Multi-object Detection in Night Time

Pavan Sai Vemulapalli, Ajay Kumar Rachuri, Heena Patel, Kishor P. Upla
Department of Electronics Engineering, Sardar Vallabhbhai National Institute of Technology (SVNIT)
{vpavansai24,ajaykumar.rachuri,hpatel1323,kishorupla}@gmail.com

***Abstract*— This paper discusses the work on detecting multi-objects such as person and car in thermal image captured during night time using deep learning architecture. Thermal images are superior to the visible images when it comes to the amount of useful information required to detect the objects during night time. Thermal imager uses radiation emitted by the objects to create an image and improve the visibility of objects in a dark environment. Contrast to that, visible image does not provide useful information in darkness. Hence, it is better to use thermal images to detect objects present in darkness. The state-of-the-art, Yolo-v3, deep learning convolutional neural network model is the latest version of the Yolo model in which the feature extraction layer contains a much deeper network. The results of detecting person and car in the thermal images obtained by the proposed model are compared with the results of Yolo- v3. Experimental results show that there is a significant improvement in detecting person and car in the thermal images in terms of mean average precision (mAP) using the proposed method.**

Keywords—Thermal imaging; Deep Learning; object detection; convolutional neural network; Yolo; Multi class detection.

I. INTRODUCTION

Surveillance systems have seen development in such a short span of time. Also, most major cities have been equipped with surveillance cameras which are located at tourist destinations, busy intersections, etc., to allow the local police and authorities to monitor public places. Due to the significant improvement in computer vision technologies and the neural network applications in object detection, now-a-days more interest is focused on making these surveillance systems automated. However, these surveillance systems are facing difficulties in night vision. Thermal imaging cameras can be used in surveillance applications in certain climatic conditions like fog, rain, and also in total darkness in which visible cameras captured poor images. Thermal images detect thermal radiation and they do not need a source of illumination to produce an image in the night and can see through rain, smoke, light fog (to a definite extent). Thermal imaging cameras create and shows little temperature variations visible. They are referred to as FLIR ("forward-looking infrared"). Once associated with additional cameras (for example, SWIR or a visible camera) multispectral sensors are possible, which takes the advantage of benefits of each detection band's capabilities. Thermal imagers cannot see through solid objects, nor can they see-through glass or perspex as both materials are having their own thermal signature and are opaque to long-wave infrared radiation. Classification and detection of objects in the thermal imagery have been an active and emerging research area in computer vision [3, 10, 18, 26]. Even this application is often extended to the domain of personal security, border protection, national security, and military surveillance operations [2] due to the global terrorist threat. Also, the importance of thermal images in the application of self-car driving is significant. Many kinds of research are being carried out in this domain to translate models into deployment in the real-world environments. The main aim of the object detection is to classify objects present in the image and to administer their actual position. Most of the efforts are placed on detecting objects and humans in the RGB images. Many successful machine learning algorithms have been developed for the detection of objects like full human figures [4] or human faces [25] in RGB images. Currently, models based on convolutional neural networks are the most successful models for detecting objects in RGB images. The development of the image recognition task started with the great achievement of AlexNet in the ImageNet Large Scale Visual Recognition Challenge in 2012 [15]. By using convolutional neural networks the performance of object detection in the RGB domain has been significantly increased. The main concern of object detectors using deep neural networks is how the spatial information about the object is to be acquired. One of the first approaches to solve this problem using deep learning is to extract candidate regions from the image using selective search, and subsequently classify these regions as if they were individual images. This way, the object location is given by the region from which it originates. This method was first proposed by Girshick et al. [8] who named it Regions with CNN (R-CNN). More amount of time is required by the model to train the network. To resolve a number of the drawbacks of R-CNN, Fast R-CNN [7] is introduced. In fast R-CNN, the image is being fed as an input to CNN to get a convolutional feature map rather than feeding the region proposals to the CNN. The region of proposals is identified from the convolutional feature map. The reason fast R-CNN is faster than R-CNN is that every time it is not needed to feed region proposals to the convolutional neural network. Instead, the convolution operation is performed one time per image and a feature map is generated from it. Both R-CNN and Fast R-CNN uses selective search to find out the region proposals. The performance of the network is affected by selective search as it is a slow and time-consuming process. Therefore, Shaoqing Ren et al. came up with Faster R-CNN [23], an object detection algorithm that permits the network to learn region proposals by eliminating the selective search algorithm. Similar to Fast R-CNN, image is provided to a convolutional network as input that provides a convolutional feature map. The separate network referred to as Region Proposal Network is employed to predict the region proposals instead of using a selective search algorithm on the feature map. Region of Interest (RoI) pooling layer is used to reshape the predicted region proposals which are then used to classify the image within the proposed region and predict the offset values for the bounding boxes. Most of the earlier object detection models use regions to locate the object in the image. The network does not examine the entire image. Instead, regions of the image that have higher chances of containing the object. You Only Look Once (Yolo) is an

object detection algorithm that is much different from the region based algorithms [20]. It predicts the presence of an object, as well as a bounding box, for a fixed size grid that tiles the input image, similar to that of DNN based regression [24] except that only a single pass through the network is sufficient for detection.

TABLE I. DARKNET-53 [22].

Type	filters	stride	Size	Output
Conv2d	32	3×3	1×1	320×320
Conv2d	64	3×3	2×2	160×160
(×1)res 1	–	–	–	160×160
Conv2d	128	3×3	2×2	80×80
(×2)res 2	–	–	–	80×80
Conv2d	256	3×3	2×2	40×40
(×8)res 3	–	–	–	40×40
Conv2d	512	3×3	2×2	20×20
(×8)res 4	–	–	–	20×20
Conv2d	1024	3×3	2×2	10×10
(×4)res 5	–	–	–	10×10

In Yolo, the bounding boxes and the class probabilities for these boxes are predicted by a single convolutional network. The bounding boxes having the class probability more than a threshold value is selected and used to identify the object if present in the image. The Yolo model is much faster than the other object detection algorithms. The major limitation of the Yolo algorithm is that it struggles with tiny objects present in the image. To handle small objects, the input image is up-scaled before fed to the network. From that point forward, many successful CNN architectures have been developed for the task of object detection. The abovementioned object detection methods depend on the models that have been trained on available standard datasets such as PASCAL-VOC, MS-COCO, and ImageNet. Less availability of such huge datasets in the thermal domain has limited the development of such frameworks on thermal images. Some researchers came up with a solution of using a transfer learning approach for object detection tasks in thermal images such as Abbott et al. [1] used a transfer learning approach, Yolo architecture trained on high-resolution thermal images containing vehicles and pedestrians is used for the classification of vehicles and pedestrians in low-resolution thermal images. Marina Ivai-Kos [12] also used a transfer learning approach with the Yolo framework to train a network on thermal images to detect persons. In this paper, we consider the task of object detection using the convolutional neural network in images captured using thermal cameras. The results obtained by the proposed model and Yolo-v3 are compared. For the detection task, we use Yolo-v3 [22] and the proposed model. Yolo-v3 is one of the faster object detection algorithms. So it is a good choice to use Yolo when there is a need for real-time detection, without loss of too much accuracy. The remainder of this paper is organized as follows. Section 2 gives a brief overview of early versions of the Yolo object detection models and the proposed model. Section 3 describes the experimental setup and details of the dataset. Section 4 discusses the results and gives comparisons of mAP scores and ends with a conclusion.

II. THE YOLO OBJECT DETECTOR

The paper on Yolo [20] depicts the object detection model that uses a single convolutional network to correctly predict bounding boxes of multiple objects in images as well as class confidences for those boxes simultaneously. The network architecture of Yolo model consists of 24 convolutional layers and two fully connected layers. The convolutional layers extract features in the images while the fully connected layers try to predict the bounding box coordinates and their class probabilities. The framework first

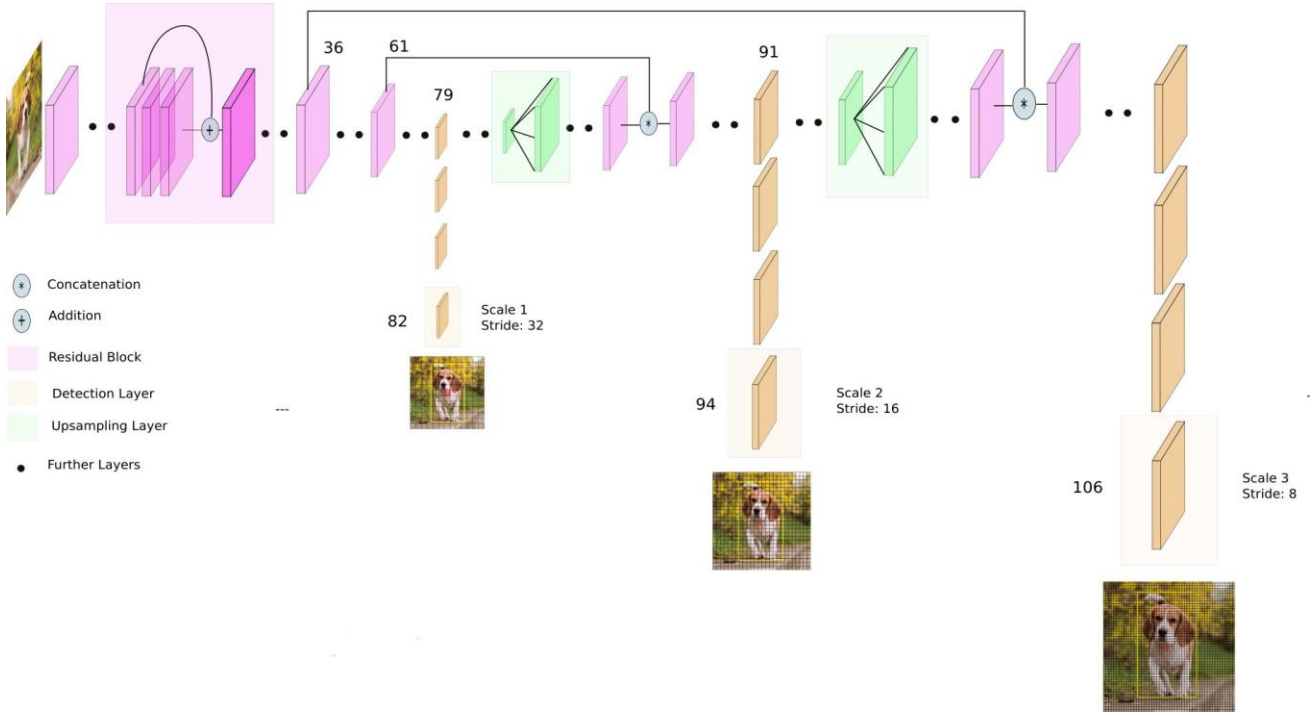


Fig. 1. Yolo-v3 architecture [14].

divides the input image into an S×S matrix. Two bounding boxes and their corresponding class confidences are associated with each grid cell, so at max two objects can be detected in a cell, and if an object is present in more than one cell, then the center cell is taken as a prediction holder for that object. When training the network, a bounding box with no objects will be assigned a confidence value of zero, a bounding box around an object has a confidence value that depends on the intersection over-union (IoU) score of the ground truth box and the bounding box. Yolo-v2 [13, 21] is the extension of the Yolo detector is faster and accurate than the previous version (Yolo). This is often as a result of Yolo-v2 uses some techniques that Yolo didn't use, like Batch-Normalization [11] and Anchor Boxes. Batch-Normalization is employed to normalize the outputs of hidden layers. This makes learning much faster. Anchor-Boxes is an assumption on the shapes of the bounding boxes. Since the shapes of objects to be detected don't vary such a lot, therefore there's no ought to realize boxes that do not seem like any of the objects we wish to detect. For example, to detect humans the shapes of anchor boxes are usually vertical rectangles and it is less likely that they are squares or horizontal rectangle. Hence, there is ought to search such boxes. This makes prediction much faster. It replaces five convolution layers of the original model with max-pooling layers and changes the approach of generating bounding box proposals. Rather than fully connected layers, predefined anchor boxes are used to predict the bounding box coordinates for every cell. To outline the anchor boxes, Yolo-v2 uses K-means clustering in a training set of ground truth bounding boxes where boxes translations are relative to a grid cell. One grid cell is responsible for detecting 5 bounding boxes, therefore it will detect up to 5 boxes on each grid cell. Yolo-v2 often struggled with the detection of small objects as there is a loss of fine-grained features as the layers down-sampled the input. To overcome this, Yolo-v2 used an identity mapping, concatenating feature maps from a previous layer to acquire low-level features. However, the architecture of Yolo-v2 is still lacking with some of the most important elements which are now introduced in most of the state-of-the-art algorithms.

Yolo-v3[22] incorporates residual blocks, skip connections and up sampling techniques. It uses a Darknet variant, which consists of 53-layer network as shown in Table I and it is trained on the COCO dataset [17]. These 53 layers are used for feature extraction. The newer architecture consists of residual skip connections inspired by ResNet [9].

TABLE II. SPECIFICATIONS OF THE PROPOSED NETWORK MODEL

layer	kernel	stride	Output size	Skip connection
Input	–	–	320×320×3	–
Conv2d	3×3	1×1	320×320×32	–
Conv2d	3×3	2×2	160×160×64	–
(×1)res 1	1×1	1×1	160×160×64	–
Conv2d	3×3	2×2	80×80×128	–
(×2)res 2	1×1	1×1	80×80×128	–
Conv2d	3×3	2×2	40×40×256	–
(×8)res 3	1×1	1×1	40×40×256	–
Conv2d	3×3	2×2	20×20×512	–
(×8)res 4	1×1	1×1	20×20×512	to concat
Conv2d	3×3	2×2	10×10×1024	–
(×4)res 5	1×1	1×1	10×10×1024	–
CB 1	1 3	1×1	10×10×1024	to upsample
Conv2d	×1,3×	1×1	10×10×255	–
	1×1			
Upsample	–	–	20×20×512	from CB 1
concat	–	–	20×20×512	from res 4,Upsample
CB 2	1×1,3×3	1×1	20×20×512	to upsample
Conv2d	1×1	1×1	20×20×255	–

For detection task, 53 extra layers are stacked onto it, summing up a total of 106 layers fully convolutional underlying architecture for Yolo-v3. This is the reason behind the slowness of Yolo-v3 compared to previous versions of Yolo. The architecture of Yolo-v3 is shown in Fig. 1.The detections at three different scales is the most important feature of Yolo-v3. The network extracts features from these scales using a similar concept to feature pyramid networks [16]. In Yolo-v3, 1×1 kernels are applied on feature maps of three completely different scales at three different layers within the network for detection functions. The dimension of the detection kernel is 1×1×(B×(5+C)). Here, B is the maximum number of bounding boxes predicted by a cell on the feature map, C denotes the number of classes, and 5 is for the 4 bounding box coordinates and one object class confidence score. For Yolo-v3 trained on COCO, B=3, and C=80, therefore the size of the kernel is 1×1×255. The feature map produced by the kernel obtained above has same width and height of the previous feature map and has attributes detected along with the depth. The stride of the layer or the network is defined as the proportion by which it downsamples the input. For an input image of size 320×320, three scale prediction is done by Yolo-v3, which are correctly obtained by downsampling the dimensions of the input image by 32, 16 and 8 respectively. The primary detection is made at the 82nd layer. The image is down-sampled by the network for the first 81 layers, such that the 81st layer contains a stride of 32. The feature map obtained here would be of size 10×10. The first detection is made here using the 1×1 detection kernel, giving us a 10×10×255 detected feature map. Then, the feature map from 79th layer is subjected to a number of convolutional layers before being up sampled by 2× to dimensions of 20×20. The feature map obtained above is then depth concatenated with the feature map from 61st layer. Then the combined feature maps are again subjected to a few 1×1 convolutional layers to fuse the features from the previous layer (61). Then, the second detection is made at the 94th layer, yielding a detection feature map of 20×20×255. An analogous procedure is followed once more, where the feature map from layer 91 is subjected to few convolutional layers before being depth concatenated with a feature map from layer 36. Like before, a number of 1×1 convolutional layers follow to fuse the information from the previous layer (36). We make the ultimate detection at 106th layer, yielding a feature map of size 40×40×255. Detections at completely different layers facilitate to deal with the difficulty of detecting small objects in Yolo-v2. The up sampled layers coupled with the previous layers facilitate to preserve the fine-grained features that help in detecting small objects.

In total nine anchor boxes are used by Yolo-v3, three for each layer where detection takes place. One ought to use K-Means clustering to come up with nine anchors to train Yolo-v3 on their own dataset and organize them in descending order according to dimension. The three biggest anchors are assigned to the primary scale, successive three for the second scale, and the last three for the third scale. For an input image of the identical size, Yolo-v3 predicts more number of bounding boxes compared to Yolo-v2. For the

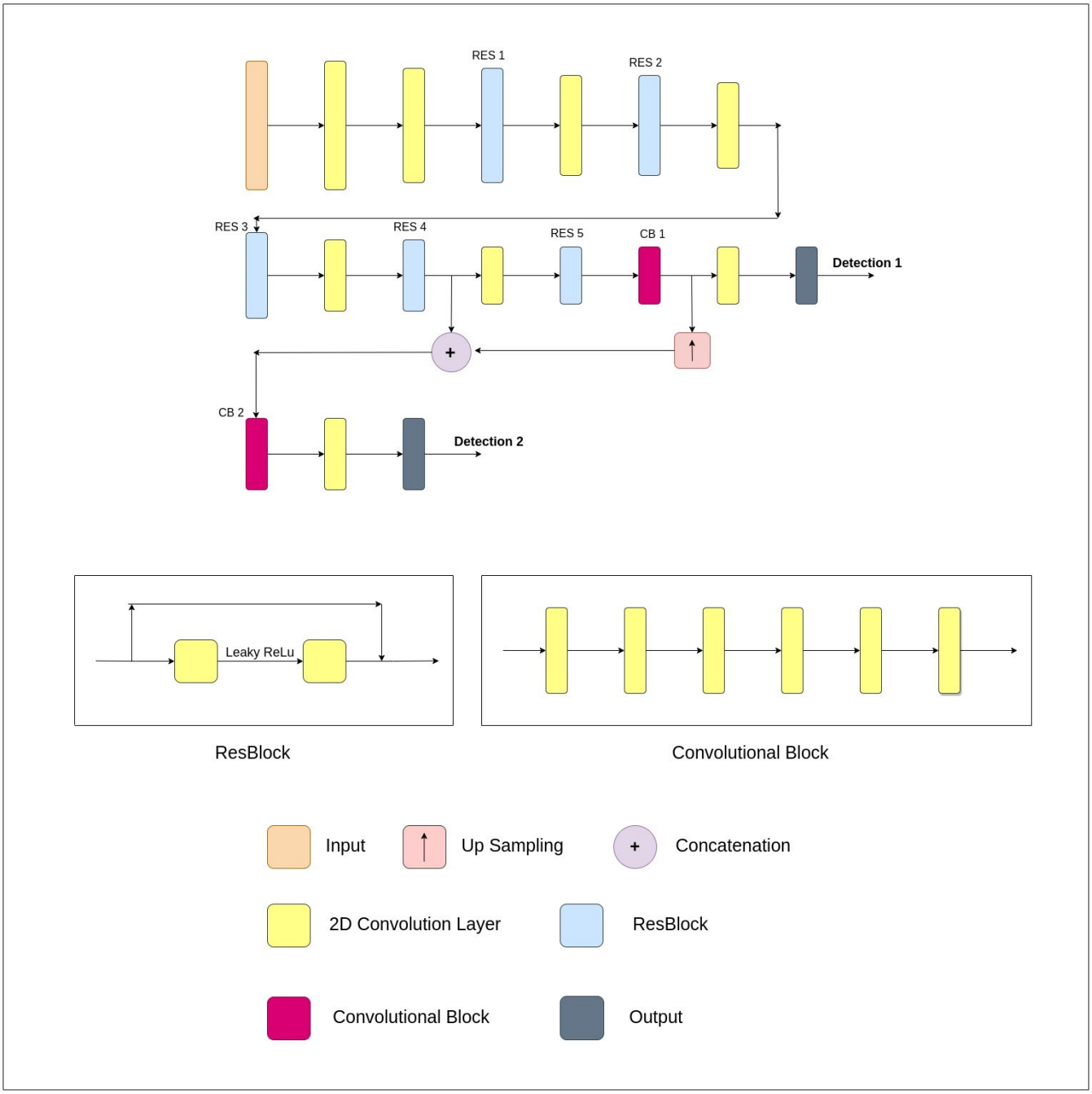


Fig. 2. Network Architecture of the proposed model.

image resolution of 320×320, Yolo-v2 predicts 10×10×5=500 boxes. At every grid cell, five anchors are used to detect five bounding boxes. On the other hand, Yolo-v3 predicts bounding boxes at three completely different scales. For the same image of size 320×320, 6,300 bounding boxes are predicted. This suggests that Yolo-v3 predicts 12× the number of boxes predicted by Yolo-v2. Hence, Yolo-v3 is slower compared to Yolo-v2. The classification methodology has additionally been changed in Yolo-v3. Earlier in Yolo, soft-max classification is used by the authors to classify the confidence scores and take a class with the utmost score to be the class of the object contained within the bounding box. Soft-max classes rest on the belief that classes are mutually exclusive, or in simple words, if an object belongs to one class, then it cannot belong to the other class. Now, multi-label classification is employed. In this case, an object may belong to more than one class simultaneously, which is achieved by replacing the soft-max classification with logistic regression. Logistic regression predicts each class score and multiple labels for an object are predicted by using a threshold value. Classes with confidence scores higher than this threshold value are assigned to the bounding box. The architecture of the proposed model shown in Fig. 2 increases the speed with acceptable levels of accuracy. It is the shortened version of Yolo-v3 formed by removing one of the object detection blocks. Out of the three detection blocks of different scales, the one with the larger scale which is used for detecting small objects is removed.

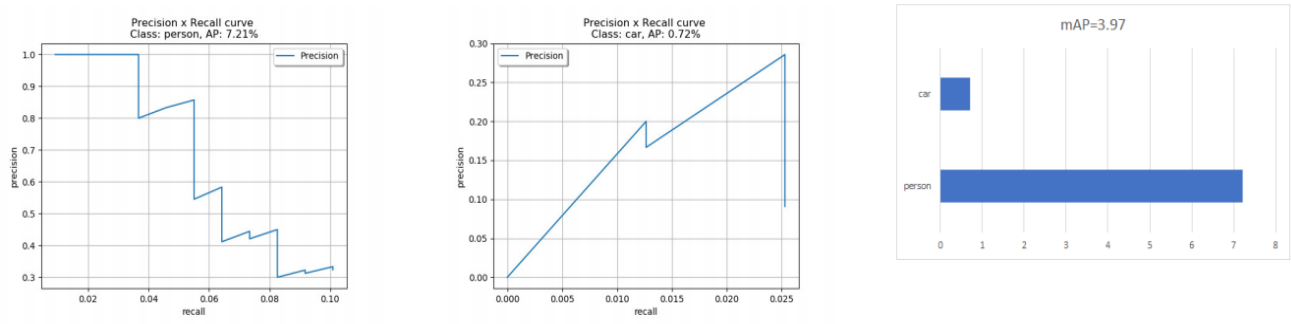


Fig. 3. Precision/recall curves and mAP score of classes person and car for Yolo-v3 (trained on COCO).

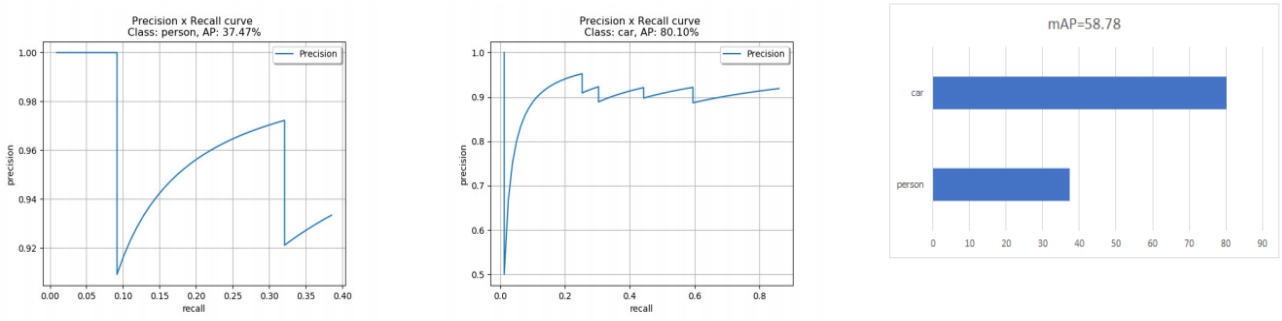


Fig. 4. Precision/recall curves and mAP score of classes person and car for proposed model (trained on COCO and thermal dataset).

The reason for removing that block is, in thermal images small objects don't give proper structure and appears as one heated spot and gets dominated by the surrounding which gets very difficult for the model to detect and there is a chance that it might get false detection. Specifications of the proposed network architecture are depicted in Table II. The proposed model predicts boxes at 2 different scales. For the same image of 320×320, the proposed model predicts 1,500 bounding boxes. This concludes that Yolo-v3 predicts 4× the number of boxes which are predicted by the proposed model. Hence, Yolo-v3 is slow compared to the proposed model.

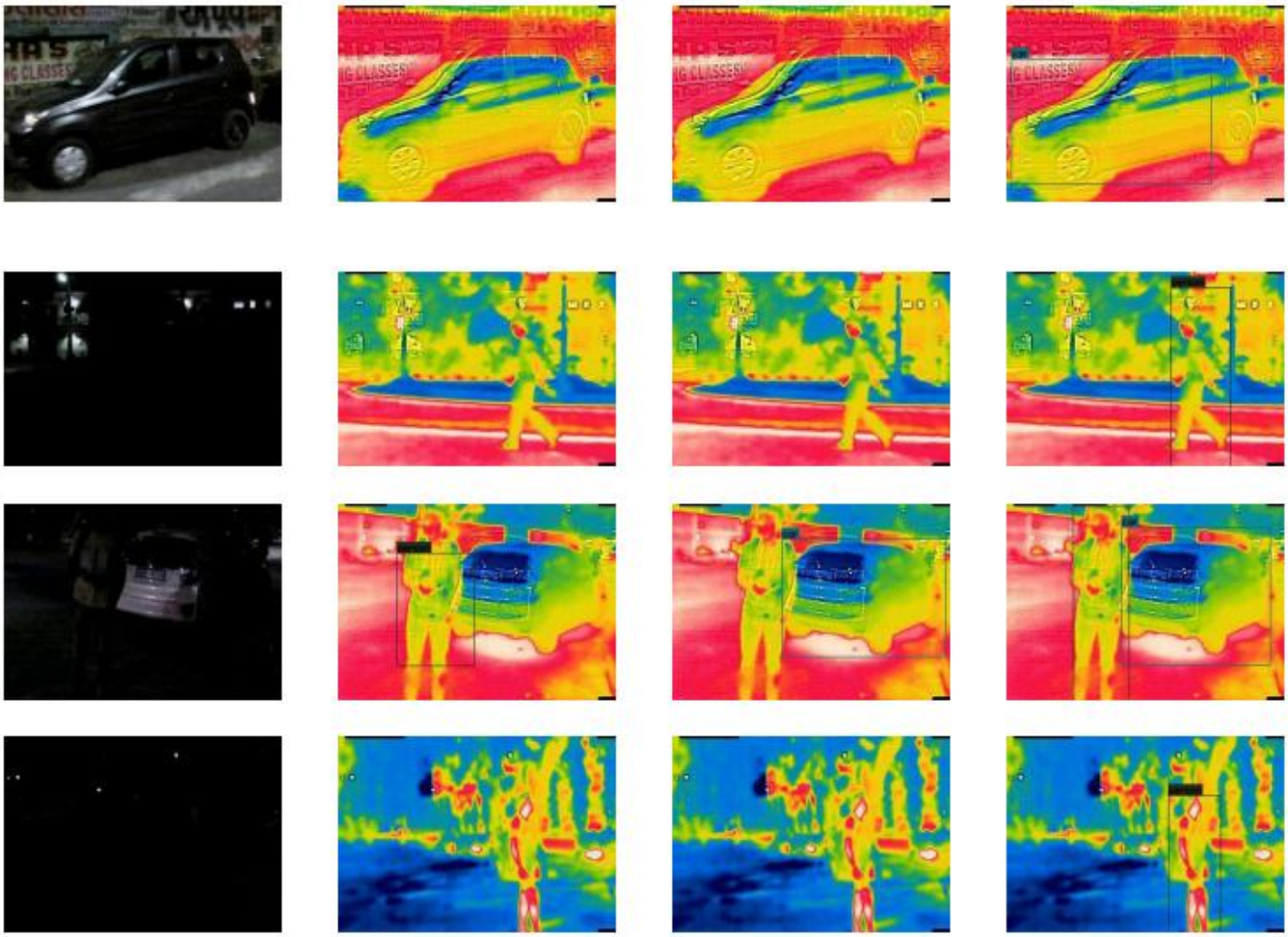
III. EXPERIMENTAL SETUP

In order to evaluate the potential of the proposed model, the experiments are carried out on the images captured under various conditions. All experiments have been performed on the system with the following configuration: Intel 7th Generation i7-7700k processor, 3.60GHz, GPU NVIDIA GeForce GTX 1070, 8GB GPU. The TensorFlow libraries are used as a backend to the implementation of the proposed network architecture. We are primarily focusing on detecting two classes, person and car in thermal images.

DATASET: We use the FLIR E8-XT camera to prepare a dataset of images in the night time under different weather conditions. The resolution of the camera is 320×240 pixels. We collected images of cars and persons both stationary and moving under different lighting and weather conditions. Finally, the images taken were manually annotated using VGG Image Annotator [5]. We compare the performance of three types of networks. First is the state-of-the-art Yolov3 model pre-trained on a COCO image dataset [17]. Second is the extension of Yolo-v3 with additional training on our thermal image dataset. The third is the proposed model trained on thermal images of our dataset. We compared the performance using the mean average precision (mAP) which is the one used as a performance metric in PASCAL VOC 2012 competition [6].

IV. RESULTS AND DISCUSSION

Before going for the comparison of results, some important terms related to mAP are discussed herewith. Precision is the fraction of relevant instances among the retrieved instances also called positive predictive value. Recall is the fraction of the total amount of relevant instances that were actually retrieved which is also called sensitivity. Average precision (AP) [19] is a popular metric to measure the accuracy of object detectors such as Fast R-CNN, Faster RCNN, SSD, etc. It finds the area under the precision-recall curve. It computes the average precision value for recall value over 0 to 1. AP for each class is calculated separately. All the predictions made for the respective class in all the images are collected and according to the predicted confidence level ranked in descending order. The prediction is correct if IoU is greater than the threshold value. After arranging them in descending order to calculate precision and recall for each class, draw the precision versus recall curve. Recall value increases as prediction ranking goes down while precision is having a zigzag pattern, it increases with true positives and decreases with false positives. To smooth out the zigzag pattern each of the precision value is replaced with the max precision value to the right of that recall level. AP of that class is given by the area under the obtained curve. Recall and precision value lies between 0 and 1. Therefore, AP falls between 0 and 1. mAP is the average of AP. In our context, we calculate the AP for both classes and average them. Fig. 3 presents the mAP score for the original Yolo-v3 model that is not trained with the thermal dataset. Fig 4 presents the mAP score for the proposed model trained with the thermal dataset. Here, we can observe that the mAP score of the Yolo-v3 is 3.97% which is very less compared to the proposed model which is 58.78% as one can be seen in Fig. 4.



(a) Visible (b) Yolo-v3 (trained on COCO) (c) Yolo-v3 (trained on COCO & thermal dataset) (d) Proposed model

Fig. 5. Experimental results obtained using the different scenarios. Results of multiple detection using (a) visible (b) Yolo-v3 (trained on COCO) (c) Yolo-v3 (trained on COCO & thermal dataset) and (d) Proposed model, respectively.

Images displayed in the results section are acquired in different lighting conditions (Fig. 5). We compare the results of the proposed model with Yolo-v3 (trained only on COCO dataset) and Yolo-v3 (trained on both COCO and thermal dataset). The first row of Fig. 5 contains the image of a car under dull light. Both models of Yolo-v3 which are displayed in Fig. 5(b,c) failed to detect the car present in the image (false-negative detection). Proposed model detects the car present in the image (see Fig. 5(d)). Even it can be seen from the Average Precision (AP) of the class car. Proposed model has the highest AP for the class car. Results of the second row of Fig. 5 is taken at low light condition, normal object detectors trained on the RGB image dataset are unable to find the person present in the image using the visible image as input (see Fig. 5(a)). Again, both Yolo-v3 models failed to identify the person present in the image (false-negative detection) i.e. Fig. 5(b,c). Proposed model which is displayed in Fig. 5(d) detects the person present in the image. Even Yolo-v3 which is trained on both COCO and the thermal dataset is unable to find the person. Third row of Fig. 5, is taken in low light conditions and it contains both the classes car and person. Yolo-v3 (Fig. 5(b)) detects the person present in the image but it misses some of the important features of the person. It is unable to find the car present in the image (false-negative detection). Yolo-v3 trained on the thermal dataset i.e. Fig. 5(c) detects the car present in the image but is unable to find the person present in the image (false-negative detection). Proposed model is able to detect both car as well as person in the image (see Fig. 5). Hence, the proposed model is better for multi-object detection. The images in the last row of Fig. 5 are taken in darkness, it contains a person. Both Yolov3 models failed to identify the person present in the image (false-negative detection). Proposed model (see Fig. 5(d)) detects the person present in the image. Proposed model is detecting both the person and car present in the thermal image where both Yolo-v3 models are failed to detect.

V. CONCLUSION

In this paper, we study and apply the deep learning methods available for detection on thermal images. Our idea was to collect images for the dataset in night time under low light conditions, and also we wanted to include moving and stationary objects. As our purpose of the experiment was on surveillance, we decided to collect images of person and car as they are of major concern.

We tried to relate the results of different approaches we made in detecting persons and cars in thermal images. Our initial idea was to use state-of-the-art Yolo-V3 architecture to detect two basic classes, person and car in thermal images. Even thermal images greatly differ from RGB images in

appearance we assumed that Yolo-v3 trained on the COCO dataset will still give a reasonable baseline for thermal images, but the obtained mAP was 3.97% which is very poor. So we trained the model with our custom thermal dataset and the results were significantly better with the mAP score of 58.84%. Further, we made a few changes to the model by removing one of the blocks with an intention to reduce the number of parameters. The results obtained are relatable to the previous one with the mAP score of 58.78%.

Through this experiment, we observed that with further additional training, Yolo is giving significantly much better results. We further plan to extend this experiment by focusing on the effect of different hyperparameters on this model and also investigate how different weather, range and lighting conditions going to affect the result. The idea is to make the model much better so that it can be implemented in real-time as this application has an enormous scope in military, vehicles and in alarm systems in restricted areas.

REFERENCES

[1] R. Abbott, J. Del Rincon, B. Connor, and N. Robertson. Deep object classification in low resolution lwir imagery via transfer learning. In *Proceedings of 5th IMA Conference on Mathematics in Defence*, volume 2, 2017.

[2] M. Bertozzi, A. Broggi, C. H. Gomez, R. Fedriga, G. Vez- zoni, and M. DelRose. Pedestrian detection in far infrared images based on the use of probabilistic templates. In *2007 IEEE Intelligent Vehicles Symposium*, pages 327–332. IEEE, 2007.

[3] T. P. Breckon, A. Gaszczak, J. Han, M. L. Eichner, and S. E. Barnes. Multi-modal target detection for autonomous wide area search and surveillance. In *Emerging Technologies in Security and Defence; and Quantum Security II; and Un- manned Sensor Systems X*, volume 8899, page 889913. In- ternational Society for Optics and Photonics, 2013.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. 2005.

[5] A. Dutta and A. Zisserman. The vgg image annotator (via). arXiv preprint arXiv:1904.10699, 2019.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303– 338, 2010.

[7] R. Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[10] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1037–1045, 2015.

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[12] M. Ivasi˘ c-Kos, M. Kri ´ sto, and M. Pobar. Human detection ˘ in thermal imaging using yolo. In Proceedings of the 2019 5th International Conference on Computer and Technology Applications, pages 20–24. ACM, 2019.

[13] K. Jo, J. Im, J. Kim, and D.-S. Kim. A real-time multi-class multi-object tracker using yolov2. In 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pages 507–511. IEEE, 2017.

[14] A. Kathuria. What's new in yolo v3? https://towardsdatascience.com/yolo-v3-object-detection53fb7d3bfe6b. Accessed: 2019-07-05.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[16] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and ´ S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ´ mon objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.

[18] S. Mangale and M. Khambete. Moving object detection using visible spectrum imaging and thermal imaging. In 2015 International Conference on Industrial Instrumentation and Control (ICIC), pages 590–593. IEEE, 2015.

[19] I. D. Melamed, R. Green, and J. P. Turian. Precision and recall of machine translation. In Companion Volume of the Proceedings of HLT-NAACL 2003-Short Papers, pages 61– 63, 2003.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[21] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.

[22] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[24] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In Advances in neural information processing systems, pages 2553–2561, 2013.

[25] P. Viola, M. Jones, et al. Rapid object detection using a boosted cascade of simple features. CVPR (1), 1(511-518):3, 2001.

[26] T. T. Zin, H. Takahashi, and H. Hama. Robust person detection using far infrared camera for image fusion. In Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), pages 310–310. IEEE, 2007.