# Transfer Learning of Deep Neural Networks for Visual Collaborative Maritime Asset Identification

Nicholas Warren
*Texas State University*
San Marcos, TX, USA
ndw30@txstate.edu

Benjamin Garrard
*Texas State University*
San Marcos, TX, USA
bag131@txstate.edu

Elliot Staudt
*Mayachitra, Inc.*
Santa Barbara, CA, USA
staudt@mayachitra.com

Jelena Tešić
*Texas State University*
San Marcos, TX, USA
jtesic@txstate.edu

***Abstract*—Recent advances in deep learning for visual recognition demonstrate high performing pipeline for building and deploying well-performing content models. These advances come with underlying assumptions of the data characteristics pertaining to consumer image and video and availability of the large set of annotated data. In this paper we show how to apply lessons learned in the consumer domain to overhead maritime video corpora. We present how to successfully tune deep learning network to overhead maritime domain and tune parameters to new domain characteristics to achieve high performance metric with smaller set of domain annotations. This approach improves the state-of-the-art metric by 80% on maritime IPATCH data [1]. Next, we present challenges and propose several approaches on user collaboration for maritime asset identification, and introduce the notion of persistent and intermittent models.**

***Keywords*-Training, Collaboration, Data models, Multi-layer neural network, Machine learning, Computational modeling Machine Vision, Data Science**

## I. Introduction

Machine Learning algorithms will always do exactly what one instructs them to: they will learn by example, and the models they produce will be an exact product of the training examples that you provided. Complex Machine Learning algorithms, such as Deep Neural Networks (DNN), require a significant number of labeled training samples to perform well. Recent advances have shown that DNN systems perform with a superb degree of accuracy in visual object detection and recognition benchmarks, having been trained on millions of training examples [2]–[4]. The effectiveness of the Deep Convolutional Neural Networks (DCNNs) has been demonstrated for various computer vision tasks such as image classification, object detection, semantic-segmentation, human body joint localization, face recognition and so on [5]. Amassing and labeling massive amounts of data for each of these applications in a single domain has led to a set of breakthrough benchmarks e.g. Pascal VOC [6], ImageNet [7],and COCO [8]. These benchmarks were complex and expensive in nature, requiring overwhelming amount of human time and resources, while focusing on consumer imagery. state-of-the-art for DCNN systems is based on consumer visual data e.g. all systems and benchmarks focused on imagery created by consumers using their hand-held devices. When considering domain translation for other applications, one has to consider the replication process of similar scale. In most of the domains that come to mind - news, agriculture, archived cultural data, climate science, medical science, astronomy, space, underwater exploration, aerial imagery, satellite imagery, underwater imagery, drone-captured imagery - there exist no crowd sourcing effort or labeling uniformity to achieve comparable benchmark at such a large scale.

This work has been supported by NAVAIR SBIR N68335-18-C-0199 award.

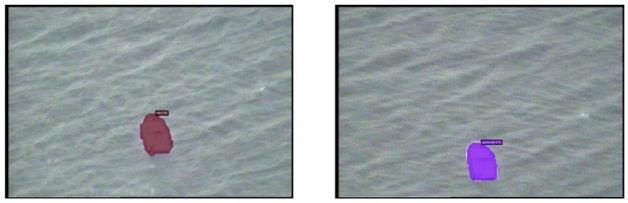

Fig. 1. State-of-the-art in Deep Convolutional Neural Networks (DCNN) is consumer data: slight change in operating conditions (overhead camera, cloudy day, low resolution) causes the system to identify boat as sports ball in one frame and kite in the other.

In this paper we consider maritime domain application where there is no maturity of annotated corpus or clear definition of semantically relevant objects. We utilize *transfer learning* approach, a machine learning method designed to mitigate the lack of training data. We start with reusing an existing pretrained model from consumer domain as the starting point for a model on a second task [5], [9]. Vast compute, time, and data labeling resources are required to develop neural network model from scratch. Since we do not have this type of data labels available in maritime datasets. we will investigate the success of transfer learning from consumer domain, where models were trained on images captured by hand held devices in social or land-bound settings, to maritime domain, where videos are taken by overhead camera mounted on an unmanned airborne system or a large ship.

*Collaborative tagging* describes the process by which many users add labels in the free-form to shared or related content [10]. Collaboration among diverse groups have been analyzed to show the need for simplified information sharing. Here, we propose the foundations for collaborative approach to sharing pretrained models based on consumer datasets to non-consumer domains, and tools that enable users to adapt those

models to their own application within the target domain, as illustrated with the proof of concept in maritime domain.

## II. Related Work

Recent DCNNs [2], [4], [11]–[15] have raised the bar with respect to what is expected of an image and video achievements in regards to machine learning. However, DCNNs continue to exhibit short-comings which has spurred great activity in the research community, but limited its effectiveness in real-life situations. Due to the large number of network parameters that need to be trained, DCNNs require a significant number of training samples. For tasks where a sufficient number of training samples is not available, a DCNN trained on a large dataset for a different task is tuned to the current task by making necessary modifications to the network and retraining it with the available data [5], [16]–[19]. Deep convolutional neural networks for overhead imagery have been used to produce pixel-wise classification maps of satellite imagery [20]. Authors compensate for imperfect training data through a two-step training approach: CNNs are first initialized by using a large amount of possibly inaccurate reference data, and then refined on a small amount of accurately labeled data to provide fine-grained classification maps [20]. One shot learning is another name for learning with few labels [21]. Lately, multiple groups proposed an one shot learning approach for deep learning setup, and showed it to be consistent with normal methods for training deep networks on large data [9].

## III. Domain Translation Considerations

*a) Domain Analysis:* Low resolution quality of operational data, size of objects of interest, view occlusions, and crowded scenes degrade the performance of state-of-the-art DCNN when applied to overhead sensor and shipboard data. For Maritime datasets, the best algorithms struggle with objects that are small (distant objects) or with the distorted view (sun glare), which are common problems in ocean environments. Humans have no issues in recognizing objects in videos with similar conditions, but state-of-the-art machine learning algorithms break when there is a slight change in the operational environment. Figure 1 illustrates how the state-of-the-art DCNN model [15] trained on consumer data classifies same boat into Sports Ball and Kite categories in 2 different frames.

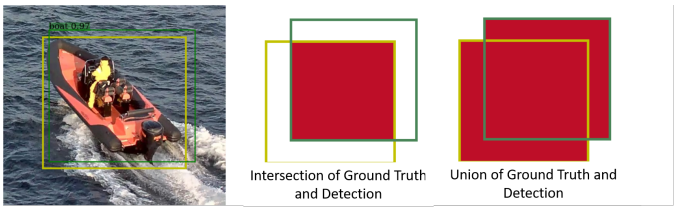


Fig. 2. IoU illustration for object localization: yellow boxes are ground truth, green boxes are detections.

*b) Deep Neural Network Learning Considerations:* Deep neural networks trained on large corpora of labeled consumer images provide a robust generalized modeling start, and initializing a network with transferred features from almost any number of layers produces a boost to generalization [22]. In our work, we rely on this finding and expand from a consumer dataset using domain translation to non-consumer datasets with our proof of concept in the maritime domain.

*c) Evaluation Metric:* Intersection over Union (IoU) is an evaluation metric typically used to measure the accuracy of object localization in an image, and it has been used in object recognition benchmarks such as PASCAL VOC [6], ImageNet [7], and COCO [8] benchmark. Any algorithm that provides predicted bounding boxes as output can be evaluated using IoU. IoU takes the set A of proposed object pixels within the proposed bounding box by the detector and the set of true object pixels B and calculates: $IoU(A, B) = A \cap B \div A \cup B$ as seen in Fig. 2. In consumer benchmarks the detector performance is a hit if IoU of proposed detection A and ground truth B is larger than a threshold, typically 0.5 e.g. if IoU $> 0.5$ it is a hit, otherwise it was a fail. In the Section VI we evaluate the performance of detectors using different measures of IoU, and evaluate performance sensitivity for maritime domain.

*d) Performance Evaluation:* We adopt COCO benchmark evaluation metric [8]. We calculate the True Positive $TP(c)$ for class $c$ as a proposal was made for class $c$ with probability higher than the threshold, and there actually was an object of class $c$, and the IOU is larger than set threshold. We calculate False Positive $FP(c)$ for class $c$ as : a proposal was made for class $c$, but there is no ground truth object of class $c$. False Negative $FN(c)$ for class $c$ as : a proposal was made for class $c$, but it is lower than the threshold; or IoU with the ground truth object for class $c$ is lower than than IoU threshold.

Thus, the average precision (AP) for set IoU as $AP(c) = \frac{|TP(c)|}{(|TP(c)|+|FP(c)|}$. Thus, the Recall for set IoU as $Recall(c) = \frac{|TP(c)|}{(|TP(c)|+|FN(c)|}$. The mAP (mean average precision) is computed over all classes. $mAP = \frac{1}{|classes|} \Sigma classes \frac{|TP(c)|}{|TP(c)|+|FP(c)|}$ for specific value of IoU and threshold.

## IV. Towards Collaborative Asset Identification

Modeling a dynamic domain, as discussed in the maritime environment, is challenging. Consider the maritime piracy monitoring scenario: there are multiple camera sensor feeds, and multiple analysts are accessing all these feeds from different location and for different applications e.g. monitoring, prevention, and alert. A Deep Neural Network framework is inherently static, as described in Section III. Training is mostly done in one location offline and the model is utilized for mass consummation e.g. image-to-text, identify consumer object in cell phone images or recognize a face. High confidence of the trained model is ensured by a high number of training data and context filtering.

Deep Convolutional Neural Networks for surveillance and monitoring needs to be utilized in a more dynamic environ-

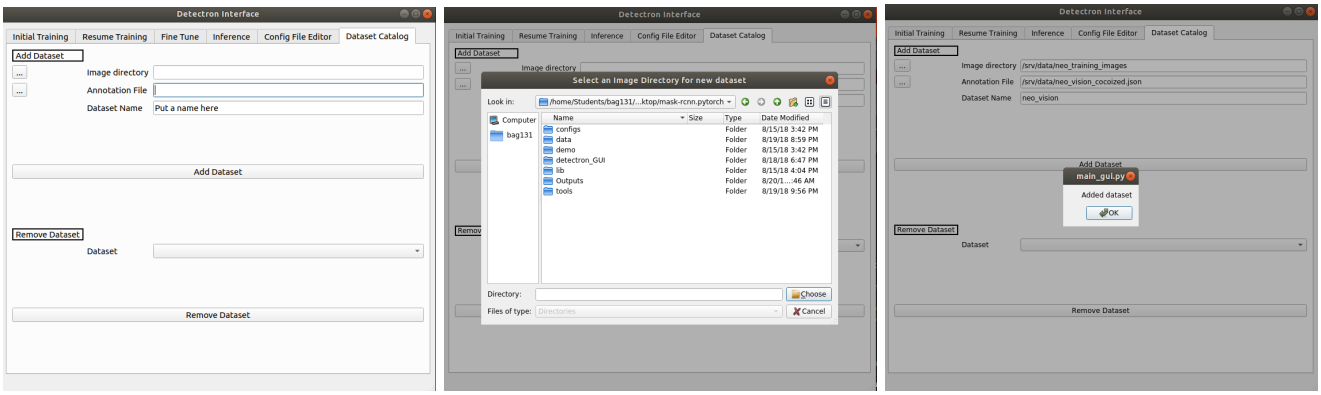


Fig. 3. Collaborative Training and Inference Tool (CTIT): The users add new domain-relevant datasets to the catalog used to identify data-sets on a computer (left). This is can be done with directory browsing (center) or using a path input (right).

ment: the sensor feeds have greater variance than consumer images. There is less labeled data available and the application of machine learning models for asset localization and identification varies due to the different surveillance goals. Users may require localizing, identifying, monitoring assets, or generating alerts. We identify the need to develop collaborative tools aimed specifically at these unstructured domains. Especially where collaboration across different agencies and seamless sharing of models (without sharing data) can improve different maritime missions. We have developed two interactive tools for analyst to foster this collaboration. The first tool, Collaborative Training and Inference Tool (CTIT), enables the building of more persistent models that can be shared among collaborators. The second, Asset Identification and Monitoring (AIM) Tool, enables persistent target labeling and intermittent modeling for real-time asset monitoring.

### A. *Collaborative Model Refinement in Dynamic Scenarios*

We have developed the Collaborative Training and Inference Tool (CTIT) to aid the training and fine-tuning of Detectron [24] models using domain-specific datasets, as well as inferencing those models on validation sets. The tool started out as a simplified interface to Detectron to allow more effective training. The tool aids in effective training by allowing analysts of all skill levels to be able to interact with Detectron without the need for in depth knowledge of the Detectron system. Screen shots of the CTIT Tool are shown in Figures 3,5, 6, and 7.

The Collaborative Training and Inference Tool (CTIT), shown in Figure 3, is an easy to use interface for the domain-specific training of the Detectron deep learning algorithms. The analyst can navigate to their data locations on the system using a directory browser and the CTIT will insert the data to a data-set catalog. Advanced functionality of the CTIT allows the analyst to edit configuration files used by the system using ”Config File Editor” functionality, as illustrated in Figure 5. The tools are designed to aid novice users in configuring the deep learning training module as it allows selection on all valid and system supported options. We are currently working on extending this tool to provide more information about the meaning of the parameters and how that can influence the training process. We plan to incorporate the findings from the Section VI to aid the analyst in training the domain specific module.

Collaborative Training and Interface Tool (CTIT) allows analyst to seamlessly access, improve, and fine tune existing models. Our goal is to enable seamless hand off of models between collaborators. For example, one analyst may access a specific maritime data (either new data or archived) and label several maritime vehicles as different or identify imagery that was captured under varying conditions (e.g. snowstorm, high glare, night). The data then can be added to the existing best-to-date model, fine tuned, re-evaluated on data, and leaves it in the repository for the next analyst to use it in the same manner. By the previous example this allows users with different domain expertise and skills to seamlessly reuse models. The CTIT enables analyst to select one of the three options for training the model: (1) Start training the model from scratch, (2) resume training an existing model from a checkpoint, and (3) fine-tune the checkpoint to newly added dataset, where each option initiates the call to different sub-process of the deep learning system. All three options are illustrated in Figure 6, as well as how the tool allows for configuration files to be changed. In the design, we have anticipated the wide variation in the dataset size, available memory across the workstations and servers, and this option allows analyst to adjust the batch size, iteration size, number of loading workers for a specific training task. With these parameters easily changeable any dataset size should be usable as long as the analyst sets the parameters accordingly.

We have extended the Collaborative Training and Inference Tool (CTIT) to support the experimentation and seamless inferencing of models on the validation dataset or new dataset, as shown in Figure 7. Analysts get to select where to store the prediction(s) (modeling results), if to use GPU or CPU (”Don’t Use Cuda” checkbox). If there is ground truth available for the validation dataset, the ”Reference JSON” option is used to generate evaluation scores from that JSON file. This easy to use interface makes the evaluation of the analyst’s current

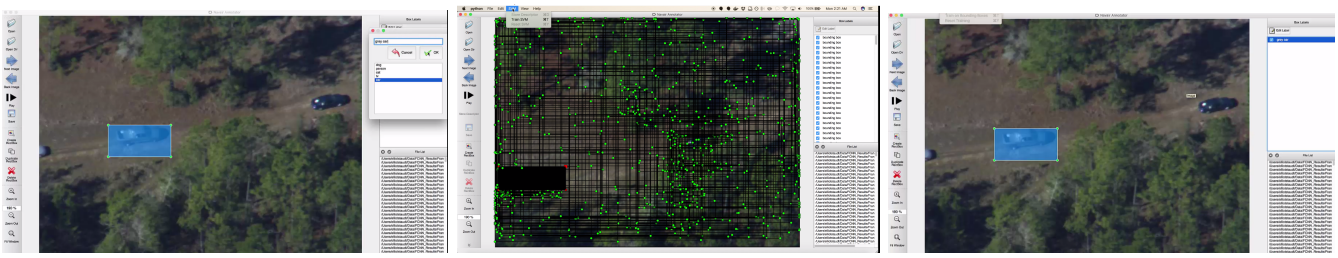

Fig. 4. AIM for Intermittent Modeling: Analyst can add new annotation and draw a new bounding box (left). The annotation will automatically label the region that has the with highest IoU score with that bounding box (center). Analyst can choose to refine the label or make it more specific (right). [23].

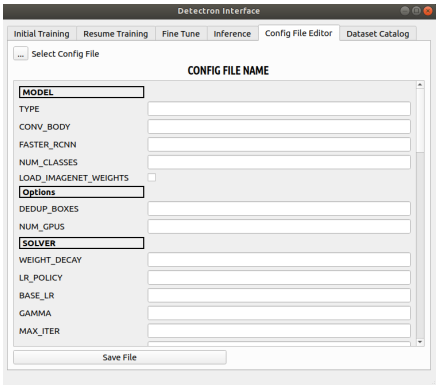


Fig. 5. The Collaborative Training and Inference Tool (CTIT) model training configuration setup: Analyst are able to seamlessly review all options for configuration parameters and select the valid combination. This collaborative feature greatly reduces error and allows novice users to access training deep learning module with a predefined set of options.

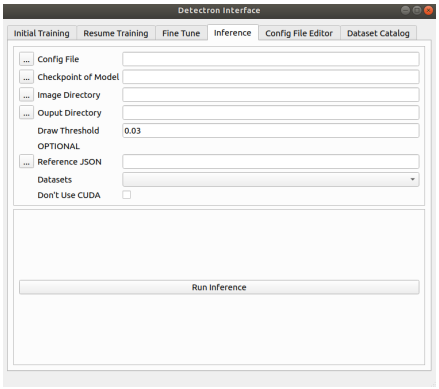


Fig. 6. The Collaborative Training and Inference Tool (CTIT) inference on validation set: ease of parameter setup allows for fast evaluation cycle of the trained models on the target domain e.g. location of the validation set, model threshold, and what model is evaluated. GPU/CPU toggling is also allowed with the "Don't Use Cuda" checkbox and it provides additional flexibility for the user in dynamic setup.

model quicker and easily located.

We have refined the tool by using it in collaborating on this project, and have utilized the tool for the experiments described in Section VI. Near term plan is to release it on git hub for research purposes and gather feedback on the useful features in collaborative model training setup.

The CTIT is to be released as open source software under the LGPL license on Github. The CTIT will be updated with new functionality when the new functionality completes and passes functionality checks. The CTIT will eventually support other deep learning models that coincide with research needs. In other words, completing a new portion of the interface to ease the creation custom model architectures by using Detectron and our CTIT. Video tutorials showing the use of the CTIT will be linked on the Github repository.

### *B. Asset Identification and Monitoring (AIM)*

*a) Asset Identification and Monitoring (AIM) Tool for Annotation:* provides analysts with an interface to (1) identify new objects of interests in maritime video feeds, (2) initialize light-weigh model training when enough samples were labeled, and (3) apply this intermittent model to the incoming video stream for real-time monitoring application. An analyst spots an asset of interest in a frame, and using the AIM annotation functionality, then localizes and annotates the asset, as illustrated in Figure 8. Full demo of the tool is available on YouTube [25].

*b) Asset Identification and Monitoring (AIM) Tool for Intermittent Modeling:* extends this basic features to to support building intermittent models: asset models persistent to particular sensors, time frame or location. Here we utilize the discriminate power of raw features produced by deep neural network system before the classification step, as demonstrated in [16], [17], [19]. The final form of the features cannot capture aspects that separate one member of a generic class (e.g. car) from another. If an analyst is looking for a specific kind of car, as illustrated in Figure 4(right), and labels it as a grey car, we can use this new specialized labeling to separate what characterizes grey car from all the other cars using underlying DCNN features. In the deep neural network inference phase, we save top region proposal network candidates [4], [12], [15] and illustrated in Figure 4(center), and associated high dimensional features for those regions. When an analyst uses AIM to select the bound box, the system automatically snaps to the closest detected region by DNN as illustrated in YouTube video [23]. Region with the highest IoU from analyst selected regions is selected as the best candidate. When an analyst labels the box, either adds missing label Figure 4(left) or adds more descriptive label Figure 4(right), the system saves the annotation to be used both for persistent and intermittent training. The flow when

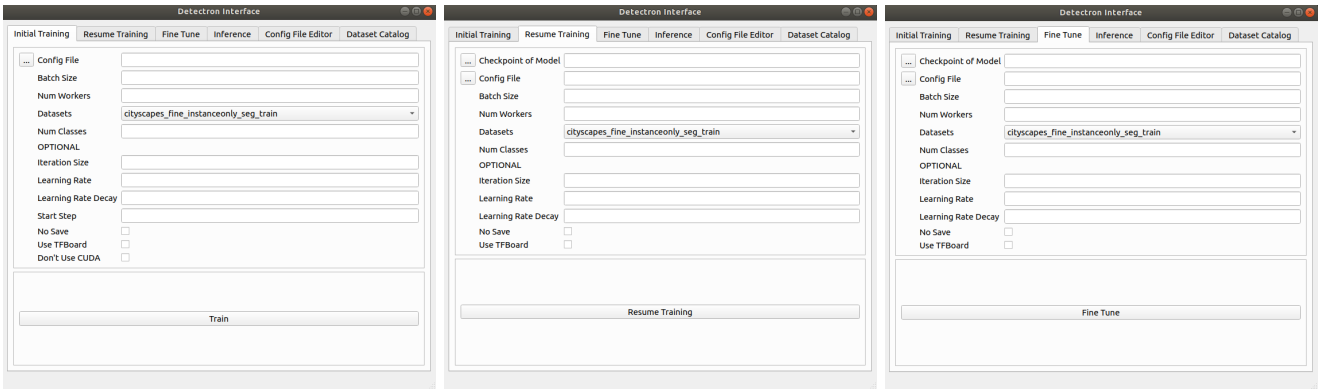


Fig. 7. The Collaborative Training and Inference Tool (CTIT) for (left) initial training, (center) resume training, and (right) fine tuning mode of deep learning model training process. This capability makes it easy for the analyst to make the last minute changes to their configurations e.g. number of loading workers, image batch size, and learning rate.

| dataset name | including data from COCO | including data from IPATCH | number of images/frames | number of annotations | number of annotated boats (IPATCH) |
|---|---|---|---|---|---|
| COCO Train | 2017 Training Set | - | 118287 | 860001 | **10759** |
| COCO Test | 2017 Testing Set | - | 5000 | 36781 | 430 |
| Control Set | COCO Train | 2016 PETS Low Level | 121868 | 8863 | **17064**(6305) |
| Validation Set | - | 2016 PETS Mid Level | 7049 | 8151 | **8151**(8151) |

TABLE I
COCO AND IPATCH DATA USED IN THE EXPERIMENT

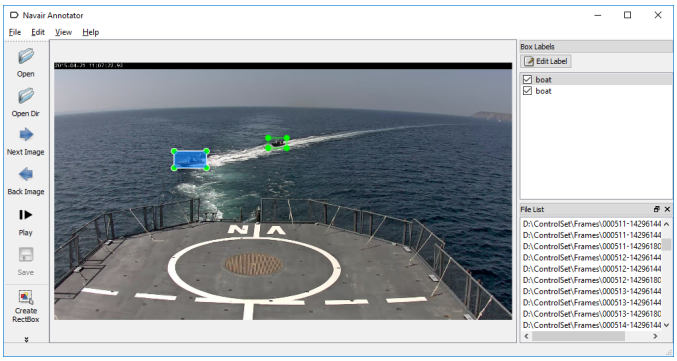


Fig. 8. AIM for Data Annotation: existing annotations and deep learning results are displayed (bounding box and a label). Analyst can add new label, and select or refine the label. The information is automatically saved to feed into deep learning processing.

using the AIM tool for intermittent modeling is shown at Figure 10. We are currently experimenting with single layer neural networks and traditional machine learning algorithms to create lightweight robust framework that generates intermittent models. The proposed system helps an analyst to identify and mark assets of interest and to utilize the existing persistent model in efficient way.

## V. EXPERIMENTAL SETUP

*a) Baseline Consumer Dataset:* We use COCO benchmark [8] for performance evaluation of our transfer learning strategy [8]. COCO, Common Objects in Context dataset consists of images with complex everyday scenes containing common objects in their natural context. COCO dataset contains 91 objects types common in consumer photography, and total of 2.5 million labeled objects in 328k images. We use a model trained with COCO only as a baseline analysis and metric to report the performance of our persistent models [8].

*b) Baseline Maritime Dataset:* IPATCH is a maritime dataset [26] collected in April 2015, addressing the application of multi sensor surveillance to protect a vessel at sea from piracy. The recordings represent a series of realistic maritime piracy scenarios. For the close range detection of threats, the IPATCH project added visual and thermal cameras to the VN Partisan vessel. Specifically, Four AXIS P1427-E Network cameras with five megapixel resolution were added; three of them at the starboard side and one facing the stern. They have day and night functionality, wide temperature range, weatherproofing, progressive scan CMOS, a frame rate of 30 fps, and digital PTZ [27]. We use the [27] Low Level Challenge Dataset as domain specific dataset added to the training pipeline. Sample annotation is shown in image 8. Mid and High Level Challenge Datasets from PETS Workshop [27] are annotated and used as the Validation Set to evaluate domain translation of persistent models. Note that all three PETS Datasets contain scenes from different days and activities, and we use them as robust proof of concept to demonstrate sensor data variation in domain translation for maritime application. Number of annotation instances and dataset characteristics are shown in Table IV-A.

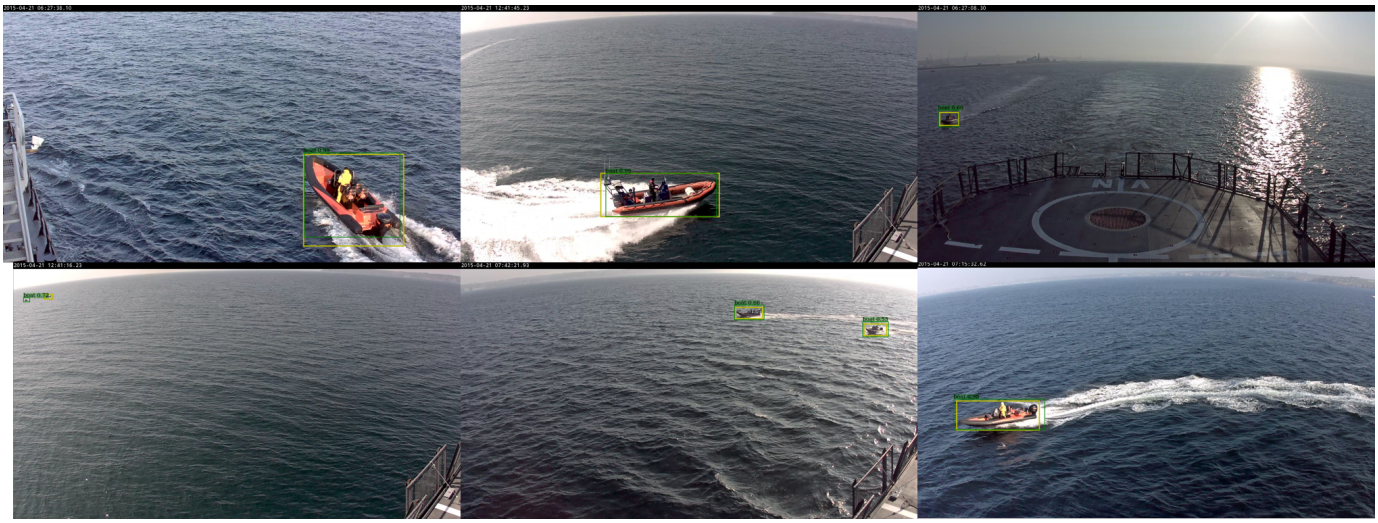

Fig. 9. Examples of images from Validation Set with ground truth annotations (yellow box) and model results (green box) visualized

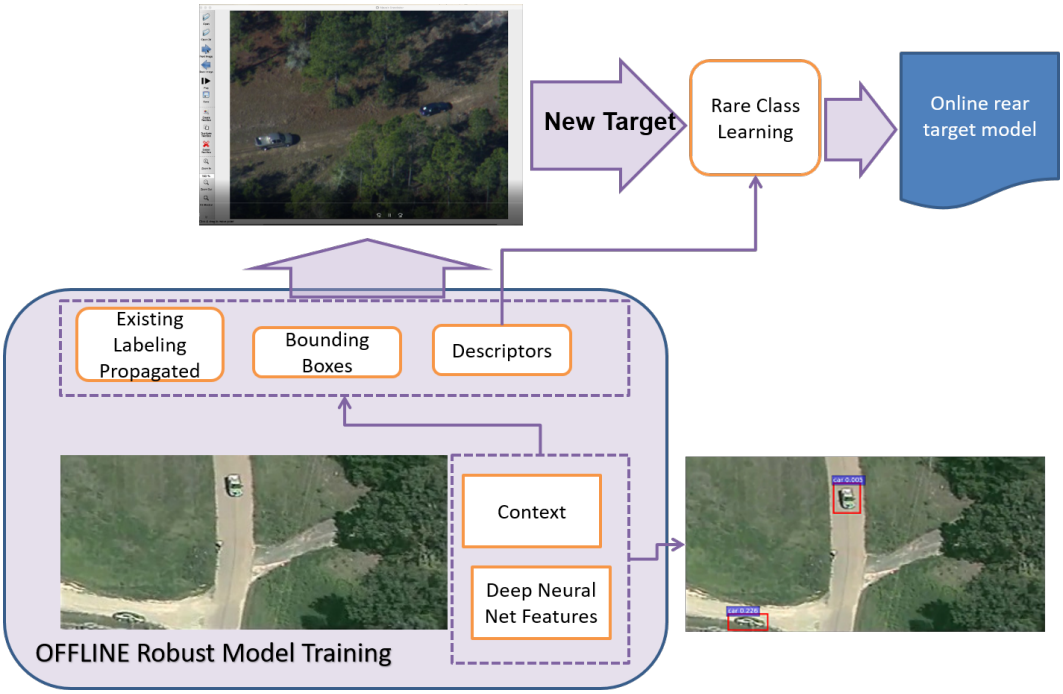


Fig. 10. AIM for Intermittent Modeling: system identifies target using existing set of target models. If the new target is spotted and few examples labeled, system triggers learning pipeline branch and applies the label to subsequent frames.

*c) Deep Learning Framework:* We rely on the baseline pytorch implementation of Detectron [24]. Our DNN is created using ResNet50 [13] architecture and for each network we train 180,000 epochs.

*d) System:* Server with four NVIDIA GeForce GTX 1080 Ti GPUs is used for training and inferencing.

## VI. Experiments, Results, and Findings

*a) Experiment 1: Domain Translation:* was to compare a model trained using COCO2017 training dataset to a model trained under the same condition using COCO2017 training dataset and IPATCH Low Level Challenge Dataset, as described in Table IV-A. The goal for this experiment was to provide an accurate assesment that features gathered from a large dataset of a different domain could help provide statisfactory results when applied to a dataset in a more obscure highly variant domain. Evaluation Metric used in this experiment setup was defined by COCO benchmark [8]. Note that $mAP@[0.5, 0.95]$ means average mAP over different IoU thresholds, from 0.5 to 0.95 with the step 0.05 e.g. (0.5, 0.55,

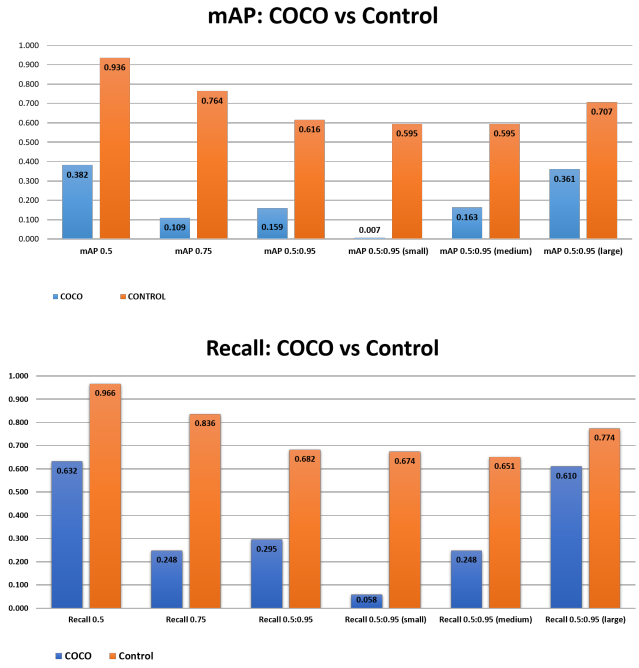


Fig. 11. mAP and Recall for different values of IoU and averaging over multiple IoU for all objects, and for small medium and large objects [8] for Detectron model trained on COCO data and trained on COCO + IPATCH data.

0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95). mAP@.75 means the mAP with IoU=0.75. Same applies for Recall. Numerical comparison is charted in Figure 11 where we compare average precision (top) and Recall (bottom) for both models for various values of IoU. Visual examples are presented in Figure 9. Note that model performance significantly improves across the board when domain relevant data was introduced to the learning pool. Even though the training set is much smaller than COCO training set and Validation set, it still offered a significant performance boost, specifically for small regions, as illustrated in Average Precision and Recall figures in Figure 11. These results show that many of the difficulties in the ocean environment can be captured by creating a dataset that encompasses the domain specific challenges.

| Train | Test | for IoU | 0.3 | 0.5 | 0.75 |
|---|---|---|---|---|---|
| COCO Train | Validation Set | AP | 0.518 | 0.382 | 0.19 |
| Control Set | Validation Set | AP | **0.936** | 0.936 | 0.764 |
| COCO Train | Validation Set | Recall | 0.814 | 0.636 | 0.248 |
| Control Set | Validation Set | Recall | **0.969** | 0.966 | 0.836 |

TABLE II
AVERAGE PRECISION AND RECALL VALUES FOR "BOAT" CLASS AND DIFFERENT MODELS FOR DIFFERENT VALUES OF IOU.

*b) Experiment 2: IoU Metric Sensitivity:* Maritime objects are small compared to the overall area of the image or vieo frame, see Figure 9 for examples. Traditional consumer IoU setup of 0.5 is too restrictive, as system marks hits as

Fig. 12. Multiple detections and scores and one ground truth frame for maritime boat object. Lowering IoU requirement from consumer data setup allows for better evaluation for the system.

missed in the evaluation. By relaxing the IoU threshold to account for sparsity of objects in maritime data, and including Experiment 1 domain data in the training, we have managed to boost the performance of the detectron on new domain from AP 38.2% to 93.6% and recall frm 63.6% to 96.9%. This experiment confirms our hypothesis. All models were then re-evaluated with the a lower bound of 0.3, reduced from 0.5. In natural scenes, lowering IoU results in lowering detectron precision tradeoff: more true positives **and** false positives pass. This assumption does not hold true due to sparsity of data: increase of recall was accompanied by increase in precision for the same models, as illustrated in Table II. Figure 12 offers an insight of typical relation between ground truth and detections in maritime IPATCh dataset.

| Detection Score Threshold | 0.05 | 0.025 | 0.01 | 0.005 | 0.001 |
|---|---|---|---|---|---|
| mAP 0.3 | 0.936 | 0.940 | 0.943 | 0.943 | 0.943 |
| mAP 0.5 | 0.936 | 0.940 | 0.943 | 0.943 | 0.941 |
| mAP 0.75 | 0.764 | 0.764 | 0.766 | 0.766 | 0.766 |
| mAP 0.3:0.95 | 0.707 | 0.709 | 0.709 | 0.710 | 0.710 |
| mAP 0.3:0.95(small) | 0.698 | 0.699 | 0.699 | 0.699 | 0.699 |
| mAP 0.3:0.95(medium) | 0.693 | 0.697 | 0.697 | 0.699 | 0.700 |
| 0.3:0.95(large) | 0.783 | 0.783 | 0.783 | 0.783 | 0.784 |
| Detection Score Threshold | 0.05 | 0.025 | 0.01 | 0.005 | 0.001 |
| Recall 0.3 | 0.969 | 0.977 | 0.980 | 0.983 | **0.986** |
| Recall 0.5 | 0.966 | 0.973 | 0.976 | 0.979 | **0.983** |
| Recall 0.75 | 0.836 | 0.839 | 0.840 | 0.841 | **0.843** |
| Recall 0.3:0.95 | 0.764 | 0.769 | 0.771 | 0.772 | **0.776** |
| Recall 0.3:0.95(small) | 0.764 | 0.768 | 0.768 | 0.769 | **0.771** |
| Recall 0.3:0.95(medium) | 0.737 | 0.743 | 0.746 | 0.749 | **0.753** |
| Recall 0.3:0.95(large) | 0.838 | 0.839 | 0.839 | 0.840 | **0.841** |

TABLE III
SCORE THRESHOLD REDUCTION WITH INCREASE OF RECALL WITH NO CORRESPONDING DECREASE IN MAP FOR A MODEL TRAINED ON CONTROL SET AND EVALUATED ON VALIDATION SET.

*c) Experiment 3: Model Threshold Sensitivity:* The goal of this experiment is to show that given domain specific characteristics, that hyper-parameter tuning can greatly affect results. Note that models are optimized for consumer imagery and natural scenes, that are often crowded. Maritime data has

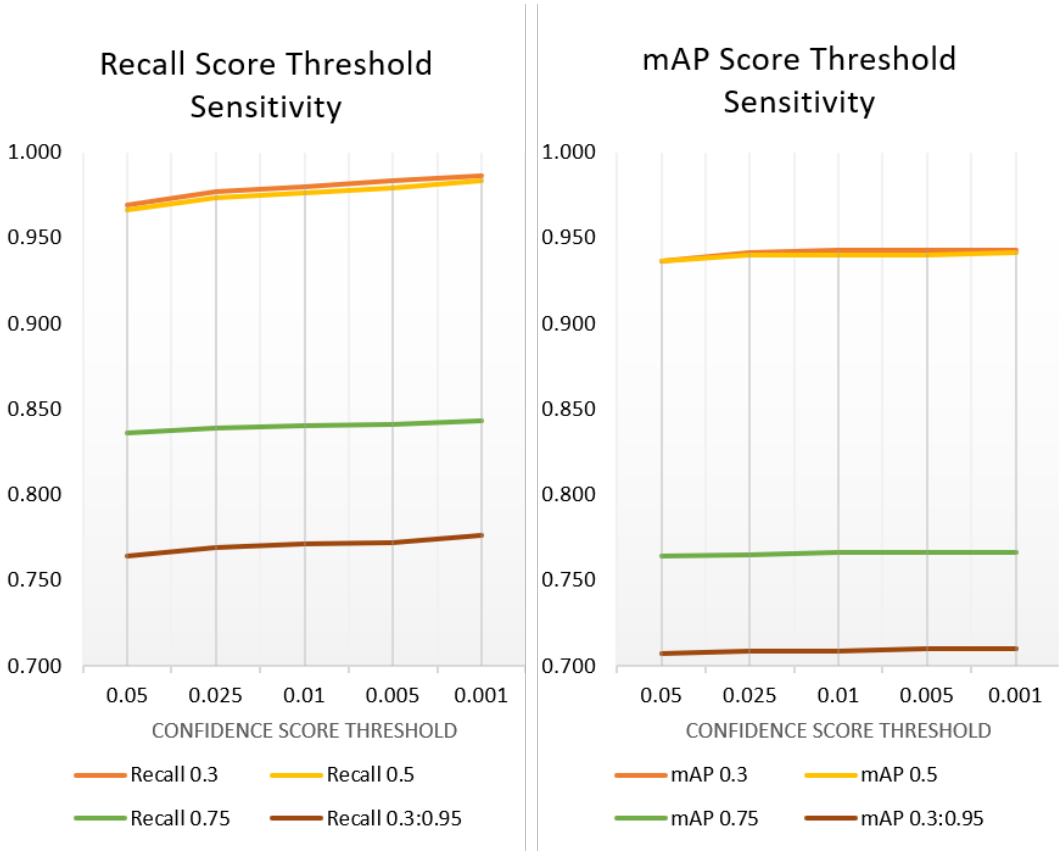


Fig. 13. Decreasing score threshold increases recall without corresponding reduction to mAP

prominent characteristics such as that objects are rare and different than the background, and crowding rarely occurs at sea. We use this domain characteristics to justify lowering the threshold of the expected predictions, and evaluate its influence on precision. The results are presented in Figure 13 and Table III. When the predictions are created by the model, it assigns a confidence score to each, which is used to rank the predictions. To continue in the evaluation, the detections must have a score above a detection score threshold. In an effort to increase recall, the control model was evaluated with lower thresholds as seen in figure 13. This lowering of the score increased the recall without dropping precision, and adjusting this parameter to domain characteristics allowed recall boost from 0.969 to 0.986, while precision gain went from 0.936 to 0.943 for IoU of 0.3.

*d) Experiment 4: RPN experiment:* The Region Proposal Network(RPN) is used to create proposed detections. Non-maximum suppression(NMS) is then applied to these proposals. NMS is ranking the detections by confidence score from highest to lowest. The highest ranked is then compared to each subsequent proposal using IoU. Any lower ranked proposal that has an IoU over a certain threshold is suppressed for being too similar. This reduces the number of proposals that need to be considered. The goal of this experiment was to increase the mAP and Recall scores by looking at mutliple RPN thresholds. This is due to the success of lowering the scoring threshold and acquiring satisfactory results. The RPN NMS threshold was increased to from the default of an IoU of 0.7 to 0.75, 0.8, 0.85, 0.9, and 0.95 table IV. This increase of the RPN NMS threshold was only applied during testing.

*e) Experiment 5: CTIT Fine-tuning:* The goal of this experiment was to provide results that show the CTIT is able to be used as an interface and assist with domain translation and fine tuning. In this experiment, we refined baseline COCO model by adding IPATCH Low Level Challenge Dataset to the training pipeline. The results are in line with findings, and adding domain data for generic model fine tuning significantly improves model performance, as demonstrated in Table V and Table VI.

*f) Summary:* We have demonstrated robust way of increasing model performance when adjusted to domain characteristics. The greatest discriminator is domain sensitive training data. Maritime domain lack alternative targets that would be incorrectly associated as maritime vehicles allowed us to relax the parameter constraints learned on urban natural scenes in consumer photos, adjust parameters of the model inference (IoU, RPN, and threshold), and achieve robust performance and high precision and recall numbers for this challenging dataset, as illustrated in Figure 14.

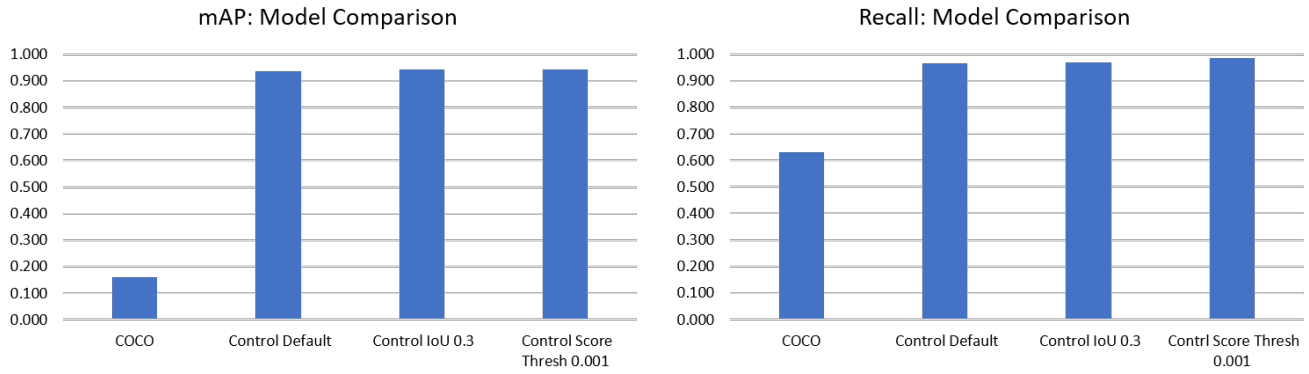


Fig. 14. Domain adaptation modeling sees most significant boost in the performance

| RPN NMS Threshold | 0.7 | 0.7 | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
|---|---|---|---|---|---|---|---|
| mAP 0.3 | 0.518 | 0.936 | 0.945 | 0.945 | 0.945 | **0.946** | 0.939 |
| mAP 0.5 | 0.382 | 0.936 | **0.939** | **0.939** | **0.939** | **0.939** | **0.939** |
| mAP 0.75 | 0.109 | 0.764 | **0.762** | 0.761 | 0.761 | 0.761 | 0.756 |
| mAP 0.3:0.95 | 0.252 | 0.707 | **0.710** | **0.710** | **0.710** | **0.710** | **0.706** |
| mAP 0.3:0.95(small) | 0.129 | 0.698 | **0.699** | **0.699** | 0.698 | 0.697 | 0.697 |
| mAP 0.3:0.95(medium) | 0.273 | **0.693** | **0.693** | 0.692 | 0.692 | 0.692 | 0.687 |
| mAP 0.3:0.95(large) | 0.484 | 0.783 | 0.785 | 0.784 | **0.787** | **0.787** | 0.786 |
| RPN NMS Threshold | 0.7 | 0.7 | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| Recall 0.3 | 0.814 | 0.969 | 0.970 | **0.972** | **0.972** | **0.972** | 0.966 |
| Recall 0.5 | 0.632 | 0.966 | 0.967 | **0.969** | **0.969** | 0.968 | 0.962 |
| Recall 0.75 | 0.248 | **0.836** | **0.836** | 0.837 | 0.833 | 0.833 | 0.829 |
| Recall 0.3:0.95 | 0.431 | 0.764 | **0.765** | **0.765** | **0.765** | **0.765** | 0.760 |
| Recall 0.3:0.95(small) | 0.247 | 0.764 | **0.765** | 0.765 | 0.762 | 0.762 | 0.760 |
| Recall 0.3:0.95(medium) | 0.382 | 0.737 | 0.738 | **0.739** | 0.738 | 0.738 | 0.731 |
| Recall 0.3:0.95(large) | 0.712 | 0.838 | 0.838 | **0.843** | 0.841 | 0.841 | 0.841 |

TABLE IV
REGION PROPOSAL NETWORK THRESHOLD VARIATION AND PERFORMANCE CHANGE FOR A MODEL TRAINED ON CONTROL SET AND EVALUATED ON VALIDATION SET.

| **IoU** | 0.3 | 0.5 | 0.75 | 0.3:0.95 | 0.5:0.95 | 0.3:0.95 | 0.5:0.95 | 0.3:0.95 | 0.5:0.95 | 0.3:0.95 | 0.5:0.95 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** / Area | all | all | all | all | all | small | small | medium | medium | large | large |
| **COCO Train** | 0.518 | 0.382 | 0.109 | 0.252 | 0.159 | 0.129 | 0.007 | 0.273 | 0.163 | 0.484 | 0.361 |
| **Control** | 0.936 | 0.936 | 0.764 | 0.707 | 0.616 | 0.698 | 0.595 | 0.693 | 0.595 | 0.783 | 0.707 |
| **COCO Train, finetuned with IPATCH** | 0.945 | 0.945 | 0.792 | 0.723 | - | 0.691 | - | 0.707 | - | 0.810 | - |

TABLE V
**mAP** SCORES, AS DEFINED BY COCO BENCHMARK [8], FOR DOMAIN MODELS

| **IoU** | 0.3 | 0.5 | 0.75 | 0.3:0.95 | 0.5:0.95 | 0.3:0.95 | 0.5:0.95 | 0.3:0.95 | 0.5:0.95 | 0.3:0.95 | 0.5:0.95 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** / Area | all | all | all | all | all | small | small | medium | medium | large | large |
| **COCO Train** | 0.814 | 0.632 | 0.248 | 0.431 | 0.295 | 0.247 | 0.058 | 0.382 | 0.248 | 0.712 | 0.610 |
| **Control** | 0.969 | 0.966 | 0.836 | 0.764 | 0.682 | 0.764 | 0.674 | 0.737 | 0.651 | 0.838 | 0.774 |
| **COCO Train, finetuned with IPATCH** | 0.973 | 0.973 | 0.857 | 0.774 | - | 0.767 | - | 0.749 | - | 0.849 | - |

TABLE VI
**Recall** SCORES FOR VALIDATION SET, AS DEFINED BY COCO BENCHMARK [8], FOR DOMAIN MODELS

## VII. Conclusion

We propose a new approach for transfer learning and persistent model reuse for domain adaptation of Deep Convolutional Neural Networks. Varying resolution quality of operational data, size of objects of interest, view occlusions, and large variation in sensors due to sheer nature of overhead systems as compared to consumer devices contribute to degradation of the classification and recognition when applied to overhead sensor data. In this paper, we propose a clear path towards object recognition solution for overhead sensor feeds, and demonstrate its usability for collaborative maritime asset identification. First, we exploit the domain characteristics to refine the deep learning framework, and show that our transfer learning strategy produces models that reliably and accurately discriminate sea objects from overhead imagery data comparable to consumer data benchmarks. Next, we introduce the notion of persistent and intermittent modeling strategies in collaborative environments, and propose as well as implement two collaborative tools to aid the object recognition: one that support persistent modeling, and the other that supports labeling and intermittent modeling. We propose a data science approach to bring deep learning application in maritime domain to the same level as for consumer data, as our persistent models achieve over 94.5% precision and 97.5% recall rate for generic boat classification in IPATCH data [1].

## VIII. Acknowledgments

This material is based upon work supported by NAVAIR under contracts STTR N68335-16-C-0028 and SBIR N68335-18-C-0199. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Authors thank Fiona Chandrasekaran, undergraduate researcher from University of California Los Angeles, for the extensive annotations of IPATCH dataset, and to Jason Bunk for Fig. 1 and for proving us right. The authors would like to thank Texas State University REU Software Systems and Analysis for providing a framework for this work(http://reussa.cs.txstate.edu/). We would also like to acknowledge Roy Tseng, a graduate student at National Tsing Hua University in Taiwan, for a pytorch implementation of Detectron(https://github.com/roytseng-tw/Detectron.pytorch).

## References

[1] L. Patino, T. Cane, A. Vallee, and J. Ferryman, "Pets 2016: Dataset and challenge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems (NIPS)*, 2012.

[3] K. H. J. S. Jifeng Dai, Yi Li, "R-FCN: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, 2016.

[4] 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, pp. 1137–1149, June 2017.

[5] K. Hara, R. Vemulapalli, and R. Chellappa, "Designing deep convolutional neural networks for continuous object orientation estimation," *arXiv*, 2017.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *CoRR*, 2014.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*. Springer International Publishing, 2014, pp. 740–755.

[9] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4077–4087.

[10] S. Golder and **Jelena Tešić**, "Collaborative tagging of multimedia," *IEEE MultiMedia*, vol. 15, no. 3, pp. 12–13, July 2008.

[11] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[12] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, 2014.

[15] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[16] H. evikalp, G. G. Dordinejad, and M. Elmas, "Feature extraction with convolutional neural networks for aerial image retrieval," in *2017 25th IEEE Signal Processing and Communications Applications Conference (SIU)*, May 2017, pp. 1–4.

[17] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 3320–3328.

[19] F. J. Huang and Y. LeCun, "Large-scale learning with svm and convolutional for generic object categorization," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, June 2006, pp. 284–291.

[20] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, Feb 2017.

[21] Y.-X. Wang and M. Hebert, "Learning from small sample sets by combining unsupervised meta-training with cnns," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 244–252.

[22] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, "LSDA: Large scale detection through adaptation," in *Neural Information Processing Systems (NIPS)*, 2014.

[23] J. Tesić and E. Staudt, "annotator: train a new model," https://youtu.be/QoUSB0pj0u8, 2017.

[24] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[25] J. Tesić and E. Staudt, "annotator: label new instances," https://youtu.be/YDQ18-lPs9Q, 2017.

[26] L. Patino, T. Nawaz, T. Cane, and J. Ferryman, "Pets 2017: Dataset and challenge," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 2126–2132.

[27] L. Patino, T. Cane, A. Vallee, and J. Ferryman, "Pets 2016: Dataset and challenge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.