Landmine Detection Efforts during Summer 2007

Seniha Esen Yuksel*

August 13, 2007

Abstract

This paper presents the summer 2007 efforts for the landmine detection project. The efforts have been concentrated around understanding the basics of landmining, executing the current algorithms and defining the current problems with these current algorithms with some emphasis on improving them. For this purpose, I started the summer semester with some image processing on the GPR data. The image processing techniques mostly included the implementation of the anisotropic diffusion filter and frequency based filtering techniques. These algorithms were tested on the baseline hidden Markov Models (HMM); and didn't show any significant improvement simply on the testing. The idea that they could actually give a better output if the image processing techniques were applied to the training data, made us progress into the continuous density HMM code. With the frequency domain image filters, I tried to get rid of the EMI noise, and was actually successful in doing so with a median filter. Luckily, I also saw that we were able to get rid of a good bunch of noise by taking the mean in the down track direction. After understanding the CDHMM code, I tried to improve it with batch training and receiver operating characteristic (ROC) training. The result of the training is open to improvements.

INDEX TERMS: Landmine detection, CDHMM, ROCA training, Batch training, Anisotropic Diffusion Filter.

*Department of Computer and Information Science and Engineering, University of Florida

Contents

1 INTRODUCTION

Over thirty-nine countries – mostly spread into Africa, Asia, Europe, Middle East and South America – suffer from the threat of currently buried 60 million landmines; around 26000 people a year are wounded or killed by landmines; yet there are around 35 countries that are still producing mines; and only about twenty-five nations renounced or prohibited the use of mines [2, 6]. With the production of more modern mines, the conventional mine detection techniques; namely using metal detectors and probing (pushing a probe into the ground and lifting gently), needed to get supplemented with more modern detection techniques since probing can initiate some of the mines, and metal detectors may miss the more modern mines with plastic casings and very low metal content.With the initiative of the United States in 1997 to start a humanitarian demining effort to eliminate the threat of landmines by 2010 [1], long range detection from airborne platforms and the short-range imaging detection using Ground-Penetrating Radars (GPR) have found considerable interest. The long range methods proved to be useful to identify safe areas to start a mine search; whereas the ground penetrating radars, mostly vehicle-mounted; operate between the ranges 0.3 to 2GHz, can detect individual landmines.

Fig.1 shows a vehicle mounted GPR. As the vehicle moves, the GPR system radiates short pulses of electromagnetic energy from its transmitting antenna; and receives back the reflected energy with its receiver antenna. If the EM waves hit a surface with a different dielectric constant, the waves that are collected back give a hint of the underlying object. As shown in Fig.1, landmines appear as shapes similar to a hyperbola in time-domain GPR signals.

So researches have been trying to detect these hyperbolic shapes with image processing and machine learning steps, however; it is not very easy to detect the mines considering the facts that, (i) the permittivity and hence the output of the GPR varies with the frequency of the applied field, the type of soil, humidity and temperature; (ii) there is a large variety clutter that can be found in the soil (grass, wood pieces, bottles, nails to name a few) that affects the outcome of the learning steps; and (iii) there are hundreds of different mine types. Hence, research is going on to differentiate the mines from the other clutter and the soil itself; and this report will give our efforts in using the GPR data to successfully detect the mines.

In Figures [3 - 9], examples of low metal anti personal mines, high metal anti personal mines, high metal anti tank mines, low metal anti tank mines, metallic clutter and GPR clutter are shown respectively. It can be observed that with bigger mines such as anti-tank mines, the hyperbolic signatures become more significant. On the other hand, clutter may have very similar signatures to mine.

2 PREPROCESSING

In many cases, vehicle-mounted system cannot maintain the radar antenna at a fixed distance above the ground, and hence the ground should be aligned before

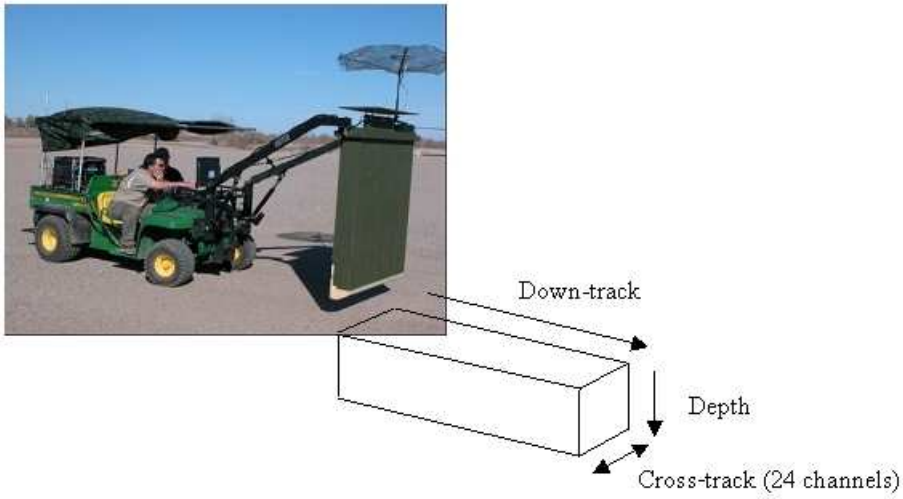


Figure 1: A GPR mounted vehicle showing the time doming GPR signal collection. Adopted from [4].

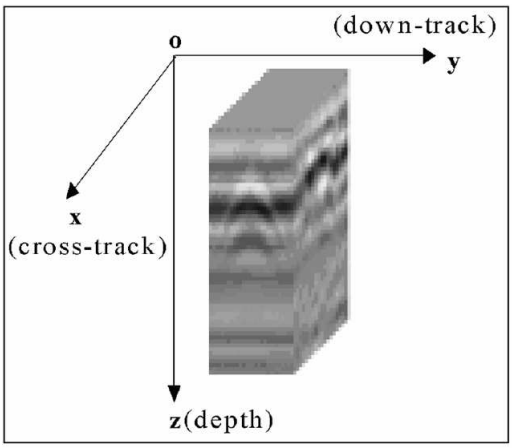


Figure 2: Landmines appear as hyperbolas in time-domain GPR data. Adopted from [3].

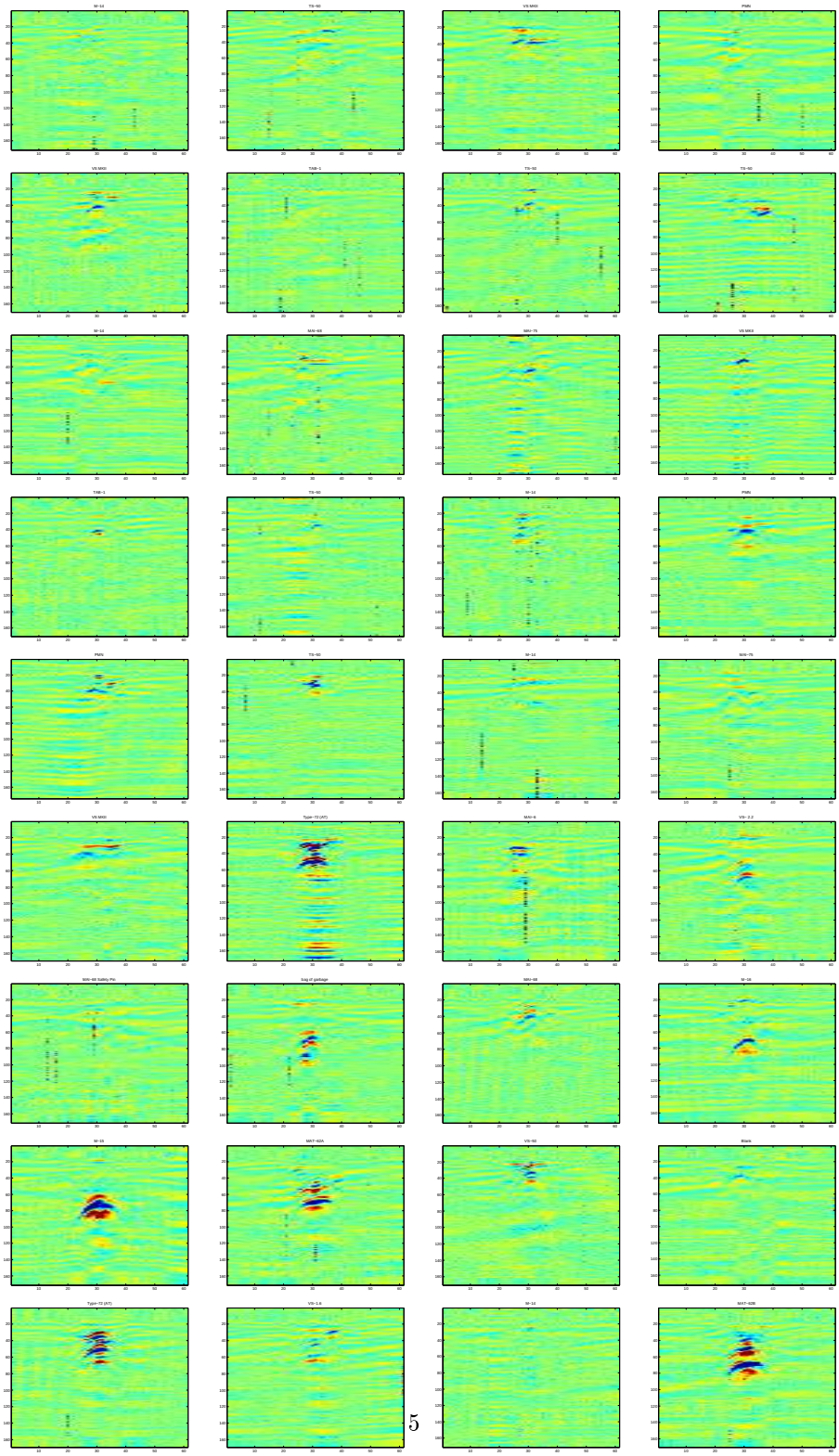


Figure 3: Representatives of Low Metal Anti Personal Mines

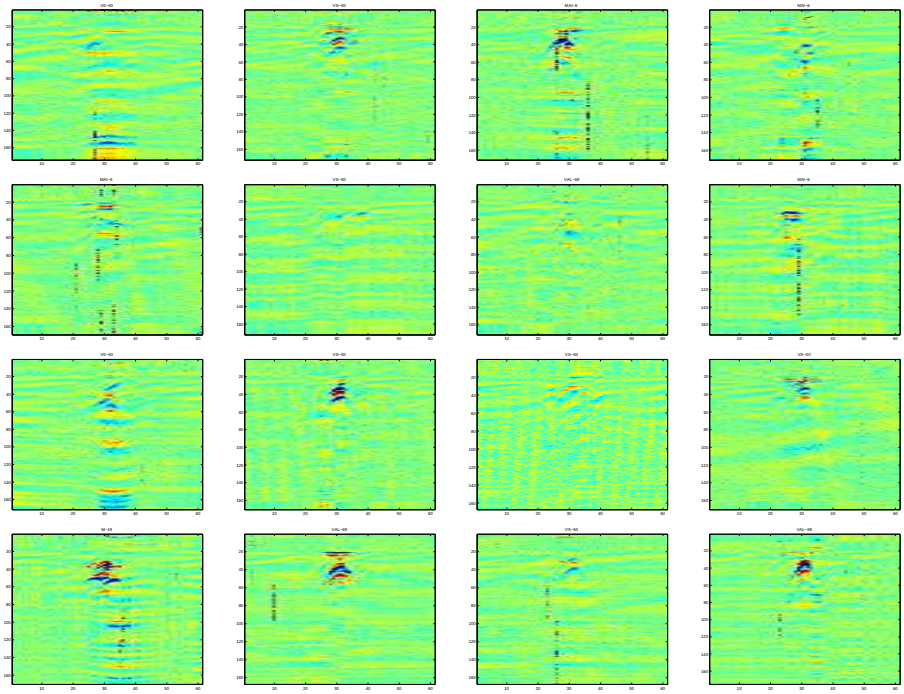


Figure 4: Representatives of High Metal Anti Personal Mines

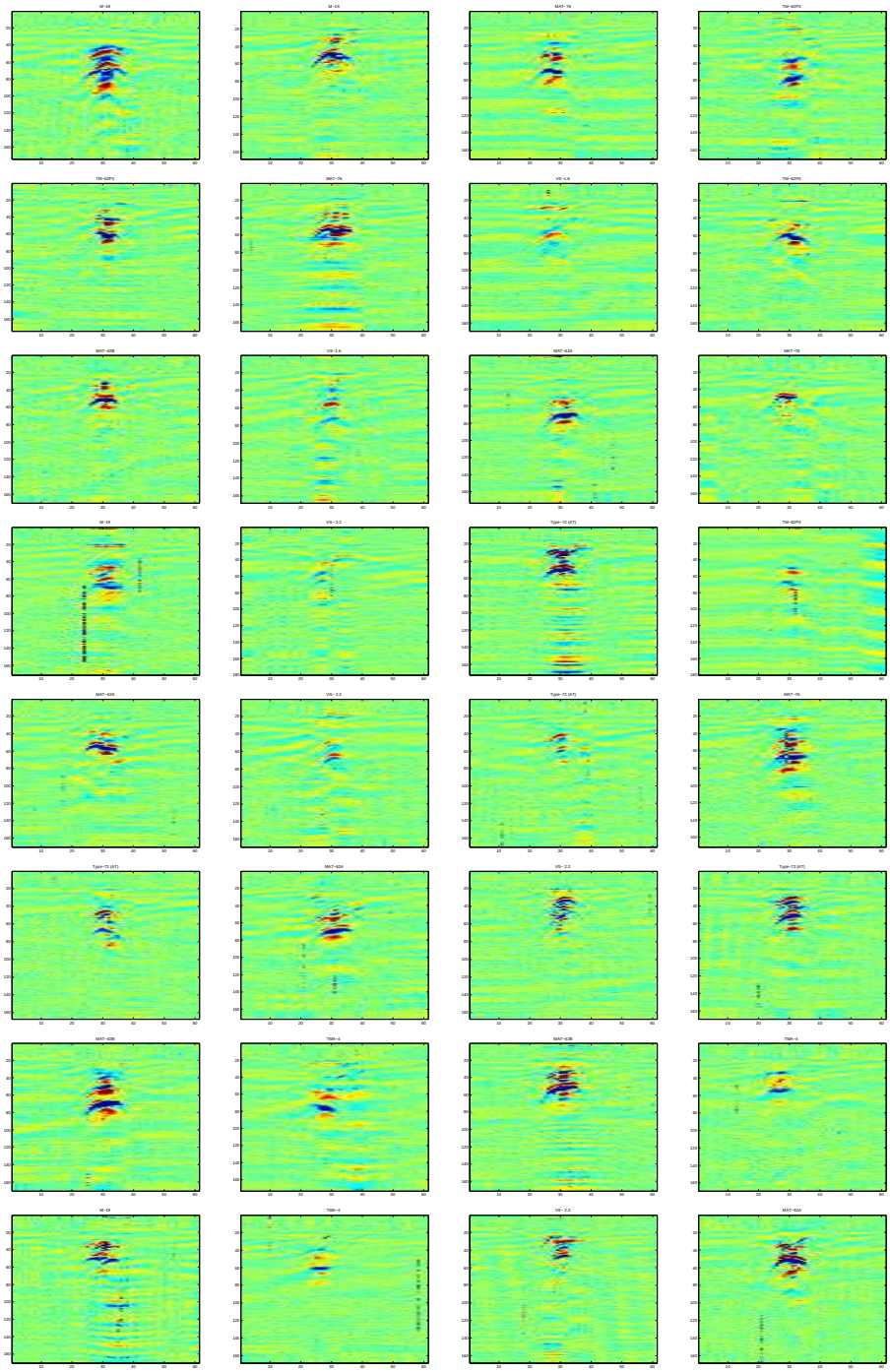


Figure 5: Representatives of Low Metal Anti Tank Mines

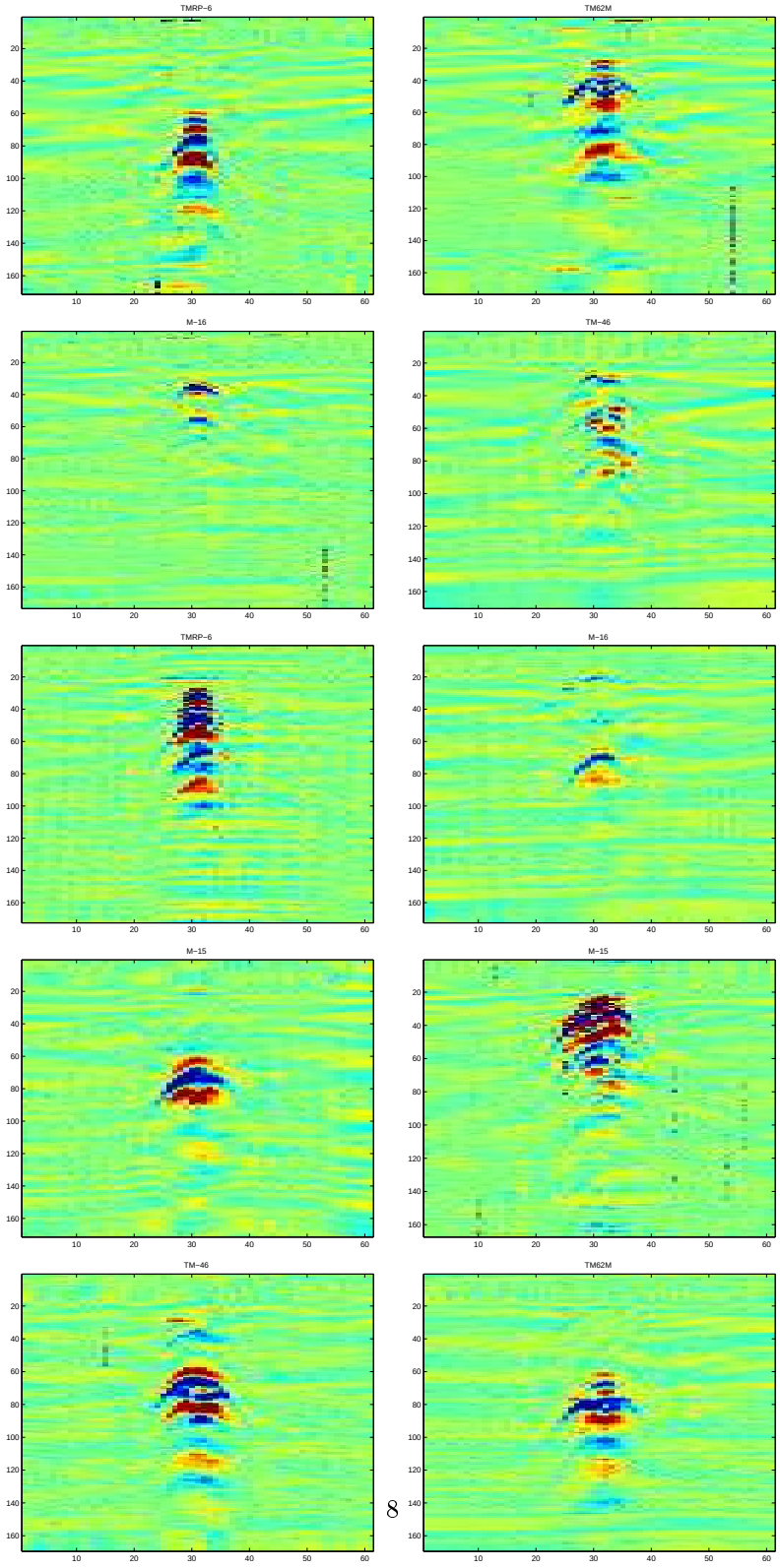


Figure 6: Representatives of High Metal Anti Tank Mines

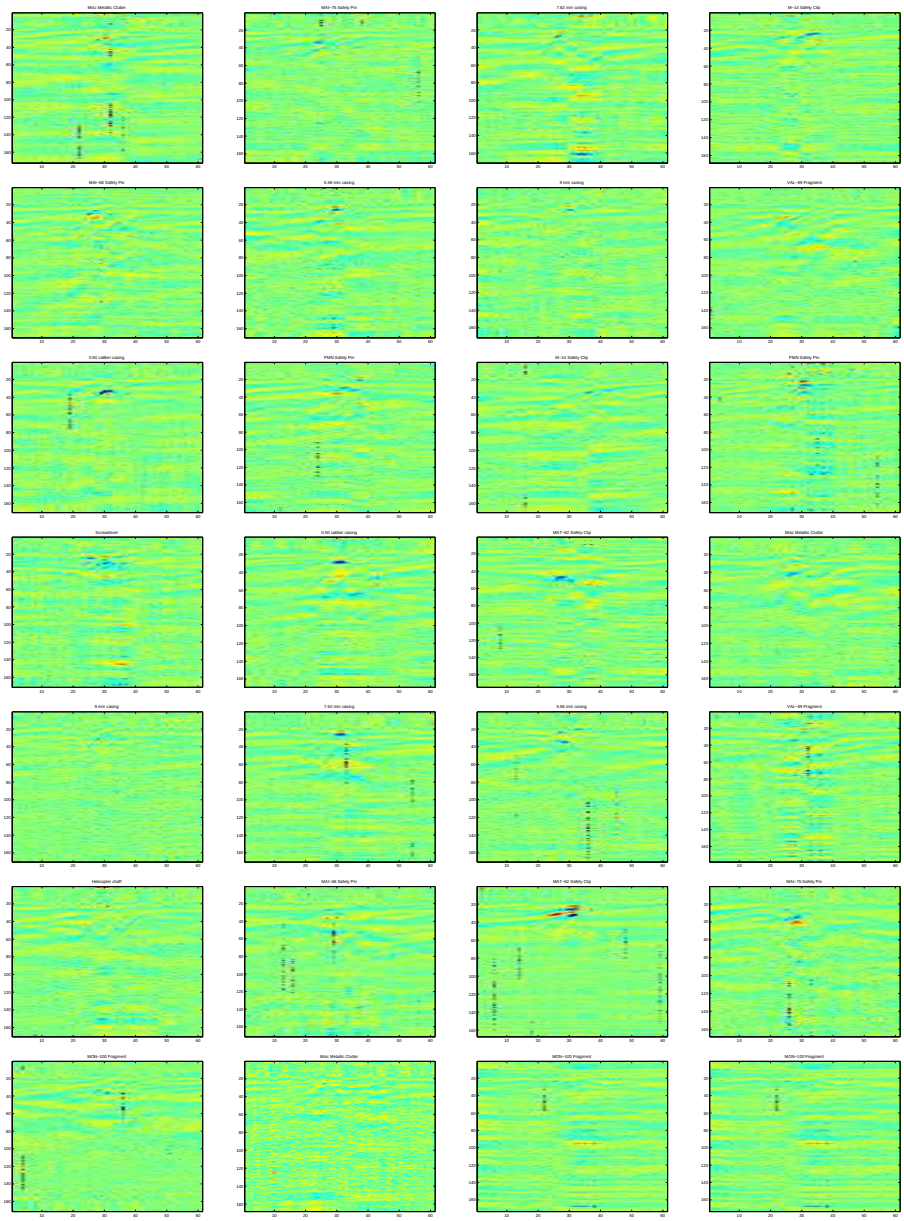


Figure 7: Representatives of Metallic Clutter

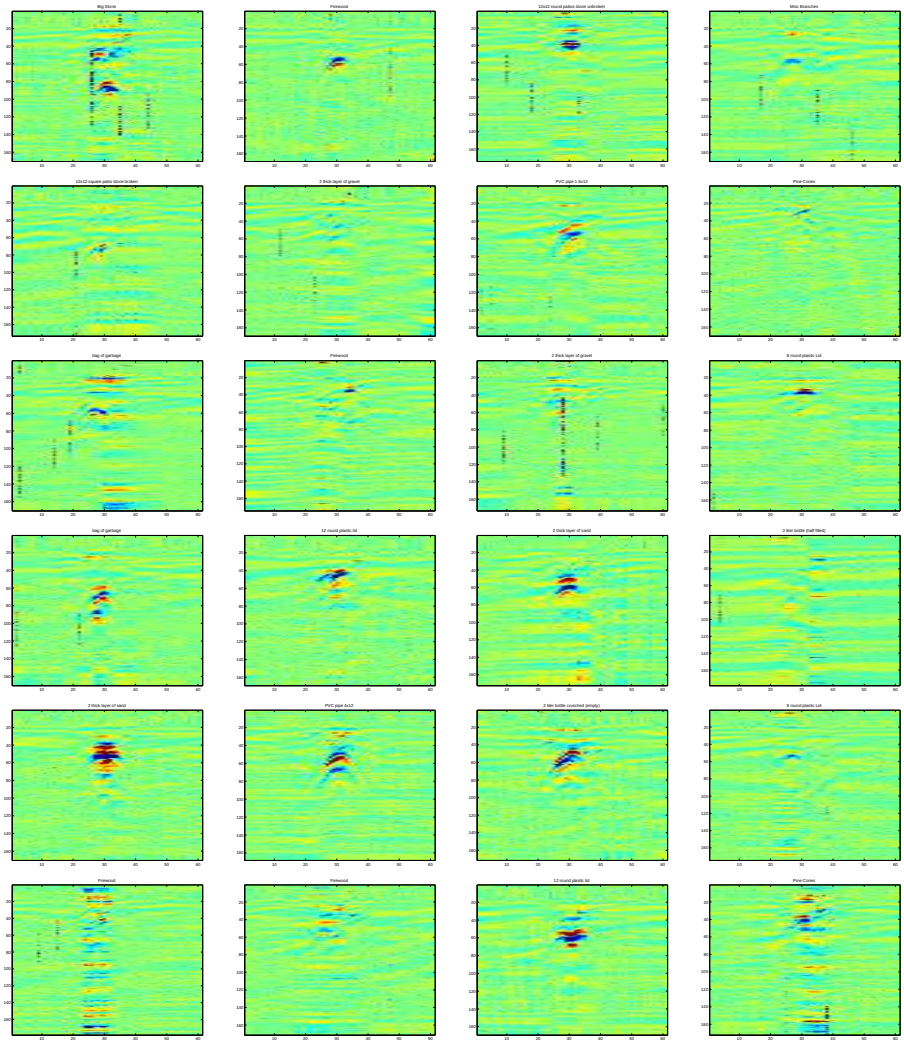


Figure 8: Representatives of GPR Clutter

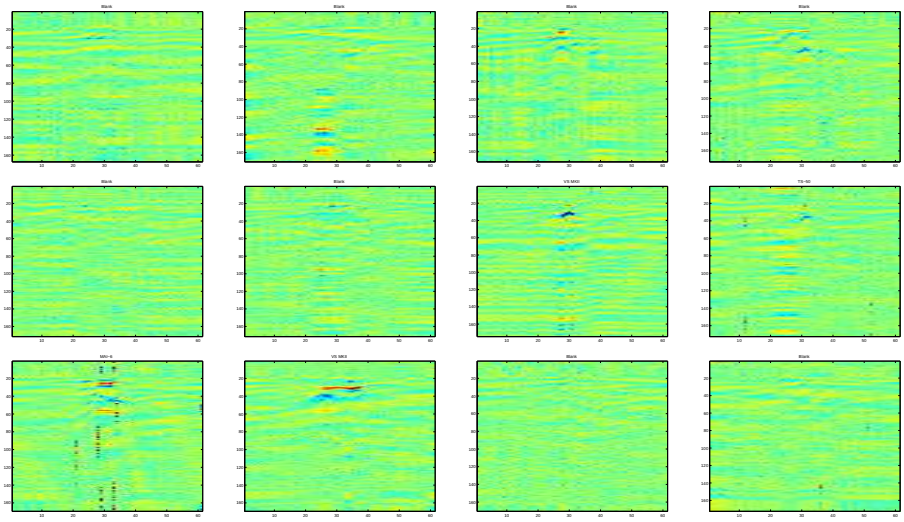


Figure 9: Representatives of Background

further processing. Fortunately, the radar reflection from the ground (ie. ground bounce) dominates the rest of the signal, and it can be used to align the ground position [4]. For this reason, and also to clean the data from the noise and other artifacts, a number of preprocessing steps is necessary. The processing steps will not be described in detail here, but it will be illustrated with the intermediate results, and we hope that it will give an idea to the reader why these steps were needed. In the later sections, we will show our results for trying to improve these processing steps. As seen in Fig. 10, the preprocessing steps get the data from a non-visible condition Fig. 10(a) to a clean output after a couple of steps. The first step is to subtract the mean in the depth direction from the data. This results in Fig. 10(b) is avid of the DC noise from the GPR. In both Fig. 10(a) and (b), it is clear that the ground (the top line) is not aligned. This is because of the ground bounce of the vehicle the GPR is mounted to. To remove this artefact, the haircut algorithm best explained in [4] identifies the peak signals which give the location of the ground bounce, and aligns these peaks as shown in Fig. 10(c). Later, the top few samples are discarded, and the remaining data is down-sampled for speed purposes as shown in Fig. 10(d). The data given in the figures was originally $343 \times 24 \times 61$ and got sampled to $171 \times 24 \times 61$ where the structure of the 3D data is (depth x channel x scan). In the last step of preprocessing, the background is removed by normalizing the data in x direction (cross-track) as best explained in [5]. In Fig. 11, the effects of the preprocessing steps are shown on various kinds of mine and clutter; as well as the resultant second derivatives that are used in feature extraction are displayed.

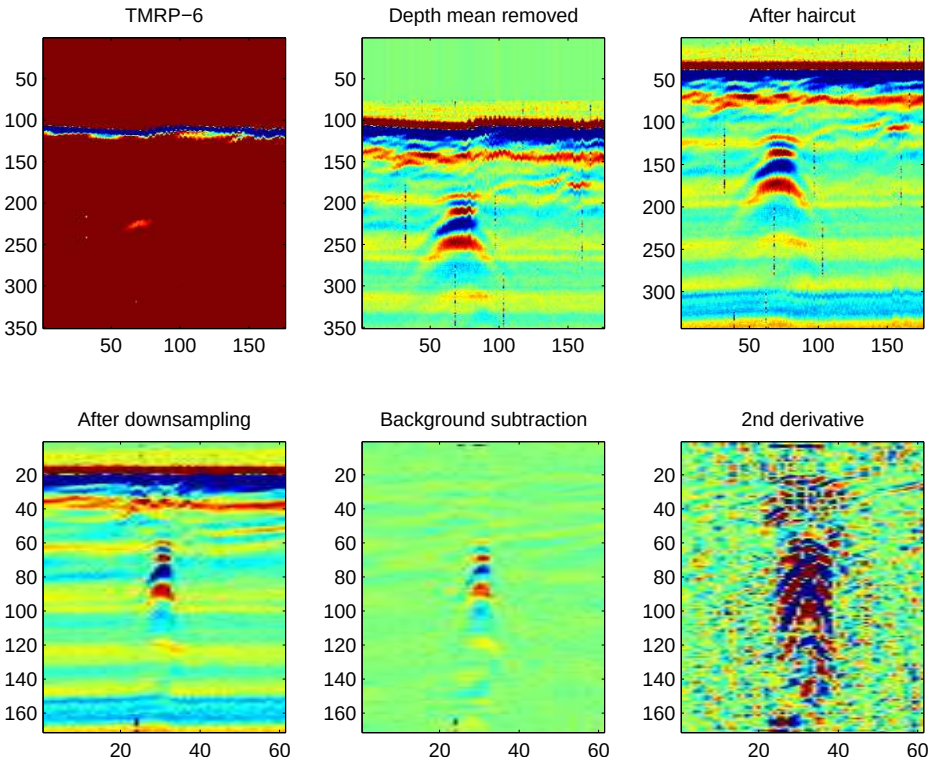


Figure 10: Preprocessing steps from left to right, top to bottom shown on an anti-tank mine signature: The original data (a); The mean in the depth direction is subtracted from the data to delete the GPR noise (b). The ground is aligned by the haircuts algorithm (c). Haircuts removes the bouncing effects of the GPR by first identifying the signal peaks where the ground bounce occurs, then it aligns these peaks. After the haircuts algorithm, the data is down-sampled (d), and normalized (e). The second derivatives can now be computed (f) for feature extraction.

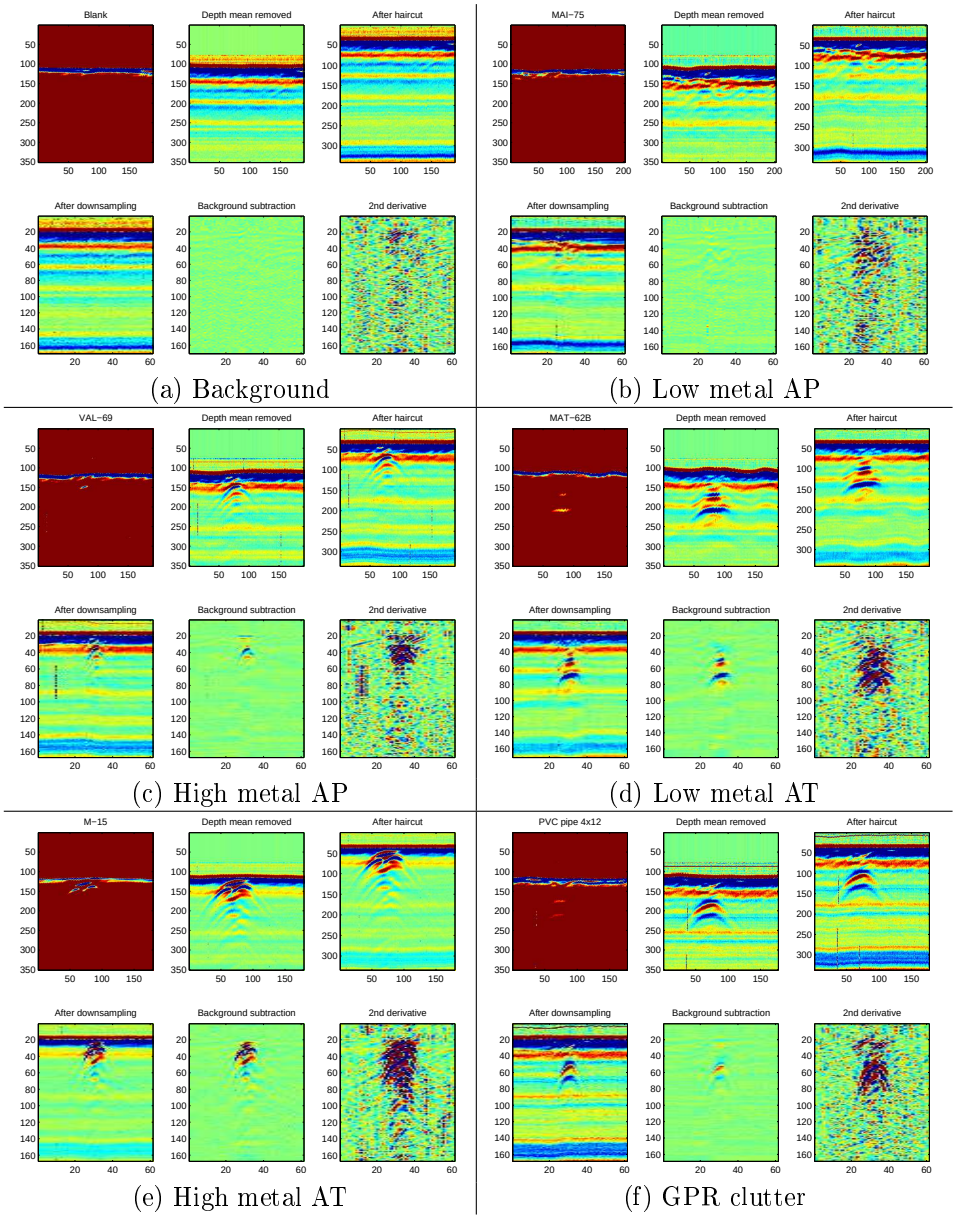
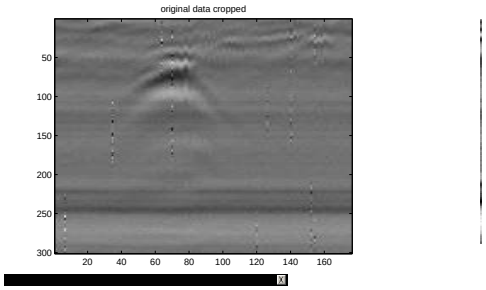


Figure 11: The effects of preprocessing are shown on the different kinds of mine and clutter: Background data in (a); Low metal anti personnel mine in (b); High metal anti personnel mine in (c); Low metal anti tank mine in (d); High metal anti tank mine in (e); GPR clutter in (f).



(a) (b)

Figure 12: The original mine data (a) and a single channel from the mine data which shows strong EMI (b) .

2.1 Frequency Domain Filtering

A careful look at the data showed the existence of EMI (electromagnetic interference) noise in the data as shown in Fig. 12(a). To understand the properties of the EMI noise, I looked at both the time domain and frequency domain characteristics of the data channel by channel as shown in Fig. 12(b). The channel by channel signal display for one mine data is given in Fig. 2.1(a) and (b). Although the EMI noise stands distinct from the background in the time domain signals, it is not occurring at the same place or at the same frequency; but rather it is spreading into various frequencies. To cancel this noise, I tried lowpass filters that would attenuate the high-frequency content of the data that appear as sharp changes as in EMI; the characteristics of the Butterworth lowpass filter used for this purpose is given in Fig. 14(a). I also tried to understand if the EMI falls into a specific frequency; and hence used a Butterworth band reject filter as shown in Fig. 14(b). The experiments with the band reject filter indicated that there is no specific frequency that the EMI is occurring at, and the low-pass filter is actually not a very good idea as it also attenuates the edges which could possibly belong to mines. As a result, I found taking the median to be a very quick and effective way to get rid of the EMI without distorting the image too much.

While playing with the frequencies, the FFT spectrum of the 2D landmine data revealed the existence of a DC frequency in the data. Deleting this frequency (Fig. 15(a)) corresponds to subtracting the mean in the downtrack direction and cleans some of the background noise very well as shown in Fig. 15(c).

2.2 Edge Enhancing Anisotropic Diffusion Filtering

The diffusion equation

$$\partial_t u = div(D \nabla u)$$

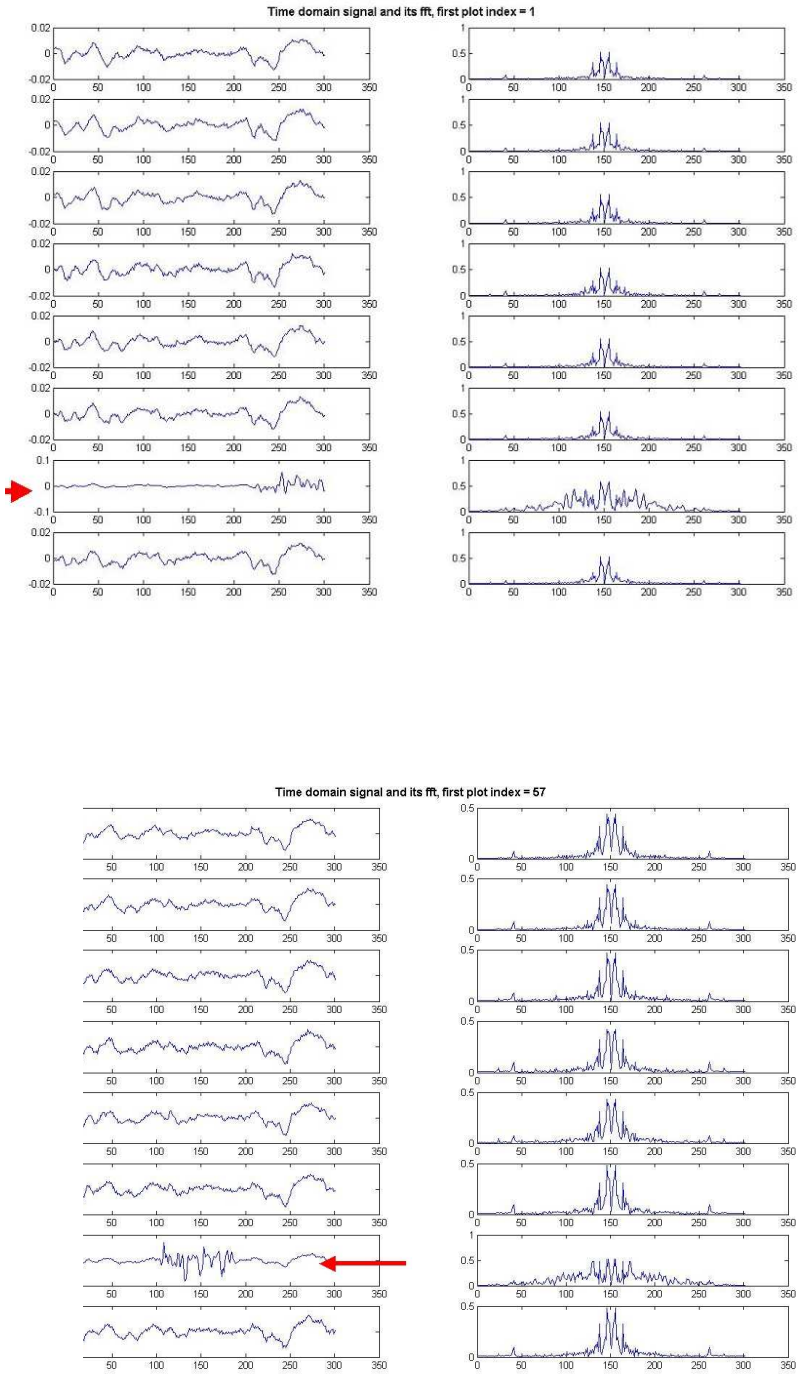
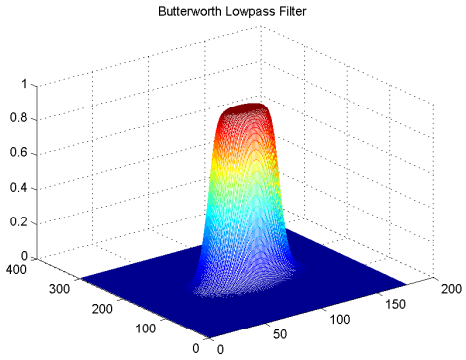
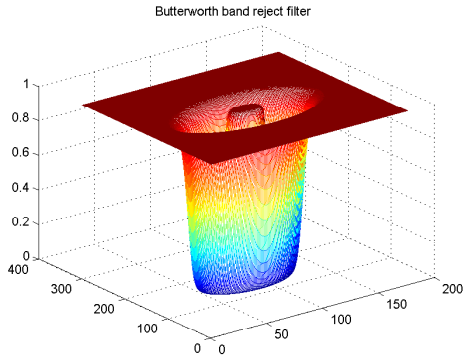


Figure 13: EMI shows a difference in frequency domain, but the signal is not structured and can be spread into various frequencies.



(a) Butterworth Lowpass Filter

$$H(u,v) = \frac{1}{1+\left[\frac{\sqrt{u^2+v^2}}{r_0}\right]^{2n}}$$



(b)Butterworth Band Reject Filter:

$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)*W}{D^2(u,v)-D_0^2}\right]^{2n}}$$

Figure 14: The filter parameters are n: filter order, r_0:cutoff frequency, D_0: the distance from the center of the spectrum to the middle part of the band which is desired to be rejected, W: width of the band, D: the distance of the point (u,v) to the origin of the Fourier Transform cutoff frequency

where the original image is the initial condition $u(x,0) = f(x)$ is used as an edge enhancing filter with a careful selection of the diffusion tensor D, such that the diffusion is preferred along the edges with respect to the diffusion perpendicular to them.Therefore, we construct an orthonormal system of eigenvectors v_1, v_2 of the diffusion tensor D such that

$v_1 \parallel \nabla u_\sigma$ and $v_e \perp \nabla u_\sigma$. So the choice of eigenvalues $\lambda_1 = 1 - exp(\frac{-1}{|\nabla u_\sigma|^2})$ and $\lambda_2 = 1$ solves such a problem. Then the eigenvectors are calculated from

$$v_1 = [\partial u_x \quad \partial u_y]/\sqrt{(\partial^2 u_x + \partial^2 u_y)}$$

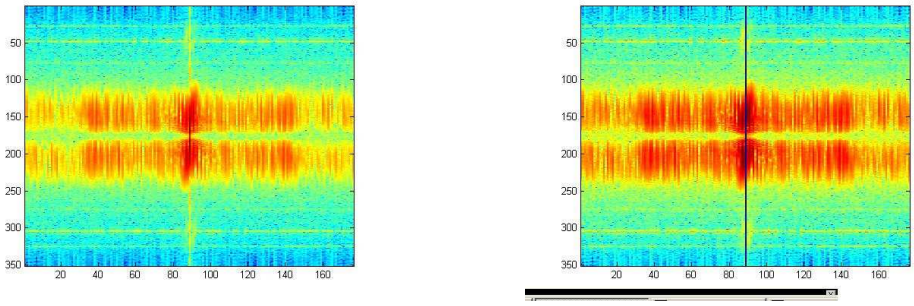
$$v_2 = [-\partial u_y \quad \partial u_x]/\sqrt{(\partial^2 u_x + \partial^2 u_y)}$$

are used to form the diffusion tensor D as:

$$D = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}$$

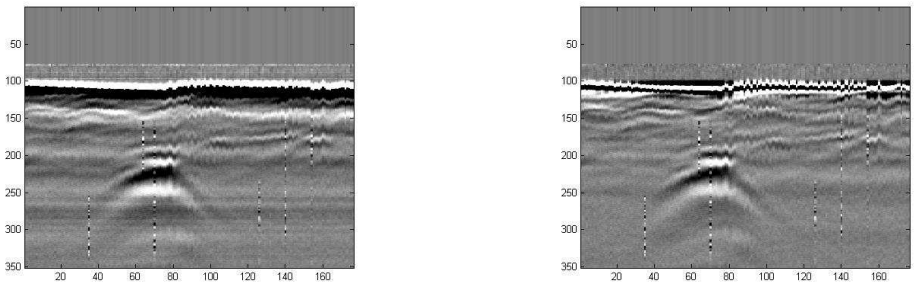
Then the filtered image is found through iterations as:

$$Im_{t+1} = Im_{t+1} + \Delta T * div(D\nabla u)$$



(a) The 2D FFT has a single vertical frequency.

(b) The vertical frequency is filtered.



(c) Original data

(d) Some noise is gone after filtering.

Figure 15: The FFT spectrum of the 2D landmine data revealed the existence of a DC frequency in the data. Deleting this frequency corresponds to subtracting the mean in the down-track direction and cleans some of the background noise very well.

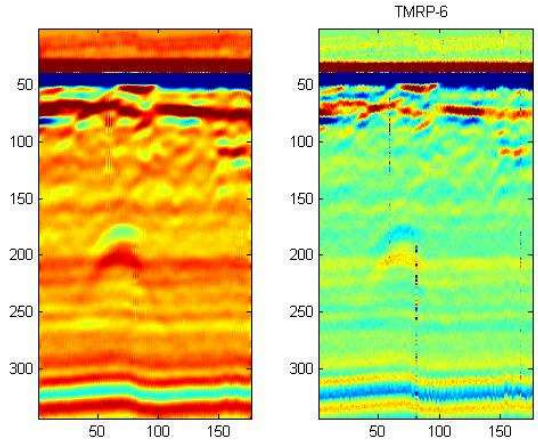
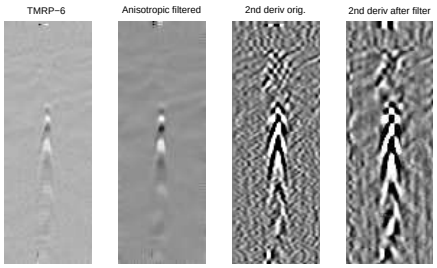


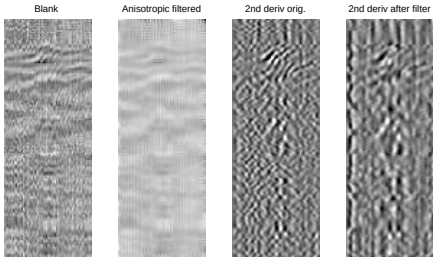
Figure 16: At first, I applied the anisotropic diffusion filter on the data just after haircuts. The original mine on the left is enhanced on the right with anisotropic diffusion filter. For this particular image, the filter parameters are set to $\Delta T = 0.1$, No.Of Iterations = 25, Filter size = [3 3], $\sigma = 0.2$. However, with this approach, the ground is also filtered and the mines close to the ground can get mixed with the ground and get deleted at the background removal process. Hence we thought that it might be a better idea to apply it after the background removal which is shown in the next picture.

At first, I tried the anisotropic diffusion filter on the data just after haircuts as shown in Fig. 16. The original mine on the right is enhanced on the left with anisotropic diffusion filter. For this particular image, the filter parameters are set to $\Delta T = 0.1$, No.Of Iterations = 25, Filter size = [3 3], $\sigma = 0.2$. However, with this approach, the ground is also filtered and the mines close to the ground can get mixed with the ground and get deleted at the background removal process. Hence we thought that it might be a better idea to apply it after the background removal which is shown in the next picture.

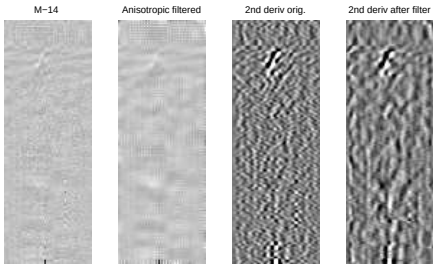
With the application of the anisotropic filter after the background removal, the results were as shown in in Fig. 17. Although there is some enhancement, it is dependent on the parameter selection and did not improve the test results. It might be worth giving a try both on the training and testing data with carefully selected parameters.



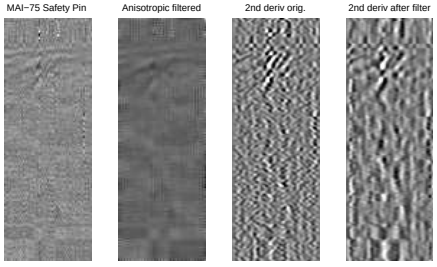
(a) High Metal AT Mine



(b) Blank



(c)Low Metal AP Mine



(d) Metallic Clutter

Figure 17: Anisotropic diffusion results on four different signatures. Displayed are from left to right, the original image, the filtered image, the original second derivatives for feature extraction, the effect of filtering on the second derivatives. Filtering has somewhat reduced the noise, but its parameters should be cleverly tuned so as not to lose the small mines, and not to introduce extra sharpened edges that look like mines.

3 BASELINE HIDDEN MARKOV MODELS

The landmine detection framework introduced in [5] uses the Hidden Markov Models to assign a confidence that a mine is present at a point. The notation for a discrete HMM is given as:

T the length of the observation sequence (total number of time steps)
N number of states in the model
M number of observation symbols
S $S_1, S_2, ..S_N$ states
Q $q_1 q_2 ..q_T$ state sequence
V $v_1, v_2, ..v_M$ discrete set of possible observation symbols
q_t state visited at time t
A $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$, state transition probability distribution;
B $b_{jk} = P(v_k att | q_t = S_j)$, observation symbol probability distribution in state j;
π $\pi i, \pi_i = P(q_1 = S_i)$, initial state probabilities;

The compact notation $\Lambda = (A, B, \pi)$ is used as the parameters of the HMM. Given an observation sequence $O = O_1, O_2, .., O_T$ and a model Λ, the Viterbi algorithm is used to find the optimal state sequence associated with the observation sequence, and the Baum-Welch algorithm is used to learn the model parameters. So, we would like to have a confidence that a mine is present at a position (x,y) on the surface being traversed. Hence, the observation sequence at a point (x, y) is taken as the sequence of O(x,y-7), ... O(x,y-2), O(x,y-1), O(x,y), O(x,y+1), . O(x,y+7). Each of these observation sequences encode information about a mine signature; and are obtained from the second derivatives of the processed data as explained in detail in [5]. These second derivatives are clipped and normalized and a sixteen dimensional observation vector is formed from the positive anti diagonal, negative anti diagonal, positive diagonal and negative diagonal edges associated with a point. In our implementation, the for both the background and the mine models, $M = 3, N = 3, dim = 4; sequencelenght = 15, minimumsequencelength = 7$; hence the states of the mine model are the leading edge, center and trailing edge of a mine. Using the output of the Viterbi algorithm, a confidence map is produced as $C(x, y) = log(P(O(x, y), Q^* | \lambda_{mine})) - log(P(O(x, y), Q^* | \lambda_{background}))$.

Below is the results of this algorithm. Fig. 18 shows the scatter plot using the baseline HMM algorithm, Fig. 19 shows the scatter plot using a metal detector, and Fig. 20 is the fusion of the two using geometric and arithmetic means. Mathematically, the fusion is done as

$$PD = (\frac{x_1 + x_2}{2} + \sqrt{x_1 \times x_2})$$

In a scatter plot, we ideally we want all the mines above the background and clutter confidences. With these plots, the receiver operating characteristic curve is shown in Fig. 21 with 90/48.1 detection for HMM alone, 90/23.1 with

metal detector alone, and 90/23.1 by the fusion of both.

4 MCE TRAINING OF CDHMMS

Continuous Density Hidden Markov Models (CDHMMs) are very similar to the aforementioned discrete HMMs, with the difference that the matrix B is now a set of probability density functions, one for each state. Hence, in addition to the parameters defined in the previous section, we have

$$b_j(o_t) = \Sigma_{k=1}^{M} c_{jk} N(o_t; \mu_{jk}, \sigma_{jk})$$

where

$$N(o_t; \mu_{jk}, \sigma_{jk}) = \frac{1}{(2\pi)^{D/2} |\sum_{jk}|^{1/2}} e^{\frac{1}{2}(o_t - \mu_{jk})^T \sum_{jk}^{-1} (o_t - \mu_{jk})}$$

is a multivariate Gaussian density, D is the dimension of the feature vector o_t and $c_{jk}, \mu_{jk}, \sigma_{jk}$ are the weight, mean and covariance of the k-th Gaussian component of the Gaussian mixture density at the state j.

In [8], minimum classifier error (MCE) training is applied to the continuous density HMMs. The parameters of the MCE training are updated using a probabilistic descent approach. In MCE training, a misclassification measure is introduced as:

$$d_i(O) = -g_i O; \Lambda + log(\frac{1}{M-1} \sum_{j, j \neq i} exp[g_t(O, \Lambda)\eta])^{1/\eta}$$

where the class conditional likelihood functions are

$$g_i(O, \Lambda) = P(O|\Lambda^{(i)})$$

and the classifier operates under the following decision rule:

$$C(O) = C_i if g_i O; \Lambda = max_j g_j(O, \Lambda)$$

For an ith class observation O, $d_i(O) > 0$ means misclassification and $d_i(O) < 0$ is correct classification. This misclassification rule is embedded into a sigmoid function as

$$l(d) = \frac{1}{1 + e^{-\gamma d + \theta}}$$

Hence the expected loss is

$$L(\Lambda) = \sum_{i}^{M} \int l_i(O; \Lambda) p(O) dO$$

With the generalized probabilistic descent algorithm, we try to update the parameter set Λ so that the expected loss can be minimized, and the update equation for Λ is

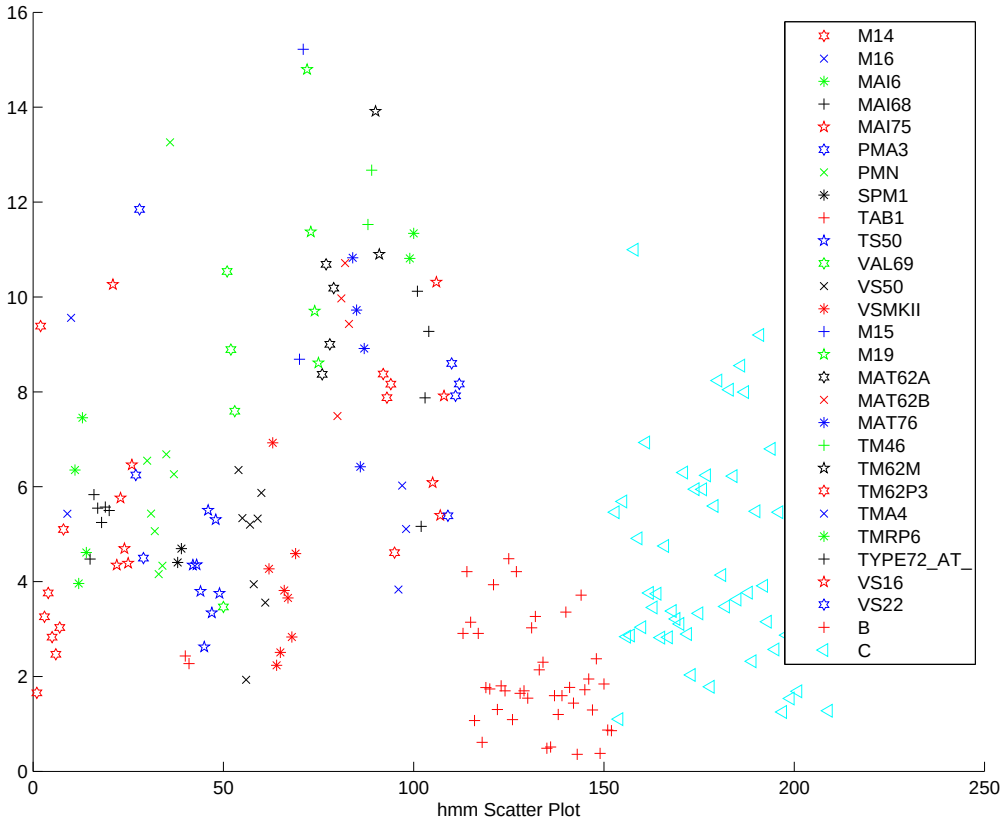


Figure 18: HMM Scatter Plot. In the scatter plots, we ideally want all the mines above the background and the clutter confidences.

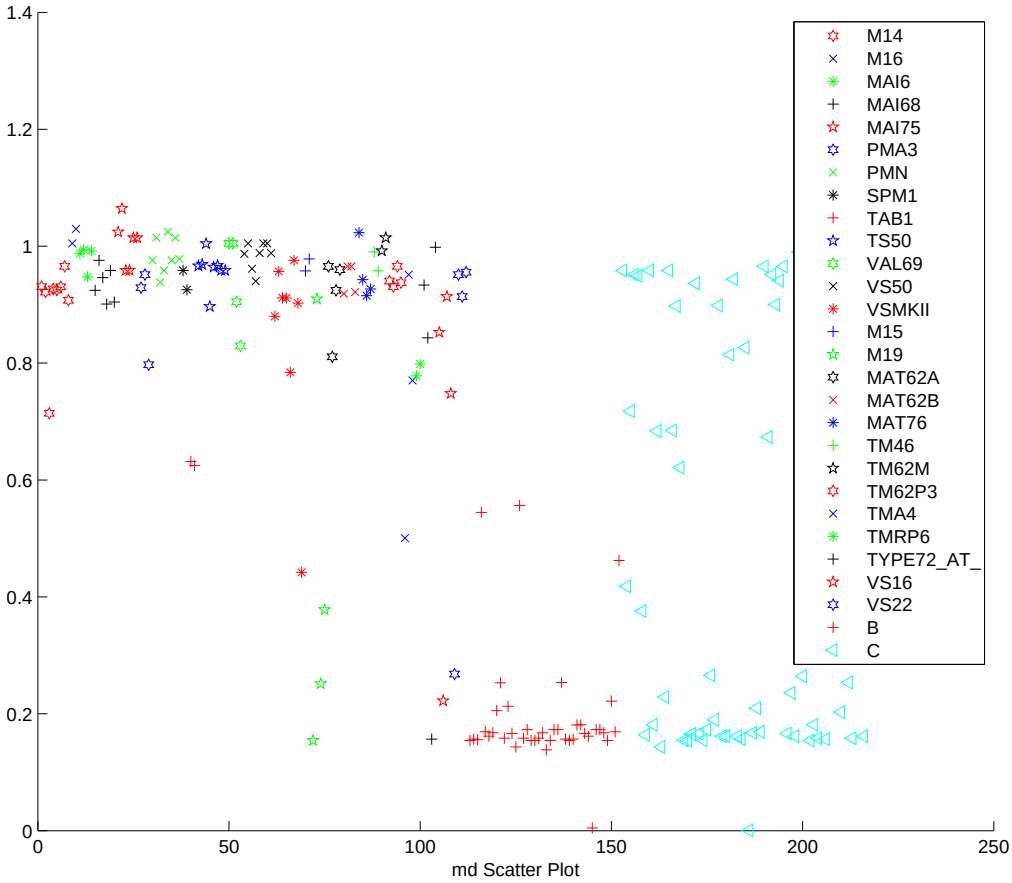


Figure 19: Metal Detector Scatter Plot. In the scatter plots, we ideally want all the mines above the background and the clutter confidences.

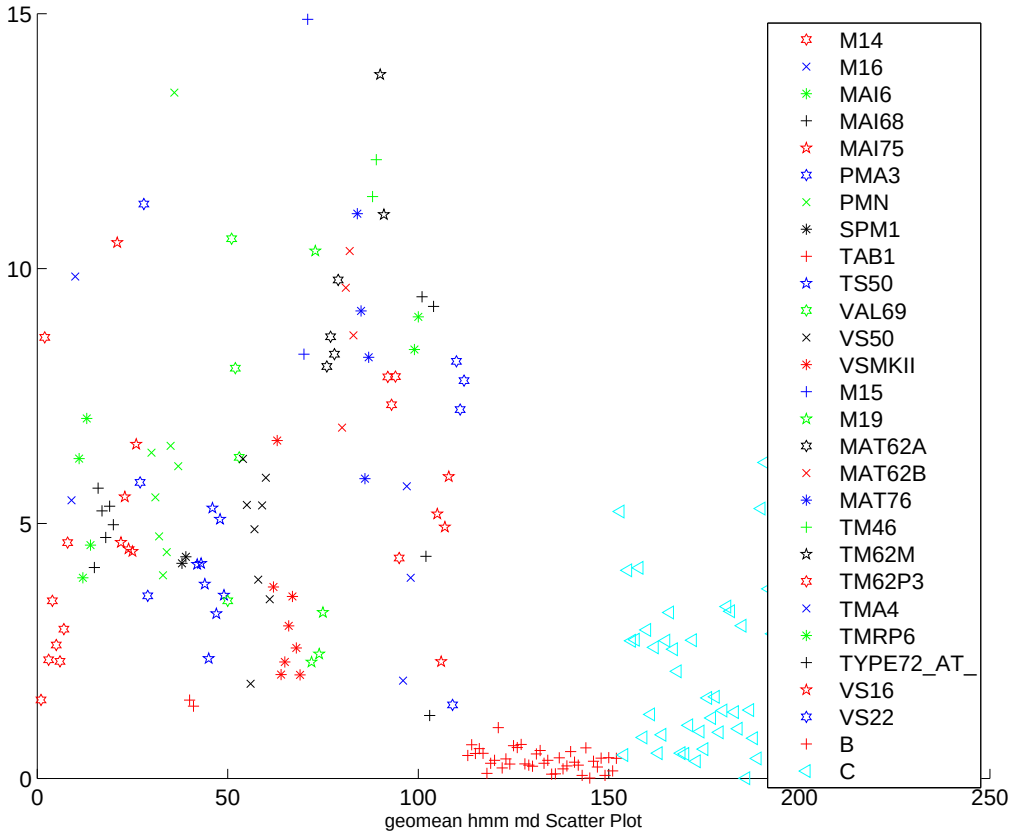


Figure 20: Fusion of HMM and Metal Detector using Geometric and Arithmetic means

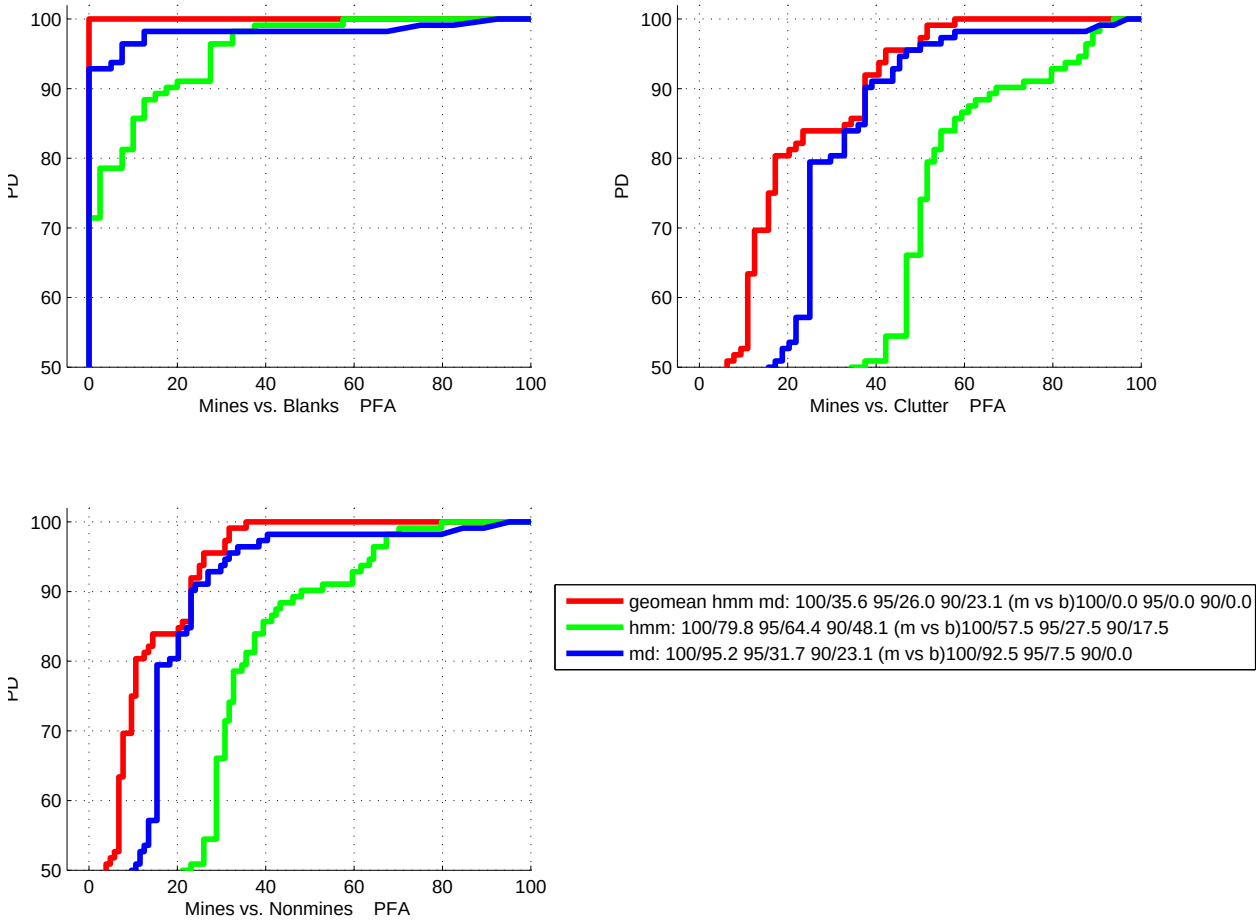


Figure 21: Landmining ROC's! : 90/48.1 detection for HMM alone, 90/23.1 with metal detector alone, and 90/23.1 with the fusion of both.

$$\Lambda_{t+1} = \Lambda_{t+1} - \epsilon_t \nabla l(O_t; \Lambda_t)$$

The gradients of the loss function are given in [8] in Eqns. (26) through (42). Using this algorithm, a glimpse into the training updates is shown in Fig. 22.

The MCE is a nice approach in the sense that we are not fitting a distribution to the training data; hence it is not affected by the mismatch between the assumed and true distributions of the data. On the other hand, it is a gradient descent method, and the algorithm is highly dependent on the sigmoid function as well as the step sizes.

5 IMPROVEMENTS TO MCE TRAINING OF CDHMM

Although the MCE training looked like a good training algorithm (ie. on the training set, it was learning beautifully), when it comes to testing, it is not even close to the baseline HMM within my evaluations. It looks like it is overtraining too much, and one reason for this might be the online training nature of the code, which takes the training sequences one by one and learns for every single one of them. For this reason, I implemented the batch training.

Another idea that we worked on is the ROCA training. In landmining, ultimately, the success of an algorithm is judged by its receiver operating characteristic (ROC) curves; hence it is a good idea to train the algorithm such that the area under the ROC curve is maximized. ROCA training algorithm introduced in [7], requires differentiable functions, which were already available in the MCE training. Hence, using the previously introduced loss function, we were able to implement the ROCA training.

5.1 Batch Training

In the online DCHMM algorithm, the data is taken one by one, and as the data is taken, the loss function and hence the parameter set is updated. This algorithm is quick in updating the parameters; however, since it learns for every data, it can overlearn. On the other hand, with batch training, all the data is collected at once, the gradients are accumulated as the data is encountered, then the loss function and hence the parameters are updated from the accumulated gradients after all the data is seen. The procedure for batch training can be outlined as shown in Fig. 23. The scatter plot is given in Fig.24, and the ROC curve is given in Fig.25. The result are too good so I believe something is wrong. I am still working on it.

5.2 ROCA Training

It is clear that at the end of the learning we look at the receiver operating characteristic (ROC) curves. Hence, a genius algorithm in [7] takes the ROC curves

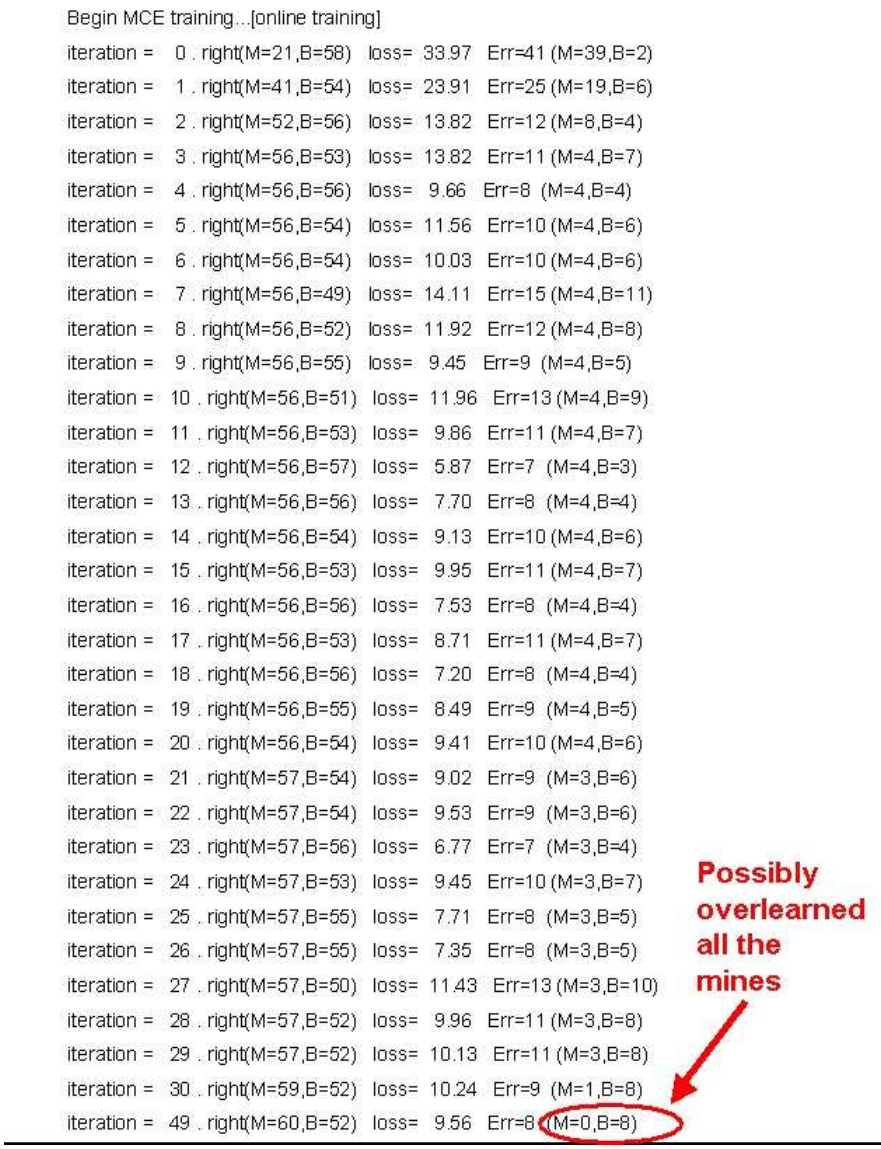


Figure 22: The learning steps of Online MCE learning for CDHMM. At the end of the iterations, the algorithm learned all the mines, but possibly overlearned.

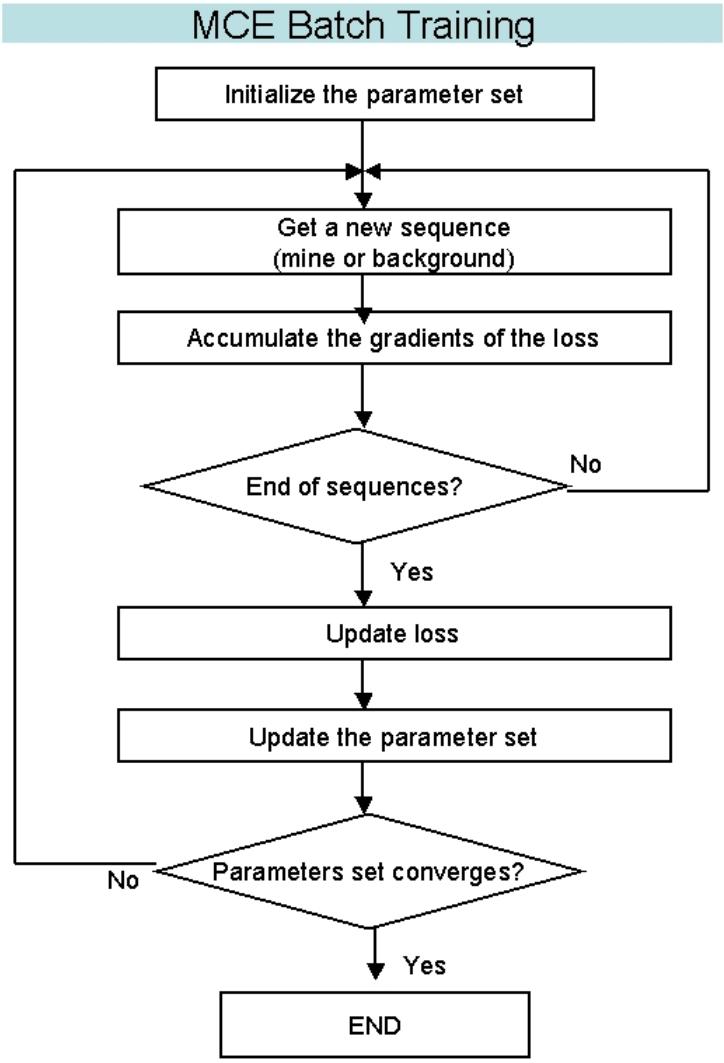


Figure 23: Procedure For Batch Training

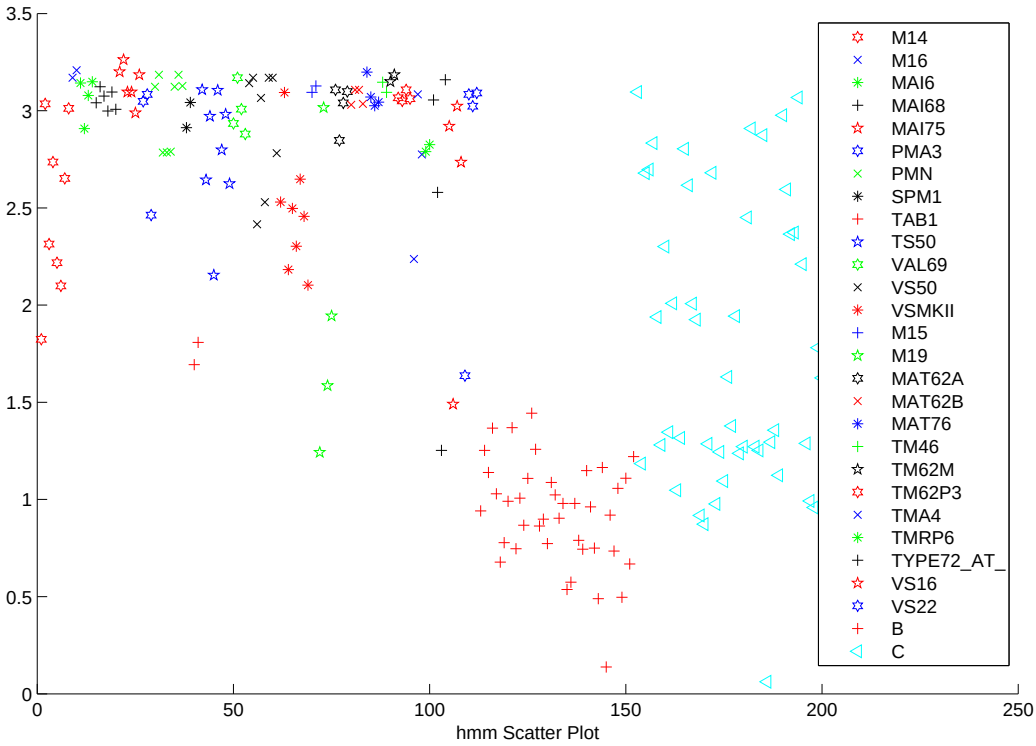


Figure 24: Scatter plot from batch training

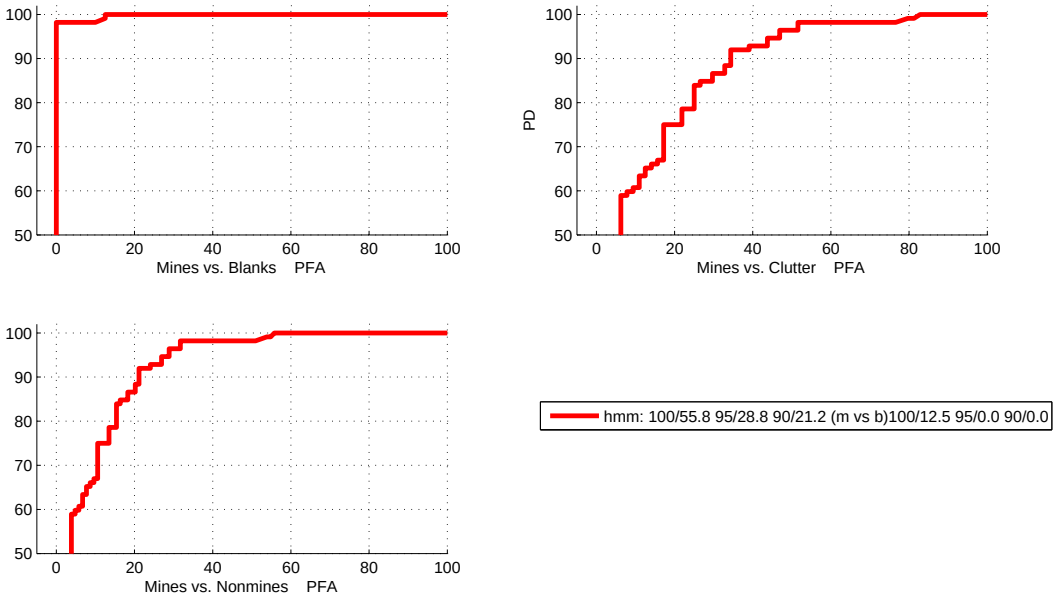


Figure 25: ROC curve after batch training

and tries to minimize the distance under them. The algorithm presented there requires the derivatives be known, which makes the MCE a perfect candidate to apply this algorithm. So the ROCA training algorithm is given as:

$$d_{ij}(\theta) = f(x^i;\theta) - f(y^j;\theta)$$

$$J(\Delta T) = \frac{1}{\Delta T.MA} \sum_{i=1}^{M} \sum_{j=1}^{N} T_{ij}$$

$$T_{ij} = \begin{cases} a) & 0 & if\ (i,j) & | & d_{ij}(\theta) \leq 0 \\ b) & d_{ij}^{\theta} & if\ (i,j) & | & d_{ij}(\theta) \in (0, \Delta t] \\ c) & \Delta t & if\ (i,j) & | & d_{ij}(\theta) > \Delta t \end{cases}$$

Algorithm: ROCA Training

Initialize θ
Do until stopping criterion reached

1. Compute $f(x^i;\theta)$ and $f(y^i;\theta)$ for every training sample
2. Set $\nabla_{\theta}^{ave} J(\Delta t) = 0$;
3. For each pair of training samples x_i and y_j
4. Check if case (b) is satisfied. If yes,
   - Compute $\nabla_{\theta}^{i,j} J(\Delta t)$ from the gradient update formulas of MCE training
   - Set $\nabla_{\theta}^{ave} J(\Delta t) = \nabla_{\theta}^{ave} J(\Delta t) + \nabla_{\theta}^{i,j} J(\Delta t)$
5. Update $\theta = \theta + \mu \nabla_{\theta}^{ave} J(\Delta t)$

6 CONCLUSION

I have found landmine detection as a very interesting real-world problem with endless future directions into machine learning, computer vision, fusion, statistics and even heuristics. The well-known machine learning problems such as parameter selection and overtraining are easily observed here with no doubt; and the detection rates are in 90/30's with lots of possibility for enhancement. I find myself lucky to have touched all the areas of this problem during my summer semester; in the very basics with image processing applications, HMM's for machine learning and metal detector and GPR fusion. And I have to admit, in some of the clutter cases where I assumed a mine from GPR signatures; the fusion of metal detectors and GPS's made a far better job than me, so my hopes are high that this research will really benefit those who suffer from the consequences of wars even long time after them.

References

[1] Hidden killers: The global landmine crisis. Technical report, U.S. Dept. State Rep. Publ 10 575, Washington, DC, Sept 1998.

[2] Landmines, mine action news from the united nations. Technical report, 1998.

[3] H. Frigui, K. C. Ho, and P. Gader. Real–time landmine detection with ground–penetrating radar using discriminative and adaptive hidden markov models. *EURASIP Journal on Applied Signal Processing*, (12):1867–1885, 2005.

[4] P. Gader, W. Lee, and J.N. Wilson. Detecting landmines with ground–penetrating. radar using feature–based rules, order. statistics, and adaptive whitening. *IEEE Transactions on Geoscience and Remote Sensing*, 42(11):2522–2534, 2004.

[5] P. D. Gader, M. Mystkowski, and Y. Zhao. Landmine detection with ground penetrating radar using hidden markov models. *IEEE Trans. Geosci. Remote Sensing*, 39:1231–1244, June 2001.

[6] Colin King, editor. *Jane's Mine and Mine Clearance*. Jane's Information Group, Great Britain, second edition, 1998.

[7] W.H. Lee, P.D. Gader, and J.N. Wilson. Optimizing the area under a receiver operating characteristic curve with application to landmine detection. *IEEE Transactions on Geoscience and Remote Sensing*, 45(2):389–397, February 2007.

[8] Chuanhong Ma. Mce training based continuous density hmm landmine detection system. Ms, University of Missouri – Columbia, 2003.