Applying Deep Learning to Improve Maritime Situational Awareness

Kathy Tang
Stottler Henke Associates, Inc.
1650 S. Amphlett Blvd. Ste. 300
San Mateo, CA 94402

David Crandall
School of Informatics and Computing
Indiana University
Bloomington, IN 47401

ABSTRACT
We describe a system called ExPATSS (Extensible Platform for Automated Tactical Sensor Screening) that we are developing for the Navy to automatically detect and classify ships from onboard an aircraft carrier. ExPATSS simultaneously processes several video streams for ship detection and classification, in order to reduce the attention and concentration currently required of human sensor operators, who presently have to manually monitor all the video streams at once. ExPATSS leverages recent developments in deep learning, specifically Convolutional Neural Networks (CNN), to accurately detect and classify ships. ExPATSS has been developed and tested using real-world data and this paper discusses the effectiveness of using CNN within the system.

CCS Concepts
• Computing methodologies ➔ Artificial Intelligence ➔ Computer Vision

Keywords
Computer Vision; Deep Learning; Convolution Neural Networks; Object Recognition; Image Processing

1. INTRODUCTION
Accurate, rapid acquisition and interpretation of sensor data is critical to modern naval warfare. The need to maintain situational awareness in both open water and coastal zones necessitates a broad array of sensing apparatuses distributed across sea, land, and air vessels. Accurate interpretation and integration of the data from these sensors can mean the difference between early detection of an enemy threat and mission failure, including the loss of lives.

Proper interpretation of data from cameras, including both visual-spectrum and infrared sensors such as the FLIR and multi-mode Inverse Synthetic Aperture Radar (ISAR) installed on a modern Anti-Submarine Warfare (ASW) platform (e.g., an MH-60R helicopter) requires intense concentration by a highly trained human operator. The tactical information acquired from even one of these sensors can include ten or more unique contacts of interest, making the potential for cognitive overload a significant risk even for a single data stream. As the number of data streams increases, so too does the likelihood of missed or mischaracterized contacts, resulting in potentially erroneous analysis and delay or disruption of the tactical decision-making process.

Systems have been developed and deployed to assist with the processing of ASW sensor data streams. However, these systems still rely heavily on human operators to carry out the initial identification and classification of contacts, as well as to assess their significance to their mission objectives — all tasks that require intense attentional commitment. As such, current systems can only offload a portion of overall operator effort and are really only helpful for on-craft processing of a small number of simultaneous sensor streams.

To make matters worse, next-generation Navy systems are evolving toward integrated multi-platform, multi-sensor platforms that incorporate real-time transmissions of raw sensor data from multiple vehicles into a centralized command and control framework for analysis. This shift has a number of advantages, including a potential to significantly increase the speed at which changes in the tactical picture are understood and communicated up the chain of command, and an opportunity for increased fusion of data from disparate sensor sources to generate a more complete and accurate picture of the tactical situation as it develops. However, it is likely that available manpower will remain constant or even decrease in the future, resulting in individual sensor operators in centralized command and control analysis positions required to analyze significantly greater quantities of simultaneous streaming data. In a realistic ASW scenario with four airborne helicopters simultaneously transmitting two sensor streams each (FLIR and ISAR), this increase in sensor data relative to available operator attention is almost certain to degrade overall operator performance and reduce situational awareness.

In this paper, we describe a system called ExPATSS (Extensible Platform for Automated Tactical Sensor Screening) that Stottler Henke Associates is developing for the Navy in order to reduce operator overload and to minimize the effort and attention required of the operator per-stream. The overall goal of ExPATSS is to automatically perform some of the raw sensor data analysis involved in ASW in order to aid sensor operators in detecting and tracking contacts across a large number of simultaneous data streams. By freeing operators to focus their attention on higher-priority events and contacts, we have found that the advantages of centralized multi-sensor video processing can be realized without additional manpower requirements and with no reduction in accuracy or latency of analysis. In particular, here we describe applying deep learning with Convolutional Neural Networks (CNNs) to image data, in order to automatically detect and classify ships from onboard an aircraft carrier. We evaluate the ability of CNNs to recognize whether or not ships are present in images, and if so, to classify the type of ship into one of several categories of interest. For training the classifier, we use publicly available imagery downloaded from social media websites as a convenient, inexpensive, and large-scale data source. We also present results using real-world data.

2. RELATED WORK
Object detection and recognition is a central problem in computer vision and has been studied extensively for many decades. Until recently, the standard approaches for object recognition involved using hand-designed algorithms that tried to abstract important visual characteristics into statistical feature vectors, for input into

standard machine learning algorithms like Support Vector Machines (SVMs). However, since a deep learning-based technique won the 2012 ImageNet challenge [4], there has been a dramatic shift in interest within the computer vision community towards techniques that automatically learn the feature extraction stage, instead of relying on hand-designed algorithms. The most popular technique so far is based on Convolutional Neural Networks, which are similar to the classic feed-forward neural networks that have been studied for decades, but typically are much deeper, have many more parameters, and have a unique architecture that encourages them to cue on spatially-contiguous regions within an image. CNN-based and deep learning techniques in general have now been shown to outperform traditional techniques on a wide variety of problems, ranging from image classification [4], to object detection [5], to image captioning [8], among many others. Much of the rapid progress in this area is also due to large-scale image collections (like ImageNet, which contains millions of labeled images [3]), and high-quality, open-source CNN implementations like Caffe [2].

3. APPLYING DEEP LEARNING TO SHIP CLASSIFICATION

We now describe how we apply CNNs to our novel ship classification application. We first give an overview of the ExPATSS system in Section 3.1, and then discuss how we apply CNNs to ship recognition in Section 3.2.

3.1 ExPATSS System Overview

The ExPATSS system receives video input streams from multiple sources. There are two general types of input streams: ISAR (radar input) and FLIR (visible light and infrared input). In this paper, we focus on processing of the FLIR visible light input.

Each stream is individually processed by a Recognition Engine, which performs object detection, classification, and tracking. Then the data from each Recognition Engine is aggregated by the Correlation Module, which merges the tracking results to identify and label identical objects caught across multiple video streams. Finally, the Prioritization Module processes the detection/classification results and assigns a priority value to each video stream, which aids the operator in appropriately directing his or her attention towards the most important stream.

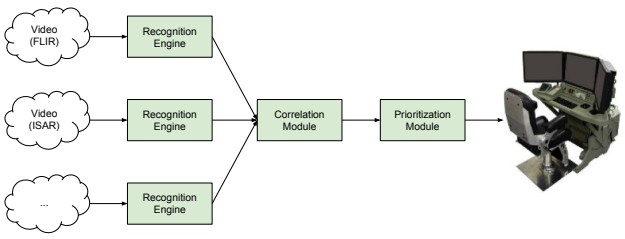

Figure 1: ExPATSS Overview Diagram

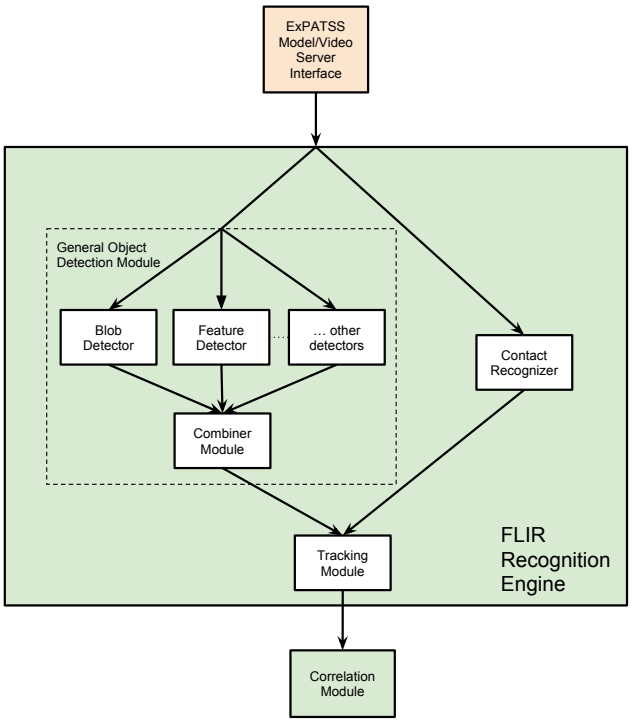

Figure 2: FLIR Recognition Engine

Within ExPATSS, the FLIR Recognition Engine contains the core vision capabilities. A single FLIR Recognition Engine is created for each FLIR video stream, which in turn processes FLIR video data to automatically detect and classify ships.

The FLIR Recognition Engine, shown in Figure 2, is composed of three major components: the General Object Detection Module, the Contact Recognizer, and the Tracking Module. The General Object Detection Module consists of many different types of object detectors, each using a different algorithm to detect objects in the frame. The Combiner Module aggregates all objects detected by the individual object detectors and removes duplicates. The Contact Recognizer is tasked with recognizing/classifying ship types, while the Tracking Module collects the output from the Combiner Module and the Contact Recognizers to keep a track or history of each ship detected.

Within the FLIR Recognition, the General Object Detection Module detects non-specific objects in the video frame by using various object detectors such as a Blob Detector, which looks for contiguous homogeneous image regions, and a Feature Detector, which cues on corner detection to detect objects. The Combiner Module combines similar objects from different detectors into a single object to avoid detecting duplicates, by comparing candidate object detections using size and pixel location, and combining them into a single bounding box that minimally encloses both original boxes if the detections are significantly overlapping. The Contact Recognizer performs classification by using the Convolutional Neural Network (CNN)-based recognition that we describe in more detail here. Finally, the Tracking Module compares object locations from frame to frame. It assigns every contact a track ID in each frame, and if there is sufficient positional overlap between a contact on the current frame and one on a previous frame, a single ID is assigned.

For each frame, the FLIR Recognition Engine outputs the pixel location of each object detected as well as its ID and classification.

3.2 Application of CNNs

The Contact Recognizer is the main image classification module. We use a Convolutional Neural Network in the Contact Recognizer to estimate a class label for each detected object. The Contact Recognizer uses Caffe, an open source deep learning framework, to perform classification at the frame level.

For our initial experiments presented here, we focus on an image-level classification task, i.e. deciding if a photo contains a given ship category or not. In particular, we classify an image into either open water (no ship visible), or one of five ship types: warship, speedboat, sailboat, merchant (cargo) ship, or cruise ship. These classifications were chosen based on the general ship types of interest to the Navy. We included a class for open water so that we can classify frames with no ship activity, which is useful for skipping to segments of interesting footage during playback. The open water class is also helpful to minimize the false positive rate of the object detectors. The object detectors will sometimes detect water movement (waves, splashing water, reflections, etc.) as objects of interest. We are able to lower the false positive detection rate by filtering out detected objects when the CNN classification result for a frame is open water.

3.2.1 Training Data

Deep learning-based models like CNNs typically require large amounts of training data, which is challenging for applications like our task where training data can be expensive to collect. We thus used publicly-available images downloaded from a social photo sharing website (Flickr.com) as our training dataset. These are consumer-style photos that may have significantly different properties from the images collected by cameras in a real application. For example, most users only upload their "good" photos – sharp, well-composed, with interesting or unusual content – whereas real applications may capture images with significant blur, noise, etc. Nevertheless, because images on Flickr are uploaded by so many different users under many different imaging scenarios, we find that they are diverse enough to create a reasonable, low-cost training dataset for our task.

In particular, two human coders were asked to browse Flickr to find images from each of the five categories of ships (cruise, cargo, speed, sail, and war) as well as photos of open water. They used different strategies including searching based on keywords and browsing ocean-related Flickr groups and photo collections. They were asked to try to collect as wide a variety of images as possible, e.g. featuring ships in a variety of different poses, illumination conditions, sizes, etc., and to avoid collecting duplicate or near-duplicate photos or many photos taken by the same photographer. They were also asked to ignore photos that were obviously synthesized or edited (e.g. cartoons of ships, or photos with prominent watermarks).

We then downloaded the photos they selected from Flickr, at a resolution of 500 pixels on the longest side. The human coders were then asked to label the images by drawing bounding boxes around all instances of ships visible in any of the images. We used the publicly-available LabelMe [7] tool. This yielded a set of 3,040 images, or about 500 images per class (the distribution across classes was approximately uniform), with a total of 3,533 bounding boxes. We split the images into training, validation, and testing sets of 2,128, 456, and 456 images, respectively.

3.2.2 Training the models

We used the open-source Caffe package [2] to train a 6-way classifier. Instead of training a network from scratch, we used a model trained on the ImageNet dataset of millions of images [3] to initialize the parameters of the network, and then "fine-tuned" on our much smaller dataset. This approach is often used to overcome limited training data; the intuition is that although the initialization parameters were derived for a completely different task and dataset, they still incorporate a generic enough representation of the visual world to be useful on other tasks. We used the AlexNet architecture [4], consisting of 5 convolutional layers (with rectified linear (ReLU) activation functions, and spatial max-pooling) followed by three fully-connected layers (with drop-out), except that we replaced the last layer with a six-output layer to fit our six-class problem. We trained the network for a total of 100,000 iterations using a batch size of 50, a base learning rate of 0.001, a step size of 20,000, a momentum of 0.9, and a weight decay of 0.005. Training took approximately nine hours on a single system with an NVidia Tesla K40 GPU.

4. EXPERIMENTAL RESULTS

We now present preliminary experimental results of our ship detection and classification system based on deep learning. We present results on the Flickr data in Section 4.1. And we present some initial results for real-world data in our EXPATSS application in Section 4.2.

4.1 Results on Flickr dataset

We first evaluated the image-level classifier on the held-out portion of our Flickr dataset. To do this, we resized each image to 227 x 227 pixels, and then presented it as input to the CNN trained in the last section. The CNN generates a confidence for each of the six classes, and we chose the single highest as the predicted class.

The CNN achieved an overall correct classification rate of about 91.4% on this 6-way problem, versus a majority class baseline (i.e. predicting the most frequent class) of 19.7%. Table 1 presents the confusion matrix for this experiment. As the table shows, the majority of errors occur because the classifier misses the ship in an image, incorrectly predicting it as open water. Many of these cases are small ships far in the distance, or ships photographed

Table 1 Confusion matrix for image-level ship classification on the Flickr test dataset

		Detected class					
		Cruise	**Cargo**	**Open**	**Sail**	**Speed**	**War**
Actual class	Cruise	86.4%	1.2%	7.4%	0.0%	4.9%	0.0%
	Cargo	1.5%	88.1%	4.5%	0.0%	3.0%	3.0%
	Open	0.0%	0.0%	97.8%	1.1%	1.1%	0.0%
	Sail	0.0%	1.2%	4.8%	91.4%	1.2%	1.2%
	Speed	0.0%	1.5%	3.0%	1.5%	94.0%	0.0%
	War	1.4%	2.8%	4.2%	0.0%	1.4%	90.1%

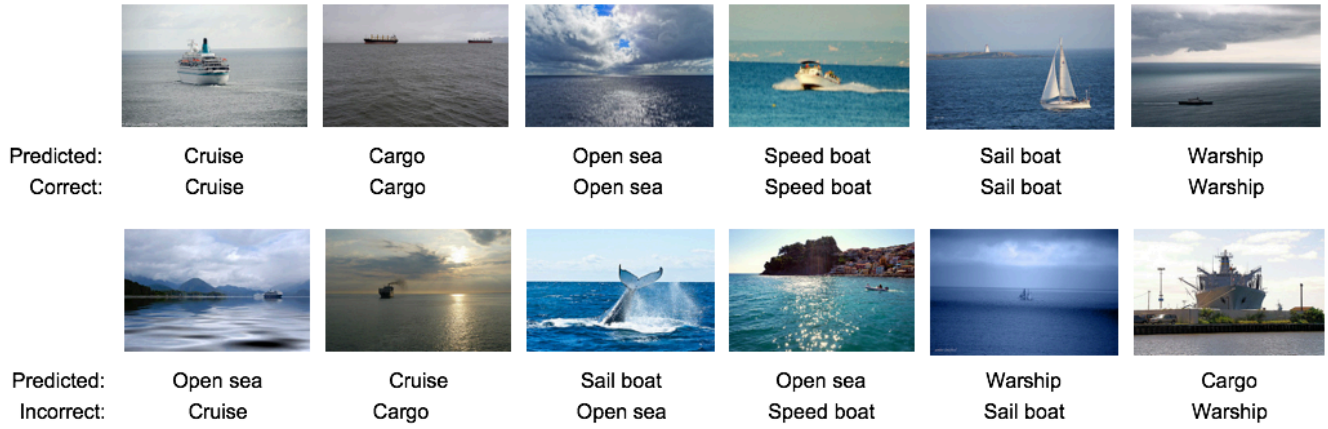

Figure 3: Randomly-sampled correctly (top) and incorrectly (bottom) classified images for each class in the Flickr test dataset.

from very unusual perspectives. Excluding this error mode, the correct classification rate rises to 95.1%, indicating that once a ship is found in an image, the chance of it being identified as the correct type is quite high. Many of the remaining errors are due to ambiguity between ship types in the training data, because we did not define them precisely to our human coders (who were also not maritime experts). For instance, a yacht could be identified as a cruise ship or a speedboat, while small military crafts could be classified as warships or speedboats. Figure 3 presents some sample correctly- and incorrectly-classified images.

Given the relatively small amounts of real-world training data in the target ExPATSS application, another important consideration of applying CNNs to this problem is how much training data they need to achieve acceptable levels of classification accuracy. To measure this, we performed an ablation study on our Flickr dataset, in which we trained on subsets of the training data of different sizes. Figure 4 presents results ranging from only 10 training images total (i.e., an average of less than two images per class) to nearly the full training dataset of 2,000 images (averaging about 330 images per class). For each training set size, we performed five trials and show the mean and error bars (plus and minus two standard errors). Of course, accuracy is quite low with a training set as small as 10 images, and the variation across trials is quite high, indicating of course that which images are chosen in the training set is critical. However, even at these small training sets, classification accuracy is still about twice that of the majority class classifier (about 50% versus 19%), which is likely due to the fact that we fine-tune a classifier trained on ImageNet instead of training from scratch (as has been observed for one-shot learning with CNNs in other work [6]). Every factor of 10 increase in training set size reduces the error rate by a factor of roughly two (50% error at 10, 20% error at 100, 11% error at 1,000), and it appears that increasing the training set further will likely continue to improve results above our training set size of 2,000.

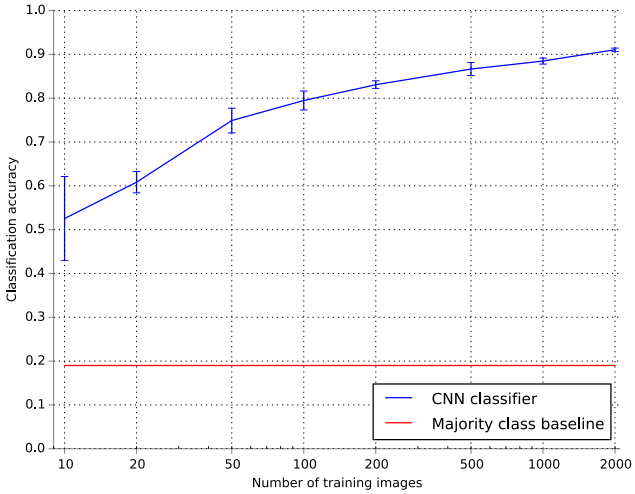

Figure 4: Six-way classification accuracy on the Flickr dataset, as a function of total training set size, averaged over five trials. Error bars show plus/minus two standard errors.

Although training the CNN model for 100,000 iterations requires significant computational resources (about 9 hours on a single Tesla K40 GPU), in a real application the classification running time is much more critical. After the network has been loaded and initialized in memory and images have been preprocessed, we measured a GPU-based classification time of about 0.048 seconds per image, or about 20.7 frames per second, suggesting that the technique could be run in near real-time if a GPU is available. CPU-only classification was significantly slower, at about 0.91 seconds per image.

4.2 Results of ExPATSS Using CNN

Table 2, shows the result of ExPATSS in classifying ships in our test videos.

Each video has one main ship in view. The correct classification rate represents the performance of the system in classifying the main ship in view. The correct classification rate is calculated based on a per-frame basis. It represents the percentage of frames the system correctly classifies the ship. Using the CNN classifier, the ExPATSS system does very well in 9 out of 11 test videos.

Table 2 Classification results on ExPATSS data.

Video	Number of frames	Correct Classification Rate
speed_boat_1	3,625	0.79
speed_boat_2	4,320	0.68
warship_1	750	0.91
sail_1	570	0.00
warship_2	600	0.97
speed_boat_3	480	0.67
warship_3	1,890	0.14
warship_4	1,980	0.91
open_water_1	114,497	1.00
stormy _waters_1	7,223	1.00
stormy_waters_2	441	1.00

Figure 5 below shows sample frames that the ExPATSS system was able to correctly classify.

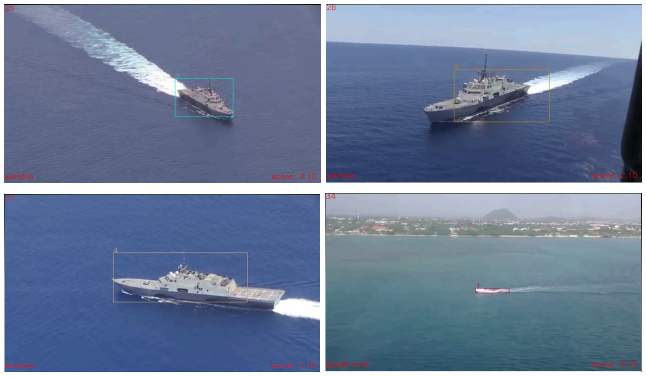

Figure 5: Sample ExPATSS frames correctly classified.

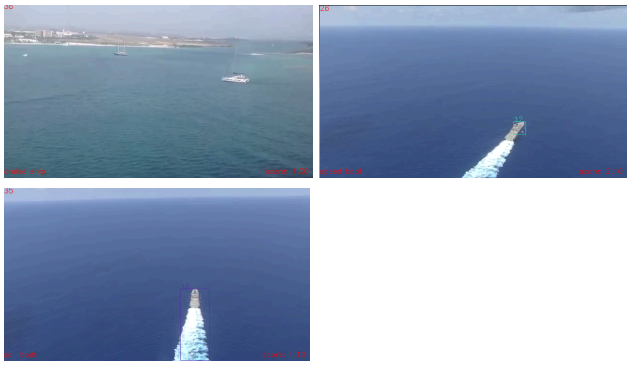

Figure 6: Sample incorrectly-classified ExPATSS frames.

Figure 6 above shows a sample collection of incorrect classifications. The system performance decreases as the resolution of the ship decreases. Additionally, the system fails to classify the sailboat without the sail up, as well as the warships from the bow view.

ExPATSS can very accurately categorize frames with no activity as open water. We achieved perfect results (no false positives) over 1:03:27 (114,197 fames) of open ocean, and 0:04:18 (7,664 frames) of stormy choppy water.

Figure 7: Sample Open Ocean Frame

Figure 8: Sample Stormy Ocean Frame

These results suggest that ExPATSS can be used to easily skip over uninteresting segments in playback.

5. CONCLUSION

To improve maritime situational awareness and to reduce operator overload, we have designed and implemented a system to automatically detect and classify ships in maritime scenes. The system uses CNNs for image classification and achieves higher than 79% correct classification rate for 7 out of 11 videos of real maritime data. The system achieves higher than 67% correct classification rate for 9 out of 11 videos. The system classifies long segments of open water (both stormy and calm) with 100% accuracy over 114,197 fames. This allows the system to be used during video playback to skip uninteresting segments of footage, which can save enormous amounts of human-hours.

Our future work includes improving the current CNN model by using a larger and more diverse training set. We also plan to train new CNN models with more specific ship classes, i.e. training a CNN model to distinguish between different types of warships. Finally, we plan to apply CNN-based approaches directly to the object detection problem, to not only classify images but also identify where in the image the ships of interest are. Until recently, CNN-based detection techniques have been too slow for online applications. However, we have achieved promising initial results with the recently-proposed technique of Redmon *et al.* [5], which can deliver near real-time object detection in some scenarios.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the Naval Sea Systems Command under Contract No. N00024-14-C-4082.

We also thank Luke Song, Stottler Henke Associates, Inc., for his contribution in the development of the ExPATSS system.

7. DISCLAIMER

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Sea Systems Command.

8. REFERENCES

[1] Everingham, M., S. M. A. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. "The PASCAL Visual Object Classes Challenge – A Retrospective." *International Journal of Computer Vision,* 111:98-136, 2015.

[2] Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. "Caffe: Convolutional Architecture for Fast Feature Embedding," *arXiv preprint 1408.5093*, 2014.

[3] Russakovsky, O., 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," *arXiv preprint 1409.0575*, 2014.

[4] Krizhevsky, A., I. Sutskever, and G. Hinton. "ImageNet Classification with Deep Convolutional Networks," *Advances in Neural Information Processing Systems,* 2012.

[5] Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. "You Only Look Once: Unifed, Real-time Object Detection," *arXiv preprint 1506.02640,* 2015.

[6] Hoffman, J., E. Tzeng, J. Donahue, Y. Jia, K. Saenko, and T. Darrell. "One-shot adaptation of supervised deep convolutional models," *arXiv preprint 1312.6204,* 2013.

[7] Russell, B., A. Torralba, K. Murphy, and W. Freeman. "LabelMe: a database and web-based tool for image annotation." *International Journal of Computer Vision,* 77:157-173, 2008.

[8] Karpathy, A. and L. Fei-Fei. "Deep visual-semantic alignments for generating image descriptions," *IEEE Conference on Computer Vision and Pattern Recognition,* 2015.