CAP 5415: Main Project : ATR data-set

Abraham Jose, Ganesh Aravind
ID :5068109,5068570 CAP6614
abraham@knights.ucf.edu,ganesharavind@knights.ucf.edu

Abstract

Deep learning and CNNs are undoubtedly the state of the art for image classification and other important tasks in computer vision. Here, we address image classification, detection and tracking on the ATR data-set. Considering factors such as the robustness issues, we have tried to come up with our models that make use of the pre-trained ResNet 101 and UNet with a depth of 4 for classification and detection respectively. Also, we have performed certain data augmentation and fine tuning to achieve the best possible results. In spite of the data set being extremely challenging, the results look promising for both classification and detection. The use of multi frame approach with difference in the image has been the key as it helped extremely in detection using segmentation. We demonstrated tracking by making use of Kalman filter in this 2d image through motion of the target. The approach we have taken to track is by predicting the next location of the vehicle and reducing the noise generated by incorrect detection of targets. We have also tried many other approaches to solve this problem, including a multi-frame CNN and point based detection as well on trial basis which is mentioned in this report. The model and the code is provided with a set of consecutive data taken from 3.5 km range to test our model using Kalman tracker.

1 Introduction

Deep learning in computer vision has helped us achieve great miles. It is impossible to imagine computer vision without deep learning today. The feature extraction and response when cascaded helps us to extract higher level features for 3 important tasks. Image classification, Image Detection and Image Segmentation. We approach this problem as a segmentation problem and classification problem. We chose to use UNet architecture for segmentation after out literature survey as UNet is more flexible enough to learn the spatial features like pattern, textures accurately. For the Classification of the detected model, we are using ResNet 101 model to learn the classes. This compared to ResNet 152 provides much better accuracy.

The data-set is challenging for the given test and training data-sets but the models we chose are promising in terms of learning. We went ahead using multiple frame approach for detection to simply the input fed into the model and to reduce the noise due to wind and clutter. We found that the absolute difference of the image was able to fit the model really well to the problem statement and we propose a UNet that uses the absolute difference of images as input to our final solution.

Another challenging task we tried to perform is to track the object. We implemented a Kalman filter by tracking the predicted object location with significant accuracy by suitably initializing the state variables for 2D single target tracking.

2 Dataset and Preprocessing

The given ATR data-set is a challenging given the test and train data-sets that we have. The range of values for the training data-set is 1000 meters to 2000 meters and for testing is 2500 meters to 3500 meters. This makes the problem to detect the target harder. However, we decided to go ahead with this training and testing to see the room for improvement given this is the data-set. However, we used the data-set on the best available data-set on the mixed ranges for prediction.

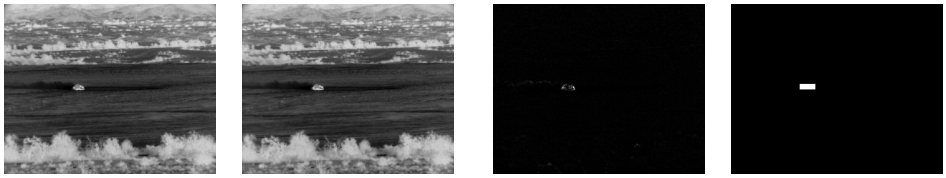
The initial processing of the images and ground truth were geared towards cleaning the data and making it as simple as possible. We have found many sequences of incorrect data with wrong annotations in the given pickle file as in the following figure. The sequences are [1] creg01927_0002* [2] creg01927_0005*, [3] creg02005_0005* [4] creg02007_0005*.

Figure 1: The red dot is the ground truth. However, the actual target is the white spot. This sample is from the sequence cegr01927_0005*. We removed all inconsistent data-points from the data-set.

Further while examining the data-set, we found that the target categories 'D20' and 'MTLB' co-occur all time as one pulls the other. So we combined both the targets to single target 'D20_MTLB'. This allowed us to have only single target per image. For classification we used it as a single target.

All images are batch-wise normalized based on intensity of the pixel in the range 0-1. The ground truth for all segmentation are in range 0 to 1.

For UNet (Multi-Frame approach) : Initially we tried to built a model with single frame UNet input and the model never learnt. Hence, we went ahead with multi-frame approach on the data-set. The data-set has the sequence of data of 180 frames where 4 in consecutive 5 frames are skipped. So in the training data-set there are 53 such 180 frame sequences. We have tried to exploit this temporal dimension by taking consecutive 5 frames and then calculating the absolute difference of the consecutive images leaving us with 4 images that shows the difference. The images of difference will be normalized between 0 and 1. The ground truth to this data would be 3rd image in the original 5 consecutive image.



(a) Image Input 1 (b) Image Input 2 (c) Difference of 2 consecutive images (d) Segmentation output/Ground truth

Figure 2: Data generator input and output

Tried 2 different experiments with this data-set while training the UNet model:

1. Averaged the difference frames(4) and using ground truth of 3rd frame.
2. Feeding as a 4 channel input to the model.

For Locating Objects Without Bounding Boxes(single frame approach) : The dynamic range of the data-set has short range in the given image. Also, the data-set has challenging day and night dynamic distribution. Hence we did normalize the histogram using CLAHE(Contrast Limited Histogram Equalization) algorithm. These images are used along with csv annotation of the center of each target. The CLAHE really helped to improve the training accuracy while training as it equalises the input data distribution.

3d CNN model(Multi-Frame approach): In this test, we used to feed in with 5 consecutive images[1] and the ground truth for the 3rd image to train the model. However, we found that the model is inefficient in the learning process with 2000 parameter when compares to UNet with 1.6M

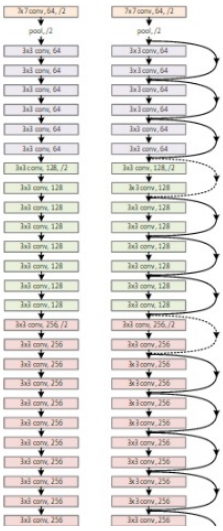
[1]The Data loader for 5 consecutive images and difference is a custom function we wrote to interface with Tensorflow's training protocol

parameters.Training for longer duration also did not helped. The model showed the same pattern as the UNet with single frame, and the gradients dies out eventually. The model we used is as below.

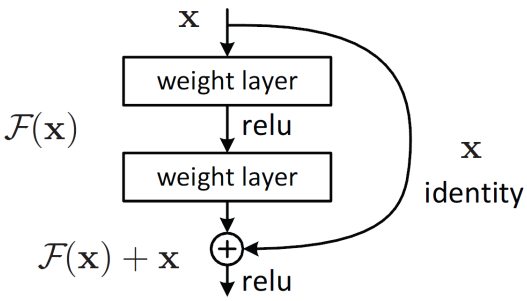
3 Classifier

The resent architecture was released by Microsoft Research Asia in 2015 with 3 realisations. ResNet50, ResNet101, ResNet 152 being the 3 realisations were able to achieve successful results in Imagenet and MS COCO competitions.

In resent architecture, the residual connections helps in greatly improving the gradient flow. This moreover helps us to allow training of much deeper models with hundreds of layers. The pertained network has an input image size of 224 by 224. As we can see in the left and right architectures of ResNet in the below figure, the left is a plain net and the right is the residual net which makes use of the concept of residual blocks shown in Figure 4 (b).



(a) Network Layer .



(b) Residual block intuition

Figure 3: Resnet Architecture

In our approach we take into the following training considerations Optimisation algorithm used here is SGD. As the weights are updated after looping each training sample, unlike each training data-set. We have observed in the most of the trained models with the available data, we see the loss decrease gradually and reach some plateau, thereby the possibility of reaching a local minima.

The problem could be the optimizer's old nemesis, pathological curvature. Pathological curvature is, simply put, regions of f which are not scaled properly. The landscapes are often described as valleys, trenches, canals and ravines. The iterates either jump between valleys, or approach the optimum in small, timid steps. Progress along certain directions grind to a halt. In these unfortunate regions, gradient descent fumbles.[1]

Now before training the network for 500 epochs, it is advisable to use the Xavier initialisation scheme.[2] Avoiding over-fitting is also a major concern. Hence, we can make use of a early stopping criterion to check if the testing error increases before reaching the final epoch. As we make use of necessary data augmentation, we can avoid the use of dropout as we already make use of a batch-form. The concept of using a residual block is the key to make better predictions

4 Detection

UNet Model: UNet architecture was introduced in May 2015 and was really useful in many of the binary segmentation where the model need to identify a particular pattern spatially. Also, the model uses skip connection and concatenation to preserve the residues from the previous layers which makes the gradient landscape smoother which will help the training process. We trained UNet model with input size $224x224$ and single channel input with the same output size dimensions. We

have batch-normalization on each layer and to make sure that the model is not over-fitting we used a dropout of 0.25 in each layer. This helped us to generalize on the data that we had so as to improve accuracy in the testing data-set.

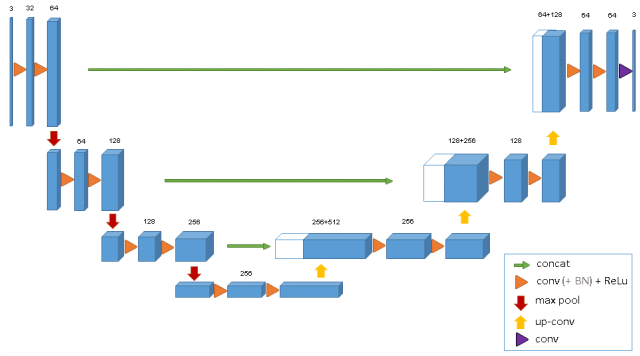


Figure 4: UNet Model of depth 4 and skip connections at each layer.

The initial approach was to train a UNet model with $depth = 5$ on each frames to get predictions for the corresponding segmentation results. However, because the data-set is very hard and differentiating the clutter from the target was really hard in the given image set, the model was not able to train adequately in the given training set. It dies out eventually even if we tries various training parameters and architectures. We used the training model with bath normalization and the learning rate was reduced for both $Adam$ and SGD. We used $reLU$ and $tanh$ for middle layers and $sigmoid$ for the last layer for output in range$[0 - 1]$. We concluded that the model was not able to distinguish between the clutter and the target.

Unet MultiFrame Approach: We have tried using the single images for UNet model with the ground truth annotation as described above. The model has input $(224, 224, 1)$ and $(224, 224, 1)$ with depth 4 and with residuals concatenated layers in the UNet model. We used $reLU$ and $sigmoid$ for middle and final layers with $Adam$ optimizer.
Further, we tried to train a 4 channel UNet model with single output segmentation. The 4 channels are absolute difference of 5 consecutive images as described earlier in section Data-set. The performance of this model was not as good as the single channel input.

3D CNN Model : The 3D model is as follows. It has fully convolutional model with $reLU$ activation for middle layers and $Sigmoid$ for the final layer with batch-normalization in each layer. The input to the model has size $250x200x4$ and with output size of $96x121x1$. The model was not able to learn attributed to less number of parameters. However, the computation required for the model is huge even with less parameters due to many 3D convolution operations.

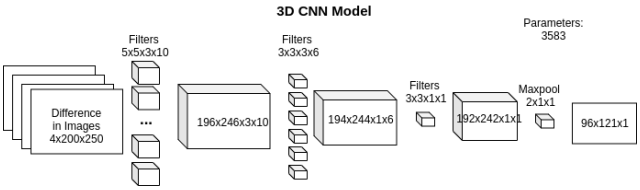


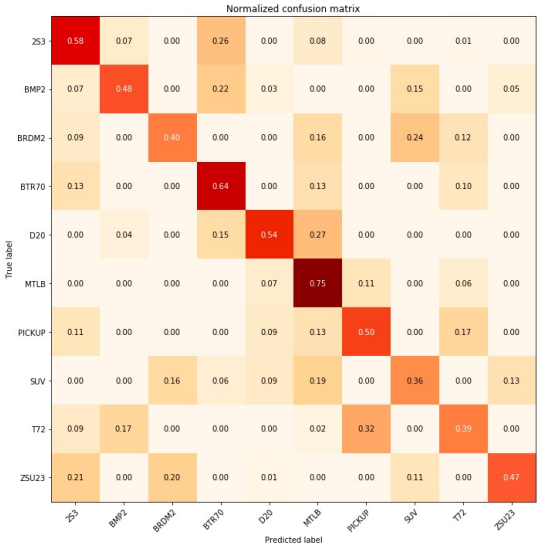
Figure 5: 3D CNN Architecture

5 Results

Under this section we have reported our numerical experiments to evaluate the classification, detection and tracking performances for our proposed models. We first report our classification results with a the confusion matrix and the accuracy for each class individually on figure 6. We then have also reported the results for detection on figure 7.

5.1 Classification

The classifier built on the ResNet101 model when tested on the 5410 test samples from 10 different classes gave an overall accuracy of 51.05 percent. The accuracy for each class individually is also shown in Figure 6 (b).



(a) Confusion matrix for classification.

Class	Prediction Accuracy
2S3	58
BMP2	48
BRDM2	40
BTR70	64
D20	54
MTLB	75
PICKUP	50
SUV	36
T72	39
ZSU23	47

(b) Prediction Accuracy

Figure 6: Classification Results

5.2 Detection

UNet : The most successful model that we trained was UNet with averaged 4 channel consecutive difference input with size $224x224x1$ as input channel and $224x224x1$ as output channel.

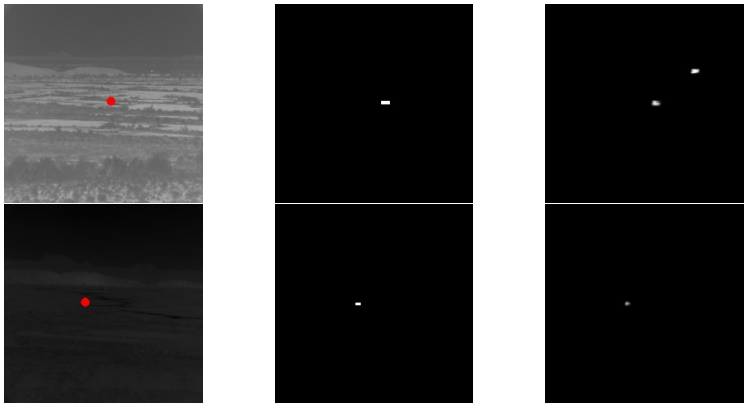


Figure 7: [1] Image input: 3^{rd} in 5 frames [2] Ground truth. [3] Prediction. -3.5 km range

We were able to achieve these results using the UNet with difference in images as input. In Row 1, there are false predictions due to clutters. The performance is always better during night time and low lit scenes, like in image Row 2, attributed to the fact that there are less variations during night for an infrared image. During day-time, these variations are prominent in difference map and thus accuracy is less in the testing data-set.

We tried to Qualitatively analyze the model based on the following parameters including $DiceCo-efficient, Recall, Precision, f1\_score$ etc.. These metrics helps to quantify our results. We trained our model on both the data-sets we created. ie, the $Dataset-1$ with range $[1000-$

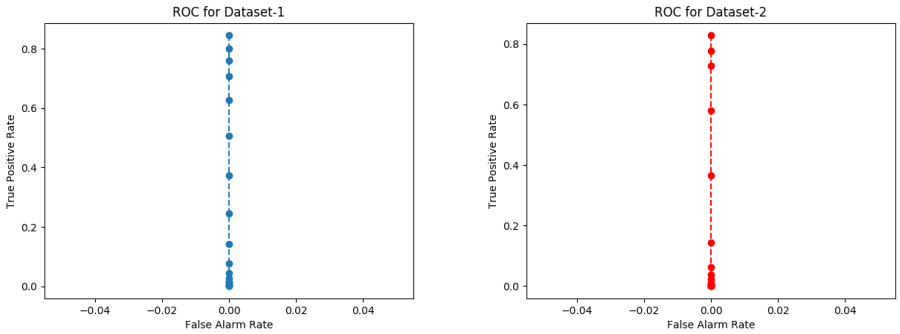


Figure 8: [1] ROC curve for detection: Dataset-1 [2] ROC curve for detection: Dataset-2

$2000]$ -training, $[2500 - 3500]$ -testing and the $Dataset - 2$ with range $[1000, 2000, 3000]$ -training, $[1500, 2500, 3500]$ -testing. Quantitative results of the proposed model is as follows for both the data-sets. We counted detection to be true if there exists an overlap between ground truth and

Table 1: Table for performance UNet- MultiFrame approach.

Metric	Dataset-1	Dataset-2
Average_dice_score	0.4484	0.706
Average_recall	0.535	0.6895
Average_precision	0.5001	0.7075
Average_F1_score	0.403	0.4333
Average_IoU	0.505	0.664

predicted values. However, since all of our data-set has a target, there exists no True_Negative or False_Positives, the false alarm rate coincides in a single line while plotting ROC curve.

6 Tracking

We used the Kalman filter tracker which uses a linear quadratic estimation function to model track of the data points that is being detected. We have used 2-dimensional tracking with a particular velocities in both directions for tracking and we have not used varying acceleration terms as we are assuming the targets moves with some certain acceleration. The Kalman tracker was able to learn the tracking of the points well with the given parameter initialization. We applied tracker on center of the targets to get the predictions and uses all prior values as well for tracking of that particular point. The uncertainties we used includes mean and standard variation along x,y and velocities in x and y directions and their error correlation.[4]

7 Conclusion

We decided on UNet since it is able to learn pixel-wise patterns in the data-set which is really important in this data-set as farther image sets in range $[3000, 3500]$ meters has really small targets. We have tried to train the model on each-frame to see the performance of the model. However, as mentioned above the weight value dies slowly. Proceeding with average of difference in 5 consecutive frame has made a huge difference in the same UNet model.

On trial basis we tried the exact model 'Locating Objects Without Bounding Boxes'[5] from Javier et. al. to see the performance of the model for single images. The results were not promising, however, they were close to the actual points which is attributed to the Hausdorff distance loss that they used in their UNet model. Also, we have tried a fully convolutions 3D CNN architecture as described above in section Detection to see if the model was able to learn. The input image fed to the model is 5 consecutive images. We have not tried the model with 4 consecutive image difference as there were only 3K parameters in the fully connected 3D CNN network. We would explore further in this

data-set on it's temporal dimensions to see where the model can be improved and developed further to produce robust results.

The loss function we used for this experiment is Binary cross-entropy and Mean Square Error loss. But looking at the paper *Locating Objects Without Bounding Boxes*,[6] we should have used Weighted Hausdorff distance in tandem with MSE or Binary cross-entropy for better detection of the targets. In first row of Fig 6., we have the model performing badly with two 2 detention which can be improved using the Hausdorff distance. However we might loose the bounding boxes, which is important in the experiment with classifier. Hence we did not introduced the Weighted Hausdorff distance as a metric for the detection. UNet will be able to give the actual bounding box that we require.

References

[1] Biboswan roy. Understanding adam vs sgd. *, https://medium.com/@Biboswan98/optim-adam-vs-optim-sgd-lets-dive-in-8dbf1890fbdc*, 2019.

[2] Yoshua Bengio Xavier Glorot. Understanding the difficulty of training deep feedforward neural networks. *,DIRO, Universit e de Montr eal, Montr eal, Qu ebec, Canada*, 2010.

[3] Philipp Fischer Olaf Ronneberger and Thomas Bro. U-net: Convolutional networks for biomedicalimage segmentation. *,CVPR*, 2019.

[4] Github repo. *https://iitmcvg.github.io/summer$_s$chool/$Session4/$*, 2019.

[5] Yuhao Chen Edward J. Delp Javier Ribera, David Guera. Locating objects without bounding boxes. *, CVPR*, 2019.

[6] Github repo. *https://github.com/javiribera/locating-objects-without-bboxes*, 2018.