Mini-Project Report
Sensitive Regions for Black-Box Adversarial Attacks

Sowrya Gali
B.Tech 3rd Year
Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Abstract

Deep neural networks are known to be vulnerable to adversarial attacks. Most of the emphasis, in adversarial attacks, has been on image data. The study on adversarial attacks for videos has started recently, and much work has been done in the area of White-Box attacks. In this project, we tried to visualize the saliency heat maps for YOLOv3 model, which aids in developing adversarial attacks on datasets like MS-COCO, ImageNet, etc.

Introduction

The Convolutional Neural Networks are susceptible to adversarial attacks i.e., a cleverly crafted noise is added to the image so that the network is fooled and misclassifies it. These adversarial attacks can be broadly classified into two types : (i) White-Box Attacks and (ii) Black-Box Attacks. In white box attacks, one has access to the model architecture, dataset used by it, weights of the model etc. and using this information adversarial noise is generated. Whereas in Black-Box attacks, one usually has no access to model's parameters and hence uses a substitute model locally to craft examples by performing White-Box attacks on the local model.

Ever since adversarial attacks were proposed, much emphasis has been on image data and numerous attacking strategies and defense techniques were developed focusing on the image data. The study of attacks and defenses for video data and autonomous driving has been started recently.

In this project, we aim to detect key regions in an image that are responsible for classification and localization, so that we can perform attacks, using this information, on the dataset to fool robust models like YOLOv3[13]. Most of the works in this area are focused on developing techniques generating perturbations for general CNN architectures and they cannot perform well on robust architectures like YOLO, Faster-RCNN[14], SSD[9], etc. evident from figure 1.

So to attack datasets, in order to fool such robust architectures, needs more investigation into the regions which activate neurons for the detection of object and it's class. Many visualization techniques have been proposed to detect feature maps for general CNNs and of these, GradCAM++ is considered to be state-of-the-art.

But GradCAM++, and it's predecessors are designed keeping in mind of general convolutions networks like VGG-16.

Figure 1: *Top to bottom: Predictions of YOLOv3, faster-RCNN, and SSD on original image(left) and attacked image(right). The PGD attack is used to craft the adversarial samples using ResNet-51 as surrogate network.*

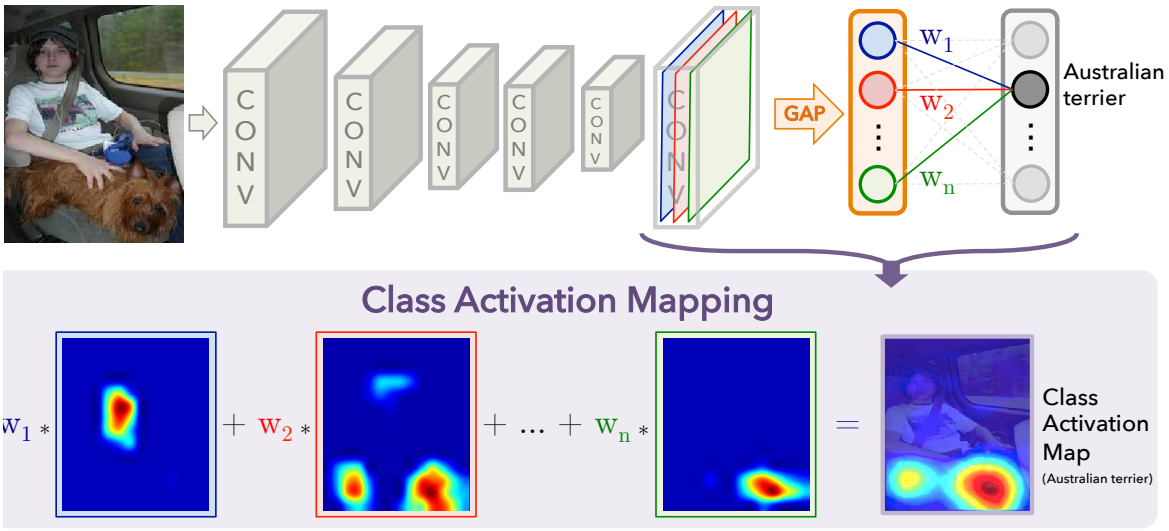

Figure 2: Schematic visualization of CAM model showing how scores are mapped back from GAP layer.[22]

Though produce good visualizations with excellent localization capabilities, there primarily based on convolutional architectures and there is no straight forward way to generalize them to non-trivial models like YOLOv3.

In this project, we explore possible ways of extending GradCAM++[1], to non-trivial architectures and how to utilize this information to perform attack on the dataset.

Prior Work

A lot of work has been done in the area of adversarial attacks and defense techniques. [18] made the intriguing discovery of adversarial example's misclassification. Later many approaches have been proposed to create adversarial samples, [3] proposed FGSM, [10] proposed PGD attack, [4] proposed SimBA attack etc.

Recently, more advanced attack strategies have been proposed like [6] [12] [16] [20] [2].

Ever since neural networks became popular, many efforts have been made to understand how these black-box models work under the hood. [21] made one of the first attempts in this regard. [22] followed up with a novel approach of using Global Average Pooling(GAP) values as the saliency maps with each channel having weight same as that of it's counter part in the adjacent flattened layer, shown in figure 2. Later, [15] came up with GradCAM model, which utilizes class conditional probabilities fused with pixel-based gradient visualization techniques like Deconvolution[21] and Guided Backpropagation (GBP)[17] to generate finer Heat Maps that generalize well non-GAP architectures. [1] came up with an improved version of GradCAM, called GradCAM++, which shows good localization properties on multi-object images and it is considered as one of the *state-of-the-art* visualization techniques for general CNN architectures.

Experimental Details

We first started to assess how a simple PGD attack on MS-COCO dataset could affect performance of YOLO*v3*. The results of this experiment are shown in figure 3. As evident from results, YOLO*v3* is robust against these attacks. Hence we need to look into which parts of the image are responsible for predictions the model to localize attacks to these regions, and which results in better attack success rate.

To understand the failure of PGD attack, we contrasted GradCAM++ generated heat maps of original image and attacked image. As it is evident, the attack could only slightly disperse regions of heat and there is no change especially in multi-object case as shown in figure 4.

In order to come up with a more successful attack we started investigating which regions play crucial role in determining objectness and class scores in robust models like YOLO*v3*. Towards this end, we started working on GradCAM++. But the problem with such techniques is they are engineered keeping in mind the general CNN architectures which are convolutional layers stacked upon fully connected layers and final layer gives out class conditional probabilities.

But architectures like Darknet (backbone of YOLO*v3*) are not straight forward since they embed bounding box information into the layers, and predictions are given at various stages, moreover the structure of prediction is non trivial to infer analogy with formulations of GradCAM++.

An example prediction structure of YOLO*v3* is shown in figure 5(a) and architecture in figure 5(b). An image goes through a series of residual blocks and down sampling is done through increasing stride. After reaching a fixed scale, we get a grid with information about bounding boxes stacked sequentially as shown in figure 5(a). Then this grid is up sampled to go get two more predictions with different number of grid boxes as shown in figure 5(b).

Since our focus is on video data for autonomous driving, we studied how well GradCAM++ performs on the traffic symbols and we tried to answer the question: "Do we need to formulate a new approach for robust models like YOLO*v3* or GradCAM++ is good enough to proceed?".

Figure 3: A simple PGD attack is performed on some examples from MS-COCO dataset using ResNet-51 as a surrogate model. The YOLO*v3* is quite resistent to this simple attacks owing to its highly complex architecture.

To answer this we chose GSTRB[5] dataset and trained VGG-16 model on it. Using this model, we generated heat maps for this dataset and result were satisfactory, as can be seen from figure 6.

Since GradCAM++ gives good visualizations for traffic signs and it also claimed to have good localization properties[1] on mutli-object scenario, we studied how well these localized regoions can be leveraged with objectness score heat maps from YOLO*v3*. To obtain the "objectness" heat maps from we used the object score channels of each anchor box for every grid. And then these scores were interpolated back to image to understand where exactly the model looks for object within the bounding box.

To conduct this experiment, we used CUB-2011[19] dataset, with VGG-16 trained on ImageNet and YOLO*v3* trained on MS-COCO dataset. The results include 2 images for each image input, where one of them signifies heat map from YOLO*v3* as discussed above and the other one is GradCAM++ heatmap and both with the bounding box drawn onto the heat maps to understand their layout with respect to the localization by the non-max supressed bounding box of YOLO*v3*. The results are shown in figure 7.

As evident from the heat maps we need to combine these two aspects to detect more effective regions for attack purposes.

Technologies and Resources

Most of the architectures are implemented using PyTorch. Except GradCAM++, which uses TensorFlow. YOLO*v3* is implemented from scratch[7] in PyTorch to extract "objectness" heat maps. The PGD attack is designed in PyTorch with reference to the tutorials[11]. The VGG-16 nets on GSTRB and CUBD are trained locally using PyTorch. The GradCAm++ implementation is adopted from the paper itself, with modifications required on task-to-task basis.

Most of the computations are run locally on a system with Intel i7 processor, 8GB RAM, and Nvidia GeForce GTX 1050Ti.

Results and Conclusion

In conclusion, we can see that attacking datasets with simple attacks like PGD do not give satisfying success rates and we need to look deeper into regions of activation of the datasets with respect to robust models like YOLO*v3*.

So in this direction, we contrasted how visualization techniques like GradCAM++, are specific for general CNNs and we need to come up with a hybrid approach which blends both GradCAM++ heat maps with YOLO*v3*'s objectness heat maps to come with finer and feasible regions for attacks.

Future Work

We are thinking of extending this approach and devise new formulation for architectures like YOLO*v3* and utilize these mappings to create noise patches that could fool the object detectors.

References

[1] Aditya Chattopadhay et al. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (Mar. 2018). DOI: 10.1109/wacv.2018.00097. URL: http://dx.doi.org/10.1109/WACV.2018.00097.

[2] Ranjie Duan et al. *Adversarial Camouflage: Hiding Physical-World Attacks with Natural Styles*. 2020. arXiv: 2003.08757 [cs.CV].

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[4] Chuan Guo et al. *Simple Black-box Adversarial Attacks*. 2019. arXiv: 1905.07121 [cs.LG].

[5] Sebastian Houben et al. "Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark". In: *International Joint Conference on Neural Networks*. 1288. 2013.

[6] Yunhan Jia et al. *Enhancing Cross-task Transferability of Adversarial Examples with Dispersion Reduction*. 2019. arXiv: 1905.03333 [cs.LG].

[7] Ayoosh Kathuria. *Tutorial on implementing YOLO v3 from scratch in PyTorch*. Dec. 2019. URL: https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/.

[8] Ayoosh Kathuria. *What's new in YOLO v3?* Apr. 2018. URL: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b.

[9] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *Lecture Notes in Computer Science* (2016), pp. 21–37. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[10] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2019. arXiv: 1706.06083 [stat.ML].

[11] Zico Kolter Madry and Aleksander. URL: https://adversarial-ml-tutorial.org/.

[12] Ali Rahmati et al. *GeoDA: a geometric framework for black-box adversarial attacks*. 2020. arXiv: 2003.06468 [cs.CV].

[13] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018. arXiv: 1804.02767 [cs.CV].

[14] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV].

[15] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[16] Yucheng Shi, Siyu Wang, and Yahong Han. *Curls & Whey: Boosting Black-Box Adversarial Attacks*. 2019. arXiv: 1904.01160 [cs.CV].

[17] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net*. 2015. arXiv: 1412.6806 [cs.LG].

[18] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].

[19] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[20] Chaowei Xiao et al. *Generating Adversarial Examples with Adversarial Networks*. 2019. arXiv: 1801.02610 [cs.CR].

[21] Matthew D Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*. 2013. arXiv: 1311.2901 [cs.CV].

[22] Bolei Zhou et al. *Learning Deep Features for Discriminative Localization*. 2015. arXiv: 1512.04150 [cs.CV].

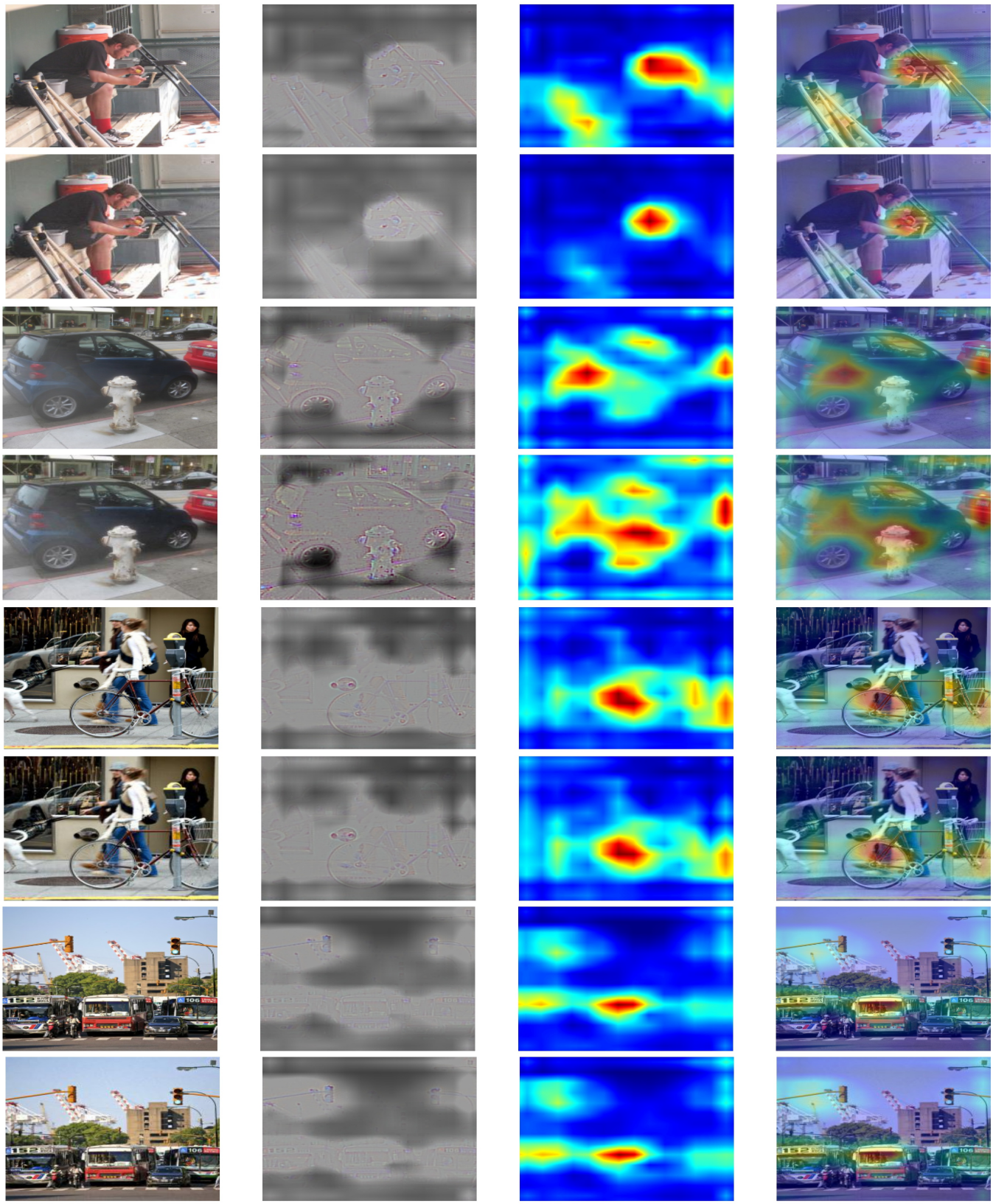

Figure 4: Heat maps generated by GradCAM++ on original and attacked images. In every pair, the first one is original and second one is attacked.

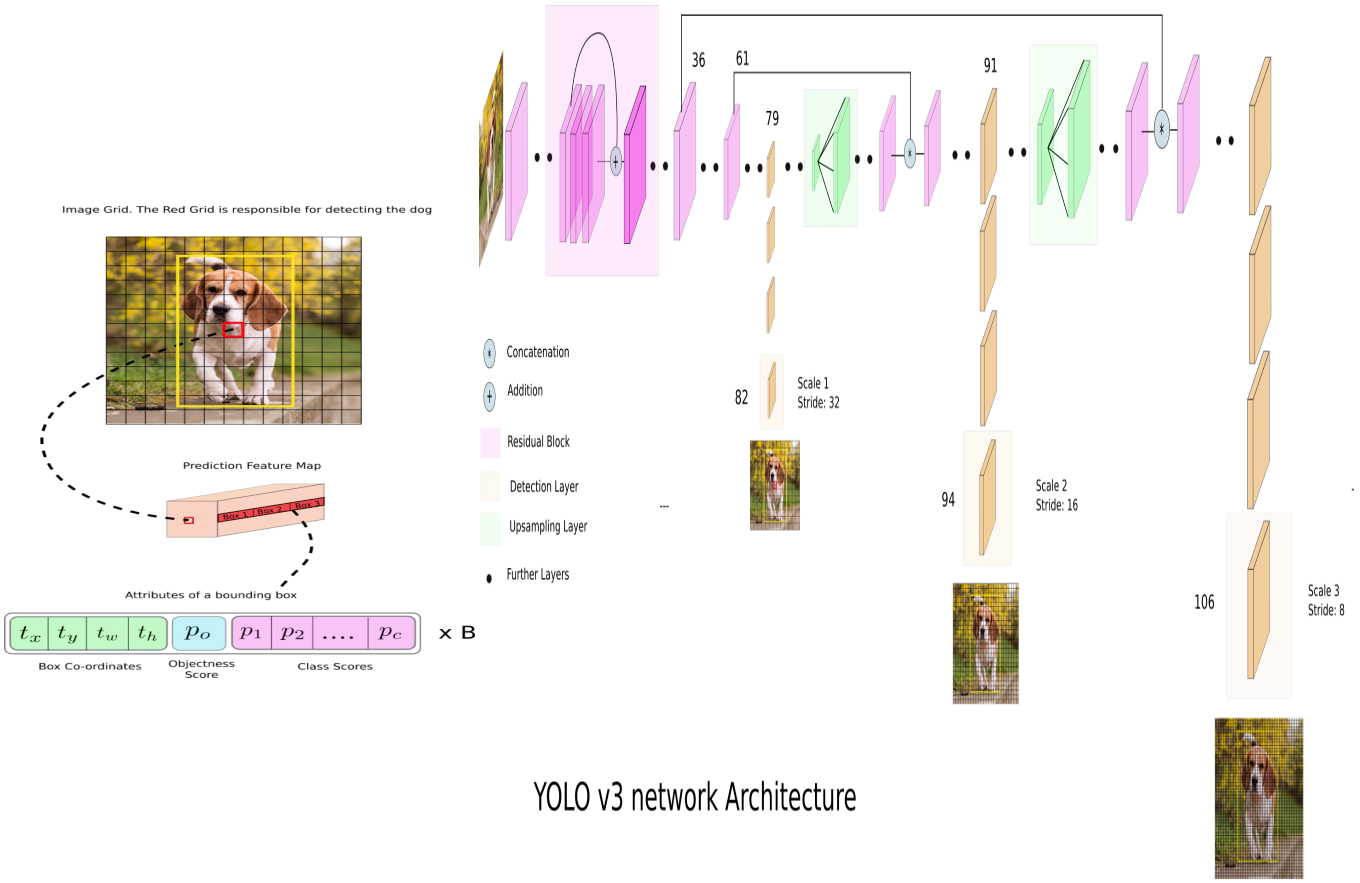

Figure 5: (a) An example prediction from YOLO*v3* on 13x13 grids [7]. (b) Predictions from YOLO*v3* [8].

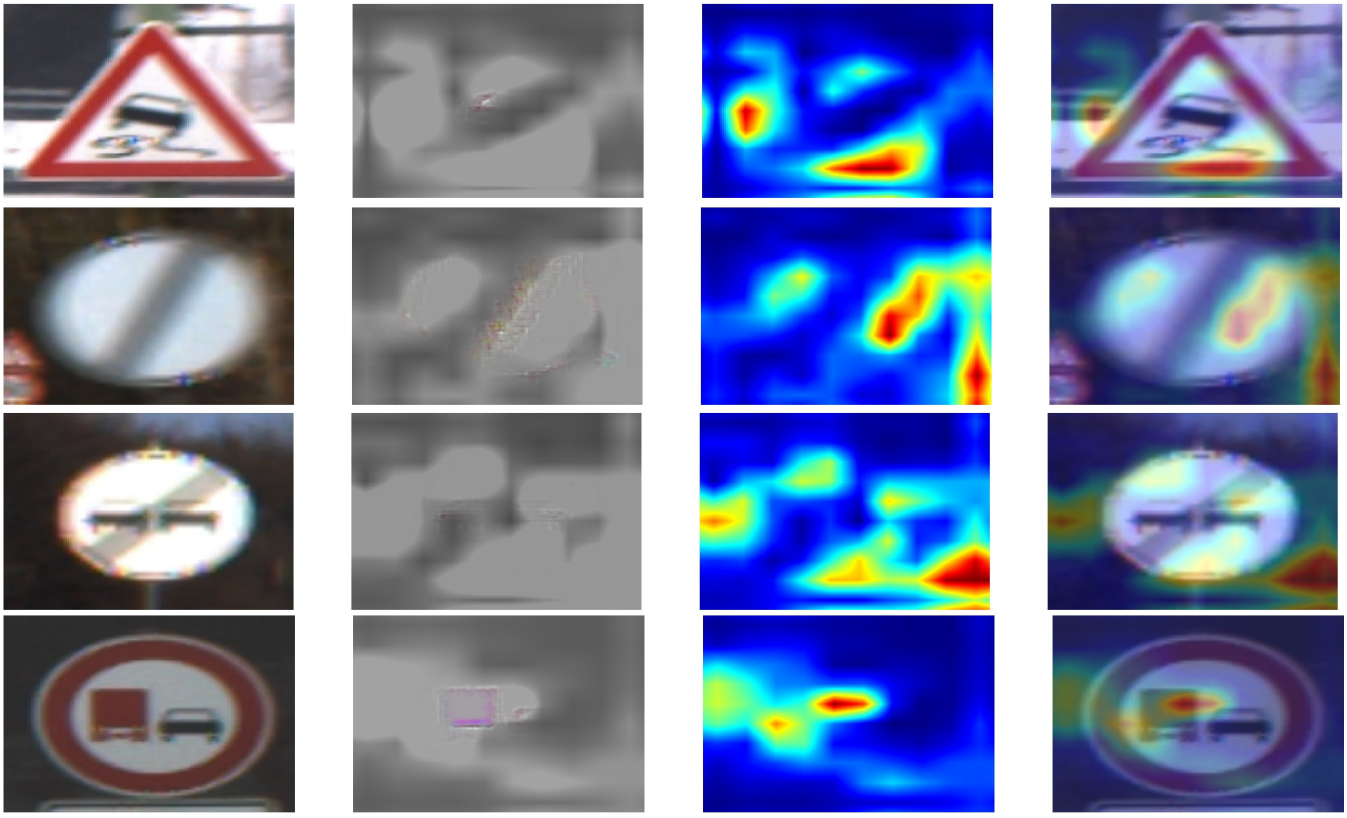

Figure 6: Heat maps generated by GradCAM++ on GSTRB dataset.

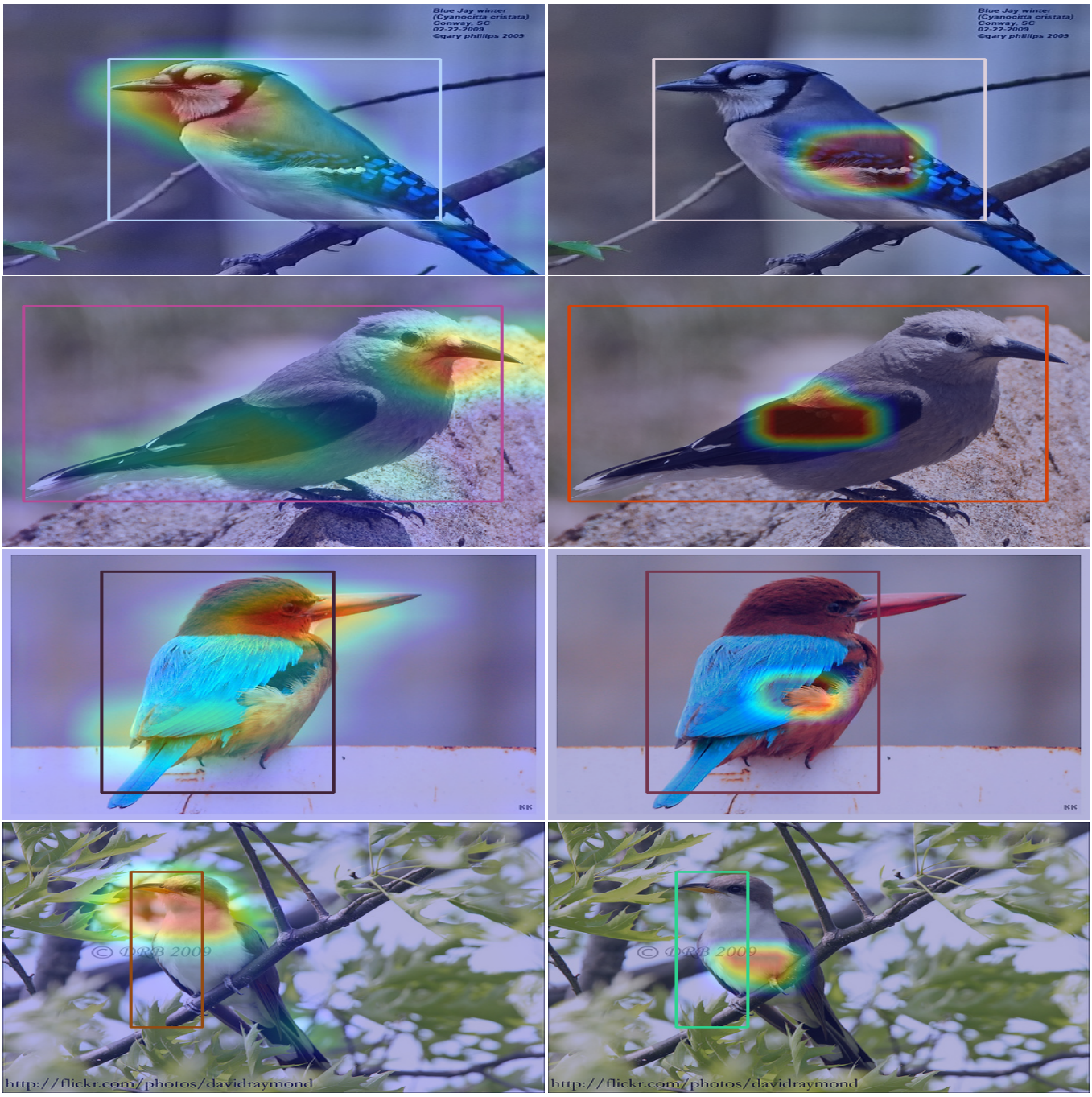

Figure 7: YOLO*v3* bounding boxes with heat maps generated through GradCAM++ and YOLO*v3* objectness scores.