Application for Vehicle Detection

Ruhul Quddus Tamim, Md Yusuf Bin Forkan, Shafi Ahmed, Syafiq Ibnu Ramadhan, Rasheaduzzaman

Abstract— Autonomous driving relies on vehicle detection-based computer vision to recognize and locate automobiles in digital photos or videos. Detecting "blocks" in photos or videos representing a vehicle's location is the underlying principle of vehicle detection. Also discussed here are 3D vehicle identification algorithms based on stereo perception derived from advanced planar vehicle detection. Finally, the differences between the feature extraction technique and the perceived outcomes are summarised in this publication, which highlights recent years' worth of vehicle identification systems. Vehicle detection techniques are examined in further detail here.

Index Terms — R-CNN, YOLO, IOU, CNN, Probability, Loss Function.

1 INTRODUCTION

WHO released "the State of Global Road Safety Report 2018" on Dec. 7th, 2018. In 2018, 1.35 million people (the population of Maine) died in traffic accidents yearly, even though road safety has improved over time. Families and communities have suffered tremendously because of traffic accidents and substantial financial losses. Each country's GDP is reduced by 3% because of the losses [2]. The conventional closed-loop control mechanism of road-vehicle-person is a major contributor to road traffic accidents. This management method's stability and security largely depend on the unpredictable nature of human behavior.

Society should investigate new modes of transportation to solve the transportation problem. The "vehicle-road" must be created by inviting "people" out of the closed-loop system. On the other hand, instead of relying on people to make traffic choices, automobiles would use human perception techniques. The ultimate objective is self-driving cars.

Autonomous driving relies heavily on computer vision for vehicle detection. According to studies on visual thinking, more than 80% of what we see in the environment comes from our visual senses [3]. Self-driving cars have a major challenge: how to implement visual perception.

Visual intelligence systems may be difficult to develop since they have to replace human perception. Vehicle detection is a classic scientific problem since it is important to intelligent perception systems. These algorithms mainly aim to support human drivers in their efforts to protect the safety of other road users.

Obstacle detection research focuses on how to identify and track cars ahead, which is a major issue in the area of safety-aided driving. The use of vehicle edge symmetry [4] or specialist hardware (color CCD [5-7] and binocular [8-10] computer vision technologies) or a variety of algorithms and implementation approaches have all been suggested in recent years both domestically and internationally.

The approach teaches to locate things accurately while using classification images to expand its vocabulary and resilience, using labeled detection photos.

An object detection system called YOLO9000, which is capable of detecting over 9000 distinct types of objects, is trained this way. To begin, we built upon the original YOLO detection technology to create YOLOv2, a cutting-edge, real-time detection system. A hybrid training strategy is then used to train a model using more than 9000 classes from ImageNet and COCO detection data.

2 Unified Detection

An object detection neural network is created by integrating the various components of the detection process. Each bounding box is predicted by our network based on information from the complete picture. For an image, it also forecasts all bounding boxes across all classes at once. This implies that our network considers the whole picture and all its components. High average accuracy and real-time training are also possible because of the YOLO design. Our method divides the input picture into a S x S grid. A grid cell contains an object's centroid, and that cell is responsible for detecting it. Each grid square predicts boxes and confidence ratings. If the model is certain that the box contains an item, it will give it a higher confidence score. Pr(Object) IOUtruth pred. is the formal definition of confidence.

On the safe side, all cells should have their confidence ratings set to zero. For an appropriate confidence score, the sum of the predicted box's intersection over union (IOU) and the dataset contains must be equal. x, y, w, h, and certainty all go into creating a bounding box. The midpoint of the box is indicated by the (x, y) coordinates in relation to the grid cell's borders. The image dimensions are predicted in relation to the entire image. Lastly, the confidence forecast represents the debt between the projected box and any actual box that may exist. In addition, probabilities for Pr(Class|Object) are predicted for each grid cell. These probabilities are determined based on which grid cell contains an item. We can only anticipate one set of class probabilities for each cell, regardless of the number of boxes B in the grid. During the test, we multiply conditional probability for each class by individual box confidence estimates based on the actual circumstance. IOU pred = Pr(Class I) IOU pred, which gives us class-specific confidence ratings for each box. Pr(Class I |Object) Pr(Object) IOU pred. These statistics represent both the chance of that class being in the box and how well the predicted box fits the item.

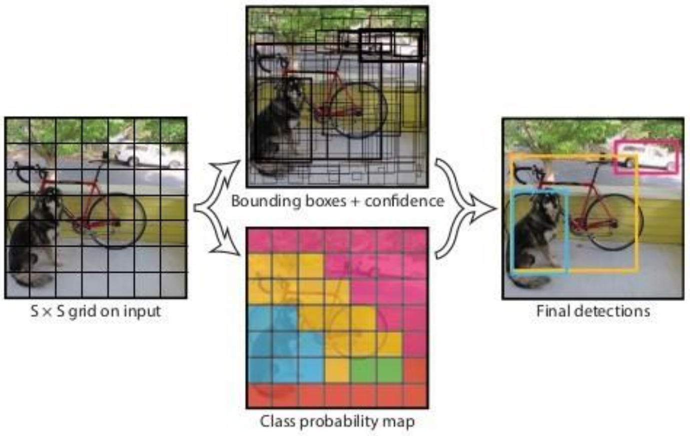


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

2.1 Network Design

On the PASCAL VOC detection dataset, a convolutional neural network is employed to generate this model. When it comes to visual qualities, convolutional layers are used in the network's early phases, whereas fully connected layers are used later on. Using the GoogLeNet paradigm of picture classification, we constructed our network. Our 24-layer network is followed by two completely connected convolutional layers. Lin et al [22] update GoogLeNet's inception modules with 1 1 reduction layers and 3 3 convolutional layers. The whole network is seen in Figure 3. For those who want to push the boundaries of object identification, we've developed an even faster version of YOLO we're training. Fast YOLO reduces the number of convolutional layers in the neural network from 24 to 9, as well as the number of filters. The only difference between YOLO and Fast YOLO in training and testing is the network size.

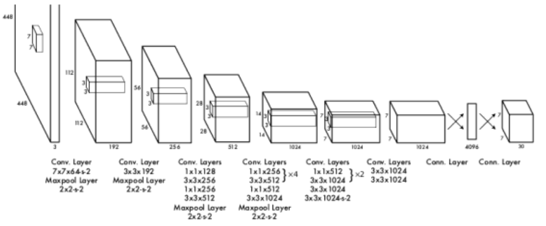


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

2.2 Training

We use the ImageNet 1000-class competition data set to pretrain our convolutional layers [29]. For pre-trained neural networks, we use an average-pooling layer and a fully connected layer after the first 20 convolutional layers shown in Figure 3. When trained for a week on the ImageNet 2012 validation set, a single crop top-5 accuracy of 88% is comparable to the GoogLeNet models in Caffe's Model Zoo [24]. Models are used to create detection tools. and his colleagues. Demostrate that pretrained networks may benefit from the addition of convolutional and linked layers [28]. The weights of the four convolutional and two fully linked layers are set to random. The network's input resolution was raised from 224 224 to 448 448 since fine-grained visual information is often required for detection. The last layer of our model predicts class probabilities and bounding box coordinates. Images' dimensions are used to normalise the bounding box's dimensions, such that they fall between the range of 0 and 1 for both dimensions. Using a grid cell location as an offset, we can maintain the x and y coordinates of the bounding box inside the 0 and 1 range. Last but not least, a linear activation function is used as opposed to a leaky rectified linear activation, which has been used in the preceding layers

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

The output of our model is tailored to minimize sum-squared error. It's easy to maximize sum-squared error, but our goal of maximizing average accuracy doesn't quite match this strategy. It offers equal weight to mistakes in classification and localization, which may not be the optimal method. In addition, a large number of grid cells in each image are empty. Cells that don't contain objects have their "confidence" values set to zero as a countermeasure to this impact. Training may begin to diverge early in the process due to model instability.

There will be several bounding boxes for each grid cell, according to YOLO. During training, just one bounding box predictor should be accountable for each item. It is decided which prediction has the greatest current IOU with the ground truth that will be designated as "responsible" for making predictions for a certain item in our system. As a consequence, the bounding box predictors became increasingly specialized. Overall recollection improves as each predictor grows better at predicting various dimensions and aspect ratios. During training, the multi-part loss function described below is optimized:

$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \\
&+ \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i-\hat{C}_i\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i-\hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c\in\text{classes}} \left(p_i(c)-\hat{p}_i(c)\right)^2
\end{aligned}
$$

2.3 Limitations of YOLO

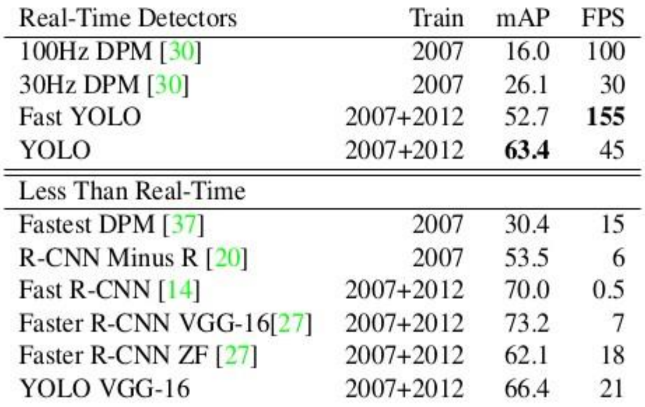
There are substantial geographical limits on predictions since each grid cell only predicts two bounding boxes and can only have one class in the YOLO model. Our model's ability to anticipate adjacent items is constrained by this restriction on the model's spatial range. Our approach has a hard time dealing with tiny groupings of things, like flocks of birds. Due to how our model has been trained, it cannot generalize to more complex shapes or configurations. As a result of our architecture's several down-sampling layers, our model employs rather coarse information for predicting bounding boxes.

Despite training on a loss function that approximates detection performance, our loss function treats errors equally in small and large bounding boxes. Small mistakes in a huge box may be devastating, but massive mistakes are usually harmless in the grand scheme. Our challenges are exacerbated by inaccurate localization.

2.4 Comparison to Other Real-Time Systems

There is much work on speeding up detection pipelines in object detection research. [37] [30] [14] [17] [27] [5] On the other hand, the real-time detection system developed by Sadeghi et al. is only 30 frames per second or better [30]. A 30Hz or 100Hz GPU implementation of DPM is compared to YOLO. However, we also evaluate their relative mAP and speed to look at the accuracy-performance tradeoffs present in object detection systems, even though they do not hit the real-time milestone.

Regarding object recognition on PASCAL, Fast YOLO is the quickest solution we have seen. It is more than twice as accurate as previous real-time detection studies, with an mAP of 52,7%. When using YOLO, the mAP is increased by 63.4%, but real-time performance is maintained.



Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

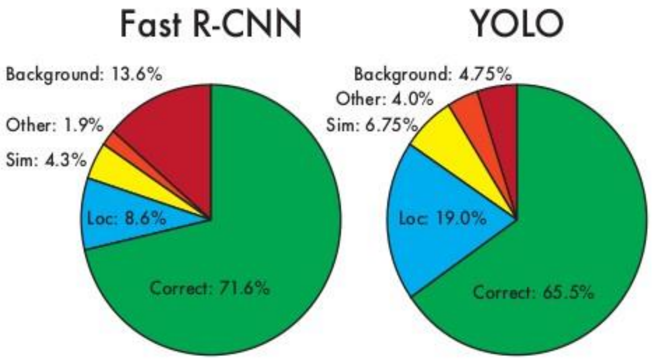
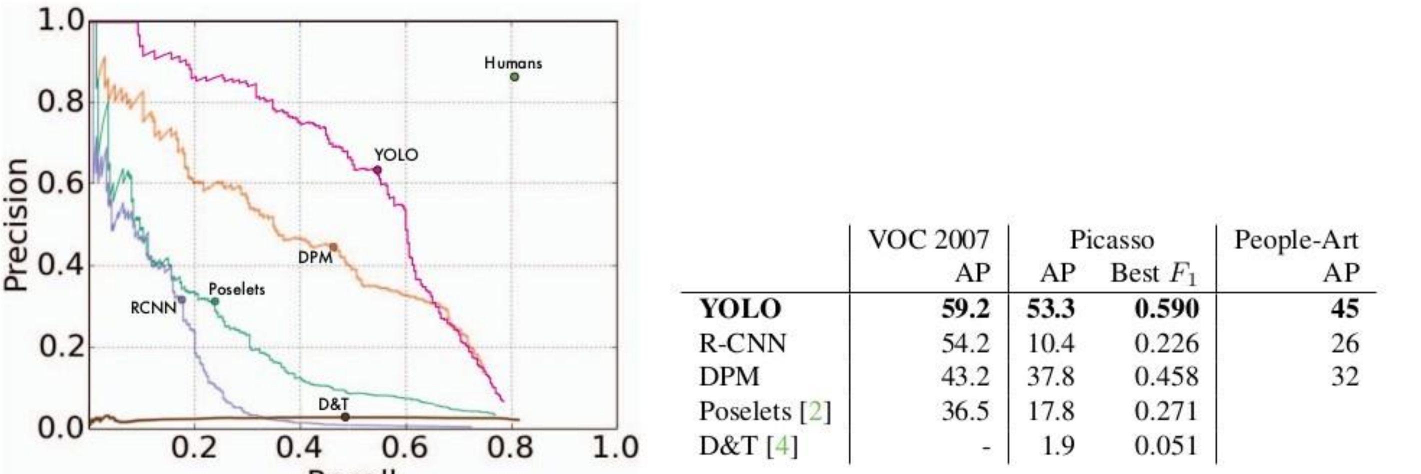


Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

- Other: class is wrong, IOU $>$ .1
- Background: IOU $<$ .1 for any object

2.5 Real-Time Detection in The Wild

YOLO is a good option for computer vision applications because of its quick speed and high accuracy in identifying objects. The real-time performance of YOLO is tested using a webcam, including the time it takes to receive camera photos and show detections. As a consequence, it is a fun and participatory experience. Using Yolo to monitor moving or changing things is possible by connecting it to a camera.



	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

(a) Picasso Dataset precision-recall curves.

(b) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets. The Picasso Dataset evaluates on both AP and best F_1 score.

Figure 5: Generalization results on Picasso and People-Art datasets.

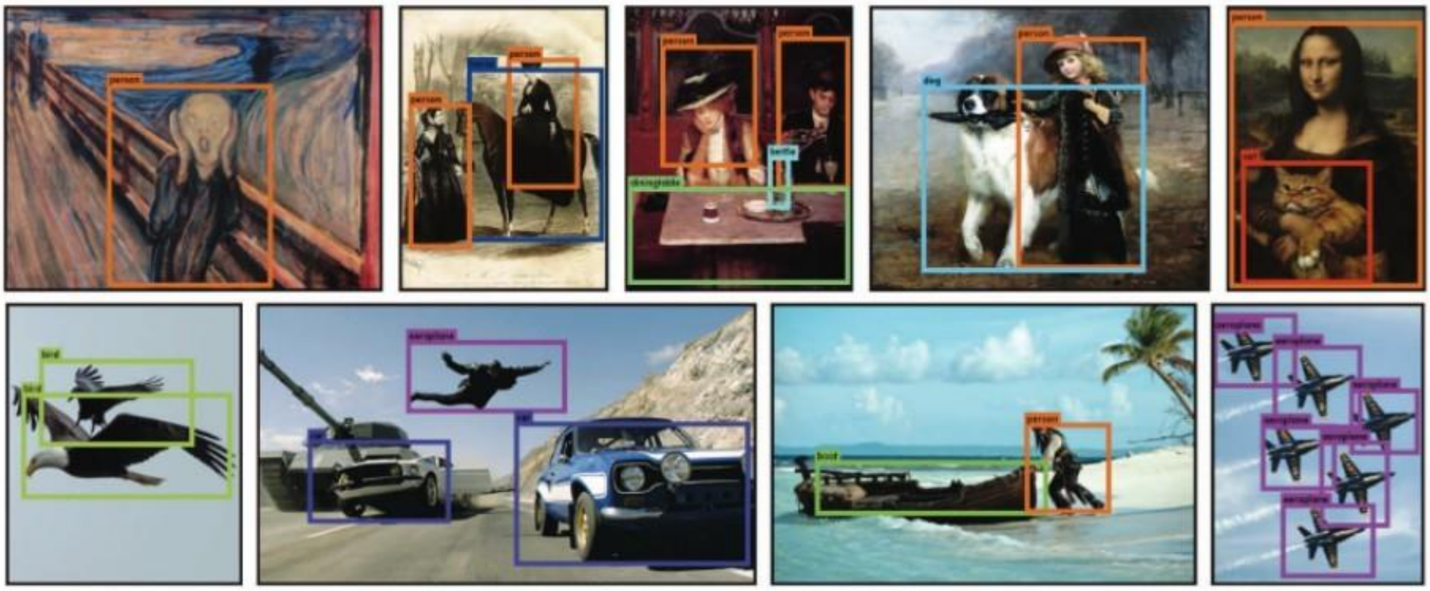


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

3 CONCLUSIONS

YOLO, a unifying paradigm for object detection, is introduced here. Our model is easy to build and can be trained on exclusive photos. A loss function directly related to detection performance is used to train YOLO instead of classifier-based techniques. The complete model is also trained together.

As the quickest general-purpose object detector available, Fast YOLO pushes the boundaries of object detection technology in real-time. YOLO is well-suited for applications that need quick and reliable object identification, as it can easily be extended to other domains.

ACKNOWLEDGMENT

The leader (Tamim) wishes to thank Yusuf, Shafi, Syafiq, and Rashead. This work was under the supervision of Dr. NOOR HAYATI BINTI MOHD ZAIN

REFERENCES

[1] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In Computer Vision–

[2] ECCV 2008, pages 2–15. Springer, 2008. 4

[3] [2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In International Conference on Computer Vision (ICCV), 2009. 8

[4] [3] H. Cai, Q. Wu, T. Corradi, and P. Hall. The cross-depiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. ArXiv preprint arXiv:1505.00110, 2015. 7 N. Dalal and B. Triggs. Histograms of oriented gradients for

[5] human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005. 4, 8 T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijaya- narasimhan, J. Yagnik, et al. Fast, accurate detection of

[6] 100,000 object classes on a single machine. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1814–1821. IEEE, 2013. 5 J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang,

[7] E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531, 2013. 4 J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified

[8] object detection and semantic segmentation. In Computer Vision–ECCV 2014, pages 299–314. Springer, 2014. 7 D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable

[9] object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014. 5, 6 M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I.

[10] Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, Jan. 2015. 2

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysi and Machine Intelligence, 32(9):1627–1645, 2010. 1, 4

[12] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware CNN model. CoRR, abs/1505.01749, 2015. 7

[13] S. Ginosar, D. Haas, T. Brown, and J. Malik. Detecting peo- ple in cubist art. In Computer Vision-ECCV 2014 Workshops, pages 101–116. Springer, 2014. 7

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014. 1, 4, 7

[15] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015. 2, 5, 6, 7

[16] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In Advances in neural information processing systems, pages 655–663, 2009. 4