INTERNSHIP REPORT

We, **Achin** and **Harekrissna** worked as a team to complete the project given to us on Buried threat detection using ground penetrating radar.

We applied Machine Learning techniques specifically CNN to identify the threats hidden underground by analysing the radar data. We implemented the techniques given in the research paper (Some Good Practices for Applying Convolutional Neural Networks to Buried Threat Detection in Ground Penetrating Radar, by Daniël Reichman, Leslie M. Collins, Jordan M)

The implementation goes as follows: -

1. We use grayscale imagery from the Cifar10 dataset to pre-train our GPR CNN.
2. The parameters for all layers of a CNN are learned jointly using adam optimizer
3. In training, the dataset is provided to the network in mini-batches (i.e., small, randomly selected subsets) and the whole dataset is traversed times (epochs). To update the weights, the network is evaluated on a mini-batch of data, the prediction is compared to the training labels, and the error is back-propagated through the network. At the end of an epoch, the network is typically evaluated on a validation set to measure whether the network is learning general patterns or overfitting to the training data The final network chosen for testing is the one with the highest validation accuracy.
4. Network pretraining: Pretraining here was done by transferring at most 4 convolutional layers from the network trained on the Cifar10 dataset to the GPR network.
5. Training data augmentation: In the vertical dimension, patches at every other pixel above and below the maximum keypoint location for threats are chosen, up to 4 pixels away. In the horizontal dimension, one patch is chosen 2 pixels away, in both directions from the maximum energy keypoint for threats. For non-threats, 33% of the data is randomly sampled from 3 A-scans: the central A-scan, one A-scan 2 pixels to the right and to the left of the central A-scan.
6. The CNN network is same as that proposed in the research paper

CNN configurations			
Input (18 × 18 image)			
(a)	(b)	(c)	(d)*
Conv3-16	Conv3-16	Conv3-24	Conv3-4
Conv3-16	Conv3-16	Conv3-24	Maxpool (2*2, 2)
Maxpool (2*2, 2)			Conv3-4
Conv3-32	Conv3-32		Maxpool (2*2, 2)
Conv3-32	Conv3-32		
Maxpool (2*2, 2)			
Conv3-64			
Conv3-64			
Maxpool (2*2, 2)			
	FC-64	FC-32	FC-2
		FC-32	
FC-16			
Soft-max			
	*used for simpler data		

Results

Accuracies for network (a) for 35 epochs on complicated non-threat images, was found out to be : -

Training	92.8
Testing	92.25

Accuracies for network (a) for 100 epochs on simpler non-threat images, was found out to be : -

Training	92
Testing	82.5

Accuracies for network (d) for 100 epochs on simpler non-threat images, was found out to be : -

Training	96.8
Testing	96.7

Accuracies for network (d) for 100 epochs on complicated non-threat images, was found out to be : -

Training	67.5
Testing	65