2014

DEEP LEARNING METHODS FOR MULTIBAND EXPLOSIVE HAZARD DETECTION USING L-BAND AND X-BAND FORWARD-LOOKING GROUND-PENETRATING RADAR

John T. Becker
Michigan Technological University

Recommended Citation

Becker, John T., "DEEP LEARNING METHODS FOR MULTIBAND EXPLOSIVE HAZARD DETECTION USING L-BAND AND X-BAND FORWARD-LOOKING GROUND-PENETRATING RADAR", Master's Thesis, Michigan Technological University, 2014.
https://digitalcommons.mtu.edu/etds/968

DEEP LEARNING METHODS FOR MULTIBAND EXPLOSIVE HAZARD DETECTION USING L-BAND AND X-BAND FORWARD-LOOKING GROUND-PENETRATING RADAR

By

John T. Becker

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2014

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical Engineering.

Department of Electrical and Computer Engineering

Thesis Advisor:	*Dr. Timothy Havens*
Committee Member:	*Dr. Michael Roggemann*
Committee Member:	*Dr. Laura Brown*
Department Chair:	*Dr. Daniel Fuhrmann*

Contents

List of Figures

List of Tables

Important Acronyms and Abbreviations:

UTM: universal transverse mercator

FLGPR: forward-looking groud-penetrating radar

CFAR: constant false alarm rate

FA: false alarm

ROC: recievcer operating characteristic

DOM: difference of means

Bhatt: Bhattacharya

AUR: area under ROC

DBN: deep belief network

SDAE: stacked denoising autoencoder

CNN: convolutional neural network

NN: neural network

SVM: support vector machine

SKSVM: single kernel support vector machine

RBF: radial basis function

MKLGL: multiple kernel learning group lasso

HOG: histogram of ordered gradients

LBP: local binary pattern

LSTAT: local statistics

FFST: fast finite shearlet transform

Preface

Several papers were written as a result of this research, they are:

Havens, T. C., Becker, J., Pinar, A., and Schulz, T. J. “Multi-band sensor-fused explosive hazards detection in forward-looking ground-penetrating radar.“ *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014.

Contributions:

J. Becker and A. Pinar - Data Analysis and Testing

T. C. Havens and T. J. Schulz - Image Formation and Preprocessing

T. C. Havens - Wrote Paper

J. Becker - Editting and Advising

Becker, J., Havens, T. C., Pinar, A., and Schulz, T. J. ”Deep Belief Networks for False Alarm Rejection in Forward-Looking Ground-Penetrating Radar.“ Abstract accepted for *SPIE Defense+ Security.* International Society for Optics and Photonics, 2015.

Contributions:

J. Becker and A. Pinar - Data Analysis and Testing

T. C. Havens and T. J. Schulz - Image Formation and Preprocessing

J. Becker - Wrote Paper

T. C. Havens - Editting and Advising

Becker, J., Havens, T. C., Pinar, A., and Schulz, T. J. ”Stacked Denoising Autoencdoers for Explosive Hazard Mitigation in Sensor-Fused Forward-Looking Ground-Penetrating Radar.” Awaiting review for *Geoscience and Remote Sensing, IEEE Transactions on,*

Contributions:

J. Becker and A. Pinar - Data Analysis and Testing

T. C. Havens and T. J. Schulz - Image Formation and Preprocessing

J. Becker - Wrote Paper

T. C. Havens - Editting and Advising

Acknowledgments

I would like to thank my friends and family for their incredible support and my advisor for his patience and guidance.

This work has been funded in part by U.S. Army grants W909MY-13-C-0013 and W909MY-13-C-0029. We would like to acknowledge U.S. Army RDECOM CERDEC NVESD for vital support in this project.

Superior, a high performance computing cluster at Michigan Technological University, was used in obtaining results presented in this publication.

Abstract

Explosive hazards are one of the most deadly threats in modern conflicts. The U.S. Army is interested in a reliable way to detect these hazards at range. A promising way of accomplishing this task is using a *forward-looking ground-penetrating radar* (FLGPR) system. Recently, the Army has been testing a system that utilizes both L-band and X-band radar arrays on a vehicle mounted platform. Using data from this system, we sought to improve the performance of a *constant false-alarm-rate* (CFAR) prescreener through the use of three deep learning architechtures; *deep belief networks* (DBNs), *stacked denoising autoencoders* (SDAEs), and *convolutional neural networks* (CNNs). We also compare these deep learning classifiers with two more conventional shallow learning classifiers; *single kernel support vector machines* (SKSVMs) and *multiple kernel learning group lasso* (MKLGL). By training the deep learners on a combination of image features and comparing the test results to the conventional shallow learners, we were able to significantly increase the probability of detection over both the CFAR prescreener and the shallow learners while maintaining a nominal number of false alarms per square meter. Our research shows that deep learners are a good candidate for improving detection rates in FLGPR systems.

Chapter 1

Introduction

1.1 Introduction

An important goal for the U.S. Army is remediating the threats of explosive hazards as these devices cause uncountable deaths and injuries to both civilians and soldiers throughout the world. Since 2008, explosive hazard attacks in Afghanistan have wounded or killed nearly 10,000 U.S. Soldiers; worldwide, explosive devices on average cause 310 deaths

---

The material contained in this chapter was either previously published in *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014 *OR* has been submitted to the *Geospace and Remote Sensing, IEEE Transactions on*, 2015.

and 833 wounded per month [1]. Systems that detect these threats have included *ground-penetrating-radar* (GPR), *infrared* (IR) and visible-spectrum cameras, and acoustic technologies [2, 3, 4]. Handheld and vehicle-mounted GPR-based systems have been the subject of recent research and great progress has been made in increasing detection capabilities [5, 6]. The ability of *Forward-looking* synthetic aperture GPR (FLGPR) to detect hazards before they are encountered makes these systems especially attractive; allowing standoff distances to range in to tens of meters. FLGPR systems have been applied to the detection of surface, side-attack, and buried devices [7, 8, 9]. An unfortunate drawback of FLGPR systems is that in addition to being sensitive to explosive devices, UXO, and landmines, they are also sensitive to other objects, both above and below the ground. Because FLGPR is a standoff sensor, the area being examined for targets is much larger than with downward-looking systems; thus, clutter is a serious concern. Furthermore, the explosive hazard threat is very diverse—they are made from many different materials, including wood, plastic, and metal, and come in many different shapes and sizes. This threat also continues to evolve. Hence, it is nearly impossible to detect explosive hazards solely by a modeling-based approach. We have shown in previous work that if *forward-looking infrared* (FLIR) or visible-spectrum imagery is combined with L-band FLGPR, false alarm rates can be reduced significantly [10, 11, 12, 13, 14]. We also demonstrated that fusing multiple sub-bands and spectral features in L-band FLGPR improves detection performance [15, 16, 17]. Therefore, in this thesis, we will extend this line of sensor fusion research by examining methods for fusion of multiple bands of FLGPR.

Recently, the U.S. Army *Night Vision and Electronic Sensors Directorate* (NVESD) has been working with an integrated L-band and X-band FLGPR system. Our focus was on the developement of computer-aided classification algorithms for this dual-band system. We explored several approaches that use both L-band and X-band FLGPR to improve detection statistics. These approaches include *support vector machines* (SVM), *multiple kernel learning* (MKL), and a variety of deep learning architechures which will be the focus of this document. Deep learning is a relatively new appraoch to the explosive hazard detection problem [22, 32]; however, they have been shown to achieve desirable results. With that in mind, we sought to examine several different architechures to both see what we could achieve with different learners and also to provide a simple survey of deep learning architechures, and their relative performance on this particular data set. The three basic deep learners that we implemented were the *deep belief network* (DBN), the *stacked denoising autoencoder* (SDAE), and the *convolutional neural network* (CNN). To achieve a more complete understanding of the performance of these classifiers, we also compare them to so-called shallow learners such as the SVM and the MKL. In doing this we hope to achieve a better understanding of the effectiveness of deep learning algorithms and what, if any, benefits they might provide over more traditional approaches.

1.1.1 Radar System and Data

The radar systems used to collect these data were an L-Band *Multiple-Input-Multiple-Output* (MIMO) radar and an X-band radar. The L-Band MIMO radar was capable of operating in both *dual-polarization* (dualpol) and *all-polarization* (allpol) mode. Dualpol ony collects the HH and VV polarizations, allpol additionally collects the HV and VH cross polarizations as well. For this paper, we will be primarily concerned with the dualpol mode. The X-band radar only collects in the VV polarization.

The X- and L-band FLGPRs operate as stepped-frequency arrays—each transmitter individually illuminates the scene and all receivers then measure the complex return at each frequency and polarization, repeating for each transmitter. The parameters of the radars are shown in Table 1.1. The *government-furnished data* (GFD) are represented as I/Q values at each frequency for each *transmit/receive* (T/R) pair, GPS location, and pitch, roll, and yaw of the array. From these data we are able to localize each T/R pair in 3D *universal transverse mercator* (UTM) coordinates, allowing for fully motion-compensated imaging (within the error of the platform motion estimation).

Table 1.1
FLGPR Parameters[24] (see reprint permissions in Appendix A)

	L-band	**X-band**
Waveform	Stepped frequency	Stepped frequency
Transmitters	8	32
Receivers	8	4
Bandwidth	0.5–3.4 GHz	8.4–10.4 GHz
# Frequencies	2702	1024
Pulse rate	12 Hz	50 Hz
Polarizations*	HH, VV, HV, VH	VV

*Note that we only have HH and VV polarizations for the L-band data sets used in this paper.

We use data collected from four different lanes in this work. Each lane is between 300 and 600 meters in length and about 10 meters in width. These lanes vary from each other in the types of targets buried, the depth of the targets and the soil composition of the lane. The data were collected as follows. First, the targets were buried and the exact GPS coordinates of each target recorded. Later, the test platform traversed the lane first in one direction (either north or east), then reversed direction. We label the lanes used here as lanes A, B, C, and D. For lanes A and B, we used the northbound data and for lanes C and D we use the eastbound data. Lane A contained a total of 38 targets, lane B had 94 targets, lane C had 38 targets and lane D contained 35 targets. The overall goal of this research is to get

Table 1.2
Symbols Used in This Paper [24] (see reprint permissions in Appendix A)

Symbol	Description
$\mathbf{x}(t_{GPS})$	Position of vehicle in UTM at time t_{GPS}
$\mathbf{v}(t_{GPS})$	Velocity (m/s) of vehicle at time t_{GPS}
$\mathbf{x}_j(t_{GPS})$	Position of jth antenna element at time t_{GPS}
$\mathbf{w}_{jk}(f, t_{GPS})$	I/Q signal of jkth T/R pair at time t_{GPS} and frequency f
$a_w(f)$	frequency-domain window (Hamming)
$a_r(j, k)$	aperture window
c	speed of light, 2.998×10^8 (m/s)

the radar system to detect as many of the targets as possible while detecting as few false positives as possible. The aim of this research is to improve the detection statistics. By using varied lanes and target types, we hope to provide a more generalized solution to the explosive hazard detection problem.

Chapter 2

Image Formation and Preprocessing

2.1 Image Formation and Preprocessing

Using the provided radar data, images were formed using a simple backpropagation algorithm. The Superior Computing Cluster was used to run multiple lanes simultaneously.

The material contained in this chapter was either previously published in *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014 *OR* has been submitted to the *Geospace and Remote Sensing, IEEE Transactions on*, 2015.

2.1.1 Backpropagation

The radar images, denoted as $I_p(u, v)$, where p is the polarization and (u, v) are the horizontal and vertical UTM coordinates of the image are formed by a backpropagation procedure as illustrated in Fig. 2.1. The steps of this process are as follows:

1. Remove self-interference by subtracting a windowed time-average of $\mathbf{w}_{jk}(f, t_{GPS})$ over the variable t_{GPS} at each frequency f and for each T/R pair.

2. For each frame (as indicated by t_{GPS}) and polarization:

    (a) Inverse Fourier transform the zero-padded (up-sampled), windowed signals

    $$a_w(f)\mathbf{w}_{jk}(f, t_{GPS}),$$

    where $a_w(f)$ is a Hamming window, producing the range/time signals $\mathbf{r}_{jk}(t, t_{GPS})$.

    (b) Interpolate and coherently integrate the windowed range signals $a_r(j, k)r_{jk}(t, t_{GPS})$ onto the predetermined grid (u, v), and apply the amplitude and phase correction

    $$r^2 \exp\left\{-i4\pi f_1 t\right\},$$

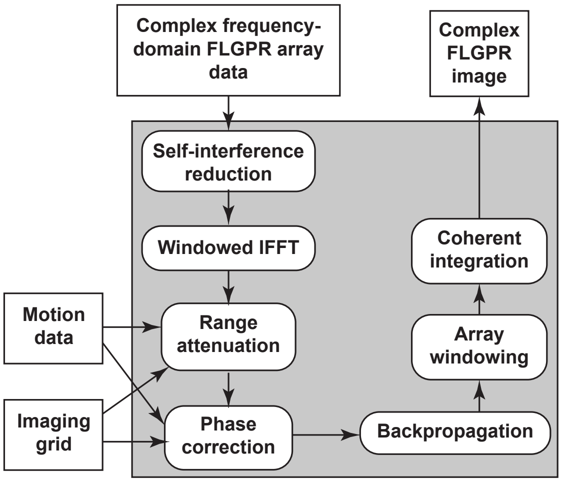


Figure 2.1: Block diagram of FLGPR backpropagation-based imaging algorithm.[24] (see reprint permissions in Appendix A)

> where f_1 is the lowest frequency in the stepped frequency transmission. Note that the windowed range signals are only interpolated and integrated onto grid points that are between r_{min} and r_{max} range in front of the array (we use $r_{min} = 5$ and $r_{max} = 10$ meters for most of the results in this report).

We used a grid spacing of 2.5cm in both cross-range and down-range and upsample $\mathbf{w}$ by a factor of 16 (to the nearest power of 2). This form of backpropagation is the most basic synthetic aperture radar imaging method. We also experimented with a phase correction to compensate for the motion of the vehicle. The self interference reduction and phase correction are discussed in the following sections.

2.1.1.1 Self-Interference Reduction

A known problem in MIMO radars is self-interference. This self-interference causes spatial correlation in the receiver, which in turn lowers the systems capacity for performance [33]. This correlation can come from several factors such as improper antenna spacing, small scattering angles, and angle of arrival [33]. Self-interference can be corrected for in a signal processing approach called self-interference reduction (SIR). If we assume we have a radar image $I(x, y, t)$, can apply SIR as

$$I(x, y, t) = I(x, y, t) - \frac{\sum\limits_{t} w(t) I(x, y, t)}{\sum\limits_{t} w(t)}. \tag{2.1}$$

where $w(t)$ is a window function. Currently, we are using $w(t) = 1$ as our window. This esentially gives us a time average as our method of SIR; however, other window functions are possible, such as

$$w(t) \begin{cases} tri(t/T) & \text{if } |t - t_c| < T \\ 0 & \text{else} \end{cases}, \tag{2.2}$$

which is the triangle function over a certain time range. It is also possible to make the SIR causal by only windowing over prior frames. For our approach, however, we assume a full time average is sufficient.

2.1.1.2 Phase Correction

The Akela L-band radar produces wideband FLGPR data by transmitting each frequency individually. Due to platform motion, the pulses experience a time-dependent phase shift. The phase of each pulse at the receiver from a target located at $\mathbf{x} = (x, y, z)$ is

$$\phi(t) = k_f(R_{tx}(t) + R_{rx}(t)), \tag{2.3}$$

$$R_{tx}(t) = \|\mathbf{x} - \mathbf{x}_{tx}(t)\|_2, \tag{2.4}$$

$$R_{rx}(t) = \|\mathbf{x} - \mathbf{x}_{rx}(t)\|_2, \tag{2.5}$$

where k_f is the wavenumber of frequency f, and $\mathbf{x}_{tx}(t)$ and $\mathbf{x}_{rx}(t)$ are the locations of the transmitter and receiver at time t. If we assume that the vehicle during a frame is only moving in the positive x direction then R_{tx} (and, similarly, R_{rx}) can be written as

$$R_{tx}(t_b) = \left[(x - x_{tx}(t_b) - v\Delta t_s)^2 + (y - y_{tx})^2 + (z - z_{tx})^2\right]^{1/2},$$

where t_b is the time at the beginning of the frame, v is the velocity of the vehicle, and Δt_s is the time between t_b and the sth frequency pulse. If we assume that frequency pulses are spaced equally in time, then $\Delta t_s = s\Delta t$, where $s = 0, 1, \ldots, F - 1$ is the index of the frequency pulse and Δt is the transmit time of each pulse.

We simulated the effect of the velocity of the vehicle versus the frame rate of FLGPR to see what velocity-to-frame-rate ratios significantly degraded the conventional backpropagation image. One can think of the velocity-to-frame-rate ratio as the distance the vehicle travels during one sweep of all the frequencies. Figure 2.2 shows images of point targets located at (down range, cross range, height) = $(0, 10, 0)$, $(3, 10, 0)$, and $(-4, 7, 0)$ meters. The array geometry is equivalent to the geometry of the actual L-band FLGPR. As these images show, the image of the point targets degrades significantly at velocity-to-frame-rate ratios greater than 0.25. Furthermore, the maximum return at 0 m/s velocity is more than 6dB greater than the return at a velocity-to-frame-rate ratio of ~ 0.42. In terms of signal-to-noise ratio, this is equivalent to a 6dB loss in SNR. Although the simulated noise is low in these simulated images, one could imagine that a 6dB loss in SNR for the real system could be catastrophic. Based on the data that we have currently, the velocity-to-frame-rate ratios are about 0.1 to 0.2 (corresponding to speeds of about 4-5 mph), which we do not expect to cause a large amount of blurring in the image. But if speeds exceeding 10mph are desired, then motion during the swept frequency pulses can cause significant degradation in image quality.

Even though the real platform velocity to frame rate ratio is very small, we examined a way to compensate for the phase error caused by the platform motion. We first estimate the ranges as $R_{tx}(t)$ and $R_{rx}(t)$ as $R_{tx}(t_s) \approx R_{tx}(t_b) - v\Delta t_s$ and $R_{rx}(t_s) \approx R_{rx}(t_b) - v\Delta t_s$ and then apply a phase compensation to the swept frequency signal as

$$w'_{t_b}(f_s) = w_{t_b}(f_s) \exp\left\{-j4\pi \frac{f_s}{c} v(t_b)\Delta t_s\right\},$$

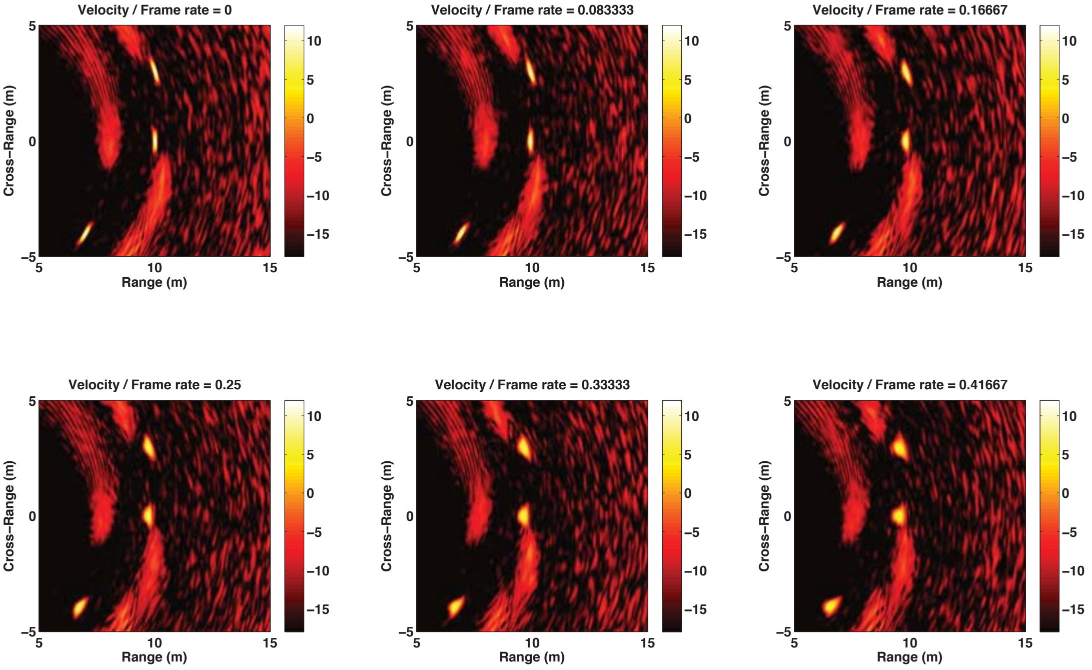
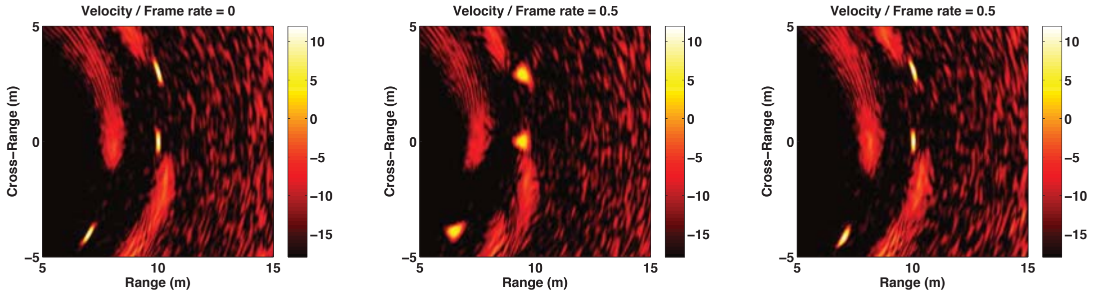


Figure 2.2: Effect of platform velocity and frame rate on simulated backpropagation image of L-Band FLGPR

where $w_{t_b}(f_s)$ is the received signal at frequency f_s, $s = 0, 1, \ldots, F-1$, F is the number of frequencies, and c is the speed of light. Note that this corresponds to a non-linear phase shift proportional to $f_s \Delta t_s$. Hence, the compensation for motion cannot *simply* be applied in the time (range) domain as a time-delay. This compensation is a rough estimate of the true effect of the motion on the 2D back-propagated image, as it is easy to see that it is only exact when cross-range (y) and height (z) are both 0-valued. However, simulation shows that this approximation is very effective at reducing the motion effect on the swept-frequency signal. Figure 2.3 shows the effect of this compensation on simulated L-band FLGPR data; view (a) shows the image with no platform motion, view (b) shows the uncompensated image, and view (c) shows the phase-compensated result. View (c) shows that the backpropagated



(a) No Motion **(b)** Uncompensated **(c)** Compensated

Figure 2.3: Effect of phase compensation for motion effects on simulated backpropagation of L-band FLGPR

image is significantly more focused. Furthermore, the 6dB of SNR loss is negated. The compensated image does show, when compared to the no motion image in view (a), that the target located off the bore-sight still has a slightly blurred signature; however, this loss is negligible.

Table 2.1 shows the AUR of our DOM prescreener (see Section 2.1.2 for a description of this prescreener) on four lanes with and without the phase adjustment. As these results show, there is little difference between the results. We attribute this to the slow speed that the vehicle is traveling. However, if faster speeds are used in future data collections, we believe that the motion-induced blurring will be more significant.

Table 2.1
Effect of Phase Adjustment on DOM Prescreener AUR

	with adjustment		without adjustment	
Lane	**HH**	**VV**	**HH**	**VV**
A	0.4982	0.4230	0.4949	0.4386
B	0.2594	0.2319	0.2865	0.2499
C	0.0697	0.0128	0.0753	0.0126
D	0.1118	0.0823	0.1271	0.0943

2.1.1.3 Coherent Integration Length

Several coherent integration lengths were experimented with for the L-band radar. These lengths represent the down range area in which each frame of the FLGPR is imaged. This area is also a representation of the possible stand-off distance (maximum distance at which a target can be detected). A longer coherent integration distance would mean that the images are formed farther from the vehicle; thus, objects can be detected at longer distances. However, while it is desirable to increase stand-off distance, it is far more important to have excellent detection statistics. A hazard that is detected closer is far preferable to a hazard that is not detected at all. Table 2.2 shows the prescreener AUR over various coherent integration lengths.

Table 2.2
AUR Comparison for Different L-Band Coherence Lengths

Lane A	xmin = 5, xmax = 10		xmin = 10, xmax = 15		xmin = 15, xmax = 20		xmin = 5, xmax = 20	
	DOM	Bhatt	DOM	Bhatt	DOM	Bhatt	DOM	Bhatt
HH	**0.4982**	**0.4281**	0.3155	0.2428	0.2535	0.1932	0.2833	0.2450
VV	**0.4230**	**0.3802**	0.3788	0.3463	0.2635	0.2183	0.3860	0.3300
Lane B								
HH	0.2594	0.2395	**0.2743**	**0.2543**	0.1934	0.1539	0.2132	0.1906
VV	0.2319	0.1767	**0.2482**	**0.1968**	0.1754	0.1456	0.2184	0.1562

In this table we can see that, while the coherent integration length of 10-15 has the best performance for lane B, it only is slight better than the 5-10 meter integration length. Comparatively, lane A is vastly better over the coherent integration length of 5-10 than it is over 10-15. For this reason, we will be using the 5-10 meter coherent integration length for the results presented in this document. Once the images had been formed, they were preprocessed using a *Constant False Alarm Rate* (CFAR) prescreener. This prescreener finds the intial hits and labels them using the ground truth data. It should be noted that the labels are only used in the training lanes. Once the CFAR prescreener provides the hit list, a feature extraction algorithm is used to pull important image features from the hit locations.

2.1.2 Constant False Alarm Rate Prescreener

The result of the radar imaging procedure above is a coherently integrated image I at predetermined UTM coordinates (u, v), one for each polarization of the L-band FLGPR and one image (the VV polarization) of the X-band FLGPR. It is well known that penetration depth increases with wavelength; hence, the L-band will have a deeper penetration than the X-band radar. Thus, we use the L-band radar as the detection radar for the method proposed here; although, we will show results for X-band detection too.

The prescreening detector is the first algorithm that indicates candidate detection locations—a block diagram is shown in Fig. 2.4. We employed two methods to indicate the presence of a target, both of which could be considered to be a *constant false alarm rate* (CFAR) detector. Consider an FLGPR image $I(u, v)$, where u is the cross-range coordinate, and v the down-range. We then produce four images from $I(u, v)$, denoted as $I_{\mu_c}(u, v)$, $I_{\mu_h}(u, v)$, $I_{\sigma_c^2}(u, v)$, $I_{\sigma_h^2}(u, v)$, calculated as

$$I_{\mu_c}(u, v) = \frac{\{I * H_c\}(u, v)}{\sum H_c}; \tag{2.6a}$$

$$I_{\mu_h}(u, v) = \frac{\{I * H_h\}(u, v)}{\sum H_h}; \tag{2.6b}$$

$$I_{\sigma_c^2}(u, v) = \{I^2 * H_c\}(u, v) - \{I * H_c\}^2(u, v); \tag{2.6c}$$

$$I_{\sigma_h^2}(u, v) = \{I^2 * H_h\}(u, v) - \{I * H_h\}^2(u, v); \tag{2.6d}$$

where I^2 indicates the image with each element squared, $*$ indicates convolution, and H_c and H_h are elliptical convolution kernels as shown in Fig. 2.5. In essence, I_{μ_c} and I_{μ_h} are the mean values of the pixels in the center and halo, respectively, surrounding each pixel and $I_{\sigma_c^2}$ and $I_{\sigma_h^2}$ are the corresponding variances. Detections can now be indicated by either of the difference in the means (or size-contrast filter) or the modified Bhattacharya distance:

$$
\begin{aligned}
I_{sc}(u,v) =& I_{\mu_c}(u,v) - I_{\mu_h}(u,v); && \text{(2.7a)}\\
I_B(u,v) =& \operatorname{sgn}\{I_{sc}(u,v)\} \cdot \left[\log\left(\frac{1}{4} \left[\frac{I_{\sigma_c^2}(u,v)}{I_{\sigma_h^2}(u,v)} + \frac{I_{\sigma_h^2}(u,v)}{I_{\sigma_c^2}(u,v)} + 2 \right] \right) \right. && \text{(2.7b)}\\
& \left. + \frac{\left(I_{\mu_c}(u,v) - I_{\mu_h}(u,v)\right)^2}{I_{\sigma_c^2}(u,v) + I_{\sigma_h^2}(u,v)} \right]; && \text{(2.7c)}
\end{aligned}
$$

where the Bhattacharya distance is modified so that it is signed such that positive distance indicates that the mean of the center is greater than the mean of the outer. In our experiments, we have determined the following prescreener parameters to be good choices for this system: down-range radius $= 0.25$m; cross-range radius $= 0.5$m; and halo width $= 0.75$m.

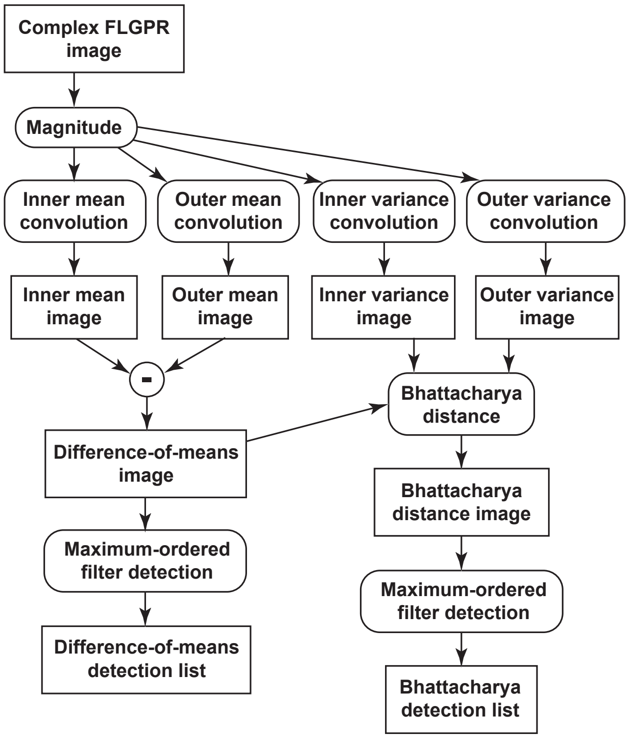


Figure 2.4: Block diagram of prescreener detection algorithm[24]. (see reprint permissions in A)

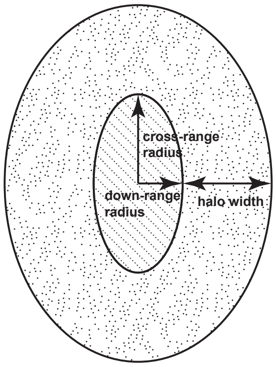


Figure 2.5: Elliptical convolution kernels used in prescreener. Detection is indicated by comparing the distribution of pixel intensities in inner ellipse to the distribution of pixel intensities in outer halo[24]. (see reprint permissions in A)

One could simply threshold I_{sc} or I_B to indicate a detection; however, this can result in many detections in one local region. We wish to have one prototype detection location for each candidate target; hence, we first calculate a maximum order-filtered image, denoted $I_o(u,v)$, with a 3m (cross-range) by 1m (down-range) rectangular kernel. Detection locations are indicated by

$$A = \arg_{(u,v)}\{I_*(u,v) = I_o(u,v)\}, \tag{2.8}$$

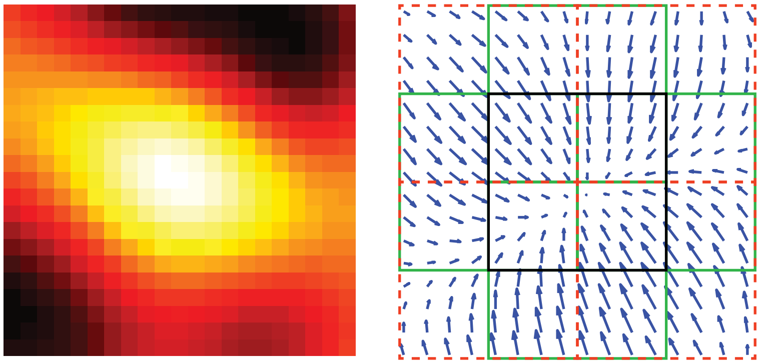
where I_* is either I_{sc} or I_B and A is the set of cross-range and down-range locations of detections. At each detection location, we also extract a set of shape- or texture-based features, which we now describe.

2.1.3 Image Feature Extraction

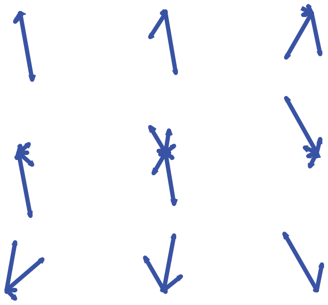
Once the hit list is generated, image features are extracted from the hit locations. The primary image features that were used were the *Histogram of Ordered Gradients* (HOG), *Local Binary Patterns* (LBP) and the *Local Statistics* (LSTAT). Additionally, the imagelet and *Fast Finite Shearlet Transform* (FFST) of each hit was also extracted. For each of the following features, a sub-image, or imagelet, was extracted for each detection location and the features were then extracted from the imagelet.

2.1.3.1 Histogram of Ordered Gradients

At each detection location, we calculate an image-based texture feature called the *histogram of ordered gradients* (HOG)[47]. This feature represents the texture by calculating local gradients and then compiling these gradients into a histogram descriptor. We use cell patterns of 3×3, 4×4, and 5×5 for this feature. These cells contain, 8×8, 16×16, and 24×24 pixels each respectively. A histogram is computed for each cell of pixels surrounding the detection location. The center cell is centered on the detection location. The surrounding cells are organized such that there is 50% overlap between neighboring cells. The histogram in each cell has 9 bin centers. Thus, there are a total of 9 feature values per cell, for a total of 81 feature values in the 3×3 cell pattern that describe each detection. Figure 2.6 illustrates the HOG calculation for an example detection.



(a) Sub-image at hit location **(b)** Gradient calculation and 3×3 cell arrangement with 50% overlap



(c) Cell-based 9-bin histogram of gradients feature

Figure 2.6: Example of histogram of ordered gradients (HOG) with 3×3 cell arrangement, 50% overlap of cells, 8×8 pixels per cell. Feature is a $3 \times 3 \times 9 = 81$-length vector of histogram components[24]. (see reprint permissions in Appendix A)

2.1.3.2 Local Binary Patterns

The second feature we use is called *local binary pattern* (LBP). This feature uses gray-scale variations to capture the texture of objects in an image. An effective method for capturing this texture information is the LBP that was developed by Ojala et al [29]. First, the LBP captures a binary pattern for each pixel in the image. To accomplish this, we use an 8 pixel neighborhood with a radius of 1. For each neighborhood we calculate

$$LBP_{8,1} = \sum_{p=1}^{8} s(t_p - t_c) 2^p, \tag{2.9}$$

where

$$s(x) = \begin{cases} 1 & x \geq 0, \\ \\ 0 & x < 0. \end{cases}$$

Each value of the summation in (2.9) contributes a unique bit to the binary representation of LBP, giving this feature its name. The LBP operator is calculated for each pixel in the image. The calculation of the histogram is the final step of the LBP feature extraction,

$$h_{LBP}(m) = \sum_{u,v \in \text{image}} S\{LBP_{8,1}(u,v) = m\},\ m = 1, \ldots, 256, \tag{2.10}$$

where $S\{H\}$ is a Boolean function that takes the value of 1 if the argument H is true and 0 else. The histogram contains 256 bins; each bin contains the count of the pixels in the image with the corresponding LBP pattern. The histogram is then normalized by

$$\tilde{h}_{LBP}(m) = \frac{h_{LBP}(m)}{\sum_{i=1}^{256} h_{LBP}(i)}.$$

The normalized histogram values comprise the LBP feature.

2.1.3.3 Local Statistics

The third feature extracted is the *local statistics* (LSTAT). These are simply the first four central moments of the imagelet. For reference, the equations for each are given here. In these equations, $x_{m,n}$ refers to the *m-th* row and *n-th* column of the imagelet and $\mathbf{x_n}$ refers to the column vector *n* in the imagelet.

$$\mathbf{U_1} = \left[\left| \frac{\sum_{i=1}^{m} x_{i,1}}{m} \right|, \left| \frac{\sum_{i=1}^{m} x_{i,2}}{m} \right|, ..., \left| \frac{\sum_{i=1}^{m} x_{i,n}}{m} \right| \right] \tag{2.11}$$

$$\mathbf{U_2} = \left[\left| \sum_{i=1}^{m} (x_{i,1} - \mu_1)^2 \right|, \left| \sum_{i=1}^{m} (x_{i,2} - \mu_2)^2 \right|, ..., \left| \sum_{i=1}^{m} (x_{i,n} - \mu_n)^2 \right| \right] \tag{2.12}$$

$$\mathbf{U_3} = \left[\left| \frac{\mathbf{E}(\mathbf{x_1} - \mu_1)^3}{\sigma_1^3} \right|, \left| \frac{\mathbf{E}(\mathbf{x_2} - \mu_2)^3}{\sigma_2^3} \right|, ..., \left| \frac{\mathbf{E}(\mathbf{x_n} - \mu_n)^3}{\sigma_n^3} \right| \right] \tag{2.13}$$

$$\mathbf{U_4} = \left[\left| \frac{\mathbf{E}(\mathbf{x_1} - \mu_1)^4}{\sigma_1^4} \right|, \left| \frac{\mathbf{E}(\mathbf{x_2} - \mu_2)^4}{\sigma_2^4} \right|, ..., \left| \frac{\mathbf{E}(\mathbf{x_n} - \mu_n)^4}{\sigma_n^4} \right| \right] \tag{2.14}$$

These features are then normalized and strung together as shown.

$$LSTAT(i) = [\mathbf{U_{1,i}}, \mathbf{U_{2,i}}, \mathbf{U_{3,i}}, \mathbf{U_{4,i}}] \tag{2.15}$$

The reasoning behind using the absolute value of each statistic is to condition the data for use with the deep learning algorithms, which require inputs of $\mathbf{x} \in [0, 1]$ for the use of the sigmoid function. Due to this requirement, the LSTAT feature is somewhat limited in the information it presents and thus is not expected to perform as well as the other two features. It may, however, be useful in conjuction with other features.

2.1.3.4 Fast Finite Shearlet Transform

The final feature pulled from the imagelets was the *Fast Finite Shearlet Transform* (FFST) coefficients. This feature was computed using Häuser's FFST algorithm [30, 31]. This algorithm is freely available and regularly updated. The FFST has been shown to work well for *convolutional neural networks* (CNNs) in the *Forward-Looking Long-Wave Infrared* (FL-LWIR) buried explosive hazard problem [32]. The FFST is a discrete version of the Shearlet Transform, computed as

$$\|f - f_N\|_2^2 \leq CN^{-2}(logN)^3 \qquad as\ N \rightarrow \infty \tag{2.16}$$

In this equation, f_N is a nonlinear Shearlet approximation of the function f and N is the largest Shearlet coefficient in absolute value. In order to discretize this, finite Shearlets must be introduced. These finite Shearlets rely on three factors: dilation, shear, and translation. Assuming an image size of $M \times N$, we let $j_0 = \left\lfloor \frac{1}{2} \log_2 \max\{M, N\} \right\rfloor$ which is equal to the number of considered scales. We also assume this image is on a grid of $\mathscr{G} = \{(m_1, m_2) :$

$m_1 = 0, ..., M-1, m_2 = 0, ..., N-1\}$. Given this, the dilation, shear, and translation can be defined as

$$\text{Dilation: } a_j = \frac{1}{4^j}, j = 0, ...j_0 - 1; \tag{2.17a}$$

$$\text{Shear: } s_{j,k} = k2^{-j}, -2^j \leq k \leq 2^j; \tag{2.17b}$$

$$\text{Translation: } t_m = \left(\frac{m_1}{M}, \frac{m_2}{N}\right), m \in \mathscr{G}. \tag{2.17c}$$

With these properties, the Shearlets can be written as

$$\Psi_{j,k,m}(x) = \Psi_{a_j, s_{j,k}, t_m}(x) = \Psi(A^{-1}_{a_j, \frac{1}{2}} S^{-1}_{s_{j,k}}(x - t_m)) \tag{2.18}$$

in the time domain, where $A_a = \begin{pmatrix} a & 0 \\ 0 & \sqrt{a} \end{pmatrix}$ is the dilation matrix and $S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$ is the shear matrix. Now we can take the shearlets into the Fourier domain to get

$$\hat{\Psi}_{j,k,m}(\omega) = \hat{\Psi}(A^T_{a_j} S^T_{s_{j,k}} \omega) e^{-2\pi \mathbf{i} \langle \omega, t_m \rangle} \tag{2.19a}$$

$$= \hat{\Psi}_1(4^{-j}\omega_1)\hat{\Psi}_2(2^j \frac{\omega_2}{\omega_1} + k) e^{-2\pi \mathbf{i} \left\langle \omega, \binom{m_1/M}{m_2/N} \right\rangle}. \tag{2.19b}$$

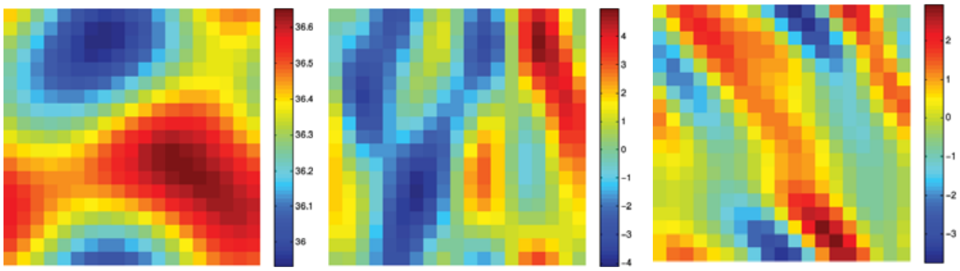
From here, the dicrete shearlet transform is simply a matter of multiplying the function of interest by the shearlets and applying the inverse FFT. Parsevel's formula is used for first step to obtain (2.20), then the 2D IFFT is applied to obtain

$$SH(f)(h,j,k,m) = \frac{1}{MN}\sum_{\omega\in\Omega}\hat{\Psi}(4^{-j}\omega_1, 4^{-j}k\omega_1 + 2^{-j}\omega_2)\hat{f}(\omega_1,\omega_2)e^{2\pi\mathbf{i}\left\langle \omega, \binom{m_1/M}{m_2/N}\right\rangle} \tag{2.20}$$

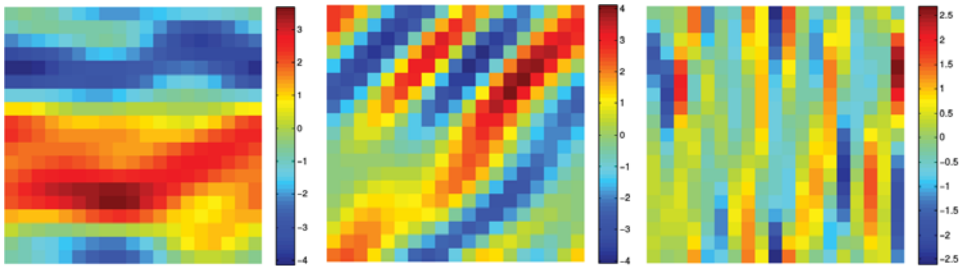
This equation can be notationally simplified by defining $\hat{g}_{j,k}(\omega) := \hat{\Psi}(4^{-j}\omega_1, 4^{-j}k\omega_1 + 2^{-j}\omega_2)\hat{f}(\omega_1,\omega_2)$. Using this and applying the 2D IFFT, we arrive at

$$SH(f)(h,j,k,m) = \mathbf{ifft2}(\hat{g}_{j,k}), \tag{2.21}$$

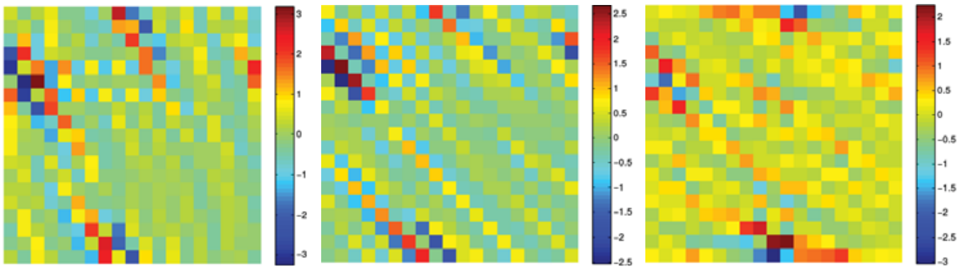
which is the discrete shearlet transfom. A more in-depth derivation of the fast finite shearlet transform may be found in [31]. Using the algorithm freely available at Häuser's website [49], we are able to compute the shearlet coefficients of each hit's imagelet. For this data set, the FFST provided 13 shearlet coefficients for each imagelet. A sample of these shearlets from a hit in Lane A are shown in Fig. 2.7.



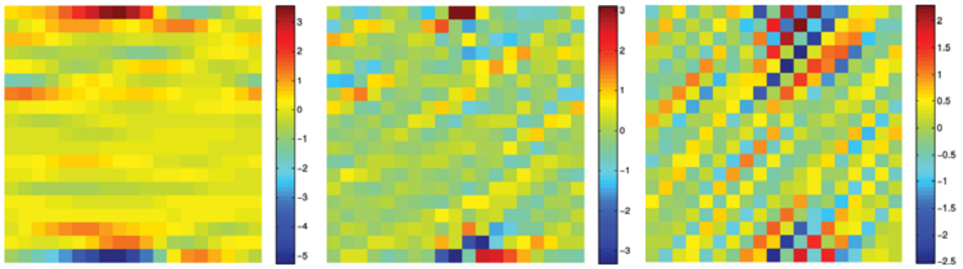
(a) Shearlet 1 (b) Shearlet 2 (c) Shearlet 3



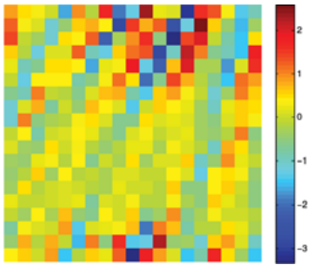
(d) Shearlet 4 (e) Shearlet 5 (f) Shearlet 6



(g) Shearlet 7 (h) Shearlet 8 (i) Shearlet 9



(j) Shearlet 10 (k) Shearlet 11 (l) Shearlet 12



(m) Shearlet 13

Figure 2.7: Shearlet coeffiecients for one lane hit

Since this feature is primarily extracted for use in the *Convolution Neural Network* (CNN), it seemed advantageous to pick the most detailed coeffieicents. Having observed these plots, the first shearlet coeffieicents were chosen; however, the first four were all rather detailed. The reason only one set of shearlet coeffieicents was chosen is two-fold. First, the concatenation of two 2D features is a difficult problem that typically results in a loss of clarity of the features. Second, the prescreener finds many hits. Using 2D features to represent these hits as is required for the CNN leaves us with large 3D matrices which are $N \times N \times P$ in size, where N is the feature length and P is the number of hits. In order to keep all the shearlets, these matricies would have to become 4-dimensional, giving us a feature matrix of size $N \times N \times P \times 13$ for *each* polarization. While this could certainly be done, it would have been very time-consuming to compute. Furthermore, the curse of dimensionality would have resulted in a poorly generalized classifier. Thus, only the first Shearlet coefficients were used. Given more time, it would have been interesting to see what the other shearlets could have provided. This topic will be revisited in the Section 4.5. We now present a more formal introduction to the deep learning methods used in this document.

Chapter 3

Classification Methods

3.1 Deep Learning Methods

The motivation for development and use of Deep Learning architechures starts at the human brain. More specifically, the way the human brain identifies what we see [21]. Recent understanding of the mind's recognition shows that the neocortex uses six layers and a forward-backward structure to classify image data collected by the eye [46]. Additional motivation for deep learning comes from the limitations of so-called shallow architechures, such as *Neural Networks* (NNs). It is often the case that adding more than two hidden

---

The material contained in this chapter was either previously published in *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014 *OR* has been accepted for publication in *SPIE Defense+ Security.* International Society for Optics and Photonics, 2015 *OR* has been submitted to the *Geospace and Remote Sensing, IEEE Transactions on*, 2015.

layers to an NN is detrimental to the network's performance [45]. Deep learning seeks to transend this boundary and allow multiple hidden layers, which in turn allows the nodes of each hidden layer to act as more generalized feature detectors and thus allows a higher level of recognition.

Let us first consider the classic neural net, with one input, one hidden, and one output clasification layer. Since there is only one hidden layer, its nodes can only represent a general feature information. Say for example that the nodes in the NN hidden layer represent Gabor-like filters. Since a deep learner is heirachical, its hidden layers can represent varying classes and dimensions of feature detectors. The deep learner's first layer could be Gabor-like, then the second layer could be edge and corner detectors, then pixel intensities in the layer after that. It is in this manner that a deep learner can reduce the dimensionality of an input.

In the case of the famous MNIST dataset, the input image is a 28×28 greyscale image of a handwritten digit from 0 to 9. This yields a network input of 1×784. In the case of the neural network, this input would typically be expanded out over a larger hidden layer, then fanned back in to a 1×10 clasification layer. Even with the limitation of a single hidden layer, NNs have been shown to achieve testing errors as low as 7.6% [27]. This is good, but far from state-of-the-art. Using appropriate deep learning algorithms, a testing error of 0.7% [27] or lower can be achieved. The improvements do not stop at MNIST; deep learning algorithms have been shown to perform excellently on a variety of image classification tasks such as facial recognition [44], document classification [27], and even speech recognition [43]. With this motivation in mind we will now describe the three most common deep learning architechures, the *Deep Belief Network* (DBN), *Stacked-Denoising Autoencoder* (SDAE), and the *Convolutional Neural Network* (CNN).

3.1.1 Deep Belief Networks

DBNs are a type of deep learning network formed by stacking *Restricted Boltzmann Machines* (RBMs) in succesive layers to reduce dimensionality by making a compressed representation of the input. DBNs are trained layer by layer using greedy algorithms and information from the previous layer. In this subsection, we will first discus RBMs and how to train them, then move on to training DBNs.

3.1.1.1 Restricted Boltzmann Machines

In this section, we will denote $\sigma(x)$ as the sigmoid activation function,

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{3.1}$$

Restricted Boltzmann Machines are simple binary learners that generate stochastic representations of the input data. They consist of two layers, one visible and one hidden. The visible layer is the input layer and typically consists of a $1 \times N$ vector of normalized, grayscale pixel values. The hidden layer can then be thought of as a feature representation layer. The defining equation of the RBMs is the energy equation,

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{bv} - \mathbf{ch} - \mathbf{vhW}, \tag{3.2}$$

where **v** is the input vector, **h** is the hidden feature vector, **b** and **c** are the visible and hidden layer biases, respectively, and **W** is the weight matrix that connects the layers. It should be noted that weights only exist between the hidden and visible layers, that is to say, that the nodes in either layer are not interconnected. **v** is the input and used to train hidden layer **h** as

$$\mathbf{h} = \sigma(\mathbf{c} + \mathbf{vW}). \tag{3.3}$$

The hidden layer is then used to reconstruct the visible layer in the same manner,

$$\mathbf{v}_{recon} = \sigma(\mathbf{b} + \mathbf{hW}^T). \tag{3.4}$$

The reconstruction of the visible layer is put back through (3.3) to form $\mathbf{h}_{recon}$ and then the weight updates are given by

$$\Delta\mathbf{W} = \epsilon(\langle\mathbf{vh}\rangle_{data} - \langle\mathbf{vh}\rangle_{recon}), \tag{3.5}$$

where ϵ is the learning rate. Itereated over several epochs, this weight update performs a type of gradient descent called Contrastive Divergence [21].

3.1.1.2 Training DBNs

Training a DBN is done layer by layer, where each layer is an RBM. Once the first RBM is trained, its reconstructed hidden layer is used in (3.6) to create the *visible* layer of the next RBM. Once the first RBM is trained, its reconstructed hidden layer h_{recon} is used to create the visible layer of the next RBM by

$$\mathbf{v_{n+1}} = \sigma(\mathbf{c}_n^T + \mathbf{h}_n \mathbf{W}_n^T) \tag{3.6}$$

where n denotes the layer number. Once the visible layer has been created, the layer is trained as an RBM. This cycle is repeated for the number of layers desired. After all layers have been trained, the DBN is typically then mirrorred to make an encoder-decoder as shown in the *Unrolling* column of Fig. 3.1 [18]. Once this has been done, passing an input through the encoder-decoder will produce a reconstruction of the same size as the input. This pass through is done simply using

$$x_{recon,n+1} = \mathbf{W}_n \mathbf{x}_{recon,n} \tag{3.7}$$

$$x_{recon,n-1} = \mathbf{W}_n^T \mathbf{x}_{recon,n} \tag{3.8}$$

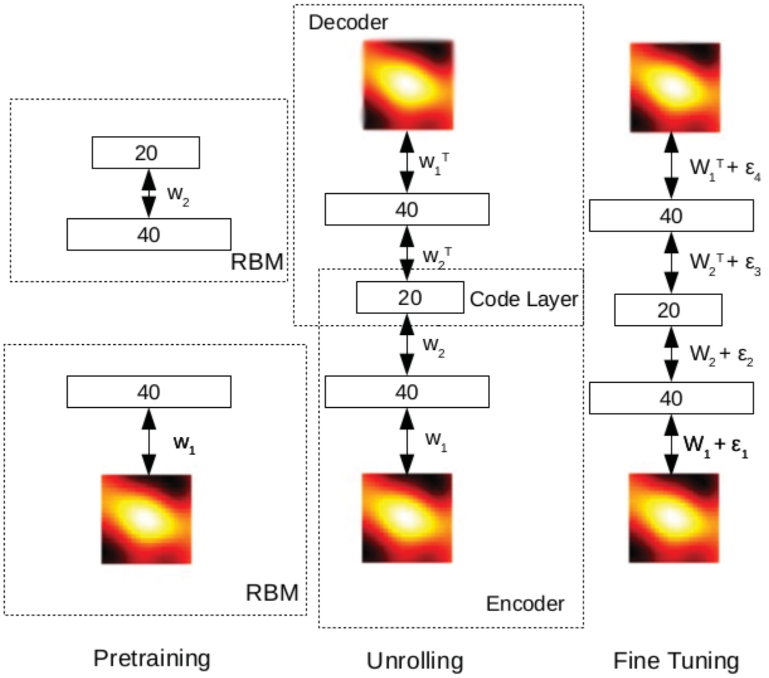


Figure 3.1: Training of a deep belief network[51]

where $x_{recon,1} = x_{data}^T$ for the encoding side and $x_{recon,1} = x_{recon}$ on the decoding side. This reconstruction along with the input can then be passed though a cost function and fine-tuning can be performed as shown in the rightmost column of Fig. 3.1. This fine-tuning is often done using stochastic gradient descent or Hinton's *up-down* algorithm [19].

3.1.2 Stacked Denoising Autoencoders

SDAEs are deep learning architechures formed by stacking *Denoising Autoencoders* (DAEs). Autoencoders and Denoising Autoencoders will be explained in the next two subsections and the final subsection will talk about training SDAEs.

3.1.2.1 Autoencoders

Autoencoders, like RBMs, have two layers: one input and one hidden. The input vector is mapped to the hidden layer via a deterministic mapping function. This representation is then used to generate a reconstruction of the input vector using the same mapping function and transposing the connecting weights and biases. Using this reconstruction, one can use a cost function to find the reconstruction error. Common cost functions are the squared error and cross-entropy. Using these errors, one can optimize the Autoencoder through methods, such as stochastic gradient descent [20]. Equation (3.9) shows the mapping to the hidden layer, often called the encoding step. Equation (3.10) shows the reconstruction, or decoding, step [21].

$$\mathbf{y} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}_1), \tag{3.9}$$

where **W** is the NxD weight matrix and $\mathbf{b}_1$ is the encoding bias and x is the D-dimensional input vector.

$$\mathbf{z} = g_{\theta'}(\mathbf{y}) = \sigma(\mathbf{V}^T\mathbf{y} + \mathbf{b}_2), \tag{3.10}$$

where **V** is the NxD decoding matrix and $\mathbf{b}_2$ is the decoding bias.

This leaves a reconstruction error to be optimized,

$$L(\mathbf{X}, \mathbf{Z}) = \frac{1}{2}\sum_{i=1}^{m} ||\mathbf{z}_i - \mathbf{x}_i||_2^2. \tag{3.11}$$

This is the traditional squared error; an alternative is the reconstruction cross-entropy:

$$\begin{aligned} L_{\mathbf{H}}(\mathbf{x}, \mathbf{z}) &= \mathbf{H}(\mathcal{B}_x || \mathcal{B}_z) \\ &= -\sum_{k=1}^{d} [\mathbf{x}_k \log \mathbf{z}_k + (1 - \mathbf{x}_k) \log(1 - \mathbf{z}_k)]. \end{aligned}$$

This approach is suggested if **x** and **z** can be interpreted as either bit vectors or vectors of bit probabilities.

3.1.2.2 Denoising Autoencoders

A Denoising Autoencoder (DAE) is designed to reconstruct a *repaired* input from a corrupt input. A corrupt input $\tilde{\mathbf{x}}$ is created from input **x** by means of stochastic mapping $\tilde{\mathbf{x}} \sim q_D(\tilde{\mathbf{x}}|\mathbf{x})$. This is then encoded, $\mathbf{y} = \sigma(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b}_1)$ and decoded, $\mathbf{z} = \sigma(\mathbf{V}^T\mathbf{y} + \mathbf{b}_2)$. The objective is still to minimize the average reconstruction error $L_{\mathbf{H}}(\mathbf{x}, \mathbf{z}) = \mathbf{H}(\mathcal{B}_x || \mathcal{B}_z)$ over the training set. For the method we are using, the function $q_D(\tilde{\mathbf{x}}|\mathbf{x})$ is a Masking Noise in which a percentage of the input nodes are chosen at random and forced to zero. Fig. 3.2 shows the training of a DAE more clearly [20].

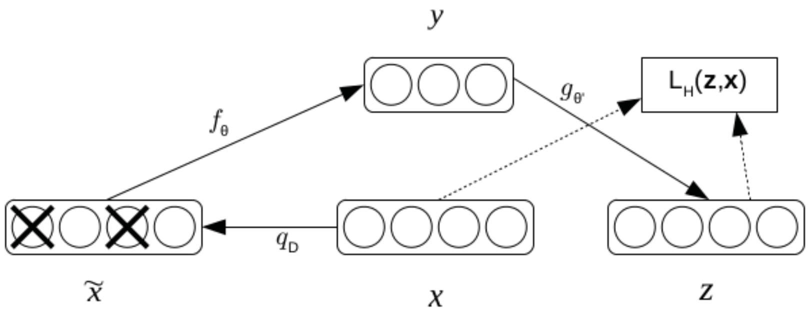


Figure 3.2: Training of a denoising autoencoder[50]

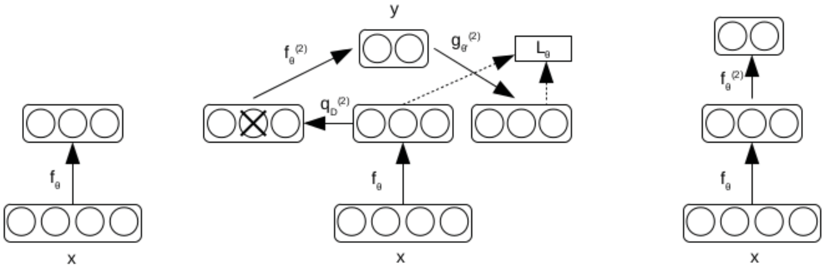


Figure 3.3: Training of a stacked denoising autoencoder[50]

3.1.2.3 Training SDAEs

Stacking DAEs to form SDEA deep architechures is much like stacking RBMs to form DBNs. It is important to note that input corruption is only used for the initial training of each layer [20]. After the mapping function f_θ is learned, it is used on clean inputs. Fig. 3.3 shows the training process of SDEAs [20]. Once the stack has been built, the output may be used as an input for a supervised learning algorithm, such as an SVM. Another popular method involves using the layer weights to initialize a NN and then using that network to do the fine-tuning and classification [21].

3.1.3 Convolutional Neural Networks

CNNs are a type of neural network with a unique architechure. Inspired by the visual system, these networks consist of alternating convolutional layers and sub-sampling layers. The convolutional layers generate feature maps by convolving kernels over the data in the previous layers and the sub-sampling layers downsample the feature maps [21]. CNNs work directly on the 2D data as opposed to the other forms of deep networks which string out the data into 1D feature vectors.

The convolutional layer l is generated from a feature map j by

$$a_j^l = \sigma(b_j^l + \sum_{i \in M_j^l} a_j^{l-1} * k_{ij}^l), \tag{3.12}$$

where σ is the activation function, usually tanh or sigmoid, b_j^l is a scalar bias, M_j^l is an indice vector of feature maps *i* in layer *l* - 1, * is the 2D convolution operator and k_{ij}^l is the kernel used on map *i* in layer *l* - 1. A sub-sample layer l is generated from a feature map j by

$$a_j^l = down(a_j^{l-1}, N^l), \tag{3.13}$$

where *down* is a downsampling function, such as mean-sampling, that downsamples by factor N^l [21]. The output layer is then generated by

$$o = f(b^o + W^o fv), \tag{3.14}$$

where *fv* denotes a feature vector concatenated from the feature maps of the previous layer, b^o is a bias vector and W^o is a weight matrix. The parameters to be learned are thus k^l_{ij}, b^l_j, b^o and W^o. Gradient descent is used to learn these parameters and this can be efficiently performed through the use of convolutional backpropagation [21]. Figure 3.4 from on the Comparison of Learning Algorithms for Handwritten Digit Recognition shows an example of the layer structure and function of a Convolutional Neural Network very similar to the one used in Palm's toolbox [21, 48].

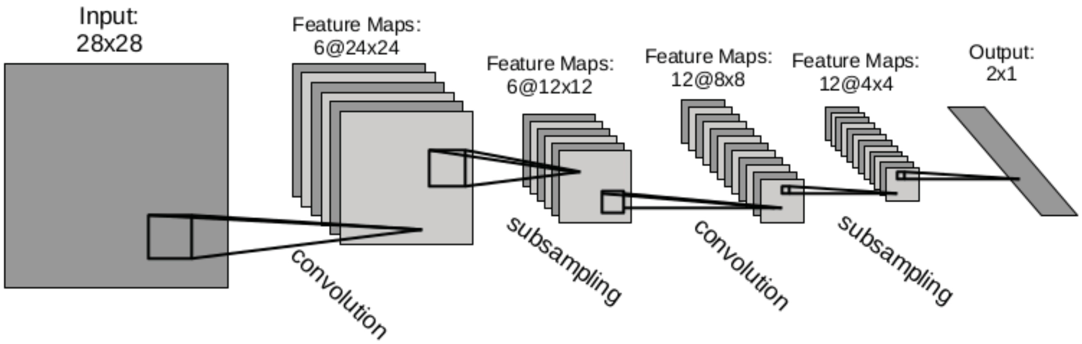


Figure 3.4: Example of a convolutional neural network

3.2 Shallow Learning Methods

In order to better understand the performance of the deep learning algorithms, two shallow algorithms were also tested. The algorithms tested for comparision were the *Single Kernel Support Vector Machine* (SKSVM) and the *Multiple Kernel Learning - Group Lasso* (MKLGL). These algorithms have been used previously in the FLGPR detection problem and their benefits are known [31]. The following two subsections will discus in detail the two shallow algorithms.

3.2.1 Single Kernel Support Vector Machine

The optimization problem is the most general definition of the SVM algorithm

$$\min_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2}, \tag{3.15}$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1,\ i = 1, \ldots, n, \tag{3.16}$$

where $y_i \in \{-1, +1\}$ are the class labels and $(\mathbf{w}^T\mathbf{x}_i - b)$ is the equation of the class-separating hyperplane. In this form, the SVM does not support overlapping classes. The soft-margin SVM was thus introduced to compensate for the overlapping classes case,

$$\min_{\mathbf{w},\xi,b}\left\{\frac{\|\mathbf{w}\|^2}{2}+C\sum_{i=1}^{n}\xi_i\right\}, \tag{3.17a}$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i-b)\geq 1-\xi_i,\ \xi_i>0,\ i=1,\ldots,n, \tag{3.17b}$$

where C determines how many errors are allowed in the training [42]. Note that (3.17) represents a linear soft-margin SVM. This may be extended to the more general kernel soft-margin SVM in which the optimization problem is solved using Lagrange multipliers. The *single-kernel* SVM (SKSVM) is defined as

$$\max_{\alpha}\left\{\mathbf{1}^T\alpha-\frac{1}{2}(\alpha\circ\mathbf{y})^TK(\alpha\circ\mathbf{y})\right\}, \tag{3.18a}$$

subject to

$$0\geq\alpha_i\geq C,\ i=1,\ldots,n,\ \alpha^T\mathbf{y}=0, \tag{3.18b}$$

where $\mathbf{1}$ is the n-length vectors of 1s, $K=[\kappa(\mathbf{x}_i,\mathbf{x}_j)]\in\mathbb{R}^{n\times n}$ is the kernel matrix, and $\circ$ indicates the Hadamard product [40]. Note that for the kernel $\kappa(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$ (which is simply the Euclidean dot product), the SKSVM reduces to the linear SVM.

One of the drawbacks of using the above SVM formulation is that it treats each datum equally; hence, when there is an imbalance between the number of datum in each class, then the SVM decision boundary is driven primarily by the data from the class with more data points. This is a problem in explosive hazards detection as there are typically many more false alarm detections than there are true targets—the true targets only comprise a small overall area of the lane. To attack this issue, we use a formulation of the SVM for imbalanced data which uses a different error cost for positive (C^+) and negative (C^-) classes. Specifically, we change the constraints of the kernel SVM formulation at (3.18) to

$$0 \geq \alpha_i \geq C^+, \forall i | y_i = +1;\ 0 \geq \alpha_i \geq C^-, \forall i | y_i = -1;\ \alpha^T \mathbf{y} = 0; \tag{3.19}$$

where C^+ is the error constant applied to the positive class and C^- is the error constant applied to the negative class. In our application, the positive class is true positives and the negative class is false alarms. We set $C^+ = n^- / n^+$ and $C^- = 1$, where n^- is the number of objects in the negative class and n^+ is the number of objects in the positive class. This essentially allows for fewer errors in the true positives class.

The popular LIBSVM is used here to efficiently solve the SKSVM problem [26]. A classifier model is the output of LIBSVM. This model contains the vector α and the bias b. We can then classify a measured feature vector $\mathbf{x}$ by computing

$$y = \operatorname{sgn}\left[\sum_{i=1}^{n} \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) - b\right], \tag{3.20}$$

where sgn is the signum function.

3.2.2 Multiple Kernel Learning - Group Lasso

MKL extends the idea of kernel classification by allowing the use of combinations of multiple kernels. The kernel combination can be computed in many ways, as long as the combination results in a Mercer kernel [35]. In this paper we assume that the kernel K is composed of a weighted combination of pre-computed kernel matrices, i.e.,

$$K = \sum_{k=1}^{m} \sigma_k K_k, \tag{3.21}$$

where there are m kernels and σ_k is the weight applied to the kth kernel. The composite kernel can then be used in the chosen classifier model. For this paper, we will use the SVM. Thus, MKL SVM extends the SKSVM optimization at (3.18) by also optimizing over the weights σ_k,

$$\min_{\sigma \in \Delta} \max_{\alpha} \left\{ \mathbf{1}^T \alpha - \frac{1}{2} (\alpha \circ \mathbf{y})^T \left(\sum_{k=1}^{m} \sigma_k K_k \right) \alpha \circ \mathbf{y}) \right\}, \tag{3.22a}$$

subject to (typically)

$$0 \leq \alpha_i \leq C, \; i = 1, \ldots, n, \; \alpha^T \mathbf{y} = 0, \tag{3.22b}$$

where Δ is the domain of σ. Note that, if the kernel weights are assumed constant, this is the exact *same* problem as SKSVM [34]. Many researchers have used this property to propose *alternating optimization* (AO) procedures to solve the problem of min-max optimization. This method solves the inner maximization given a constant kernel K, and then updates the kernel weights σ_k in order to solve the outer minimization. This is done iteratively until convergence.

Many MKL implementations only work for a single domain, thus the domain of σ is very important. For example, $\Delta = \{\sigma \in \mathbb{R}^m : \|\sigma\|_2 < 1, \sigma_k > 0\}$ is the ℓ_2-norm MKL [36, 37]. Our MKL instantiation is generalized to allow for an ℓ_p-norm domain $\Delta = \{\sigma \in \mathbb{R}^m : \|\sigma\|_p < 1, \sigma_k > 0\}$ [39]. We use the MKL *group lasso* (MKLGL) optimization procedure proposed by Xu et al. [39]. This method uses a closed form solution for solving the outer minimization to improve efficiency in (3.22), i.e.,

$$\sigma_k = \frac{f_k^{2/(1+p)}}{\left(\sum_{k=1}^{m} f_k^{2p/(1+p)}\right)^{1/p}}, \; k = 1, \ldots, m; \tag{3.23a}$$

$$f_k = \sigma_k^2 (\alpha \cdot \mathbf{y})^T K_k (\alpha \cdot \mathbf{y}). \tag{3.23b}$$

We use a modified MKLGL algorithm which uses the SKSVM for unbalanced classes—i.e., we apply the constraints with C^+ and C^- as shown at (3.19). The MKLGL training algorithm is outlined in Alg. 1. The MKLGL is simple to implement and is efficient as the update equations for σ_k are closed-form. MKL can find the optimal kernel among a set of candidates by tuning the weights on each kernel. In this way, it can be thought of as a classifier fusion algorithm. The individual kernels can be computed in many ways—see [38] for more discussion on the formation of the kernel matrices.

Algorithm 2: Appeared in [24] (see reprint permissions in Appendix A)

```
Data: $(\mathbf{x}_i, y_i)$ - feature vector and label pairs; $K_k$ - kernel matrices
Result: $\alpha$, $\sigma_k$ - MKLGL classifier solution
Initialize $\sigma_k = 1/m$, $k = 1, \ldots, m$ (equal kernel weights) ;
while not converged do
    Solve unbalanced SKSVM for kernel matrix $K = \sum_{k=1}^{m} \sigma_k K_k$;
    Update kernel weights by eqs. (3.23);
```

Chapter 4

Results and Discussions

4.1 Deep Networks for False Alarm Rejection

On its own, the CFAR prescreener does a good job of finding targets. However, through the use of deep learners, we believe we can further improve these results. A deep learner's ability to learn high-dimensional features and represent them by compressed, low-dimensional vectors makes them an excellent candidate for false alarm rejection. For this section, we train on one lane and test on another. The two lanes used here will be denoted as Lane A and Lane B. In addition to testing on both lanes, we also tested with and without the phase

---

The material contained in this chapter was either previously published in *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014 *OR* has been accepted for publication in *SPIE Defense+ Security.* International Society for Optics and Photonics, 2015 *OR* has been submitted to the *Geospace and Remote Sensing, IEEE Transactions on*, 2015.

adjustment. Each algorithm will thus have four tables associated with it, one for single image features with phase adjustment, one for single features without phase adjustment, one for combinations of features with phase adjustment, and one for combinations of features without phase adjustment. The numbers in these tables represent the AUR improvement of the algorithm as compared to just the prescreener.

The *area under ROC* (AUR) is here used to show the relative performance of our different detection methods and classifiers. This metric is calculated by normalizing the area under the *receiver operating characteristic* (ROC) curve for a given method. Figure 4.1 shows the calculation of the AUR for an example ROC curve. We chose to limit the maximum FA per squared-meter rate (FAR) to 0.1 for our AUR calculation as a FAR > 0.1 results in digging about every 1 meter down a 10 meter wide lane). The AUR equation is

$$AUR = \frac{1}{0.1}\int_0^{0.1} p_d(FAR)d(FAR), \tag{4.1}$$

where $p_d(FAR)$ is the probability of detection for the given FAR. The minimum possible AUR is 0, which indicates that no targets were detected, and the maximum possible AUR is 1, which indicates that all targets were detected with 0 FAR. For each ROC, we will also show the ROC curve and corresponding AUR of a uniform random detector, which is a detector such that one indicates hits at a predetermined uniform random spatial rate—the uniform random detector will be shown by a red dotted line. The AUR improvement then is the percentage of improvement of the AUR over the prescreener. After all the tables have been presented, two figures will be shown. One figure is the ROC curve for the best single

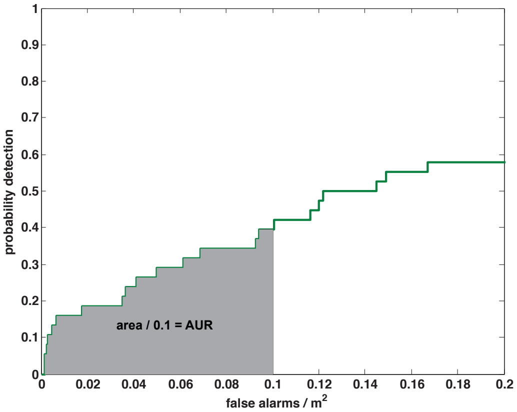


Figure 4.1: Illustration of AUR calculation[24](see reprint permissions in Appendix A)

feature and the other is the ROC curve for the best combination of features.

4.1.1 DBNs for False Alarm Rejection

Deep belief networks are very useful for learning representations and patterns in large data sets. Because of this, they can be trained to generate accurate representations and reconstructions of data and images and thus could be useful for learning to detect and reject false alarms in FLGPR data. One way to do this is to use a DBN to learn the notion of a false alarm and then use the reconstruction *root mean-square error* (RMSE) to determine if the object is a target or not. Our process is as follows. First, we use the CFAR Prescreener to determine where the hits are in the training lane and then extract the image features as described in the Section 2.1.3. Second, we remove all the true hits so that only the false alarms remain. Third, we train three DBNs, one for each FLGPR channel, on the image features of the false alarms. In doing this, we hope that the DBNs will learn to very accurately model false alarms and, by contrast, do a very poor job of reconstructing true positives. Once the DBNs are trained, we load the testing lane. We once again use the CFAR Prescreener to find hits in the testing lane and then extract their features. We then push each of the prescreener hits through the DBN and take the RMSE between the reconstruction and the feature input. This RMSE is then used as the hits' confidences in the ROC curve. The main idea is that since the DBN is well trained on false alarms, it will model targets very poorly, thus leading to a high RMSE and therefore a high confidence for the ROC curve.

Since DBNs have many parameters to adjust, many different architechures, learning rates, and epoch amounts were tested. The best combination we found for this data set is an architechure using two hidden layers of sizes 40 and 20, giving a full encode-decode stack architechure of [$\mathbf{x}$ 40 20 40 $\hat{\mathbf{x}}$], where $\mathbf{x}$ is the $1 \times N$ feature vector input and $\hat{\mathbf{x}}$ is the $1 \times N$ reconstruction of the input.This architechure is used for all three channels, as is the learning rate of 0.9 and 30 epochs of contrastive divergence for the RBM training. Such a low number of epochs were used as this network was prone to overfitting on the training data. In order to better compare the performance, this process was repeated for every combination of image features over three cellsizes. In each trial, the AUR of the prescreener alone was compared to the AUR of the DBN FA rejector and the improvement percentage was calculated. The AUR improvements of all of these trials can be seen in Tables 4.1 and 4.3. These trials were also performed on the images with no phase correction; thoseresults can be seen in Tables 4.2 and 4.4. These tables are ordered as follows: single image feature with phase correction, single image features without phase correction, combinations of image features with phase corrections, and combinations of image features without phase correction. The number in each cell is the percentage improvement of the classifier AUR over the prescreener AUR. The maximum improvement of each feature set for each polarization and each run is bolded for clarity.

Table 4.1

Percent AUR Improvemets using Single Features with DBN

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-53.52	-18.42	**64.35**	-92.35	0.39	**9.38**
Mimo VV:	-28.02	**57.97**	31.07	**-0.66**	-55.84	-84.62
Set VV:	**-84.86**	-98.44	-99.77	-76.67	-80.50	**-71.65**
LBP						
Mimo HH:	**-68.56**	-90.16	-94.89	**-15.89**	-86.87	-87.48
Mimo VV:	**-65.66**	-76.08	-86.00	**-16.59**	-52.64	-73.94
Set VV:	-23.52	**-22.57**	-39.63	-23.25	-1.94	**13.26**
LSTAT						
Mimo HH:	**-41.96**	-74.36	-83.76	**-54.70**	-89.53	-71.26
Mimo VV:	**-49.91**	-79.70	-67.29	**-38.90**	-74.84	-67.14
Set VV:	**-49.83**	-62.22	-75.42	**1.27**	-28.21	-78.71
FFST						
Mimo HH:	**-38.75**	-58.58	-57.30	-66.68	**-63.39**	-69.71
Mimo VV:	-32.39	**-30.87**	-49.86	**-48.58**	-63.70	-57.29
Set VV:	-51.23	-91.83	**-38.58**	**-14.36**	-33.77	-25.28

Table 4.2
Percent AUR Improvemets using Single Features with DBN and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-55.06	-18.93	**22.61**	-93.21	3.51	**10.34**
Mimo VV:	-46.21	**54.02**	7.75	-88.14	-82.35	**-8.54**
Set VV:	**-84.86**	-98.44	-99.77	-90.56	**-89.14**	-91.78
LBP						
Mimo HH:	**-77.09**	-85.56	-96.02	-85.85	**-27.63**	-55.80
Mimo VV:	**-63.77**	-87.30	-93.54	**-85.32**	-86.92	-91.45
Set VV:	-23.52	**-22.57**	-39.63	-23.89	**-16.24**	-33.17
LSTAT						
Mimo HH:	**-69.82**	-90.01	-81.96	**-56.91**	-89.24	-77.58
Mimo VV:	**-46.96**	-60.91	-88.09	**-41.18**	-71.22	-79.73
Set VV:	**-49.83**	-62.22	-75.42	**-16.06**	-26.08	-59.87
FFST						
Mimo HH:	**-31.91**	-57.54	-58.20	**-59.00**	-67.33	-60.71
Mimo VV:	**-30.15**	-53.17	-73.24	-65.75	-64.60	**-59.35**
Set VV:	-57.40	**-37.81**	-48.65	-79.11	-51.96	**-51.43**

Table 4.3
Percent AUR Improvemets using Combinations of Features with DBN

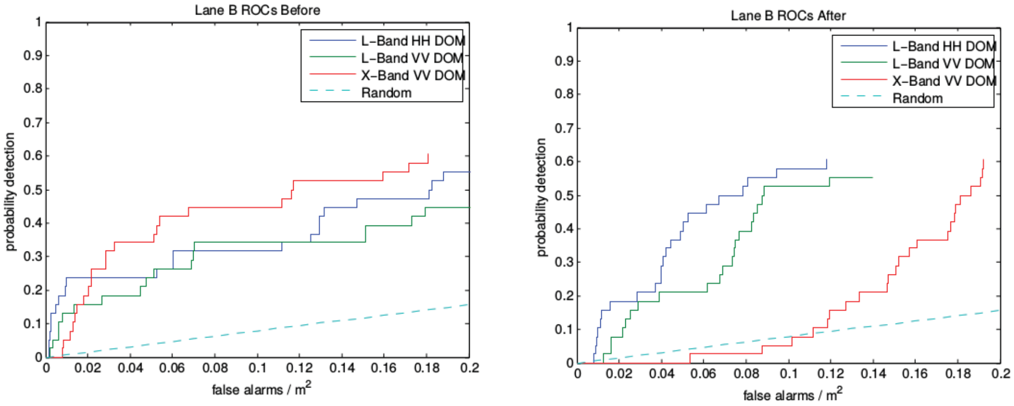
	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-17.44	-7.44	**36.02**	-64.17	-50.11	**12.91**
Mimo VV:	-27.29	7.80	**62.56**	-85.62	-23.33	**22.28**
Set VV:	-24.59	4.21	**6.93**	10.15	39.89	**48.79**
HOG & LSTAT						
Mimo HH:	-44.12	**-1.15**	-26.45	-57.99	-67.80	**-52.42**
Mimo VV:	-41.88	-30.14	**-27.19**	**-15.61**	-57.60	-55.56
Set VV:	**-37.58**	-39.88	-48.66	-10.76	**-5.43**	-26.67
LBP & LSTAT						
Mimo HH:	-40.29	**-37.59**	-66.45	**-54.60**	-74.57	-83.28
Mimo VV:	**-48.30**	-65.87	-60.89	**-41.19**	-66.66	-80.98
Set VV:	**-59.21**	-64.90	-67.95	-10.06	**-5.05**	-8.63
HOG, LBP & LSTAT						
Mimo HH:	-30.28	**-21.91**	-46.21	**-46.44**	-71.90	-77.92
Mimo VV:	**-48.55**	-58.41	-48.79	**-43.58**	-59.67	-65.15
Set VV:	**-57.12**	-62.12	-59.76	-11.38	**3.82**	3.02

Table 4.4
Percent AUR Improvemets using Combinations of Features with DBN and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-26.04	-8.43	**36.69**	-90.51	-16.99	**17.42**
Mimo VV:	-46.41	21.83	**60.06**	-51.57	-12.21	**15.07**
Set VV:	-24.59	4.21	**6.93**	5.82	39.58	**60.51**
HOG & LSTAT						
Mimo HH:	-42.77	**-25.73**	-34.54	**-44.25**	-51.01	-72.08
Mimo VV:	-66.01	-42.61	**-14.11**	**-30.96**	-61.10	-40.39
Set VV:	**-37.58**	-39.88	-48.66	**5.88**	-16.75	-36.17
LBP & LSTAT						
Mimo HH:	**-47.56**	-71.32	-62.73	**-55.96**	-80.36	-83.38
Mimo VV:	**-65.34**	-68.07	-81.06	**-37.70**	-60.76	-79.89
Set VV:	**-59.21**	-64.90	-67.95	**0.53**	-8.40	-3.68
HOG, LBP & LSTAT						
Mimo HH:	**-38.89**	-52.78	-47.51	**-64.61**	-68.13	-77.06
Mimo VV:	-58.38	-58.23	**-47.24**	**-31.84**	-58.92	-72.16
Set VV:	**-57.12**	-62.12	-59.76	-11.94	**6.17**	-2.87

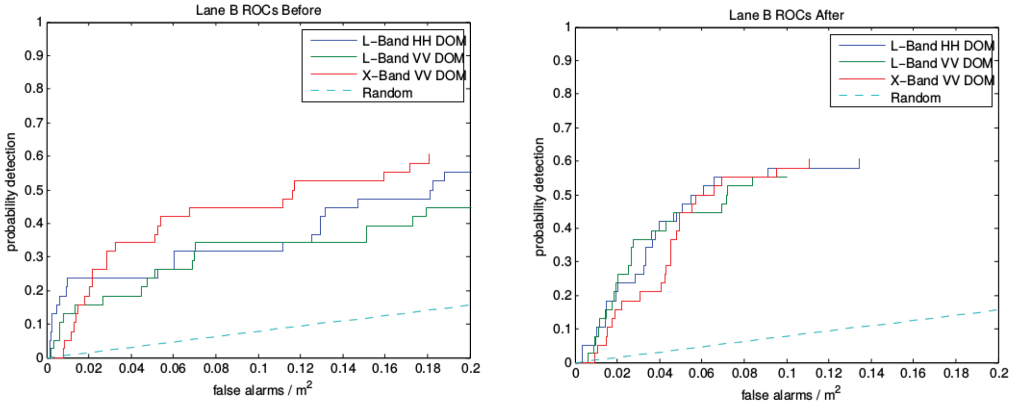
These tables tell us a few things about the DBN false alarm rejector. We see that the HOG is the best single feature. We see that though the LBP on its own does not perform very well, when combined with the HOG it helps form the best overall improvement of the network. We can also see that the LSTAT features seem to degrade the performance of the rejector. The best theory as to why the LSTAT features are so harmful is that they are not very unique between the false alarms and the actual hits and thus blur the line between the two classes. Because of this, the RMSE for the hits would be very similar to that of the false alarms and thus using the RMSE as the hit confidence is useless.

In almost every case, the phase adjusted data outperformed the non-phase adjusted data. This is a bit unexpected as the non phase adjusted data preformed better in the prescreener. It is possible that, because the prescreener does a better job on the non phase adjusted data, that there is less room for improvement and thus the network does not appear to work as well. This theory will be touched on in more detail in the Section 4.4. Figures 4.2 and 4.3 show the best overall ROC curves for the single features and combinations of features, respectively. For the single features, we use the HOG feature; for the combination, we use the HOG and LBP. Note that not all the FLGPR channels are positively improved. In the case of the single image features, the X-Band is not improved positively.



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.2: Best improvements of DBN using single image features



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.3: Best improvements of DBN using combinations of image features

4.1.2 SDAEs for False Alarm Rejection

Stacked Denoising Autoencoders have been used very successfully in feature learning and image recognition applciations [20, 25]. This makes them another possible candidate for a false alarm rejector. Two methods were attempted; the first was to train the SDAE and then use the layers as layers of a Neural Network. In this case, the backpropagation of the NN is used for further fine-tuning of the network as well as for classification of the testing data. The second method uses the top layer of the SDAE as the input to an SVM, which will be used to classify the data. In the former case, there are many options that can be changed to optimize the SDAE. These include SDAE learning rate, SDAE activation function, SDAE zero mask percentage, number of epochs to train the SDAE, layer sizes, NN activation function, NN learning rate, and number of epochs to train the NN. Taking inspiration from the DBN trials, a seperate SDAE and subsequent NN was trained on each channel of radar data. The parameters of both the SDAE and NN that we found to work the best after much trial and error can be seen in Table 4.5, where $\mathbf{x}$ is the length of the input feature vector. An additional measure that was taken was reducing the number of false alarms in the training data to prevent the network from becomeing too biased. Using the training labels, we reduce the numebr of false alarms in random order until there are equal targets and FAs. For the latter approach, we use the same SDAE parameters. We found that the linear kernel in the SVM did a better job than the RBF kernel in this case, so that is what was used. The new confidences are thus the distance of the point from the hyperplane found by the SVM.

Table 4.5
SDAE-NN Parameters

Parameter Name:	**Value:**
SDAE Activation:	Sigmoid
SADE Learning Rate:	0.9
SDAE Zero Mask:	60%
SDAE Epochs:	350
SDAE Layers:	[**x** 30 15]
NN Activation:	sigmoid
NN Learning Rate:	0.9
NN Epochs:	200

4.1.2.1 SDAE into FFNN

The first attempt to use an SDAE for false alarm rejection was to train the SDAE as mentioned in Section 3.1.2.3 and then use the layers of the SDAE as layers of a Neural Network (NN). The process for doing this is simple. Once the SDAE has been trained, the weights that make up the mapping function f_θ of each layer are simply used as the weights of the NN. Likewise, the layer sizes are used as the NN's layer sizes and an output layer of size **c** is added, where **c** is the number of classes in the training data. In the FLGPR case, **c** = 2 as each hit is either a target or a false alarm. The weights connecting the top layer of the SDAE to the output layer of the NN are initialized randomly. The training features are then pushed through the NN and classified based on the maximum node value in the output layer. This is then compared with the training label for that feature set and the errors are backpropagated through the network to fine-tune the weights. This process is repeated for a certain number of epochs and then the network is set. At this point, the testing data can be fed forward though the network and classified. The testing data is not used to modify the network. The testing data was gathered the same way as in the DBN case, using the CFAR prescreener to locate hits and determine confidences. The prescreener results are then used to generate a basis ROC curve and the AUR is calculated. These hit locations are then pushed through the NN and classified as either targets or FAs. If a hit is classified as a false alarm, the network throws out the hit location and confidence from the prescreener hit list. The newly revised hits list is used to form a new ROC curve and the new AUR is calculated. The prescreener and SDAE AURs are then compared to determine the percentage improvement. Tables 4.6 and 4.8 show the AUR improvements of the SDAE-NN with the phase adjustment while Tables 4.7 and 4.9 show improvements of the SDAE-NN without the phase adjustment. As before, the best overall improvement for each feature combination is bolded.

Table 4.6
Percent AUR Improvemets using Single Image Features with SDAE-NN

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	38.66	48.97	**69.01**	-39.35	-14.19	**12.69**
Mimo VV:	11.19	71.23	**89.63**	15.89	11.11	**23.28**
Set VV:	**2.92**	-100.00	-100.00	-6.16	**23.43**	11.49
LBP						
Mimo HH:	**0.00**	0.00	0.00	**0.00**	0.00	0.00
Mimo VV:	**0.00**	0.00	0.00	**0.00**	0.00	0.00
Set VV:	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00
LSTAT						
Mimo HH:	-39.42	**-23.06**	-32.24	-66.57	-76.38	**-25.07**
Mimo VV:	-37.94	-30.71	**7.49**	-36.82	-21.20	**0.00**
Set VV:	-57.02	-35.00	**-8.64**	**2.97**	-41.32	-2.23
FFST						
Mimo HH:	**0.00**	-100.00	-100.00	-100.00	0.00	**0.00**
Mimo VV:	**-100.00**	-100.00	-100.00	-100.00	-100.00	**0.00**
Set VV:	**0.34**	0.34	0.34	**1.82**	-100.00	-100.00

Table 4.7

Percent AUR Improvemets using Single Image Features with SDAE-NN and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	16.11	35.29	**64.87**	-3.67	-7.21	**7.86**
Mimo VV:	15.00	**64.94**	54.72	2.74	20.64	**45.95**
Set VV:	**2.92**	-100.00	1.56	-13.86	**21.55**	-65.99
LBP						
Mimo HH:	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00
Mimo VV:	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00
Set VV:	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00
LSTAT						
Mimo HH:	-60.71	**-36.48**	-46.71	-27.34	-27.12	**-13.07**
Mimo VV:	-53.35	-19.80	**-4.57**	**-23.23**	-61.57	-79.11
Set VV:	-57.02	-35.00	**-8.64**	-100.00	-35.75	**3.08**
FFST						
Mimo HH:	**-5.26**	-100	-100	**0**	-100	0
Mimo VV:	-100	**-5.26**	-100	**0**	-100	0
Set VV:	-4.94	**-4.77**	-100	**1.82**	1.82	-100

Table 4.8

Percent AUR Improvemets using Combinations of Image Features with SDAE-NN

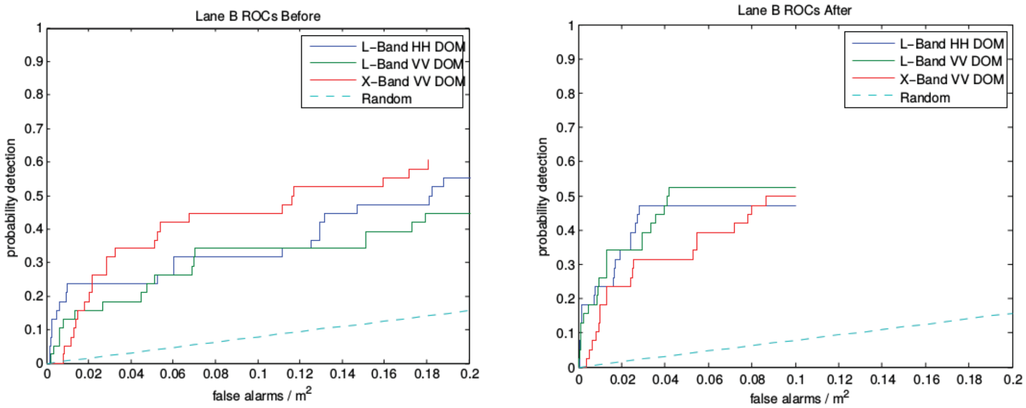
	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	25.75	45.02	**68.48**	-46.29	-19.45	**16.03**
Mimo VV:	-100.00	**55.90**	18.94	4.10	**19.73**	-100.00
Set VV:	-100.00	**38.49**	0.34	-15.84	27.73	**48.16**
HOG & LSTAT						
Mimo HH:	-4.97	**23.46**	-100.00	-8.72	8.61	**24.23**
Mimo VV:	-18.37	**25.76**	16.02	11.80	5.78	**28.12**
Set VV:	-31.15	-8.40	**0.81**	18.23	**27.95**	8.27
LBP & LSTAT						
Mimo HH:	-8.94	-29.27	**-5.17**	-35.52	-14.31	**-7.66**
Mimo VV:	**-12.76**	-16.43	-30.10	**-14.05**	-32.75	-69.86
Set VV:	-29.02	-23.73	**-0.22**	-38.85	**-25.05**	-39.07
HOG, LBP & LSTAT						
Mimo HH:	14.20	22.87	**24.27**	4.56	-23.04	**11.84**
Mimo VV:	1.32	29.24	**53.41**	-16.94	**32.27**	23.99
Set VV:	-42.46	**-1.82**	-74.13	24.57	-14.37	**42.58**

Table 4.9

Percent AUR Improvemets using Combinations of Image Features with SDAE-NN and No Phase Adjustment

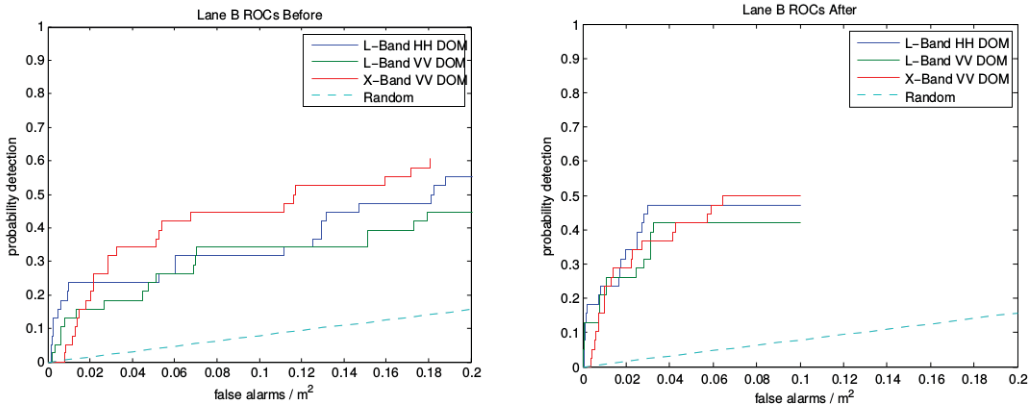
	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	17.44	47.58	**63.90**	-30.25	-16.43	**19.63**
Mimo VV:	-100.00	**50.84**	20.50	1.57	33.97	**46.39**
Set VV:	-100.00	**17.05**	-100.00	-32.89	-1.52	**58.19**
HOG & LSTAT						
Mimo HH:	-43.38	18.33	**32.29**	-6.21	2.18	**16.10**
Mimo VV:	-20.27	25.30	**36.81**	10.46	16.08	**18.33**
Set VV:	-31.15	**-6.04**	-93.91	-24.50	**34.00**	3.32
LBP & LSTAT						
Mimo HH:	**1.00**	-26.07	-18.82	**-18.94**	-62.23	-40.62
Mimo VV:	-39.77	-11.79	**1.62**	0.00	-46.71	**7.35**
Set VV:	-29.02	-23.73	**-0.22**	6.26	**13.49**	-74.30
HOG, LBP & LSTAT						
Mimo HH:	-32.55	**34.24**	31.86	**-16.69**	-100.00	-100.00
Mimo VV:	-20.59	9.54	**57.13**	9.55	-5.02	**20.78**
Set VV:	-42.46	**5.54**	-52.58	27.80	**33.94**	33.50

Much like the DBN, we can see that the SDAE-NN performs much better with combinations of image features than it does on any single feature; however, the difference is not as large as with the DBN. Much like the DBN, the SDAE-NN also prefers the HOG and LBP features. The HOG is again the best single feature and the HOG and LBP is the best combination. Dissimilarly, the LSTAT does not perform as poorly as it did in the DBN, either alone or in a combination. This is likely due to the SDAE's methods of training. Again, the phase adjusted data provided for better improvements than the non phase adjusted data. We also see that the SDAE-NN, in general, outperforms the DBN. Figures 4.4 and 4.5 again show the best improvements for the single and combinations of features.



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.4: Best improvements of SDAE-NN using single image features



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.5: Best improvements of SDAE-NN using combinations of image features

4.1.2.2 SDAE into SVM

The second attempt to use an SDAE for false alarm rejection involves again training the SDAE as in Section 3.2.3 and then using the top layer representation of the data as the input to an SVM. Unlike the case of the SDAE-NN, we found it was better not to limit the number of false alarms. Thus, the SDAEs and subsequent SVMs were trained using all the hits of the training lane. To train, we used the same size SDAEs as in the previous section with the same number of epochs. After the SDAE was trained, each hit was pushed up through the SDAE and a matrix of the compressed representations was generated. This new training matrix was then used as the training input to the SVM. We trained two models for each sensor, one with a linear kernel and one with a *radial basis function* (RBF) kernel. To perform the testing, the test lane's hits were pushed up through the pretrained SDAE's layers and a matrix of compressed representations was saved. This compressed feature matrix was used in the SVM prediction equation at (3.20). The distance from the SVM classification hyperplane was then used as the new confidence value for the hits. Results showed that the linear kernel outperformed the RBF kernel; thus, results in Tables 4.10 and 4.12 represent the testing over multiple cellsizes and feature combinations using the linear kernel SVM and the phase adjustment. Results in Tables 4.11 and 4.13 show the improvements made by using the linear kernel SVM and non phase adjusted data.

Table 4.10

Percent AUR Improvemets using Single Image Features with SDAE-SVM

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-2.85	-39.16	**22.03**	-56.50	-24.04	**-6.98**
Mimo VV:	-25.28	-5.92	**60.71**	-20.74	**18.69**	-51.49
Set VV:	-90.68	-29.79	**10.41**	-7.74	8.21	**24.22**
LBP						
Mimo HH:	-16.33	-13.94	**-13.40**	-43.01	**-34.32**	-41.48
Mimo VV:	-24.53	-47.32	**-21.41**	-69.54	-64.19	**-63.59**
Set VV:	-38.28	**-12.17**	-38.80	-20.50	**11.84**	6.02
LSTAT						
Mimo HH:	-59.85	-34.98	**-23.54**	-30.66	-64.32	**-28.86**
Mimo VV:	-53.69	**-39.96**	-64.50	**-43.45**	-67.87	-84.17
Set VV:	-41.63	-68.29	**-31.23**	-1.86	**24.62**	-41.37
FFST-						
Mimo HH:	-40.52	**-31.26**	-31.33	-65.18	-67.06	**-60.97**
Mimo VV:	**-65.65**	-67.19	-75.97	**-45.20**	-60.53	-59.47
Set VV:	-35.17	-59.88	**-27.76**	**-33.32**	-49.80	-40.84

Table 4.11

Percent AUR Improvemets using Single Image Features with SDAE-SVM and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-11.09	-74.62	**5.96**	**-30.01**	-76.19	-30.52
Mimo VV:	-48.93	1.37	**47.82**	-53.16	5.96	**22.67**
Set VV:	-90.68	-29.79	**10.41**	-17.96	13.94	**37.70**
LBP						
Mimo HH:	-29.49	**-20.19**	-29.09	-25.70	-51.59	**-14.05**
Mimo VV:	**-22.68**	-47.78	-47.16	-62.20	-65.54	**-39.56**
Set VV:	-38.28	**-12.17**	-38.80	7.67	15.47	**28.55**
LSTAT						
Mimo HH:	-69.73	**-38.76**	-42.30	-49.22	**-44.00**	-70.92
Mimo VV:	**-21.84**	-74.44	-37.18	-76.60	**-74.20**	-81.80
Set VV:	-41.63	-68.29	**-31.23**	-42.43	-54.01	**-37.90**
FFST						
Mimo HH:	-98.11	**-43.94**	-63.87	-62.08	**-57.68**	-59.16
Mimo VV:	**-30.05**	-53.59	-67.35	-69.88	-61.14	**-57.03**
Set VV:	**-48.16**	-61.30	-73.48	-34.41	-57.93	**-32.66**

Table 4.12

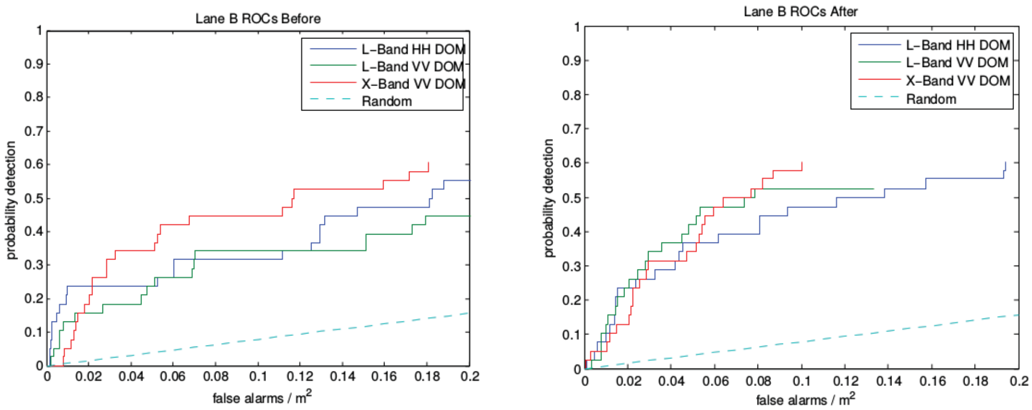
Percent AUR Improvemets using Combinations of Image Features with SDAE-SVM

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	7.62	19.98	**68.73**	-60.08	-24.42	**8.02**
Mimo VV:	-39.98	-82.88	**54.98**	-42.28	-27.85	**19.49**
Set VV:	-15.30	**9.30**	-8.99	-35.82	**22.40**	-32.13
HOG & LSTAT						
Mimo HH:	-29.69	**22.82**	-1.04	-38.75	-67.39	**-30.77**
Mimo VV:	-46.10	-52.95	**-27.08**	**-45.48**	-60.80	-87.21
Set VV:	-29.83	**-1.21**	-52.74	-22.04	-69.55	**37.95**
LBP & LSTAT						
Mimo HH:	-51.75	-40.92	**-37.97**	-62.87	**-36.68**	-82.14
Mimo VV:	-50.87	-45.59	**-9.06**	-59.91	-60.50	**-51.85**
Set VV:	**-65.69**	-70.56	-69.65	**-12.73**	-52.72	-22.75
HOG, LBP & LSTAT						
Mimo HH:	-3.25	-23.26	**42.29**	**-34.57**	-35.64	-60.97
Mimo VV:	-32.62	-50.03	**14.80**	-57.40	**-55.55**	-62.51
Set VV:	-82.41	**-14.26**	-43.79	-55.85	-18.33	**43.21**

Table 4.13
Percent AUR Improvemets using Combinations of Image Features with SDAE-SVM and No Phase Adjustment

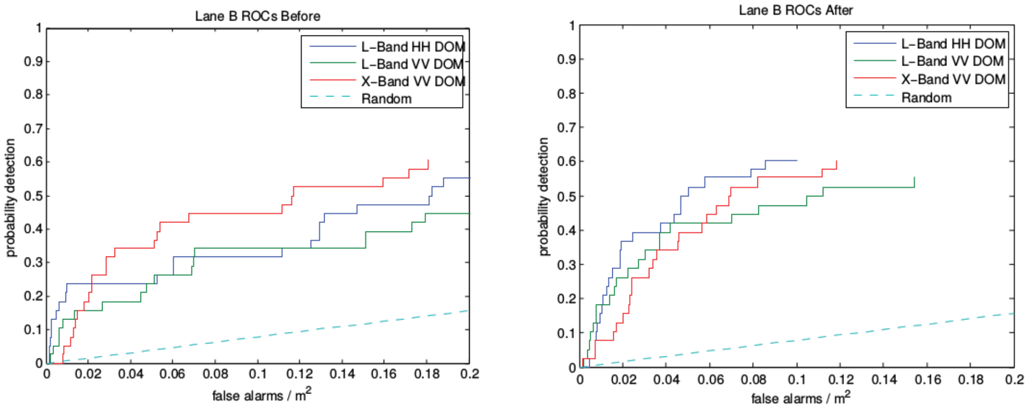
	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-22.62	-4.78	**41.96**	-43.22	-57.21	**-2.79**
Mimo VV:	-35.40	-84.97	**51.73**	-60.72	-26.57	**-21.00**
Set VV:	-15.30	**9.30**	-8.99	6.13	7.17	**49.98**
HOG & LSTAT						
Mimo HH:	-28.27	**8.08**	-13.91	-80.49	**-43.19**	-51.69
Mimo VV:	**-30.57**	-50.62	-43.38	-48.13	**-6.64**	-92.39
Set VV:	**-29.83**	-38.94	-51.90	7.33	-20.18	**26.29**
LBP & LSTAT						
Mimo HH:	-57.59	-40.30	**-37.92**	-49.40	-63.49	**-50.32**
Mimo VV:	**-23.80**	-30.67	-26.57	**-48.57**	-75.46	-72.71
Set VV:	**-65.69**	-70.56	-69.65	-58.43	**-3.05**	-42.97
HOG, LBP & LSTAT						
Mimo HH:	-39.16	**3.90**	-9.55	-45.80	**-30.74**	-32.38
Mimo VV:	-28.49	-37.19	**14.70**	-24.12	-47.22	**-8.37**
Set VV:	-82.41	**-14.28**	-35.47	-21.63	-19.31	**17.30**

Tables 4.10-4.13 seem to show that using the SVM to finetune the SDAE may be undesired. Indeed, if one compares the SDAE-SVM results with the SDAE-NN results, it is clear that the SDAE-NN performs almost uniformly better than the SDAE-SVM. This could likely be attributed to the power of the single kernel SVM combined with the power of the SDAE. Perhaps the two combined algorithms cause such an overfit to the training data that the testing data becomes horribly misclassified. Unlike in the case of the SDAE-NN, the SVM cannot be cut off at a certain number of epochs to generalize better to the testing data. Because of this, we believe the SDAE-SVM is overtraining and, hence, the testing results are poor. Figures 4.6 and 4.7 further show the shortcomings of this network.



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.6: Best improvements of SDAE-SVM using single image features



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.7: Best improvements of SDAE-SVM using combinations of image features

4.1.3 CNNs for False Alarm Rejection

The final deep architechure explored for FA rejection was the CNN. This architechure has already shown to achieve state-of-the-art performance on machine learning benchmarks such as the MNIST data [27]. Because of its previous performace and its ability to use $N \times N$ input features (i.e. raw imagelets), we are exploring ways of using this network for FA rejection. As before, we train a different CNN for each channel, all having the same layer setup. These layers are slightly different depending on which feature is used for training due to the math behind the network. Each network has an $N \times N$ input layer and an output layer of size 2×1. In between are two convolutional layers (C1 and C2) and two subsampling layers (S1 and S2). The subsampling layers both use a scale of 2. The details on the convolutional layers are given in Table 4.14. The overall structure is thus [input C1 S1 C2 S2 output].

Table 4.14

CNN Layer Parameters

Layer	Output Maps	Kernel Size Imglet	Kernel Size HOG	Kernel Size LBP	Kernel Size FFST Coeffs
C1	6	4	2,3,2	5	4
C2	12	5	3,2,2	5	5

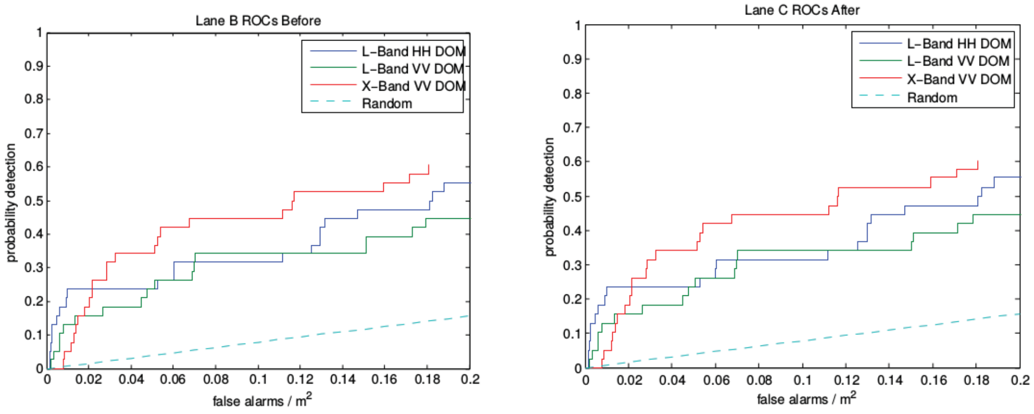
We randomly reduce the number of FAs to twice the number of hits per channel for training. The learning rate was 0.9 and 350 epochs were used in for training. These numbers were found to be optimal in preliminary testing. Since the CNN operates on 2D inputs, combinations of image features were not an option. Table 4.15 shows the results of using phase adjusted data and Table 4.16 shows the results using non phase adjusted data.

Table 4.15
Percent AUR Improvemets Using CNN

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
Imagelet						
Mimo HH:	**0.00**	-100.00	-100.00	-100.00	**0.00**	-100.00
Mimo VV:	**0.58**	-15.56	0.00	**0.33**	-100.00	0.00
Set VV:	-36.53	**0.34**	-100.00	-100.00	-100.00	**1.82**
HOG						
Mimo HH:	-41.33	-100.00	**-32.63**	**-40.90**	-45.87	-79.46
Mimo VV:	-40.54	**0.00**	-100.00	-23.28	**-12.21**	-86.52
Set VV:	0.34	-24.64	**0.34**	**1.82**	-100.00	-100.00
LBP						
Mimo HH:	-100.00	-100.00	**-100.00**	0.00	0.00	**0.00**
Mimo VV:	0.76	0.76	**0.76**	0.00	0.00	**0.00**
Set VV:	0.34	0.34	**0.34**	1.82	2.70	**4.39**
FFST						
Mimo HH:	**-5.26**	-100.00	-100.00	**-5.26**	-100.00	-100.00
Mimo VV:	**-5.26**	-5.26	-100.00	**-5.26**	-5.26	-100.00
Set VV:	-12.23	**-4.94**	-100.00	-12.23	**-4.94**	-100.00

Table 4.16
Percent AUR Improvemets Using CNN and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
Imagelet						
Mimo HH:	**0.00**	-100.00	-100.00	-100.00	**0.00**	-100.00
Mimo VV:	**0.65**	-25.61	0.00	0.00	-100.00	**0.00**
Set VV:	-36.53	**0.34**	-100.00	-3.49	-100.00	**1.82**
HOG						
Mimo HH:	**-50.00**	-100.00	-55.78	**-37.88**	-100.00	-73.15
Mimo VV:	-76.36	**0.00**	-100.00	-43.61	**-37.68**	-100.00
Set VV:	**0.34**	-24.64	0.34	-100.00	1.82	**1.82**
LBP						
Mimo HH:	**-100.00**	-100.00	-100.00	-100.00	-100.00	**-100.00**
Mimo VV:	**0.37**	0.37	0.37	0.00	0.00	**0.00**
Set VV:	**0.34**	0.34	0.34	-1.66	-2.04	**-0.03**
FFST						
Mimo HH:	**-5.26**	-100.00	-100.00	-100.00	-100.00	**0.00**
Mimo VV:	**-5.26**	-5.26	-100.00	0.00	-100.00	**0.00**
Set VV:	-12.23	**-4.94**	-100.00	-49.16	-100.00	**1.82**



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.8: Best improvements of CNN using single image features

The results of the CNN are actually disappointing. This network has been shown to work very well on a wide range of image classification tasks, but it falls short here. Additionally, as seen in our conclusions in Section 4.4, it is by far the slowest of the deep learning networks. Since CNNs have also been shown to perform rather well on similar data [32], we believe they are still worth consideration. Figure 4.8 further displays the inadequete performance of the CNN.

4.2 Comparison with Shallow Methods

To achieve a better understanding of the possible benefits of using deep leanring algorithms for this problem, we also tested with two shallow architechures that have been explored previously for this problem [24]. These architechures are the previously described SKSVM and MKLGL. Again we used the same two lanes, A and B, and again we represent the same results, the percentage AUR improvement over the prescreener. Additional comparisons between the deep and shallow algorithms, such as run time and memory costs, can be found in Section 4.4.

4.2.1 SKSVM

One popular method that we have looked into in the past is using SVMs to classify the data and generate confidences. The way this works is that the SVMs learn a hyperplane to seperate the data points into two classes. We use LIBSVM to train the SVMs and make predictions on them. The predictions give the labels of the testing data as well as the data's distances to the hyperplane. These distances are then used as the hits' confidences when creating the ROC curve. In this way, a point that is farther from the hyperplane should be more correctly classified and thus have a higher confidence value. Additionally, we choose -1 as the label for FAs, meaning they have a negative distance to the hyperplane and thus a negative confidence. We tested two different kernels for the SVM. The first was the linear kernel. As the name implies, this kernel seeks to find a linear hyperplane to seperate the two classes. Tables 4.17 and 4.19 show the AUR improvements using linear kernel SVMs over various cellsizes and image features with the phase adjusted data. Tables 4.18 and 4.20 show the results for the non phase adjusted data. For a more accurate comparison to the deep learning algorithms, three SVMs were trained, one for each channel. The best overall improvements are bolded for each channel and feature set.

Table 4.17

Percent AUR Improvemets using Linear Kernel and Single Image Features with SKSVM

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-24.57	14.22	**64.12**	-24.62	-9.72	**8.48**
Mimo VV:	-19.78	**22.18**	10.86	-30.47	**-5.69**	-12.07
Set VV:	-44.39	-29.39	**-29.34**	-27.20	4.73	**28.29**
LBP						
Mimo HH:	**23.77**	19.75	-2.40	-11.43	**-5.92**	-26.19
Mimo VV:	-9.65	-17.75	**10.64**	**-26.82**	-55.77	-30.57
Set VV:	-36.90	-32.48	**-13.14**	-38.93	3.50	**14.72**
LSTAT						
Mimo HH:	9.04	**9.84**	-31.52	**-24.44**	-35.37	-38.29
Mimo VV:	**-24.57**	-62.19	-33.56	**-40.17**	-60.58	-43.00
Set VV:	**-50.09**	-52.19	-63.48	-33.57	-50.19	**-6.12**
FFST						
Mimo HH:	-74.28	-60.69	**-32.40**	-72.47	-69.38	**-58.19**
Mimo VV:	**-6.88**	-60.62	-38.54	**-40.51**	-70.23	-56.68
Set VV:	-73.00	-64.25	**-55.65**	**-42.98**	-53.95	-45.30

Table 4.18
Percent AUR Improvemets using Linear Kernel and Single Image Features with SKSVM and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-46.98	6.72	**41.94**	-19.78	-11.90	**7.82**
Mimo VV:	-18.26	**22.88**	20.44	-37.88	-13.67	**2.24**
Set VV:	-44.39	-29.39	**-29.34**	-27.20	4.73	**28.29**
LBP						
Mimo HH:	-18.38	**16.84**	2.07	**-1.45**	-8.56	-39.84
Mimo VV:	-8.24	**15.22**	0.24	-16.39	**-11.19**	-28.62
Set VV:	-36.90	-32.48	**-13.14**	-38.93	3.50	**14.72**
LSTAT						
Mimo HH:	-36.38	**-30.06**	-45.19	-32.73	-32.00	**-30.25**
Mimo VV:	**-20.55**	-47.66	-36.64	-46.41	-53.57	**-17.00**
Set VV:	**-50.09**	-52.19	-63.48	-33.57	-50.19	**-6.12**
FFST						
Mimo HH:	-56.00	**-41.99**	-59.20	-70.04	**-50.96**	-56.49
Mimo VV:	-53.75	**-41.08**	-83.05	**-49.13**	-59.82	-69.81
Set VV:	**-27.35**	-49.16	-77.35	-35.61	**-7.75**	-61.33

Table 4.19
Percent AUR Improvemets using Linear Kernel and Combinations of Image Features with SKSVM

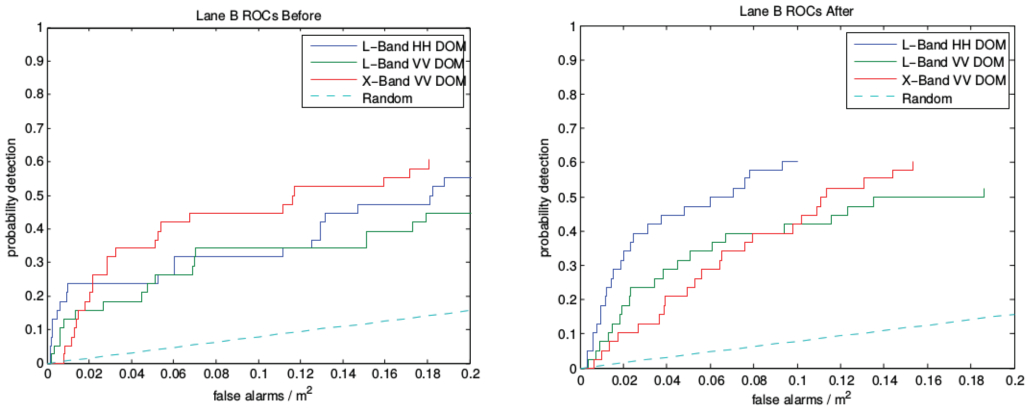
	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-2.98	22.86	**63.62**	-32.07	-9.30	**8.77**
Mimo VV:	-29.90	**19.52**	12.00	-20.25	**-7.14**	-10.76
Set VV:	-44.19	**-27.86**	-30.79	-23.03	5.39	**26.91**
HOG & LSTAT						
Mimo HH:	6.21	**25.75**	22.73	-38.75	**3.32**	-5.44
Mimo VV:	-47.27	**11.51**	6.55	-36.76	-34.89	**1.48**
Set VV:	-35.76	**-19.55**	-31.47	-50.65	-12.71	**20.42**
LBP & LSTAT						
Mimo HH:	**16.17**	9.77	-30.91	**-13.47**	-29.80	-36.59
Mimo VV:	**-28.05**	-62.93	-33.91	**-38.87**	-61.67	-42.55
Set VV:	**-40.63**	-51.14	-63.37	-20.47	-41.19	**-5.62**
HOG, LBP & LSTAT						
Mimo HH:	5.51	**25.81**	22.82	-38.80	**3.50**	-5.48
Mimo VV:	-47.27	**11.72**	6.55	-35.13	-34.68	**1.56**
Set VV:	-35.26	**-19.50**	-31.52	-46.17	-12.80	**20.42**

Table 4.20

Percent AUR Improvemets using Linear Kernel and Combinations of Image Features with SKSVM and No Pase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-37.59	7.78	**43.22**	-15.09	-10.92	**7.70**
Mimo VV:	-16.59	**27.36**	22.50	-36.25	-13.00	**2.13**
Set VV:	-44.19	-27.86	**-30.79**	-23.03	5.39	**26.91**
HOG & LSTAT						
Mimo HH:	-28.57	3.61	**19.60**	-36.78	**-3.86**	-8.70
Mimo VV:	-20.08	-0.33	**26.24**	-31.85	-13.97	**14.65**
Set VV:	-35.76	**-19.55**	-31.47	-50.65	-12.71	**20.42**
LBP & LSTAT						
Mimo HH:	-32.99	**-31.21**	-44.95	-32.41	-30.57	**-30.21**
Mimo VV:	**-9.68**	-49.03	-36.31	-38.57	-54.00	**-13.95**
Set VV:	**-40.63**	-51.14	-63.37	-20.47	-41.19	**-5.62**
HOG, LBP & LSTAT						
Mimo HH:	-26.53	3.55	**19.56**	-32.27	**-3.80**	-8.72
Mimo VV:	-19.71	0.12	**26.24**	-29.50	-14.02	**14.62**
Set VV:	-35.26	**-19.50**	-31.52	-46.17	-12.80	**20.42**

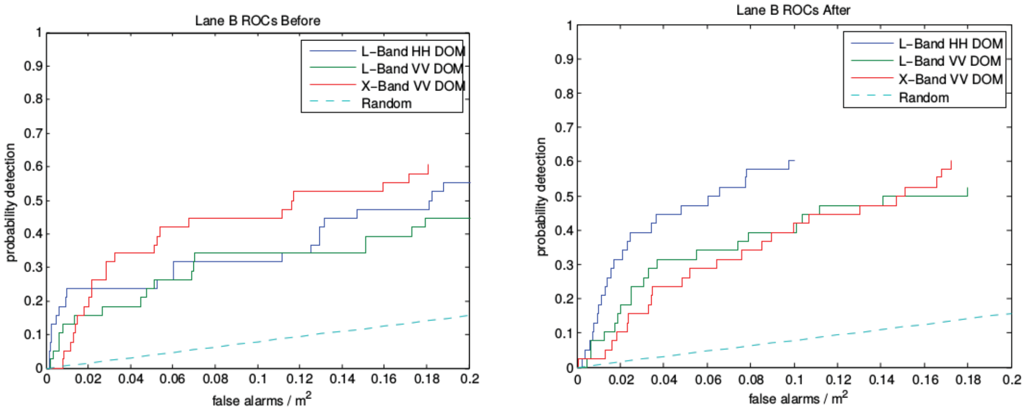
From these tables, we can see that much like the deep learners, the HOG and LBP are the strongest features. In a departure from the deep learners, the singluar features actually outperform the combinations in several instances. For example, the HOG is the best of the single features and the HOG/LBP combo is the best of the combinations. The HOG alone does better overall than the HOG/LBP combo. Another observation that can be made is that the LSTAT does not perform as poorly as with the deep learners. We believe the SKSVM handles the LSTAT better due to the shallowness of the architechure. In this way, the SKSVM combined with the CFAR prescreener is not overtraining and thus can yield better test results. We also see that in some cases the non phase adjusted data outperforms the phase adjusted data. This may be due again to the shallowness of the architechure preventing overtraining. Figures 4.9 and 4.10 show the ROCs of the best performing SKSVM classifiers.



(a) Lane B ROCs Before

(b) Lane B ROCs After

Figure 4.9: Best improvements of linear kernel SKSVM using single image features



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.10: Best improvements of linear kernel SKSVM using combinations of image features

The other kernel tested was the RBF, or Radial Basis Function, kernel. As opposed to the linear kernel, the RBF uses a Gaussian proximity to map the features to a much higher dimensional space. This kernel is very widely used in SVMs [41] and thus was thought to be a good candidate for testing the SKSVM. The training and testing process is the same as with the linear kernel; the only real difference is declaring the kernel model in LIBSVM to be RBF. Tables 4.21 and 4.23 show the resulting AUR improvements using the RBF kernel.

Table 4.21
Percent AUR Improvemets using RBF Kernel and Single Image Features with SKSVM

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	10.73	29.48	**71.47**	-11.40	-0.65	**1.29**
Mimo VV:	-46.91	1.44	**18.75**	-8.96	**5.83**	-17.29
Set VV:	-40.01	-40.90	**-25.33**	-29.08	-4.48	**26.73**
LBP						
Mimo HH:	**17.38**	-36.52	-86.81	**-27.86**	-87.20	-87.68
Mimo VV:	**-31.27**	-94.91	-79.29	-91.61	**-88.57**	-95.69
Set VV:	-15.54	**7.79**	-75.53	**-51.52**	-73.53	-70.91
LSTAT						
Mimo HH:	**-1.68**	-5.50	-26.53	-28.42	**-20.50**	-32.66
Mimo VV:	**-18.76**	-50.64	-25.03	-37.24	-53.57	**-30.21**
Set VV:	-59.83	-66.30	**-55.03**	-58.11	-45.45	**-19.18**
FFST						
Mimo HH:	**-100**	-100	-100	-100	**-100**	-100
Mimo VV:	**-100**	-100	-100	-100	**-100**	-100
Set VV:	**-100**	-100	-100	-100	**-78.30**	-100

Table 4.22

Percent AUR Improvemets using RBF Kernel and Single Image Features with SKSVM and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-18.25	26.06	**48.95**	-23.29	-5.75	**-1.42**
Mimo VV:	-40.09	4.89	**7.08**	-31.16	-6.66	**-2.48**
Set VV:	-40.01	-40.90	**-25.33**	-29.08	-4.48	**26.73**
LBP						
Mimo HH:	**-36.04**	-40.62	-80.50	**-12.67**	-59.03	-71.21
Mimo VV:	**-53.98**	-96.98	-77.97	**-65.26**	-95.75	-97.23
Set VV:	-15.54	**7.79**	-75.53	**-51.52**	-73.53	-70.91
LSTAT						
Mimo HH:	**-20.44**	-30.78	-30.26	-34.42	-34.09	**-34.00**
Mimo VV:	-33.78	-34.30	**-20.30**	-33.98	-69.36	**-31.72**
Set VV:	-59.83	-66.30	**-55.03**	-58.11	-45.45	**-19.18**
FFST						
Mimo HH:	**-48.58**	-100	-100	**-56.18**	-100	-100
Mimo VV:	**-53.75**	-100	-100	**-44.66**	-100	-100
Set VV:	**-71.34**	-100	-100	**-52.04**	-100	-100

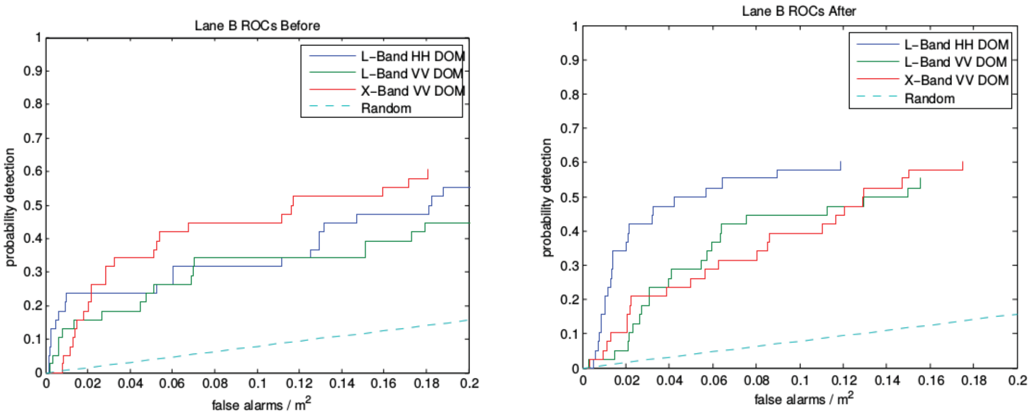
Table 4.23
Percent AUR Improvemets using RBF Kernel and Combinations of Image Features with SKSVM

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	7.64	29.43	**79.89**	-15.98	-2.14	**-1.15**
Mimo VV:	-49.55	-1.50	**0.14**	-20.89	**-11.36**	-27.17
Set VV:	-47.91	-32.86	**-29.52**	-28.17	**13.81**	4.57
HOG & LSTAT						
Mimo HH:	8.75	**26.27**	18.21	-19.19	**5.22**	-2.96
Mimo VV:	-5.08	**22.05**	12.51	-12.60	-22.97	**10.06**
Set VV:	-28.45	-30.21	**-27.55**	-34.79	-14.09	**11.34**
LBP & LSTAT						
Mimo HH:	**-1.91**	-6.29	-26.24	-29.58	**-22.13**	-32.35
Mimo VV:	**-17.61**	-50.47	-25.38	-41.39	-53.07	**-27.65**
Set VV:	-57.61	-68.07	**-55.12**	-22.26	-43.17	**-19.49**
HOG, LBP & LSTAT						
Mimo HH:	7.91	**25.61**	18.39	-17.91	**4.75**	-3.11
Mimo VV:	-5.72	**20.73**	12.33	-15.85	-24.59	**9.90**
Set VV:	-28.01	-31.83	**-26.12**	-33.76	-13.58	**10.37**

Table 4.24
Percent AUR Improvemets using RBF Kernel and Combinations of Image Features with SKSVM and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-18.06	25.28	**48.91**	-15.00	**-3.02**	-5.88
Mimo VV:	-57.70	**2.01**	-1.45	-29.45	**-6.12**	-16.50
Set VV:	-47.91	-32.86	**-29.52**	-28.17	**13.81**	4.57
HOG & LSTAT						
Mimo HH:	-18.90	-4.76	**11.62**	-21.43	**-8.70**	-10.94
Mimo VV:	-15.11	9.35	**35.37**	-31.58	-16.36	**11.39**
Set VV:	-28.45	-30.21	**-27.55**	-34.79	-14.09	**11.34**
LBP & LSTAT						
Mimo HH:	**-29.35**	-29.47	-30.13	**-29.57**	-32.95	-32.99
Mimo VV:	**-19.11**	-34.25	-20.17	-43.39	-69.45	**-28.69**
Set VV:	-57.61	-68.07	**-55.12**	-22.26	-43.17	**-19.49**
HOG, LBP & LSTAT						
Mimo HH:	-17.82	-5.48	**11.23**	-23.87	**-10.26**	-11.77
Mimo VV:	-15.38	8.13	**34.37**	-34.23	-16.99	**9.79**
Set VV:	-28.01	-31.83	**-26.12**	-33.76	-13.58	**10.37**

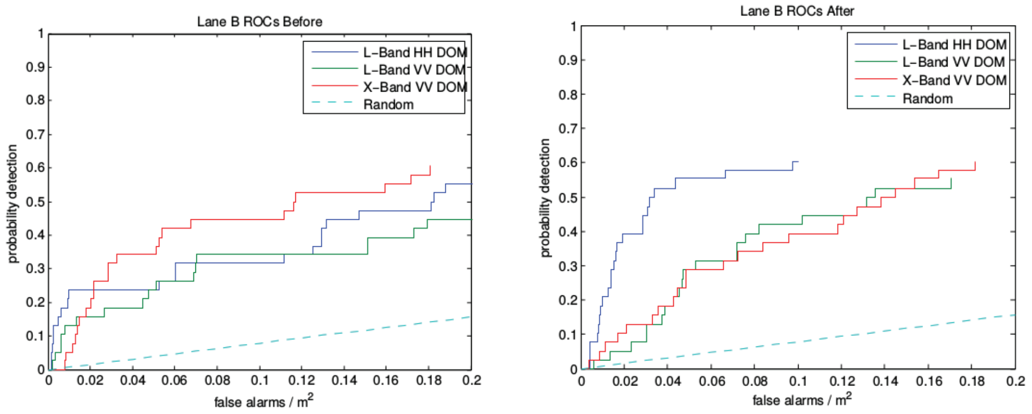
Similarly to the SDAE-SVM, the RBF performs worse than the linear kernel. It is possible in both cases that the freedom of shape provided by the RBF kernel provides a better fitted hyperplane to the training data, and thus the distance of the points from that hyperplane is shorter. This would correspond directly to an overtraining situation. We can also see an interesting comparison between the phase and non phase adjusted data. While it is consistent with the other learners in that the phase adjusted data outperforms the non phase adjusted data in general, the non phase adjusted data seemed to consistently do better for the Mimo VV polarization when using combinations of image features.



(a) Lane B ROCs Before

(b) Lane B ROCs After

Figure 4.11: Best improvements of SKSVM rbf kernel using single image features



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.12: Best improvements of SKSVM rbf kernel using combinations of image features

4.2.2 MKLGL

Another popular method we have previously explored is the MKLGL algorithm. This learner uses multiple kernels to further optimize the hyperplane in an attempt to provide better classification than the standard SKSVM. Our particular approach uses 10 kernels, the first five are RBF kernels,the next three are polynomial kernels, and the final two are linear kernels. After training, this gives a kernel matrix size of $N \times N \times 10$, where N is the input feature length. Once trained, one simply must build a testing matrix in the same fasion, just using the testing data vectors in place of the training data vectors, then use LIBSVM's *svmpredict* function. From here, the process is the same as it was in the SKSVM case; the distance from the hyperplane become the new confidence of the alarm. Tables 4.25 through 4.28 show the results of the MKLGL appraoch.

Table 4.25
Percent AUR Improvemets using Single Image Features with MKLGL

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-70.49	-50.96	**0.57**	-81.62	-66.27	**-39.38**
Mimo VV:	-76.84	**-40.98**	-60.42	**-94.25**	-97.75	-98.02
Set VV:	-24.32	**-16.37**	-42.86	0.62	-8.22	**21.40**
LBP						
Mimo HH:	-85.48	-58.44	**-43.74**	-73.98	**-12.64**	-32.19
Mimo VV:	**-24.44**	-61.82	-70.90	**-83.17**	-100.00	-96.27
Set VV:	-30.40	**-25.03**	-37.81	**13.53**	-31.96	-77.65
LSTAT						
Mimo HH:	**-42.63**	-59.55	-64.65	-80.52	**-45.81**	-70.43
Mimo VV:	**-47.62**	-49.25	-62.50	-58.30	-58.56	**-42.49**
Set VV:	-70.81	**-31.52**	-64.00	**-28.97**	-45.64	-34.42
FFST						
Mimo HH:	-100	-50.84	**-30.56**	-100	-84.04	**-58.35**
Mimo VV:	-100	-58.36	**-53.54**	-100	**-45.75**	-76.32
Set VV:	-100	-55.52	**-46.32**	-100	-52.38	**-30.75**

Table 4.26

Percent AUR Improvemets using Single Image Features with MKLGL and No Phase Adjustment

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG						
Mimo HH:	-65.16	-45.66	**-14.42**	**-76.67**	-92.80	-98.67
Mimo VV:	-64.40	**-54.41**	-64.97	**-94.35**	-100.00	-97.35
Set VV:	-24.32	**-16.37**	-42.86	0.62	-8.22	**21.40**
LBP						
Mimo HH:	-79.64	-68.86	**-48.65**	**-50.82**	-95.16	-89.93
Mimo VV:	**-22.29**	-90.76	-86.97	-82.89	-99.70	**-62.41**
Set VV:	-30.40	**-25.03**	-37.81	**13.53**	-31.96	-77.65
LSTAT						
Mimo HH:	**-56.94**	-92.96	-75.78	-72.44	-17.59	**-16.26**
Mimo VV:	-67.58	-56.26	**-55.76**	-68.43	**-58.16**	-61.41
Set VV:	-70.81	**-31.52**	-64.00	**-28.97**	-45.64	-34.42
FFST						
Mimo HH:	-100	-58.63	**-45.38**	-100	-78.48	**-57.89**
Mimo VV:	-100	-76.14	**-48.64**	-100	-77.28	**-53.39**
Set VV:	-100	**-46.08**	-56.91	-100	**-31.03**	-72.25

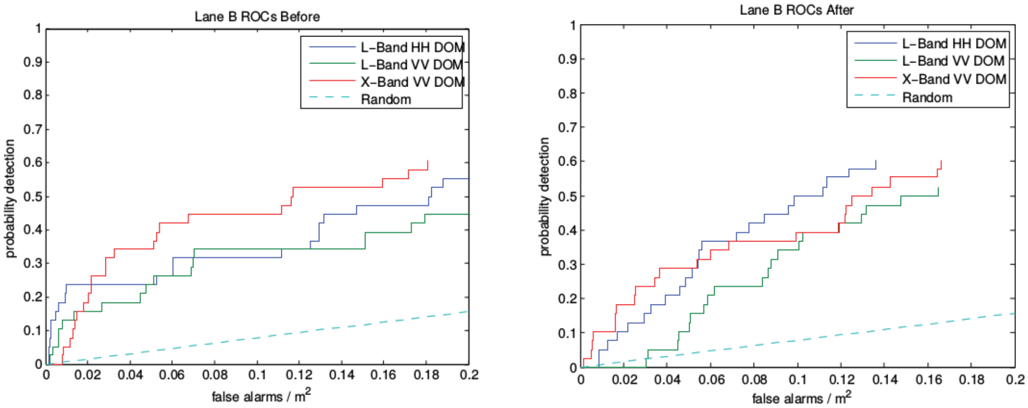
Table 4.27
Percent AUR Improvemets using Combinations of Image Features with MKLGL

	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-70.78	-51.09	**0.42**	-81.76	-66.34	**-39.70**
Mimo VV:	-76.63	**-41.08**	-60.34	**-93.01**	-97.70	-98.02
Set VV:	-24.37	**-16.35**	-42.55	0.81	-7.90	**21.97**
HOG & LSTAT						
Mimo HH:	-70.29	-67.24	**-47.23**	-81.28	**-50.92**	-70.65
Mimo VV:	-79.35	**-55.83**	-61.84	-77.69	-82.58	**-70.29**
Set VV:	-65.12	**-18.97**	-48.05	**-11.20**	-28.41	-18.81
LBP & LSTAT						
Mimo HH:	**-42.51**	-59.44	-64.56	-80.54	**-45.36**	-70.41
Mimo VV:	**-47.73**	-49.48	-62.75	-58.78	-59.98	**-43.23**
Set VV:	-70.74	**-31.30**	-63.96	**-28.72**	-45.39	-34.36
HOG, LBP & LSTAT						
Mimo HH:	-70.29	-67.24	**-47.25**	-81.34	**-50.93**	-70.65
Mimo VV:	-79.25	**-55.85**	-61.84	-77.61	-82.61	**-70.29**
Set VV:	-64.97	**-18.77**	-48.07	**-11.32**	-28.53	-18.81

Table 4.28
Percent AUR Improvemets using Combinations of Image Features with MKLGL and No Phase Adjustment

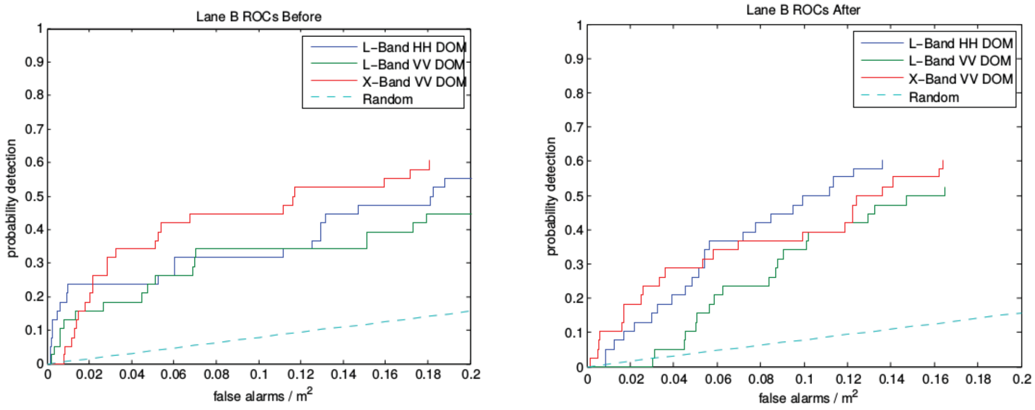
	Train A, Test B			**Train B, Test A**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP						
Mimo HH:	-65.51	-45.86	**-14.63**	**-75.69**	-92.78	-98.68
Mimo VV:	-64.38	**-54.83**	-64.99	**-94.46**	-100.00	-97.38
Set VV:	-24.37	**-16.35**	-42.55	0.81	-7.90	**21.97**
HOG & LSTAT						
Mimo HH:	-66.13	-74.96	**-61.83**	-74.05	**-34.16**	-37.59
Mimo VV:	-76.06	-58.03	**-49.99**	**-77.44**	-80.07	-81.31
Set VV:	-65.12	**-18.97**	-48.05	**-11.20**	-28.41	-18.81
LBP & LSTAT						
Mimo HH:	**-57.46**	-92.96	-75.82	-73.06	-18.10	**-16.35**
Mimo VV:	-67.70	-56.43	**-55.66**	-68.67	**-59.30**	-61.94
Set VV:	-70.74	**-31.30**	-63.96	**-28.72**	-45.39	-34.36
HOG, LBP & LSTAT						
Mimo HH:	-66.61	-74.85	**-61.88**	-74.10	**-34.20**	-37.56
Mimo VV:	-75.94	-58.03	**-49.89**	**-77.36**	-80.07	-81.31
Set VV:	-64.97	**-18.77**	-48.07	**-11.32**	-28.53	-18.81

Unfortunately, the MKLGL fails to outperform the SKSVM. In fact, comparing all the tables in this section, we can actually see that the MKLGL is one of the worst classifiers attempted. It is likely that MKLGL is simply too powerful in conjunction with the CFAR prescreener, which would lead to overtraining and thus poor testing results. This overtraining is likely amplified by the small amount of training data. In the next section, we use data from three different lanes to train each network and then test on a single lane. In that case, we would expect the MKLGL, as well as all the other networks, to perform much better given the larger and more diverse training set.



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.13: Best improvements of MKLGL using single image features



(a) Lane B ROCs Before **(b)** Lane B ROCs After

Figure 4.14: Best improvements of MKLGL using combinations of image features

4.3 Multi-Lane Training for Single Test Lane

Given the size of the data we had, we also explored using multiple lanes to train the classifiers and then testing them on one lane. We use the same two lanes as above, however we also add two more lane, which we will call Lane C and Lane D. We then ran each classifier by training on three of the lanes and testing on the fourth. Again, our main testing criterion is the AUR out to a FAR of 0.1 FAs per meter squared. To get a full comparision of the effects of using multiple lanes to train, we also trained the shallow learners using the same methods. Algorithm parameters remained the same for all learners; this test was simply to find if adding more training data provided any benefits. We also compared the results of both the phase corrected data and the non phase corrected data. Since no parameters were changed in any of the networks, we will forgo unnecessary introductions to each again and simply present the results followed by a discussion of them.

4.3.1 DBN

Table 4.29
Percent AUR Improvemets using Single Image Features with DBN and Phase Adjustment[51]

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train C,B,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	231.86	**283.75**	281.68	**90.64**	19.61	46.63	-48.13	0.42	**5.67**	18.14	**67.75**	11.62
Mimo VV:	305.15	**518.99**	380.84	151.75	**263.55**	219.96	-76.58	-17.13	**29.27**	3.78	51.24	**61.88**
Set VV:	-92.90	512.74	**530.04**	-40.40	**73.30**	49.84	11.53	**28.67**	-5.70	-12.30	6.32	**16.36**
LBP												
Mimo HH:	234.48	**255.52**	102.44	-80.34	-11.72	**2.86**	**-66.04**	-70.09	-69.72	-79.83	**-79.28**	-94.81
Mimo VV:	7.50	-92.43	**106.36**	**-49.25**	-68.00	-96.79	-62.15	-71.82	**-47.81**	-81.17	**-23.87**	-68.28
Set VV:	59.67	**168.25**	144.56	**-29.99**	-62.15	-99.85	-71.93	-47.99	**-33.85**	-77.88	-81.81	**-29.50**
LSTAT												
Mimo HH:	**89.72**	-68.50	-12.09	**-16.59**	-86.72	-89.23	-76.29	-92.03	**-67.82**	**-53.56**	-83.39	-69.20
Mimo VV:	**195.38**	-100.00	47.71	-72.40	-96.50	**-54.98**	**-47.92**	-79.13	-73.28	-61.27	-73.51	**-45.54**
Set VV:	**195.49**	-67.94	111.87	**-57.62**	-74.24	-71.37	**-47.23**	-48.64	-79.76	**-42.54**	-42.70	-72.66
FFST												
Mimo HH:	**51.57**	46.89	33.89	**35.50**	-45.87	-6.51	**-59.27**	-63.19	-67.67	-51.29	**-22.73**	-54.79
Mimo VV:	102.18	26.63	**156.05**	-6.77	-17.36	**37.68**	-71.34	**-61.28**	-72.01	-53.00	-84.47	**-46.81**
Set VV:	**332.11**	211.60	221.08	-68.15	-94.58	**-36.80**	-53.85	-56.13	**-40.12**	-75.66	**-43.68**	-69.92

Table 4.30

Percent AUR Improvemets using Single Image Features with DBN and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	-77.44	-97.74	**38.80**	-30.66	21.61	**26.23**	-56.29	3.37	**11.48**	-87.79	**-0.36**	-47.66
Mimo VV:	-66.96	-100.00	**68.98**	-63.60	92.03	**117.60**	1.13	-92.46	**-30.00**	-60.61	**37.85**	-1.25
Set VV:	-74.65	**-42.32**	-100.00	**-72.09**	-99.99	-92.69	**-82.53**	-92.35	-93.27	-91.04	-98.99	**4.31**
LBP												
Mimo HH:	**-57.68**	-79.45	-65.44	**-16.38**	-56.72	-63.54	**-16.40**	-27.09	-46.81	-88.25	-75.69	**-72.79**
Mimo VV:	-52.79	-72.45	**-49.22**	-56.07	0.06	**10.93**	**-69.33**	-84.75	-88.74	**-37.23**	-54.78	-65.62
Set VV:	**-12.20**	-63.53	-93.45	**-9.83**	-19.91	-40.50	**-15.74**	-78.60	-54.07	-74.68	**-39.34**	-75.95
LSTAT												
Mimo HH:	**-14.99**	-78.74	-89.02	**-50.46**	-80.29	-96.10	**-56.33**	-83.55	-72.69	**-63.31**	-75.02	-81.64
Mimo VV:	-78.63	-88.23	**-70.52**	**-62.31**	-89.11	-95.64	**-56.15**	-65.60	-74.00	-64.93	-84.81	**-58.55**
Set VV:	-35.96	**-13.25**	-59.36	-62.65	**-55.97**	-65.53	-20.85	**-18.50**	-49.93	-64.84	**-56.90**	-71.41
FFST												
Mimo HH:	-76.56	**-41.43**	-57.08	-83.70	**-10.38**	-43.18	-63.05	-63.11	**-60.16**	**-35.11**	-57.79	-55.51
Mimo VV:	-53.38	-75.55	**-45.11**	-74.99	-54.71	**8.32**	-65.63	-66.01	**-54.02**	**-31.61**	-51.07	-32.39
Set VV:	-24.39	**31.36**	28.60	-21.57	**-18.36**	-24.00	**14.72**	-46.55	-50.47	**-31.09**	-82.41	-47.15

Table 4.31
Percent AUR Improvemets using Combinations of Image Features with DBN and Phase Adjustment[51]

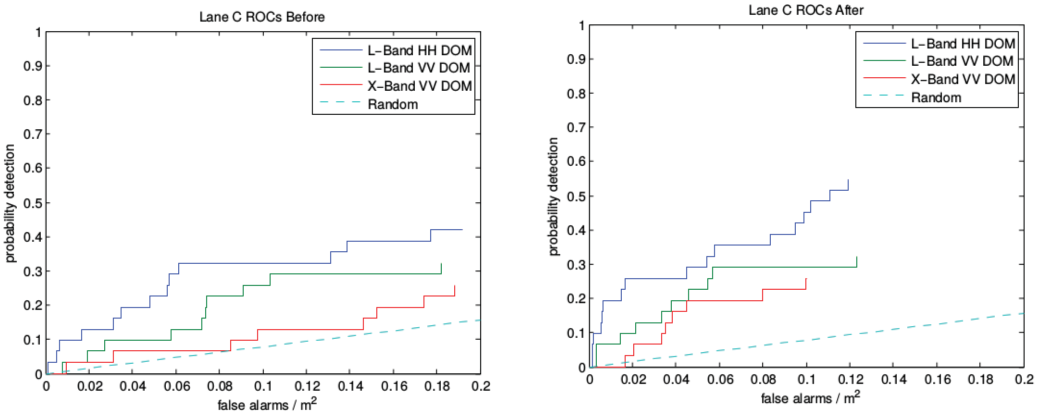
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	94.75	126.19	**231.12**	88.07	**131.30**	125.96	-94.47	-53.42	**11.61**	1.99	-36.17	**63.21**
Mimo VV:	355.44	**518.67**	385.07	206.32	327.83	**344.22**	-55.90	-28.35	**21.31**	2.02	22.38	**78.51**
Set VV:	308.43	499.35	**849.49**	1.86	83.08	**88.97**	-1.54	41.02	**54.18**	-28.83	**9.03**	0.22
HOG & LSTAT												
Mimo HH:	**220.00**	125.39	39.46	**39.80**	-98.08	-99.00	**-30.75**	-48.40	-63.63	**-8.06**	-20.60	-17.64
Mimo VV:	**295.63**	109.56	91.04	**22.44**	12.73	-54.53	**-6.70**	-56.44	-45.14	**-32.62**	-37.70	-38.71
Set VV:	**291.85**	101.93	162.88	**-49.41**	-50.96	-52.62	-6.11	**1.31**	-11.22	**-24.63**	-43.51	-54.98
LBP & LSTAT												
Mimo HH:	**111.17**	44.02	-17.11	**-47.77**	-35.83	-62.69	**-56.01**	-76.45	-83.82	-40.47	**-40.08**	-71.63
Mimo VV:	**135.24**	-29.93	-80.78	-48.13	**32.31**	5.26	**-33.69**	-65.83	-75.94	**-50.03**	-72.11	-60.46
Set VV:	174.62	**204.50**	99.20	-75.25	-62.21	**-54.57**	-7.49	**-4.30**	-2.74	**-60.03**	-68.03	-65.86
HOG, LBP & LSTAT												
Mimo HH:	**93.99**	78.86	28.38	**-2.67**	-29.39	-60.81	**-57.03**	-72.92	-73.67	-27.30	**-16.46**	-35.79
Mimo VV:	**169.42**	-12.38	-38.15	-43.46	**55.73**	34.32	**-44.87**	-64.68	-64.91	-47.16	-46.89	**-39.67**
Set VV:	80.80	**259.97**	161.69	-56.27	**-47.70**	-48.00	-11.50	-1.79	**6.22**	-60.61	-55.26	**-52.68**

Table 4.32
Percent AUR Improvemets using Combinations of Image Features with DBN and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	-70.13	2.02	**35.74**	**76.70**	56.14	63.02	-63.42	**-16.63**	-20.22	-20.22	30.69	**57.43**
Mimo VV:	37.35	**74.71**	72.83	47.12	115.79	**125.17**	-14.70	-2.93	**20.02**	-6.43	47.60	**59.19**
Set VV:	-0.84	109.88	**152.57**	14.85	**90.24**	69.12	-67.60	43.15	**53.55**	-27.21	-29.14	**19.50**
HOG & LSTAT												
Mimo HH:	**-5.09**	-47.11	-50.79	**-18.45**	-83.08	-99.99	**-56.72**	-60.36	-64.40	-44.75	-39.35	**-31.47**
Mimo VV:	**-16.64**	-52.22	-46.89	**-55.33**	-85.68	-70.72	**-16.56**	-48.74	-58.49	-54.20	-37.51	**-17.84**
Set VV:	**24.66**	-19.95	-24.63	-64.97	-61.57	**-57.55**	-11.22	**-5.40**	-22.19	-40.68	-44.21	**-38.70**
LBP & LSTAT												
Mimo HH:	**-33.88**	-70.75	-86.41	**-45.35**	-46.18	-79.82	**-52.75**	-80.08	-82.54	**-57.07**	-69.34	-58.48
Mimo VV:	**-62.81**	-64.39	-89.81	-70.47	**-33.10**	-55.31	**-32.83**	-62.77	-74.22	-72.73	**-69.24**	-72.48
Set VV:	-9.79	**9.04**	-37.04	-71.59	**-58.22**	-58.61	-15.95	-0.45	**0.77**	**-56.32**	-62.48	-67.27
HOG, LBP & LSTAT												
Mimo HH:	**-27.58**	-71.73	-87.02	**-5.87**	-55.11	-64.43	**-55.24**	-75.13	-78.49	-57.26	-62.45	**-33.43**
Mimo VV:	**-29.78**	-50.35	-74.47	-58.75	**3.10**	-58.15	**-22.70**	-61.68	-63.37	-54.40	-63.70	**-43.95**
Set VV:	0.90	**9.67**	-25.26	-73.54	-62.74	**-45.52**	-13.95	**1.41**	-0.11	-55.54	-56.04	**-52.74**

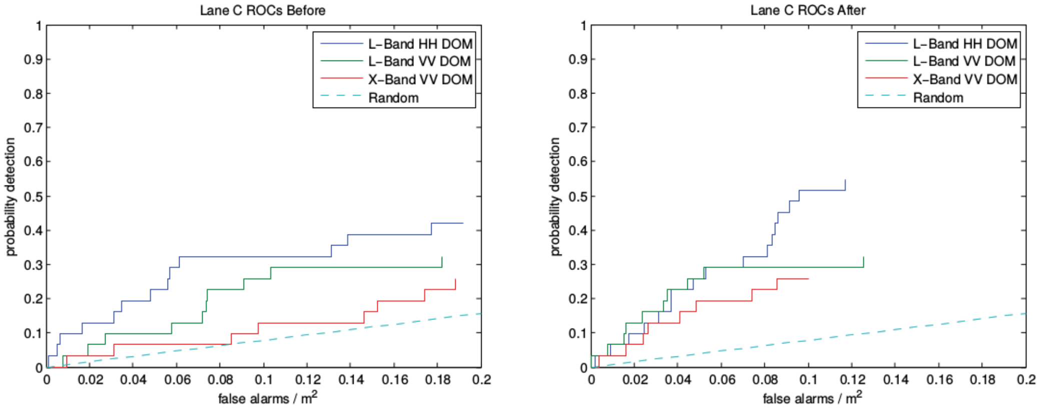
Comparing Tables 4.29, 4.30, 4.31, and 4.32, one can easily see that by using multiple lanes for training, we generally get a much better performance from the DBN as opposed to training on a single lane. The most likely reason behind this is that the increased training data allows the DBN to learn a better representation of the false alarms and thus will do a worse job at reconstructing the targets. This then provides a larger RMSE which directly translates into a larger confidence for the target locations and thus a lower confidence for the false alarms. These results are further evident when looking at Figures 4.15 and 4.16.

We can also see that some lanes are better for testing than others. For instance, it seems that in the single feature case, training on A, B, and D and then testing on C yields much better results than any other combination of lanes. This result is also seen when combinations of features are used. Looking at Fig. 4.15(a), which is the prescreener ROC for Lane C, we see that Lane C has much to gain from testing with these networks. Also keeping in line with the single train, single test results, we see that the phase adjusted data generally outperforms the non phase adjusted data.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.15: Best improvements of DBN using single image features[51]



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.16: Best improvements of DBN using combinations of image features[51]

4.3.2 SDAE-NN

Table 4.33
Percent AUR Improvemets using Single Image Features with SDAE-NN and Phase Adjustment[50]

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	280.63	**286.02**	157.06	100.88	58.56	**110.21**	-34.74	**-0.13**	24.65	11.98	**49.38**	36.65
Mimo VV:	253.28	365.10	**450.64**	100.35	**317.22**	315.88	-8.28	30.02	**44.96**	1.98	41.64	**73.38**
Set VV:	121.52	467.38	**682.43**	16.00	56.17	**97.29**	-7.38	-0.20	**56.12**	-56.53	**8.25**	-23.45
LBP												
Mimo HH:	**203.23**	203.23	203.23	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00
Mimo VV:	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00
Set VV:	**203.23**	203.23	203.23	**0.00**	0.00	0.00	**1.82**	1.82	1.82	**0.34**	0.34	0.34
LSTAT												
Mimo HH:	-34.95	56.36	**146.68**	**-19.36**	-55.56	-57.69	**-53.14**	-95.19	-70.41	-59.43	-52.71	**-32.51**
Mimo VV:	**171.65**	-53.80	-39.17	-62.55	**0.92**	-100.00	-71.49	**-42.97**	-94.10	-37.01	-51.70	**-12.41**
Set VV:	78.07	-100.00	**173.08**	-87.09	-88.51	**9.62**	-59.44	**-41.17**	-44.24	**-47.71**	-63.51	-73.51
FFST												
Mimo HH:	0.00	**203.23**	-100.00	**203.23**	-100.00	-100.00	0.00	-100.00	**0.00**	-100.00	-100.00	**0.00**
Mimo VV:	0.00	-100.00	**203.23**	**-100.00**	-100.00	-100.00	-100.00	-100.00	**0.00**	-100.00	-100.00	**-100.00**
Set VV:	0.00	-100.00	**203.23**	**203.23**	-100.00	0.00	0.00	1.82	**1.82**	-100.00	-100.00	**-100.00**

Table 4.34

Percent AUR Improvemets using Single Image Features with SDAE-NN and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	15.47	**28.60**	9.12	32.40	**71.27**	70.44	-23.59	-7.51	**28.49**	-3.84	38.08	**52.35**
Mimo VV:	53.54	**81.91**	63.22	20.74	**135.33**	0.00	-13.83	32.54	**48.42**	44.99	38.28	**70.16**
Set VV:	-7.99	95.58	**129.59**	36.02	42.49	**73.35**	-16.31	22.05	**31.01**	19.72	-100.00	**36.87**
LBP												
Mimo HH:	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00	-100.00	-100.00	**-100.00**
Mimo VV:	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00	0.00	0.00	**0.00**
Set VV:	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00	0.34	0.34	**0.34**
LSTAT												
Mimo HH:	**-12.99**	-19.92	-24.57	**51.44**	-38.64	-28.26	**-34.04**	-36.41	-34.81	**-6.38**	-41.86	-34.94
Mimo VV:	-48.36	**41.60**	-3.60	**-0.41**	-9.87	-2.42	**-16.06**	-41.54	-29.75	-28.93	-33.33	**-16.64**
Set VV:	-5.05	-16.40	**22.11**	-96.88	-19.67	**20.69**	**-3.52**	-6.03	-6.43	-15.53	-17.44	**-7.21**
FFST												
Mimo HH:	**0**	-100	-100	**0**	-100	-100	0	0	**0**	-100	**0**	0
Mimo VV:	**0**	-100	-100	**0**	0	0	0	-100	**0**	**0**	-100	0
Set VV:	**0**	-100	-100	**0**	0	-100	-100	1.82	**1.82**	**0.34**	0.34	-100

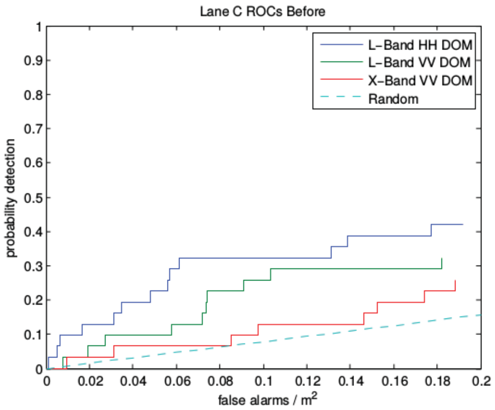
Table 4.35
Percent AUR Improvemets using Combinations of Image Features with SDAE-NN and Phase Adjustment[50]

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	**272.98**	144.20	155.03	64.79	133.42	**118.61**	-43.45	12.79	**25.17**	13.57	**52.82**	38.03
Mimo VV:	92.01	369.65	**448.69**	115.40	335.38	**427.92**	-22.38	23.88	**41.91**	-3.83	20.72	**81.23**
Set VV:	367.18	461.64	**707.30**	22.96	3.54	**86.05**	-14.81	-45.18	**21.58**	-56.33	11.60	**31.34**
HOG & LSTAT												
Mimo HH:	105.62	224.97	**226.42**	**12.96**	-26.78	-2.82	-45.54	-33.68	**-28.28**	-45.91	-24.37	**-24.14**
Mimo VV:	310.49	267.54	**443.49**	**91.63**	28.12	67.27	**-1.03**	-65.17	-35.65	-24.77	-19.71	**46.69**
Set VV:	101.67	**598.18**	433.50	-54.77	**29.44**	17.17	-17.63	-40.89	**5.84**	-41.22	**9.74**	-65.47
LBP & LSTAT												
Mimo HH:	28.78	11.43	**199.72**	**-45.78**	-52.66	-78.92	**-54.30**	-69.40	-64.57	-46.72	-71.57	**-23.87**
Mimo VV:	**200.95**	-100.00	100.05	-66.77	-75.38	**-9.99**	**-67.87**	-100.00	-95.36	**-45.96**	-52.51	-48.65
Set VV:	237.66	1.47	**247.13**	-67.62	**-37.83**	-60.86	-66.21	**4.71**	-75.74	-70.86	**-40.41**	-77.04
HOG, LBP & LSTAT												
Mimo HH:	103.46	**116.17**	108.80	**15.31**	-52.23	-7.07	**-29.80**	-53.49	-50.37	-42.98	-62.49	**-24.91**
Mimo VV:	299.77	303.34	**509.20**	-74.20	**80.47**	-32.53	-54.13	**-13.82**	-57.51	-22.63	-19.61	**30.21**
Set VV:	247.77	-100.00	**362.17**	-34.67	57.77	**95.25**	-11.20	-3.59	**5.63**	-51.82	**8.18**	-23.42

Table 4.36
Percent AUR Improvemets using Combinations of Image Features with SDAE-NN and No Phase Adjustment

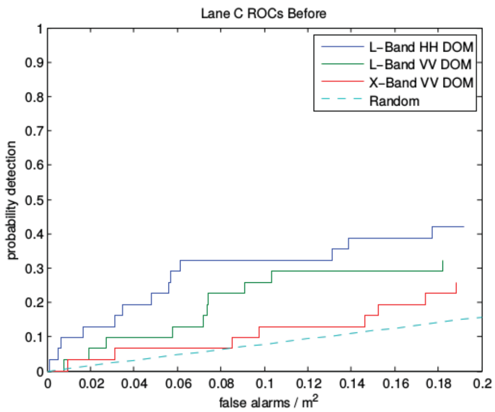
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	15.58	**33.21**	21.32	37.06	**81.93**	-16.56	-13.45	-5.70	**27.24**	-16.96	38.00	**64.98**
Mimo VV:	35.95	**82.10**	66.97	25.73	136.24	**148.34**	-33.65	19.13	**36.61**	15.72	29.92	**63.45**
Set VV:	6.76	110.48	**142.48**	24.57	35.29	**103.86**	4.06	**46.91**	31.76	18.74	7.41	**36.25**
HOG & LSTAT												
Mimo HH:	-10.96	17.00	**37.85**	46.48	13.47	**53.35**	-17.11	-4.35	**17.04**	-16.70	15.97	**30.88**
Mimo VV:	1.74	82.40	**86.85**	-6.90	46.38	**99.92**	-17.27	14.20	**14.39**	4.50	**26.48**	16.95
Set VV:	**136.98**	82.91	97.66	-29.65	-2.79	**69.90**	29.54	26.69	**37.95**	-6.88	-8.48	**16.72**
LBP & LSTAT												
Mimo HH:	**-14.73**	-32.03	-15.77	**19.92**	-47.63	-43.88	**-38.21**	-34.82	-39.19	**-12.98**	-25.75	-20.54
Mimo VV:	-33.83	41.60	**43.28**	-36.53	-18.21	**28.72**	**-13.76**	-46.37	-35.71	-38.06	-34.02	**-8.53**
Set VV:	**67.77**	-26.28	-10.96	-66.75	-29.50	**40.50**	-18.16	-28.47	**3.13**	-58.64	-10.06	**-7.15**
HOG, LBP & LSTAT												
Mimo HH:	8.96	-20.42	**23.03**	61.61	14.58	**63.05**	-11.34	-22.14	**7.41**	-8.72	8.58	**33.92**
Mimo VV:	22.70	**125.70**	103.26	14.98	39.97	**140.83**	-9.59	9.15	**29.59**	9.80	**24.35**	13.35
Set VV:	74.08	57.68	**101.05**	-5.27	22.73	**64.00**	11.02	**46.22**	25.88	-15.33	**20.34**	13.23

Much like the DBN case, we see a massive improvement over the single-train-single-test results in the SDAE-NN. The HOG remains the best single feature and that combinations of features still outperform single features. The interesting difference we see in the SDAE-NN is that when testing on Lanes A and B, the non phase adjusted data generally outperforms the phase adjusted data. This may be due to a combination of the SDAE being a more forgiving network than the DBN. In support of this, we see that, just like the single-train-single-test results the SDAE-NN outperforms the DBN in terms of AUR improvement. Comparing the AUR improvements in Figs. 4.15 and 4.16 with those in Figs. 4.17 and 4.18 further shows the abilities of the SDAE-NN to dramatically improve the prescreener results of Lane C over the DBN.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.17: Best improvements of SDAE-NN using single image features[50]



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.18: Best improvements of SDAE-NN using combinations of image features[50]

4.3.3 SDAE-SVM

Table 4.37

Percent AUR Improvemets using Single Image Features with SDAE-SVM and Phase Adjustment[50]

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	**48.52**	33.35	-27.22	**20.99**	3.77	-18.56	-74.58	**-16.68**	-37.44	-35.29	-80.43	**-14.79**
Mimo VV:	-19.85	**268.33**	-20.50	16.16	-25.21	**61.44**	-71.71	**-25.00**	-84.05	-48.54	-80.90	**12.26**
Set VV:	358.17	**404.89**	307.43	-50.91	-52.28	**30.62**	-6.08	**31.99**	17.64	-32.81	-42.18	**-28.43**
LBP												
Mimo HH:	220.74	67.58	**235.16**	23.09	**157.84**	73.15	**-44.19**	-58.72	-51.91	**-0.33**	-29.24	-43.08
Mimo VV:	312.86	309.66	**442.93**	94.09	200.93	**340.52**	**-53.92**	-72.32	-58.97	-21.53	-20.23	**5.66**
Set VV:	150.85	550.18	**711.13**	-2.00	27.46	**55.25**	-16.12	7.89	**35.78**	-46.86	**9.03**	-5.79
LSTAT												
Mimo HH:	**110.76**	53.98	63.19	**73.22**	-38.55	25.44	-46.75	-76.53	**-38.09**	-78.82	**-46.41**	-54.20
Mimo VV:	48.73	32.85	**69.86**	11.72	-8.54	**22.90**	-76.96	-78.83	**-33.72**	-86.73	**-45.49**	-56.30
Set VV:	200.04	11.12	**281.56**	-76.08	**-51.95**	-55.79	**9.64**	-38.62	-44.07	-60.46	-69.48	**-58.43**
FFST												
Mimo HH:	**31.50**	-100.00	-100.00	**-31.56**	-100.00	-100.00	**-80.92**	-100.00	-100.00	**-54.75**	-100.00	-100.00
Mimo VV:	**148.90**	-100.00	-100.00	**25.73**	-100.00	-100.00	**-55.70**	-100.00	-100.00	**-62.42**	-100.00	-100.00
Set VV:	**150.30**	-100.00	-100.00	**-56.85**	-100.00	-100.00	**-40.22**	-100.00	-100.00	**-100.00**	-100.00	-100.00

Table 4.38
Percent AUR Improvemets using Single Image Features with SDAE-SVM and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	-17.52	-89.30	**-55.72**	-38.77	-56.54	**-27.75**	-64.68	**-32.45**	-81.86	**-7.05**	-60.43	-25.88
Mimo VV:	-52.51	-54.50	**-34.38**	17.67	-34.40	**71.25**	**-19.03**	-68.13	-46.44	-65.21	-77.84	**3.63**
Set VV:	-3.76	81.23	**98.53**	-49.50	-4.95	**45.86**	8.86	9.78	**16.61**	-29.05	**0.36**	-2.49
LBP												
Mimo HH:	-34.26	-20.46	**-4.81**	**22.28**	-54.44	-2.31	**-22.82**	-73.65	-37.65	-24.17	**-12.93**	-32.30
Mimo VV:	-1.27	-7.54	**25.70**	-33.35	-5.57	**83.75**	-51.72	**-51.54**	-54.85	-76.95	-25.45	**0.49**
Set VV:	108.80	62.93	**257.29**	46.19	52.04	**52.52**	-4.67	-2.13	**34.53**	-17.65	**22.54**	14.32
LSTAT												
Mimo HH:	-96.92	**-31.78**	-67.91	**-13.07**	-32.48	-14.28	-41.48	**-36.93**	-83.51	**-42.32**	-51.99	-44.36
Mimo VV:	-32.75	-48.26	**12.55**	**-28.64**	-65.67	-41.45	-61.55	-60.31	**-51.13**	**-24.18**	-41.86	-69.99
Set VV:	-6.34	**39.74**	-3.76	**-13.05**	-44.54	-74.92	**-30.21**	-34.10	-38.83	-71.08	-63.00	**-59.06**
FFST												
Mimo HH:	**-6.56**	-100.00	-100.00	**-29.88**	-100.00	-100.00	**-63.34**	-100.00	-100.00	**-70.84**	-100.00	-100.00
Mimo VV:	**-48.84**	-100.00	-100.00	**-34.14**	-100.00	-100.00	**-56.50**	-100.00	-100.00	**-63.61**	-100.00	-85.02
Set VV:	**-48.84**	-100.00	-100.00	**-26.82**	-100.00	-100.00	**-49.35**	-100.00	-100.00	**-100.00**	-100.00	-100.00

Table 4.39

Percent AUR Improvemets using Combinations of Image Features with SDAE-SVM and Phase Adjustment[50]

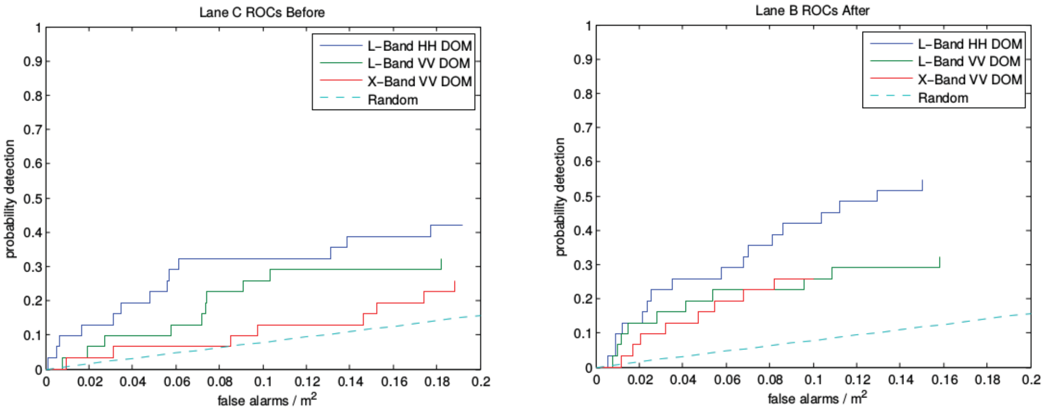
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	133.45	**145.87**	82.41	16.93	**63.46**	56.51	-84.61	-18.96	**0.63**	-55.02	-68.47	**5.55**
Mimo VV:	84.26	31.23	**347.92**	-17.43	84.31	**167.72**	-87.85	**1.30**	-35.33	-16.94	6.27	**44.93**
Set VV:	127.90	468.02	**519.75**	-0.73	17.29	**40.26**	-38.27	**24.78**	12.41	-31.27	**-20.10**	-45.18
HOG & LSTAT												
Mimo HH:	-7.99	**75.89**	-26.39	-57.50	-57.89	**17.43**	-52.45	-83.54	**-47.88**	-27.72	-21.52	**-6.36**
Mimo VV:	106.31	**161.34**	149.31	-10.01	**131.08**	-67.87	-76.17	-60.90	**-50.77**	-76.65	-20.21	**-10.06**
Set VV:	**291.48**	51.29	232.01	-79.00	-69.94	**-4.73**	-41.98	-21.63	**15.29**	-64.96	**15.25**	-5.54
LBP & LSTAT												
Mimo HH:	22.48	**33.91**	29.85	-33.64	-40.19	**4.06**	**-68.76**	-79.18	-75.12	-71.03	**-43.56**	-49.68
Mimo VV:	128.27	34.11	**149.27**	**-4.01**	-36.21	-79.73	-84.94	**-48.72**	-27.52	-43.92	-39.32	**-38.56**
Set VV:	-6.73	44.37	**46.74**	**-7.45**	-34.85	-11.88	-60.90	-17.97	**9.74**	-61.44	**-45.08**	-61.66
HOG, LBP & LSTAT												
Mimo HH:	-13.21	**78.33**	22.59	**-2.95**	-28.77	-30.79	-89.43	-78.76	**-77.40**	-76.86	-63.39	**-71.85**
Mimo VV:	**52.35**	25.05	-63.18	**75.92**	49.89	35.92	-84.28	**-51.79**	-76.35	-29.39	-24.52	**-4.54**
Set VV:	-93.53	171.26	**200.04**	-64.68	-35.05	**11.54**	-55.61	10.42	**31.87**	-60.46	-15.39	**16.61**

Table 4.40
Percent AUR Improvemets using Combinations of Image Features with SDAE-SVM and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	-69.22	**-36.64**	-92.05	-93.40	**91.26**	-0.83	-66.90	**-42.23**	-44.61	-67.14	**-32.93**	46.15
Mimo VV:	64.63	25.39	**51.57**	-2.81	**97.36**	44.14	-90.80	-40.49	**12.67**	-73.92	-42.24	**38.07**
Set VV:	55.57	158.04	**193.15**	-21.27	-8.60	**40.30**	-3.52	-8.59	**-0.01**	-28.92	-18.67	**6.70**
HOG & LSTAT												
Mimo HH:	-81.21	**-67.31**	-93.23	-32.49	-88.51	**17.63**	-89.11	-65.27	**-21.00**	**-46.70**	-57.57	-87.44
Mimo VV:	**-15.33**	-57.77	-74.35	-52.26	-67.58	**17.86**	-73.61	**-47.23**	-54.55	-87.80	-84.52	**-61.74**
Set VV:	17.69	67.35	**193.33**	4.95	-25.80	**73.01**	-15.05	10.99	**10.90**	-76.90	**5.86**	-50.48
LBP & LSTAT												
Mimo HH:	-50.18	-61.74	**-0.59**	-24.66	-28.07	**-19.65**	-81.72	**-58.58**	-73.73	-45.25	-44.19	**-10.48**
Mimo VV:	-88.99	**11.07**	-22.63	**-21.97**	-40.87	-25.65	**-30.55**	-63.28	-56.47	-58.73	-39.79	**-34.23**
Set VV:	-64.64	-15.95	**-10.72**	-62.34	**-24.73**	-63.47	-61.96	**-7.34**	-57.46	-50.12	-64.04	**-36.18**
HOG, LBP & LSTAT												
Mimo HH:	-45.46	-90.19	**-40.42**	**-23.68**	-44.86	-95.76	-75.12	-70.86	**-55.14**	-58.00	-15.25	**11.82**
Mimo VV:	-99.69	-59.01	**-20.20**	-38.61	-19.76	**91.19**	**-7.85**	-75.74	-55.23	-71.36	-63.80	**-8.26**
Set VV:	-100.00	**164.56**	49.50	-81.38	**15.18**	8.48	-76.98	**-46.66**	-56.30	-44.88	**9.21**	-0.66

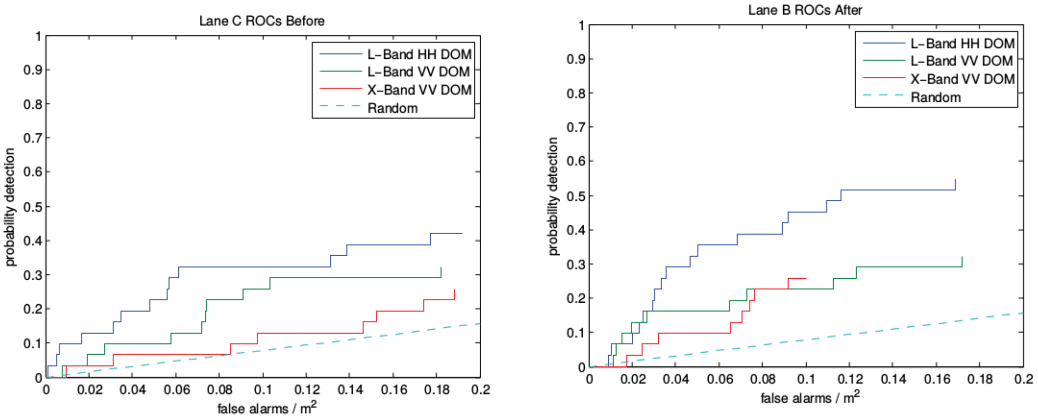
Keeping consistent with the results thus far, the multi-lane SDAE-SVM also outperforms its single-lane trained counterpart by a good margin. Unlike the SDAE-NN, the LBP is actually the strongest single feature for the SDAE-SVM. In fact, the LBP alone actually outperforms the HOG and LBP combination, which is the strongest combination of features. This is a strange occurance in this set. It does not seem logical that a single feature would provide better information than a combination of features. Perhaps the addition of the other features causes the SDAE to slightly overtrain, which is then amplified by the SVM and thus the testing classification rates are lower.

Also unlike the SDAE-NN, the phase adjusted data performs better in general overall. Even the combinations of image features tested on Lanes A and B, the phase adjusted data performs better than the non phase adjusted data. By comparing both forms of the SDAEs, we see that the SDAE-SVM is the better overall performer for single image features, while the SDAE-NN seems to be better for combinations of image features. This difference can likely be atttributed to how the finetuning mechanisms of each approach work. Adding more data in the form of combinations of image features seems to be helpful when using a NNs backpropagation to finetune the architechure. Likewise, it would appear that using solitary image features allows the SVM to perform better when finding a hyperplane to seperate the classes. Figures 4.19 and 4.20, show the best improvements of the single and combination features of the SDAE-SVM respectively.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.19: Best improvements of SDAE-SVM using single image features[50]



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.20: Best improvements of SDAE-SVM using combinations of image features[50]

4.3.4 CNN

Table 4.41

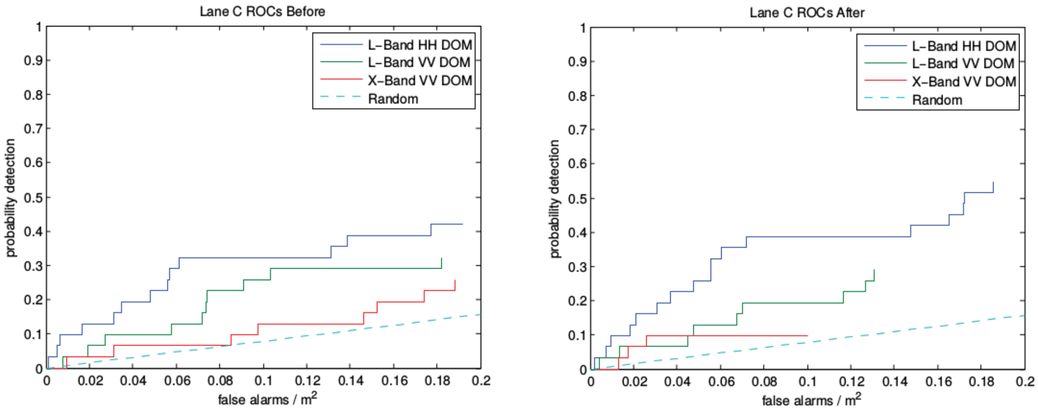
Percent AUR Improvemets Using CNN

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
Imglet												
Mimo HH:	-100.00	**216.49**	203.23	0.00	**0.60**	0.00	-5.27	-4.66	**0.00**	**0.00**	0.00	0.00
Mimo VV:	-100.00	-100.00	**203.23**	-100.00	**0.00**	0.00	**-93.99**	-100.00	-100.00	**0.00**	-100.00	0.00
Set VV:	**221.72**	-100.00	-100.00	-11.40	**0.00**	0.00	-2.04	**1.82**	-100.00	**-24.64**	-76.29	-100.00
HOG												
Mimo HH:	49.59	134.71	**203.23**	**-7.38**	-12.01	-100.00	-48.89	-100.00	**0.00**	-30.93	-100.00	**-16.39**
Mimo VV:	156.37	13.91	**206.48**	**52.68**	-100.00	-100.00	-75.10	-47.64	**-24.01**	-27.99	**0.00**	-60.03
Set VV:	**315.36**	282.93	203.23	-100.00	**-15.92**	-100.00	**1.82**	-19.98	-100.00	**0.34**	0.34	0.34
LBP												
Mimo HH:	**203.23**	203.23	203.23	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00
Mimo VV:	**203.23**	203.23	203.23	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00
Set VV:	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**4.39**	4.39	-0.03	**-100.00**	-100.00	-100.00
FFST												
Mimo HH:	-100.00	**0.00**	-100.00	**0.00**	0.00	-100.00	-100.00	-100.00	**0.00**	**-5.26**	-5.26	-100.00
Mimo VV:	**-100.00**	-100.00	-100.00	**0.00**	-100.00	-100.00	-100.00	**0.00**	0.00	-100.00	-100.00	**-5.26**
Set VV:	-100.00	**0.00**	0.00	-100.00	-100.00	**0.00**	**1.82**	-100.00	-100.00	**-4.94**	-4.94	-4.94

Table 4.42
Percent AUR Improvemets Using CNN and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
Imglet												
Mimo HH:	-100.00	**0.00**	-100.00	0.00	**11.09**	-100.00	-100.00	-100.00	**0.00**	-12.43	**-5.26**	-100.00
Mimo VV:	-100.00	**-100.00**	-100.00	**-12.71**	-100.00	-100.00	-100.00	**0.00**	0.00	-100.00	-100.00	**-5.26**
Set VV:	42.48	**0.00**	0.00	-22.20	-100.00	**0.00**	**-27.13**	-100.00	-100.00	-15.02	-15.34	**-4.94**
HOG												
Mimo HH:	-32.30	-48.50	**0.00**	-48.01	**7.77**	-100.00	**0.00**	-58.37	-100.00	**-12.97**	-52.21	-54.41
Mimo VV:	-18.93	-100.00	**0.00**	0.00	**0.00**	0.00	**-3.44**	-58.40	-12.85	**-5.26**	-66.10	-100.00
Set VV:	-25.26	**38.06**	-100.00	-100.00	**-48.54**	-64.66	-100.00	**17.70**	1.82	**-4.25**	-100.00	-100.00
LBP												
Mimo HH:	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**-100.00**	-100.00	-100.00
Mimo VV:	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00	**0.00**	0.00	0.00	**-100.00**	-100.00	-100.00
Set VV:	**-100.00**	-100.00	-100.00	1.41	**1.80**	1.80	**-100.00**	-100.00	-100.00	**-4.94**	-4.94	-4.94
FFST												
Mimo HH:	-100.00	**0.00**	0.00	**0.00**	0.00	-100.00	-100.00	-100.00	**0.00**	**-5.26**	-5.26	-100.00
Mimo VV:	-100.00	-100.00	**0.00**	**14.23**	-100.00	-100.00	-100.00	**0.00**	-100.00	-100.00	-100.00	**-5.26**
Set VV:	**27.52**	20.82	0.00	**9.82**	-100.00	0.00	-58.44	**-10.17**	-100.00	-49.35	**-5.51**	-100.00

While the multi-lane CNN yields much better results than the single-lane verison, they are still disappointing. Considering how long the CNN takes to run a single cell (see Table 4.57), the performance is even less desirable. This may be due several factors. Perhaps the layer size, mapping size, and sampling rate are not yet optimal. Perhaps the network is over or under training. Possibly the data sampling could use improvement. Without more time to explore these options, we cannot be certain. Due to this, much of the future work section is dedicated to improving the CNNs.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.21: Best improvements of SDAE-NN using single image features

4.3.5 SKSVM

Table 4.43
Percent AUR Improvemets using Single Image Features with SKSVM Linear Kernel and Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	91.99	**219.42**	214.94	-3.21	**31.49**	37.72	-35.30	-17.34	**-8.34**	-20.27	17.66	**34.53**
Mimo VV:	120.71	**468.28**	289.78	16.73	205.67	**206.40**	-10.53	-9.98	**-8.13**	-29.88	**59.47**	48.68
Set VV:	170.62	681.16	**702.75**	-3.71	-6.22	**33.59**	-8.34	2.19	**15.89**	-53.69	-18.86	**11.97**
LBP												
Mimo HH:	**205.85**	200.34	120.04	**2.07**	-64.26	-49.51	**-21.52**	-43.67	-47.71	**11.93**	-23.93	-39.50
Mimo VV:	93.68	7.78	**285.56**	**112.82**	29.26	60.01	**-47.67**	-68.83	-51.71	-17.67	-14.14	**-9.50**
Set VV:	-19.39	69.87	**313.62**	-41.43	5.64	**12.42**	-32.35	-14.02	**1.02**	-39.05	-46.32	**-10.66**
LSTAT												
Mimo HH:	129.12	198.32	**159.34**	76.00	**78.82**	53.72	-23.76	**-20.87**	-48.62	-66.88	**-37.58**	-46.07
Mimo VV:	264.75	**302.78**	61.09	54.86	**116.76**	78.24	-50.11	**-39.33**	-43.16	**-46.35**	-49.65	-47.19
Set VV:	9.21	**190.47**	-1.72	-39.29	**30.77**	10.13	-52.85	**-25.40**	-39.72	-58.39	-39.54	**-37.83**
FFST												
Mimo HH:	11.36	70.14	**135.15**	**39.46**	-16.54	-7.44	-56.48	-62.50	**-48.86**	-51.27	-56.67	**-49.27**
Mimo VV:	-71.26	**8.10**	-8.20	60.54	69.16	**139.59**	-57.24	-57.83	**-49.03**	-60.04	-64.91	**-44.63**
Set VV:	210.24	**218.53**	133.64	**-30.02**	-94.78	-94.20	**-36.95**	-72.25	-44.85	-77.19	-76.88	**-68.08**

Table 4.44

Percent AUR Improvemets using Single Image Features with SKSVM Linear Kernel and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	-58.83	0.12	**25.46**	-80.62	15.75	**63.02**	-34.26	-14.66	**-7.62**	-13.43	-9.90	**32.57**
Mimo VV:	-47.38	**51.18**	15.62	-6.44	100.64	**115.92**	-26.18	-5.14	**-0.94**	-47.75	37.50	**71.46**
Set VV:	-10.75	157.62	**164.74**	-3.71	-6.22	**33.59**	-8.34	2.19	**15.89**	-53.69	-18.86	**11.97**
LBP												
Mimo HH:	-9.67	-36.59	**-4.25**	**0.61**	-51.31	-40.32	**-20.38**	-32.64	-44.05	**7.59**	0.36	-7.45
Mimo VV:	**40.50**	22.64	2.47	**48.60**	-46.18	-60.75	**-4.84**	-23.16	-59.10	**9.17**	3.88	-35.67
Set VV:	-73.42	-43.98	**36.41**	-41.43	5.64	**12.42**	-32.35	-14.02	**1.02**	-39.05	-46.32	**-10.66**
LSTAT												
Mimo HH:	5.09	**17.46**	-47.88	**21.45**	12.53	-18.17	-40.15	**-26.05**	-40.92	-29.28	**7.48**	-36.42
Mimo VV:	4.53	**58.80**	30.93	-12.13	1.08	**18.17**	-40.98	-46.66	**-36.89**	**5.54**	-50.64	-49.83
Set VV:	-63.98	**-4.21**	-67.59	-39.29	**30.77**	10.13	-52.85	**-25.40**	-39.72	-58.39	-39.54	**-37.83**
FFST												
Mimo HH:	-72.12	-89.87	**-64.24**	-55.01	**-32.93**	-45.94	-73.30	-65.90	**-62.19**	-98.67	**-58.51**	-70.63
Mimo VV:	-45.41	**-24.83**	-75.77	-20.29	**19.72**	-78.16	-71.00	-62.34	**-51.58**	-71.21	-62.78	**-45.72**
Set VV:	2.31	**5.05**	-22.95	**-30.02**	-94.78	-94.20	**-36.95**	-72.25	-44.85	-78.39	-78.09	**-69.76**

Table 4.45

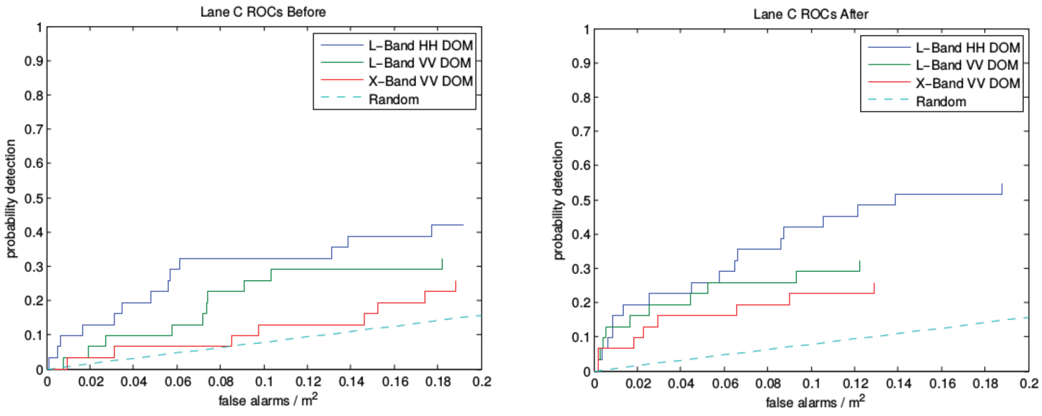
Percent AUR Improvemets using Combinations of Image Features with SKSVM Linear Kernel and Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	197.45	214.06	**218.55**	26.75	22.43	**23.91**	-29.68	-19.31	**-13.27**	**55.46**	22.99	33.76
Mimo VV:	282.96	**476.41**	287.46	124.25	**219.25**	197.36	-23.09	-12.21	**-6.98**	9.49	**56.20**	52.87
Set VV:	62.77	612.93	**681.07**	-51.45	14.76	**36.36**	-31.53	17.23	**17.96**	-38.36	-15.54	**8.90**
HOG & LSTAT												
Mimo HH:	157.41	**272.87**	190.21	79.61	**124.91**	108.59	-28.79	-15.57	**-10.53**	14.51	-4.12	**23.43**
Mimo VV:	297.54	409.77	**464.38**	25.80	178.53	**327.17**	-21.61	-4.29	**14.27**	-13.30	31.09	**43.61**
Set VV:	155.95	**378.02**	283.11	-40.45	**31.88**	31.34	-31.00	-13.42	**10.31**	-45.96	-19.03	**-15.30**
LBP & LSTAT												
Mimo HH:	**282.98**	251.38	227.80	23.20	**60.44**	50.60	**-9.18**	-21.36	-43.51	**30.67**	-30.51	-34.60
Mimo VV:	143.79	**541.42**	149.27	34.30	**95.47**	71.97	**-19.20**	-31.47	-44.12	**14.75**	-32.08	-50.49
Set VV:	-13.11	**285.48**	32.25	8.66	**58.55**	14.46	-42.66	**-22.89**	-38.77	-50.23	-39.21	**-37.80**
HOG, LBP & LSTAT												
Mimo HH:	168.11	**271.72**	190.35	68.47	**122.51**	107.23	-31.74	-16.19	**-10.80**	**25.20**	-3.28	23.34
Mimo VV:	318.02	416.92	**466.33**	21.54	174.67	**327.62**	-20.13	-4.97	**14.35**	-9.17	30.56	**43.71**
Set VV:	217.34	**374.83**	292.67	-36.76	**36.65**	31.64	-22.22	-11.79	**10.24**	-50.39	-19.01	**-14.47**

Table 4.46
Percent AUR Improvemets using Combinations of Image Features with SKSVM Linear Kernel and No Phase Adjustment

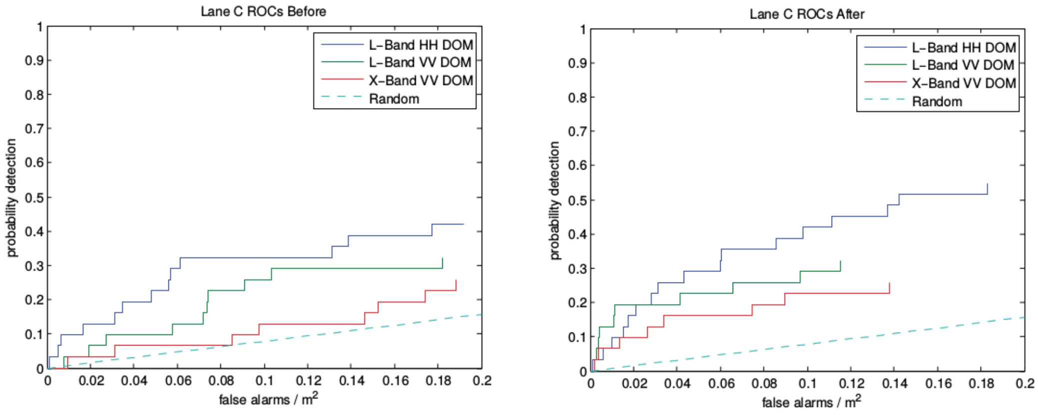
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	-18.92	-6.94	**24.03**	-1.89	-7.11	**61.05**	-17.05	-13.61	**-9.11**	-8.88	-3.54	**32.74**
Mimo VV:	20.83	**39.90**	20.47	30.27	92.48	**116.38**	-15.64	-8.65	**-2.82**	-12.74	42.23	**70.88**
Set VV:	-46.32	135.12	**157.59**	-51.45	14.76	**36.36**	-31.53	17.23	**17.96**	-38.36	-15.54	**8.90**
HOG & LSTAT												
Mimo HH:	-18.82	**40.00**	6.73	19.35	**57.64**	48.46	-29.58	**-4.50**	-10.17	-37.57	-9.14	**11.45**
Mimo VV:	31.12	48.79	**99.41**	-3.98	78.43	**117.15**	-45.90	2.55	**3.86**	-7.44	13.80	**23.05**
Set VV:	-15.59	**57.65**	26.34	-40.45	**31.88**	31.34	-31.00	-13.42	**10.31**	-45.96	-19.03	**-15.30**
LBP & LSTAT												
Mimo HH:	11.10	**22.22**	-25.66	**15.69**	1.18	-9.71	**-4.90**	-28.38	-43.19	**13.92**	-10.66	-35.34
Mimo VV:	37.52	**89.33**	48.51	**35.12**	-21.65	19.53	-38.03	**-27.97**	-32.08	**12.11**	-36.96	-47.46
Set VV:	-71.34	**27.13**	-56.38	8.66	**58.55**	14.46	-42.66	**-22.89**	-38.77	-50.23	-39.21	**-37.80**
HOG, LBP & LSTAT												
Mimo HH:	-26.43	**39.56**	6.46	33.33	**55.81**	48.36	-30.17	**-4.57**	-10.55	-29.55	-9.38	**11.40**
Mimo VV:	42.87	49.09	**99.41**	-4.43	80.70	**117.28**	-40.52	2.37	**3.80**	-2.42	13.90	**23.07**
Set VV:	4.66	**56.59**	29.50	-36.76	**36.65**	31.64	-22.22	-11.79	**10.24**	-50.39	-19.01	**-14.47**

If one compares the results seen in Tables 4.43 and 4.45 to the deep learning results in this section, one would see that the performance margin is much smaller than anticipated. While the deep learners tend to perform better using combinations of image features, the SKSVM with the linear kernel does better than most deep learners with single image features. We see that the SKSVM with linear kernel even performs rather well with the LSTAT feature, which has easily been the weakest image feature used. The SKSVM even outperforms some of the deep learners using combinations of image features that include the LSTAT feature. This is not good news for the deep learners.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.22: Best improvements of linear kernel SKSVM using single image features



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.23: Best improvements of linear kernel SKSVM using combinations of image features

Table 4.47

Percent AUR Improvemets using Single Image Features with SKSVM RBF Kernel and Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	240.38	208.14	**240.94**	71.26	**63.77**	47.00	**-2.40**	-8.79	-17.43	42.49	31.83	**18.16**
Mimo VV:	158.69	**319.69**	197.05	82.13	149.72	**173.04**	-22.66	-12.47	**-10.64**	4.11	35.84	**32.06**
Set VV:	428.48	**638.99**	591.08	-39.82	-13.38	**23.70**	**8.52**	0.31	7.87	-50.23	-22.94	**-30.05**
LBP												
Mimo HH:	**86.64**	-63.77	-74.79	**-8.04**	-78.53	-61.27	**-12.16**	-53.61	-49.22	**0.49**	-14.78	-42.31
Mimo VV:	-68.38	-93.69	**233.64**	-35.75	**26.31**	-100.00	**-31.94**	-90.26	-57.44	**-24.33**	-95.06	-84.29
Set VV:	**177.81**	85.18	-100.00	-57.14	**-38.36**	-45.46	**-18.56**	-29.44	-84.84	**-42.43**	-45.61	-90.48
LSTAT												
Mimo HH:	181.73	**248.90**	227.38	69.88	**92.64**	44.72	**-14.27**	-18.35	-40.08	-46.33	**3.97**	-24.95
Mimo VV:	125.21	304.73	**310.26**	27.73	**85.16**	66.80	-39.93	**-31.05**	-37.54	**-16.73**	-51.02	-29.55
Set VV:	176.99	**252.96**	238.93	-57.62	-14.12	**-10.42**	-87.53	-37.31	**-25.39**	-66.81	-57.86	**-36.15**
FFST												
Mimo HH:	-100.00	**48.77**	-100.00	**20.37**	-55.05	-18.52	**-42.03**	-100.00	-100.00	-100.00	**-66.67**	-100.00
Mimo VV:	-100.00	**119.27**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**-93.81**	-100.00	-100.00	**-100.00**	-100.00
Set VV:	-100.00	**45.47**	-100.00	-100.00	**-47.66**	-59.26	-100.00	**-88.31**	-100.00	-100.00	**-55.46**	-100.00

Table 4.48

Percent AUR Improvemets using Single Image Features with SKSVM RBF Kernel and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	-21.89	1.91	**40.93**	7.48	64.74	**90.11**	-10.63	-13.19	**-5.36**	7.30	11.92	**38.73**
Mimo VV:	13.13	**39.63**	38.04	0.32	106.79	**146.21**	-14.54	**5.12**	-5.56	-26.02	**49.32**	45.11
Set VV:	74.29	**143.71**	127.91	-39.82	-13.38	**23.70**	**8.52**	0.31	7.87	-50.23	**-22.94**	-30.05
LBP												
Mimo HH:	**-28.67**	-85.13	-69.13	-71.21	-62.76	**-43.20**	**-5.20**	-85.44	-92.20	**-16.99**	-84.79	-76.75
Mimo VV:	**-20.38**	-93.55	-100.00	**0.65**	-53.85	-98.44	**-81.12**	-97.88	-96.96	-82.57	-99.60	**-53.04**
Set VV:	**-8.38**	-38.93	-100.00	-57.14	**-38.36**	-45.46	**-18.56**	-29.44	-84.84	**-42.43**	-45.61	-90.48
LSTAT												
Mimo HH:	-2.46	**35.90**	-24.17	6.77	**30.24**	-17.50	-28.99	**-24.08**	-42.66	-6.79	**-3.08**	-33.49
Mimo VV:	10.36	**64.83**	34.27	-6.29	3.21	**42.57**	-62.14	**-38.73**	-39.14	-30.15	-36.26	**-18.78**
Set VV:	-8.65	**16.40**	11.78	-57.62	-14.12	**-10.42**	-87.53	-37.31	**-25.39**	-66.81	-57.86	**-36.15**
FFST												
Mimo HH:	-100.00	**-80.35**	-100.00	-100.00	**-100.00**	-100.00	-100.00	**-62.48**	-100.00	**-54.53**	-62.16	-100.00
Mimo VV:	-100.00	**-49.37**	-100.00	-100.00	**-34.47**	-100.00	-100.00	**-65.08**	-100.00	**-59.39**	-100.00	-100.00
Set VV:	-100.00	**-52.03**	-100.00	-100.00	**-47.66**	-59.26	-100.00	**-88.31**	-100.00	-100.00	**-57.80**	-100.00

Table 4.49
Percent AUR Improvemets using Combinations of Image Features with SKSVM RBF Kernel and Phase Adjustment

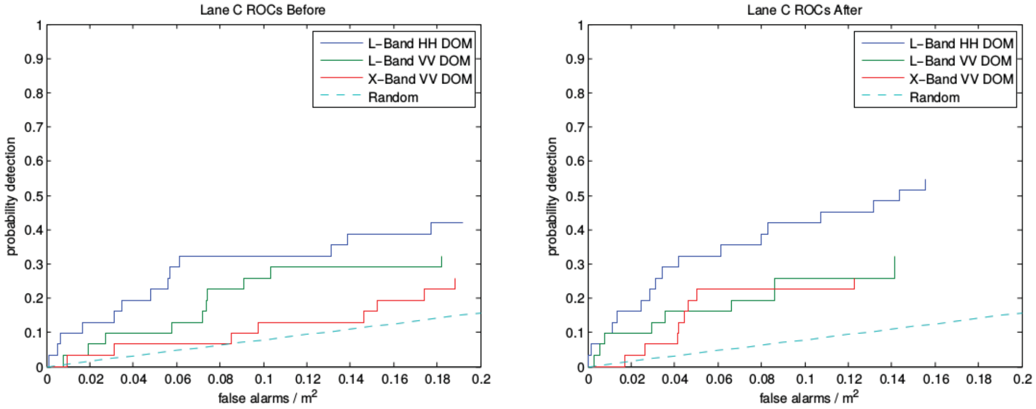
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	273.14	196.44	**213.91**	34.39	**42.79**	38.60	**-20.58**	-29.51	-27.24	**52.70**	22.50	16.69
Mimo VV:	193.47	**356.46**	144.30	112.68	**198.25**	86.52	**-12.85**	-22.43	-19.97	0.58	-0.15	**32.41**
Set VV:	267.71	71.33	**648.46**	-67.08	-16.78	**7.69**	**-4.58**	-12.16	-20.62	-52.70	**-17.06**	-24.17
HOG & LSTAT												
Mimo HH:	264.04	**308.11**	247.29	62.82	99.04	**114.33**	-3.00	**5.74**	-4.68	14.79	30.15	**52.87**
Mimo VV:	326.51	510.54	**607.13**	65.97	222.70	**297.37**	0.62	-2.52	**10.16**	0.15	28.78	**51.34**
Set VV:	334.12	341.04	**341.13**	-32.22	**22.92**	16.06	-31.57	-15.64	**21.77**	-51.63	-23.10	**-14.85**
LBP & LSTAT												
Mimo HH:	215.81	269.43	**271.29**	21.95	**73.98**	28.72	**-15.35**	-23.23	-36.61	**-4.24**	-12.51	-18.55
Mimo VV:	364.31	336.92	**341.10**	3.92	**102.65**	59.59	**-36.06**	-37.80	-37.53	**-22.34**	-43.99	-25.36
Set VV:	-11.83	99.93	**311.62**	-32.76	-7.16	**-4.82**	-51.46	**-26.24**	-29.09	-51.50	-50.08	**-31.65**
HOG, LBP & LSTAT												
Mimo HH:	260.72	**311.56**	245.42	79.81	95.70	**111.11**	-13.85	**2.77**	-4.34	22.11	27.88	**53.46**
Mimo VV:	340.86	526.14	**608.43**	72.35	219.51	**294.85**	-2.00	-4.14	**8.56**	-2.79	28.58	**49.90**
Set VV:	222.35	283.57	**315.54**	-58.91	**19.90**	15.24	-35.20	-21.25	**17.95**	-54.83	-21.68	**-14.50**

Table 4.50

Percent AUR Improvemets using Combinations of Image Features with SKSVM RBF Kernel and No Phase Adjustment

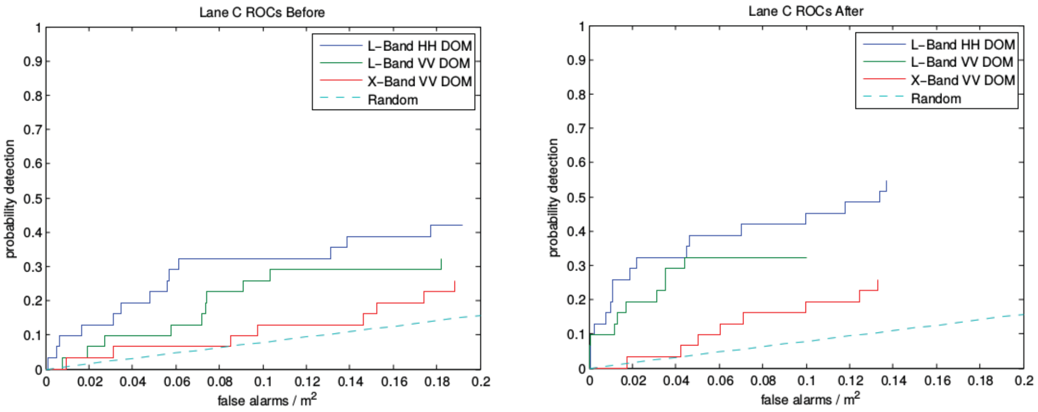
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	-22.49	-8.42	**33.93**	-39.84	49.03	**60.76**	-16.96	-24.31	**-11.61**	-1.13	2.60	**33.35**
Mimo VV:	-53.39	34.00	**37.23**	-7.09	102.78	**111.97**	-23.09	**-8.62**	-28.75	-19.50	30.12	**32.28**
Set VV:	21.27	-43.50	**146.83**	-67.08	-16.78	**7.69**	**-4.58**	-12.16	-20.62	-52.70	**-17.06**	-24.17
HOG & LSTAT												
Mimo HH:	9.24	60.72	**36.55**	56.86	**84.53**	31.69	-14.34	**4.08**	0.99	-1.15	**26.63**	20.34
Mimo VV:	42.57	107.72	**101.88**	-11.23	**110.94**	93.65	-29.09	3.23	**13.73**	14.94	8.15	**49.44**
Set VV:	43.17	45.45	**45.48**	-32.22	**22.92**	16.06	-31.57	-15.64	**21.77**	-51.63	-23.10	**-14.85**
LBP & LSTAT												
Mimo HH:	9.74	**28.13**	-14.82	-15.54	**14.58**	-17.55	**-13.21**	-23.69	-38.15	-8.36	**-2.02**	-32.19
Mimo VV:	27.85	**75.99**	29.03	-25.21	1.72	**39.46**	-58.72	-44.26	**-31.84**	-21.92	-27.80	**-14.63**
Set VV:	-70.92	-34.07	**35.75**	-32.76	-7.16	**-4.82**	-51.46	**-26.24**	-29.09	-51.50	-50.08	**-31.65**
HOG, LBP & LSTAT												
Mimo HH:	4.11	**59.68**	35.40	46.92	**85.25**	30.15	-12.77	**3.36**	0.75	-7.07	**26.13**	20.19
Mimo VV:	18.84	**106.04**	101.39	9.16	**106.86**	94.88	-30.66	2.92	**13.02**	-6.20	7.68	**48.27**
Set VV:	6.31	26.50	**37.04**	-58.91	**19.90**	15.24	-35.20	-21.25	**17.95**	-54.83	-21.68	**-14.50**

Continuing our investigation of the SKSVM, we see from Tables 4.47-4.50 that the RBF kernel performs rather well also. In fact, it out performs the linear kernel for the single LSTAT, as well as the combinations of {HOG & LSTAT} and {HOG, LBP, & LSTAT}. For the rest of the features, the differences vary from marginal to large. Like the linear kernel, the non phase adjusted data produces much worse results than the phase adjusted data.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.24: Best improvements of RBF kernel SKSVM using single image features



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.25: Best improvements of RBF kernel SKSVM using combinations of image features

4.3.6 MKLGL

Table 4.51

Percent AUR Improvemets using Single Image Features with MKLGL and Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	**294.99**	210.30	203.06	105.53	**109.19**	36.92	-26.42	-21.86	**-14.35**	**67.64**	38.11	56.63
Mimo VV:	194.17	463.08	**479.66**	98.32	257.71	**296.70**	-26.08	-12.42	**5.41**	-6.70	45.00	**56.58**
Set VV:	392.78	**674.78**	533.60	-29.01	24.25	**38.70**	-21.75	**8.55**	7.31	-40.01	-17.03	**-0.61**
LBP												
Mimo HH:	**288.33**	258.68	235.02	25.84	59.86	**173.11**	-1.35	**8.33**	5.43	76.01	**83.91**	75.24
Mimo VV:	533.62	481.33	**552.52**	98.09	227.71	**325.02**	-12.81	-19.23	**16.43**	30.51	**86.40**	82.98
Set VV:	491.70	634.52	**776.80**	23.95	**58.66**	57.09	20.84	**48.22**	15.32	1.91	**20.14**	-4.81
LSTAT												
Mimo HH:	147.19	**305.66**	222.76	63.81	**102.69**	70.71	**-14.10**	-17.17	-38.76	-73.26	**4.19**	-16.37
Mimo VV:	51.66	126.00	**325.86**	56.28	85.01	**92.31**	-44.84	-38.33	**-36.40**	**-14.22**	-43.18	-27.70
Set VV:	83.99	170.71	**432.31**	-37.57	-20.29	**-3.30**	-74.45	**-25.59**	-36.30	-81.24	-51.50	**-34.98**
FFST												
Mimo HH:	**61.33**	53.63	6.12	-17.69	-47.94	**-7.71**	-56.16	-67.40	**-52.09**	-39.40	-53.72	**-35.31**
Mimo VV:	-58.26	**45.48**	-64.52	62.40	**87.02**	-61.64	-65.41	**-57.47**	-78.87	-71.83	-56.74	**-39.48**
Set VV:	**363.45**	161.78	-64.66	-15.48	**-4.67**	-62.01	-20.16	-58.45	**-5.40**	-72.30	**-41.85**	-48.43

Table 4.52

Percent AUR Improvemets using Single Image Features with MKLGL and No Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG												
Mimo HH:	-13.73	9.95	**30.98**	29.24	51.53	**97.50**	-22.69	-15.70	**-1.65**	-16.14	19.71	**41.68**
Mimo VV:	3.74	46.52	**48.02**	-20.04	90.61	**145.32**	-10.60	5.53	**8.79**	-8.66	50.34	**67.03**
Set VV:	62.51	**155.51**	108.95	-29.01	24.25	**38.70**	-21.75	**8.55**	7.31	-40.01	-17.03	**-0.61**
LBP												
Mimo HH:	15.74	27.09	**32.84**	38.81	38.28	**81.98**	10.31	**18.85**	8.04	46.82	49.08	**58.03**
Mimo VV:	33.78	**73.72**	61.77	96.25	113.34	**121.75**	-2.36	6.51	**16.53**	20.12	**72.80**	72.15
Set VV:	95.13	142.24	**189.16**	23.95	**58.66**	57.09	20.84	**48.22**	15.32	1.91	**20.14**	-4.81
LSTAT												
Mimo HH:	-22.44	**-2.73**	-28.62	16.09	**25.54**	-14.52	**-34.01**	-31.71	-36.53	-21.83	**-19.00**	-34.79
Mimo VV:	9.37	24.52	**69.08**	-7.78	-25.91	**68.07**	-62.25	**-35.50**	-36.74	-49.20	-33.88	**-21.45**
Set VV:	-39.32	-10.72	**75.55**	-37.57	-20.29	**-3.30**	-74.45	**-25.59**	-36.30	-81.24	-51.50	**-34.98**
FFST												
Mimo HH:	-82.69	**-55.43**	-75.13	**-40.46**	-62.85	-87.34	-68.01	**-56.82**	-60.51	**-34.31**	-65.13	-73.22
Mimo VV:	-63.88	**-49.54**	-51.24	-48.58	-4.35	**35.06**	-49.38	**-33.21**	-63.54	-56.61	**-31.49**	-66.32
Set VV:	**52.84**	-13.67	-88.35	-15.48	**-4.67**	-62.01	-20.16	-58.45	**-5.40**	-73.76	**-44.91**	-51.15

Table 4.53

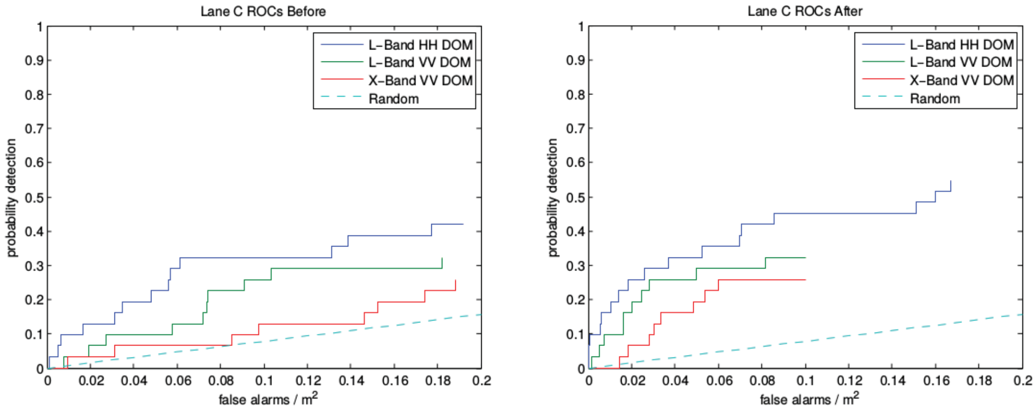
Percent AUR Improvemets using Combinations of Image Features with MKLGL and Phase Adjustment

	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	**293.11**	207.83	204.36	99.63	**109.29**	34.51	-28.84	-24.99	**-15.59**	**75.31**	39.38	56.68
Mimo VV:	251.38	467.63	**470.88**	105.80	258.16	**296.48**	-22.15	-11.09	**6.30**	-2.05	45.48	**55.08**
Set VV:	433.50	**668.41**	556.56	-33.97	38.36	**43.56**	-23.88	**15.26**	7.87	-39.14	-13.32	**-3.50**
HOG & LSTAT												
Mimo HH:	277.02	**310.42**	211.87	78.95	**162.14**	119.68	-4.60	**10.03**	-7.43	8.05	33.33	**46.10**
Mimo VV:	298.51	412.70	**515.09**	106.64	178.10	**317.76**	-8.21	0.54	**22.36**	-15.66	37.99	**55.49**
Set VV:	319.36	**431.58**	415.73	-15.86	**42.88**	26.33	-34.79	-10.69	**11.15**	-55.35	-20.70	**-11.39**
LBP & LSTAT												
Mimo HH:	176.22	**325.73**	250.51	33.51	**93.08**	63.01	**-12.93**	-16.68	-37.75	-68.09	**4.64**	-15.33
Mimo VV:	101.44	173.46	**349.92**	46.13	71.89	**79.41**	-41.78	-37.09	**-35.63**	**-12.11**	-37.49	-27.67
Set VV:	47.01	143.20	**437.41**	-39.50	-20.00	**-1.79**	-65.61	**-22.83**	-36.62	-79.35	-47.73	**-34.49**
HOG, LBP & LSTAT												
Mimo HH:	280.49	**310.57**	212.16	72.23	**161.16**	118.80	-5.95	**9.85**	-7.52	9.16	33.44	**46.19**
Mimo VV:	298.84	415.30	**516.07**	103.08	176.91	**317.91**	-7.94	0.56	**22.36**	-16.40	38.05	**55.43**
Set VV:	328.93	**419.47**	412.54	-11.29	**43.90**	26.38	-35.42	-9.22	**10.96**	-55.23	-20.58	**-11.92**

Table 4.54
Percent AUR Improvemets using Combinations of Image Features with MKLGL and No Phase Adjustment

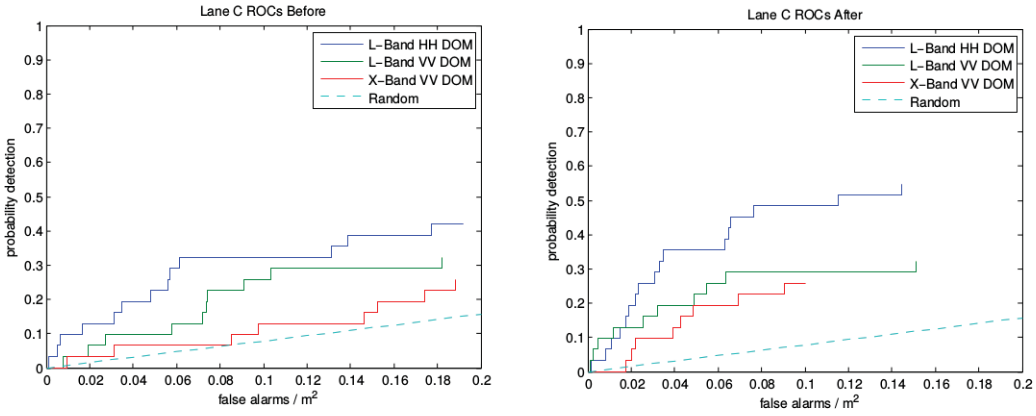
	Train B, D, A, Test C			**Train C, B, A, Test D**			**Train C, B, D, Test A**			**Train A,C,D, Test B**		
Cellsize:	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5	3x3	4x4	5x5
HOG & LBP												
Mimo HH:	-17.02	**10.83**	29.78	36.25	**43.32**	95.43	-19.79	-15.95	**-2.34**	-25.51	20.51	**41.96**
Mimo VV:	15.99	51.97	**46.24**	-3.39	92.42	**143.24**	-7.10	6.84	**7.74**	-3.36	50.09	**66.93**
Set VV:	75.94	**153.41**	116.52	-33.97	**38.36**	43.56	-23.88	**15.26**	7.87	-39.14	-13.32	**-3.50**
HOG & LSTAT												
Mimo HH:	6.51	**43.39**	34.48	64.88	**78.19**	59.98	-13.37	**5.90**	-0.03	-3.49	-0.03	**19.45**
Mimo VV:	49.80	85.97	**107.71**	29.12	62.40	**93.52**	-31.09	9.45	**12.72**	19.32	19.86	**43.37**
Set VV:	38.30	**75.31**	70.08	-15.86	**42.88**	26.33	-34.79	-10.69	**11.15**	-55.35	-20.70	**-11.39**
LBP & LSTAT												
Mimo HH:	-10.83	**0.82**	-23.63	8.93	**24.72**	-15.05	**-31.87**	-31.91	-36.06	-20.09	**-19.42**	-33.10
Mimo VV:	24.09	29.95	**79.45**	-24.10	-30.97	**73.83**	-56.44	-36.27	**-34.01**	-40.22	-30.89	**-12.39**
Set VV:	-51.52	-19.80	**77.23**	-39.50	-20.00	**-1.79**	-65.61	**-22.83**	-36.62	-79.35	-47.73	**-34.49**
HOG, LBP & LSTAT												
Mimo HH:	4.81	**43.50**	34.53	64.31	**78.04**	59.83	-13.57	**5.75**	-0.12	-4.45	-0.68	**19.45**
Mimo VV:	47.62	85.58	**108.10**	27.95	62.59	**93.52**	-31.77	9.56	**12.61**	18.11	20.04	**43.47**
Set VV:	41.45	**71.31**	69.03	-11.29	**43.90**	26.38	-35.42	-9.22	**10.96**	-55.23	-20.58	**-11.92**

The results of the multi-lane MKLGL are outstanding to say the least. Again, the non phase adgusted results are many times worse than the phase adjusted results, so we will focus on those. If we compare the MKLGL to the SKSVMs, we see that the MKLGL performs much better than either kernel in almost every case. It even does well on the Test A and Test B runs, which are notoriously the weakest. Looking back even further, we see that the MKLGL even outperforms the deep learning networks in the Test B. However, the DBN and SDAE-NN both come very close on certain feature sets, particularly the HOG and the HOG/LBP combination. Looking into the other runs, however, we see that for the HOG/LBP combination, the MKLGL is outperformed by the DBN and the SDAE-NN almost uniformly.



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.26: Best improvements of MKLGL using single image features



(a) Lane C ROCs Before **(b)** Lane C ROCs After

Figure 4.27: Best improvements of MKLGL using combinations of features

4.4 Conclusions

Reviewing our results, some interesting conclusions can be drawn. At first glance it appears that the deep learning algorithms perform very well in the single-lane training experiments and, in contrast, the shallow learning algorithms seemed to perform better when multiple lanes were used for training. Once multiple lanes were used for training, the shallow learning algorithms seemed to perform better. In the all around sense, this is more or less true. The shallow learners using certain features, such as the LBP or the LSTAT, do outperform the deep learners when multiple lanes are used for training. However, as Tables 4.55 and 4.56 show, if we closely examine the HOG and HOG/LBP feature combinations, we see a different story unfold.

Table 4.55
Best Performing Algorithm for Select Features using Single-Lane Training

HOG	**Test B**	**Test A**
Mimo HH:	SKSVM-RBF	SDAE-NN
Mimo VV:	SDAE-NN	SDAE-NN
Set VV:	SDAE-SVM	SKSVM-Lin
HOG & LBP		
Mimo HH:	SKSVM-RBF	SDAE-NN
Mimo VV:	DBN	DBN
Set VV:	SDAE-NN	DBN

Table 4.56
Best Performing Algorithm for Select Features using Multi-Lane Training

HOG	**Test C**	**Test D**	**Test A**	**Test B**
Mimo HH:	MKLGL	SDAE-NN	DBN	DBN
Mimo VV:	DBN	SDAE-NN	SDAE-NN	SDAE-NN
Set VV:	MKLGL	SDAE-NN	SDAE-NN	DBN
HOG & LBP				
Mimo HH:	MKLGL	SDAE-NN	SDAE-NN	MKLGL
Mimo VV:	DBN	SDAE-NN	SDAE-NN	SDAE-NN
Set VV:	DBN	DBN	DBN	SDAE-NN

In this examination, we see that when restricted to these two features the deep learners really shine. The SDAE-NN does particularly well as an all around FA rejector when trained and tested on the HOG and HOG/LBP features. Likewise, the DBN also does rather well, getting the top slot on some and falling not far behind on most others. This speaks to the robustness of these algorithms. The idea behind the DBN is to learn a representation of the input data and then use this representation to attempt to reconstruct new data in the same way. The SDAE attempts to learn a mapping between the corrupted input to a lower dimensional output. It then uses the NN to finetune the layer connections and then uses the FFNN to attempt to classify new data. Conversly, the MKLGL seeks to take multiple kernel matricies and find an optimal hyperplane between the feature vectors of the two output classes. Because of this, the MKLGL requires more memory to store its relevant variables.

As Table 4.57 shows, for the multi-lane training the MKLGL runs in about half the time of the DBN and both SDAEs. This is likely due to the speed of LIBSVM, MKLGL's primary function call. However, the MKLGL requires about 1.4 GB more memory to store the variables generated than the DBN and SDAEs. This is because the MKLGL needs, in our case, 10 $N \times N$ kernel matricies in its model to perform its predictions; conversely the DBN only needs two weights matrices and two bias vectors.

Table 4.57
Run Time and Memory Requirement Comparison of Different False Alarm Rejection Classifiers

Network Type	Lanes Trained	Run Time/Cellsize(sec)	Memory Used(MB)
DBN	1	54	438
DBN	3	337	266
SDAE-NN	1	284	439
SDAE-NN	3	319	264
SDAE-SVM	1	343	439
SDAE-SVM	3	500	264
CNN	1	2592	502
CNN	3	4198	305
SKSVM Linear	1	49	439
SKSVM Linear	3	40	271
SKSVM RBF	1	49	439
SKSVM RBF	3	40	271
MKLGL	1	56	597
MKLGL	3	146	1846

Overall, the SDAE-NN seems to be the best overall choice out of the networks explored. When combined with the HOG feature or the combination of the HOG and LBP features, the SDAE-NN had the most improvements over all the results. Following that are the DBN and the MKLGL. The DBN is simple, yet rugged. Its performance could possibly be enhanced with a fine-tuning step, however the methods we attempted to achieve this, both were detremental to the results. This was attributed to potential over-training since the DBN's confidence relies on the RMSE of the reconstructed inputs. Thus a better reconstruction is actually less desirable for the true targets. The MKLGL, while requiring much more memory than both the SDAE-NN and the DBN combined, was still the quickest. When allowing all other considered image features, the MKLGL also assumes the top spot in AUR improvement. While there is no one-size-fits-all solution to this problem, it seems clear that deep learning algorithms are both a viable and perhaps even desirable approach moving forward.

4.5 Future Work

There are a few recommendations for further developement that could likely improve the performance of the deep learning algorithms as false alarm rejectors. If the training portions of the deep learning algorithms could be parallelized, it could significantly reduce runtimes. This gives two advantages. One, the results would be available sooner to the vehicle crew which could increase stand-off distance to the potential threat. Two, it would allow the researcher to perform more test runs to better fine-tune the parameters of the network. These algorithms have many different parameters to tune and at current run times it is very time consuming to change one at a time and re-run to check for changes. Along these lines, we believe the CNN has much more potential than was found in our experiments. Due to both the CNNs' long run time, the CNNs were not as heavily tested as they could have been. Additionally, the FFST should be investigated further to see if other coeffieicents or even the full transform would provide better results in not only the CNN but perhaps the other algorithms as well.

References

[1] MID, J. C., “Global IED monthly summary report,“ (August 2012).

[2] Cremer, F., Chavemaker, J. G., deJong, W., and Schutte, K., ”Comparison of vehicle-mounted forward-looking polarimetric infrared and downward-looking infrared sensors for landmine detection,“ in [*Proc. SPIE*], **5089**, 517526 (2003).

[3] Playle, N., Port, D. M., Rutherford, R., Burch, I. A., and Almond, R., ”Infrared polarization sensor for forward-looking mine detection,” in [*Proc. SPIE*], **4742**, 1118 (2002).

[4] Costley, R. D., Sabatier, J. M., and Xiang, N., “Forward-looking acoustic mine detection system,” in [*Proc. SPIE*], **4394**, 617626 (2001).

[5] Collins, L. M., Torrione, P. A., Throckmorton, C. S., Liao, X., Zhu, Q. E., Liu, Q., Carin, L., Clodfelter, F., and Frasier, S., “Algorithms for landmine discrimination using the NIITEK ground penetrating radar,” in [*Proc. SPIE*], **4742**, 709718 (2002).

[6] Gader, P. D., Grandhi, R., Lee, W. H., Wilson, J. N., and Ho, K. C., “Feature analysis

for the NIITEK ground penetrating radar using order weighted averaging operators for landmine detection," in [*Proc. SPIE*], **5415**, 953962 (2004).

[7] Bradley, M. R., Witten, T. R., Duncan, M., and McCummins, R., "Anti-tank and side-attach mine detection with forward-looking GPR," in [*Proc. SPIE*], **5415**, 421432 (2004).

[8] Cosgrove, R. B., Milanfar, P., and Kositsky, J., "Trained detection of buried mines in SAR images via the deflection-optimal criterion," *IEEE Trans. Geoscience and Remote Sensing* **42**(11), 25692575 (2004).

[9] Sun, Y. and Li, J., "Plastic landmine detection using time-frequency analysis for forward-looking ground-penetrating radar," in [*Proc. SPIE*], **5089**, 851862 (2003).

[10] Stone, K., Keller, J. M., Ho, K. C., and Gader, P. D., "On the registration of FLGPR and IR data for the forward-looking landmine detection system and its use in eliminating FLGPR false alarms," in [*Proc. SPIE*], **6953** (2008).

[11] Stone, K., Keller, J. M., Popescu, M., Havens, T. C., and Ho, K. C., "Forward-looking anomaly detection via fusion of infrared and color imagery," in [*Proc. SPIE*], **7664**, 766425 (2010).

[12] Havens, T. C., Stone, K., Keller, J. M., and Ho, K. C., "Sensor-fused detection of explosive hazards," in [*Proc. SPIE*], **7303**, 73032A (2009).

[13] Havens, T. C., Ho, K. C., Farrell, J., Keller, J. M., Popescu, M., Ton, T. T., Wong, D. C., and Soumekh, M., ”Locally adaptive detection algorithm for forward-looking ground-penetrating radar,“ in [*Proc. SPIE*], **7664**, 76442E (2010).

[14] Havens, T. C., Spain, C. J., Ho, K. C., Keller, J. M., Ton, T. T., Wong, D. C., and Soumekh, M., ”Improved detection and false alarm rejection using ground-penetrating radar and color imagery in a forward-looking system,“ in [*Proc. SPIE*], **7664**, 76441U (2010).

[15] Havens, T. C., Keller, J. M., Ho, K. C., Ton, T. T., Wong, D. C., and Soumekh, M., ”Narrow band processing and fusion approach for explosive hazard detection in FLGPR,” in [*Proc. SPIE*], **8017**, 80171F (2011).

[16] Farrell, J., Havens, T. C., Ho, K. C., Keller, J. M., Ton, T. T., Wong, D. C., and Soumekh, M., “Detection of explosive hazards using spectrum features from forward-looking ground penetrating radar imagery,” in [*Proc. SPIE*], **8017**, 80171F (2011).

[17] Farrell, J., Havens, T. C., Ho, K. C., Keller, J. M., Ton, T. T., Wong, D. C., and Soumekh, M., “Evaluation and improvement of spectral features for the detection of buried explosive hazards using forward-looking ground-penetrating radar,” in [*Proc. SPIE*], **8357**, 8357C (2012).

[18] Hinton, G. E., and Salakhutdinov, R. R. “Reducing the dimensionality of data with neural networks.” *Science* 313.5786 (2006): 504-507.

[19] Hinton, G., Osindero, S., and Teh, Y. W. "A fast learning algorithm for deep belief nets." *Neural computation* 18.7 (2006): 1527-1554.

[20] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. A. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *The Journal of Machine Learning Research* 11 (2010): 3371-3408.

[21] Palm, Rasmus Berg. "Prediction as a candidate for learning deep hierarchical models of data." *Technical University of Denmark, Palm* (2012).

[22] Besaw, L. E., and Stimac, P. J. "Deep learning algorithms for detecting explosive hazards in ground penetrating radar data." *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014.

[23] LeCun, Y., Cortes, C., and Burges, C. "THE MNIST DATABASE." *MNIST Handwritten Digit Database, Yann LeCun, Corinna Cortes and Chris Burges.* N.p., n.d. Web. 13 Aug. 2014.

[24] Havens, T. C., Becker, J., Pinar, A., and Schulz, T. J. "Multi-band sensor-fused explosive hazards detection in forward-looking ground-penetrating radar." *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014.

[25] Shin, H. C., Orton, M. R., Collins, D. J., Doran, S. J., and Leach, M. O. "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013): 1930-1943.

[26] Chang, C. C., and Lin, C. J "LIBSVM: a library for support vector machines." *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011): 27.

[27] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[28] Bengio, Y. "Learning deep architectures for AI." *Foundations and trends in Machine Learning* 2.1 (2009): 1-127.

[29] Ojala, Timo, Matti Pietikinen, and Topi Menp. "A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification." *Advances in Pattern RecognitionICAPR 2001.* Springer Berlin Heidelberg, 2001. 399-408.

[30] Huser, S., and Steidl, G. (2013). "Convex Multiclass Segmentation with Shearlet Regularization." *International Journal of Computer Mathematics.* 90, (1), 62-81.

[31] Huser, S., and Steidl, G. (2014). "Fast Finite Shearlet Transform: a tutorial." *ArXiv.* (1202.1773)

[32] Stone, K., and Keller, J. M. "Convolutional neural network approach for buried target recognition in FL-LWIR imagery." *SPIE Defense+ Security.* International Society for Optics and Photonics, 2014.

[33] Nordin, Rosdiadee, and Mahamod Ismail. "An investigation of self-interference reduction strategy in correlated SM-OFDMA systems." *Communications (APCC), 2011 17th Asia-Pacific Conference on.* IEEE, 2011.

[34] Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. ”Learning the kernel matrix with semidefinite programming.” *The Journal of Machine Learning Research* **5** (2004): 27-72.

[35] Mercer, J. ”Functions of positive and negative type, and their connection with the theory of integral equations.” *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character* (1909): 415-446.

[36] Kloft, M., Brefeld, U., Laskov, P., and Sonnenburg, S. ”Non-sparse multiple kernel learning.” (2008).

[37] Cortes, C., Mehryar Mohri, and Afshin Rostamizadeh. ”L 2 regularization for learning kernels.” *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence.* AUAI Press, 2009.

[38] Farrell, J., Havens, T. C., Ho, K. C., Keller, J. M., Ton, T. T., Wong, D. C., and Soumekh, M. ”Evaluation and improvement of spectral features for the detection of buried explosive hazards using forward-looking ground-penetrating radar.” *SPIE Defense, Security, and Sensing. International Society for Optics and Photonics,* 2012.

[39] Xu, Z., Jin, R., Yang, H., King, I., and Lyu, M. R. ”Simple and efficient multiple kernel learning by group lasso.” *Proceedings of the 27th International Conference on Machine Learning* (ICML-10). 2010.

[40] Boser, B. E., Guyon, I. M., and Vapnik, V. N. ”A training algorithm for optimal margin

classifiers.” *Proceedings of the fifth annual workshop on Computational learning theory.* ACM, 1992.

[41] Hsu, C. W., Chang, C. C., and Lin, C. J. ”A practical guide to support vector classification.” (2003).

[42] Cortes, C., and Vapnik, V. ”Support-vector networks.” *Machine learning* 20.3 (1995): 273-297.

[43] Sarikaya, R., Hinton, G. E., and Ramabhadran, B. ”Deep belief nets for natural language call-routing.” *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on.* IEEE, 2011.

[44] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. ”Face recognition: A convolutional neural-network approach.” *Neural Networks, IEEE Transactions on* 8.1 (1997): 98-113.

[45] De Villiers, J., and Barnard, E. ”Backpropagation neural nets with one and two hidden layers.” *Neural Networks, IEEE Transactions on* 4.1 (1993): 136-141.

[46] Kandel, E. R., Schwartz, J. H., and Jessell, T. M. *Principles of neural science.* Vol. 4. New York: McGraw-Hill, 2000.

[47] Dalal, N., and Triggs, B. ”Histograms of oriented gradients for human detection.” *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE, 2005.

[48] LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., M´uller, U., S´ackinger, E., Simard, P., and Vapnik, V. ”Comparison of learning algorithms for handwritten digit recognition.” *International conference on artificial neural networks.* Vol. 60. 1995.

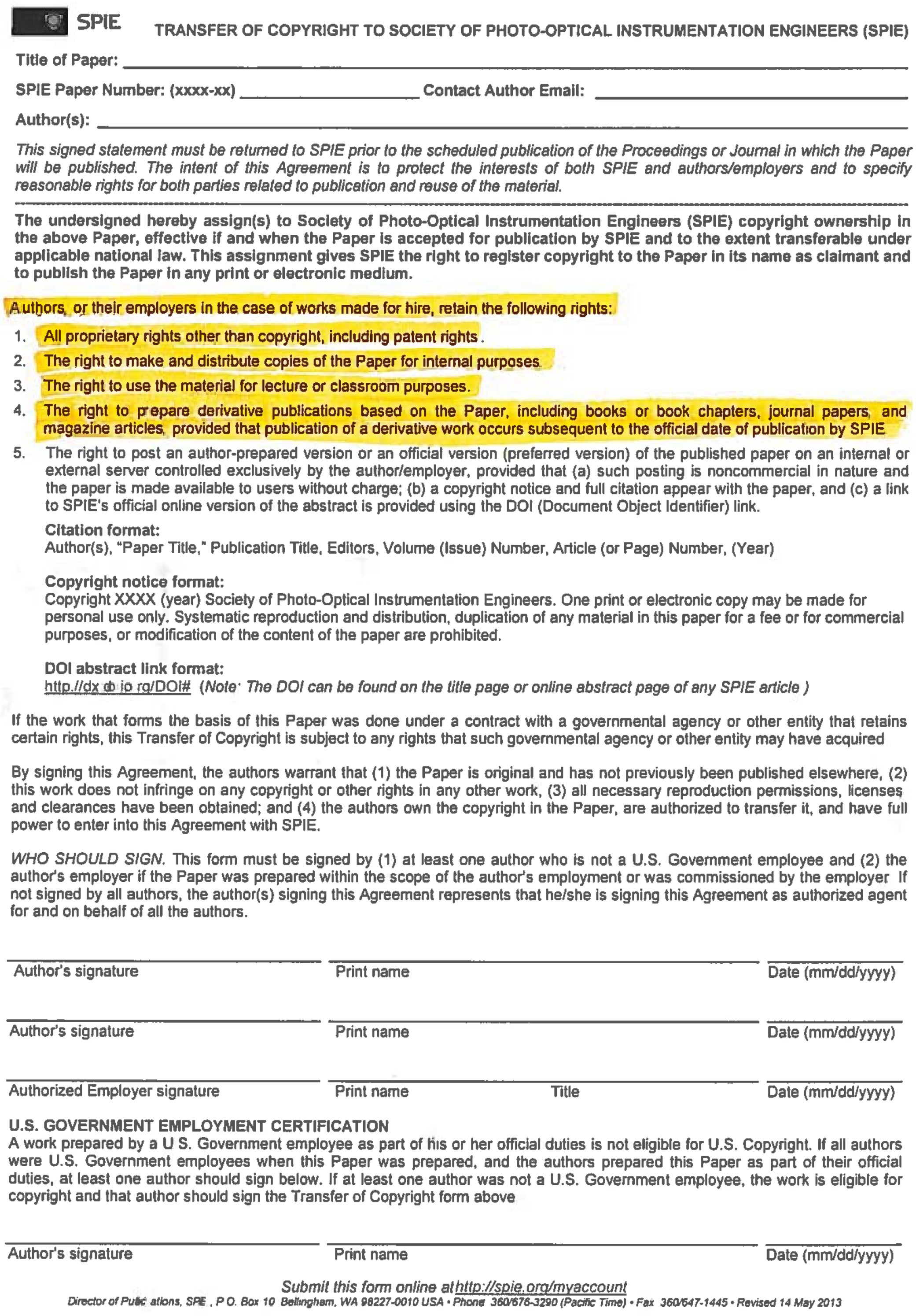
[49] http://www.mathematik.uni-kl.de/imagepro/members/haeuser/ffst/

[50] Becker, J., Havens, T. C., Pinar, A., Schulz, T. J. “Stacked Denoising Autoencoders for Explosive Hazard Detection in Multiband Forward-Looking Ground-Penetrating Radar.” Under Review for *Geoscience and Remote Sensing, IEEE Transactions on*

[51] Becker, J., Havens, T. C., Pinar, A., Schulz, T. J. “Deep Belief Networks for False Alarm Rejection in Forward-Looking Ground-Penetrating Radar” Submitted to *SPIE Defense+ Security.* International Society for Optics and Photonics, 2015.

Appendix A

Letters of Permission



SPIE TRANSFER OF COPYRIGHT TO SOCIETY OF PHOTO-OPTICAL INSTRUMENTATION ENGINEERS (SPIE)

Title of Paper: ______________________________

SPIE Paper Number: (xxxx-xx) ______________ **Contact Author Email:** ______________

Author(s): ______________________________

This signed statement must be returned to SPIE prior to the scheduled publication of the Proceedings or Journal in which the Paper will be published. The intent of this Agreement is to protect the interests of both SPIE and authors/employers and to specify reasonable rights for both parties related to publication and reuse of the material.

The undersigned hereby assign(s) to Society of Photo-Optical Instrumentation Engineers (SPIE) copyright ownership in the above Paper, effective if and when the Paper is accepted for publication by SPIE and to the extent transferable under applicable national law. This assignment gives SPIE the right to register copyright to the Paper in its name as claimant and to publish the Paper in any print or electronic medium.

Authors, or their employers in the case of works made for hire, retain the following rights:

1. All proprietary rights other than copyright, including patent rights.
2. The right to make and distribute copies of the Paper for internal purposes.
3. The right to use the material for lecture or classroom purposes.
4. The right to prepare derivative publications based on the Paper, including books or book chapters, journal papers, and magazine articles, provided that publication of a derivative work occurs subsequent to the official date of publication by SPIE
5. The right to post an author-prepared version or an official version (preferred version) of the published paper on an internal or external server controlled exclusively by the author/employer, provided that (a) such posting is noncommercial in nature and the paper is made available to users without charge; (b) a copyright notice and full citation appear with the paper, and (c) a link to SPIE's official online version of the abstract is provided using the DOI (Document Object Identifier) link.

   Citation format:
   Author(s), "Paper Title," Publication Title, Editors, Volume (Issue) Number, Article (or Page) Number, (Year)

   Copyright notice format:
   Copyright XXXX (year) Society of Photo-Optical Instrumentation Engineers. One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

   DOI abstract link format:
   http://dx.doi.org/DOI# (*Note: The DOI can be found on the title page or online abstract page of any SPIE article.*)

If the work that forms the basis of this Paper was done under a contract with a governmental agency or other entity that retains certain rights, this Transfer of Copyright is subject to any rights that such governmental agency or other entity may have acquired

By signing this Agreement, the authors warrant that (1) the Paper is original and has not previously been published elsewhere, (2) this work does not infringe on any copyright or other rights in any other work, (3) all necessary reproduction permissions, licenses and clearances have been obtained; and (4) the authors own the copyright in the Paper, are authorized to transfer it, and have full power to enter into this Agreement with SPIE.

WHO SHOULD SIGN. This form must be signed by (1) at least one author who is not a U.S. Government employee and (2) the author's employer if the Paper was prepared within the scope of the author's employment or was commissioned by the employer. If not signed by all authors, the author(s) signing this Agreement represents that he/she is signing this Agreement as authorized agent for and on behalf of all the authors.

Author's signature ______ Print name ______ Date (mm/dd/yyyy) ______

Author's signature ______ Print name ______ Date (mm/dd/yyyy) ______

Authorized Employer signature ______ Print name ______ Title ______ Date (mm/dd/yyyy) ______

U.S. GOVERNMENT EMPLOYMENT CERTIFICATION

A work prepared by a U.S. Government employee as part of his or her official duties is not eligible for U.S. Copyright. If all authors were U.S. Government employees when this Paper was prepared, and the authors prepared this Paper as part of their official duties, at least one author should sign below. If at least one author was not a U.S. Government employee, the work is eligible for copyright and that author should sign the Transfer of Copyright form above

Author's signature ______ Print name ______ Date (mm/dd/yyyy) ______

Submit this form online at http://spie.org/myaccount

Director of Publications, SPIE, P.O. Box 10 Bellingham, WA 98227-0010 USA • Phone 360/676-3290 (Pacific Time) • Fax 360/647-1445 • Revised 14 May 2013