Non-Native Phragmites Colony Detection System

Deepak Sharma
Rochester Institute of Technology
ds5930@g.rit.edu

Abstract

Phragmites is an invasive species of grass that has shown harmful effects on biodiversity of North American flora. It displaces native vegetation in the region. Currently New York Department of Environmental Conservation has no effective means to locate infested areas.

We propose a solution to locate such areas infested with phragmites using Google Street Images. Recent research and developments in the field of Deep learning have enabled us to produce state of the art object classification and localization systems which possess better discriminating abilities than humans. This project proposes a deep learning based system that can consume geo-referenced images from Google Street Images and produce the location of phragmites on the maps of New York.

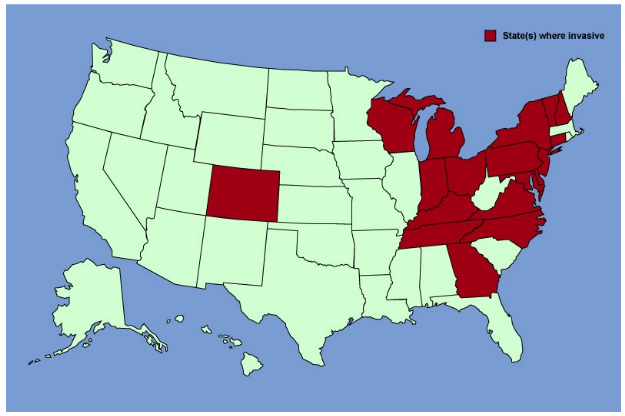


Figure 1: Phragmites affected states (Image source [14])

1. Introduction

During the era of industrial revolution in 18th and 19th centuries ancestors of the non-native phragmites reached to the ports of the New York. Non-native phragmites is ubiquitous in the eastern coastal states of the USA as shown in the map 1. In many parts of these regions, non-native phragmites is overthrowing the native flora of marshlands and wetlands. New York state is the most affected state where non-native phragmites can be located in brackish marshlands, wetlands, swales and even in small ditches near roadsides. The rich ecosystem and biodiversity of these areas are hampered due to the fast growth of phragmites. Each plant can produce more than a thousand seeds and a large colony of the non-native phragmites can stay alive for decades. However, there are many methodologies available for removing the non-native phragmites like burning, mowing and grazing. But New York Department of Environmental Conservation is facing challenges in locating the non-phragmites clumps [13, 18, 9, 19].

The objective of the project is to develop a deep-learning based image localization and classification system which can detect non-native phragmites colonies in the Google Street View images of the New York state. Ultimately the proposed system will provide geo-coordinates of the locations of phragmites colonies.

Major challenge for the proposed classification model is not only to differentiate phragmites from other similar looking species of grass but also taking into account the fact that looks and shape of the phragmites change during the year, for example, it becomes gold and purple from green in months of August.

2. Related work

Most of the early attempts made for performing image classification were based on shallow learning approaches. In these approaches, before passing image data from classification model, feature extractors like SIFT [12], HOG [2] and fisher vector[16] were used for extracting features, these extracted features were used for classifying images. The accuracy of these approaches was highly dependent on the quality of features extracted by the feature extractors. Due to the subtle difference between the species of grass, the performance of these techniques might not be effective for our problem.

In the domain of deep learning, the architecture of Convolutional neural networks provide computation platform for the learning algorithm to develop complex nonlinear relationships between attributes and class labels. The success of AlexNet[10] has inspired computer vision society to explore more efficient solutions for problems related to the object classification and localization using CNN based models. Though for training a CNN from scratch, we need sufficiently large training dataset and computation power in form of high-speed GPU, but in a number of applications high classification accuracy has been achieved with relatively small training labeled dataset using the concept of transfer learning. CNN architecture extracts features hierarchically, hence initial layers of the CNN learn to extract the basic features like edges and color blob of images. These learned parameters of the initial layers are also applicable for images of other domains. Recently, a number of elegant computer-vision related applications have been developed using transfer learning [22, 11]. Researchers Zeng and Ji have used pre-trained CNN VGG16 model for classification of mouse brain section images [22]. Similarly, researchers Lee et al. have used pre-trained CNN network for tree leaves classification [11]. Tree leaves and brain section images are highly domain specific but the reported results indicate that transfer learning can be effective for such domain-specific classification tasks.

A pre-trained CNN can be used for classification problem, but in an image there can be multiple objects. Either we can train the CNN for detecting phragmites present with any kind of objects or we can have a system which first performs object localization by identifying the location of multiple objects and then perform classification task on these objects. The Later approach is more efficient because it processes each object independently. Though the task of identifying the non-native phragmites in geo-referenced Google Street images is unprecedented but recent work by Wegner et al. for cataloging urban trees in Google maps images is similar to our problem. While designing the image classification system they have used publicly available implementation of the Faster R-CNN. Faster R-CNN performs training and testing tasks faster than its successor R-CNN [6]. Faster R-CNN system first generates region proposals(bounding boxes on objects) using RPN network and then performs object detection task for each generated proposal using a Fast R-CNN network [15]. Results have shown that this system is much efficient in object reorganization, therefore we have proposed faster R-CNN network-based object localization and reorganization network. Wegner et al. has also provided a transformation function for converting the georeferenced pixel coordinates to geographical coordinate which will also be useful in the proposed application in generating exact location of non-native phragmites colony on the map.

3. Creating the Dataset

In this section process of data acquisition from Google Street view image server, pre-processing of raw data, data labeling, and data preparations have been explained.

3.1. Data Sources

No labeled dataset of native and non-native Phragmites was available. So for the initial stages of the project, a dataset was constructed using images extracted from various sources. Using Fatkun Batch Downloader [4], around 600 google search images of phragmites were downloaded. Many of these downloaded images were not relevant to Phragmites. In many of these images, other objects like human along with phragmites were present. A manual cleaning was performed on these images by removing irrelevant samples.

The second major source of the phragmites samples was publicly available phragmites videos on some video hosting services like YouTube. Using FFmpeg [8] utility frames of the videos were extracted. In these videos, anchors were holding a camera and walking around the phragmites or camera was still but the wind was moving the phragmites or anchor was changing the zoom parameter of the camera. In all these scenarios, by extracting frames from such videos, we collected samples of same objects not only with different resolution and shape but also from different viewpoints. Training set containing such samples have improved the descriptive abilities of the learned classification model.

The third source of the samples was Google Street View images. All the samples of the 'Other' class have been collected from Google street view API. Images downloaded from Google Street View were manually segregated while labeling. Since the quality of images downloaded from Google Street View API was low, in many samples it was difficult to know whether they are containing phragmites or some other grass. we needed experts help to label these samples efficiently. Phragmites samples downloaded from google street view API were critical for the performance of the proposed application because, at the execution time of the application, the only source of data will be Google street view images. But at the same time finding phragmites samples on Google street view images were a real challenge as it requires manually exploring long highways on the google map.

In table 1 count of samples received from above mentioned sources has been listed.

3.2. Data acquisition using Google Street View API

An auxiliary task of the proposed application was to access images from the Google Street View Image server. Images in Google Street View are in the panorama format. Using Google Street view API [7], a script has been

Table 1: Showing count of the labeled samples taken from respective data Sources.

Source	Phragmaties	Others
Google Images	438	0
Videos frame	1862	0
Google Street View	558	3990

developed to automatically download image samples from Google Street View servers.The size of downloaded images were 600X300. For a given coordinate, script downloaded images from the both sides of the road by adjusting the camera angle parameter. This script also stores associated geo-coordinates of the image so that application can report these geo-coordinates after performing object detection step. For downloading images, the script first generates a set of equidistant geo-coordinates for the entire route between provided start point and end points. Using each geo-coordinate, a static URL was generated for creating a HTTP request which downloads associated image from Google Street View server. A sample of a downloaded image has been shown in figure 2. But the script was unable to generate geo-coordinates for a route if the path between source and destination was not a straight line. In such scenario, the download application also needs geo-coordinates of the turning points and equidistant points on the curves associated with the path so that it can download images for the complete path. Another user-friendly and interactive web application was developed which enabled the user to create a list of geo-coordinates just by clicking on starting, turning and destination points on the Google map as shown in the figure 3. By providing this list of coordinates to the image downloader application, images for a complete path between the given source and destination points can be downloaded.

Figure 2: An image downloaded using Google Street View API

[ht]

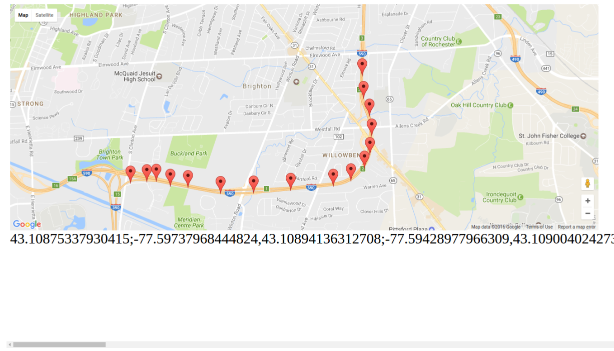


Figure 3: Web Application for selecting geo-coordinates from map.

3.3. Data Augmentation and Preparation

Though I wasn't able to use Faster-R-CNN network for this project, the created training dataset can be prepared for training the publicly available Faster-RCNN implementation [5] by providing labels for region proposals in Annotations files [17]. Creating annotation files can be a very time-consuming process, but recently darrenl tzutalin have developed a Qt based GUI application name LableImg. Using this application annotations files can be created with a faster pace. One can create required annotation files in a weeks time for a dataset containing around 12,000 images.

I was unable to train faster RCNN, professor Kanan suggested me to create an object detection system using transfer learning with VGG-16 convolutional network. Since an image can contain many objects, without having region proposals, we can't localize the target object. So instead of using RPN network, I have cropped each image in multiple equal parts. In my observations, I found most of the times the upper half the images contains trees, advertisement boards, and polls, the lower half contains phragmites colonies, other small plants, trunk of the trees, part of roads and vehicles. For the cropping process, each sample was divided into (length of minor axis)/2x(length of minor axis)/2 size parts. This process created sub-images of same width and height which doesn't lose their aspect ratio while performing re-sizing for CNN pre-processing step. For most of the sample images, this way of cropping created four sub-images of equal size. In order to remove unwanted parts from images downloaded from Google and extracted frames from videos, same cropping process was applied. This cropping process not only substituted the RPN network's working but also performed data augmentation for training purpose because after dividing each sample into multiple parts the training dataset size was improved as shown in table 2. For training and testing of the CNN network, these cropped sub-images have been used.

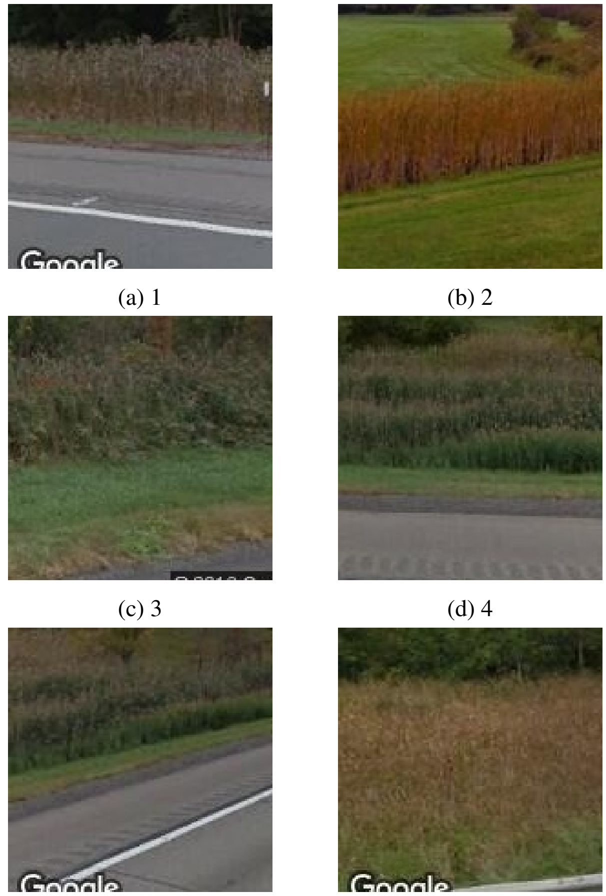
The actual number of samples of phragmites downloaded from Google street view were much higher than reported because for many samples, I was not completely sure whether they were containing phragmites or any other kind of grass, therefore such samples were not included in the dataset. Also, a cropped sub-image was labeled as phragmites only if the phragmites in the sub image was covering more than 35 percent area of it. Samples where phragmites were covering an area less than 35 percent(approx) were not added to the dataset because the cropping process was simply mimicking the RPN network and we are training an object detection system. For example images shown in figure 4 have been included in the dataset because the area covered by phragmites is significant while the samples shown in figure 5 have been excluded from the dataset.

Table 2: count of the labeled samples generated for respective sources after performing cropping operation.

Source	Phragmaties	Others
Google Images	1325	0
Videos frame	5032	0
Google Street View	735	6739

3.4. Training Dataset limitations:

As discussed in section 1, we know that phragmites change their colors and looks during the year but the images acquired from Google street API lacks seasonal diversity as most of the highways in Rochester NY. were covered during the same month of the year. Google map does not provide any additional information using which we can know when a particular road was covered. Also, along with phragmites other objects like grass, trees (lose their leaves after fall) and background(due to snow) change their color, shape and look during the year. Also, for maintaining diversity in the dataset, we can't store samples from two nearby geo-locations. Hence for improving geographic diversity of the dataset, a knowledge of local geography is required, so that we can improve the diversity of the dataset. Again the size of training dataset was only 12,000 images only which is relevantly small for such applications.



(a) 1 (b) 2 (c) 3 (d) 4 (e) 5 (f) 6

Figure 4: Samples included in the dataset

(a) 1 (b) 2

Figure 5: Samples excluded from the dataset

4. Classification Pipeline

In this section classification process of the proposed network has been explained.

4.1. Pre-Processing

The pre-processing step is applicable for both training and testing datasets. It is a good practice to zero centers and normalizes the images before passing to the CNN model [21]. All samples of the dataset were zero centered by offsetting the red, green and blue planes by their respective mean values calculated for the entire dataset. Images were then re-sized to the standard input size of 224X224 before passing to the CNN model. As CNNs are not rotation invariant, we require data augmentation techniques like horizontal flipping. Since any sample of training or testing dataset will not experience rotation in this application so it was not applied. However, data augmentation techniques like inversion and the random crop can boost the accuracy of the classifier.

4.2. Region Proposals

As discussed in section 2, Faster R-CNN model uses RPN network and Fast R-CNN Network for object localization and detection receptively.

Both object detection and localization networks share initial convolution layers as shown in figure 6. These layers work as feature extractor for both RPN and Fast R-CNN network (object classification network). For training the entire network end to end alternating training approach can be used. This approach suggests that, after initializing all the shared layers of the Faster-R-CNN network by weights gained by training respective classification model(VGG-16) on ImageNet dataset, we should train RPN network with a prepared dataset. Using trained RPN network we can train subsequent Fast R-CNN network for classification. Then using this trained Fast R-CNN network we can fine tune the RPN network. Using the parameters gained by fine-tuning, we can re-initialize the RPN network and start training for RPN network again. By performing this iterative training approach, we can improve the accuracy of the entire end to end network[15].

The implementation of Faster R-CNN has been provided by Girshick and it's publicly available [5]. For executing the provided training script, a training dataset should be in a certain format as discussed in section 3.

In order to Conduct end to end training of the Faster-RCNN network,not only requires a large label dataset like VOC segmentation dataset which contains 27000 annotations with 12,000 images, but also requires high computation power of GPUs like Titan, K20, K40. **(Till the end of the semester I wasn't able to build a dataset of this much size and diversity with annotation files [?].**

4.3. Image classification using transfer learning.

As professor Kanan suggested for developing an object detection system using transfer learning with VGG-16 convolutional network, shown in figure 7. The implementation of VGG-16 and parameters(weights) gained after training it with ImageNet dataset has been made publicly available by baraldilorenzo.[1].

After removing the final layer from the pre-trained CNN VGG16 classification model, the residual architecture works as a feature extractor. Last classification layer of pre-trained CNN model generates a feature vector of 4096 dimensional for the pre-processed image input. After extracting feature vectors from the images of the dataset, L2 normalization was performed on these feature vectors and all the negative values in feature vectors were replaced by 0. Subsequently, using a SoftMax classifier model, feature vectors were classified into Phragmites and Other classes.

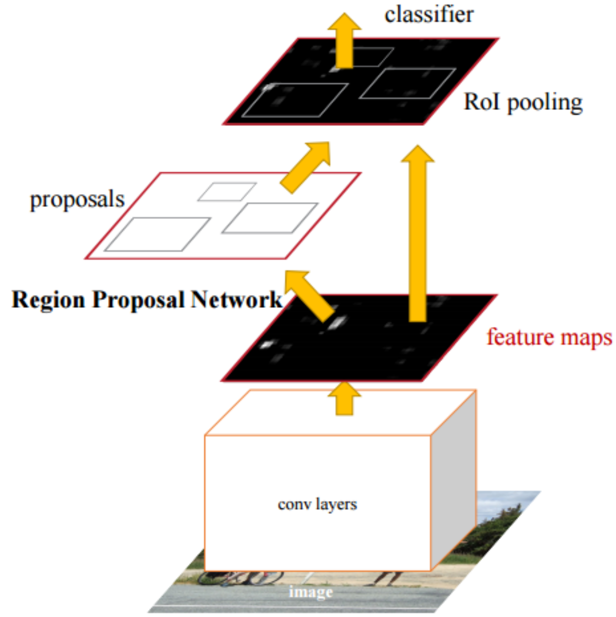


Figure 6: Faster-R-CNN (Image source: [15])

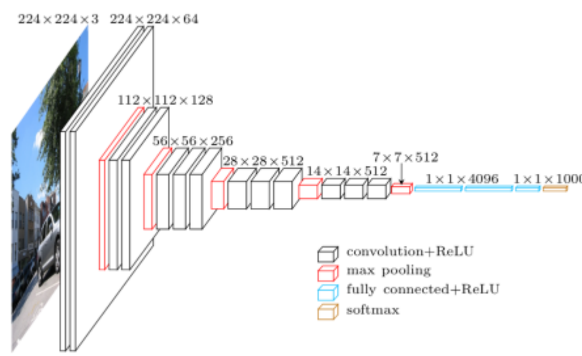


Figure 7: VGG-16 Convolution Neural Network (Image Source:[1])

5. Experiments

The created dataset was divided into training and testing dataset by separating randomly selected 1000 samples. The resultant training dataset contained 12,000 samples of feature vectors. During training the softmax classifier, a constant learning rate of .7 was used. 20 percent of training data was used for validation during each epoch. Training was performed with mini batch stochastic gradient descent algorithm with a batch size of 32. As shown in figure 8 After 120 epochs loss function converged to near zero value.

The model was tested with two different testing datasets.

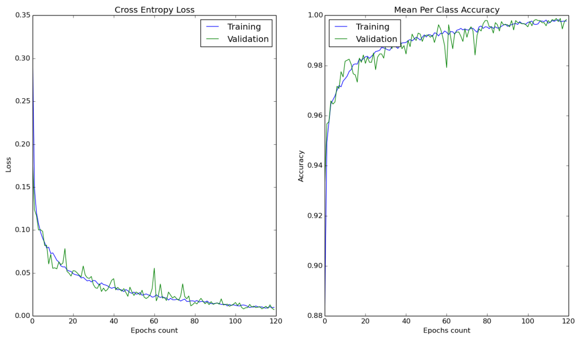


Figure 8: Epochs Vs Training/validation Loss(left) and Epochs Vs Training/validation curve(right).

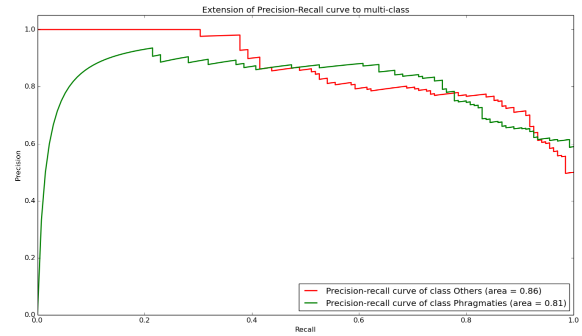


Figure 9: Precision Recall Curve for second test dataset (only containing Google Street Images).

The constructed dataset was partitioned into the training and first testing dataset. This testing dataset was containing 1000 samples. Final accuracy on this test dataset was 0.9832. Since for most of the samples, the source was video frames where adjacent frames can be similar for a video, the accuracy achieved on test dataset was not a good measure of the efficiency of the classifier.

Since the proposed application was designed to work only on Google Street view images, for measuring the accuracy of the classifier, a dataset containing only the Google Street view Images was also used. This second testing dataset was containing total 400 samples of both phragmites and other class in equal ratios. Phragmites images were taken from I-90 highway between Rochester and Syracuse, Andrew Memorial Drive and East River Road. These routes were covered respectively during summer, fall and winter by Google Map team. Images for 'Other' class were collected from different highways. These images were containing images of trees belonging to different weather conditions, different vehicles, roads, advertisement boards, corn fields and grass fields images. The proposed classifier achieved only 77 percent accuracy on this dataset. Confusion matrix has been shown in the table 3 and precision recall curve has been shown in figure 9. The Classifier has achieved .81 average precision for phragmites class for this testing dataset.

Table 3: Confusion Matrix for the testing dataset which contains only Google Street view Images.

class	Phragmaties	Others
Phragmaties	141	72
Others	21	166

6. Conclusion

The classifier didn't perform well on Google street view image testing dataset. Possible reason for low accuracy could be the low-resolution quality of the Google street view images which became further lower for the images where phragmites colonies were far away from the camera. Also, there were less number of Google street view phragmites images in the training dataset. improving the number of samples of Google street view image in training dataset, might result in increase of overall accuracy on the test dataset.

In my understanding for achieving over all higher accuracy, we should use faster R-CNN network for generating region proposals and object detection task. For creating training dataset the faster R-CNN LabelImg application can be used for preparing annotation data files [3] [5].

References

[1] baraldilorenzo. Vgg16 model for keras. URL `https://gist.github.com/baraldilorenzo/8d096f48a1be4a2d660d`.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. doi: 10.1109/CVPR.2005.177.

[3] darrenl tzutalin. Labelimg is a graphical image annotation tool and label object bounding boxes in images. URL `https://github.com/tzutalin/labelImg`.

[4] Image Downloader. Fatkun batch download image. URL `https://chrome.google.com/webstore/detail/fatkun-batch-download-ima/nnjjahlikiabnchcpehcpkdeckfgnohf?hl=en`.

[5] Ross Girshick. Faster r-cnn (python implementation). URL https://github.com/rbgirshick/py-faster-rcnn.

[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. URL http://arxiv.org/abs/1311.2524.

[7] Google. Google maps api. URL https://developers.google.com/maps/documentation/javascript/streetview.

[8] FFmpeg Software group. Ffmpeg. URL https://ffmpeg.org/.

[9] Corey L Gucker. Phragmites australis. in: Fire effects information system, [online]. u.s. department of agriculture, forest service, rocky mountain research station, 2008. URL http://www.fs.fed.us/database/feis/plants/graminoid/phraus/all.html.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. Deep-plant: Plant identification with convolutional neural networks. *CoRR*, abs/1506.08425, 2015. URL http://arxiv.org/abs/1506.08425.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[13] Mariann Marks. Element stewardship abstract for phragmites australis. URL http://www.nps.gov/plants/alien/fact/phau1.htm.

[14] Pinterest. Phragmities affected states. URL https://in.pinterest.com/pin/279997301805477221/.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. 2015.

[16] Jorge Sanchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 105(3):222–245, December 2013. doi: 10.1007/s11263-013-0636-x. URL https://hal.inria.fr/hal-00830491.

[17] SGSAI. Training faster r-cnn on custom dataset. URL http://sgsai.blogspot.com/2016/02/training-faster-r-cnn-on-custom-dataset.html.

[18] B. Slattery K. Reshetiloff Swearingen, J. and S. Zwicker. Plant invaders of mid-atlantic natural areas, 4th ed national park service and u.s. fish and wildlife service. washington, dc. 168pp, 2011.

[19] Cornell University. Ecology and management of invasive plants program, phragmites australis, 2011. URL http://www.invasiveplants.net/monitor/9CommonReed.aspx.

[20] Jan D. Wegner, Steven Branson, David Hall, Konrad Schindler, and Pietro Perona. Cataloging public objects using aerial and street-level images - urban trees. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] Xiu Shen Wei. Must know tips/tricks in deep neural networks. URL http://lamda.nju.edu.cn/weixs/project/CNNTricks/CNNTricks.html.

[22] T. Zeng and S. Ji. Deep convolutional neural networks for multi-instance multi-task learning. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 579–588, Nov 2015. doi: 10.1109/ICDM.2015.92.