Anomaly Detection using One Class SVM

Harpreet Singh
Autonomous Robots Lab
harpreets@nevada.unr.edu

July 3, 2018

1 Introduction

Anomaly detection is the problem of correctly classifying abnormal data given a large set of normal data examples. In this project, images of the normal state of an environment represent the normal data, target class, while images containing objects uncharacteristic to the environment represent abnormal data, outlier class. The difference between a two-class classifier and anomaly detection is that data of only the target class is available during training. This means that an anomaly detection classifier, a one class classifier, must learn the target class and then be able to classify similar and outlier examples.

While other methods for one class classification exist, this project used one-class SVMs, a statistical method that learns a decision boundary encapsulating the target class data. The primary advantage of this technique is that it does not rely on the probability density estimation of the target data; this is useful when only sparse target data is available or when one does not know the normal and abnormal states of the environment [Tax(2001)].

A disadvantage of the one-class SVM technique with image data is the sensitivity to the features representing the images; features that change due to illumination or viewpoint will cause misclassification. For this reason, this project tested two different techniques to extract image features: visual bag of features and deep features. Furthermore, deep features were fine-tuned for the one class classification task by further training a pre-trained deep network to reduce the variance of the target class image features.

1.1 One-Class SVM

There are two different formulations of a one-class SVM: Support Vector Data Descriptor(SVDD) and v support vector classifier(v-SVC). Both are briefly described below.

SVDD constructs a boundary around the target dataset, a hypersphere that encapsulates all of the target data examples. The objective of SVDD is to minimize the volume of the hypersphere in order to detect outliers. The following figure shows a simple, 2D case of SVDD.

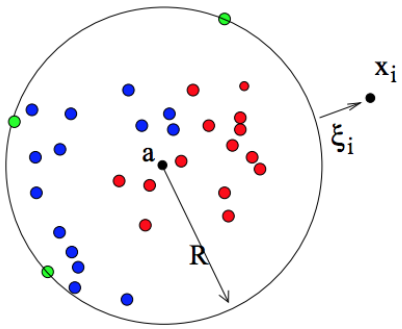

Figure 1: A 2D sphere encapsulating the training data. R is the radius, a the center, and ξ_i the non-zero slack variable of an example. [Tax(2001)]

SVDD is defined as a constrained optimization problem. The objective is to minimize the radius of the hypersphere with the addition of slack variables, to account for outliers in the training data:

$$\epsilon(R, a, \xi) = R^2 + C \sum_i \xi_i \tag{1}$$

The objective is constrained with the condition that almost all training examples are within the hypersphere:

$$\|x_i - a\|^2 \leq R^2 + \xi_i,\ \xi_i > 0,\ \forall i \tag{2}$$

SVDD's optimization problem is solved using Lagrange multipliers. The full derivation along with a modified SVDD formulation for outlier examples in training can be found at [Tax(2001)].

V-SVC is a different implementation of a one-class SVM; v-SVC constructs a hyperplane that separates the training data from the origin as far as possible.

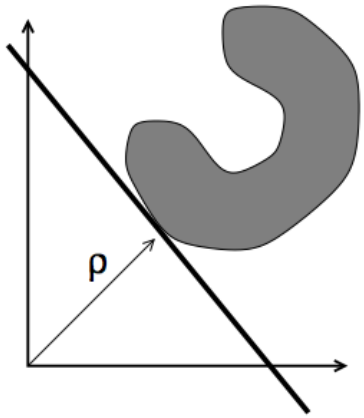

Figure 2: Example ν-SVC goal. [Tax(2001)]

The optimization problem for v-SVC is:

$$\max(\rho - \frac{1}{\nu N} \sum_i \xi_i) \tag{3}$$

where ρ is the distance from the origin to the separating hyperplane, w, and ν is the regularization parameter $\in (0, 1)$ that represents the fraction of training error. Equation 3 is subject to the following constraints:

$$w \cdot x_i \geq \rho - \xi_i,\ \xi_i > 0,\ \|w\| = 1 \tag{4}$$

1.2 Image Feature Extraction

In visual anomaly detection, images have to be mapped to the feature space. In this project, two methods were tested: visual bag of features and deep features.

Visual bag of features is a classical computer vision technique to quantize an image based on common visual features in the training set [Csurka et al.(2004)Csurka, Dance, Fan, Willamowski, and Bray]. The first step in the process is to extract feature descriptors in a grid pattern from each image in the training set. Then find k clusters, which represent the size of visual words, in the combined set of features of each image. Finally, construct a normalized histogram for each image in the training set using the extracted features in the image and k-clusters; this normalized histogram is the quantized feature vector representing the image. Histograms of similar images will result in feature vectors closes together.

Deep features are those that are learned by a deep network. Most deep networks contain fully connected layers before the final output layer. The output of the fully connected layers are the features that the network has learned and uses for the task. Deep convolutional networks are very effective in image classification tasks because of the discriminative features they learn. As such, pre-trained deep networks can be used for the anomaly detection problem to extract discriminative features. [Perera and M. Patel(2018)] describes a process to further improve one class deep features by fine-tuning a pre-trained network to reduce the variance of the target class features, making them more compact.

2 Implementation

The project is implemented in Python with the following libraries: Keras for deep networks, Scikit-Learn for the one class SVM classifier, and OpenCV for the visual bag of features implementation.

OpenCV provides two data structures to support visual bag of features: one to perform clustering with the feature descriptors from the images and another descriptor mapper to perform image vector quantization. In this project, SURF features were utilized to construct the bag of words. For all of the tests ran, 2048 bag of words/clusters were selected and SURF feature descriptors were computed in a 16x16 image patch across the entire image.

Keras provides common deep network architectures along with pre-trained weights through a simple functional API. Three architectures were tested: ResNet50, VGG16, and InceptionV3. These architectures were chosen based on their performance on the ImageNet challenge as shown in Table 1. High accuracy on the ImageNet challenge portrays discriminative capabilities of the network's learned features. Keras was also used to implement deep one class feature tuning [Perera and M. Patel(2018)]; its flexible functional API allows customization of the training process.

Network	Parameters	ImageNet Accuracy
VGG 16	138,357,544	71.5%
ResNet 50	25,636,712	75.9
Inception V3	23,851,784	78.8%

Table 1: Stats on the deep networks used in the project. ImageNet accuracy refers to the classification accuracy achieved on the ImageNet challenge. [Keras()]

Since the architectures of the three deep networks differ, the dimensionality of the features varies. The following table shows the feature dimensionality based on the extraction mechanism:

Feature Extraction Mechanism	Feature Dimensionality
VGG 16	4096
ResNet 50	2048
Inception V3	2048
Bag of Features	2048

Table 2: This table shows the number of dimensions of the feature vector.

LibSVM, a popular SVM library, is the underlying library that Scikit-learn' SVM package utilizes; LibSVM provides an implementation of the ν-SVC one class classifier. The only modifications made by Scikit-learn are related to data structure changes to accommodate for data transfers between Scikit-learn's and LibSVM's API and a sparse dataset version of SVM.

The implementation runs in two phases: the training phase and the testing phase. During the training phase, image features of the entire training dataset is extracted and then a one class SVM classifier is trained. In the testing phase, a subset of the training images, a new set of target class images, and a set of outlier images are classified.

The code for the project is available on Github [Singh(2018)].

3 Results

The initial dataset consisted of images of a dessert terrain with various bushes. Outlier images contain bright colored objects different from the environment's tone.

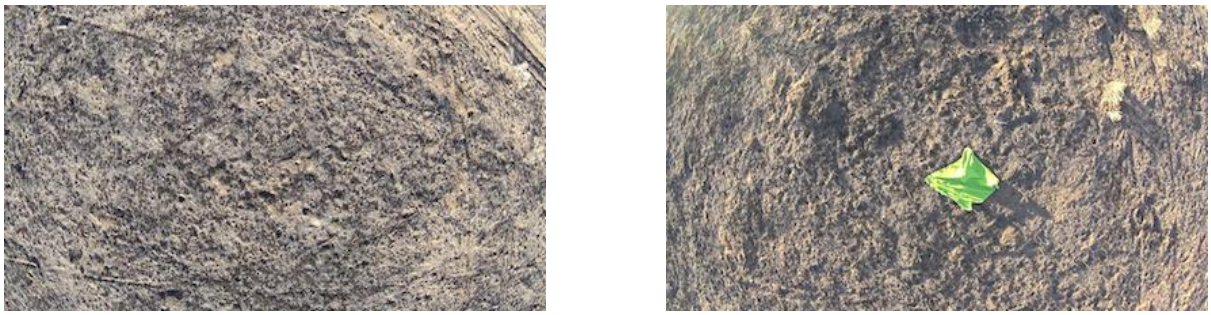

Figure 3: The image to the left shows an example of the normal state of the environment and the right shows the environment with an outlier.

Data Subset	Count
Training Set (all target class)	1220
Target Testing Set	268
Outlier Testing Set	164

Table 3: Size of each subset of the initial dataset

Figure 3 shows an example inlier and outlier image and Table 3 shows the number of images in each testing subset. The one class SVM was configured with $\nu = 0.1$ and an RBF kernel with the gamma set to 0.1. The results from these runs are presented below:

Data Subset	VGG16	InceptionV3	BagOfFeatures	ResNet50
Training Set Accuracy	40.00%	73.77%	83.60%	47.95%
Target Testing Set Accuracy	36.94%	76.49%	86.89%	50.37%
Outlier Testing Set Accuracy	100.00%	100.00%	47.56%	100.00%
Number of Support Vectors	1220	1217	127	1220

Table 4: Accuracies of anomaly detection using different sources of features. The entire training set was evaluated after the classifier was trained; Target testing set consists of images not in the training set. InceptionV3 achieves the highest accuracy and least number of support vectors from the deep networks, albeit by a small margin. While Bag of Features achieved the overall highest accuracy on the target class sets, it did not perform well in the outlier class set.

Results show that Inception V3 achieved the highest accuracy in the target class set with the least number of support vectors among the deep networks utilized. Bag of Features achieved the highest overall accuracy with the target class set and the lowest number of support vectors but performed very poorly on the outlier dataset.

In regards to the deep networks, the primary issue recognized from this run was the number of support vectors: all or nearly all of the training data was a support vector even with a low gamma value of the RBF kernel. And while the Bag of Features resulted in a small number of support vectors, its performance on detecting outliers is very low compared to the other methods. Furthermore, the uniformity and the lack of features in the dataset may have also caused the low accuracies. In order to alleviate these issues, the size of the dataset was increased as well as a new site was chosen. The following figure shows sample images from the new site:

Figure 4: The image to the left shows an example of the normal state of the environment and the right shows the environment with an outlier.

Data Subset	Count
Training Set (all target class)	21273
Target Testing Set	5000
Outlier Testing Set	3145

Table 5: Size of each subset of the new dataset

All the tests were rerun and the following table shows the results:

Data Subset	VGG16	InceptionV3	BagOfFeatures	ResNet50
Training Set Accuracy	1.06%	53.88%	90.98%	45.62%
Target Testing Set Accuracy	3.12%	26.96%	89.58%	21.60%
Outlier Testing Set Accuracy	100.00%	100.00%	11.73%	100.00%
Number of Support Vectors	17403	16365	2136	17508

Table 6: Similar to the initial results, InceptionV3 achieves the highest accuracy and least number of support vectors from the deep networks. Bag of Features also performed similarly.

Results from the new dataset show that performance for all of the deep network features decreased. Most notably, VGG16 performed significantly worse and Inception V3 still achieved the highest accuracy among the deep features. Bag of Features performed consistently by achieving high accuracy on the target class dataset and low accuracy on the outlier set.

After the results from the new site dataset, only results with the Bag of Features and InceptionV3 were attempted to be improved. Since the performance from the Bag of Features was only lagging with classifying outlier data, the gamma value of the RBF kernel was increased in order to increase the variance of the decision boundary and allow the SVM to overfit. The results are provided below:

Dataset\Gamma	0.1	0.5	0.9
Training Set Accuracy	90.98%	91.18%	91.18%
Target Testing Set Accuracy	89.58%	89.64%	89.62%
Outlier Testing Set Accuracy	11.73%	11.67%	11.67%
Number of Support Vectors	2136	2163	2164

Table 7: Accuracy of classification with different gamma parameter of the RBF kernel using the Bag of Features. As gamma increases, the SVM would overfit.

Table 3 shows results using the BOF with varying gamma. Since the results do not improve, they seem to indicate that the features of outlier images are very similar to the target dataset, causing misclassification. The features extracted via BOF need to be improved in some manner. No further methods to improve this mechanism were studied.

To further improve results from InceptionV3, the pre-traineed network was tuned specifically for one class classification as proposed by [Perera and M. Patel(2018)]. The main objective of the fine-tuning is to maintain the discriminative capability of the learned features while decreasing the variance. The following figure shows the proposed training model:

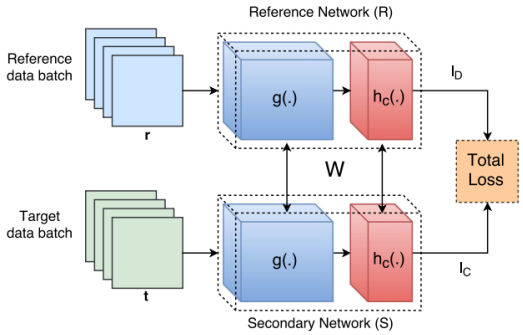

Figure 5: The reference network is original pre-trained model that computes a cross-entropy loss based on a dataset it was trained on and is not updated; the secondary model is trained with the one class image data and computes the a loss for the variance of the deep features at each batch. The weights in the dotted box are shared between the models. As the secondary model optimizes on variance, the reference network assures descriptiveness of the features, and the total loss sums the two individual losses.

The InceptionV3 secondary network was tuned with DOC using the new site training data; the reference network was evaluated with the validation set from the 2012 ImageNet challenge. The following figure shows the total training loss over 2 epochs with a batch size of 64:

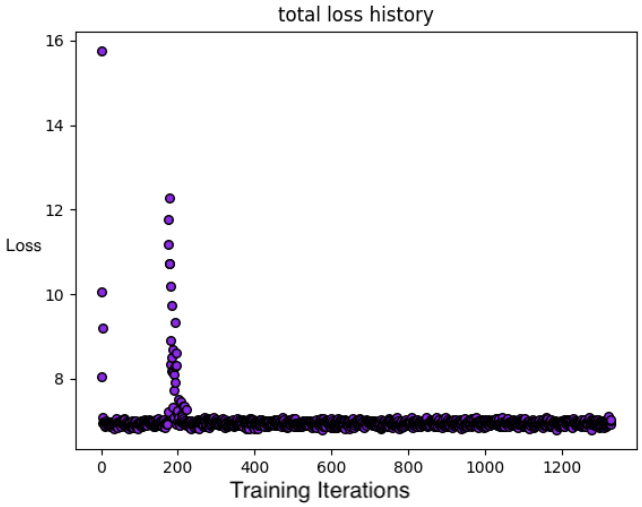

Figure 6: The total loss quickly converges as noted in [Perera and M. Patel(2018)].

The results of the deep one class tuned InceptionV3 are presented below:

Data Subset	Pre-trained Inception V3	DOC InceptionV3
Training Set Accuracy	53.88%	63.50%
Target Testing Set Accuracy	26.96%	64..80%
Outlier Testing Set Accuracy	100.00%	58.72%
Number of Support Vectors	16365	2128

Table 8: Results of the DOC tuned InceptionV3 network compared with the pre-trained InceptionV3 network.

The tuned InceptionV3 network has significantly smaller number of support vectors and a higher accuracy on the target class datasets. Although, the accuracy on outlier dataset has decreased, the results show that it is possible to improve the learned deep features for one-class classification problems. No further methods to improve these results were studied.

4 Future Work

A different approach to this problem may be utilizing semantic segmentation and performing anomaly detection at the object level rather than feature level. This may be done by combining image segmentation deep networks with a relational network [Santoro et al.(2017)Santoro, Raposo, Barrett, Malinowski, Pascanu, Battaglia, a

References

[Tax(2001)] David Tax. One-class classification; concept-learning in the absence of counter-examples. 01 2001.

[Csurka et al.(2004)Csurka, Dance, Fan, Willamowski, and Bray] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[Perera and M. Patel(2018)] Pramuditha Perera and Vishal M. Patel. Learning deep features for one-class classification. 01 2018.

[Keras()] Keras. Keras applications. `https://keras.io/applications/`. Accessed: 2018-07-02.

[Singh(2018)] Harpreet Singh. One class svm study. `https://github.com/harpreets652/anomaly-svm-study`, 2018.

[Santoro et al.(2017)Santoro, Raposo, Barrett, Malinowski, Pascanu, Battaglia, and Lillicrap] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.