

Using Image Processing Methods to Improve the Detection of Buried Explosive Threats in GPR Data

by

Rayn Sakaguchi

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

Leslie M. Collins, Supervisor

Loren W. Nolte

Guillermo Sapiro

Carlo Tomasi

Stacy Tantum

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University

2016

ABSTRACT

Using Image Processing Methods to Improve the Detection of
Buried Explosive Threats in GPR Data

by

Rayn Sakaguchi

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

Leslie M. Collins, Supervisor

Loren W. Nolte

Guillermo Sapiro

Carlo Tomasi

Stacy Tantum

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Electrical and Computer
Engineering
in the Graduate School of Duke University
2016

Copyright © 2016 by Rayn Sakaguchi
All rights reserved except the rights granted by the
Creative Commons Attribution-Noncommercial Licence

Abstract

Current state of the art techniques for landmine detection in ground penetrating radar (GPR) utilize statistical methods to identify characteristics of a landmine response. This research makes use of 2-D slices of data in which subsurface landmine responses have hyperbolic shapes. Various methods from the field of visual image processing are adapted to the 2-D GPR data, producing superior landmine detection results. This research goes on to develop a physics-based GPR augmentation method motivated by current advances in visual object detection. This GPR specific augmentation is used to mitigate issues caused by insufficient training sets. This work shows that augmentation improves detection performance under training conditions that are normally very difficult. Finally, this work introduces the use of convolutional neural networks as a method to learn feature extraction parameters. These learned convolutional features outperform hand-designed features in GPR detection tasks. This work presents a number of methods, both borrowed from and motivated by the substantial work in visual image processing. The methods developed and presented in this work show an improvement in overall detection performance and introduce a method to improve the robustness of statistical classification.

Contents

Abstract	iv
List of Figures	viii
Acknowledgements	xix
1 Introduction	1
2 Background	7
2.1 GPR System and Data	7
2.2 Current GPR Processing Methods	9
2.2.1 Pre-processing and Pre-screening Methods	10
2.2.2 Pre-screener	12
2.2.3 Pre-Screener Alarm Classification Methods	14
2.3 Summary	18
3 Detecting Buried Threats Using Visual Image Processing Methods	19
3.1 Image Processing Methods Applied to GPR Data	20
3.1.1 Visual Image Discrimination Methods	20
3.1.2 Results	41
3.1.3 Conclusions	53
3.2 GPR Patch Representation	54
3.2.1 Feature Extraction Methods	55
3.2.2 Results	60

3.2.3	Conclusions	64
4	Physics-Based Augmentation of GPR Target Signatures	67
4.1	Variation in GPR Data	68
4.1.1	Variation in Target Configuration	68
4.1.2	Variation Due to Environmental Factors	69
4.2	Understanding the Effects of Varying Ground Height	72
4.2.1	Simulating Environmental Conditions	72
4.2.2	Real Data Examples of Rough Ground	74
4.3	Mitigation of Rough Ground	74
4.3.1	Reversing the Effects of Uneven Ground On Simulated and Real Data	76
4.3.2	Augmenting the Dataset With Varying Ground Surfaces	79
4.4	Dataset Expansion From One Observation Using Uneven Ground Aug- mentation	79
4.4.1	Dataset and Classification for One Observation Expansion . . .	81
4.4.2	Augmentation of Single Observation Results	81
4.5	Additional Augmentation Experiments	87
4.5.1	Augmentation of Dataset to Tested on Different Soil Type . . .	88
4.5.2	Augmentation of a Large Training Dataset	90
4.6	Conclusions	91
5	Convolutional Neural Networks	93
5.1	Convolutional Neural Network Model	93
5.1.1	Convolutional Layers	94
5.1.2	Fully Connected Layers	95
5.1.3	Training a Neural Network with Back-propagation	97
5.2	Application of Convolutional Neural Networks to GPR Data	99

5.2.1	Experimental Dataset	100
5.2.2	CNN Layer Parameters	101
5.3	Results	101
5.3.1	Alternative Classifiers	104
5.4	Conclusions	105
6	Conclusions and Future Work	107
6.1	Implementation of Methods Developed for Image Processing	107
6.2	Physics-Based Augmentation	109
6.3	Application of Convolutional Neural Nets to Detection in GPR Data	110
6.4	Future Work and Alternative Applications	110
	Bibliography	112
	Biography	119

List of Figures

1.1	An example GPR signal received at one antenna is shown in the figure above. This A-scan represents the received signal after a radar pulse is projected into the ground. The initial high energy response is due to the air/ground interface. Subsequent energy in the signal is primarily caused by sub-surface conditions including buried objects.	3
1.2	The above figure displays two examples of 2-Dimensional GPR data slices (B-scans). B-scans are composed of multiple A-scans (see Figure 1.1) collected along a line on the surface of the earth. The image on the left contains data taken from a non-target location. The signal response was generated due to natural clutter of some kind. The image on the right was taken from a known target location. The response's hyperbolic response shape is due to the sensor moving closer to and further away from the buried object.	4
2.1	The flow of the on-board processing architecture is shown here. The pre-screener identifies locations to be further analyzed by the feature extractor and classifier. The feature extractor calculates features from the GPR data around each pre-screener location. The classifier uses the features to calculate a confidence of how likely the pre-screener alarm contains a target response.	9
2.2	Figure 2.2(a) shows the original data as received from the on-board system hardware. Figure 2.2(b) shows the data after it has been aligned by the maximum energy peak. Figure 2.2(c) shows the data after the ground bounce has been removed as well as the lower reflections. The data has also been normalized to account for the low energy signal of deeper targets.	12
2.3	B-Scan of a target response and associated HMM states. A confidence value for a given B-scan would be based on the probability that the HMM progressed through the states shown above in the order shown left to right.	16

3.1	This figure presents the algorithm flow for instance matching between an example image and a cluttered scene. First (top left) feature points are extracted from both the example image and the cluttered scene. Next (top right), features are extracted from each patch located around a feature point. Finally (bottom), matching between points is performed using the features extracted from around each point. . . .	22
3.2	Figure 3.2(b) shows the corner response at each pixel given the original image in 3.2(a). Figure 3.2(c) shows the final selected points overlaid on the original image.	24
3.3	Figure 3.3(b) shows the response to the blob detector at each pixel given the original image shown in in 3.3(a). Figure 3.3(c) shows the final selected points overlaid on the original image after non-maximal supression.	25
3.4	A comparison of all three feature point detectors applied to a pair of target images. The Harris corner detector (a), tends to locate dots on the outer edges of the high energy areas. The blob detector (b), finds small high energy areas, or around the edges of larger high energy masses. The 2-D energy method (c), focuses purely on areas of high energy regardless of the surrounding area. Many of the points from the 2-D energy method lie on the central A-scan as that is typically the highest energy column of data.	26
3.5	Example generation of SIFT descriptors on a patch of GPR data. The patch of data taken from around a feature point location is shown in 3.5(a). The region is partitioned into the 4x4 grid of sub-patches. Plot 3.5(b) shows a quiver plot representing the gradients at each pixel location. Plot 3.5(c) shows the result after binning the gradients within each sub-patch. Each arrow in this plot originating from the center of the sub-patch represents the magnitude of the bin in that angular direction.	27
3.6	Example generation of a SURF descriptor on a patch of data. Plot 3.6(a) shows the patch of data around a feature point. The statistics of each sub-patch are shown in 3.6(b). The statistics of all 16 sub-patches will be concatenated to form a 64 dimensional feature vector.	28
3.7	The original patch of GPR data to be processed is shown in 3.7(a). The same data is smoothed and shown in 3.7(b). A small number of comparison features are also displayed. The gray arrows represent location pairs where the first location has lower intensity than the second. The black arrows represent pairs where the first location has higher intensity.	29

3.8	Example flow of the object detection method applied to face detection in images. Example training samples are generally obtained via hand labeling. The classifier is trained on the descriptor representation of all example images. Locations within new images are converted to their descriptor representation and classified using the trained classifier.	31
3.9	Patches #1 and 2 both exhibit the same geometric structure. However the gradient directions along each edge are reversed. The HOG descriptor will bin the gradients in the 90° and 270° in the same bin. The SIFT descriptor will bin the gradients in separate bins as the range for the sift angular bins goes from $0-360^\circ$.	33
3.10	Figure 3.10(a) shows possible decision lines that separate the two shown classes without error. Many of these lines however, are not desirable for classifying new unseen testing data points. The solid line shown in Figure 3.10(b) is the desired robust decision boundary with the maximum margin and is better suited to classify unseen data given these distributions.	36
3.11	Processing steps for the Bag-of-Words model applied to visual images. The patch vocabulary is generated based on a random sample of patches within the training data (top). Each image is in turn represented by the frequencies of each patch type (bottom). These frequency histograms are then used to represent the image for both training and testing stages.	40
3.12	The results above illustrate the performance of various descriptor methods over the best performing parameters utilized in instance matching. The results on the left show the performance of the descriptors without fusing with the pre-screener confidence. The results on the right include the pre-screener fusion that shows additional information is being introduced by the fusion with instance matching. These results show that without the pre-screener information, the SIFT descriptor method outperforms other descriptors. In contrast, BRIEF features, perform very badly without pre-screener information perform much better after fusion with the pre-screener.	43

3.13	The results above evaluate the performance of various point localization methods utilizing the best performing descriptors. The results on the left show the performance without fusion with the pre-screener confidence. The results on the right include the pre-screener fusion that shows additional information is being introduced by utilizing the instance matching confidences. These figures show the large disparity between performance with and without the fusion with the pre-screener, especially in relation to the BRIEF descriptors.	44
3.14	This figure illustrates matching between two different instances of the same target type. The target on the right is the closest match over all training alarms available. The lines represent a match was found between the given SIFT descriptor whose patch center is represented by the white dots.	44
3.15	This figure shows the down-track matching between a target and its top match, a false alarm. The second matching image in 3.15(b) shows the feature point matching between the same target and an instance of the same target type at the same depth at a different location. The feature points are not located similarly and the overall responses do not share all of the same visual characteristics. Thus the two B-scans were not matched well during testing	45
3.16	These results illustrate the differences in performance between the various descriptor methods. The best performing patch size is used in conjunction with the SVM classifier. The SIFT descriptor consistently performed the best among all methods tested despite the similarities to the HOG descriptor. The BRIEF descriptor performed poorly in both trials, especially when not fused with the pre-screener.	46
3.17	This figure shows the comparison between different classifiers utilizing the HOG descriptor results as shown above. The SVM classifier performed the best on all trials.	47

3.18	This figure shows the results of attempts to automatically generate isolated training examples from the training data instead of using hand labeled examples. It can be observed that the hand labeled training data continues to perform better at certain operating points, most notably at 0.003 FA/m^2 . In the fused results, the hand-labeled data would be the clear choice. However the method utilizing three different training examples per training image does have better performance at some operating points in the un-fused results. It is important to note that this is not necessarily a fair comparison. By having 3 training samples per positive image, there are three times the number of positive training samples.	48
3.19	The figures above illustrate the differences in centroids for different sized patches. The smaller patch centroids can only represent parts of the landmine response. The larger patch centroids are able to represent both the rising and falling edge of the landmine response allowing it to represent the entire landmine structure.	49
3.20	This figure shows the performance of the descriptors over a range of square patch sizes independent of the pre-screener confidence. Performance is measured by the probability of detection at a given false alarm rate (FAR). Each of the images above represent the performance at a given FAR value. The darker bars within each image represent the normalized patch's performance as a descriptor. The whitened patch performance was found to be more consistent across various patch sizes than the HOG method.	50
3.21	B-scans of the same target visualizing the likelihood ratios of the centroids representing each region. Each dot's intensity is determined by the rank of the centroids likelihood ratio. The centroids when using HOG descriptors as shown in (a) is much more consistent than the whitened patches shown in (b). These results were obtained using a 16×16 patch size, with a sampling rate of 4 in both dimensions. . . .	51
3.22	This figure helps visualize the likelihood ratios of centroids that represent regions of a false alarm B-scan. Note that the energy range in these images is much smaller than the images of a buried target. The HOG patches in (a) have a much higher response to these areas where the energy is low, but structure in the data recognized by HOG remains. In the whitened patches shown in (b), the likelihood is much less consistent.	52

3.23	This figure presents the top ROC operating points from the three different detection frameworks considered. All performance levels presented include fusion with the pre-screener confidences. The object detection framework provided the best performance over all methods. This is likely due to the ability this framework has shown to account for intra-class variation in image processing tasks.	53
3.24	This figure presents an example of processing a GPR patch of data using the EHD descriptor. The patch is separated into 7 vertical areas. Within each vertical area, the pixels are counted according to their edge type. The edge types are: horizontal (H), vertical (V), diagonal (D), anti-diagonal (AD), and non-edge (NE). The counts for each edge type are shown in the bottom panel for each overlapping vertical area.	56
3.25	The above images are the first 10 principal component vectors of all training patches, both target and non-target. All GPR data from the testing set were represented as a linear combination of the principal components creating the descriptor for each patch.	59
3.26	This figure shows the first 10 principal components for sub-patches. All sub-patches regardless of position within the image region utilize the same principal components. Each sub-patch was represented by a linear combination of these principal components.	60
3.27	This figure shows the best performance achieved using each of the feature extraction methods tested as shown on the y-axis. Gradient methods that produced an angular histogram or weighted histogram of the gradients in small local regions outperformed all other methods. LBP features, which share many traits with the top performing gradient methods also performed very well.	63
3.28	This figure shows the partial AUC performance of each feature extractor (y-axis) with and without fusion with the energy based pre-screener confidence. The increase in performance represents the additional information that the signal energy provides to each feature extraction method.	64
3.29	This figure presents the performance of each feature extractor method utilizing different classification algorithms. While some feature extraction methods had performance that was more consistent between classifiers, all methods performed the best utilizing the non-linear SVM classifier.	65

3.30	This figure presents the ROC curves from the top performing feature extraction method within each category. As previously shown, the gradient based method performs the best while the best extraction method based on raw pixel values fails to provide good performance. .	66
4.1	B-scans from four different types of non-metal cased landmines are shown in the above figure. Each column contains four GPR responses randomly selected from the same target type. Each of these target types have consistent distinguishing shape features differentiating them from each other. For instance, the response generated by landmine type 2 contains much less visible ringing than the other landmine types.	69
4.2	This figure displays a GPR response from a metal target that is buried under a flat surface, (a), and a response given the same target when buried beneath a soil mound seen in (b). Note that the presence of rough ground can substantially change the shape appearance of the GPR response. This shape warping is caused by time delays due to the additional soil the signal must pass through to reach the target.	70
4.3	This figure shows an example target response. The solid bar on the left of the target is a line spanning 13 time samples. A change in soil level between the target and ground of 1cm can cause a shift of 13 time samples.	71
4.4	The leftmost figure shows an example gprMax scene that is used for simulating GPR signals. The scene contains a rectangular metal object buried 3 inches deep in homogeneous soil. The resulting GPR response is shown on the right. This data was generated using similar parameters to those of the actual detection system used in this research.	73
4.5	These figures show the simulation of a rough surface with a metal target-like object. The leftmost image shows the physical configuration of the ground and target. The middle image shows the simulated GPR data resulting from (a). The image on the right is the pre-processed patch that would be the input into the feature extraction/classification system.	73

4.6	This figure shows actual data collected for this target type. The image in (a) shows data collected from an area with a relatively smooth ground surface. The image in (b) shows data collected over the same target but with an uneven ground surface (a mound). The target response extracted from the smooth ground GPR data is shown in (c), while (d) shows the extracted target response from the data collected over the rough ground. The shape is noticeably different between the responses shown in (c) and (d). This variation is important to capture in the training data if these conditions will occur in actual system operation.	75
4.7	This figure presents the warping method used to simulate targets emplaced under rough surfaces. The process begins with the simulated data under flat ground conditions shown on the left. The center image shows the data after it has been warped according to the ground contour from the data shown in Figure 4.5. The resulting patch shown on the right is the selected patch from this process that approximates the patch shown in Figure 4.5(c).	77
4.8	This figure shows the process of shifting A-scans from data collected over smooth ground conditions to appear as though it was collected over rough ground conditions. The image in (a) is the smooth ground data that will be shifted. The resulting GPR data after it has been shifted is shown in (b). The bottom row of images shows the extracted patches: (c). the original smooth ground GPR data, (d). the time shifted GPR that approximates uneven ground, (e). the field actual data collected under uneven ground conditions.	78
4.9	This figure shows example ground contours generated according to the process described in Section 4.3.2. The standard deviation of the filter determines the roughness of the contours while the multiplier determines the scale. The top plot consists of deformation samples with a low standard deviation parameter on the top, while the bottom plot has a much higher standard deviation.	80
4.10	Baseline performance detection on test set of PlasticAs. The solid black line shows performance if no PlasticAs are included in the training set. The dotted gray line is the performance when a single observation of a PlasticA is included in the training dataset.	82

4.11	Cross-validation performance when additional emplacements of PlasticAs are added to the training set. For each emplacement, all other test emplacements are added to the existing training set. The cross-validated performance is compared to detection performance with a single observation of the PlasticA in the training set. This performance shows the effectiveness of collecting additional data when possible.	82
4.12	This figure shows the single observation used in these tests. This target data will be replicated and warped to simulate a variety of ground conditions that can be used to improve the ability of the classifier to detect PlasticA targets	83
4.13	This figure shows example warped observation for a variety of parameters shown in the title of each subplot. These images can be compared to Figure 4.12 as they are all generated from the same data observation.	84
4.14	This figure presents the AUC results as a function of parameter values used to augment the single observation. The test set in this figure is composed of only PlasticAs and false alarms. Along the y and x-axis are the standard deviation values and multipliers used in the generation of distortion vectors. As a point of reference, the AUC performance of the classifier when using a single observation of PlasticA is 0.8862.	85
4.15	This figure presents the AUC of the augmented classifier on a test set consisting of non-PlasticA target types that are found in the training dataset. A black circle indicates that the parameter pair maintained performance above the AUC value of .957 that was achieved with no augmentation.	85
4.16	The figure above displays the AUC performance when multiple realizations of noise are added to the single PlasticA training observation and added to the training set. The performance does increase over no augmentation, but does not reach the level of augmentation with simulated rough ground surfaces.	86
4.17	This figure shows the ROC performance of augmentation with simulated rough ground surfaces as well as augmentation with noisy instantiations of the single observation. Baseline ROC plots are also presented here as a reference to performance with only the single observation as well as cross-validated performance if additional emplacements of PlasticAs are included in the training set.	87

4.18	This figure shows the results of training on data collected over one soil condition and tested on data collected over a different soil type. This result shows that when the training set utilizes the augmentation presented in this chapter, the performance is improved over no augmentation and an augmentation method developed to assist in mitigating differing dielectric constants.	89
4.19	This figure presents the performance obtained when a large-scale dataset uses augmentation within the training cross-validation folds. In the case of normalized pixel features, a slight performance improvement is achieved. However when SIFT features are extracted from each data patch, it was found that augmentation does not improve performance.	91
5.1	This figure displays example processing of an input patch by a convolutional layer. The input data (top left) is convolved with a bank of filters. The output of each filtering operation (top right) is then processed using the layers activation function. The output of the activation function (bottom left) is then locally aggregated across small pixel regions. The output of this layer (bottom right) can then be passed along to the next convolutional layer or rasterized to be further processed by the fully connected layers	95
5.2	This figure shows a graphical representation of a multi-layered neural network model. As the data is introduced into the fully connected layers at the top of the model, forward propagation flows downward. The connected lines indicate that the value of each node is dependent on all nodes in the previous layer. Each connection between nodes has a unique weight that dictates the amount of influence each node in the previous layer have.	96
5.3	This Figure shows the architecture used for the convolutional layers of the CNN. Due to the differences in spatial sampling, downtrack and crosstrack data need to have independent filters. After the convolutional layer the outputs are joined when supplied to the fully connected layers	102
5.4	This figure shows the training of a CNN where test-set performance begins to decline even though performance on the training set is still increasing. If only the training set is known during training, it is very difficult to determine when test set performance is optimized. This over-fitting of the training data is avoided in this research by using a validation set, a portion of the training data that is held out to determine where performance peaks.	103

5.5 This figure shows the performance of a CNN as the number of filters are varied. Performance of networks with relatively few filters is very good, showing the discrimination ability with a single filter of above 90%. The dotted line shows the performance when HOG features are used along with a non-linear SVM to classify the same patch pairs. In general the performance of the CNN is slightly lower than that achieved with the HOG features. 104

5.6 This figure shows the performance of features extracted using the CNN convolutional layers. The features extracted are the output from the convolutional layer of the CNN. These features from the training data are used to train a non-linear SVM classifier. The extracted features from the test dataset are then classified to calculate a confidence for each pair of patches. The resulting performance of this system is consistently higher than that achieved using a basic CNN. In addition, when enough filters are utilized, these filters out perform the HOG descriptors when the same classifier is used. 105

Acknowledgements

I would love to brag that I am where I am today because of hard work and persistence in the face of unimaginable obstacles. The truth is much less glamorous and relies on a bit of luck and the numerous people in my life that motivated me, helped me, and kept me sane throughout the PhD process.

MOTIVATION!

I was born to a loving family that have always believed in me and supported my education despite constantly losing me to a frigid, far away land known as the east coast. Mom, you've watched both your kids go off to school as far away as possible, yet you've always supported me and sounded excited about every new opportunity that further cemented my life on the east coast. Dad, we never talk as much as we'd both like, but I have always known that you are proud of me and everything I do. Ga and Grampy, I could not be where I am today without your love and support. Thanks for valuing my education before I did.

One of the biggest motivating factors was having a depressing job for a large aerospace company that helped me realize that I really wanted to do something else. This paired with an undergrad advisor who thought grad school was a great option. Dr. Ning, thank you for planting the original PhD seed in my head, and helping me navigate the application process 2 years later.

HELP!

I somehow stumbled onto the greatest lab ever. I've come into contact with a lot

of PhD students, but I have yet to hear of a place that even comes close to rivaling the SSPACISS group. Thank you Leslie, for accepting me sight unseen and for assembling the coolest group of PhD students. You have created a lab atmosphere of camaraderie and fun while still creating an environment that allows us to challenge each other to become the best scientists we can be. Thank you especially for your support when my personal life priorities were in conflict with the labs. You made it possible for me to continue on with my life and be a PhD student at the same time.

My lab experience would not have been what it was without the dynamic duo that is Pete and Kenny. Thanks for turning a guy with a BS who didn't take any undergrad DSP into a PhD who doesn't know any undergrad DSP but knows a little bit about machine learning/computer vision. You guys gave me enough room to explore some ideas and talked me down from the edge every time "none of it worked." I enjoyed our chats about random ideas and look forward to continuing them.

Of course any SSPACISS educational experience would not be complete without the research and presentation input from the other research staff. Sandy, Stacy, and Mary - thank you for all your input. You guys always said what needed to be said for my benefit. It was often scary, sometimes painful, but at the end of every meeting we've all had a few laughs and hopefully learned something.

Also a special shoutout to all of the friends and family outside of the lab that listened to my various exam practices. Even though you weren't familiar with any of the topics, you still managed to help me improve.

SANITY!

The graduate students of SSPPACISS are a diverse bunch of extraordinarily smart and talented individuals, but more importantly y'all kept me sane. Jordan, pretty much the sole reason I got any sleep my first year. Thanks so much for always having the time to explain random concepts I didn't get. Our meal time chats over chinese

food were always enjoyable. Chris, thanks for getting me out of the house. You put a lot of effort into organizing fun things like softball, bike rides, and of course some game involving a bag of wine. Sara, thanks for always reminding me of the world outside of work. You might even have more random hobbies/interests than I do. Ken, thanks for being the kind of friend that was always up for anything. Nothing could beat a good townhouse gathering. Jill, you could lighten the mood of any situation. Achut, your full body laughs were contagious. Patrick, I always figured you'd beat me out, thanks for being gracious about it. Boyla, thanks for reminding us how crazy Americans are. Dima, thanks for making things enjoyable because you brought excitement to absolutely everything you did. Nick, your discipline across all aspects of your life is admirable and often very confusing. Kedar, you're the happiest person I know. Joe, never fails in getting a laugh out of anyone. Daniel, I'm not even sure I know who you are anymore. Jillian, it was always fun when you were actually able to make it to something outside of working hours. And of course a special thanks to all of the wonderful people my lab mates chose as partners, Allie, Austin, Christine, Justin, and Lili - you guys are all kinds of awesome.

Outside of the lab I was lucky to be surrounded by awesome people as well. From my paintball teams, to my online video game teams, I have had unimaginable amounts of fun with you all. FrankerZ

Those who know me can probably spot the big hole in my story. I don't even know where to start with Alicia. The process of getting a PhD might have actually been more stress for her than me. She moved far away from the clutches of family and friends in New England just to be with me. It was a sacrifice that I will never forget. A ton has changed since then and I couldn't imagine being in a better place right now. We are so different in so many ways, but we tend to be what each other needs. I look forward to all our future adventures together. And of course my daughter Talyn. She has no idea how much it means to me to see her smile at the end of a

long day, or horse around on the bed in the morning before I have to leave. I look forward to a day when I can repay the favor and do my best to cheer her up when she is facing down her own challenges.

1

Introduction

Buried victim activated explosive devices such as landmines are often placed during times of conflict. While the intended target is most often enemy forces, civilians currently suffer the majority of injuries and deaths as a result of these buried threats [1]. Efforts to clear landmines and educate populations have greatly reduced the number of incidents around the world. However, the threat of buried explosives continue to plague regions that are often still recovering from deadly conflicts by restricting access to vital resources.

While global efforts have succeeded in banning the use of anti-personnel landmines, the use of anti-vehicle landmines is not currently restricted. These landmines in particular, often placed on roads, can have a drastic effect on the economy of a recovering region long after conflict has ended. The restricted use of vehicles hampers economic development and can impede access to humanitarian aid [2]. In current conflicts, vehicular targeted explosives cause the majority of injuries inflicted on American troops serving in Afghanistan [3]. This research focuses on improving the detection of buried anti-vehicular explosive threats including landmines and improvised devices.

Traditionally, landmine detection has been performed using an electro magnetic induction (EMI) sensor to flag underground targets by inducing a current in the metal content of a buried threat. The resulting field that is induced can be sensed by a receiver coil. While this method is simple and detects many different landmine types, there are two significant issues that cannot be overcome by EMI sensors. The first problem is the manufacturing of low-metal landmines and the use of low metal components for IEDs. While most of these devices generally have a small amount of metal that can be detectable, the detector's threshold must be lowered so much such that very small metal clutter often causes false alarms [4]. The second, closely related problem is metallic clutter – often in areas of conflict large pieces of shrapnel, casings, and other high metal waste is distributed throughout the region. For example, according to data from the Cambodian Mine Action Center, of the approximately 6 million pieces dug up during land clearing in 2011, only 104,464 were actual explosives [5]. Due to problems such as these, a significant research effort has been focused on methods to mitigate the high number of false alarms caused by EMI sensors [6, 7]. However, these methods are typically either designed for offline processing or utilize high-fidelity sensors that are not widely deployed.

Due to these challenges, ultra-wideband Ground Penetrating Radar (GPR) has been introduced as a complimentary technology to EMI sensors for buried explosive threat detection [8, 9, 10]. GPR subsurface detection functions by first projecting an electronic radar pulse into the ground. As the pulse travels downwards through the ground, discontinuities in dielectric constants between materials cause a partial reflection of the pulse. The resulting reflected signal is collected near the transmitting antenna and referred to as an A-scan. An example of a received signal is shown in Figure 1.1. Notable features in this example are the two peaks discussed below. The first, largest, peak is caused by the air/ground interface which generally causes the highest energy reflection as it is the first major reflection and a point of high

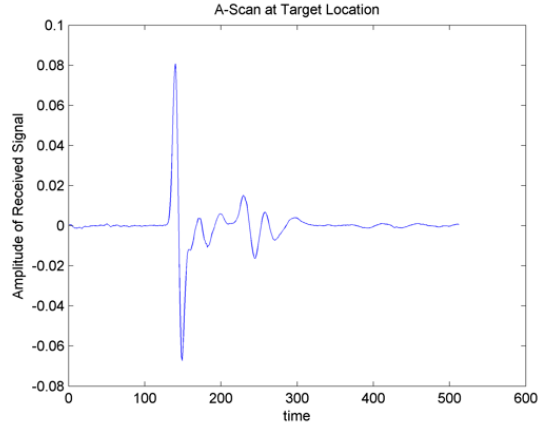


FIGURE 1.1: An example GPR signal received at one antenna is shown in the figure above. This A-scan represents the received signal after a radar pulse is projected into the ground. The initial high energy response is due to the air/ground interface. Subsequent energy in the signal is primarily caused by sub-surface conditions including buried objects.

dielectric discontinuity. In addition, due to the strong reflection, it reduces the energy of the remaining signal that progresses downwards. The second reflection in this example is caused by a buried metallic target. The magnitude of a targets response is dependent on a multitude of factors including target depth, antenna height, soil dielectric constant, soil moisture content, homogeneity of soil, and the physical makeup of the target.

While characteristics of explosive threats can be seen in single A-scans [11], most processing methods make use of the spatial characteristics of a set of A-scans near an object [10, 12, 13]. These spatial characteristics can be observed by viewing GPR data as a 2-D B-scan as seen in Figure 1.2. The hyperbolic shape is generated by the changing time it takes for the reflected signal to reach the antennae as the sensor location moves with respect to the buried object. Most algorithms, with a few exceptions [11, 14] specifically extract features that are designed to respond to hyperbolic-like shapes seen in B-scans containing targets.

The goal of this research is to improve methods currently used to detect shape

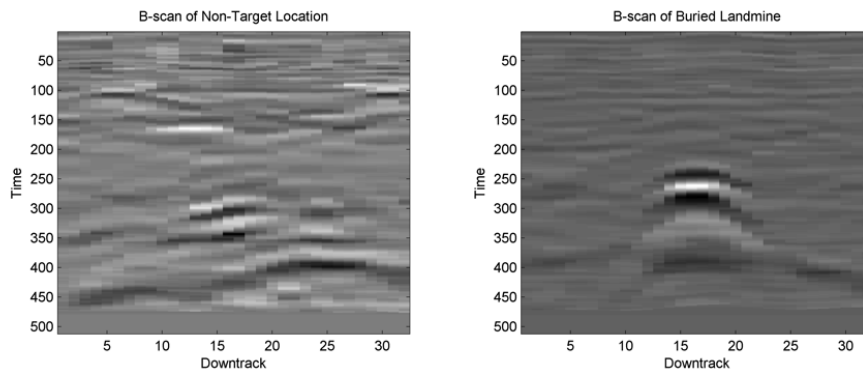


FIGURE 1.2: The above figure displays two examples of 2-Dimensional GPR data slices (B-scans). B-scans are composed of multiple A-scans (see Figure 1.1) collected along a line on the surface of the earth. The image on the left contains data taken from a non-target location. The signal response was generated due to natural clutter of some kind. The image on the right was taken from a known target location. The response’s hyperbolic response shape is due to the sensor moving closer to and further away from the buried object.

features in GPR data. Chapter 3 presents the application of methods developed specifically for finding objects within, or classifying visual images. Various detection architectures as well as features and classification methods were tested. Careful application of these methods shows improved performance over current state of the art GPR detection methods. While these methods provide state of the art performance, this research also identifies problems with these approaches.

This research addresses problems with current detection systems. The first is that these statistical detection methods all rely heavily on a varied training set that encompasses the entire statistical variability of landmine and clutter instances possible in operating conditions. In practice, given the scale of parameters involved with GPR data formation and the expense of data collection, data is only collected under a few operating conditions for various target types. This causes the potential for performance to suffer when conditions that are not in the training set are encountered in fielded operations. In addition, there may arise situations where very limited data is available. This situation can arise if a target is missed and detonates. The only

information about the target is the sensor data that was not flagged as a target. This research aims to use physics-based methods specific to GPR data in order to expand the training set and simulate conditions that were not actually encountered in the training data.

This research will focus on two areas of variance that can heavily affect a GPR target response. The first is uneven ground, or ground roughness. Ground roughness is known to be able to distort a GPR signal [15, 16]. However it is difficult to collect training data across all possible ground contours over targets. This research relies on random generation of rough ground conditions that are then applied to current training samples in order to simulate the collection of additional data observations. The second source of variance that this research addresses is changes in soil dielectric constant.

The second issue with current state-of-the-art processing is that features being used were not designed specifically for GPR data [13, 17]. These methods were tuned specifically to be invariant to the types of transformations common in visual images such as rotation and view angle. GPR responses are not subject to these same invariances and therefore could be throwing out information that is not considered important in visual image representations. This research evaluates the use of convolutional neural networks in order to learn a set of filters that optimize the discrimination between target and non-target GPR data.

The remainder of this document is organized as follows: Chapter 2 introduces the sensing system and current methods of processing data. These methods include pre-processing methods that are used in this research as well. Chapter 3 discusses research into testing and adapting methods from the image processing field to improve classification of threats in GPR data. This chapter is organized into two parts, the first part tests various image detection architectures. The second part presents results on the effectiveness of a variety of feature extraction and classification tech-

niques. Chapter 4 presents work on the augmentation of GPR datasets in order to provide a more robust detection system. Chapter 5 presents research into the application of convolutional neural networks to learn features and discriminate between target/non-target locations. Finally Chapter 6 discusses the contributions and conclusions as a result of this work. This chapter also discusses future avenues of research given the results of this research.

2

Background

This section will provide background on the data available and system used for this research. Current methods of threat detection in GPR data will be reviewed with emphasis on current algorithms implemented in fielded systems.

2.1 GPR System and Data

The data used for this research was collected with a vehicle-mounted array of GPR sensors manufactured by NIITEK Inc. The sensing system consists of a 51-channel array of sensors mounted perpendicularly to the vehicle's direction of travel. Sensors are placed at 6 cm intervals along a panel that spans 3 meters and is mounted in front of the vehicle. The ground is sampled at 5cm intervals as the vehicle moves forward. The direction of vehicle travel is referred to as the down-track direction while the perpendicular direction along the surface of the ground is referred to as the cross-track direction. The reflected GPR signal at each physical location is recorded temporally for a total of 512 time samples. The resulting data available for processing at each down-track location is 512 time samples by 51 channels. Given these parameters, the amount of data collected grows quickly; the total amount of

data collected from 50 meters of travel results in a matrix of numerical doubles of size 512x51x1000.

Since the detection system is mounted on a moving vehicle, threats need to be detected quickly as there is very limited time between when the panel passes over an object and when the wheels could potentially trigger the object to detonate. Near real time processing of the data is therefore a necessity in order to keep the operator safe. Thus, in order to be able to process more complex features, the amount of data sent to higher level processing algorithms needs to be reduced. This constraint is addressed through the use of a pre-screening algorithm that detects anomalies in subsurface energy content [18]. Locations of interest on the ground surface are identified by the pre-screening algorithm; the data associated with the location of interest is then processed by a suite of feature processing algorithms whose outputs are then fused. Data at locations not flagged by this pre-screening algorithm are not reviewed by any other processing methods. Therefore, the pre-screening algorithm is designed to operate with high sensitivity to ensure it achieves a high probability of detection while potentially allowing a moderate number of non-target areas to be flagged. This two-stage algorithm allows for the more computationally expensive feature processing methods to process smaller amounts of data, which allows the system to operate within the time constraints. While there is still a time requirement for the algorithms to complete, the two step process allows more flexibility in the feature processing computational requirements. A diagram of the system processing flow is shown in Figure 2.1. Numerous methods have been proposed and evaluated for use within the framework shown in Figure 2.1. Section 2.2 will review current methods of GPR processing, several of which are currently employed on the fielded on-board processing system.

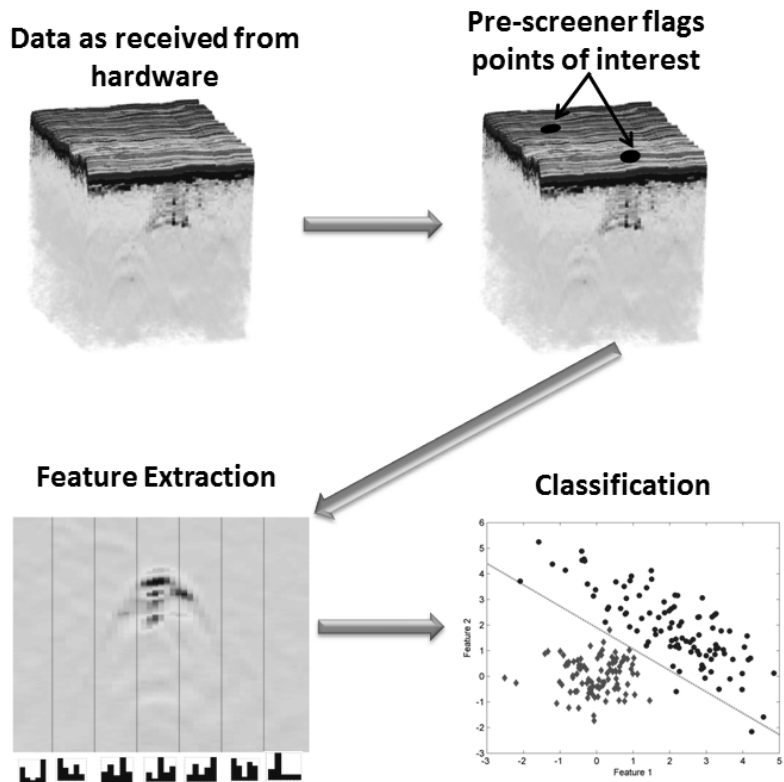


FIGURE 2.1: The flow of the on-board processing architecture is shown here. The pre-screener identifies locations to be further analyzed by the feature extractor and classifier. The feature extractor calculates features from the GPR data around each pre-screener location. The classifier uses the features to calculate a confidence of how likely the pre-screener alarm contains a target response.

2.2 Current GPR Processing Methods

This section reviews several methods that have been shown to be successful in processing GPR data for buried target detection in real time scenarios. This also section introduces standard pre-processing and pre-screening methods that have been utilized in this research. In addition, a variety of current feature processing systems are introduced with emphasis on those that have shown good performance and subsequently been implemented on the fielded detection system.

2.2.1 Pre-processing and Pre-screening Methods

Pre-processing Methods

Pre-processing the GPR data serves to reduce the effect of certain GPR characteristics that make it difficult to discriminate between targets and clutter. Three major issues are introduced and addressed in this section. The first, and one of the most challenging features for target detection in GPR is the ground-bounce, or the initial spike in energy caused by the air ground interface. The energy of the ground-bounce is generally the highest energy peak in a GPR A-scan. As most methods of GPR processing respond to areas of high energy and the ground contour can often appear hyperbolic, false alarms are commonly generated in locations where the ground-bounce is not removed properly. As a result, most GPR based systems employ some method to remove the ground-bounce from the data [19].

The second issue, one that affects the shape of a target in 2-D B-scans, is the potentially changing height of the antenna above the ground. If the antenna height changes as a sensor passes over an object, the number of time samples before the reflected signal is received at the antenna will change. Due to the GPR antenna being extended in front of a vehicle and this vehicle's use on often rough unpaved roads, the change in antenna height can be sudden and dramatic. This will cause variation in the time sample at which the target reflection is received. This leads to inconsistencies in the hyperbolic shape generated by a sub-surface object in the GPR data.

The third and final issue that pre-processing addresses is the attenuation caused by the propagation through the ground and the loss of energy due to reflections caused by heterogeneous soils [18]. This causes deep target responses to have much lower magnitude than targets found near the surface. In order to use features that are dependent on the energy of the GPR reflections at all depths, a GPR signal at a

later time sample needs to be comparable in energy to one near the ground surface.

This work utilizes previously developed methods [18] that are currently standard practice when processing data from this system. To remove the ground-bounce from a section of GPR data, it must be first identified. In downward looking GPR, it is usually a reasonable assumption that the time index associated with the highest energy within each A-scan is the ground-bounce since it is the first major dielectric discontinuity encountered by the GPR signal. In ground-bounce removal, all A-scans are aligned so that the maximum amplitude index occurs at the same time sample. Then, the all data above and including the ground-bounce is discarded. This process results in the latest time samples of consecutive A-scans to be mis-aligned. The time samples where misalignments are present are not considered in processing and are removed. The resulting trimmed data is then treated as the ground-bounce removed data.

The issue of changes in antenna height is also solved by aligning the data to the ground-bounce. When the antenna changes height, the signal propagation time between the antenna and the ground changes. However, the signal propagation time between the top of the ground and the subsurface is minimally affected by the changes in antenna height. Therefore, using the aligned data with the ground-bounce removed ensures the time indexes are in relation to the first contact with the air-ground interface.

In order to mitigate the final issue with signal strength at later time samples, the data must be normalized so that targets have approximately the same characteristics. In this work, depth normalization is performed based on the statistics of the surrounding data [18]. Independently at each time sample below the ground-bounce, the background mean and standard deviation are determined by data surrounding the area being whitened. To normalize the data at each point, the mean is subtracted and then the sample is divided by the standard deviation. After normalization, the

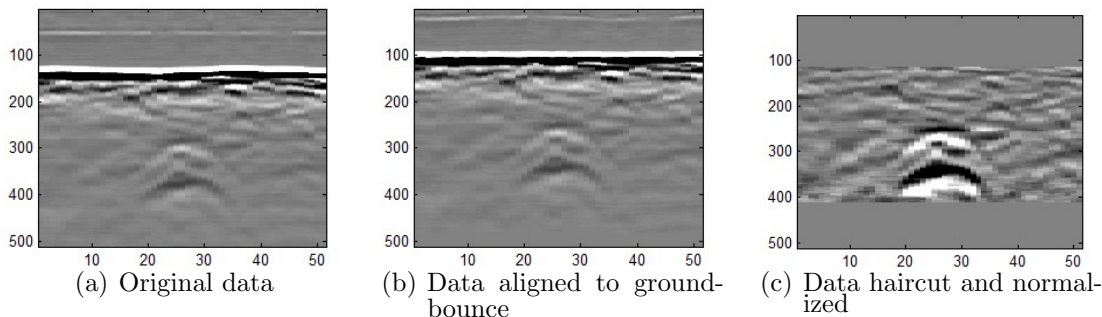


FIGURE 2.2: Figure 2.2(a) shows the original data as received from the on-board system hardware. Figure 2.2(b) shows the data after it has been aligned by the maximum energy peak. Figure 2.2(c) shows the data after the ground bounce has been removed as well as the lower reflections. The data has also been normalized to account for the low energy signal of deeper targets.

data's signal strength is more consistent throughout all depths as the data strength is based on its relation to the statistics at the same depth below the air-ground interface.

The output of these pre-processing steps is data that has been ground-bounce aligned; ground-bounce removed; and normalized by depth. An example of the pre-processing progression is shown in Figure 2.2 where a low energy deep target in the raw data has been pre-processed. Most pre-screeners and algorithms utilize these pre-processing steps so that features extracted can be easily compared across depths.

2.2.2 Pre-screener

Due to the necessity of real time detection and the benefit of computationally expensive feature extraction techniques, a low complexity pre-screener is used to flag locations of anomalous responses and thus decrease the amount of data necessary for other algorithms to process [18]. The pre-screener identifies anomalies and records the positions, which are termed pre-screener alarms. This pre-screener is designed to achieve a high probability of detection while allowing for a moderate false alarm rate

in favor of simplified processing. The only sections of data that other algorithms will process are the ones located near pre-screener alarm locations. Using a pre-screener in this manner is similar to cascading classifiers that aim to eliminate the maximum number of observations in each step using simple classifiers. This method is often used when processing time is an important consideration and data needs to be eliminated from consideration at each step [20].

The pre-screener employed by the fielded system considered in this work is based on a least mean squares algorithm [18]. The algorithm operates independently at various signal response time bins and utilizes the aligned data with the ground-bounce removed. The LMS-based algorithm operates by linearly predicting the signal data at a given location and time using knowledge of previous data. The difference between the estimate and the actual data is the output of the LMS-based algorithm. Smoothing, processing by depth, and thresholding are also applied to the output of the LMS-based algorithm. Non-maximal suppression is then performed as a final step to identify the specific location of the anomaly. Further information about specific parameters can be found in [18] where this method is introduced.

Pre-screener Fusion When creating a new type of discrimination algorithm for the detection of buried threats, it is important to consider the algorithm's capability to provide information that is complementary to the information provided by the pre-screener. The pre-screener is extremely effective at flagging subsurface objects as they generally produce a reflection with enough energy to be statistically differentiated from nearby data. Therefore an algorithm that produces a confidence highly correlated with energy features, while it may perform well, may not necessarily provide additional information to the system.

It is for this reason that fusing algorithm results with a pre-screener can offer better insight into the amount of new information being provided to the system. In

addition, as the pre-screener confidence is already calculated for each pre-screener alarm it is computationally fast to fuse the confidences of two systems with a classifier. Therefore, as algorithms are evaluated in this research with the goal of providing additional information to the detection system, results will be fused with the pre-screener confidences unless otherwise noted.

2.2.3 Pre-Screener Alarm Classification Methods

This section introduces current methods of GPR processing to detect landmines and other buried explosive threats. These methods are generally designed to serve as the sole, or final method of classification in the case of a two-stage processing system.

Model Inversion Methods

While numerous approaches to generating and processing GPR data exist, one of the most potentially accurate methods of landmine detection is via electromagnetic inversion. By inverting the signal, it is possible to reconstruct the actual physical shape of the object and its dimensions under certain assumptions [21, 22, 23, 24]. In a landmine detection task, this would involve inverting the received data over spatial area so that instead of a hyperbolic shape, the actual subsurface shapes could be imaged. Trials involving buried landmines have shown good performance under homogeneous soils such as sand [24]. These methods however, often rely on models that assume a homogeneous medium; when faced with a medium that is not well known, the resulting inverted image degrades substantially [24]. While methods have been developed to compute the subsurface characteristics, these methods are very computationally expensive and have limitations in the kinds of characteristics they can predict [25]. Methods introduced to speed up the inversion process for landmine detection [23, 21] generally work to focus the image. These methods however, still require assumptions of soil homogeneity that when violated degrade the landmine

image. Therefore, using these inversion methods still requires a statistical method to model the uncertainty in the signal after inversion. This further adds to the processing time necessary to classify a pre-screener alarm. It is for this reason that most successful methods of fast GPR processing have been designed as statistical methods that identify landmine responses in non-inverted data.

Statistical Methods

In order to develop methods that account for the variability and uncertainty commonly seen in buried target responses, researchers have developed many methods that rely on statistical modeling. These statistically based methods search for indications of buried objects in the data without attempting to invert the signal. These methods are generally composed of two steps, feature extraction, and classification based upon the feature vectors. The following sections will describe a number of statistical methods developed for the fielded GPR detection system.

Hidden Markov Models [12] Hidden Markov Models (HMMs) have shown to be a powerful tool when modeling non-stationary random processes in a variety of applications [26, 27, 28]. The HMM model assumes that data is generated according to the current hidden state of the system, and by the distributions associated with that state. This state information can be exploited to determine a common sequence of hidden states that can signify an event. HMMs used in GPR data processing attempt to identify a sequence of hidden states that signify a hyperbolic target response as illustrated in target B-scans. The states associated with a target response are illustrated in Figure 2.3.

Transition probabilities between states as well as the distributions associated with each state are learned from example B-scans in the training dataset. In evaluating a given B-scan, the algorithm processes the data from the left side of the image to

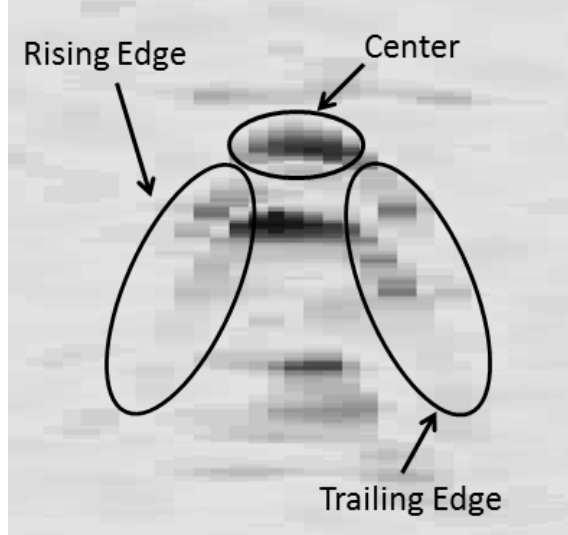


FIGURE 2.3: B-Scan of a target response and associated HMM states. A confidence value for a given B-scan would be based on the probability that the HMM progressed through the states shown above in the order shown left to right.

the right side. In testing, the confidence assigned by the HMM algorithm is the probability, given the features that are extracted, that the model progressed from background to rising edge, center, trailing edge, and finally back to background. This is known as the mine model. An important constraint on the model is that in order for the model to reach the trailing edge state, the model must have already progressed through the rising edge and center states. The final confidence of each B-scan is then the difference between the log-likelihood of the observations given a mine model and a background model.

Edge Histogram Descriptors Applied to GPR Data Edge Histogram Descriptors (EHD) were originally used as a visual texture descriptor that numerically quantifies the types of edges in images [29]. EHD features have also been utilized as a descriptor in image matching tasks [30, 31]. Similar to these applications, the EHD algorithm used in landmine detection also uses GPR 2-D B-scans to detect buried threats [13].

Similarly to the HMM method of target detection, using EHD also relies on

the detection of the hyperbolic shapes in a B-scan at target locations. For a given pre-screener alarm, multiple B-scan regions are extracted from the sensors nearby an alarm location. EHD features are generated for 7 overlapping columns of data within each B-scan section. The goal of this approach is to observe the differences in edge types across each alarm location flagged by the pre-screener. Since a typical target response has edges that change direction as a sensor passes over an object, the change in EHD descriptors across the response will reflect this. Training of the classifier is performed by collecting regions of B-scans that contain target responses and regions that contain background data. In order to avoid utilizing background sections of the data as training examples, it is useful to know the time at which the target signature begins to appear in the data. For training data, this localization is performed manually. Each target response in the training data must be manually selected by a human expert. At each alarm index used during testing, the data is broken up into 10 overlapping time regions and each is separately given a confidence value by the classifier. Order statistics [32] are then used to determine the final confidence associated with each pre-screener alarm.

Spectral Characteristics Feature The Spectral Characteristic Features (SCF) [11] method, unlike the HMM and EHD methods, is not designed to leverage the shape characteristics visible in a B-scan signal as the GPR is moved spatially over a buried object. Instead, SCF considers the individual A-scans for indications of a target. Through experiments using finite-difference time-domain (FDTD) simulations it is possible to observe the characteristics in the Energy Density Spectrum (EDS) generated by different kinds of shapes and materials under varying conditions. It was observed that the EDS generated from a given target type varied little under conditions of varying depth and soil dielectric constant.

Using the results of these simulations, SCF was tested on real data taken from

various testing sites. Similar to results seen in the the simulations, it was observed that the correlation between the SCF of targets in varying conditions remained high. Since the EDS calculation mitigates the uncertainty in depth and soil type, the classification of targets is done simply by correlating matched filters generated from known objects in the training dataset, with the test data. In testing, to reduce the effect of noise, the EDSs from A-scans near a pre-screener alarm are averaged together. The output of the averaged matched filter responses from around a pre-screener alarm determine the final confidence provided by the SCF method.

2.3 Summary

The methods presented in this chapter describe various approaches to detecting buried threats in GPR data. Inversion methods, while potentially the most accurate, suffer from a dependence on knowledge of the subsurface conditions. Statistical methods generalize very well by relying on large amounts of training data to encompass the variation in subsurface conditions.

The goal of the work presented in this document is to evaluate methods found in the image processing field to further improve the detection of buried threats in GPR data. Detection tasks in visual images is currently a very popular research topic with a variety of approaches under consideration. This work aims at adapting high performing visual detection methods and adapting them to perform well under the conditions present in GPR data.

Detecting Buried Threats Using Visual Image Processing Methods

The desire for computers to understand the physical world has driven the development of various techniques for information extraction from images. As a result, the detection of types of objects in visual images has been widely researched since the advent of digital imaging. This chapter describes the utilization of methods developed for the detection of objects in visual images in order to detect buried threats in 2-D GPR B-scans. Section 3.1 presents different methods of visual image processing that show promise in the detection of landmines from GPR data. This section reviews these methods and presents results showing the differences in performance between them. Section 3.2 delves deeper into the method that shows the best performance in detecting GPR signatures and performs analysis on the various methods used within. The outcome of the research presented in this chapter is a detection system that outperforms current state of the art detection methods over a large benchmark GPR dataset.

3.1 Image Processing Methods Applied to GPR Data

The desire for computers to understand the physical world has driven the development of various techniques for information extraction from images. The primary goal of this research is to leverage image-based technologies that identify shapes and objects and apply them to find shapes and objects within GPR B-scans. This chapter describes our work aimed at evaluating a number of image processing methods to develop a framework applicable to finding buried threats in GPR data.

3.1.1 Visual Image Discrimination Methods

This section introduces the three image discrimination methods evaluated in this work: (1) Instance recognition is the process of identifying occurrences of a single identical object, (2) object detection attempts to find a particular class of objects within images, and (3) image classification attempts to classify whole images as having certain properties [33]. This research aims to evaluate all three methods and their applicability to target detection using GPR data and improve upon this baseline performance.

Instance Matching

Instance matching aims to recognize instances of a known object within a scene. An example of this would be attempting to find all instances of a particular book within a visual scene. The specific title text and the cover design of the book can be used for recognizing it in a scene. It is assumed in this approach that the algorithm has an isolated image example of the known object. While there are applications where this problem is simply a template matching problem, modern research in this field is focused on accounting for various types of transformations commonly found in visual images. Scale, intensity, rotation, and occlusion are all common types of transformations that affect how an object appears within an image [34]. Successful

methods for instance matching must be able to identify instances of the object under these types of transformations.

The main contribution in image processing literature that allowed algorithms to reliably detect these transformations was the use of a patch based model where matching occurs between small discriminable patches of the images instead of whole objects [33]. This method not only allows for more efficient processing of the data, but also allows for partial matches and accurate detection in the case of occlusion. The patches on an object and within a scene that are selected for processing are based upon their ability to be located repeatedly and their ability to be distinguished from other patches of data. These patches are located at points known as feature points. Research has shown that corner detectors and blob detectors find the most informative image patches to be used for subsequent processing [35]. Corner detectors are designed to find areas where two edges in an image meet [36], while blob detectors look for areas within an image that are unlike the immediate area around them [37, 38, 39]. Both of these methods provide feature points with excellent localization and also generally find areas within an image that are indicative of the object they are attempting to identify.

Although patches are selected for their ability to repeatedly localize on the same physical point on an object, errors of a few pixels remain common. This is one of the reasons why an entity termed a "descriptor" is used to describe each patch of data instead of simply using the vectorized pixel intensity values. The descriptor's function in instance matching is to provide a relatively low-dimensional representation of the data within a patch while being invariant to intensity and minor pixel differences in localization of the patch.

In most modern techniques for image instance matching, the matching process begins with interest points located on the example image as well in the image of the scene in which the object is to be detected. The descriptors are calculated for

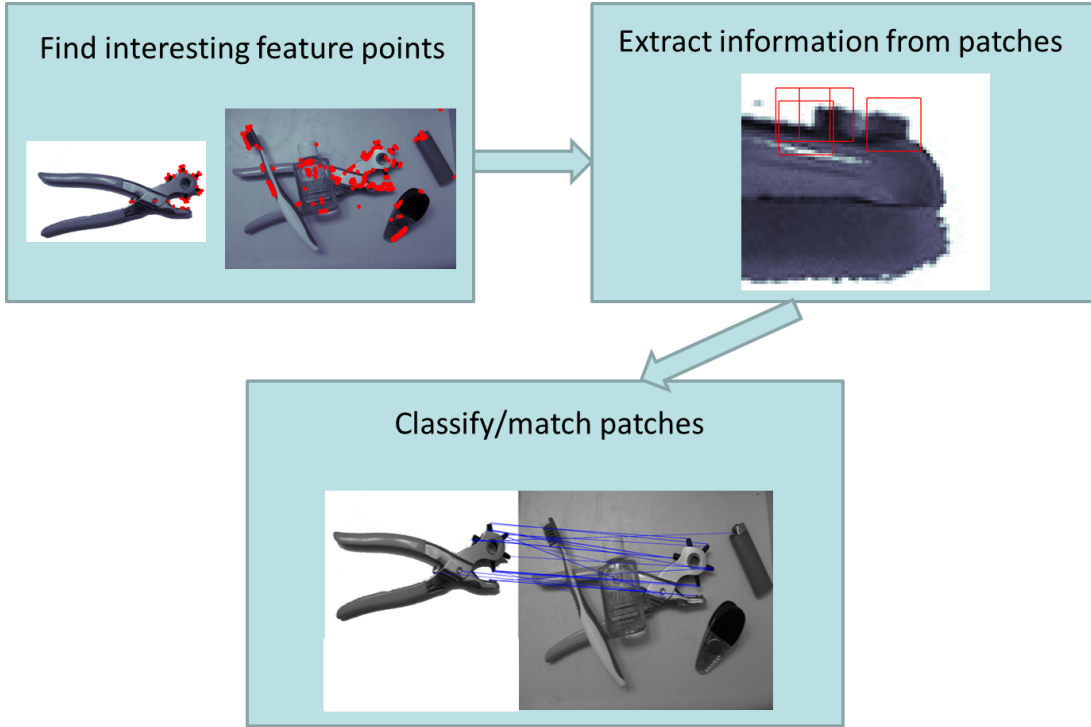


FIGURE 3.1: This figure presents the algorithm flow for instance matching between an example image and a cluttered scene. First (top left) feature points are extracted from both the example image and the cluttered scene. Next (top right), features are extracted from each patch located around a feature point. Finally (bottom), matching between points is performed using the features extracted from around each point.

each patch where a feature point is generated. To evaluate a scene, descriptors at each feature point within the scene are compared to the descriptors generated from training examples. A match is made if the distance between an example descriptor and one found in the scene is lower than a predefined threshold. These steps are illustrated in Figure 3.1 where the process of instance matching is shown with an example of finding a specific object (left) in a cluttered scene.

In this research, a number of feature point detection and descriptor methods were evaluated for instance matching in GPR data. Harris corner points [36] and a blob detector based on the determinant of the Hessian [38] were both used since their performance is generally good across a variety of applications [35]. In addition, this

work also introduces a feature point method designed specifically for GPR data based upon the smoothed energy of a 2-D B-scan. Descriptors evaluated for this method were adapted from the Scale Invariant Feature Transform (SIFT) [37], Speeded Up Robust Feature (SURF) [38], and the Binary Robust Independent Elementary Features (BRIEF) [40]. The SIFT descriptor was chosen for its good performance in tests [41], while SURF is intended to represent a computationally less expensive version of SIFT. BRIEF was chosen since it provides a complementary approach and an invariance to variation in pixel intensity differences. These different approaches are all described in the following sections.

Feature Point: Harris Corner Detector [36] The Harris corner detector is based on previous work performed by Hans Moravec [42] where a corner is found within an image by shifting a window around each pixel and noting the intensity changes. If a corner is present, the intensity should change in all directions. If the pixel lies on a single edge, or a constant area, at least one shift direction will produce a low intensity difference. Harris and Stephens proposed a method that works similarly, but uses the gradients around each pixel to approximate the sum of squared differences (SSD) surface that models the change in intensity as a window is shifted around a pixel. The equation below defines the SSD given a shift of $\bar{\phi}$ using a Taylor series approximation

$$SSD(\bar{x}_o, \bar{\phi}) \approx \bar{\phi}^T \sum_{\bar{x} \in w(\bar{x}_o)} [g(\bar{x} - \bar{x}_o)A(\bar{x})]\bar{\phi}, \quad (3.1)$$

where $w(\bar{x}_o)$ is the window around a point location \bar{x}_o ; and g is a Gaussian function. Note here that \bar{x}_o is a two dimensional vector to indicate location within the image space. At each location, A is defined as

$$A(\bar{x}) = \begin{pmatrix} I_x(\bar{x})^2 & I_x(\bar{x})I_y(\bar{x}) \\ I_y(\bar{x})I_x(\bar{x}) & I_y(\bar{x})^2 \end{pmatrix}$$

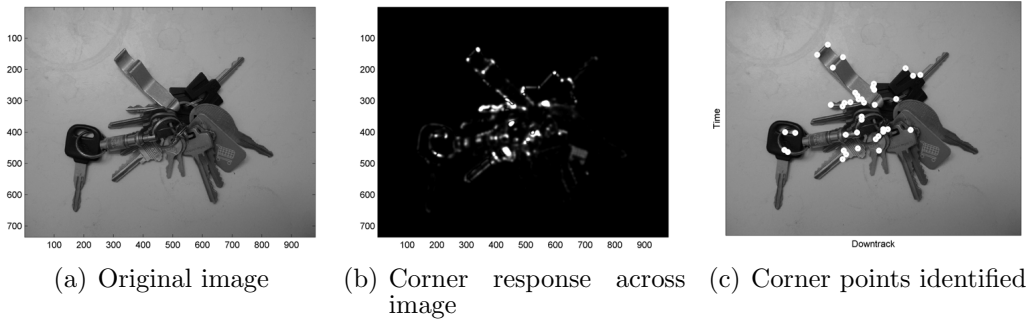


FIGURE 3.2: Figure 3.2(b) shows the corner response at each pixel given the original image in 3.2(a). Figure 3.2(c) shows the final selected points overlaid on the original image.

where $I_x(\bar{x})$ is the derivative of the image, I , in the x-direction. The corner response at each pixel is then defined by Harris and Stephens [36] as:

$$R(\bar{x}) = \det(A(\bar{x})) - k(\text{tr}(A(\bar{x})))^2,$$

where k is small constant tuned to a specific application.

Given the corner response for each pixel across the image, only local maxima are considered to be possible corner points. Those points that remain must then be above a given threshold to be recognized as a corner point by the algorithm. Figure 3.2 shows the results of the process of detecting corner points on an example image.

Feature Point: Blob Detector [38] A blob detector is designed to identify areas where the intensity within the region differs greatly from the surrounding area. This can indicate salient locations such as corners or small areas of high contrast. The blob detector that was utilized for this research is the determinant of the Hessian method that was used in the implementation of SURF [38]. This algorithm operates by calculating the Hessian matrix at each pixel location after a 2-dimensional Gaussian

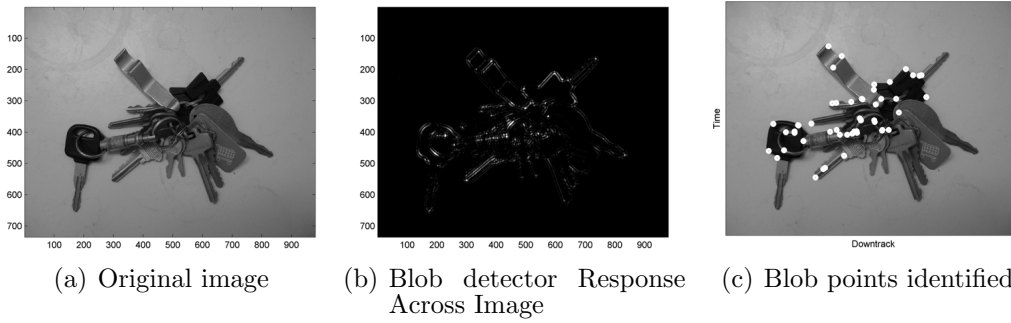


FIGURE 3.3: Figure 3.3(b) shows the response to the blob detector at each pixel given the original image shown in in 3.3(a). Figure 3.3(c) shows the final selected points overlaid on the original image after non-maximal suppression.

smoothing of the image.

$$Hessian(\bar{x}) = \begin{pmatrix} I_{xx}(\bar{x}) & I_{xy}(\bar{x}) \\ I_{yx}(\bar{x}) & I_{yy}(\bar{x}) \end{pmatrix}$$

where $I_{xx}(\bar{x})$ is the second derivative in the horizontal direction at the image pixel $I(\bar{x})$ within the image I . If the determinant of the Hessian is positive, the location is either a local minimum, or a local maximum of the function depending on the sign of I_{xx} and I_{yy} . Each pixel's response to the blob detector is represented by the determinant of the Hessian matrix at each pixel. Local maxima and minima of the blob detector that surpass a given threshold are then identified as blob locations. Figure 3.3 shows the result of processing by the blob detector on an example image. Note that a blob detector will also produce feature points on corners as corner pixels share characteristics that blob detectors respond to.

Feature Point: 2-D Energy Based Detector In addition to the feature point methods utilized from the visual image processing field, a new feature point method was developed for use on GPR data. Instead of finding corners or blobs, this method aims to identify peaks of high energy. Generally in visual images, high energy locations are not used as feature points as they can often be a result of light source reflections

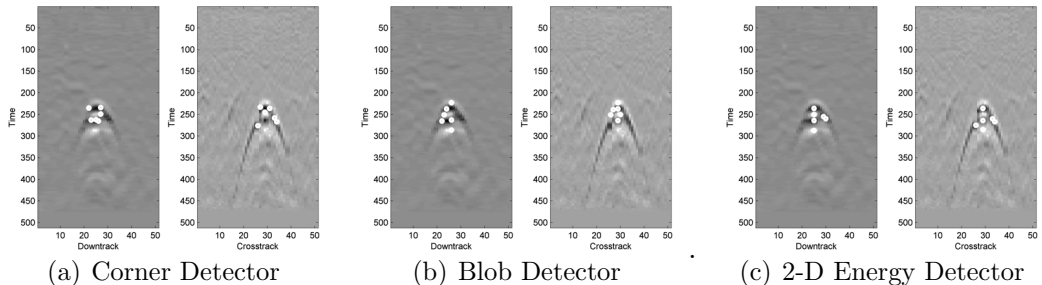


FIGURE 3.4: A comparison of all three feature point detectors applied to a pair of target images. The Harris corner detector (a), tends to locate dots on the outer edges of the high energy areas. The blob detector (b), finds small high energy areas, or around the edges of larger high energy masses. The 2-D energy method (c), focuses purely on areas of high energy regardless of the surrounding area. Many of the points from the 2-D energy method lie on the central A-scan as that is typically the highest energy column of data.

that are not indicative of aspects of the object which the reflection is located on. In GPR data however, high energy areas are desirable as they often occur as a result of a change in conducting materials, such as soil to metal. Therefore, by identifying areas of high energy in a 2-D B-scan, data caused by subsurface anomalies are processed. In order to find these areas, the energy of B-scan is calculated then smoothed with a Gaussian filter to reduce random noise. Non-maximal suppression is performed to find the local peaks. The top energy peaks are kept as feature points based on their energy value.

A comparison of the feature points extracted using each of the three methods is shown in Figure 3.4.

Descriptor: SIFT The Scale Invariant Feature Transform, [37], has proven to be a very robust and powerful feature descriptor for object matching in images [41]. This descriptor utilizes a 16x16 patch of pixels around each feature point. Within this 16x16 patch, the pixels are subdivided into 4x4 pixel sub-patches. Each pixel's gradient magnitude and angle is calculated using simple gradient filters. The magnitudes

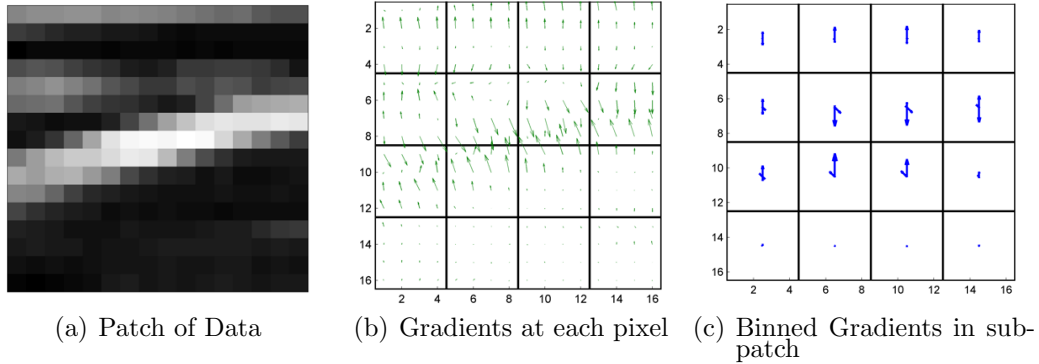


FIGURE 3.5: Example generation of SIFT descriptors on a patch of GPR data. The patch of data taken from around a feature point location is shown in 3.5(a). The region is partitioned into the 4x4 grid of sub-patches. Plot 3.5(b) shows a quiver plot representing the gradients at each pixel location. Plot 3.5(c) shows the result after binning the gradients within each sub-patch. Each arrow in this plot originating from the center of the sub-patch represents the magnitude of the bin in that angular direction.

are weighted according to their distance from the feature point using a Gaussian mask. The pixel's magnitudes are then binned within each sub-patch according to their calculated angle's distance to 8 equiangular lines between 0° and 360° . Each sub-patch generates an 8 dimensional vector where each value is the sum of magnitudes from pixels that shared a similar gradient angle. The vectors from all sub-patches are concatenated to create a 128 dimensional vector representing the SIFT descriptor for one feature point. In addition, [37] found that a particular normalization method was most effective to maximize the SIFT descriptors matching abilities. Each 128 dimensional SIFT descriptor is converted to a unit vector. All values over 0.2 are then set to 0.2 and the vector is again converted to a unit vector. This was found to provide invariance to intensity differences between images as the same edges can have vastly different gradient strengths between images depending on lighting or camera factors [37]. Figure 3.5 shows the transformation of a data patch into a set of binned gradient vectors.

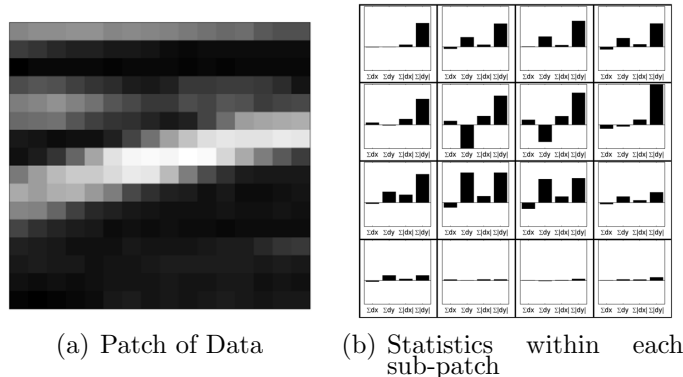


FIGURE 3.6: Example generation of a SURF descriptor on a patch of data. Plot 3.6(a) shows the patch of data around a feature point. The statistics of each sub-patch are shown in 3.6(b). The statistics of all 16 sub-patches will be concatenated to form a 64 dimensional feature vector.

Descriptor: SURF The descriptor used in SURF was intended to be a faster alternative to the SIFT descriptor with no loss in performance [38]. The SURF descriptor implementation found that a 20x20 patch of pixels around each feature point performed best for matching. Similar to the SIFT descriptor, SURF descriptors subdivide the patch of data and aggregate within each sub-patch. The SURF implementation subdivides each patch into 16 - 5x5 pixel sub-patch regions. Within each sub-patch x and y-direction gradient information is calculated using two simple Haar wavelet filters. The gradients are aggregated to obtain 4 statistics that represent the sub-patch. The four gradient statistics are: $\sum dx$, $\sum dy$, $\sum |dx|$, $\sum |dy|$. These statistics were chosen as they are very fast to calculate and provided comparable performance to the SIFT descriptors. An example descriptor calculation is shown in Figure 3.6. This calculation of a descriptor is fast as a result of using simple box filters that can be calculated quickly using an integral image [38]. Normalization for the SURF detectors simply consists of converting the 64 dimensional vector to a unit vector. As with the SIFT descriptor this is to provide invariance to changes in intensity between images.

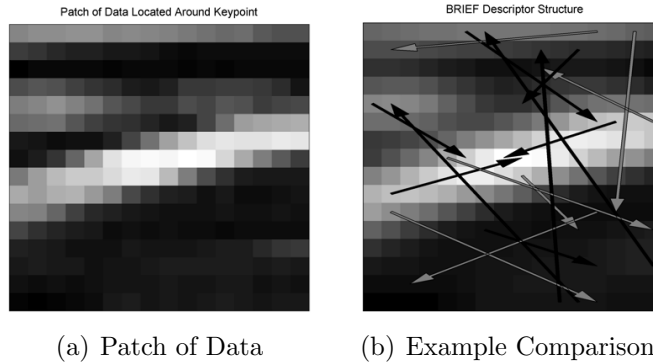


FIGURE 3.7: The original patch of GPR data to be processed is shown in 3.7(a). The same data is smoothed and shown in 3.7(b). A small number of comparison features are also displayed. The gray arrows represent location pairs where the first location has lower intensity than the second. The black arrows represent pairs where the first location has higher intensity.

Descriptor: BRIEF The BRIEF descriptor was selected in this research for its invariance to pixel intensity changes as well as its alternative method to account for feature point localization inconsistencies [40]. Unlike previous descriptors introduced, the BRIEF feature is not calculated based upon the gradients within the patch. Instead BRIEF calculates comparisons between locations within the smoothed patch around a feature point. These comparisons, when dense enough, represent the relative intensity changes within a patch, thus representing structure within the patch. As the only information retained from each comparison is the location that has a higher intensity, no knowledge of absolute pixel intensity is incorporated. By smoothing the patch with a Gaussian filter prior to any comparisons, smaller differences in localization are not as detrimental to performance. The locations utilized in BRIEF descriptors are chosen at random with replacement and kept throughout training and testing. Figure 3.7 shows an example of how a small number of brief features would be generated within a patch. In this experiment, 256 different comparisons were used per patch. Matching between patches is performed with a simple Hamming distance, the number of comparisons that do not match between the two binary vectors. In

the binary case, the matching between two vectors is then simply an XOR operation that can be optimized for very high speeds [40]. The resulting descriptor for each patch is a 256 dimensional binary vector.

Object Detection

In contrast to instance matching where an identical version of an object is identified, object detection is designed to find instances of a class of objects within images. An example of this relating to books would be trying to find books in a scene with no constraints on which specific book is to be found. Common examples of applications of this technology are face detection [20], pedestrian detection [43], and car detection [44]. For these types of detection problems, the challenges can be both the image transformation challenges seen in instance matching, as well as the uncertainty associated with interclass variations. In the example of pedestrian detection, interclass variations include the issues such as the size/weight of the person, the colors and patterns of their clothes, their poses, and all other differences that distinguish one human from another. Due to the amount of variation possible, most algorithms designed for this task tend to require more training data to effectively model the interclass differences. Classifier choice is also important in this application as over-fitting can significantly reduce classifier performance.

Training data for object detection generally consists of a large number of example images of the desired object as well a sample of background images. Naturally, the dataset used for training will also contain some portion of the expected statistical inter-class variations typical of the problem under consideration. Instead of using a patch based approach, object detection generally uses a patch size that encompasses the entire object that is being detected. While a patch based method could work well for objects that have similar small scale characteristics, generally within a class, these small-scale characteristics vary too much to provide reliable detection. However, as in

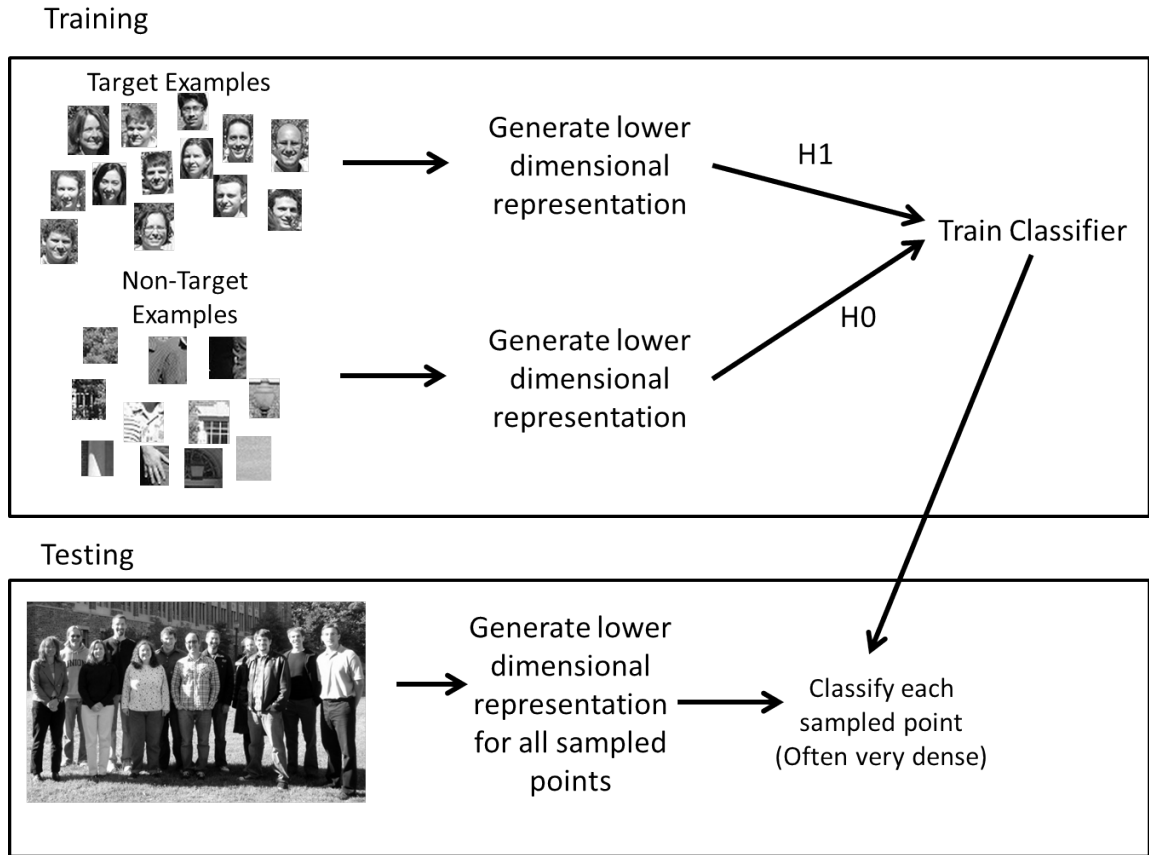


FIGURE 3.8: Example flow of the object detection method applied to face detection in images. Example training samples are generally obtained via hand labeling. The classifier is trained on the descriptor representation of all example images. Locations within new images are converted to their descriptor representation and classified using the trained classifier.

instance matching, descriptors are used to represent the data for the same advantages of localization and intensity differences.

The classifier for object detection is trained using the descriptor representation of all training example images. Detection of objects in images is then performed by computing the descriptor and classifying each location across an entire image at a sampling rate that is acceptable given the descriptor aggregation size and expected object size. These steps are illustrated in Figure 3.8 which outlines the steps that object detection generally follows.

Most of the methods utilized in the implementation of object detection are very similar to that of instance matching. Descriptor methods from both fields are generally inter-changeable; however the size of the patch is usually optimized for specific applications. In addition to the descriptors used previously, this research also utilizes a descriptor originally applied to pedestrian detection - the Histogram of Oriented Gradients (HOG) descriptor [43]. As classification takes a larger role in object matching, this research also evaluates three different types of classifiers. The first is a linear classifier that also performs dimensionality reduction known as Partial Least Squares Discriminant Analysis (PLS-DA). The Support Vector Machine (SVM) was studied because of its ability to create a decision boundary that maximizes the margins between classes [45]. Finally the Random Forest (RF) classifier is used given its performance in recent comparisons between classifiers [46].

Descriptor: HOG The Histogram of Oriented Gradients (HOG) shares many similarities with both the SIFT and SURF descriptors. As the HOG descriptor was originally developed for pedestrian detection, it was designed to detect objects with the same orientation. Like the SIFT descriptor, the HOG descriptor takes the patch of data around a given area and subdivides it into sub-patches where gradients are binned according to orientation. However, instead of using 8 equiangular bins from 0-360°, the HOG descriptor as presented in Dalal and Triggs [43], uses a number of bins between 0-180°. This difference means that while SIFT will identify a difference between the two images presented in Figure 3.9, the HOG descriptor will be the same for both.

Normalization of the HOG descriptor is necessary to account for intensity changes for the same reasons as the other gradient based descriptors used in instance matching. HOG mainly differs from other approaches in that it does not have a set number of sub-patches within which normalization occurs. Instead the number of sub-patches

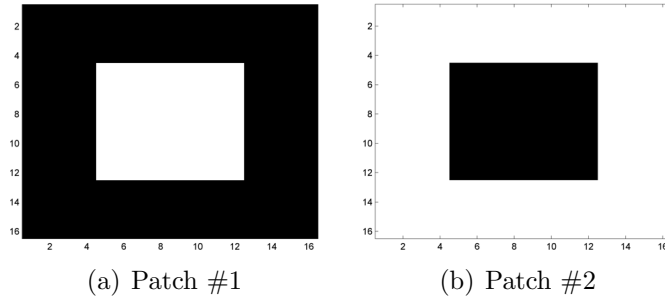


FIGURE 3.9: Patches #1 and 2 both exhibit the same geometric structure. However the gradient directions along each edge are reversed. The HOG descriptor will bin the gradients in the 90° and 270° in the same bin. The SIFT descriptor will bin the gradients in separate bins as the range for the sift angular bins goes from 0 - 360° .

included in normalization is referred to as a block size, and can be adjusted depending on the application. The original implementation of HOG found that between the methods of normalization they tried, the method of normalization generally made little difference. This research utilizes the L2-norm method, where the HOG vector for the appropriate blocks, \mathbf{v} , is normalized by

$$\mathbf{v}_{norm} = \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}} \quad (3.2)$$

Where ϵ is a small non-zero number that tends to have little effect on the descriptor. The resulting normalized HOG sub-patches are then concatenated to form the descriptor representing the area desired.

Classifier: PLS-DA The partial least-squares discriminant analysis (PLS-DA) classifier is used in this research because of its ability to perform dimensionality reduction to a space where all features are uncorrelated [47]. While this type of dimensionality reduction is often carried out using an unsupervised method like principal component analysis (PCA), PLS is advantageous because it also leverages the class labels. This is especially important if the between-class separation is not as dominant as the interclass variation [48]. Consider a data matrix \mathbf{X} of dimensions $n \times m$, contains n

observations, each with m features. For the i^{th} observation, there is a corresponding class label, y_i . The goal of these dimensionality reducing methods is to project \mathbf{X} onto a new subspace that maintains the desired information about \mathbf{X} while reducing the dimensionality. This can be modeled using

$$\mathbf{T} = \mathbf{X}\mathbf{W} \tag{3.3}$$

where \mathbf{T} is the lower dimensionality representation of \mathbf{X} , and \mathbf{W} is the transformation matrix that projects \mathbf{X} into a lower dimensional space. In this model, our class labels, \mathbf{y} are a linear function of \mathbf{U} . \mathbf{C} dictates the relationship between \mathbf{U} and \mathbf{y} while \mathbf{G} represents the error in the model.

$$\mathbf{y} = \mathbf{U}\mathbf{C}^T + \mathbf{G} \tag{3.4}$$

When training a classifier for either model, it is reasonable to assume \mathbf{y} is dependent on our representation of the data, \mathbf{T} . To learn the relation between the representation of the data and the labels, \mathbf{C} must be calculated. This is performed using least squares regression where $\|\mathbf{G}\|^2$ is minimized. The solution for \mathbf{C} is then

$$\mathbf{C} = (\mathbf{T}\mathbf{T}^T)^{-1}\mathbf{T}\mathbf{y} \tag{3.5}$$

Given projected data, \mathbf{T} , both PCA regression and PLS regression utilize the same method for training the model. The difference however is in the forming of the transformation matrix, \mathbf{W} . PCA involves maximizing the variance in the final projected data with no regard to class labels. Therefore the orthogonal directions PCA selects to project the original data, are the directions with the highest variance within the observation data. In PLS however, the goal is to find the direction vectors that maximize the covariance between observations, \mathbf{X} and labels \mathbf{y} . There are a number of different algorithms designed to complete this task. The method used in this research is known as SIMPLS [49]. SIMPLS leverages the knowledge that

the direction that maximizes covariance between \mathbf{X} and \mathbf{y} is the first left singular vector of $\mathbf{X}\mathbf{y}$. This method is iterative in that it finds a single direction vector per iteration. The residual information of \mathbf{X} and \mathbf{y} are calculated after each iteration and used as the input for the successive iterations. When the weights are generated, they are applied to \mathbf{X} to obtain a \mathbf{T} for the training data. Least squares is then used to obtain a value for the linear model variable \mathbf{C} that explains the relationship between \mathbf{T} and \mathbf{y} .

New test data that is received for classification is first reduced using the weights found during training, \mathbf{W} . The lower dimensional representation is then multiplied by \mathbf{C} to obtain estimated labels for the new data. These estimated labels are the confidences output by the PLS-DA classifier.

Classifier: SVM The Support Vector Machine (SVM) is a statistical classifier developed to maximize the margin between the decision boundary and the two classes [50]. This decision boundary is referred to as the maximum-margin hyperplane. In a linearly separable dataset, this decision boundary can be found heuristically by expanding the width of the decision boundary line until it contacts data points from either class. When the decision boundary can no longer expand, or rotate to expand further, it has achieved the maximum margin. An example of this is illustrated in Figure 3.10. The label, y_i , of a test data point, \mathbf{x}_i , is determined by which side of the decision boundary it lies in. This is performed using a linear model where

$$\begin{aligned} y_i &= 1 \text{ if } \mathbf{w}^T \mathbf{x} + b \geq 1 \\ y_i &= -1 \text{ if } \mathbf{w}^T \mathbf{x} + b \leq -1 \end{aligned} \tag{3.6}$$

In this model, \mathbf{w} is a vector whose direction is perpendicular to the decision boundary line. The data points are projected onto the \mathbf{w} subspace, then the decision is determined by this projection and the offset, b . In training the linear SVM classifier, finding the best decision boundary involves maximizing the margin until it reaches

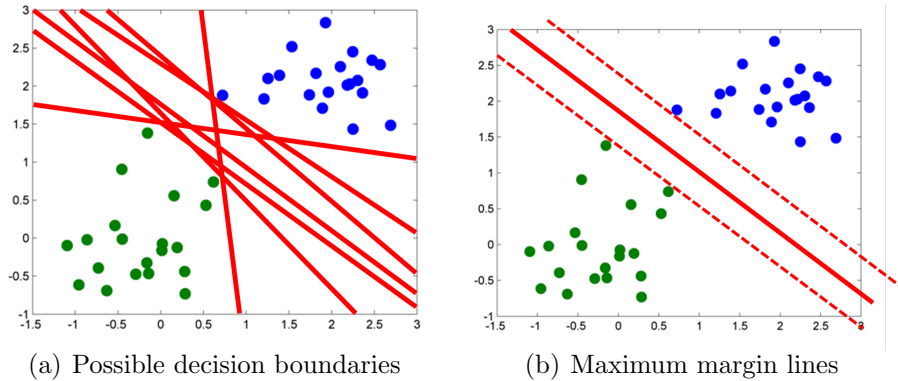


FIGURE 3.10: Figure 3.10(a) shows possible decision lines that separate the two shown classes without error. Many of these lines however, are not desirable for classifying new unseen testing data points. The solid line shown in Figure 3.10(b) is the desired robust decision boundary with the maximum margin and is better suited to classify unseen data given these distributions.

the nearest data points. Training of a linear SVM with completely separable data is then an optimization problem as the width of the margin can be expressed in terms of \mathbf{w} . The minimization required for a linearly separable dataset is performed by minimizing the following:

$$\operatorname{argmin}_{\mathbf{w}, b} \mathbf{w}^T \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^T \mathbf{x}_i + b \geq y_i \quad \forall i = 1, 2, \dots, n. \quad (3.7)$$

This optimization is possible by introducing Lagrange multipliers that reduces to optimizing the following for the Lagrange multipliers,

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \mathbf{w}^T \mathbf{w}, \quad \text{subject to} \quad \alpha_i \geq 0 \quad \forall i, \quad (3.8)$$

where because of the α_i terms, \mathbf{w} can be calculated as

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i. \quad (3.9)$$

Constraints on this minimization ensure that every point is classified correctly. However, this minimization problem fails if the data is not separable which is often the

case with noisy observations. Therefore a new term, C , that penalizes the misclassification of a training point is introduced. The minimization problem becomes

$$\operatorname{argmin}_{\mathbf{w}, b} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \epsilon_i \quad \text{s.t.} \quad y_i(b\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0 \quad \forall \mathbf{x}_i. \quad (3.10)$$

This allows for classification errors when $\epsilon_i \geq 0$ to have an effect on the minimization. The constant, C , weights how much a misclassification affects the decision boundary. The higher the value of C , the more important it is that training points lie on the side of the decision boundary that their label corresponds to.

This research utilizes a kernelized version of the SVM [51] implemented in MATLAB by [52] with a radial basis kernel. By performing the SVM learning and classification using a kernel function that maps into a higher dimensional space, the relationship between observations can be non-linear allowing for more complex relationships to be realized [53].

Classifier: Random Forest The Random Forest (RF) classifier differs with respect to other classifiers in that it does not attempt to classify the observation using its entire set of features at once [54]. The approach often taken to classify using all features generally leads to complex classifiers and decision boundaries giving little intuition of how changes affect classification. Random forests operate using an ensemble classification method, where the decision is based on the mode of a large number of simple classifiers. The simple classifiers are decision trees and the whole system of trees is appropriately known as a forest.

For each decision tree, a subset of observations is randomly selected with replacement in a process referred to as bootstrapping. The resulting data is used as the training data for the tree, while the remaining data is used to test the error of the decision tree. For each decision node, a small subset of features is selected and used

to calculate the split between classes. This is repeated for all nodes in the tree. The resulting decision tree is one of many used in the ensemble.

To classify a new sample, the observation is processed by all decision trees that have been trained on different bootstrap samples of the data. In a binary classification setting, the number of individual trees that classified the observation as a positive instance is used to assign a confidence to the observation.

Image Classification

Image classification tasks require the algorithm to classify entire images into a small number of categories. An example of this would be the categorization of pictures from the inside a library (containing books) from a school's image database. The kinds of categories can range from city vs. suburban images, to what kind of animal is present in the image. In general this term represents problems where the algorithm classifies based upon the entire image. One of the main challenges is that parts of the image consist of background data that is not indicative of the class label. However due to the variety of possible realizations of each class of image, there is no way to distinguish easily between indicative and non-indicative patches within an image. Therefore a method that takes into account this multiple instance problem must be employed.

The algorithm that this research utilizes is a categorization method known as the bag-of-words approach. This approach was selected due to its simplicity and ability to process the entire image, where many part based models [55] focus on particular areas and are more akin to object detection. The bag-of-words method is based on a method originally developed for text categorization. Instead of words however, images are classified based upon patches that are sampled densely across the entire image. In order use this model with images, a discreet set of possible types of patches, analogous to a vocabulary, must be developed. This visual vocabulary is created by

randomly sampling a large number of patches from the training data and clustering them based on their descriptor representation. This research utilizes K-means as it has shown to cluster image patches reliably and performs well on recognition tasks [56]. Patches in an image are selected using a variety of methods. These patches can be extracted densely at given intervals, or at certain interest points such as corner and blob locations [33]. Each extracted patch in an image is then represented by the cluster that its descriptor representation is most similar to. The entire image is in turn represented by the frequencies of these clusters being present. These frequency histograms are then utilized for training a classifier using both positive and negative examples. Similarly to object detection, the requirement for large amounts of training data exists in order to cover the statistical variation possible within the image categories. The final confidence of each image is the classifier output given the frequency histogram vector. The steps involved in the bag-of-words model can be seen in Figure 3.11.

Many different methods of patch representation, patch localization, and patch size parameters can be used in this method. This research evaluates the use of two descriptors, a modified HOG descriptor and the raw pixel values that have been normalized and whitened. The modified HOG descriptor is the same as normal HOG descriptors, except the block size is changed in order to match the sub-patch size of a SIFT descriptor given the patch size. Therefore the HOG descriptor block size will be such that a grid of 4x4 blocks will fit into each image patch.

Whitened Patches Utilizing normalized patches for clustering to determine a visual vocabulary has been found to perform poorly in image recognition tasks [56]. By whitening in addition to normalization as a pre-processing step, improvements were found in the ability to cluster the patches using K-means into centroids that provided better results in recognition tasks [57]. By whitening the patches, the correlation

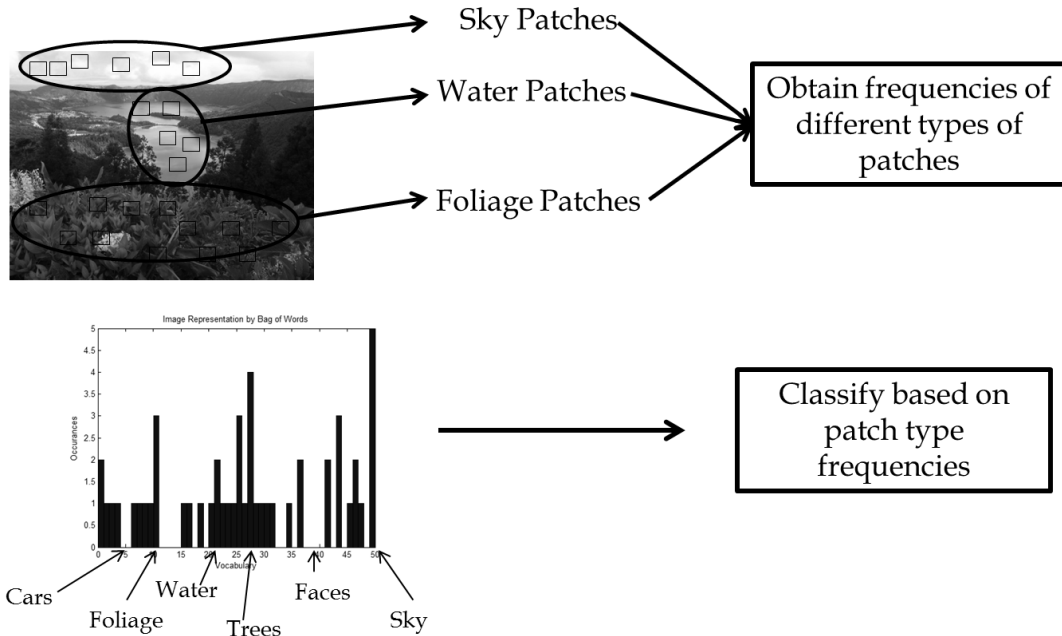


FIGURE 3.11: Processing steps for the Bag-of-Words model applied to visual images. The patch vocabulary is generated based on a random sample of patches within the training data (top). Each image is in turn represented by the frequencies of each patch type (bottom). These frequency histograms are then used to represent the image for both training and testing stages.

between pixels is removed. This allows the whitened patch centroids to be more evenly distributed across the data sub-space [57]. The whitening process used in this research is based on the method used in [57]. The training data, \mathbf{X} is a $n \times m$ matrix where each row is a random patch take from the training set of B-scans. The data is first normalized so that each row has a mean of 0 and a standard deviation of 1.

Given the decomposition of the covariance of the data matrix, $\mathbf{VDV}^T = \text{cov}(\mathbf{X})$, the whitened data is computed as

$$\mathbf{X}_{white} = \mathbf{V}(\mathbf{D} + \epsilon\mathbf{I})^{-\frac{1}{2}}\mathbf{V}^T\mathbf{X} \quad (3.11)$$

where ϵ is a regularization term to ensure that the eigenvalues, \mathbf{D} are not too close to 0. The \mathbf{V} and \mathbf{D} matrices used in to whiten the training data prior to clustering are preserved to whiten the testing data in the same manner.

3.1.2 Results

This section presents the result of testing these methods on data collected over inert targets buried at known locations marked by Global Positioning Satellite (GPS) locations. Data was collected by driving the data collection system over the targets at operational speeds multiple times from different directions. The dataset used for this chapter contains a total of 788 target pre-screener alarms and 1635 pre-screener alarms that do not contain targets. Algorithms are tested using 10 fold cross validation where the classifier training is performed on 9/10ths of the data in each fold. As multiple data collections were obtained over each target, the folds were generated so that all instances of the same target at one location are contained in one fold. This ensures that the detection methods are tested on their ability to generalize detection to other soil conditions rather than simply overcoming noise introduced by the sensing modality.

Algorithm performance is presented utilizing a variant of the receiver operator characteristic (ROC) curve. Instead of viewing detection performance in terms of the probability of false alarm (FA), this research instead utilizes the false alarm rate (FAR) with units of FA/m². This metric is used to give perspective on the overall operating characteristics of the system in real world situations. The probability of detection (PD) at set values of FAR will allow the comparison between algorithms at operating conditions acceptable to operators. The FAR levels that will be presented in this chapter are at the rates 0.001 FA/m², 0.003 FA/m², and 0.005 FA/m². Given the width of the GPR panel, these values correspond to a false alarm encounter on average every 333 m, 111 m, and 66 m traveled by the vehicle.

Instance Matching

This section presents the results obtained when utilizing instance matching as a framework for detecting targets. Typically instance matching is performed by com-

paring one example image to cluttered scenes and finding matches within the scenes. In this research however, instead of one example, this application has hundreds of training target examples to compare to. Some targets will appear more similar than others due to both target type and the surrounding soils. It is for this reason that for each test pre-screener alarm, the GPR data is compared to the complete set of pre-screener alarms in the training data. The top 15 matches, determine the confidence by using a k-nearest neighbor like approach of counting all the positive instances within the top 15 matches.

Performance as a Function of Various Descriptors Results associated with the different descriptors tested are shown below in Figure 3.12 at the FAR levels specified. These results are shown with the top performing patch sizes utilizing 2-D energy points as the feature point method. This figure shows the performance both with and without fusion with the pre-screener confidence. SIFT provided the best performance without the fusion with the pre-screener. When the pre-screener confidence is fused with these results, the performance of all methods increased. In particular, BRIEF features benefited significantly from fusion with the pre-screener. As the BRIEF feature is simply represented as a binary comparison between pixels, it has absolutely no information regarding the absolute energy in the image. Therefore with the additional energy information, performance is much improved. In comparison, the normalized SIFT descriptor is still be affected by local changes in signal energy; e.g., a number of bins with very large values, while normalized, still indicates the presence of large gradients as other gradient directions will be significantly smaller if no large edge exists. Therefore, the energy information from the pre-screener confidence is not as much of a performance boost compared to BRIEF.

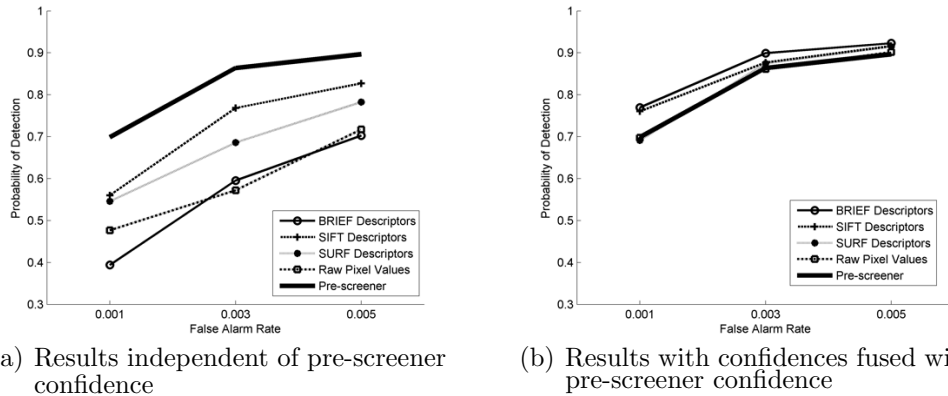


FIGURE 3.12: The results above illustrate the performance of various descriptor methods over the best performing parameters utilized in instance matching. The results on the left show the performance of the descriptors without fusing with the pre-screener confidence. The results on the right include the pre-screener fusion that shows additional information is being introduced by the fusion with instance matching. These results show that without the pre-screener information, the SIFT descriptor method outperforms other descriptors. In contrast, BRIEF features, perform very badly without pre-screener information perform much better after fusion with the pre-screener.

Performance Across Feature Point Methods Results presented in Figure 3.13 show the performance as a function of the various feature point detection methods. As in the previous section, while large differences are evident without fusion with the pre-screener, when the confidences are fused, performance differences are minimal. The performance of the point methods without fusion show that the blob detector and energy peak detectors perform consistently better than the Harris corner point method. However, when the algorithms are fused, no method is clearly superior to others.

Analysis of Matches Figure 3.14 provides an example of a very good match between two targets of the same type in different ground locations. While some points are not matched perfectly across these two B-scans, this example shows good feature point localization and most points match to similar locations within both responses.

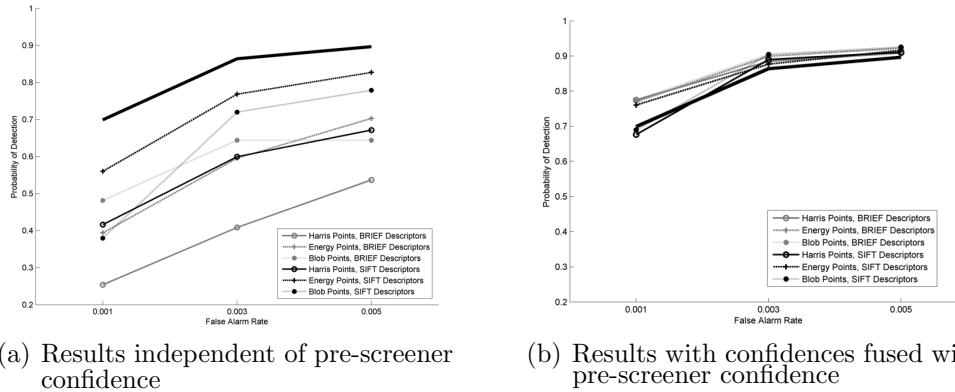


FIGURE 3.13: The results above evaluate the performance of various point localization methods utilizing the best performing descriptors. The results on the left show the performance without fusion with the pre-screener confidence. The results on the right include the pre-screener fusion that shows additional information is being introduced by utilizing the instance matching confidences. These figures show the large disparity between performance with and without the fusion with the pre-screener, especially in relation to the BRIEF descriptors.

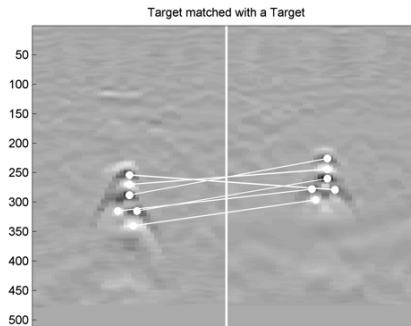
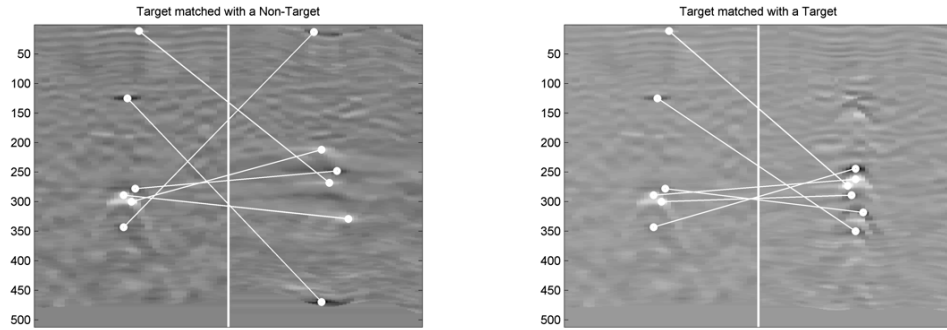


FIGURE 3.14: This figure illustrates matching between two different instances of the same target type. The target on the right is the closest match over all training alarms available. The lines represent a match was found between the given SIFT descriptor whose patch center is represented by the white dots.

In contrast, Figure 3.15(a) shows a match utilizing the same parameters where matching fails to occur between alarms of the same type. The target on the left matches most closely with the false alarm on the right. To understand why this alarm did not match with a training example of the same target type, Figure 3.15(b) shows the points matched between the two targets of the same type but different

ground locations.



(a) Target with top match, a false alarm.

(b) Target match between target of same type and depth

FIGURE 3.15: This figure shows the down-track matching between a target and its top match, a false alarm. The second matching image in 3.15(b) shows the feature point matching between the same target and an instance of the same target type at the same depth at a different location. The feature points are not located similarly and the overall responses do not share all of the same visual characteristics. Thus the two B-scans were not matched well during testing

The responses in Figure 3.15(b) look significantly different due to the soil properties around the alarm. This difference in appearance causes the points to be localized in different areas of the responses. As the points are located in different areas of the alarm response, the patch data surrounding these points will not be similar, causing large distances between descriptors. This makes it difficult to match whole target responses between B-scans. Finding multiple alarms with both similar keypoint localization and the area around those keypoints is very difficult given the variation seen in GPR responses. It would therefore require a large set of nearly identical training examples in order to correctly classify targets (ie; same subsurface characteristics). This is not possible given that subsurface characteristics are generally unknown in operating conditions, and impossible to predict fully.

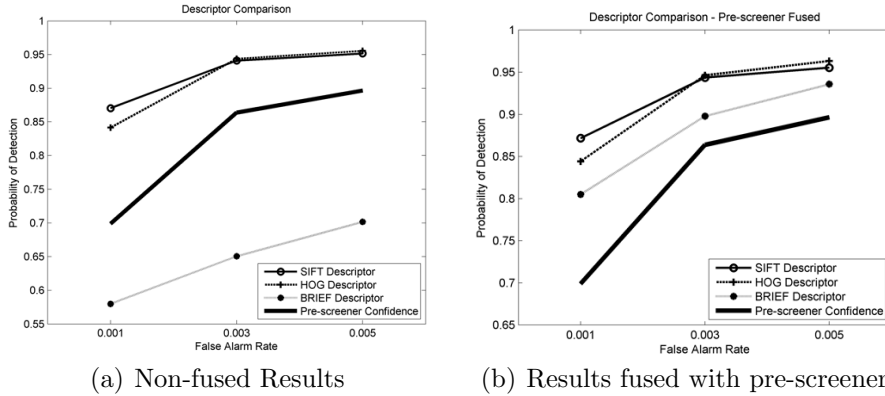


FIGURE 3.16: These results illustrate the differences in performance between the various descriptor methods. The best performing patch size is used in conjunction with the SVM classifier. The SIFT descriptor consistently performed the best among all methods tested despite the similarities to the HOG descriptor. The BRIEF descriptor performed poorly in both trials, especially when not fused with the pre-screener.

Object Detection

This section presents the results from applying an image object detection framework to the detection of buried targets in GPR data. The detector is trained using 2-D GPR image patches from pre-screener alarms both with and without targets responses present. In the case of positive examples, these patches are centered around the target response. In this chapter the target response image patch is chosen using hand labeled locations as is often done in visual image tasks.

Evaluation of Descriptor Methods A number of descriptor representations were tested in this work from both instance matching, and object detection algorithms. The results comparing the descriptors under consideration are shown in Figure 3.16. These results show very good performance in object matching methods in both the SIFT and HOG descriptors. These results also show that BRIEF features continue to perform poorly without fusion with the pre-screener. Unlike in previous methods however, after fusion, BRIEF features are not able to match the performance of the SIFT and HOG descriptors.

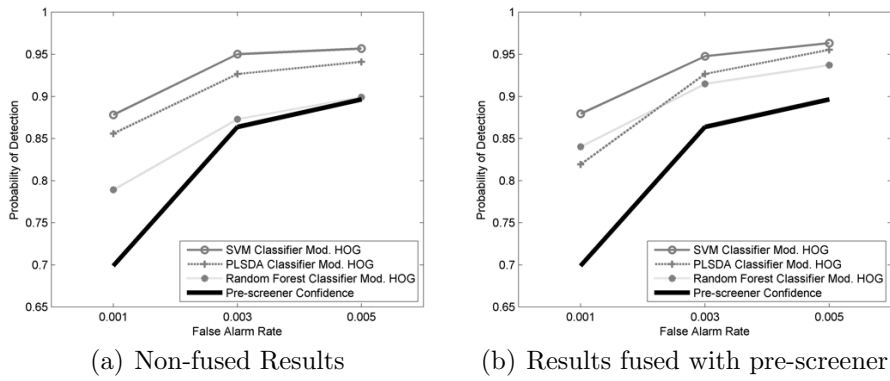


FIGURE 3.17: This figure shows the comparison between different classifiers utilizing the HOG descriptor results as shown above. The SVM classifier performed the best on all trials.

These results indicate that the more general method of object detection performs much better on GPR data than attempting to match specific alarms. This is likely due to the fact that this method primarily focuses on the overall shape of the object by utilizing a statistical classifier that can identify properties of target vs. non-target image patches across a wide variety of soil conditions and target types.

Evaluation of Classifiers Figure 3.17 provides the performance across the various classifiers tested in this research. Utilizing the SVM classifier provided the best results across all trials. The random forest classifier, however, was the only classifier that benefited from fusion with the pre-screener. This was likely due to the poorer performance of the random forest compared to the other methods. Both the SVM and PLSDA classifier had the same or worse performance when fused with the pre-screener.

Automated Training Data Labels Previously, experiments were performed using training data that was extracted based on a hand labeled row index where the first occurrence of a target response could be observed. This section provides results and analysis utilizing row indexes that were determined automatically to avoid the vari-

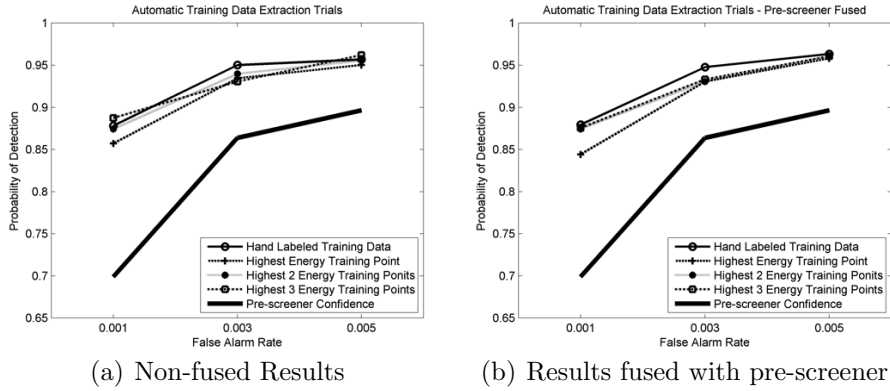


FIGURE 3.18: This figure shows the results of attempts to automatically generate isolated training examples from the training data instead of using hand labeled examples. It can be observed that the hand labeled training data continues to perform better at certain operating points, most notably at 0.003 FA/m². In the fused results, the hand-labeled data would be the clear choice. However the method utilizing three different training examples per training image does have better performance at some operating points in the un-fused results. It is important to note that this is not necessarily a fair comparison. By having 3 training samples per positive image, there are three times the number of positive training samples.

ance in human labeling. To determine the depth of the response, this method uses the smoothed energy of the A-scan at the pre-screener alarm location. As a high energy reflected signal after the initial ground bounce typically indicates a change in subsurface electromagnetic properties, it is assumed that large reflections are caused by the buried target. We can therefore attempt to identify the row at which the target response is present by the largest peak after the ground bounce.

The results shown in Figure 3.18 evaluate the effectiveness of the automatic training data extraction utilizing between one and three of the top energy peaks from each positive training example. These results indicate that utilizing automated training methods perform comparably to hand labeling positive examples. In particular, utilizing 3 training patches from each pre-screener alarm produces very similar performance to using hand labeled training data.

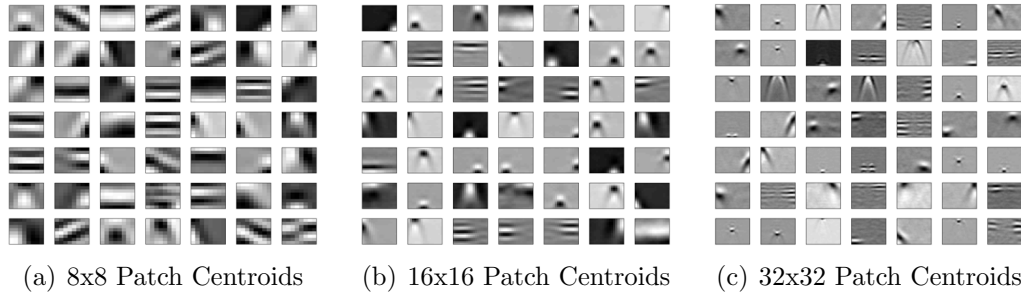


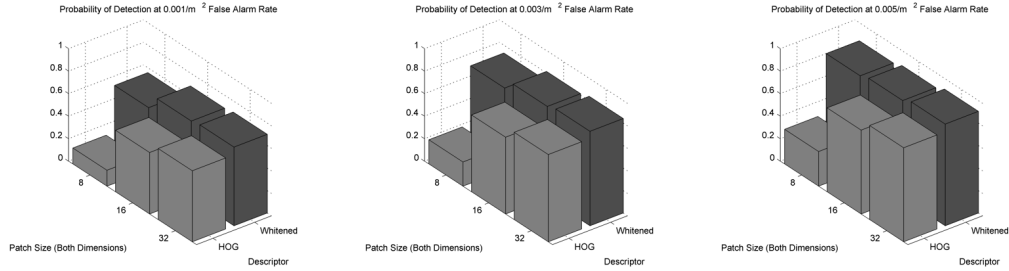
FIGURE 3.19: The figures above illustrate the differences in centroids for different sized patches. The smaller patch centroids can only represent parts of the landmine response. The larger patch centroids are able to represent both the rising and falling edge of the landmine response allowing it to represent the entire landmine structure.

Image Classification

When utilizing the bag-of-words model the first necessary step is to create a visual vocabulary. This is done by randomly sampling patches from 2-D B-scans in the training data. The raw patches are then processed either into a HOG descriptor or normalized and whitened before performing k-means. An example selection of 2-D GPR vocabulary words from a variety of patch sizes is shown in Figure 3.19. These examples are of the centroids found using whitened patches. This figure illustrates the kinds of shapes and structures that are represented by each patch size.

The first series of results shown in Figure 3.20 considers the use of the two methods of patch representation using the bag-of-words model. The first parameter evaluated is a range of patch sizes. The results presented are independent of the pre-screener fusion. This is due to the poor performance of the HOG descriptors causing the pre-screener confidence to drive ROC performance.

These results are surprising given that in previous applications of a pixel-based detector, performance has been poor. Poor performance in other methods is generally attributed to issues with localization, which if off by a single pixel can cause large differences in the resulting vector. In this application however, a representation is



(a) Probability of detection at a FAR of $0.001/\text{m}^2$ across patch sizes and descriptors
 (b) Probability of detection at a FAR of $0.003/\text{m}^2$ across patch sizes and descriptors
 (c) Probability of detection at a FAR of $0.005/\text{m}^2$ across patch sizes and descriptors

FIGURE 3.20: This figure shows the performance of the descriptors over a range of square patch sizes independent of the pre-screener confidence. Performance is measured by the probability of detection at a given false alarm rate (FAR). Each of the images above represent the performance at a given FAR value. The darker bars within each image represent the normalized patch’s performance as a descriptor. The whitened patch performance was found to be more consistent across various patch sizes than the HOG method.

learned that can encompass small shifts in location.

To better understand the performance differences between methods and patch sizes, a method of visualizing the performance across a test sample was created. Figures 3.21 and 3.22 illustrate a method of visualizing the individual patch scores. The dots in each image represent the center of an image region extracted. The intensity of the dot corresponds to the centroid that represents the region in the classification of the image. The dot intensities are determined by ordering centroids, c_i by the ratio:

$$\frac{p(c_i|H_1)}{p(c_i|H_0)}, \quad (3.12)$$

so that light colored dots correspond to centroids that have a higher likelihood ratio score.

The images shown in Figure 3.21 support a hypothesis that the whitened patches react to areas of higher energy. In both methods, areas of high energy have lighter colored dots representing centroids that appear often on target responses. In the area

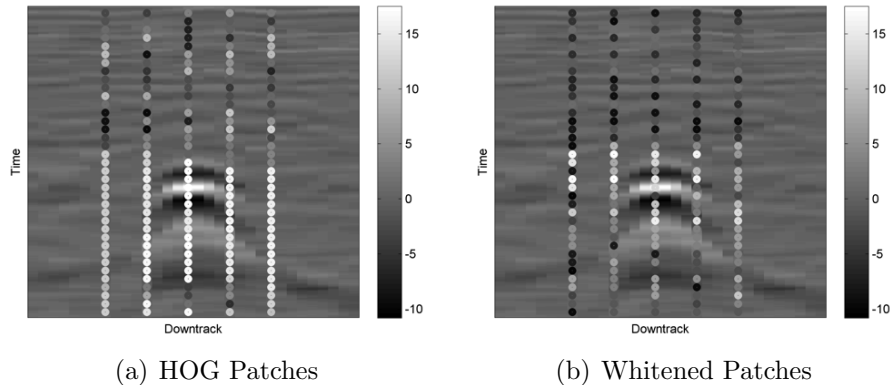


FIGURE 3.21: B-scans of the same target visualizing the likelihood ratios of the centroids representing each region. Each dot’s intensity is determined by the rank of the centroids likelihood ratio. The centroids when using HOG descriptors as shown in (a) is much more consistent than the whitened patches shown in (b). These results were obtained using a 16×16 patch size, with a sampling rate of 4 in both dimensions.

below the high energy however, the dots from the whitened patches method are no longer indicative of target responses. This is likely due to the lower energy levels of the late time reflections. The HOG descriptor however, still identifies the low energy areas as target like response patches. This illustrates the HOG descriptor’s ability to observe structure in low energy data.

In addition to the reduced energy dependence, the HOG descriptor is more consistent within the oscillations of the hyperbola. This is because the HOG descriptor angles are binned by their angle between 0° and 180° . This means that a center point located on a negative region surrounded by positive regions will look the same as a positive region surrounded by negative regions. This is not the case for whitened images because large, positive magnitude signal data will correlate poorly with large, negative magnitude signal data. Both this and the fact that HOG performs aggregation over sub-patches is why HOG descriptor centers remain more consistent within a target response’s ringing effect as shown in Figure 3.21.

While the energy invariance of HOG can be beneficial in extracting information complementary to the pre-screener, it can present issues in low energy GPR data.

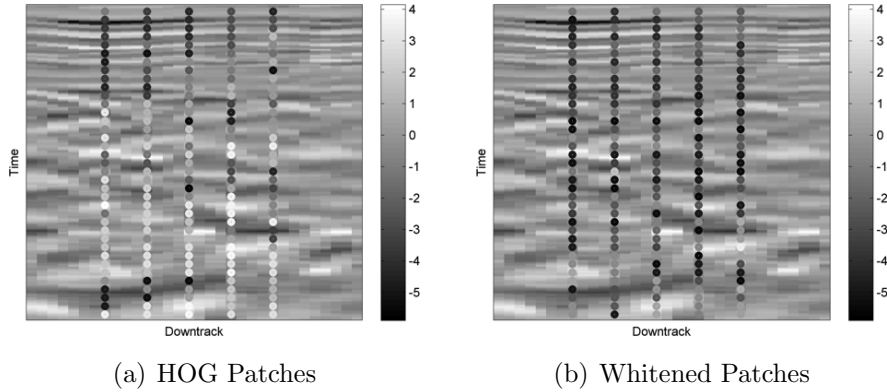


FIGURE 3.22: This figure helps visualize the likelihood ratios of centroids that represent regions of a false alarm B-scan. Note that the energy range in these images is much smaller than the images of a buried target. The HOG patches in (a) have a much higher response to these areas where the energy is low, but structure in the data recognized by HOG remains. In the whitened patches shown in (b), the likelihood is much less consistent.

Figure 3.22 provides an example where the performance obtained utilizing the HOG descriptors is poor without the pre-screener fusion. Many of the dots in the false alarm shown in Figure 3.22(a) are represented by centroids that appear often in target responses. This is due to the blob like structure observed in the B-scan data which, if viewed locally, can appear as a part of a reflection. While performance without fusion with the pre-screener is poor, with the addition of the energy information, false alarms such as these have much lower confidences.

The images shown in Figure 3.22 also provide insight into why performance with larger patches is generally better. This is particularly evident in the HOG descriptor image in Figure 3.22(a). This particular false alarm was shown because it contains significant amounts of texture where the image regions are represented by centroids commonly found on target responses. Considering Figure 3.19 which displays centroids from different sized patches, it is clear that the smaller sized patches are often looking simply for edge like structure. The larger patch sizes contain much more complete target response structure. In Figure 3.22(a), the smaller patch sizes are

identifying the edges found within the correlated noise while those same edges are also found on target responses. This causes the classifier, which is only given centroid frequencies, to calculate a much higher confidence for this false alarm.

3.1.3 Conclusions

The research presented in this chapter evaluates the use of three different detection frameworks applied to GPR data. These frameworks, originally designed for 2-D visual images, were able to extract useful information that improved the classification of pre-screener alarms. The top ROC operating points from each method tested are displayed in Figure 3.23. This figure indicates that the performance of an object detection framework is the best among all methods considered.

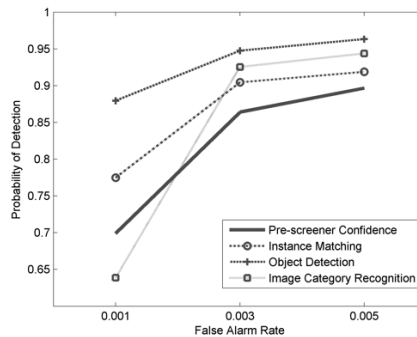


FIGURE 3.23: This figure presents the top ROC operating points from the three different detection frameworks considered. All performance levels presented include fusion with the pre-screener confidences. The object detection framework provided the best performance over all methods. This is likely due to the ability this framework has shown to account for intra-class variation in image processing tasks.

Instance matching methods can be effective at matching between target types under similar conditions. However, changes in soil composition and noise can alter a response appearance enough that both point localization and descriptor matching between two B-scans is disrupted. This caused poor performance in matching between target responses in the training and testing data which ultimately led to poor overall performance.

While the instance matching framework suffered due to the large variations in target appearance, object detection was designed to account for more intra-class variation. Performance using this framework surpassed all other methods tested. Given a set of labeled examples of target responses, object detection is the most fitting model reviewed in this work. In addition, methods to automatically generate labeled data produce comparable performance and allow for training a detector without the need for an expert to label the data.

Image classification was tested as it did not require isolated image areas in order to train a model. Utilizing the bag-of-words approach, the detector was able to add additional information to the system. However, the main observed drawback of this model is the lack of spatial awareness. The frequencies of learned centroids within the 2-D image that are used for classification are not affected by the layout of the patches. This negatively impacted performance especially when using smaller sized patches. For larger patches however, this was less of an issue as each patch contained more spatial information. The drawback with larger patches however is an increase in the amount of quantization that occurs as a result of more variation within centroid clusters.

3.2 GPR Patch Representation

Following the successful application of object detection methodology to detect target responses in GPR data, a more comprehensive study of GPR data representation was performed. This section utilizes the framework of a sliding window object detector as presented in Chapter 3.1 due to it providing the best performance as well as fitting the class of problem presented in GPR threat detection. The purpose of this study was to gain an understanding of performance implications using different families of feature extraction techniques to represent GPR data. The three types of feature extraction techniques used in this research are: pixel gradient methods,

binary comparison methods, and methods that utilize raw pixel data.

3.2.1 Feature Extraction Methods

This section introduces the feature extraction methods used in this comparative study. A total of 9 extraction methods were tested in this work.

Gradient Based Feature Extractors

The gradient based feature extraction methods tested in this work all share the commonality that each pixel within an image patch is represented by its gradient magnitude and angle. Each method then aggregates these values locally to represent the structural characteristics of the data within each local region.

Edge Histogram Descriptors The edge histogram descriptor (EHD) is a feature extraction method originally identified as a standard method to characterize the texture within small image patches [29]. The first use of EHD in landmine detection focused on the ability of EHD to recognize the changes in a texture across a B-scan that contains a target response [13]. The EHD feature extractor characterizes an area by categorizing each pixel into one of five edge types and counting the number of each edge type. The edge type categories are, vertical, horizontal, diagonal, anti-diagonal, and non-edge. Pixels whose gradient magnitude do not surpass a threshold value are categorized as non-edge pixels. In order to observe the change in the EHD feature across an image region, the region is broken up into 7 overlapping column areas. Each column area produces a histogram, and the descriptor for the overall image region is the concatenation of all 7 histograms. Figure 3.24 shows the result of processing the EHD descriptor. In this figure, the shift in the EHD feature is evident as the dominant edge shifts across the hyperbolic target response.

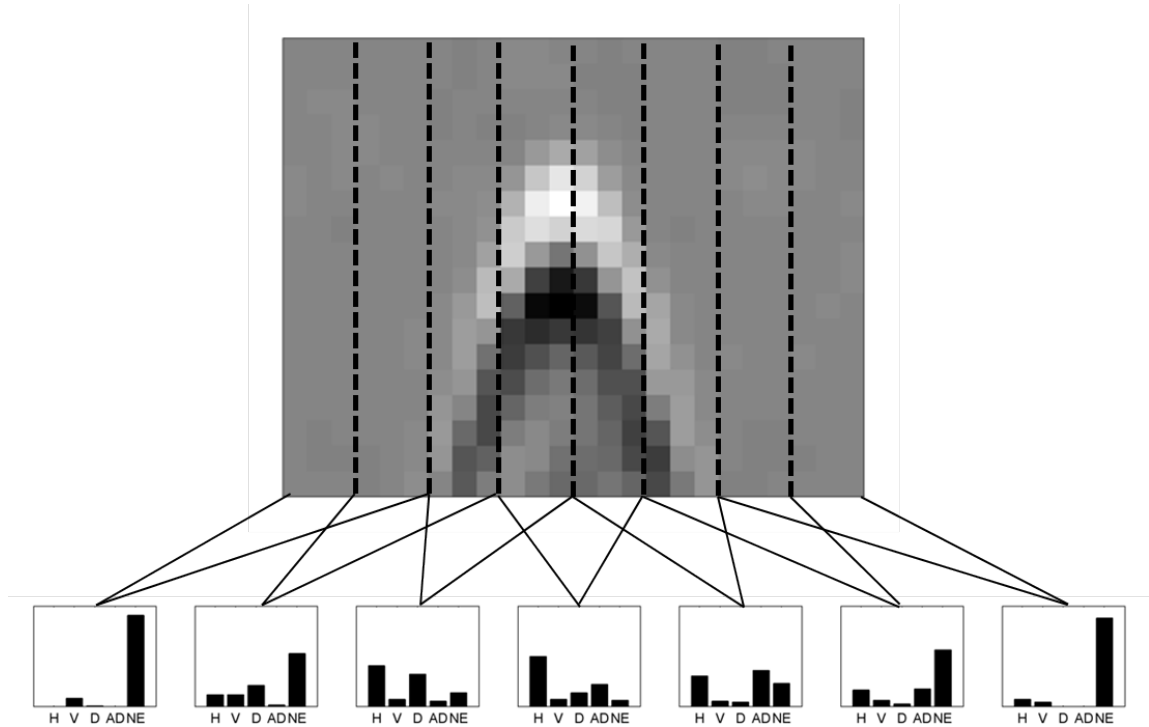


FIGURE 3.24: This figure presents an example of processing a GPR patch of data using the EHD descriptor. The patch is separated into 7 vertical areas. Within each vertical area, the pixels are counted according to their edge type. The edge types are: horizontal (H), vertical (V), diagonal (D), anti-diagonal (AD), and non-edge (NE). The counts for each edge type are shown in the bottom panel for each overlapping vertical area.

Scale Invariant Feature Transform The Scale Invariant feature transform (SIFT) is a method originally introduced for the purpose of finding instances of an identical object in a cluttered scene. The SIFT descriptor calculates the descriptor by aggregating local gradient magnitudes based upon their gradient angles. See Section 3.1.1 for a detailed explanation on the SIFT descriptor

Speeded-up Robust Features The speeded-up robust feature (SURF) was designed to be a computationally inexpensive alternative to the SIFT descriptor. Additional details regarding this descriptor as well as an example image can be found in Section 3.1.1.

Histogram of Oriented Gradients The Histogram of Oriented Gradients (HOG) is very similar to SIFT in that it operates by aggregating gradient information over small local areas. One of the primary differences is that HOG only considers angles between 0 and 180° whereas SIFT considers angles between 0 and 360°. For a detailed descriptor of the HOG descriptor, see Section 3.1.1.

Binary Comparison Descriptors

Binary comparison descriptors operate by comparing pixel intensity values between two locations within an image patch. The output of each comparison is simply a 1 or a 0 depending on which location contains a larger value. These methods are popular as this binary data has the potential to be stored and processed very efficiently.

Binary Robust Independent Elementary Features The BRIEF descriptor describes an area by comparing pixel intensities at random locations within an image patch. Additional details on the BRIEF descriptor are outlined in Section 3.1.1.

Local Binary Patterns The local binary pattern descriptor (LBP) was originally designed as to differentiate between different textures in visual images [58]. LBP characterizes texture by representing each pixel by its relation to its eight neighboring pixels. An 8 bit string is generated for each pixel where each bit represents whether or not a neighboring pixel has a higher intensity value.

A total of 256 different vectors are possible for each pixel. In practice however, it was found that 58 of the 256 possible vectors represented 90% of all pixels [59]. These 58 patterns are identified by the number of transitions (0→1 or vice versa) that occur when traversing the 8 bit vector in a circular fashion. These 58 vectors, known as uniform patterns, encompass all patterns that contain less than 3 transitions.

When using LBP features in recognition tasks [60] and in texture classification problems [59], histograms of pixel representations are used to represent small local

areas. When only uniform patterns are considered, a total of 59 bins are used with the last bin representing all pixels that have a non-uniform pattern. In this research, each image patch is split into 6x6 pixel sub-patches. Within each sub-patch a histogram is generated based upon the patterns observed for each pixel. The final descriptor for a given image patch is then the concatenation of each sub-patch histogram vector.

Pixel Representation Methods

Utilizing raw pixel values in the representation of each image patch allows for the statistical relevance of each location with the patch be evaluated by the classifier. These methods can be desirable in certain applications when the calculation of features is a very time-constrained problem.

Normalized Pixel Values Normalized pixel intensity values is a common method of representing a given image area in many image analysis applications. This method simply rasterizes a normalized image patch to create a 1-D vector. Each feature within an observation then represents a pixel location within the image patch.

Normalization is performed by first calculating the mean and standard deviation of the pixel values within the image patch. Each pixel is then normalized by subtracting the mean and dividing by the standard deviation. As the computation is minimal, this descriptor is very fast to calculate. However, as the size of the image patch is increased, the descriptor length is increased at the same rate.

Principal Component Representation A lower dimensional alternative to representing data by its pixel representation is by projecting the pixel data onto a lower dimensional basis. A notable example of this method is the recognition of faces by using the principal components as the new basis [61].

The principal components used to project the data are learned from all patches in the training data. These components are learned in an unsupervised manner aiming

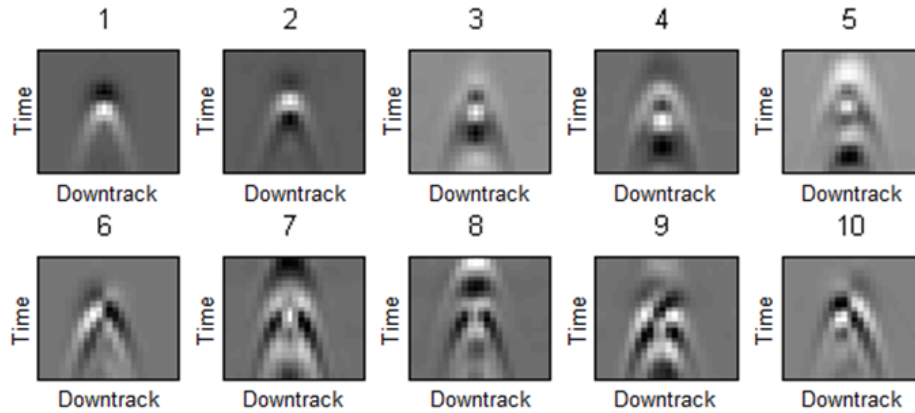


FIGURE 3.25: The above images are the first 10 principal component vectors of all training patches, both target and non-target. All GPR data from the testing set were represented as a linear combination of the principal components creating the descriptor for each patch.

at capturing the maximum amount of variance in the data. By representing each sample as a weighted sum of the principal components, the dimensionality is limited by the number of principal components used. Figure 3.25 shows an example of the top 10 principal components from one training fold. These principal components appear very similarly to visible target responses. This indicates much of the data variance is caused by these responses.

Sub-patch Principal Component Representation Representing an entire large patch by the principal components of the rasterized image patch is an effective method of preserving the overall variance of the data. However, as can be seen in Figure 3.25, the energy in each image is highly focused in the center of each patch where most of the energy response is located. This is caused by the fact that the most variance in these GPR patches occurs in the central area. Therefore, much of the variance is in the extremities that could contain some useful information is lost. It is for this reason that this research also tests a slight variant to the principal component representation method.

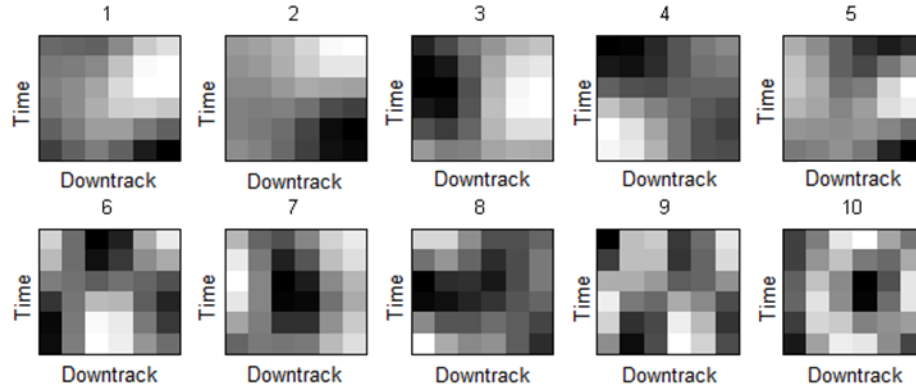


FIGURE 3.26: This figure shows the first 10 principal components for sub-patches. All sub-patches regardless of position within the image region utilize the same principal components. Each sub-patch was represented by a linear combination of these principal components.

As opposed to finding the principal components of the patch in its entirety, this method finds the principal components of small 6x6 pixel regions within each image patch. These smaller image patches are then represented by a linear combination of 6x6 pixel principal components. Figure 3.26 shows the top 10 principal components for the 6x6 sub-patch regions. The representation of each 6x6 pixel patch is concatenated to construct a descriptor for the overall image region.

3.2.2 Results

Each descriptor was tested using various image window sizes as well as different classifiers. Experimental results were generated using a variation of the receiver operator characteristic (ROC) curve. Instead of evaluating the probability of detection (PD) as a function of the probability of a false alarm (PF) given all pre-screener alarms, this research evaluates the PD as a function of the false alarm rate (FAR). The FAR is used in this application in part because we are only processing pre-screener alarms. If the PD/PF ROC curve is used, the performance of the curve will be heavily influenced by the amount and type of alarms passed through by the pre-screener. In order to visualize a large number of results on the same plot the performance of the ROC

curves are represented by the partial area under the curve (AUC) of the ROC curve. In this work the partial AUC measures the AUC between 0 and an FAR of 0.005 FA/m^2 . This represents the range from the lowest possible, and highest acceptable FAR set by the operators of the detection system. A FAR of 0.005 FA/m^2 given a 3 m wide sensor panel indicates that a false alarm occurs every 67 meters.

Experimental Setup

To ensure that all feature extraction methods are compared fairly, the data available to the feature extractors as well as the classification methods are fixed. While some of these extraction methods have been applied successfully using alternative methods [13, 62], this research utilizes an object detection framework as described in Chapter 3.1.

Pre-screener alarms from the training set are used to produce positive and negative examples in order to train the discriminating statistical classifier. The positive and negative examples are generated from 2-D image patches, where positive examples contain a target response, and negative examples contain only background GPR data. The feature extractor being tested computes a descriptor for each patch that then creates a data set where each example has a feature vector. This dataset is used to train the statistical classifier.

In testing, when given a pre-screener alarm, image patches of the same size are extracted at regular intervals across the image. Each image patch extracted is then used to calculate a feature vector using the feature extraction method being tested. The set of feature vectors from each patch is then classified by the trained classifier. The maximum confidence output among all samples taken from a given pre-screener alarm is retained.

Training Data In order to ensure reproducible results, the training data in this work must be determined automatically. Hand labeled GPR data is both a time consuming and an ambiguous task. As shown in chapter 3.1, performance using automated training data is comparable in performance to hand labeled methods on this dataset.

In this work, each pre-screener alarm that contains a target response produces a positive example for the classifier. This positive example is an image patch intended to encompass parts of a target response. As described in section 3.1.2, the location at which to extract a positive training example patch is determined by the smoothed energy in the central A-scan of a pre-screener alarm. The maximum location serves as the center of the patch extracted at a target response.

In order to properly train a statistical classifier, examples of non-targets are also necessary to extract from the data. In this work multiple negative examples are taken from each pre-screener alarm without a target. Image patches representing negative instances are taken at regular samples at each pre-screener alarm location. This ensures that the negative examples will encompass both low energy background data, as well as higher energy areas that initially caused the pre-screener alarm.

Feature Extraction Results

The results presented below in Figure 3.27 provide the partial AUC performance as described in Section 3.2.2 for each descriptor using the image window size and classifier that produces the best performance. These results indicate that the gradient methods, with the exception of SURF, consistently produce good performing detectors. The only other features that perform nearly as well are the LBP features. It is noted however that many similarities exist between the LBP features and the gradient based methods. The uniform LBP features, as described in section 3.2.1, do an excellent job at characterizing pixels along edges. The 36 LBP features within a 6x6 region are then histogrammed to form a 59 dimensional representation of the

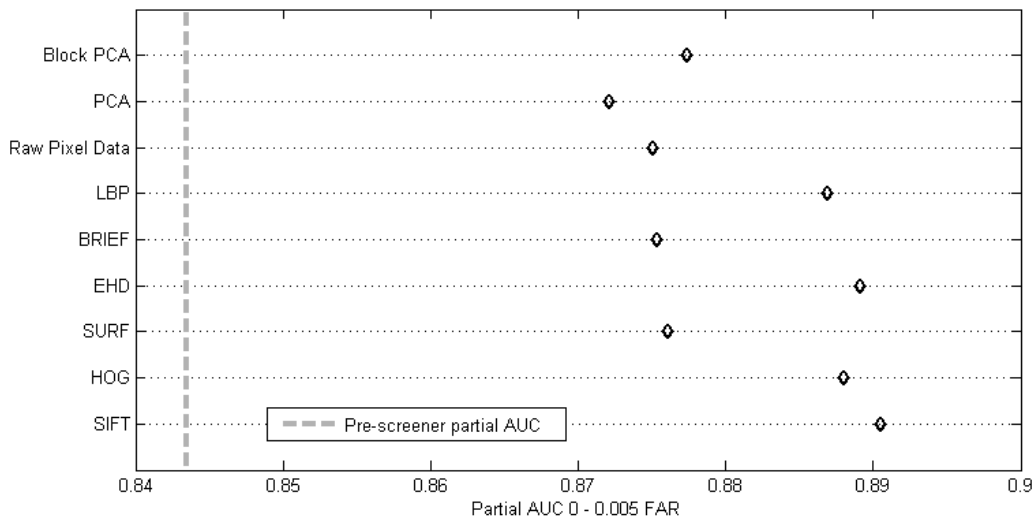


FIGURE 3.27: This figure shows the best performance achieved using each of the feature extraction methods tested as shown on the y-axis. Gradient methods that produced an angular histogram or weighted histogram of the gradients in small local regions outperformed all other methods. LBP features, which share many traits with the top performing gradient methods also performed very well.

local image. This is very similar to the histograms or weighted histograms that are found in the successful gradient methods.

Fusion with Pre-screener Confidence

In all cases, the detection performance of each classifier was improved when fusing with the pre-screener confidence. Figure 3.28 provides the performance of each feature extraction method with and without fusion. The gain in performance across each method indicates that the information provided by the pre-screener confidence is complementary to the information provided by the feature extraction method. This is likely due to the fact that each feature extraction method contains a form of normalization that is an attempt for each descriptor to be energy invariant. Therefore when fusing with the energy-based pre-screener confidence additional information about the target response is introduced.

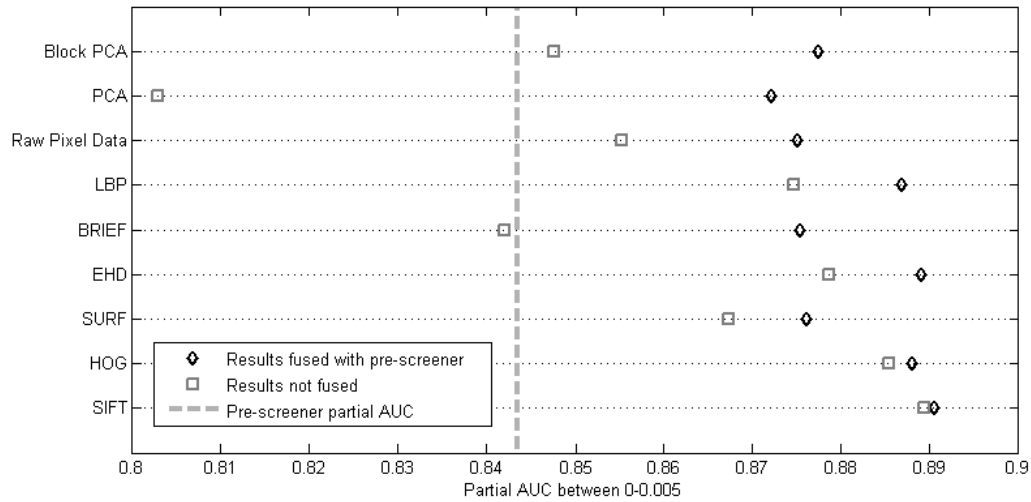


FIGURE 3.28: This figure shows the partial AUC performance of each feature extractor (y-axis) with and without fusion with the energy based pre-screener confidence. The increase in performance represents the additional information that the signal energy provides to each feature extraction method.

Classifier

This work evaluated the use of three different statistical classifiers, PLSDA, a linear SVM, and a non-linear SVM with a Gaussian radial basis function. Figure 3.29 indicates that in all trials, the non-linear SVM outperformed both linear classifiers. While some feature extraction methods benefited more from the non-linear classifier, other extraction methods, such as SIFT and HOG performed very well regardless of classifier used and could produce competitive results using a quickly trained linear classifier.

3.2.3 Conclusions

The work presented in this chapter evaluated a number of feature extraction methods applied to buried threat detection in GPR data. The results indicate that descriptors designed for visual image feature extraction perform very well when applied to GPR data. In particular, methods that characterize pixel gradients and aggregate

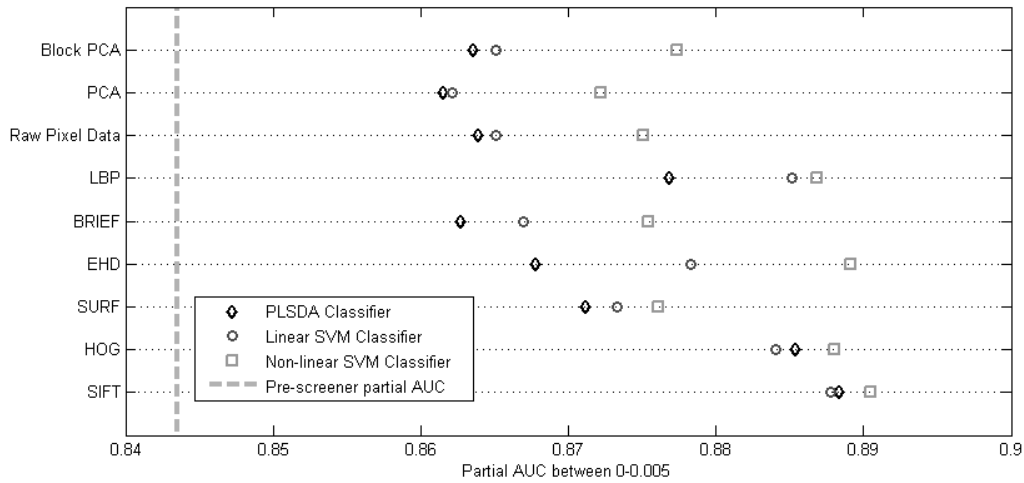


FIGURE 3.29: This figure presents the performance of each feature extractor method utilizing different classification algorithms. While some feature extraction methods had performance that was more consistent between classifiers, all methods performed the best utilizing the non-linear SVM classifier.

over local areas offer the best performance. The top performing methods from each feature extraction type is presented in Figure 3.30. These ROC curves present the performance across a range of FAR values.

The results presented in this chapter align well with the current image processing literature [41] that recognizes these gradient methods as the top performing out of the box feature extractors. This work has also demonstrated that the LBP feature extractor also generates discriminative features for GPR data. The combination of these descriptor techniques has recently shown improved performance in visual images [63] and could be explored in future work.

While this research has evaluated a number of feature extraction techniques, it is noted that this is a rapidly advancing field in the area of computer vision. Slight modifications to to these basic methods have shown performance improvements on a number of datasets [41, 63]. In addition, methods of learning features from the data are becoming more popular and outperforming state of the art methods based upon

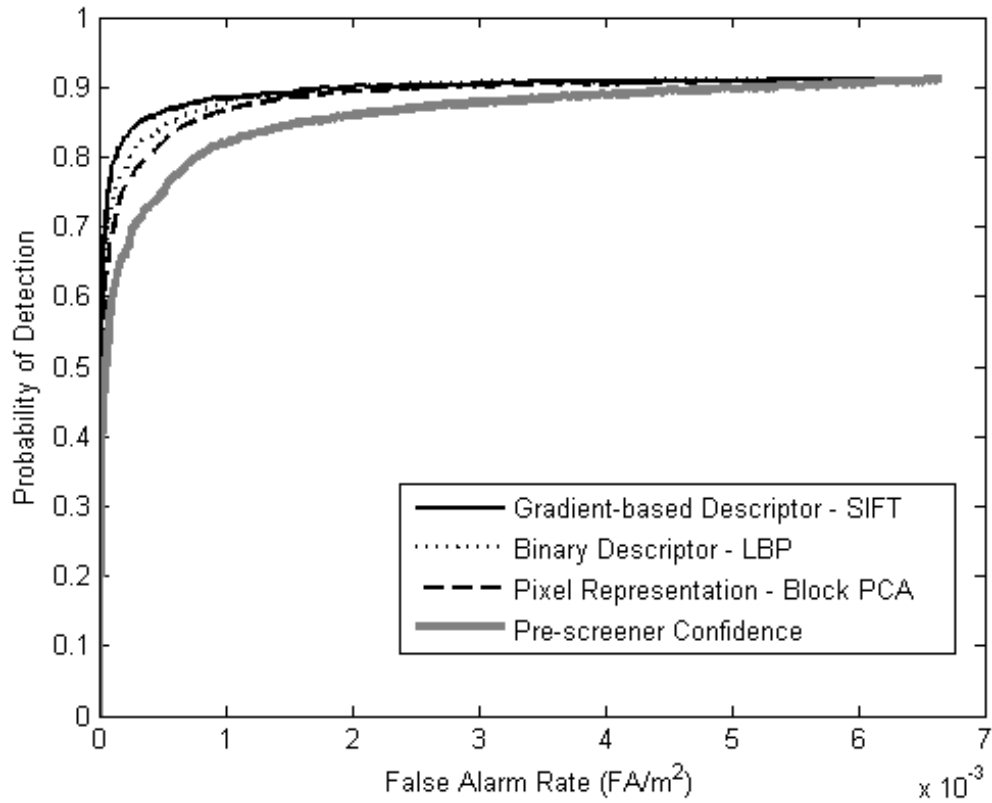


FIGURE 3.30: This figure presents the ROC curves from the top performing feature extraction method within each category. As previously shown, the gradient based method performs the best while the best extraction method based on raw pixel values fails to provide good performance.

hand designed features. The use of these methods on GPR data are explored in the following chapter.

Physics-Based Augmentation of GPR Target Signatures

The current state of the art GPR landmine detection relies on statistical classifiers to effectively discriminate between buried landmines and subsurface clutter [10, 13, 17, 64]. These statistical classifiers rely on the availability of a training dataset that must encompass the statistical variation in target signatures that is likely to be encountered by the fielded system. Unfortunately, due to the cost of collecting data and the large number of factors that affect a GPR response, it is impossible obtain training data encompassing every possible condition. This chapter aims to increase the variability found in the collected training dataset using a method of augmentation that is motivated by the physics underlying GPR sensing. This chapter shows that this augmentation method for the training dataset can improve the discrimination performance of a statistical classifier in situations of very limited training data as well as in in situations with larger training datasets.

4.1 Variation in GPR Data

To understand how to overcome challenges due to training set inadequacies, it is useful to understand how and why target/non-target signals vary beyond the usual issue of additive noise. While a portion of variation in signal responses does in fact come from noise, this research focuses on the variation in target signature shape due to various environmental parameters.

4.1.1 Variation in Target Configuration

The shape of a subsurface response as a sensor passes over a flat homogeneous region is largely dependent on the construction of the object. In addition to the size and shape of the entire object, the internal make up of a target can also affect the signature if the outer shell is permeable to radar signals. This can be seen in Figure 4.1 which shows the GPR responses from different types of plastic shelled landmines. The subsurface signatures produced due to each landmine type are visually discriminable.

Due to these types of differences, some of which are illustrated in Figure 4.1, when using a statistical classifier to detect buried explosive, it is important to have multiple examples (or at least one) of different types of targets. Without training examples of each specific target type, the classifier may be very unpredictable when assigning a confidence in a test scenario.

Unfortunately, due to the unpredictability of landmine design it is difficult to predict the kind of variation possible due to unknown landmine types. Instead, this research focuses on the variation introduced by predictable environmental factors that have simple physical models.

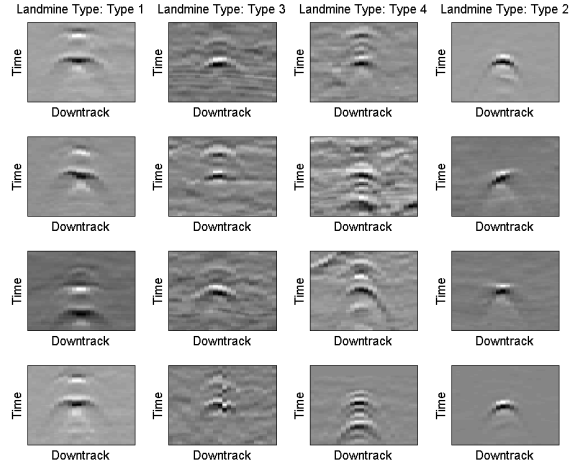


FIGURE 4.1: B-scans from four different types of non-metal cased landmines are shown in the above figure. Each column contains four GPR responses randomly selected from the same target type. Each of these target types have consistent distinguishing shape features differentiating them from each other. For instance, the response generated by landmine type 2 contains much less visible ringing than the other landmine types.

4.1.2 Variation Due to Environmental Factors

Many of the environmental factors that influence a GPR signal cause significant variation in the appearance of target responses. As discussed in Section 2.2.1, methods applied in pre-processing have been developed to mitigate the effects of certain environmental sources of variation. These pre-processing methods help to account for factors such as target burial depth, attenuation, and antenna motion. There are numerous additional environmental factors that are not accounted for in preprocessing steps. Examples of these environmental factors include soil inhomogeneities, surface contours, surface objects, uneven subterranean layers, and differences in soil dielectric constant. Each of these environmental factors can affect a signal response in different ways. For instance, soil inhomogeneities can contribute background noise without affecting the shape of a target response. On the other hand, surface contours or ground roughness can substantially affect a 2-D target response due to variable

propagation time between the sensor and target. The effect of an uneven surface over a buried target is shown in Figure 4.2.

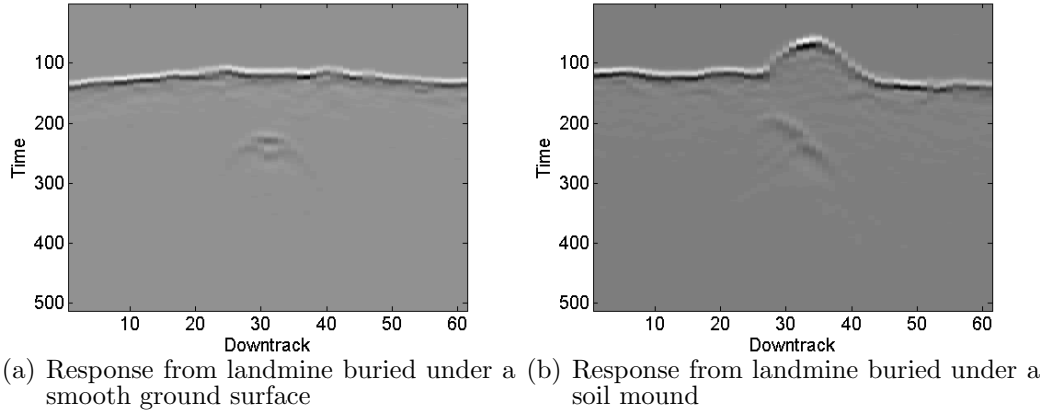


FIGURE 4.2: This figure displays a GPR response from a metal target that is buried under a flat surface, (a), and a response given the same target when buried beneath a soil mound seen in (b). Note that the presence of rough ground can substantially change the shape appearance of the GPR response. This shape warping is caused by time delays due to the additional soil the signal must pass through to reach the target.

This work focuses on the effects of surface roughness and uneven emplacements on a landmine’s GPR response shape. In both of these cases, the distance the signal propagates through the soil changes as the sensor passes over the target. This kind of environmental variation is common on unpaved, poorly maintained roads. Not only do these types of roads make it easy to conceal burying a target, these roads are also very common in current conflict areas as well as those struggling with landmine remediation for humanitarian purposes.

Effects of Uneven Ground Surface

Subsurface objects generally have a hyperbolic-like appearance in 2-D B-scans. This phenomena is caused by the time of arrival of the initial reflection from the object as the sensor approaches, passes over, and moves past the buried object. Due to the ground alignment performed during the pre-processing of the signal (see Section

2.2.1), the target response alignment is dictated by the distance between the ground surface and the target.

To theoretically predict the effect of an uneven ground surface in the data, the speed of propagation in soil must be known. Given a soil with a dielectric constant of 7 [65], the propagation speed would be approximately $1.13 \times 10^8 m/s$. Given a change in soil height of $1cm$, and given that the signal must travel both towards the target and back up to the antenna, the time shift would be the equivalent of traveling through $2cm$ of soil. This would produce a delay of $1.765 \times 10^{-10}s$. Given that the panel sampling rate is $1.303 \times 10^{-11} samples/s$, the $1cm$ ground height difference would delay the initial target reflection by about 13 time samples after the ground reflection. See Figure 4.3 for a sense of how much a 13 sample shift would affect a landmine response. Considering that these systems operate on dirt roads containing divots from vehicle traffic, washed out regions, rocks and debris, and many other symptoms of poor maintenance, it would be expected that the ground height changes will often be much more than 1 cm.

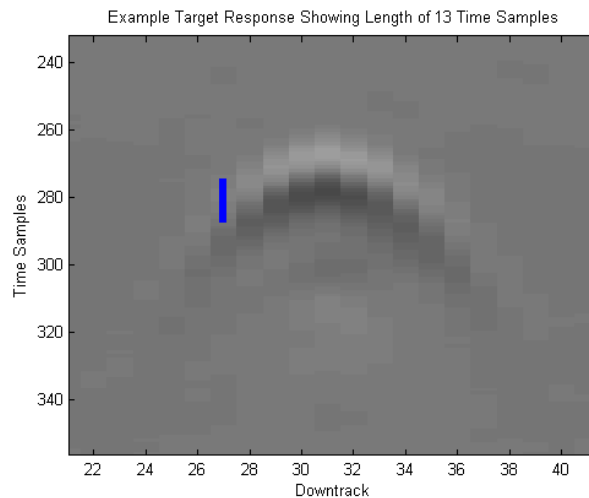


FIGURE 4.3: This figure shows an example target response. The solid bar on the left of the target is a line spanning 13 time samples. A change in soil level between the target and ground of 1cm can cause a shift of 13 time samples.

4.2 Understanding the Effects of Varying Ground Height

This section further evaluates the effects on GPR responses due to varying soil height. First, a GPR data simulation tool is used to examine the variation in a controlled environment with no noise. Next, GPR data from the actual sensing system is compared when collected over both smooth and rough surface contours.

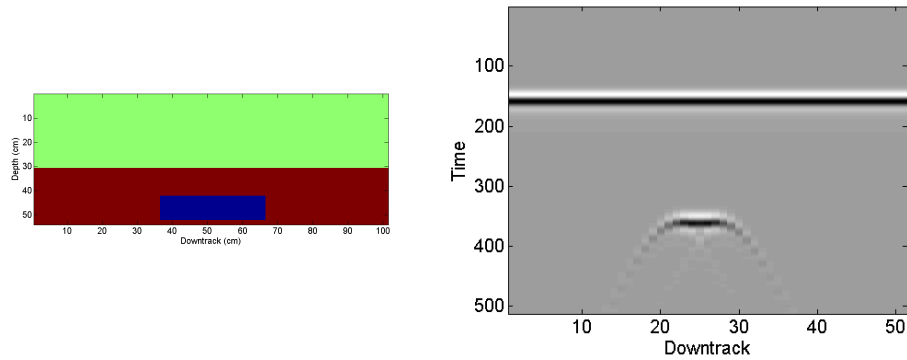
4.2.1 Simulating Environmental Conditions

In order to isolate the effect of uneven ground conditions, a simulation tool is used to create GPR data, given a 2-D scene. This section utilizes the gprMax [66] simulation tool. The gprMax tool is used to simulate the propagation of electromagnetic waves using Finite-Difference Time-Domain (FDTD) numerical methods. Using this tool it is possible to set up a scene representing an object buried in soil. The received GPR signals are simulated given the locations of transmitting and receiving antennas. The excitation signal in these experiments was set to closely resemble the frequency and time sampling of the actual GPR detection system.

An environmental setup of the gprMax simulation is illustrated in Figure 4.4(a). For simplicity, a metal rectangle is used in 2D simulations to represent the cross section of a metal-cased target. The resulting signal from the gprMax simulation is shown in Figure 4.4(b). This simulated GPR data can then be treated as a 2-D B-scan that can then undergo standard preprocessing methods.

Simulations of Rough Ground Surface

Using the gprMax simulation tool, it is possible to simulate the effects of surface roughness. Figure 4.5(a) shows a scene with a bump in the ground surface. The ground bump has a height of about 3.5 cm over the original ground surface. The resulting data simulated from the antenna passing over the top of the object is shown in Figure 4.5(b). This data can be compared to the data shown in Figure 4.4(b). As

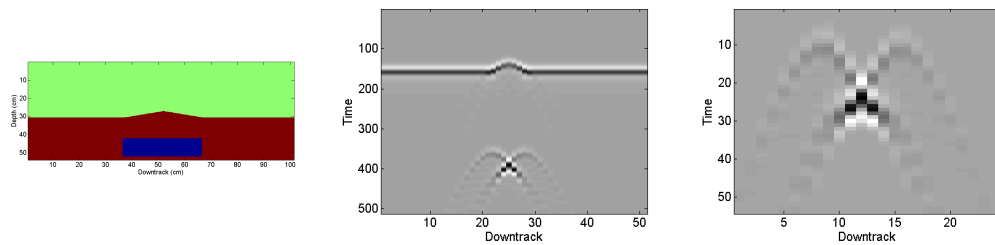


(a) Physical scene set for simulation in gprMax

(b) Simulated GPR response

FIGURE 4.4: The leftmost figure shows an example gprMax scene that is used for simulating GPR signals. The scene contains a rectangular metal object buried 3 inches deep in homogeneous soil. The resulting GPR response is shown on the right. This data was generated using similar parameters to those of the actual detection system used in this research.

would be expected given the longer travel distance for scans near the middle of the target, the initial responses are delayed substantially compared to the flat ground simulations. These results show that an uneven ground surface can substantially alter the shape of the object response that a classifier would observe.



(a) Rough Ground Physical Scene (b) Simulated GPR response (c) Simulated Rough Ground Patch

FIGURE 4.5: These figures show the simulation of a rough surface with a metal target-like object. The leftmost image shows the physical configuration of the ground and target. The middle image shows the simulated GPR data resulting from (a). The image on the right is the pre-processed patch that would be the input into the feature extraction/classification system.

4.2.2 Real Data Examples of Rough Ground

Data shown in this section was collected in a real world experiment for the purpose of isolating the effects of ground roughness. A GPR panel was suspended from a track along the ceiling over a data collection lane. This allowed the antenna to traverse the lane in a consistent horizontal trajectory without motion caused by the vehicle. The data shown in Figure 4.6(a) provides the baseline GPR signature of a metal landmine buried at 3 inches under an area of flat ground. This data can be compared to Figure 4.6(b) that shows data collected over the same target at the same depth but with a mound of soil over the target. The data collected over the rough ground surface shows a very different target shape than the baseline flat ground data shown in Figure 4.6(a). Specifically, the response shown more closely in Figure 4.6(d) appears to be shifted downwards as the antenna passes over the soil mound. This is expected because the increased travel distance through the soil increases the time that it takes for the reflected signal to reach the receiving antenna.

Given the impact of the soil mound on the landmine response shape, it would be necessary to have training examples similar to this if these mounds are found in operating environments. The following section presents a method designed to simulate the effect of rough conditions in already collected GPR data. This will enable classifiers to learn target response characteristics caused by ground conditions not present in the training dataset.

4.3 Mitigation of Rough Ground

If exact ground elevation levels and soil properties are known, it could be possible to remove some of the effects of rough ground as a sensor collects data. Unfortunately, this information is difficult to collect because the estimation of the flat ground target response would require knowledge of the dielectric constant. Acquiring the dielectric

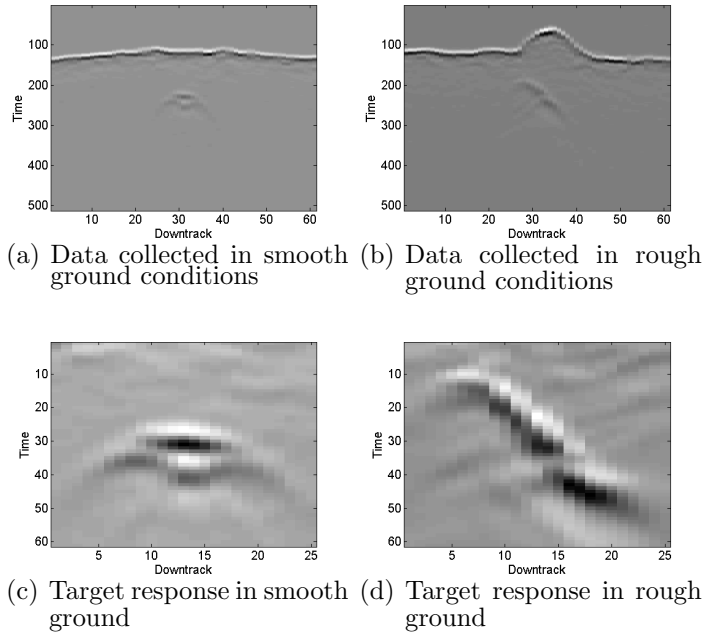


FIGURE 4.6: This figure shows actual data collected for this target type. The image in (a) shows data collected from an area with a relatively smooth ground surface. The image in (b) shows data collected over the same target but with an uneven ground surface (a mound). The target response extracted from the smooth ground GPR data is shown in (c), while (d) shows the extracted target response from the data collected over the rough ground. The shape is noticeably different between the responses shown in (c) and (d). This variation is important to capture in the training data if these conditions will occur in actual system operation.

constant of the soil is extremely difficult as the readings would need to be updated continuously in a vehicle mounted system. In addition, a target that is not laid completely flat can also cause similar response variation with no visible indication of the subsurface environment. Due to these limitations, this work must improve the robustness of the classifier without knowledge of the exact conditions of both the training and testing data. This research aims to do this by augmenting the dataset using training examples that have already been collected. The training dataset will be augmented with modified training examples that appear as though they had been collected over different ground surfaces conditions. This section introduces the A-scan time shifting methods used to modify existing data to simulate GPR responses

found under rough ground surfaces.

4.3.1 Reversing the Effects of Uneven Ground On Simulated and Real Data

Given the knowledge that uneven ground causes time shifts in the receipt of the initial target reflection, this section shows that the shifting of A-scans in time is able to create a comparable rough ground target response using data taken under flat ground conditions. This section applies time delays on the individual A-scans of a flat-ground object response based on a known contour from an uneven ground condition. The A-scans are then shifted in time to approximate the actual response recorded over the uneven ground contour. This method will be tested on both simulated and actual recorded GPR data.

In both the simulated and real GPR data rough ground examples, the suspended GPR panel (or simulated GPR sensor) is known to be on a flat trajectory over the object. Therefore, the time index of the ground reflection for each A-scan indicates the ground height when the sensor is above that location. The uneven ground contours can therefore be extracted from the time sample that contains the surface ground reflection (generally the time sample with the highest energy). These contours are a function of the ground height and the speed of propagation in air. However, simply time shifting A-scans by this amount would not provide the correct shift. This is because after ground alignment, the time shifts in the target response under uneven ground are a function of the ground contour and propagation speed in the soil. Therefore, when replicating the effects of rough ground, a multiplier is applied to the time shifts seen in the initial ground reflection. This multiplier is meant to approximate the propagation speed differences of air and soil to provide a linear relationship between the time sample differences in the initial ground reflection and the target response.

Figure 4.7(b) shows the results of warping the flat ground simulation shown in

Figure 4.7(a) using the ground contour extracted from the data in Figure 4.5. While the general structure of the target response is more similar to that shown in Figure 4.5(c), the warped target is not an exact match. This is likely due in part to multi-path reflections caused by the uneven surface that this warping method does not take into account. However, since this is simulated data, it is difficult to determine if the multi-path effects would be prevalent in noisier data with actual ground surfaces.

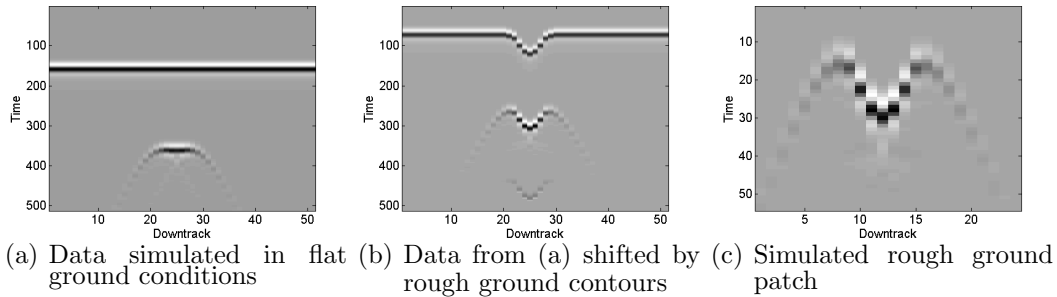


FIGURE 4.7: This figure presents the warping method used to simulate targets emplaced under rough surfaces. The process begins with the simulated data under flat ground conditions shown on the left. The center image shows the data after it has been warped according to the ground contour from the data shown in Figure 4.5. The resulting patch shown on the right is the selected patch from this process that approximates the patch shown in Figure 4.5(c).

Figure 4.8 shows the results of warping actual GPR data using this method. The target response recorded over flat ground conditions is shown in Figure 4.8(a). The ground contour was extracted from the data from Figure 4.6(b). This ground contour was then used to generate the shifted B-scan shown in Figure 4.8(b). The final patch after warping that would potentially be processed by a classifier is shown in Figure 4.8(d). This patch can be compared to the patch that was actually extracted from the rough ground condition that is shown in Figure 4.8(e).

While neither the warped simulated or warped collected GPR data exactly match the data generated in the actual rough ground conditions, they appear more similar in shape than the flat ground responses. Having exact matches is not vital in this

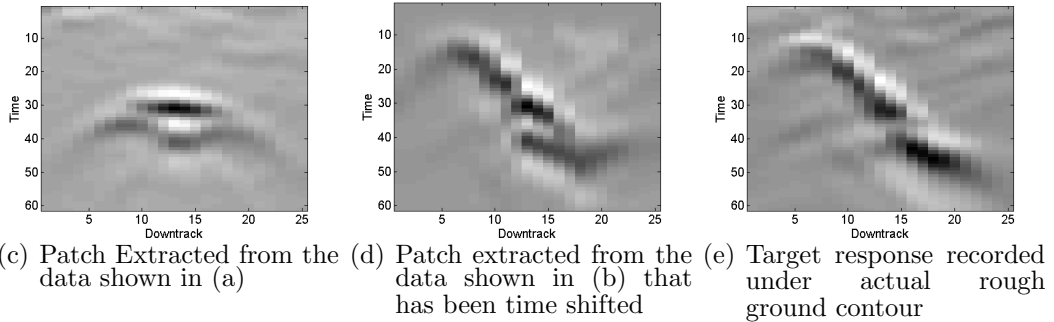
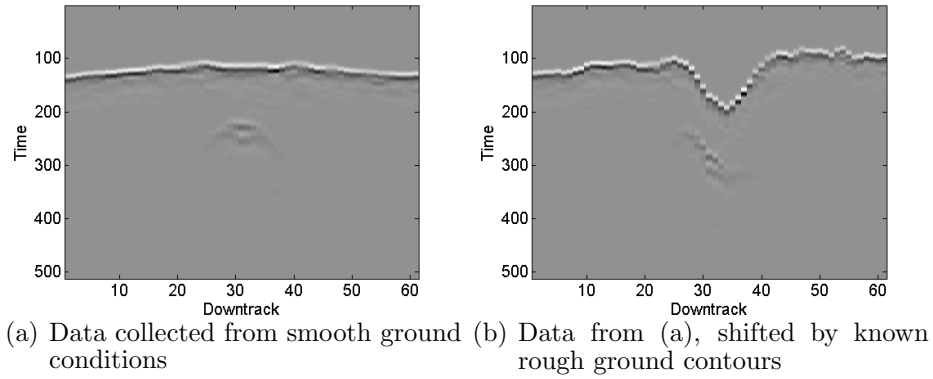


FIGURE 4.8: This figure shows the process of shifting A-scans from data collected over smooth ground conditions to appear as though it was collected over rough ground conditions. The image in (a) is the smooth ground data that will be shifted. The resulting GPR data after it has been shifted is shown in (b). The bottom row of images shows the extracted patches: (c). the original smooth ground GPR data, (d). the time shifted GPR that approximates uneven ground, (e). the field actual data collected under uneven ground conditions.

work because replicating the exact time delays is not the end goal. Due to antenna motion in real systems, the actual ground conditions of the training data is not known nor are the conditions when the detector is operating in the field. Instead, the goal of augmentation in this work is to provide a random assortment of uneven ground conditions that help increase the ability of the classifier to generalize across additional possible conditions. These results show that the warping of individual A-scans can create GPR responses that appear as though they are collected under different ground surface conditions.

4.3.2 *Augmenting the Dataset With Varying Ground Surfaces*

In order to augment a dataset of already collected GPR data where the ground contour information is not known, likely ground surface contours must be created that can then be used to time shift existing training data. In this work, A-scan time delay shifts will be created randomly. A 1-Dimensional version of elastic deformations motivated by [67] is used to create a time delay series to augment a given 2-D GPR data slice.

These distortions are generated by creating a vector the same length as the number of A-scans (columns) being shifted. This vector contains samples independently drawn from a uniform distribution between $[0, 1]$. The vector is then convolved with a 1-D Gaussian filter to control the local smoothness of the A-scan displacement. The output of that convolution is then multiplied element-wise by a scaling factor that controls the magnitude of the distortions. In this process there are two parameters, the standard deviation of the Gaussian filter and the multiplier. The standard deviation controls the smoothness of the shifts, while the multiplier controls the magnitude of these shifts.

Figure 4.9 shows example distortions created using different sets of parameters. The larger the standard deviation value, the more local smoothness between A-scan shifts. The multiplier controls how large these shifts are. In theory these parameters could be tuned depending on the ground type that the system expects to be traveling over.

4.4 Dataset Expansion From One Observation Using Uneven Ground Augmentation

This section analyzes an extreme case of limited training data - having only a single example of a particular target type. While this dataset's primary function is to tightly

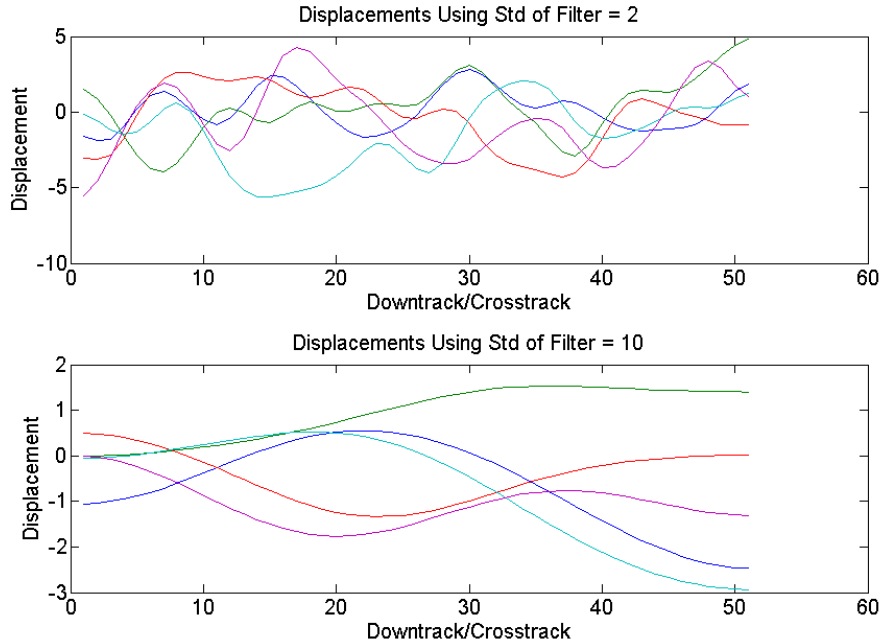


FIGURE 4.9: This figure shows example ground contours generated according to the process described in Section 4.3.2. The standard deviation of the filter determines the roughness of the contours while the multiplier determines the scale. The top plot consists of deformation samples with a low standard deviation parameter on the top, while the bottom plot has a much higher standard deviation.

control the augmentations produced, it is also a realistic situation due to the evolving design of new and improvised explosive threats. In fielded systems, missing a threat is obviously very costly. While a system that detects all possible explosive targets with a non-existent false alarm rate is the goal, this is not currently possible given existing technology. Instead, partially due to advances in vehicle safety, systems often operate with a balanced approach that tolerates a few missed detections in favor of a lower false alarm rate. In the case of a missed detection due to a new type of target in which the target is activated, at best only one observation of the target response is known. It is important to improve classification on these targets despite the little data available. This section presents the successful application of GPR response augmentation in order to improve the detection of a target type using

only a single observation.

4.4.1 Dataset and Classification for One Observation Expansion

In this experiment the existing training set contains four different landmine types, two with a metal exterior, and two low-metal content models. A total of 625 target examples and 2127 false alarm locations flagged by the pre-screening algorithm are used to train a classifier based on the GPR data patches extracted at these locations. The new target introduced is a plastic cased landmine that will be referred to as PlasticA. In this test, one observation of the landmine will be available to use for training. The test set contains 110 observations of PlasticAs taken from 16 different emplacements along with 268 false alarms not included in the training set.

In this section, a non-linear SVM is used for classification on the normalized GPR data. While feature methods provide state of the art results, the local aggregation common in high performing methods would obfuscate results. This work is focused on determining the validity of this method for general dataset augmentation.

4.4.2 Augmentation of Single Observation Results

The detection performance without the augmented data is shown in Figure 4.10. Both the performance with and without a single observation included in the training set is shown. The performance of the detector without any examples of this target type is very poor. The addition of a single training observation offers a substantial improvement in detection performance of PlasticAs.

In order to determine the kind of performance possible given more examples, Figure 4.11 shows the results under cross-validation when one emplacement, is used in the test set and the other 15 are added to the training set. These results confirm that if collecting more data is possible, it can improve the detection results of that target type.

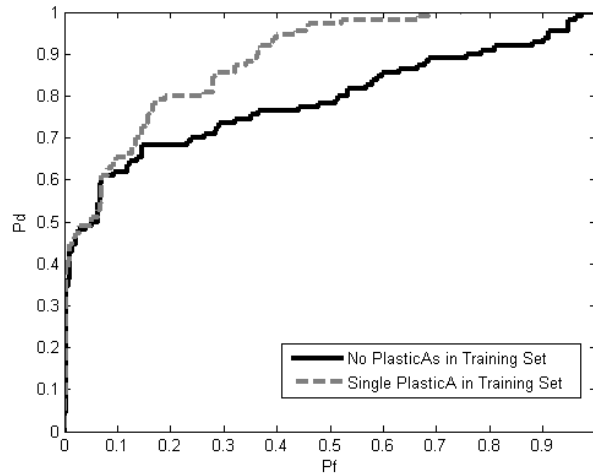


FIGURE 4.10: Baseline performance detection on test set of PlasticAs. The solid black line shows performance if no PlasticAs are included in the training set. The dotted gray line is the performance when a single observation of a PlasticA is included in the training dataset.

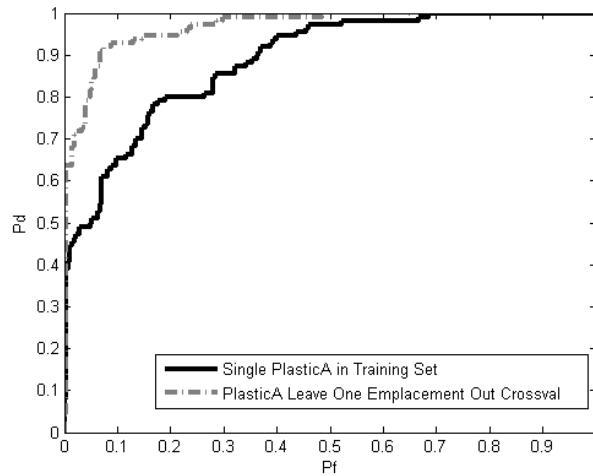


FIGURE 4.11: Cross-validation performance when additional emplacements of PlasticAs are added to the training set. For each emplacement, all other test emplacements are added to the existing training set. The cross-validated performance is compared to detection performance with a single observation of the PlasticA in the training set. This performance shows the effectiveness of collecting additional data when possible.

The single observation available for training is shown in Figure 4.12. Using this

observation, additional training data is generated by first generating ground contours using the technique described in Section 4.3.2. These generated ground contours are then used to shift the A-scans of the training set observation. These shifted observations are then added to the original training set that is then used to train the classifier.

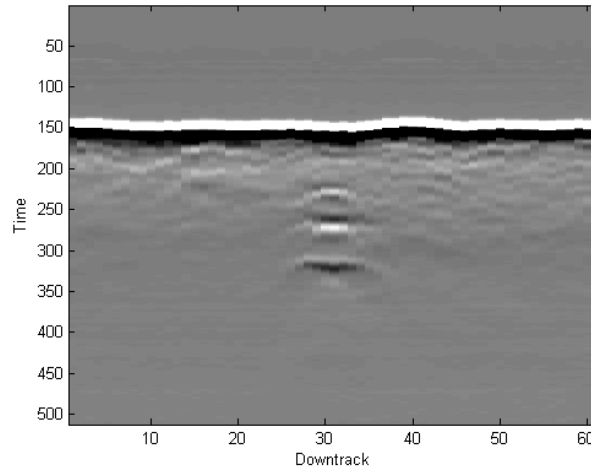


FIGURE 4.12: This figure shows the single observation used in these tests. This target data will be replicated and warped to simulate a variety of ground conditions that can be used to improve the ability of the classifier to detect PlasticA targets

Figure 4.13 shows example augmented data for a variety of parameters. As the standard deviation is increased (towards the bottom of the figure), the target response smooths out. The multiplier (increases towards the right of the figure) affects the samples by magnifying the deformations. Note that for $\sigma = 7$ the examples with higher multipliers still have only slight changes because the smoothing of the Gaussian filter ensures the deformations do not vary wildly across a B-scan.

One of the primary goals of this section is to determine what range of parameters benefits classification the most. Obviously, simulating extreme ground contours could cause a decrease in detection performance, and simulating contours that are too flat would have little to no effect. Performance using a variety of parameters is

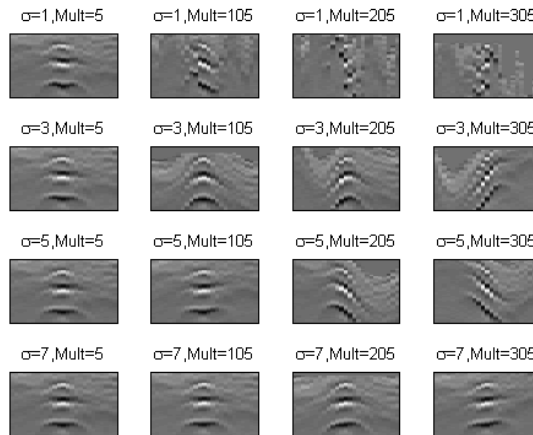


FIGURE 4.13: This figure shows example warped observation for a variety of parameters shown in the title of each subplot. These images can be compared to Figure 4.12 as they are all generated from the same data observation.

shown in Figure 4.14. Surprisingly, all of the parameters shown in this figure perform better than the classifier that had no augmentation. This is surprising because in Figure 4.13, many of the parameter values appear to provide completely unrecognizable GPR responses. Due to these unrecognizable target responses improving classification performance, the effects of augmentation are further explored.

Due to the performance gain achieved regardless of parameters, a portion of the training targets were withheld to test the performance on these types of targets when the data is augmented by PlasticAs. This validation set will ensure that while detection performance of PlasticAs is boosted, the performance on the original training targets does not suffer as a result. Using the same augmentations shown in Figure 4.14, the results on the held out portion of the training set are shown in Figure 4.15. The black circles in the figure represent parameter sets that performed as well or better than the non-augmented classifier. As might be expected the performance on the original targets degrades when augmented by the more extreme appearing observations.

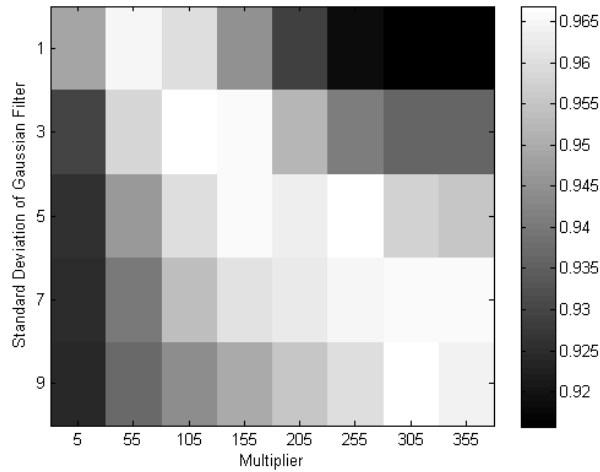


FIGURE 4.14: This figure presents the AUC results as a function of parameter values used to augment the single observation. The test set in this figure is composed of only PlasticAs and false alarms. Along the y and x-axis are the standard deviation values and multipliers used in the generation of distortion vectors. As a point of reference, the AUC performance of the classifier when using a single observation of PlasticA is 0.8862.

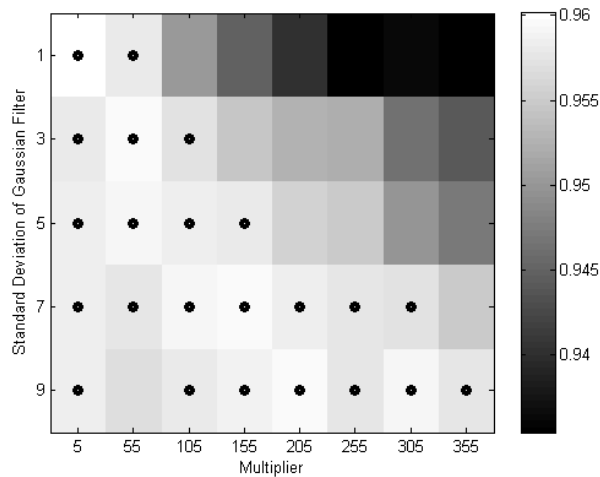


FIGURE 4.15: This figure presents the AUC of the augmented classifier on a test set consisting of non-PlasticA target types that are found in the training dataset. A black circle indicates that the parameter pair maintained performance above the AUC value of .957 that was achieved with no augmentation.

The fact that the detection performance increased despite extreme parameter values also indicates that performance can possibly be increased just by obtaining

a more general classifier. To test this theory, additive random Gaussian noise was added to the single observation at varying levels. The detection performance associated with this type of augmentation is shown in Figure 4.16. The results show that adding noise to the single observation multiple times does aid in classification over simply using the single observation. This however does not provide the same increase in performance as the augmentation using simulated rough ground surfaces given the same number of augmentations.

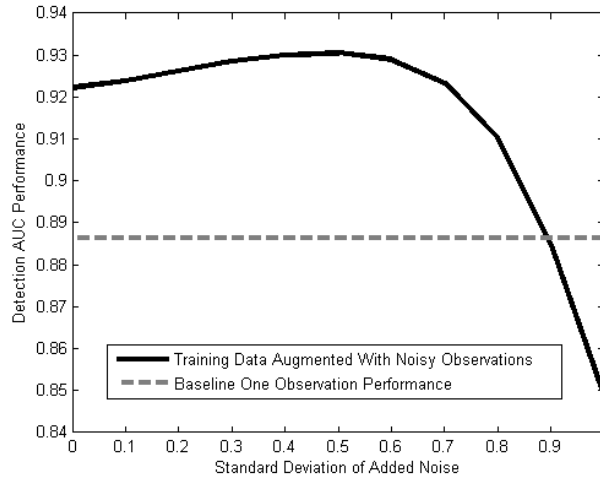


FIGURE 4.16: The figure above displays the AUC performance when multiple realizations of noise are added to the single PlasticA training observation and added to the training set. The performance does increase over no augmentation, but does not reach the level of augmentation with simulated rough ground surfaces.

To compare the methods of augmentation, the performance of one of the top performing parameter pairs is shown in Figure 4.17. This figure also shows the ROC performance using cross-validation on the testing set as well as the performance when noise is used to augment the single observation. The figure shows that performance using a single observation and ground surface augmentation is superior to the performance using no augmentation or augmentation using random noise. More training emplacements do perform better if available, however performance using ground surface augmentation and only a single observation is comparable.

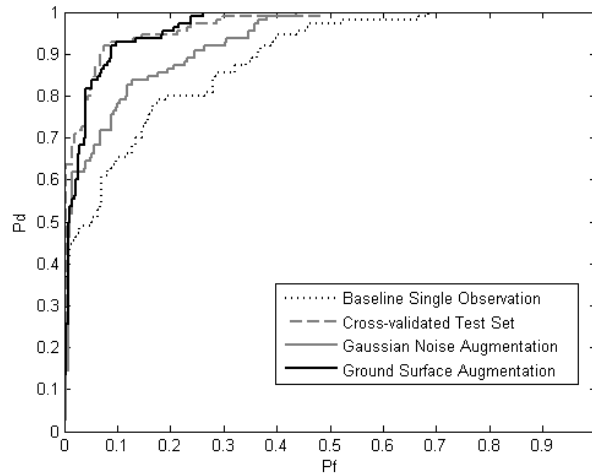


FIGURE 4.17: This figure shows the ROC performance of augmentation with simulated rough ground surfaces as well as augmentation with noisy instantiations of the single observation. Baseline ROC plots are also presented here as a reference to performance with only the single observation as well as cross-validated performance if additional emplacements of PlasticAs are included in the training set.

4.5 Additional Augmentation Experiments

This section presents additional experiments showing the application of GPR signal augmentations under different training conditions. This section builds upon work presented in Section 4.4 by using a subset of parameters that provided an increase in detection performance. These experiments utilize high performing parameters and apply augmentations to larger datasets in order to improve performance. In the first training condition, the training data is collected over a different soil type than is present in the testing dataset. These different soil types generate very different GPR responses causing poor detection performance since the training set data is not representative of the test set data. In the second training condition, a large-scale training dataset is used to evaluate augmentation performance when training samples are abundant. Results on the cross-validation of this dataset show the effectiveness of augmentation when training data is not severely limited.

In these trials, the selection of augmentation parameters were chosen based on the performance shown in Figure 4.14. The parameter pairs chosen consisting of standard deviation and multipliers respectively were [1, 55], [3, 105], [5, 155], [5, 105]. These values were chosen as they achieved the largest increase in performance on the augmented target type in Section 4.4 without a decrease in overall detection performance.

In contrast to the augmentation in Section 4.4, augmentation in this section is performed on all alarms, targets and false alarms. Each alarm is augmented 4 times, once for each parameter set. Similarly to Section 4.4, a non-linear SVM classifier is used to classify alarms based on the normalized pixel data.

4.5.1 Augmentation of Dataset to Tested on Different Soil Type

Previous attempts at augmenting datasets to combat differences in the dielectric constant between soils by stretching GPR data only showed small levels of improvement [68]. This method assumes that only the change in dielectric constant affects the target responses. The augmentation presented in this chapter is compared to the method from [68].

The data used for this experiment was collected from two different sites. The training set was collected in the Western United States in an arid environment and assumed to have a lower dielectric constant. There are a total of 3221 alarms in the training set made up of 826 target alarms and 2395 non-target alarms. The test set was collected in the Eastern United States which is assumed to have a higher dielectric constant and additional moisture in the soil. The test set is composed of 593 pre-screener alarms: 351 target alarms and 242 non-target alarms. The target population between the two datasets is kept constant so that the classifier will not be tested on any targets that for which it does not have training data.

Experimental Results

The results of this experiment are presented in Figure 4.18. Along with baseline performance and the two methods of augmentation, these results also show the performance of 10-fold cross-validation over the test set. The performance of cross-validation over the test set was expected to perform the strongest since the train/test soil conditions were constant. Instead, the small size of the test set likely hindered the cross-validation performance. With the exception of the results achieved using the rough ground augmentation presented in this chapter, the results on the test set are largely similar.

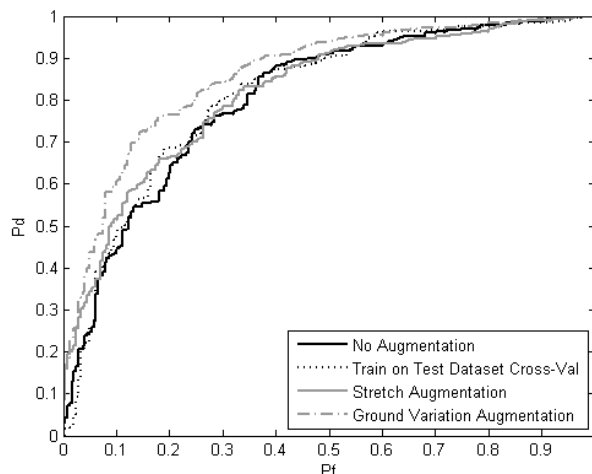


FIGURE 4.18: This figure shows the results of training on data collected over one soil condition and tested on data collected over a different soil type. This result shows that when the training set utilizes the augmentation presented in this chapter, the performance is improved over no augmentation and an augmentation method developed to assist in mitigating differing dielectric constants.

These results show that the rough ground augmentation presented in this chapter can increase detection performance when other types of variation are introduced to the test set. While it can not be proven that rough ground augmentation specifically replicates differences in dielectric constant, it does show that this method will improve the robustness of the trained detector when applied to datasets with differing

conditions.

4.5.2 Augmentation of a Large Training Dataset

This trial tests the effectiveness of GPR response augmentation when the training dataset is much larger. The dataset used comes from the same data collection used in Section 3.2. It consists of a total of 2395 alarms on targets and 3589 false alarm locations generated by the energy-based pre-screener discussed in Section 2.2.2. Cross-validation in this section is performed over 10 folds where observations of a single emplacement or physical area are restricted to the same fold. This ensures that no physical location is found in both the training and testing set.

In addition to classification on the normalized pixel values, this trial also utilizes the SIFT descriptor introduced in Chapter 3. The SIFT descriptor is included to test the effectiveness of augmentation with a feature extractor that has been found to be invariant to small changes in the GPR response shape.

Experimental Results

The results of this experiment are presented in Figure 4.19. It was expected that the performance difference due to augmentation would not be as sizable as in Section 4.4 due to the abundance of training data for all targets. However, the augmentation when using normalized pixel features is still an improvement over using no augmentation at all. This is not the case when the SIFT descriptor is used to extract features from the data patches. In this trial, there is no improvement in performance by introducing data augmentations when the SIFT descriptor is used for feature extraction. This lack of performance improvement is likely due to the quantization of gradient angles as well as aggregation that occurs over sub-patches. While these operations do not allow for augmentation to improve discrimination, they likely heavily contribute to the overall superior detection performance when the descriptor is used.

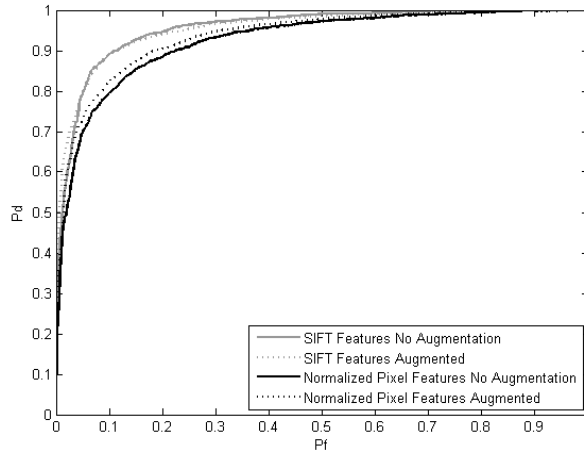


FIGURE 4.19: This figure presents the performance obtained when a large-scale dataset uses augmentation within the training cross-validation folds. In the case of normalized pixel features, a slight performance improvement is achieved. However when SIFT features are extracted from each data patch, it was found that augmentation does not improve performance.

While these results do not show an improvement when using state of the art GPR feature extraction techniques, it does not mean that other feature extractors cannot take advantage of the augmentations. It is likely that feature extraction methods that do not aggregate local areas within the data will benefit the most from this augmentation.

4.6 Conclusions

This chapter has shown the ability to augment existing training data in order to improve system performance when faced with both a challenging dataset where training data is extremely limited as well as a dataset considered large-scale. Using augmentation methods motivated by the physics behind the target variation responses, this research was able to produce a more robust classifier that improved performance over that achieved with the original training set.

In situations where very limited target response examples are collected, the gen-

eration of observations under varying ground surfaces helped improve performance on a specific target type when examples of that target type were extremely limited. This can be useful in cases where targets are detonated and the physical composition remains undiscovered. This research showed that augmenting the training set with warped versions of a single observation can improve detection performance nearing that of a classifier trained on over 100 observations of that target type.

This research also showed that performance improvements can be obtained for augmentations of larger datasets as well. These augmentations greatly increased the size of the training set, but when classifying normalized pixel features, the classifiers trained on the augmented training set outperformed the classifier trained only on collected data. However, when the SIFT feature extract is introduced, the augmented training set no longer increases the detection performance of the classifier. This is likely due to the design of the SIFT descriptor to be invariant to small changes in object shapes found in visual images.

This chapter has introduced a new method to improve the robustness of a classifier without the need to collect additional data. These augmentation methods are especially important when the amount of data available is very limited. However, due to the success of the augmentation on large scale datasets, this method has the potential to increase the performance across a variety of difficult situations. Further research into the robustness of this augmentation approach could improve current methods of collecting training data, making them cheaper and more efficient. Another future avenue for research in this area involves the use of classifiers trained on different augmentation parameter sets. Potentially, the augmentation parameters can be optimized to the severity and types of variation. This can produce multiple classifiers that are specifically trained to detect landmines in either smooth or rough conditions encountered by a fielded system.

Convolutional Neural Networks

In recent work [17, 69] it was found that the utilization of object detection methods on visual images have provided state-of-the-art performance in the detection of buried targets in GPR data. Recently, supervised deep convolutional neural networks have provided the top performing results across a number of visual image object detection benchmark datasets. CNNs generally outperform methods utilizing hand designed features in detection tasks if enough training data is available. In general however, deep convolutional neural networks require a large training dataset, something not available in most GPR applications. In this chapter, a small CNN is utilized to learn a discriminative filter bank that outperforms the use of features that were designed for images.

5.1 Convolutional Neural Network Model

The Convolutional neural network model can be thought of as a two stage process of feature extraction followed by classification. The feature extraction module consists of one or more convolutional layers. If more than one convolutional layer is used, the output of the prior layer is used as the input for the following layer. The clas-

sification stage generally consists of a multi-layered neural network usually referred to as the fully connected layers of the CNN which takes the rasterized output of the convolutional layers as an input.

5.1.1 Convolutional Layers

Each convolutional layer typically consists of convolution with a bank of learned filters along with an activation function. Between convolutional layers, the data is generally down-sampled using spatial aggregation across small blocks of data. A visual example of a convolutional layer is shown in Figure 5.1. The input 2-D patch is first convolved with a bank of filters. Each output is then processed through the activation function. The output of this example convolutional layer is then generated by aggregating values within local data blocks.

In mathematical terms with no local aggregation, the output of each convolutional layer I^j can be calculated by Equation 5.1. For this layer, a total of K filters parameterized as W provides an output of size $m \times n \times K$ where $m \times n$ are the first two dimensions of $I^{(j-1)}$. The function $f()$ represents the activation function used in the convolutional layer. Generally a non-linear function such as a sigmoid function is used.

$$I_k^j = f(I^{(j-1)} * W_k^j) \forall k \in 1, 2, \dots, K \text{ filters} \quad (5.1)$$

When a form of local aggregation is used between convolutional functions it is applied to I_k^j only on the dimensions m and n . If local aggregation occurs in 4×4 blocks of data, the input into the next layer is then $\frac{m}{4} \times \frac{n}{4} \times K$.

Of the parameters involved in each convolutional layer, only the filters, W are learned. All other parameters such as the size and number of filters are fixed.

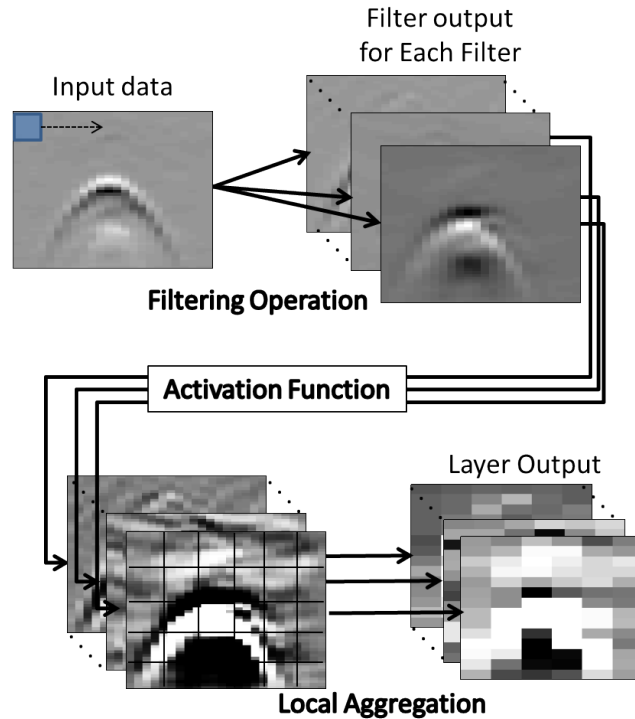


FIGURE 5.1: This figure displays example processing of an input patch by a convolutional layer. The input data (top left) is convolved with a bank of filters. The output of each filtering operation (top right) is then processed using the layers activation function. The output of the activation function (bottom left) is then locally aggregated across small pixel regions. The output of this layer (bottom right) can then be passed along to the next convolutional layer or rasterized to be further processed by the fully connected layers

5.1.2 Fully Connected Layers

The classification module of the CNN is a multi-layered neural network also referred to as the fully connected layers of a CNN. Neural networks can be seen as a way to map between an input vector and an output vector. Between these vectors neural networks contain a number of 'hidden' layers that are calculated from linear combinations of the previous layer using learned weights followed by an activation function. The activation function is generally a differentiable non-linear function such as a sigmoid. This relationship between nodes is often represented visually using connections between nodes. Figure 5.2 shows the graphical representation of an example

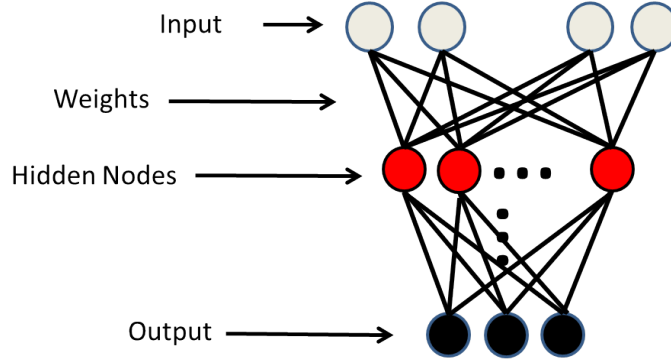


FIGURE 5.2: This figure shows a graphical representation of a multi-layered neural network model. As the data is introduced into the fully connected layers at the top of the model, forward propagation flows downward. The connected lines indicate that the value of each node is dependent on all nodes in the previous layer. Each connection between nodes has a unique weight that dictates the amount of influence each node in the previous layer have.

multi-layered neural network. The calculation of a node value can be represented mathematically by first defining the node, x_1^j , the first node in contained in layer j . The value of x_1^j is determined by

$$x_1^j = f\left(\sum_{p=1}^{n_i} x_p^{j-1} w_{p1}^j\right) \quad (5.2)$$

where n_{j-1} is the number of nodes in the previous layer and w_{p1}^j is the weight applied to x_p^{j-1} (the p th node in layer $(j-1)$) towards x_1^j . The function f is the activation function defined by Equation 5.3. In this research the task of target detection is defined as a binary task. Therefore the final layer of the neural network will be a single node whos value given a data observation will be the detection confidence.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.3)$$

5.1.3 Training a Neural Network with Back-propagation

Training a convolutional neural network is the same overall process as training a multi-layered neural network. The error between the desired outcome and network outcome is propagated backwards to adjust the weights between the layers. In the convolutional layers, it is slightly modified because the weights are shared over the entire image. This section explains this process known as back-propagation that is used to train the filters and weights in a convolutional neural network.

Parameter tuning in multi-layered perceptrons is generally performed using gradient descent. Gradient descent is used to find the minimum location on the error surface with respect to the model parameters. This process utilizes the gradient of the error in order to update the model parameters. In multi-layered perceptrons, the weights between hidden layers make up these model parameters. While the error function of the final layer is easy to calculate, a process known as back-propagation is required to train deeper layers. Note that throughout this section, some liberties are taken with linear algebra order to keep the mathematics more intuitive.

The desired gradient surface to adjust a given layers parameters, W_j is the partial derivative

$$\frac{\delta E}{\delta W^j} \tag{5.4}$$

If we define the cost function as squared error in a network with J layers, $E = \frac{1}{2}(Y - X^J)^2$, then we can find the gradient of the error surface with respect to the final layers weights:

$$\frac{\delta \frac{1}{2}(Y - X^J)^2}{\delta W^J} \text{ where } X^J = f(X^{J-1}W^J) \tag{5.5}$$

It is possible to calculate the partial derivative in Equation 5.5 using the chain

rule which gives

$$\frac{\delta \frac{1}{2}(Y - X^J)^2}{\delta(W^J)} = \frac{\frac{1}{2}\delta(Y - X^J)^2}{\delta f(X^{J-1}W^J)} \frac{\delta f(X^{J-1}W^J)}{\delta(X^{J-1}W^J)} \frac{\delta(X^{J-1}W^J)}{\delta W^J} \quad (5.6)$$

where

$$\frac{\delta \frac{1}{2}(Y - X^J)^2}{\delta f(X^{J-1}W^J)} = \frac{\delta \frac{1}{2}(Y - X^J)^2}{\delta X^J} = X^J - Y, \quad (5.7)$$

$$\frac{\delta f(X^{J-1}W^J)}{\delta X^{J-1}W^J} = f(X^{J-1}W^J)(1 - f(X^{J-1}W^J)) = X^J(1 - X^J), \quad (5.8)$$

and

$$\frac{\delta X^{J-1}W^J}{\delta W^J} = X^{J-1} \quad (5.9)$$

This gives a final derivative of

$$\frac{\delta E}{\delta W^J} = (X^J - Y)X^J(1 - x^J)X^{J-1} \quad (5.10)$$

This derivative from Equation 5.10 is then utilized to update the weights using the formula

$$W^j_{updated} = W^j - \alpha \frac{\delta E}{\delta W^j} \quad (5.11)$$

where α is a learning rate set to control the step size taken at each iteration.

For all but the last layer however, Equation 5.5 is not as straightforward as the weights of these layers effect on the error is much more complex. Equation 5.12 shows the calculation for determining the gradient of the error surface as a function of the weights at layer $J - 1$. The formula derived is recursive and can be applied to layers further from the output layer.

$$\frac{\delta E}{\delta W^{J-1}} \text{ where } E(W^{J-1}) = \frac{1}{2}(Y - f(f(X^{J-2}W^{J-1})W^J))^2 \quad (5.12)$$

Using the chain rule

$$\frac{\delta E}{\delta W^{J-1}} = \frac{\delta E}{\delta X^J} \frac{\delta f(X^{J-1}W^J)}{\delta X^{J-1}W^J} \frac{\delta X^{J-1}W^J}{\delta X^{J-2}W^{J-1}} \frac{\delta X^{J-2}W^{J-1}}{\delta W^{J-1}} \quad (5.13)$$

$$\frac{\delta E}{\delta W^{J-1}} = \frac{\delta E}{\delta X^J} \frac{\delta f(X^{J-1}W^J)}{\delta X^{J-1}W^J} W^J X^{J-1} (1 - X^{J-1}) X^{J-2} \quad (5.14)$$

where $\frac{\delta E}{\delta X^J} \frac{\delta f(X^{J-1}W^J)}{\delta X^{J-1}W^J}$ was calculated when determining the gradient of f as a function of W^J . This recursiveness where the error gradient at a layer is reliant on the next layer closer to the output leads to the following equation for all layers except the final:

$$\forall j \neq J : \frac{\delta E}{\delta W^j} = \frac{\delta E}{\delta W^{j+1}} \frac{1}{X^j} W^{j+1} X^j (1 - X^j) X^{j-1} \quad (5.15)$$

Thus the process of back-propagation is named because the error is propagated in reverse from the output to the earlier layers. An intuitive understanding of the process is that at a given layer, j , the error is a function of the next layers error, $\frac{\delta E}{\delta W^{j+1}}$ that is then distributed according to W^{j+1} . Therefore the larger the connections, or weights in W^{j+1} , the more the error is attributed to the connecting nodes.

Back-propagation in Convolutional layers is mathematically the same, however the smaller weight filters affect a large number of nodes, and the data associated, X^{j-1} is a smaller, local area for the error at each node. Therefore the gradient of convolutional layer weights is averaged across all of the local data \rightarrow node pairings.

The simplified process for the optimization of parameters in CNN can be found in the pseudo code below.

5.2 Application of Convolutional Neural Networks to GPR Data

This research seeks to compare the performance of a convolutional neural net in learning discriminative GPR features with features that have been hand-tuned to

Data: Input GPR Patch
Result: Optimized filter weights and fully connected layer weights initialization;
while *Stopping criteria not met* **do**
 Get random data sample;
 Generate output based on current filters and weights;
 Calculate error between output and training label truth;
 Calculate the derivative of the error w.r.t. the weights for each layer using the error function from the layer closer to the output;
 Update each weight based on the derivative of the error function at each weight and the desired learning rate of the network;
end

Algorithm 1: General overview for updating the weights in a CNN

perform well in visual image tasks. Since hand-tuned feature extraction methods have typically been tested on extracting data from 2-D GPR slices [17, 69] this research will similarly utilize 2-D patches of data as the dataset to compare performance. As the dataset size in this GPR application is limited, a smaller network with few parameters is used to aid in preventing over-fitting on training set noise.

5.2.1 *Experimental Dataset*

The dataset used for this research is a set of patches extracted from locations flagged by an energy based anomaly detector. There are a total of 5984 alarms, with 2395 non-target anomalies and 3589 alarms over target locations. In this section, the dataset is split into a training set and a testing set where testing emplacements are not found in the training data. There are a total of 3600 alarms in the training set with 2179 target observations. The test set consists of 1771 alarms with 1075 target observations. Performance will be measured by a methods ability to discriminate between patches extracted from a target alarm and a non-target anomaly alarm. An ROC curve will be used as a method to describe the performance of a given method of discrimination. The area under the ROC curve (AUC) will be used as a metric when comparing a number of methods simultaneously.

To determine if a CNN has been sufficiently trained, a validation set from within the training set will be withheld during training. Training will continue until performance on the validation set declines for a number of iterations. The CNN state that produced the highest score on the validation is utilized as the final classifier.

5.2.2 CNN Layer Parameters

The size of filters used in GPR data is highly dependent on the sampling rate of individual sensors both temporally and spatially. GPR data, unlike visual images, does not have issues with target signature scale variance as long as the same sensor properties are maintained. Therefore, filter and pooling sizes must be optimized on a particular sensor. For this research, the filter size used is a 7x7 pixel area as it performed best over a tested range. For the aggregation size, in order to compare closely with the aggregation performed in the HOG descriptor, a 6x6 pixel area was used. As mentioned in Section 5.1, the max pooling, or taking the max value to represent each aggregation area was used.

In designing the neural network for use in GPR data, the use of both down-track and cross-track images is important. It is for this reason that a split convolutional layer is used. The sampling rate in down-track and cross-track differs, and therefore the filters for each should not be the same. Therefore this research uses two banks of convolutional filters where the output of each is joined in at the first fully connected layer. A diagram is shown in Figure 5.3 that visualizes this setup.

5.3 Results

The section presents the results of applying simple convolutional neural networks to differentiate between target and non-target pre-screener alarms. Due to the potential flexibility and iterative training methods used to train neural networks a number of challenges exist in achieving good results. As the space of possible neural network

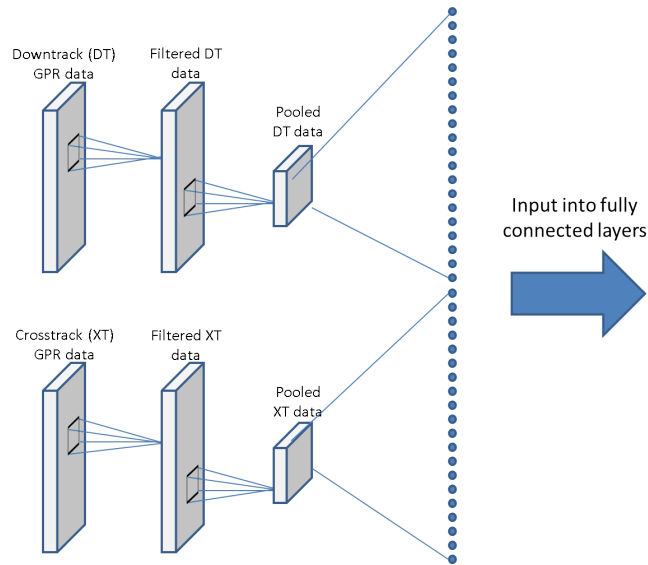


FIGURE 5.3: This Figure shows the architecture used for the convolutional layers of the CNN. Due to the differences in spatial sampling, downtrack and crosstrack data need to have independent filters. After the convolutional layer the outputs are joined when supplied to the fully connected layers

configurations is much too large to exhaustively search through all parameters, this section will focus on certain aspects that can greatly affect neural network performance on GPR data. In all results presented here, the CNN performance shown is the average from 10 independent trials to account for the random seeding of the neural network.

Over-fitting

One of the principal challenges in training a neural network is to avoid what is referred to as "over-fitting" of a neural network. Given enough training iterations, if a neural network has enough tunable parameters it will often be able to classify the training set perfectly. However, in many cases this leads to undesirable performance as the neural network may be fitting to the specific noise present in the training observations. This leads to poor generalization in the classification of test set observations. This over-fitting to the training dataset can be avoided using a number of techniques. The

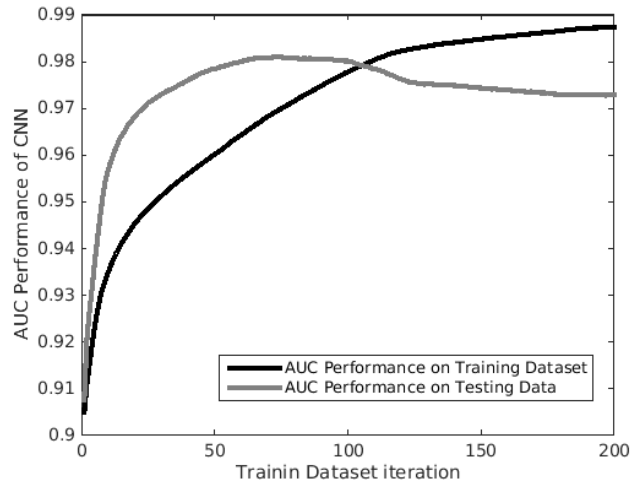


FIGURE 5.4: This figure shows the training of a CNN where test-set performance begins to decline even though performance on the training set is still increasing. If only the training set is known during training, it is very difficult to determine when test set performance is optimized. This over-fitting of the training data is avoided in this research by using a validation set, a portion of the training data that is held out to determine where performance peaks.

simplest and most obvious is to control the number of tunable parameters in the network. This work utilizes a relatively small network size as the number of training observations is limited compared to recent CNN applications [70].

Despite small network size, over training can still remain a concern in this application of CNNs. Figure 5.4 shows the test set performance measured by AUC for one fold of the dataset by the number of training iterations over the entire dataset. As the network becomes more finely tuned to perform well on the training data, it loses its ability to generalize on the held out test set. It is for this reason that a validation set from the training set is held out as a method to determine when performance peaks on an unseen dataset.

Number of Filters

Generally, an important factor in CNN performance is the number of filters used. In this single layer network however, only a few filters were required to obtain fairly high

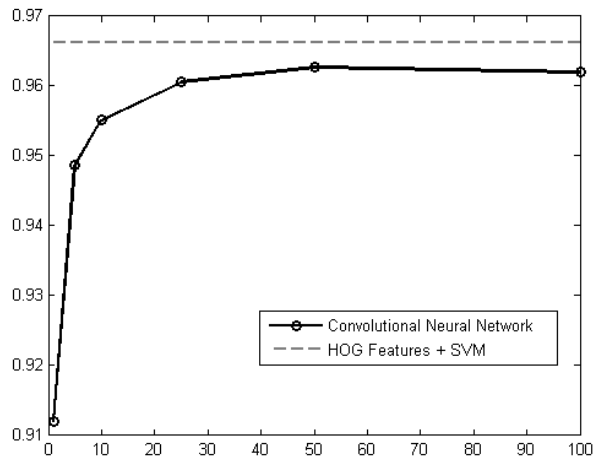


FIGURE 5.5: This figure shows the performance of a CNN as the number of filters are varied. Performance of networks with relatively few filters is very good, showing the discrimination ability with a single filter of above 90%. The dotted line shows the performance when HOG features are used along with a non-linear SVM to classify the same patch pairs. In general the performance of the CNN is slightly lower than that achieved with the HOG features.

performance. Figure 5.5 shows the AUC performance as a function of the number of filters in a given network configuration. This work has found that only about 25 filters are necessary before performance begins to level out. This implies that discriminative characteristics are fairly consistent both between observations and within each patch, hence only a limited number of filters are necessary for feature extraction.

5.3.1 Alternative Classifiers

The fully connected layers in the CNN are necessary to train the convolutional filters by the back-propagation of error. However, the learned filters can then be utilized as a method of feature extraction where the output of the convolutional layers is input into a standard statistical classifier. As this is a binary classification problem, a wide variety of statistical classifiers can be easily used. It was found in this research that a statistical classifier was able to outperform the fully connected layers in all cases

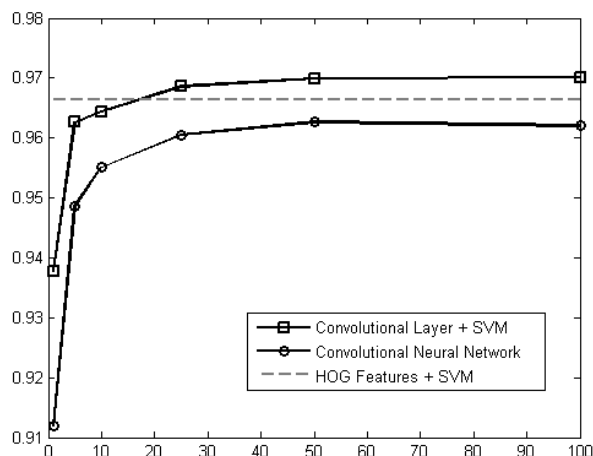


FIGURE 5.6: This figure shows the performance of features extracted using the CNN convolutional layers. The features extracted are the output from the convolutional layer of the CNN. These features from the training data are used to train a non-linear SVM classifier. The extracted features from the test dataset are then classified to calculate a confidence for each pair of patches. The resulting performance of this system is consistently higher than that achieved using a basic CNN. In addition, when enough filters are utilized, these filters out perform the HOG descriptors when the same classifier is used.

tested. Figure 5.6 shows the performance of learned features from a CNN used to extract features that are then classified by a non-linear SVM. This is compared to the performance of the fully connected layers as well as HOG features along with the same type of classifier. The figure shows that the use of CNN features outperforms the use of HOG features when the classification method is consistent. These results show that features learned specifically for GPR can outperform hand designed features in discrimination tasks.

5.4 Conclusions

The results presented in this chapter showed that the use of CNNs can produce features that perform consistently better than state of the art features that are adapted from the image processing field. It is noted, however, that the results using

the fully connected layers did not perform as well. In particular, with this network configuration, the use of an alternative classifier produces a consistent improvement in classification performance over using the fully connected layers across all trials.

Conclusions and Future Work

6.1 Implementation of Methods Developed for Image Processing

Section 3.1 presents the implementation and evaluation of three different visual image detection systems. These systems, while designed for use on visual images were able to extract information useful to the GPR threat detection system. Instance matching, object detection, and image classification methods were all evaluated for their ability to differentiate between buried threat and pre-screener false alarm. The framework found to provide the best detection performance was based off of object detection methods.

Instance matching methods had the poorest performance across all frameworks tested. Challenges using this method stemmed from the lack of consistency in threat responses between various soil conditions. Changes in subsurface conditions caused errors in the localization of feature points that prevented matching between images. In addition, as these methods are primarily used with a single positive example they are not designed to be used with a statistical classifier and often rely on heuristic matching techniques.

The image classification method tested was designed to classify an image based upon the entire contents of the image. One of the benefits of this model was that training examples did not require labeling the specific buried response. Patch frequencies over the entire B-scan were used to represent the GPR data at each pre-screener alarm. Problems with this method were often caused by correlated sub-surface clutter. Though this clutter did not have a hyperbolic shape, many of the regions around the image appeared similar to regions found on a hyperbolic shape. As the order of these patch regions was not preserved in the representation of the B-scan, they appeared very similar to buried threats and were given a high confidence.

The highest performing framework, object detection, relied simply on a training set of positive and negative image regions extracted from the training data. These image regions are then processed into a feature vector. A statistical classifier is used to calculate a decision boundary within the feature space. During testing, B-scans from pre-screener alarms are sampled regularly. At each sampled location, the image region is extracted then represented by a feature vector. The classifier calculates a confidence for each sampled location based upon the decision boundary learned in training. The highest confidence for a given pre-screener alarm is then kept to represent the pre-screener alarm.

While the object detection method's performance was much higher than all other methods, performance varied based upon a number of choices. The feature extractor used, the classifier used, and the size of the image region all had a significant effect on performance. In an effort to understand the variation, a study was designed to evaluate the performance of these factors on a much larger dataset. The results and analysis of this study can be found in Chapter 3.2.

While the performance generated in Chapter 3.1 outperformed current on-board processing methods, the results were generated on a small-scale dataset using only a subset of possible methods. The work presented in Chapter 3.2 was performed to

identify characteristics of high performing feature extractors and the various parameters that provided the most robust performance across a large dataset.

Similar to the results found in Chapter 3.1, the results from these experiments indicate that gradient based methods coupled with a support vector machine (SVM) classifier provided the best pre-screener alarm classification. These methods, specifically SIFT, HOG, and EHD, all operate by aggregating gradients over small local areas by their gradient angles. Other methods that extracted similar information, such as LBP patterns, also performed very well.

6.2 Physics-Based Augmentation

Chapter 4 introduces a number of sources of variation that complicate the training of a statistical classifier to operate in all possible conditions. This chapter focuses on one source of variation that is known to affect the target response, rough ground surface.

Section 4.4 presents research focused on simulating rough ground contours using existing collected data. The methodology of simulation was verified using both simulated and collected data where the ground surface is known. This method was tested by applying these augmentations to improve performance on a specific landmine type given that only one observation is available. It was found that this method improved performance over having only the single observation. The performance reached by the augmentation method was comparable to performance when 15 additional emplacements were added to the training dataset via cross-validation.

Section 4.5 presents experiments performed that augmented a larger variety of GPR alarms. Two different training conditions were tested in this section. The first was the case where the soil type in the training set differed from that in the test set. The second was a large-scale dataset obtained from at a single test site. In both of these cases the classifier that was trained on a dataset that included augmentation

improved detection performance.

The physics-based augmentation method has shown to help in normally challenging situations. This augmentation method can be used to improve the robustness of a system that will be operating in unknown conditions, especially if very little training data of certain target types is available.

6.3 Application of Convolutional Neural Nets to Detection in GPR Data

Chapter 5 seeks to improve on the performance of techniques outlined in Chapter 3. The techniques shown in Chapter 3 were developed for use in visual images in order to be invariant to sources of variation common in these images. Time domain GPR image formation however, is a very different process, and presents different types of variation in the target response data. Convolutional neural nets are applied in this work due to their ability to learn discriminative features from the data.

Using a slightly modified shallow convolutional layer, this work was able to discriminate between target and non-target pre-screener alarm patches better than with the use of a state of the art feature extractor from chapter 3. These results confirm that a feature extraction method tuned to the specific data performs better than a method designed with different invariances in mind.

6.4 Future Work and Alternative Applications

The most promising avenue for future work given these results is the application of larger, deep neural networks to GPR data. These networks are known to be very difficult to train and therefore would require additional data. Given the results from Chapter 4 it is believed that these physics-based augmentation methods could generate additional data that would allow a deeper neural network to be trained. These augmentations would also be necessary as convolutional neural networks are

susceptible to poor/unknown performance due to an inability to generalize properly [71].

Further research into additional sources of GPR response variation is another area that could provide additional robustness to GPR detection systems. In particular, methods to mitigate other potential sources of variation such as vehicle speed, soil/surface inhomogeneities, and subsurface layers could be helpful for current fielded systems.

In addition to finding buried threats, GPR has a number of applications including: concrete analysis [72], ice crevasse detection [73], road condition analysis [74] that could benefit from the use of these techniques when developing an automated detection system. It has been shown that transfer learning using convolutional neural networks trained on alternative datasets is effective in object detection in visual images. It is believed that this could also be used by GPR systems that use similar antenna and spatial resolution that search for different types of objects. This would be especially helpful because many GPR applications have extremely limited data sets because many of the objects are buried within concrete or otherwise inaccessible.

Bibliography

- [1] International Campaign to Ban Landmines, “Landmine monitor 2014,” November 2012.
- [2] Geneva International Centre for Humanitarian Demining, “Mines other than anti-personnel mines,” March 2012.
- [3] Joint Improvised Explosive Device Defeat Organization, “Annual report,” 2010.
- [4] J. MacDonald, “Alternatives for landmine detection,” tech. rep., RAND Corporation, 2003.
- [5] Cambodia Mine Action Centre, “Annual report 2011,” 2011.
- [6] *Improving detection of low-metallic content landmines using EMI data*, vol. 4, 2000.
- [7] S. L. Tantum, W. R. Scott, K. D. Morton, L. M. Collins, and P. A. Torrione, “Target classification and identification using sparse model representations of frequency-domain electromagnetic induction sensor data,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–18, 2012.
- [8] J. Peters, L.P., J. Daniels, and J. Young, “Ground penetrating radar as a subsurface environmental sensing tool,” *Proceedings of the IEEE*, vol. 82, pp. 1802–1822, dec 1994.
- [9] E. Rosen, “Detecting improvised explosive devices,” 2010.
- [10] J. Wilson, P. Gader, W.-H. Lee, H. Frigui, and K. Ho, “A large-scale systematic evaluation of algorithms using ground-penetrating radar for landmine detection and discrimination,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, pp. 2560–2572, aug. 2007.

- [11] K. Ho, L. Carin, P. Gader, and J. Wilson, “An investigation of using the spectral characteristics from ground penetrating radar for landmine/clutter discrimination,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 46, pp. 1177–1191, april 2008.
- [12] P. Gader, M. Mystkowski, and Y. Zhao, “Landmine detection with ground penetrating radar using hidden markov models,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 39, pp. 1231–1244, jun 2001.
- [13] H. Frigui and P. Gader, “Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k-nearest neighbor classifier,” *Fuzzy Systems, IEEE Transactions on*, vol. 17, pp. 185–199, feb. 2009.
- [14] D. Potin, P. Vanheeghe, E. Duflos, and M. Davy, “An abrupt change detection algorithm for buried landmines localization,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, pp. 260–272, feb. 2006.
- [15] S. Lambot, M. Antoine, M. Vanclooster, and E. C. Slob, “Effect of soil roughness on the inversion of off-ground monostatic gpr signal for noninvasive quantification of soil properties,” *Water Resources Research*, vol. 42, no. 3, 2006.
- [16] W. Clark, B. Burns, K. Sherbondy, J. Ralston, and C. Rappaport, “Surface effects on ground penetrating radar imagery,” in *Antennas and Propagation Society International Symposium, 2005 IEEE*, vol. 1, pp. 404–407, IEEE, 2005.
- [17] P. A. Torrione, K. D. Morton, R. Sakaguchi, and L. M. Collins, “Histograms of oriented gradients for landmine detection in ground-penetrating radar data,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 3, pp. 1539–1550, 2014.
- [18] P. Torrione, C. Throckmorton, and L. Collins, “Performance of an adaptive feature-based processor for a wideband ground penetrating radar system,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, pp. 644–658, april 2006.
- [19] K. Gu, J. Li, M. R. Bradley, J. D. Habersat, and G. B. Maksymonko, “Adaptive ground bounce removal,” in *AeroSense 2002*, pp. 719–727, International Society for Optics and Photonics, 2002.
- [20] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

- [21] F. Soldovieri, O. Lopera, and S. Lambot, “Combination of advanced inversion techniques for an accurate target localization via gpr for demining applications,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, pp. 451–461, jan. 2011.
- [22] L.-P. Song and Q. H. Liu, “Ground-penetrating radar land mine imaging: Two-dimensional seismic migration and three-dimensional inverse scattering in layered media,” *Radio Science*, vol. 40, no. 1, pp. n/a–n/a, 2005.
- [23] J. Song, Q. H. Liu, P. Torriero, and L. Collins, “Two-dimensional and three-dimensional nufft migration method for landmine detection using ground-penetrating radar,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, pp. 1462–1469, june 2006.
- [24] T. Counts, A. Gurbuz, W. Scott, J. McClellan, and K. Kim, “Multistatic ground-penetrating radar experiments,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, pp. 2544–2553, aug. 2007.
- [25] F. Roth, P. van Genderen, and M. Verhaegen, “Convolutional models for buried target characterization with ground penetrating radar,” *Antennas and Propagation, IEEE Transactions on*, vol. 53, no. 11, pp. 3799–3810, 2005.
- [26] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, feb 1989.
- [27] P. Scheet and M. Stephens, “A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase,” *The American Journal of Human Genetics*, vol. 78, no. 4, pp. 629–644, 2006.
- [28] M. Mohamed and P. Gader, “Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 548–554, may 1996.
- [29] P. Salembier, T. Sikora, and B. Manjunath, *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons, Inc., 2002.
- [30] D. K. Park, Y. S. Jeon, and C. S. Won, “Efficient use of local edge histogram descriptor,” in *Proceedings of the 2000 ACM workshops on Multimedia, MULTIMEDIA ’00*, (New York, NY, USA), pp. 51–54, ACM, 2000.

- [31] S. Rahman, S. Naim, A. Al Farooq, and M. Islam, “Performance of mpeg-7 edge histogram descriptor in face recognition using principal component analysis,” in *Computer and Information Technology (ICCIT), 2010 13th International Conference on*, pp. 476–481, dec. 2010.
- [32] P. D. Gader, R. Grandhi, W.-H. Lee, J. N. Wilson, and D. K. Ho, “Feature analysis for the niitek ground-penetrating radar using order-weighted averaging operators for landmine detection,” in *Proceedings of SPIE*, vol. 5415, pp. 953–962, 2004.
- [33] R. Szeliski, *Computer Vision: Algorithms and Applications*. Texts in Computer Science, Springer London, 2011.
- [34] L. G. Brown, “A survey of image registration techniques,” *ACM Comput. Surv.*, vol. 24, pp. 325–376, Dec. 1992.
- [35] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision*, vol. 65, pp. 43–72, 2005.
- [36] C. Harris and M. Stephens, “A combined corner and edge detector.,” in *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988.
- [37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [38] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer Berlin / Heidelberg, 2006.
- [39] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, pp. 79–116, 1998.
- [40] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *Computer Vision ECCV 2010* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), vol. 6314 of *Lecture Notes in Computer Science*, pp. 778–792, Springer Berlin / Heidelberg, 2010.
- [41] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, pp. 1615–1630, oct. 2005.

- [42] H. P. Moravec, *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford, CA, USA, 1980. AAI8024717.
- [43] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.
- [44] C. Papageorgiou and T. Poggio, “A trainable system for object detection,” *International Journal of Computer Vision*, vol. 38, pp. 15–33, 2000.
- [45] V. Vapnik, *The nature of statistical learning theory*. springer, 1999.
- [46] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proceedings of the 23rd international conference on Machine learning, ICML '06*, (New York, NY, USA), pp. 161–168, ACM, 2006.
- [47] H. Wold, “Estimation of principal components and related models by iterative least squares,” in *Multivariate Analysis* (P. Krishnaiah, ed.), New York: Academic Press, 1966.
- [48] M. Barker and W. Rayens, “Partial least squares for discrimination,” *Journal of Chemometrics*, vol. 17, no. 3, pp. 166–173, 2003.
- [49] S. de Jong, “Simpls: an alternative approach to partial least squares regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, 1993.
- [50] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [51] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [52] P. Torrione, S. Keene, and K. Morton, *PRT: The Pattern Recognition Toolbox for MATLAB*, 2011. Software available at <http://newfolderconsulting.com/prt>.
- [53] C.-W. Hsu, C.-C. Chang, C.-J. Lin, *et al.*, “A practical guide to support vector classification,” 2003.
- [54] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.

- [55] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.
- [56] A. Coates, H. Lee, and A. Y. Ng, “An analysis of single-layer networks in unsupervised feature learning,” *Ann Arbor*, vol. 1001, p. 48109, 2010.
- [57] A. Coates and A. Ng, “Learning feature representations with k-means,” *Neural Networks: Tricks of the Trade*, pp. 561–580, 2012.
- [58] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [59] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen, “Texture discrimination with multidimensional distributions of signed gray-level differences,” *Pattern Recognition*, vol. 34, no. 3, pp. 727–739, 2001.
- [60] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, “Face detection based on multi-block lbp representation,” in *Advances in Biometrics*, pp. 11–18, Springer, 2007.
- [61] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [62] P. Torrione, K. Morton, R. Sakaguchi, and L. Collins, “Histogram of gradient features for buried threat detection in ground penetrating radar data,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pp. 3182–3185, july 2012.
- [63] X. Wang, T. X. Han, and S. Yan, “An hog-lbp human detector with partial occlusion handling,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 32–39, IEEE, 2009.
- [64] S. E. Yuksel, J. Bolton, and P. Gader, “Multiple-instance hidden markov models with applications to landmine detection,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 53, no. 12, pp. 6766–6775, 2015.
- [65] S. Hubbard, J. Peterson Jr, E. Majer, P. Zawislanski, K. Williams, J. Roberts, and F. Wobber, “Estimation of permeable pathways and water content using tomographic radar data,” *The Leading Edge*, vol. 16, no. 11, pp. 1623–1630, 1997.

- [66] A. Giannopoulos, “Modelling ground penetrating radar by gprmax,” *Construction and building materials*, vol. 19, no. 10, pp. 755–762, 2005.
- [67] P. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis.,” in *ICDAR*, vol. 3, pp. 958–962, 2003.
- [68] R. T. Sakaguchi, K. D. Morton, L. M. Collins, and P. A. Torrione, “Physics-based deformations of ground penetrating radar signals to improve the detection of buried explosives,” in *SPIE Defense+ Security*, pp. 90720P–90720P, International Society for Optics and Photonics, 2014.
- [69] R. Sakaguchi, “Image processing methods applied to landmine detection in ground penetrating radar,” Master’s thesis, Duke University, 2013.
- [70] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, pp. 1106–1114, 2012.
- [71] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.
- [72] C. Maierhofer, “Nondestructive evaluation of concrete infrastructure with ground penetrating radar,” *Journal of Materials in Civil Engineering*, vol. 15, no. 3, pp. 287–297, 2003.
- [73] A. Delaney and S. Arcone, “Crevasse detection with gpr across the ross ice shelf, antarctica,” in *AGU Fall Meeting Abstracts*, vol. 1, p. 1098, 2005.
- [74] T. Saarenketo and T. Scullion, “Road evaluation with ground penetrating radar,” *Journal of applied geophysics*, vol. 43, no. 2, pp. 119–138, 2000.

Biography

Rayn Terin Tatsuma Sakaguchi was born on August 30th, 1984 in Honolulu, Hawaii. He was raised in Hawaii Kai and graduated from Punahou School in 2003. He then travelled to Hartford, CT to obtain a B.S. in general engineering from Trinity College in 2007. Following his undergraduate degree he worked as a supplier quality engineer for Sikorsky Aircraft until deciding to go back to school in 2010. In the fall of 2010 he began his graduate studies under the guidance of Prof. Leslie M. Collins at Duke University in Durham, NC. In 2013 he obtained his M.S. degree in electrical and computer engineering from Duke University. His M.S. thesis was entitled "Image Processing Methods Applied to Landmine Detection in Ground Penetrating Radar" [69]. His related dissertation, "Using Image Processing Methods to Improve the Detection of Buried Explosive Threats in GPR Data" was presented in 2016 also under the supervision of Prof. Leslie M. Collins.

Rayn's research interests include computer vision, machine learning and digital signal processing. Following his studies he will be working for CoVar Technologies in Durham, North Carolina.