of the art," *Proceedings of the IEEE*, Vol. 105, No. 10, 2017, pp. 1865-1883, DOI: 10.1109/JPROC.2017.2675998.
- [8] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [9] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580-587.
- [10] R. Girshick, "Fast R-CNN," *The 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440-1448, DOI: 10.1109/ICCV.2015.169
- [11] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, 2017, pp. 1137-1149, DOI: 10.1109/TPAMI.2016.2577031.
- [12] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv*, 2018, arXiv:1804.02767.
- [13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A.C. Berg, "SSD: Single Shot MultiBox Detector," *The 14th European Conference on Computer Vision (ECCV2016)*, vol. 9905, 2016, pp. 21-37.
- [14] J. Shermeyer and A. V. Etten, "The Effects of Super-Resolution on Object Detection Performance in Satellite Imagery," *EarthVision 2019,IEEE*, 2019.
- [15] Y. Koga, H. Miyazaki, and R. Shibasaki, "A Method for Vehicle Detection in High-Resolution Satellite Images that Uses a Region-Based Object Detector and Unsupervised Domain Adaptation," *Remote Sensing 2020*, <https://doi.org/10.3390/rs12030575>.
- [16] Z. Xin, H. Liangxiu, H. Lianghao, and Z. Liang, "How Well Do Deep Learning-Based Methods for Land Cover Classification and Object Detection Perform on High Resolution Remote Sensing Imagery?" *Remote Sensing 2020*, 2020, <https://doi.org/10.3390/rs12030417>.

- [17] S. P. DelMarco, V. Tom, H. Webb, W. Snyder, C. Jarvis, D. Fay, "Shape-based ATR for wide-area processing of satellite imagery," *SPIE 10988, Automatic Target Recognition XXIX*, 2019, <https://doi.org/10.1117/12.2518185>.
- [18] Y. Yanan, L. Zezhong, R. Bohao, C. Jingyi, L. Sudi, and L. Fang, "Broad Area Target Search System for Ship Detection via Deep Convolutional Neural Network," *Remote Sensing 2019*, 2019, <https://doi.org/10.3390/rs11171965>.
- [19] R. A. Marcum, C. H. Davis, G. J. Scott, and T. W. Nivin, "Rapid broad area search and detection of Chinese surface-to-air missile sites using deep convolutional neural networks," *Journal of Applied Remote Sensing*, Vol. 11, No. 4, 042614, 2017, DOI: 10.1117/1.JRS.11.042614.
- [20] Cannaday, A.B., C.H. Davis, and G.J. Scott (2019), "Improved search and detection of surface-to-air missile sites using spatial fusion of component object detections from deep neural networks," *Proceedings of International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, July 28 - August 2, 2019.
- [21] A. B. Cannaday II, R. L. Chastain, J. A. Hurt, C. H. Davis, G. J. Scott and A. J. Maltenfort, "Decision-Level Fusion of DNN Outputs for Improving Feature Detection Performance on Large-Scale Remote Sensing Image Datasets," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 5428-5436, DOI: 10.1109/BigData47090.2019.9006502.
- [22] A. B. Cannaday, C. H. Davis and A. J. Maltenfort, "Evaluation of Fuzzy Integral Data Fusion Methods for Rare Object Detection in High-Resolution Satellite Imagery," *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 1-10, doi: 10.1109/BigData50022.2020.9377990.
- [23] A. Neubeck and L. Van Gool, "Efficient Non-Maximum Suppression," *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, pp. 850-855, doi: 10.1109/ICPR.2006.479.
- [24] N. Bodla, B. Singh, R. Chellappa and L. S. Davis, "Soft-NMS — Improving Object Detection with One Line of Code," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5562-5570, doi: 10.1109/ICCV.2017.593.
- [25] C. Ning, H. Zhou, Y. Song, and J. Tang. "Inception single shot MultiBox detector for object detection." In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, jul 2017.
- [26] H.Zhou, Z. Li, C. Ning, and J. Tang. "Cad: Scale invariant framework for real-time object detection." In *Proceedings of the IEEE International Conference on Computer Vision*, pages 760–768, 2017.
- [27] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models. Image and Vision Computing." *arXiv preprint*, arXiv:1910.13302 . 2021
- [28] A. B. Cannaday II, C. H. Davis, G. J. Scott, B. Ruprecht and D. T. Anderson, "Broad Area Search and Detection of Surface-to-Air Missile Sites Using Spatial Fusion of Component Object Detections From Deep Neural Networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 13, pp. 4728-4737, 2020, DOI: 10.1109/JSTARS.2020.3015662.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pg. 770 –778, 2016, DOI: 10.1109/CVPR.2016.90.
- [30] B. Zoph, V. Vasudevan, J. Shlens, and Q. Le, "Learning Transferable Architectures for Scalable Image Recognition," *CVPR 2018*, pg. 8697-8710, DOI: 10.1109/CVPR.2018.00907.
- [31] T. Hastie, R. Tibshirani, J. Friedman, (2009) "Elements of Statistical Learning: data mining, inference, and prediction," 2nd Edition, *web.stanford.edu*. Retrieved 2019-04-04.
- [32] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pg. 1800-1807, doi: 10.1109/CVPR.2017.195.
- [33] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," *International Conference on Learning Representations*, 2019, [Online]. Available: <https://openreview.net/forum?id=Hy1VB3AqYm>.
- [34] M.Tan, Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *International Conference on Machine Learning*, 2019.
- [35] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *2009 conference on Computer Vision and Pattern Recognition*, 2009, DOI: 10.1109/CVPR.2009.5206848.

- [36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," Dec. 2014, arXiv:1412.6980.
- [37] N. Ye, K. M. A. Chai, W. S. Lee, H. L. Chieu "Optimizing F-measures: a tale of two approaches," *Proceedings of the International Conference on Machine Learning*, 2012.
- [38] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," *InSIGIR*, 1995, pp. 246–254.
- [39] J. Jang, "ANFIS Adaptive-Network-based Fuzzy Inference System. Systems, Man and Cybernetics", *IEEE Transactions on*. 23. 1993, pp. 665 - 685, 10.1109/21.256541.
- [40] A. Abraham, "Adaptation of Fuzzy Inference System Using Neural Learning," in Nedjah, Nadia; de Macedo Mourelle, Luiza (eds.), *Fuzzy Systems Engineering: Theory and Practice, Studies in Fuzziness and Soft Computing, 181, Germany: Springer Verlag*, 2005, pp. 53–83.
- [41] D. Karaboga, and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey," *Artificial Intelligence Review*, 2018, DOI: 10.1007/s10462-017-9610-2.
- [42] B. Ruprecht, C. Veal, B. Murray, M. Islam, D. Anderson, F. Petry, J. Keller, G. Scott, and C. Davis, "Fuzzy logic-based fusion of deep learners in remote sensing," *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019.
- [43] B. Ruprecht, C. Veal, A. Cannaday, D. Anderson, F. Petry, J. Keller, G. Scott, C. Davis, C. Northworthy, K. Nock, and E. Glimour, "Are neural fuzzy logic systems really explainable and interpretable?," *SPIE Security and Defense*, 2020.
- [44] B. Ruprecht, C. Veal, B. Murray, M. Islam, D. Anderson, F. Petry, J. Keller, G. Scott, and C. Davis, "Possibilistic Clustering Enabled Neuro Fuzzy Logic," under review, *World Congress on Computational Intelligence*, 2020.
- [45] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, B. McCord, (2018), "xView: Objects in Context in Overhead Imagery," *arXiv preprint*, arXiv:1802.07856v1.
- [46] A. B. Cannaday, C. H. Davis, and G. J. Scott (2018), "Rapid search and detection of Chinese SAM sites with Deep Neural Networks (DNNs) using component-based detections," USGIF Tech Showcase West, NGA Campus West, 17 October 2018, St. Louis, MO. Note: USGIF poster-presentation available on request.
- [47] M. Sugeno, "Fuzzy Measures and Fuzzy Integrals: A survey," *Fuzzy Automata and Decision Process*. Amsterdam: North Holland. 1977
- [48] M. Tskeno and T. Terano (1977), "A model of learning based on fuzzy information" , *Kybernetes*, Vol. 6, No. 3, pp. 157-166, 1977, <https://doi.org/10.1108/eb005448>
- [49] G. Choquet, "Theory of capacities," *Annales de l'Institut Fourier*, Tome 5, pp. 131-295, 1954, DOI: 10.5802/aif.53.
- [50] M. Islam, D. T. Anderson, A. J. Pinar, T. C. Havens, G. Scott and J. M. Keller, "Enabling Explainable Fusion in Deep Learning With Fuzzy Integral Neural Networks," *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 7, pp. 1291-1300, July 2020, DOI: 10.1109/TFUZZ.2019.2917124.
- [51] E. Mace, K. Manville, M. Barbu-McInnis, M. Laielli, M. Klaric, S. Dooley "Overhead Detection: Beyond 8-bits and RGB," *arXiv preprint*, arXiv:1808.02443.
- [52] G. Jocher, K. Nishimura, T. Mineeva, R. Vilariño: YOLOv5 (2020). <https://github.com/ultralytics/yolov5>. DOI: 10.5281/zenodo.4679653
- [53] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*. <https://github.com/facebookresearch/detectron2>, 2019.
- [54] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. "Microsoft COCO: Common objects in context." In *ECCV*. arXiv:1405.0312. 2014.
- [55] T. Nivin, (2018) "Deepnet : An Extensible Data Acquisition and Curation Framework Supporting Computer Vision Deep Learning Research"[Master's Thesis, University of Missouri], <https://mospace.umsystem.edu/xmlui/handle/10355/67622>
- [56] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

- [57] A. Bochkovskiy, C.-Y. Wang and H.-Y. Liao, "Yolov4: Optimal speed and accuracy of object detection", arXiv:2004.10934, 2020.
- [58] "ZFTurbo: Keras-RetinaNet-for-Open-Images-Challenge-2018." <https://github.com/zfturbo/keras-retinanet-for-open-images-challenge-2018>
- [59] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [60] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2901–2907.

VITA



Alan Bruice Cannaday II was born in 1984 to Alan and Francis Cannaday (né Barrett) in San Diego, California. He graduated in the class of 2002 from Taylorsville High School in Taylorsville, Utah and received a Leadership Scholarship to attend the University of Utah. Alan took a two-year leave of absence to serve as a full-time missionary in the Philippines for The Church of Jesus Christ of Latter-day Saints. Upon his return, he graduated in 2010 with a Bachelor of Science in Mathematics with an emphasis in Scientific Computing and a minor in Physics. Continuing at the University of Utah, Alan completed a Master of Science in Computational Engineering and Science in 2012. He then worked 5 for years

as a Research Engineer for FamilySearch where he focused on historical document analysis and automatic content extraction. In 2018 he began pursuing a Doctorate in Philosophy in Computer Science at the University of Missouri in Columbia, Missouri, which he completed in 2021. This was accomplished under the direction of his advisor, Dr. Curt Davis, with whom he worked as a Graduate Research Assistant as part of the Center for Geospatial Intelligence with studies and papers focused in computer vision, neural networks, object detection, fuzzy data fusion, and remote sensing. As of 2021, Alan has been married to Brooke for 15 years and the couple have seven children.