Automatic Target Recognition through Clutter Classification

Infrared dataset classification

Sabaina Haroon
University of Central Florida
Orlando FL USA
sabainaharoon@knights.ucf.edu

ABSTRACT

A lot of revolution globally has been brought using deep convolutional neural networks in medical image analysis, self-driven vehicle, crowd management and many more.

Infrared (IR) imaging has yet to make a lot of advances based on these concepts, automatic target recognition has ongoing researches where efforts are being put to understand IR data using deep learning concepts. There are multiple challenges being faced while working with IR data due to the nature of heat sensitive cameras and no color information due to which it is hard to classify and detect objects in IR by just using the machine learning approaches. In this project, by using the concept of clutter classification where we paired output from classical computer vision model to deep convolutional model, we tried reducing the computational cost of data processing and also allowed objects to be classified as targets and clutter, which will help in segmentation of targets from the background in future.

CCS CONCEPTS

• ***Interest point detector***: Oriented Fast and rotated brief, feature extractor

KEYWORDS

Infrared image classification, interest point detector, clutter classification, automatic target recognition, object detection, localization and classification

ACM Reference format:

Sabaina Haroon, 2019. Automatic Target Recognition through Clutter Classification: Infrared dataset classification. In *Proceedings of ACM conference ACM, New York, NY, USA.* https://doi.org/10.1145/1234567890

*This research paper is written as a term project of machine learning course

ACM Conference on Artificial Intelligence, Ethics, and Society, February, 2020, New York, NY USA
 978-1-4503-0000-0/18/06...$15.00
https://doi.org/10.1145/1234567890

1 Introduction

We have seen tremendous advancements in detection and localization models in machine learning in a short span, where we have come across from Voila Jones algorithm to R-CNN, ResNet, YOLO models that produce promising results on different datasets. Whereas for works related to IR datasets, simply feeding the data to machine learning model for detection and classification doesn't contribute enough since we have multiple challenges common to IR data due to lack of color information and data acquisition tool sensitive to heat changes. For our project we are going to detect and classify moving targets that are captured through IR cameras. We have 9 classes of army vehicles in this dataset, vehicles are set to move on a set route and IR camera captures their movement. We get images as six frames per second, where vehicles are moving in daytime and nighttime covering range of 1 kilometer to 3.5 kilometer. IR dataset has its own challenges; we have images in one channel having less information, time of day affects quality of images i.e. Images captured at night will have less intensity information compared to images taken in day. Whereas daylight images will contain higher noise ratio because of having high intensity values for parts of image that are other than region of interest. Third challenge will be of range, we will lose information as the object will move farther away as range increases. This will make classification more challenging as structure of vehicle would be less defining and difficult to differentiate at distant ranges. Due to the challenges of IR data set typical machine learning approaches find it very hard to classify this type of data correctly, we try a different approach where we calculate feature points through a detector designed on classical computer vision approach. The term detector is mostly used for the tool which extracts features in an image [1]. Johansson et al. in their paper on "interest point detectors and descriptors for IR images" have showed comparison of various detectors on IR images. Oriented FAST Rotated BRIEF(ORB) is basically a corner-based detector which performed very well on IR images [1]. To finalize our detector, we have also compared ORB with Harris corner detector, and Laplacian of Gaussian based interest point detection algorithms, later detectors would give detection accuracy but with more clutter points. Whereas, ORB is selected for our algorithm to suggest region of interest since it gave greater detection accuracy with least false alarm compared to the other two methods. After

we get feature points from the detector, deep convolutional neural network classifier classifies these feature points as target and clutter by fitting a window across each point. Detector will localize the interest points using confidence values, detector result with optimized confidence/threshold value will be sent to CNN which will further filter out the interest points by clutter classification

2 Dataset

For this project we are working on automatic target recognition (ATR) NVESD dataset for Infrared images. Dataset was collected through IR cameras by moving 10 different type of vehicles on a path ranging from 1 kilometer to 3.5 kilometer. For this project we have used day and night images of moving vehicles in the range of 1 km to 2.5 kilometer, and for 2.5 to 3-kilometer range we only have dataset with day images. There are 10 type of vehicles, but we have used 9 classes for this project. One type of vehicle was attached to another type and we considered both as one class for the scope of this project. Hence, we have 9 classes for vehicle and 1 class for clutter classification.

Table 1: Target types and corresponding lookups

Lookup number	Target Type/ Classes
2	Pickup
3	Sports Utility Vehicle
4	BTR70 – Armored Personnel Carrier
5	BRDM2 – Infantry Scout Vehicle
6	BMP2 – Armored Personnel Carrier
7	T72 – Main Battle Tank
8	ZSU23-4 - Anti-Aircraft Weapon
9	2S3 – Self-Propelled Howitzer
10	MTLB – Armored Reconnaissance Vehicle Towing a D20 Artillery Piece
11	Clutter class

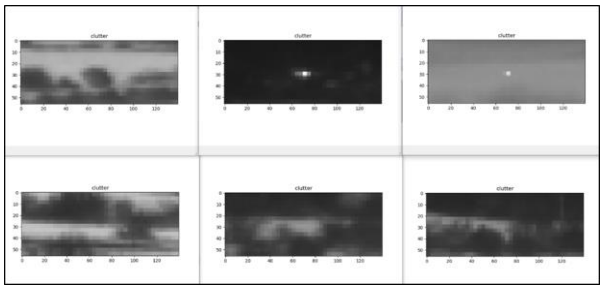

Figure 1: Sample images from clutter class

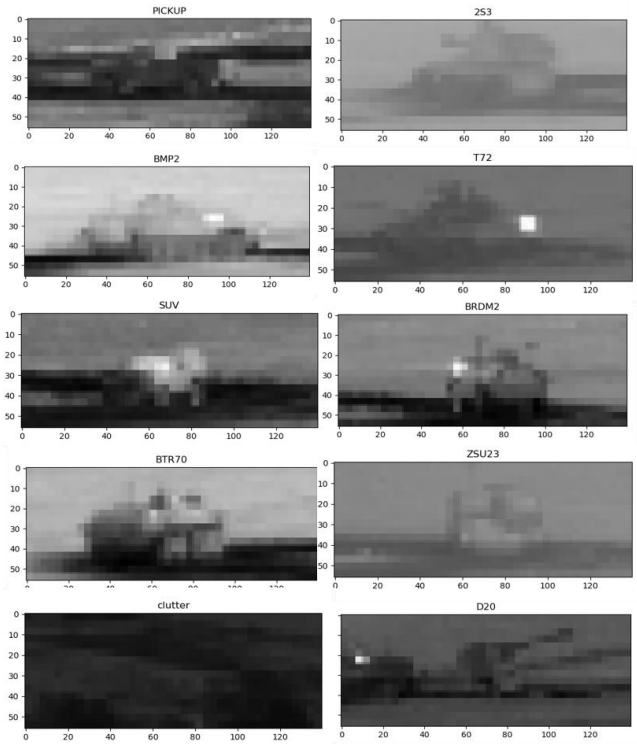

Figure 2: Sample images from 9 target classes and 1 clutter class

This project was challenging due to the nature of IR dataset as can be seen from the images above. Some of the target images are not recognizable even from human eye due to poor contrast and low resolution.

3 Detector and Classifier

To organize input data for machine learning model, in first step entire image was penalized to only those parts that are of our interest and may contain target, so instead of sliding the window we aim to fit window at selected locations in an image which decreases computation time and makes it possible for model to classify and localize target better for our dataset. To detect feature points, we selected ORB detector after testing it with other conventional detectors. Scale Invariant feature transform (SIFT) and Speeded up robust features (SURF) are two very popular detectors used for detection, but we used ORB compared to these because of two reasons, ORB performs as accurately as SIFT and SURF but is faster and SIFT, SURF are paid patented algorithms whereas ORB is free to use. Threshold is used as a confidence measure of detection in our algorithm, and we generated feature points against multiple thresholds to generate ROC for day and night. Higher threshold generated less feature points giving us less clutter but decreased detection accuracy and vice versa.

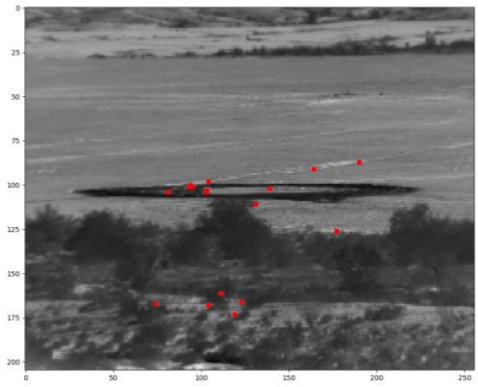

Figure 3: feature points generated using ORB detector with threshold value = 50

Our architecture for classifier explores automatic target recognition with a different approach where along with target we aim to classify clutter too, coupled with the output of detector this will increase the accuracy of detecting and classifying the target correctly. Since we have Infra-red (IR) dataset for this project, conventional machine learning (CNN) architectures such as YOLO, VGG are not enough for training. Therefore, we detect feature points from a frame through classical way using methods stated above. Depending on the threshold, detection methods refine the feature points still we get some points that are not target but have sharp features to be selected as output. These excess points will be filtered out through deep convolutional neural network. As a preprocess step to classifier, we need to compute images from these feature points to generate input for our CNN. Against each input image, detector will output multiple points, we produce mini-sampled images by taking a window around these points. We have tried different window sizes for cropping sub-images from an image. Challenges with cropping a window are:

1. If we crop a window in exact size as average bounding box size calculated from ground truth target vehicles from some classes might be large and get chopped off.
2. If we take conventional bounding boxes that have been used for ATR dataset such as 64x64, 128x128, target is cropped but clutter in the background is included in that too and classifier will classify targets with poor contrast and those in long range as clutter.

Since It's a trade-off between above stated problems therefor we faced later problem in our final bounding box approach but it improved results compared to above two methods. For cropping, average radii of bounding box are calculated from ground truth available for vehicles in our dataset. This comes out to be radius_x = 13 and radius_y = 5, we extend it by;

$$\boldsymbol{radius_x} = 1.5 * radius\_x$$
$$\boldsymbol{radius_y} = 1.5 * radius\_y$$

We crop bounding box around each point by this radius and upscale the image by a factor of 7 using inter area interpolation.

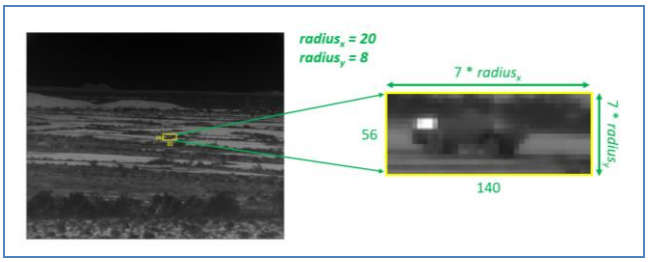

Figure 4: Procedure followed to crop window around each feature point detected

Next, we did mean normalization by taking mean image from overall data, subtracted it from each image and divided zero centered image by 255 as RGB images have pixel intensity range from 0-255. Doing mean normalization in this way increased accuracy by 30 percent. Against each image we get multiple clutter points, i.e. per image we might have 1 target image and approximately 30 clutter images, since we have around 15000 images in this dataset, this makes total sub-images around 465,000 and only around 15000 images from it would be of target classes. Which makes this a skewed dataset. Before throwing this dataset into our model, we subsample clutter class, for the scope of this project we are randomly shuffling the clutter images and picking up some percent of these images for training, in future subsampling on these images can be done using data reduction techniques where clutter images that are very similar to each other and are redundant can be removed. After subsampling clutter is still around 60 percent of the entire dataset, for which we use weight balancing, it balances our data by altering the weight that each training example carries when computing the loss. i.e. all the classes except clutter class would be given higher weightage, this would avoid the classifier to be biased towards the clutter class. After this data is fed to the deep convolutional neural network

3.1 Proposed deep CNN

We started classification for our dataset following architecture proposed in Reference. [1]. Following this approach, which was proposed for ATR dataset, we did not get satisfactory accuracy, since targets in our data were low in resolution and most of the targets in the dataset have poor contrast and low resolution that they don't seems to have any structure and are more like clutter. This problem can be seen more in day images with ranges from 3 kilometer and higher.

3.2 Gains using filter with different dimensions and Leaky ReLU

For our architecture, we used different dimensions for our filters in each layer compared to the scalar filter sizes that are commonly used in other CNN architectures. This improved accuracy by 15

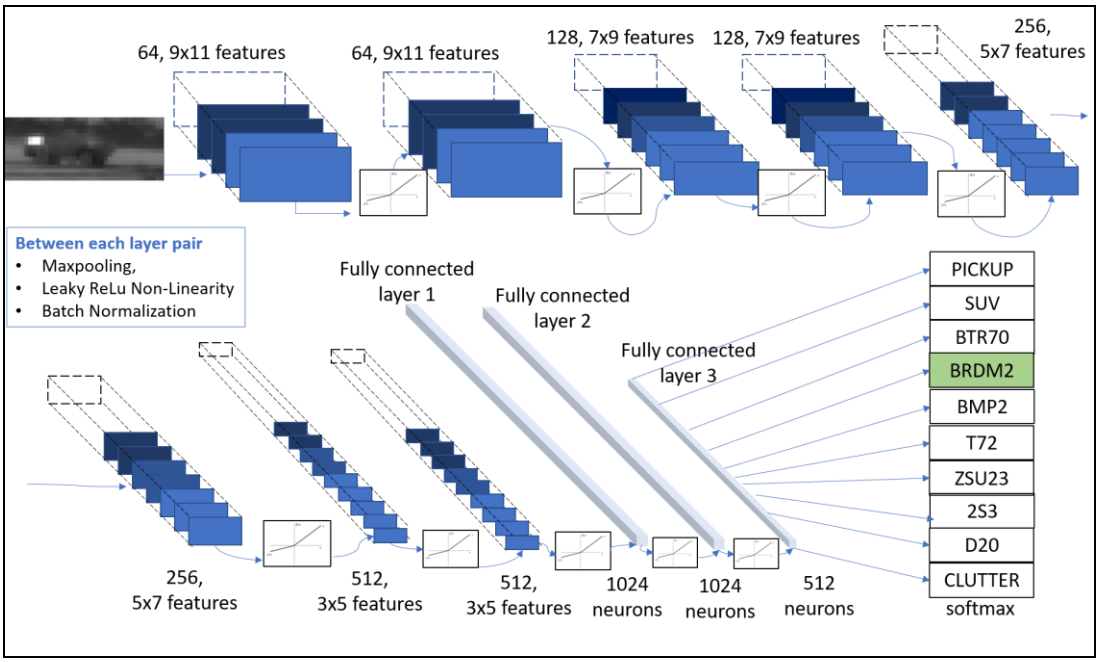

Figure 5: Deep convolutional neural network architecture

percent. Other than this we replaced our basic architecture with leaky linear rectifier (ReLU) non-linearity instead of ReLU, and it increased the accuracy by 5 percent.

3.3 Training Consideration

We use the following experimental setting for training:

1. The optimization algorithm used for training is Adam. We used following configuration parameters i.e., learning rate =0.001, β1=0.9, β2=0.999
2. We use the He et. Uniform weight initialization scheme to prepare the network before training. This initialization helps with vanishing gradient.
3. To avoid overfitting, we used dropout with a unit drop probability of 0.5 in fully connected layers.

Table 2: dCNN summary for the first two layers for reference

Layer type	Output shape	Param #
conv2d_1 (Conv2D)	(48, 130, 64)	19072
conv2d_2 (Conv2D)	(40, 120, 64)	405568
max_pooling2d_1 (MaxPooling2	(20, 60, 64)	0
leaky_re_lu_1 (LeakyReLU)	(20, 60, 64)	0
batch_normalization_1 (Batch Normalization)	(20, 60, 64)	256
conv2d_3 (Conv2D)	(14,52,128)	516224

4 Algorithm

1. Image is sent to the detector function, which runs ORB algorithm and outputs feature points as pixel location
2. A window is fit around each point from detector of fixed length and image inside the video is cropped.
3. Ground truth labels for cropped images are defined as clutter class or one of the 9 the targets
4. Step 1 to 3 are repeated for all the images to obtain training data and test data
5. We have 1 target vehicle in each image, but on average we might get 10 feature points per image, which makes ratio of clutter class to other 9 classes as 9:10, so in next step we subsample the clutter class by random votes
6. Each image is zero centered and normalized
7. Data is passed to the deep CNN model for fitting and prediction

Since it was skewed data, subsampling as preprocessing and weight balancing was done inside model and metrics therefore used for model evaluation are accuracy, f1 score, recall, precision and loss.

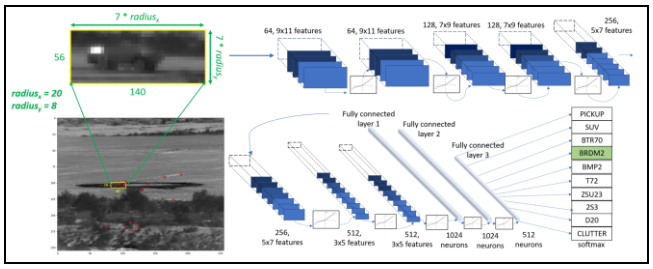

Figure 6: algorithm steps shown for a single image

5 Evaluation

Detection accuracy was evaluated from the detector by calculating probability of detection and false alarm rate at different thresholds. ROC curves for detection accuracy during day and night can be seen from figure 7 and 8. Following formulas were used to calculate detection accuracy and false alarm rate;

$$Probability\ of\ detection = \frac{No.\ of\ targets\ detected}{total\ number\ of\ targets}$$

$$Average\ False\ Alarm = \frac{Total\ Number\ of\ False\ detection}{Total\ no\ of\ Frames}$$

For results shown in figure 9 to 11, model was trained on vehicles moving at a range of 1 to 2 km but tested on those moving during day at a range of 2.5 to 3.5 km. Most images from class 1, 2 and 9, for 2.5 to 3.5km range were of poor resolution and different contrast compared to those fed in training, these images looked more like clutter and noise rather than target because of low resolution and were totally different from their training images as can be seen from figure 12 and 13. When we shuffled the data of all classes in this range of 2.5 to 3.5 km (day) and provided half of the data to retrain the model together with some of the images from day and range higher than 2.5, results improved drastically and accuracy reached from around 56 to 92.84 percent as shown from in figures 14 to 16. This showed data augmentation is crucial for this kind of data. We have moving vehicles, that are rotated at all angles and have difference in contrast due to time of day and change in scale due to ranges. If we synthetically generate images from training data, by changing scale, contrast and angle of rotation, model can work very well on unseen data also.

For results shown in figures 17 to 19 model was trained on 39810 samples, containing images from day, night of all ranges and validated on 4403-night images samples. As can be seen since night images are higher in resolution due to closer in range and have very less noise therefore, we have very few images from each class assigned as clutter.

For results obtained in figure 20 to 22 images of all ranges and all time were shuffled and combined, 70 percent of the total data was used for training and 30 percent of it was used for testing. Hence model was trained on 31622 samples and results were validated on 10450 samples

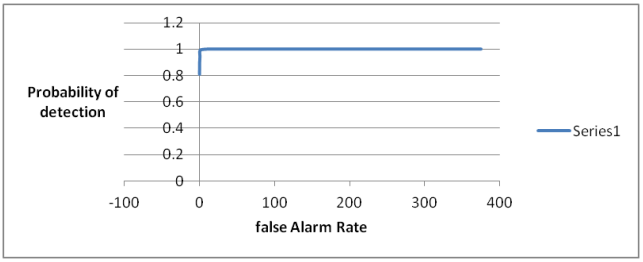

Figure 7: ROC curve using ORB detection for night images

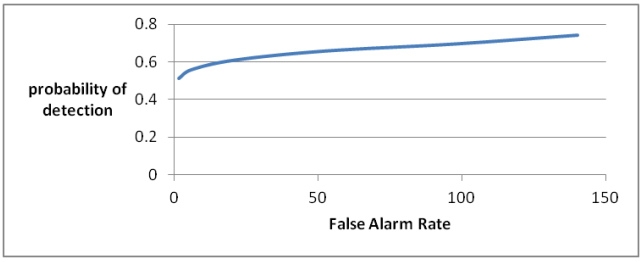

Figure 8: ROC curve using ORB detection for day images

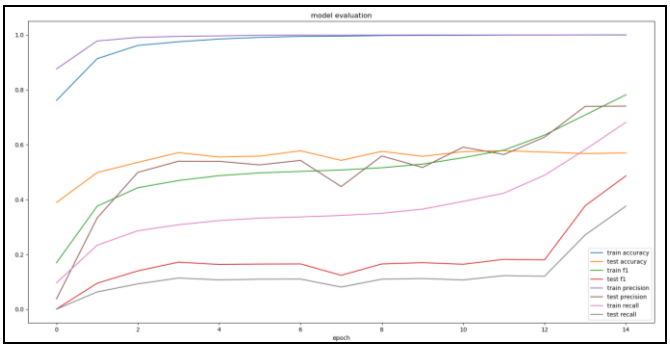

Figure 9: Model evaluation on day images of 3.5km unknown to training

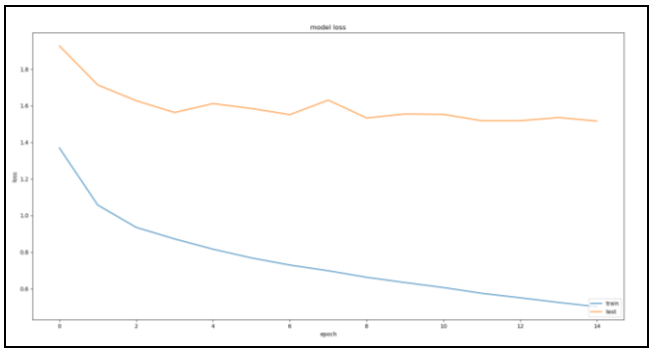

Figure 10: Model loss for day images of 3.5 km, unknown to training

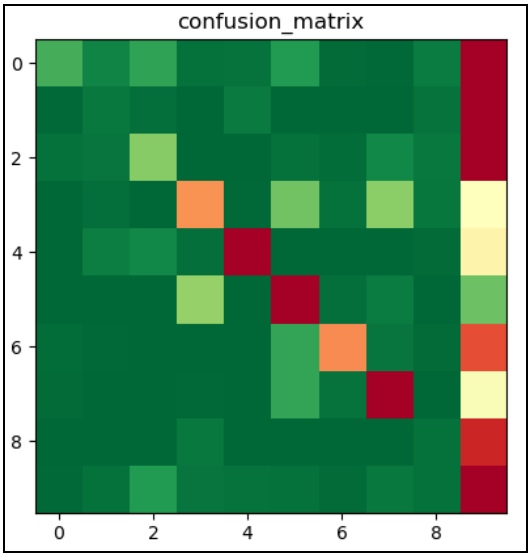

Figure 11: confusion matrix for day images of 3.5 km, unknown to training

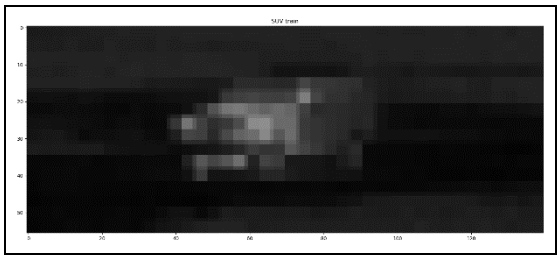

Figure 12: training set image of class 3

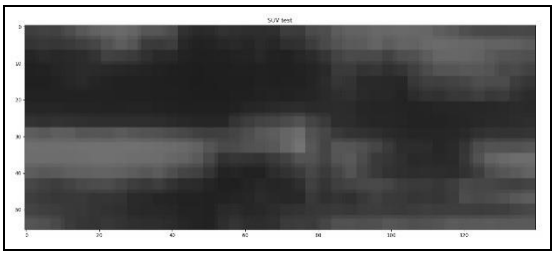

Figure 13: test set image of class 3

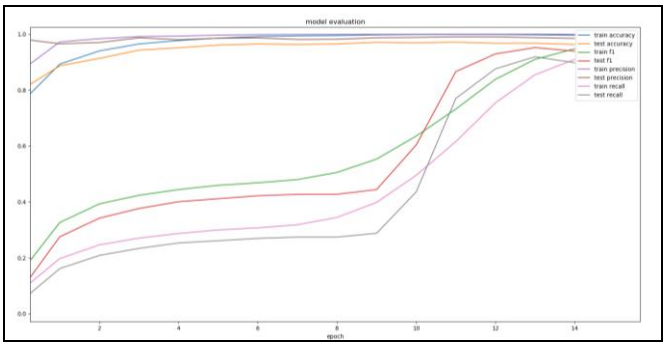

Figure 14: model evaluation for day images of 3.5 km

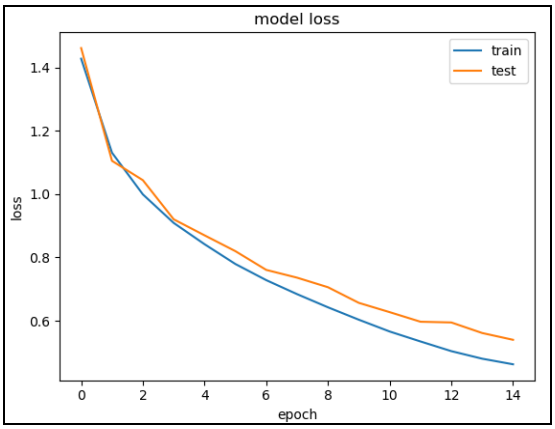

Figure 15: model loss for day images of 3.5 km

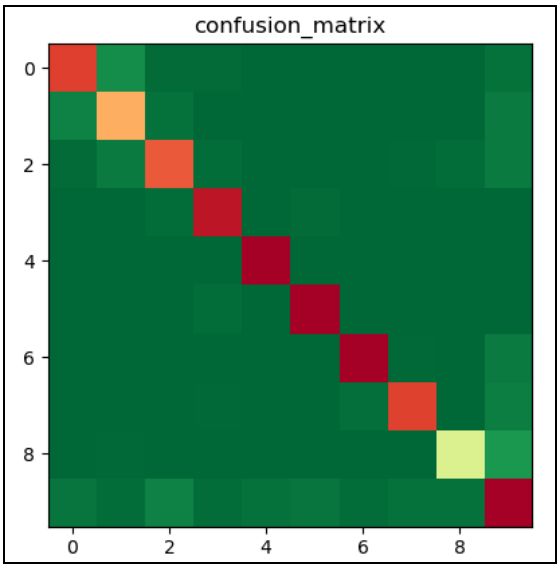

Figure 16: Confusion matrix for day images of 3.5 km

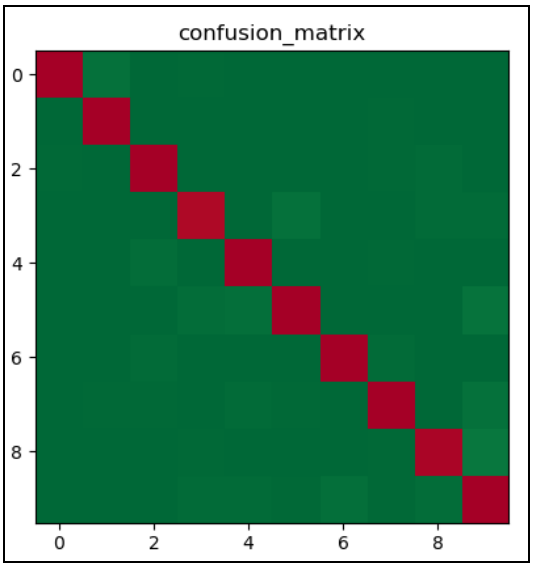

Figure 19: Confusion matrix for night images

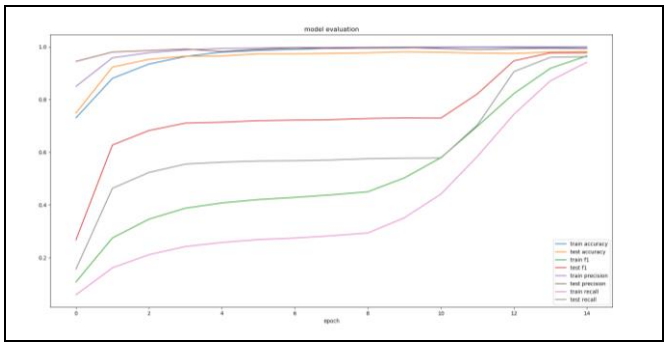

Figure 17: model evaluation for night images

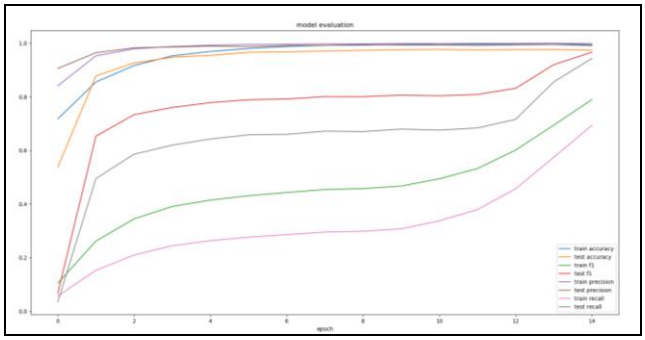

Figure 20: Model evaluation on 30 percent of total data

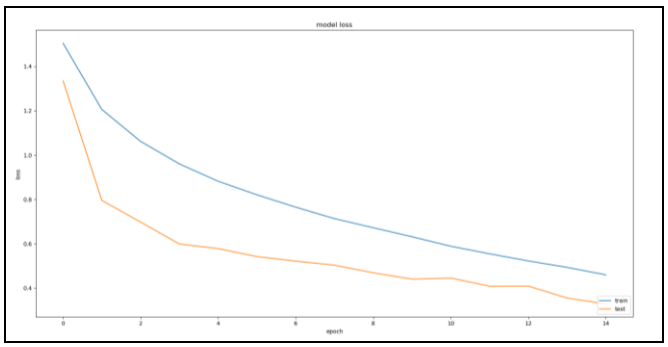

Figure 18: Model loss for night images

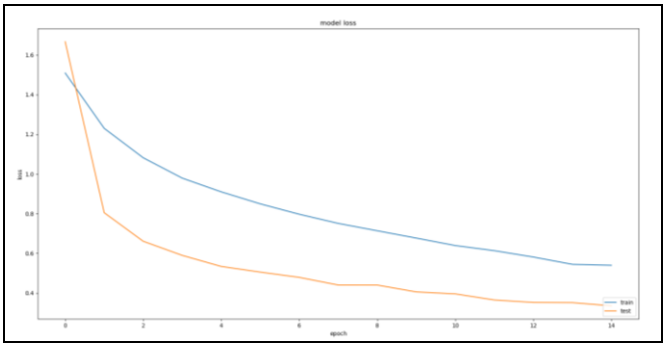

Figure 21: Model loss for 30 percent of total data

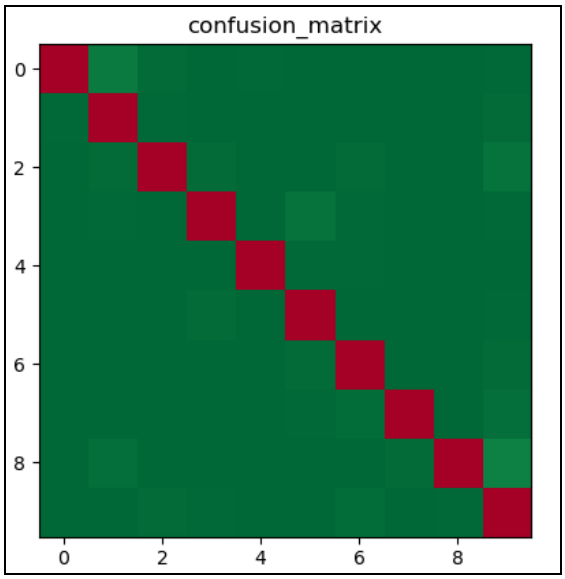

Figure 22: Confusion matrix for 30 percent of total data

6 Conclusion

The approach used in this project where classical approaches and deep convolutional learning were coupled improved the performance compared to using machine learning model for both detection and classification. Our Detector gave the accuracy of 94.9% with false alarm rate of 3.016 on night images, and 91.23% accuracy with false alarm rate of 27.95 on day images. High threshold value worked better for day images but for night images low threshold gave good performance. Threshold of zero resulted in decreased performance when used with window(grid). We found through tests and trials that classifier accuracy increases by using Leaky ReLU instead of ReLU non-linearity and adding mean normalization to the input. Validation accuracy for images above the range of 3km comes out to be around 60 percent since trained model has not seen and learned far range images in daylight from input training images of 1 to 2 km. This problem can be solved using data augmentation where we can generate synthetic images of distant ranges and color intensity to help model learn unseen data also. Other than this all the settings for validation described above give accuracy of classifier around 95 percent. Finally, the use of clutter classification paired with detector increases the accuracy of detection by discarding the feature points that are not of our use and by keeping only the target points.

REFERENCES

[1] Johansson, J. (2015). Interest point detectors and descriptors for ir images: An evaluation of common detectors and descriptors on ir images.

[2]. Antoine d'Acremont, Ronan Fablet, Alexandre Baussard, and Guillaume Quin CNN-Based Target Recognition and Identification for Infrared Imaging in Defense Systems