International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

YOLO PRESENTS A NOVEL APPROACH TO OBJECT DETECTION

ANITA[1], RUPALI CHANDRAKAR[2], DR. SHIKHA PANDEY[3]

[1]Research Scholar, Department Of Computer Science And Engineering
RSR Rungta College Of Engineering And Technology, Bhilai, Chhattisgarh, India
[2,3]Assistant Professor, Department Of Computer Science And Engineering
RSR Rungta College Of Engineering And Technology, Bhilai, Chhattisgarh, India

---

Abstract

Presenting YOLO, a fresh take on object detection, is something we're very excited about. In the past, object detection made use of repurposed classifiers to actually detect objects. We approach object detection differently by viewing it as a regression problem. This makes use of geographically separated bounding boxes and associated categorization probabilities. A single neural network can instantly evaluate complete images and provide predictions about bounding boxes and class probabilities within the context of a single evaluation. All of the detection steps may be fine-tuned in real time based on detection performance because the entire pipeline is a single network.

Our unified infrastructure delivers lightning-fast performance. Our YOLO model revolves upon the continuous processing of images at a rate of 45 frames per second in real time. A smaller variant of the network, the Fast YOLO network, achieves twice the maximum average processing speed (mAP) of competing real-time detectors while processing an incredible 155 frames per second. Everything about this is remarkable. While YOLO is less likely to predict false positives on background, it is more prone to localization errors as compared to other state-of-the-art detection methods. To sum up, YOLO can create rather generalized representations of the objects it encounters. Its ability to generalize from natural images to other domains, such artwork, is superior to that of other detection approaches, like DPM and R-CNN.

Keywords- Object Detection, Real time video, YOLO coco dataset, PyCharm, OpenCV.

1. INTRODUCTION

When people look at a picture, they immediately recognize the items that are shown in it, where they are located, and how they interact with one another. Due to the fact that the human visual system is both swift and precise, we are able to engage in physically demanding tasks such as driving with a relatively low amount of conscious thought. The development of object identification algorithms that are both speedy and precise would make it possible for computers to drive automobiles without the need for specialized sensors. It would also make it possible for assistive devices to provide real-time scene information to human users. Finally, it would open the door to the possibility of general-purpose robotic systems that are responsive.

Within the most recent generation of detection systems, classifiers are repurposed to perform the function of detection. These systems begin with a classifier that corresponds to the object in question, and then proceed to evaluate the object at a variety of various sizes and locations within a test picture. This allows the system to ultimately identify the item. Examples of such systems are deformable parts models (DPM), which make use of a sliding window methodology. In this method, the classifier is performed at points that are consistently spaced over the whole picture [10].

The utilization of region suggestion is a feature that is utilized by more contemporary methods like as R-CNN.

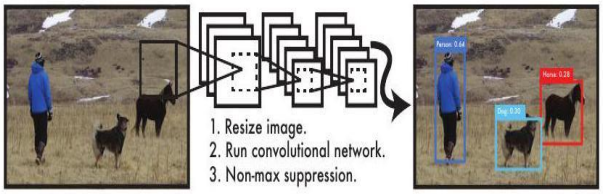

Figure 1. YOLO Detection System. YOLO picture processing is easy.

The system consists of resizing the input image to 448x448, running a single convolutional network, and throttling detections based on model confidence techniques to construct picture bounding boxes and then classify them. The post-processing step, which follows classification, involves improving the bounding boxes, removing duplicates, and rescoring them according to the items in the scene [13]. Due to the need to teach each component separately, these complex pipelines are slow and difficult to optimize.

We transform the object recognition issue into a singular regression problem by transferring the class probabilities and bounding box coordinates from the picture pixels to the bounding box coordinates. Our method relies on taking a cursory look at an image (YOLO) to deduce what objects are there and where they are positioned.

See Figure 1 for an illustration of how deliciously straightforward YOLO is. Different bounding boxes and class probabilities can be predicted by the same convolutional network across many classes. Complete image Through YOLO training, detection performance may be improved. When compared to more traditional methods, a unified paradigm for object detection offers a number of positive advantages.

The first thing is that YOLO is incredibly quick. A state-of-the-art method is unnecessary as we reframe detection as regression. In order to generate predictions regarding the detections, we train our neural network on a fresh image right before the test. On a Titan X GPU, our baseline network runs at 45 fps without batch processing, but a fast variant can reach over 150 fps. Thanks to this feature, streaming video can be processed in real time with a latency of under 25 milliseconds. Furthermore, among real-time systems that are equivalent, YOLO has a mean average accuracy that is twice as excellent. On the homepage of our project, which can be found at http://pjreddie.com/yolo/, our system is displayed live on a webcam.

The second thing that YOLO does is think about the image in a broad sense before making predictions. Because it observes the whole image during training and testing, YOLO automatically encodes class context and appearance. This is in contrast to approaches that are based on sliding windows and region proposals. A technique for detecting the top The Fast R-CNN [14] algorithm is unable to comprehend the larger context, which causes it to incorrectly identify background patches as objects. YOLO has a background error rate that is fifty percent lower than that of Fast R-CNN.

Finally, YOLO can learn to represent objects in a generalized way, which is a huge plus. You Only Live Once far outpaces DPM and R-CNN in terms of teaching on natural images and judging on artwork. The generalizability of YOLO makes it highly unlikely that it would fail when applied to new domains or given unexpected inputs. When compared to more sophisticated detection methods, YOLO's accuracy is lower. Although it struggles with smaller items, it can swiftly recognize objects in photos. We have investigated these tradeoffs in our experiments. The public has access to all of the code we use for testing and training. You can download any of the pre-trained models.

2. UNIFIED DETECTION

Combining all of the parts into one neural network allows us to detect objects. Our network predicts the contents of each bounding box based on the picture's attributes. Every class using the same image makes a prediction about its bounding boxes at the same time. Therefore, our network considers the full image and all of its parts. By utilizing the YOLO design, which permits end-to-end training and real-time speeds, a high level of average accuracy is attained.

To divide the input picture, we employ a square-by-square grid. Objects whose centers are falling into grid cells are detected by grid cells. Predictions about the B bounding boxes and confidence ratings are made in each grid cell. The model's confidence ratings show how sure it is that the box contains an item and how accurate it is at predicting. With the product of Pr(Object) and IOU_"pred" and "truth," we get the concept of confidence. To indicate that the cell is empty, set the confidence ratings to 0. That being said, we'd like it if the confidence score matched the anticipated intersection of box-ground truth over union (IOU), should the contrary be true. X, y, w, and h are the five forecasts and confidence levels for each bounding box. The center of the box with respect to the grid cell is indicated by the coordinates (x, y). Anticipation affects the entire image's width and height. When we speak about confidence prediction, we're referring to the IOU between the predicted box and any ground truth box.

The conditional class probabilities, denoted as Pr(Class _i| Object), are also predicted by each grid cell independently. These probabilities are dependent on the grid cell that contains an item in order to be calculated. There is just one set of class probabilities that we anticipate for each grid cell, and this is true regardless of the number of spaces B.

Each participant's confidence prediction for their own box is multiplied by the conditional class probabilities when testing is underway,

$$\Pr(\text{Class}_i \mid \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

so that we can get confidence scores for each class individually for each box. These scores represent the object's anticipated box fit as well as the class's likelihood of appearing in the box.

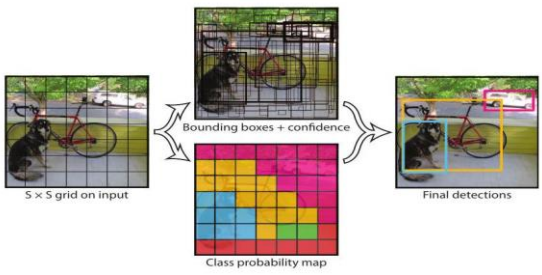

Figure 2: The Prototype. In our system, detection is modeled as a regression problem. A S×S grid is used to segment the image. The model makes predictions about C class probabilities, B bounding boxes, and confidence scores for each grid cell. The tensor S×S×(B*5+C) encodes these predictions.

For evaluating YOLO on PASCAL VOC, we use $S = 7$, $B = 2$. PASCAL VOC has 20 labelled classes so $C = 20$. Our final prediction is a $7 \times 7 \times 30$ tensor.

2.1. NETWORK DESIGN

We tested this convolutional neural network model using the PASCAL VOC identification dataset [9]. Predicting output probabilities and coordinates is the job of fully connected layers, while earlier convolutional layers are in charge of visual property extraction.

Our network was built using the principles of Google's neural network [33] for the goal of image classification. We use a network architecture consisting of twenty-four convolutional layers and two fully linked layers. We apply 3x3 convolutional layers and 1x1 reduction layers to GoogleNet's inception modules to align with the results reported in Lin et al. [22]. You can see the whole network in Figure 3.

We also train a fast YOLO to improve its object recognition speed. A neural network with fewer than twenty-four filters and nine convolutional layers is utilized by the Fast YOLO method. In terms of testing and training conditions, the sole difference between YOLO and Fast YOLO is the network size.

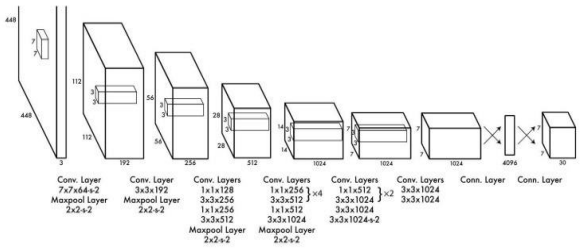

Figure 3: Architecture. We use 24 convolutional layers and 2 fully linked layers in our detecting network. Alternating 1x1 convolutional layers reduce feature space from previous layers. We train convolutional layers on ImageNet at half resolution (224x224 input picture) and then do detection at double resolution.

The final output of our network is the $7 \times 7 \times 30$ tensor of predictions.

2.2. Training

We use the ImageNet 1000-class competition dataset [29] to pretrain our convolutional layers. As illustrated in Figure 3, after the first twenty convolutional layers, we use a completely connected and an average-pooling layer for pretraining. Following just one week of training on the ImageNet 2012 validation set, this network achieves a top-5 accuracy of 88% for single crops. The GoogLeNet models created by Caffe's Model Zoo [24] achieve an accuracy level similar to this one.

We change the model so it can detect. When it comes to enhancing the performance of pretrained networks, Ren et al. show that convolutional and linked layers function better [28]. In keeping with their model, we use two fully connected layers and four convolutional layers, all of which weights are randomly assigned. Our network's fine-grained visual identification capabilities have been improved by increasing the input resolution from 224×224 to 448×448.

In the last layer, we use the class statistics and bounding box coordinates to create predictions. When the bounding box's width and height are normalized by the picture's width and height, they fall inside the range of 0 to 1. To make the x and y coordinates of the enclosing box range from 0 to 1, we offset them from the position of a grid cell.

All of the layers up to our last one used leaky rectified linear activation, whereas our final layer uses a linear activation function.

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \quad \#(2)$$

When it comes to sum-squared error, we maximize the model output. Although improving the sum-squared error is an easy process, it does not adequately achieve our objective of optimizing the average accuracy. The mistakes in localization and classification are given equal weight, which is not the best possible situation. Most of the grid cells in each and every photo are vacant. The "confidence" values of those cells tend to decrease toward zero, frequently overcoming the gradient that is produced by cells that contain objects. Instability in the model might lead to divergence before training is complete.

We have resolved this issue by raising the loss from predictions of bounding boxes' coordinates and lowering the loss from predictions of confidence for empty boxes. We use two parameters, λ_{coord} and λ_{noobj} to accomplish this. We set $\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = .5$.

The sum-squared error method gives equal weight to both large and tiny box errors. When it comes to our error measure, Smaller discrepancies in bigger boxes should be given less weight than smaller ones in smaller boxes. Predicting the square root of the width and height of the bounding box instead of both at once can help with the problem to a certain degree.

It is anticipated by YOLO that each grid cell will have many bounding boxes. During training, we wish for each item to have a single bounding box predictor. A prediction is considered "responsible" if it comes from the predictor with the highest current IOU with the ground truth. This leads to the specialization of bounding box predictors. Size, aspect ratio, and item type predictions are all enhanced with each predictor, leading to higher recall.

In training, we optimize a multi-part loss function. :

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

where$\mathbb{1}_i^{\text{obj}}$ denotes if object appears in cell i and $\mathbb{1}_{ij}^{\text{obj}}$ denotes that the j th bounding box predictor in cell i is "responsible" for that prediction.

The loss function only penalizes classification mistakes if an object is in that grid cell, which is why the class probability is conditional. The only predictor that is penalized for inaccurate bounding box coordinates is the one that is "responsible" for the ground truth box, meaning the one with the greatest IOU in that grid cell.

The network undergoes a 135-epoch training process using validation and training data from PASCAL VOC 2007 and 2012. For training purposes, we used VOC 2007 test data in our 2012 testing. We use 64 batches, 0.9 momentum, and 0.0005 decay for training. Our learning rate schedule: In the initial epochs, we gradually increase the learning rate from 〖10〗^(-3) to 〖10〗^(-2). Unstable gradients cause model divergence at high learning rates. We train with 〖10〗^(-2) for 75 epochs, then 〖10〗^(-3) for 30 epochs, and lastly 〖10〗^(-4) for 30 epochs.

To prevent overfitting, we use dropout and enhance the data significantly. Layer co-adaptation cannot occur in a dropout layer following the first linked layer with a rate of 0.5 [18]. When augmenting data, we randomly scale and translate images by up to 20% of their original size. Using HSV color space, we arbitrarily change the saturation and exposure of the image by a maximum of 1.5.

2.3. INFERENCE

Predicting test image detections takes one network assessment, like training. The network predicts 98 bounding boxes per image and class probabilities using PASCAL VOC. Test time is fast with YOLO since it only needs one network assessment, unlike classifier-based approaches.

The grid design ensures that the bounding box forecasts contain a diverse range of spatial elements. Most of the time, it is obvious which grid cell each object belongs to, and the network only predicts one box for each object. Numerous cells, on the other hand, are able to accurately pinpoint certain huge objects or items that are located on the boundary of numerous cells. Fixing these numerous detections can be accomplished through the use of non-maximal suppression. In mAP, non-maximal suppression contributes 23% to performance, despite the fact that it is not as important to performance as it is for R-CNN or DPM.

2.4. LIMITATION OF YOLO

There are strict geographical constraints on YOLO's bounding box predictions due to the fact that each grid cell can only foresee two boxes and one class. Because of this geographical restriction, our model is unable to predict as many nearby things. Crowds of small objects, such as birds, are difficult for our model to handle.

Since our method relies on data to anticipate bounding boxes, it is unable to generalize to objects with unusual aspect ratios or configurations. Our strategy uses coarse characteristics in our bounding box prediction model because we downsample the input picture multiple times.

Finally, we train on a loss function that approximates detection performance that handles mistakes equally in small and big bounding boxes. minor errors in large boxes are usually harmless, whereas minor errors in small boxes affect IOU more. Our biggest error is mislocalization.

3. COMPARISON TO OTHER DETECTION SYSTEM

Object detection is a key problem in computer vision. Detection pipelines often begin with the extraction of a set of robust characteristics from the input photos. A few examples of these features are convolutional features [6], Haar [25], SIFT [23], and HOG [4]. Classifiers [35,21,13,10] and localizers [1,31] can then be used to identify objects in the feature space. These classifiers or localizers can be applied in a sliding window fashion to either a subset or the full image [34, 15, 38]. In this article, we compare and contrast the YOLO detection approach with many industry-leading detection systems, highlighting both their similarities and differences.

Models of pliable components. Models for deformable components (DPM) use sliding window object detection [10]. The DPM utilizes an independent pipeline to carry out operations including feature extraction, region classification, bounding box prediction for high-scoring regions, etc. All of these parts are substituted by a single convolutional neural network in our system. The network simultaneously does bounding box prediction, feature extraction, contextual reasoning, and no maximal suppression. In order to optimize them for the detection task, the network trains its features in-line instead of employing static features. When compared to DPM, our simplified design is far faster and more accurate.

Implementing R-CNN... When it comes to object detection in images, R-CNN and related algorithms use region suggestions instead of sliding windows. A linear model adjusts the bounding boxes, a neural network extracts features, a support vector machine (SVM) scores the boxes, a linear model uses Prospective Bounding Boxes [34] to remove duplicate detections, and finally, non-max suppression is used to eliminate duplicates. The system's tremendous slowness, brought on by the necessity to fine-tune each stage of the complex pipeline independently, resulted in each picture taking about 40 seconds during test time[14].

There are several similarities between YOLO and R-CNN. Every cell in the grid makes a score-based suggestion for a potential bounding box using convolutional features. However, our method restricts the grid cell suggestions spatially to lessen the occurrence of identical item detections. To add insult to injury, our method only proposes 98 bounding boxes per image, whereas Selective Search provides almost 2,000. Afterwards, we utilize our technology to combine all of these components into a single, optimal model.

Various Rapid Identifiers Quite a bit quicker In an effort to speed up the R-CNN architecture, R-CNN intends to replace Selective Search with neural networks for region proposal [14, 27]. You can't compare their speed and precision to that of R-CNN, but they're still no match for real-time performance.

Research on how to speed up the DPM pipeline is prevalent [30, 37, 5]. They maximize HOG computing through the use of cascades and the acceleration of calculation on GPUs. Note that only 30" "Hz DPM [30] is capable of operating in real-time.

The YOLO algorithm eliminates the need to optimize individual components of a large detection pipeline and is quick by design.

Because they don't have to cope with as much fluctuation, detectors for single classes, such as faces or persons, can be significantly optimized [36]. With its ability to learn and detect many items at once, YOLO is a versatile detector.

The multi-box is advanced. When predicting ROIs, Szegedy et al.[8] train a convolutional neural network rather than using Selective Search, as opposed to R-CNN. Single object detection is also possible with MultiBox by replacing the confidence prediction with a single class prediction. Nevertheless, MultiBox is still limited to being a part of a larger detection pipeline and cannot perform general object identification, therefore further picture patch classification is required. While both YOLO and MultiBox use neural networks to predict the borders of images, YOLO's detection capabilities are superior.

The Defeat. By first training a convolutional neural network to do localization, Sermanet et al. [31] modify the network such that it can also detect objects. Even though it's still not a cohesive system, OverFeat does sliding window detection efficiently. Instead of focusing on detection performance, OverFeat optimizes for localization. When creating a prediction, the localizer, similar to DPM, only considers local information. OverFeat necessitates substantial post-processing to get consistent detections since it is unable to reason regarding global context.

Grab multiple objects at once. Redmon et al.'s [26] work on grasp detection is conceptually comparable to ours. The MultiGrasp method for regression to grasps forms the basis of our grid approach to bounding box prediction. But compared to object detection, grip detection is child's play. In order for MultiGrasp to function, each object in an image only has to have its graspable region predicted once. It just needs to locate an appropriate area to grab; it doesn't need to guess the object's class or even its size, position, or limits. For various objects in an image belonging to different classes, YOLO can forecast both their bounding boxes and the likelihood of each class.

4 EXPERIMENTS

To start, on PASCAL VOC 2007, we evaluate YOLO in comparison to various real-time detection methods. We compare the mistakes made by YOLO and Fast R-CNN, a top-performing R-CNN variant, on VOC 2007 [14] to learn about the distinctions between the two R-CNN variants. Rescoring Fast R-CNN detections using YOLO and reducing background false positive errors significantly improves performance, as demonstrated by various error profiles. In addition, we compare mAP to the state-of-the-art methodologies currently available and offer the findings of VOC 2012. We conclude by demonstrating, on two artwork datasets, that YOLO outperforms other detectors in terms of generalizing to new domains.

4.1. COMPARISON TO OTHER REAL TIME DATA

Standard detection pipeline speed optimization is a major area of attention in object detection research. Citations: [5] [37] [30] [14] [17] [27]. On the other hand, only Sadeghi et al. develop a real-time detection system (30 fps or higher) [30]. We evaluate YOLO in comparison to their 30Hz or 100Hz GPU implementation of DPM. To investigate the accuracy-performance tradeoffs in object identification systems, we evaluate their relative mAP and speed, even while previous efforts fail to achieve the real-time milestone.

From what we can tell, Fast YOLO is the quickest object detector currently available, and it's also the quickest way on PASCAL. Compared to previous work on real-time detection, its accuracy of 52.7%mAP is more than double.

YOLO pushes mAP to 63.4% while still maintaining real-time performance.

Using VGG-16, we also train YOLO. While this model outperforms YOLO in terms of accuracy, it is also noticeably slower. While it can be used to compare VGG-16-based detection systems, the paper's remaining sections center on our quicker models due to their superior performance in real-time.

Fastest DPM effectively speeds up DPM without sacrificing much mAP but it still misses real-time performance by a factor of 2 [37]. It also is limited by DPM's relatively low accuracy on detection compared to neural network approaches.

R-CNN minus R replaces Selective Search with static bounding box proposals [20]. While it is much faster than

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: Systems Operating in Real-Time on PASCAL VOC 2007. There is a comparison of the speed and performance of fast detectors. Not only is Fast YOLO the fastest PASCAL VOC detector ever recorded, but it is also twice as accurate as any other real-time detector. Compared to the fast version, the YOLO algorithm is 10mAP more exact; nonetheless, it is still far quicker than real-time. R-CNN, it not only does not meet the requirements for real-time, but it also suffers a large loss in accuracy due to the absence of suitable ideas.

Fast R-CNN uses selective search to speed up the classification stage of R-CNN, although it still takes about two seconds each image to provide bounding box proposals. Despite its high mAP, its frame rate of 0.5 fps is far lower than real-time.

By replacing selective search with a neural network, the newly created Faster R-CNN suggests bounding boxes in a way similar to Szegedy et al. on page 8: Our study shows that their most accurate model functions at 7 frames per second, whereas a smaller model with less precision performs at 18 frames per second. quicker R-CNN with VGG-16 architecture is 10mAP quicker than YOLO, but it's six times slower. The ZeilerFergus Faster R-CNN has worse accuracy despite being only 2.5 times slower than YOLO.

4.2. VOC 2007 ERROR ANALYSIS

Looking at a deep breakdown of VOC 2007 data, we will compare YOLO against the most advanced detectors and find out where it falls short. Given that Fast R-CNN is a publicly available, top-performing detector on PASCAL, we are comparing it to YOLO.

We utilize the technique and tools developed by Hoiem and colleagues. [19] We examine the top N guesses for each category during testing. Each forecast is either accurate or categorized according to the sort of mistake made.

- Correct: correct class and IOU > .5
- Localization: correct class, . 1 < IOU < .5
- Similar: class is similar, IOU > . 1

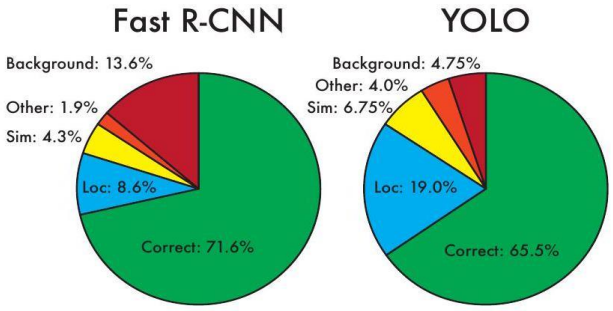

Figure 4: Analyzing Errors: Quick R-CNN vs. YOLO The charts display the percentage of top N detections for each category, where N is the number of objects in that category, and background and localization mistakes are shown as a percentage.

Other: class is wrong, IOU > . 1

- Background: IOU < .1 for any object

As a whole, all 20 classes contributed to the breakdown of mistake types, as shown in Figure 4.Things can be difficult for YOLO to pinpoint precisely. When it comes to YOLO, localization issues are far and away the most common cause of mistakes. While Fast R-CNN does increase its background mistake rate, it has drastically cut down on localization errors. Since 13.6% of its main detections are empty, they are not accurate.Fast R-CNN has a higher

probability of predicting background detections compared to YOLO, almost three times more likely.

4.3. COMBINING FAST R-CNN AND YOLO

The YOLO model outperforms Fast R-CNN in terms of background error rate. We achieve a significant speed boost by using YOLO to eliminate background detections from Fast R-CNN. We will compare YOLO's predicted bounding boxes to all of R-CNN's predicted bounding boxes to see if they are comparable. If it happens, we improve that prediction by factoring in the YOLO-predicted likelihood and the overlap between the two boxes.

A mean absolute performance (mAP) of 71.8% is achieved by the top Fast R-CNN model on the VOC 2007 test set. In a way, it's both YOLO and itself.

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**

Table 2: VOC 2007 model combination experiments. Combining different models with the best Fast R-CNN is examined. Other Fast R-CNN versions offer only a minor performance improvement, while YOLO does.

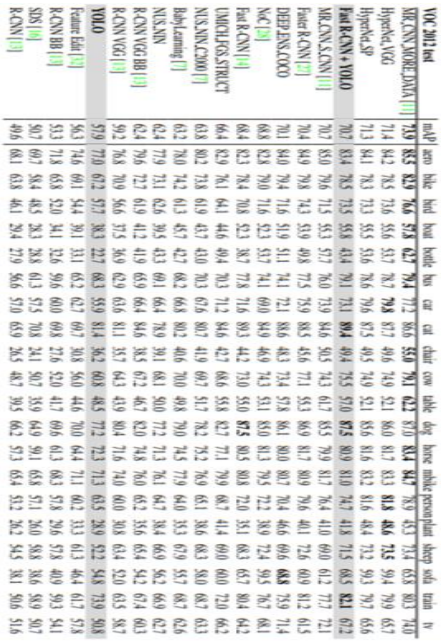

Table 3: 2012 PASCAL VOC status quo. The YOLO public leaderboard was compared to the full comp 4 (outside data allowed) as of November 6, 2015. For a number of detection approaches, we display the mean and per-class average precision. So far, YOLO is the sole real-time detector.

Fast R-CNN + YOLO comes in at number four, with a score that is 2.3% higher than Fast R-CNN.

A 3.2% gain brings the mAP up to 75.0%. Additionally, we attempted to combine the most successful Fast R-CNN model with a number of different variants of Fast R-CNN. For more information, please refer to Table 2; these ensembles resulted in mAP gains that were between 0.3 and 0.6 percent.

Because there is not much of an advantage to be gained by merging different versions of Fast R-CNN, the increase that comes from YOLO is not merely a result of the process of model assembly. It is more accurate to say that the reason YOLO is so successful in improving Fast R-CNN's performance is exactly because it creates a variety of errors throughout the testing process.

You won't get the speed boost from YOLO with this combination because we run each model separately and then combine the results. Thanks to its incredibly quick speed, YOLO, on the other hand, does not significantly increase the amount of computation time required compared to quick R-CNN..

4.4. VOC 2012 RESULTS

In 2012, YOLO achieved 57.9% mAP on the VOC test set. Table 3 shows that this is lower than the state-of-the-art and more in line with the R-CNN originally built with VGG-16. When it comes to handling relatively small objects, our system struggles, especially when compared to its main competitors. In domains including bottle, sheep, and television/monitor models, YOLO performs 8-10% worse than R-CNN or Feature Edit. But in other areas, like their train and cat divisions, YOLO has better success. One detection strategy that can reach maximum performance is the combination of our Fast R-CNN and YOLO model. Fast R-CNN jumps to number five on the public leaderboard after combining with YOLO, which increases its performance by 2.3%.

4.5. GENERALIZABILITY: PERSON DETECTION IN ARTWORK

According to the VOC 2012 test results, YOLO gets 57.9% mAP. This is lower than the state-of-the-art and more in line with the R-CNN that was initially constructed using VGG-16, as shown in Table 3. Our technology struggles to handle relatively small objects as compared to its main competitors. The bottle, sheep, and television/monitor models get 8-10% poorer results for YOLO compared to R-CNN or Feature Edit. However, YOLO has more success in other areas, including their train and cat divisions. For optimal performance, one detection strategy is to

combine our Fast R-CNN with the YOLO model. When combined with YOLO, Fast R-CNN achieves a 2.3% boost, putting it in fifth place in the public leaderboard.

5. REAL - TIME DETECTION IN THE WILD

Computer vision applications benefit greatly from the YOLO object detector's speed and accuracy. We hook up YOLO to a webcam and watch to see if it keeps running in the background.

(a) Picasso Dataset precision-recall curves.

	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	-

(b) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets. Both AP and the best F_1 score are used for evaluation in the Picasso dataset.

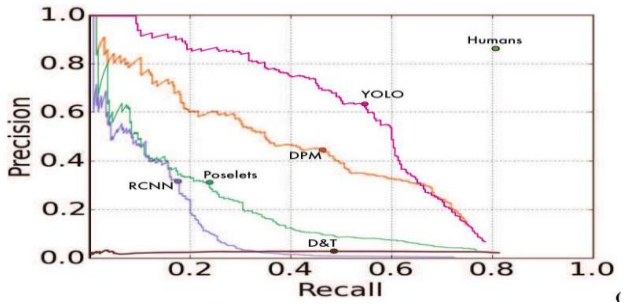

Figure 5: Generalization results on Picasso and People-Art datasets.

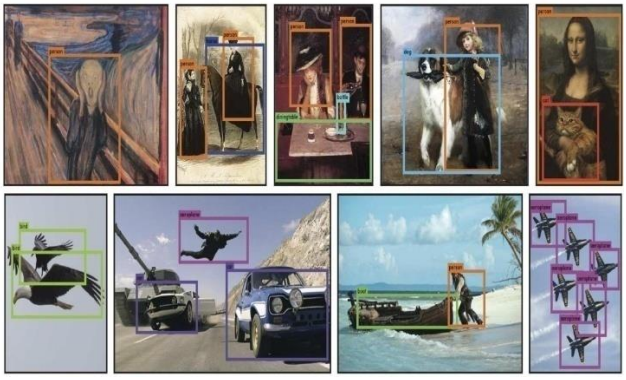

Figure 6: Qualitative Results. Internet-based YOLO using sample art and natural photos. It generally works, but it thinks one person is an airplane.

for example, the amount of time required to get photos from the camera and display the detections.

The resultant system is engaging and full of interactive features. When connected to a camera, YOLO examines each photo independently, but it also acts as a tracking system, detecting things as they move or change their appearance. Detailed instructions and a live demo of the system are available on our project website (http://pjreddie.com/yolo/).

5. RESULT

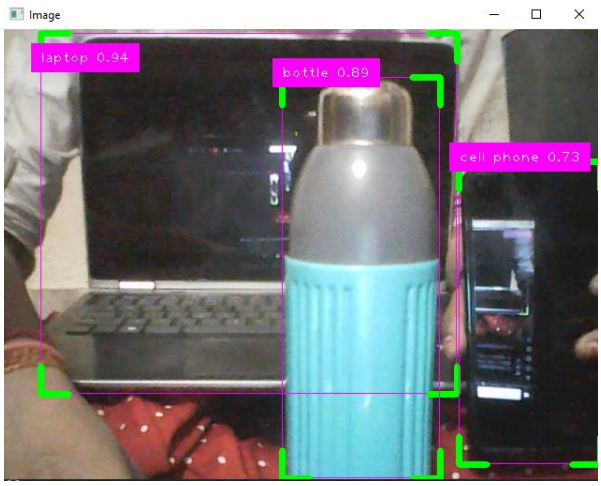

Figure 8: Object Detection using OpenCV Yolo

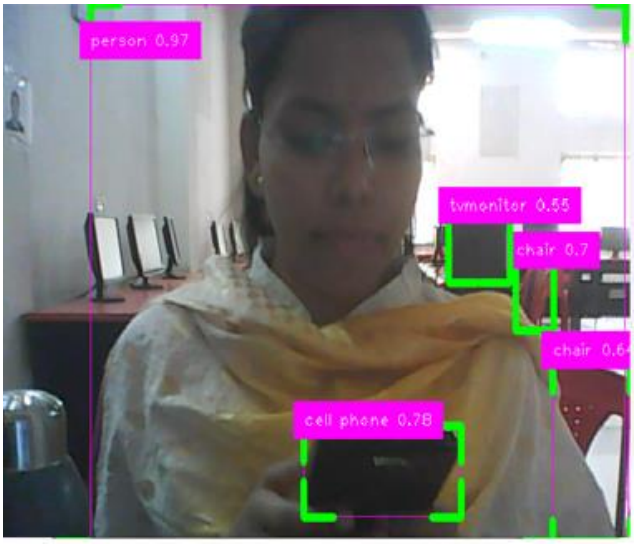

Figure 8 : Multi-Object Detection using YOLO

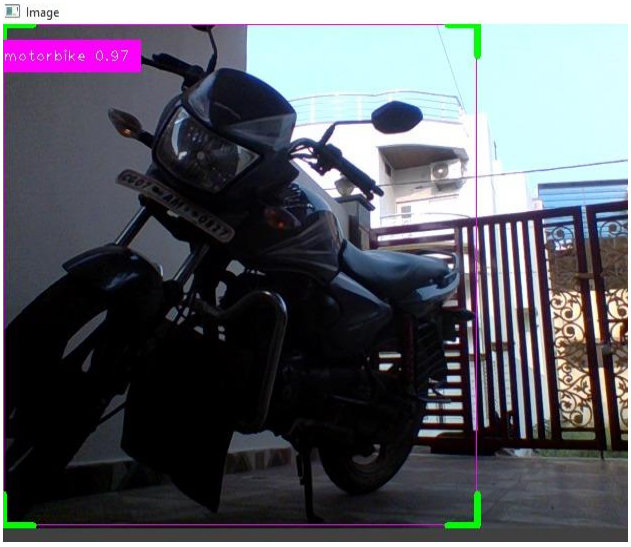

Figure 9 : Motorbike Detection using YOLO

Figure 11 : Cellphone and Bottle detection by using Yolo

6. CONCLUSION

In order to differentiate between various objects, we introduced the YOLO approach in this study with the goal of using a single neural network. After being extended from natural photos to other domains, this expanded technique outperforms existing algorithms. It is possible to train the algorithm on a complete image, and its construction is simple. Using region proposal techniques limits the classifier to a certain region. In order to do border prediction, YOLO can see the whole picture. Moreover, it anticipates a reduced amount of false positives in background regions. When applied in real time, this method outperforms all others in terms of effectiveness and speed of movement among classifier algorithms.

REFERENCES

[1] M. B. Blaschko and C. H. Lambert. Learning to localize objects with structured output regression. In Computer VisionECCV 2008, pages 2-15. Springer, 2008. 4

[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

[3] H. Cai, Q. Wu, T. Corradi, and P. Hall. The crossdepiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. arXiv preprint arXiv:1505.00110, 2015. 7

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

[5] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik, et al. Fast, accurate detection of 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

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531, 2013. 4

[7] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In Computer Vision-ECCV 2014, pages 299-314. Springer, 2014. 7

[8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155-2162. IEEE, 2014. 5, 6

[9] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. 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

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

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

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

[13] 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

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

[15] 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

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In Computer VisionECCV 2014, pages 297-312. Springer, 2014. 7

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv preprint arXiv:1406.4729, 2014. 5

[18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012. 4

[19] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In Computer Vision-ECCV 2012, pages 340-353. Springer, 2012. 6

[20] K. Lenc and A. Vedaldi. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015. 5, 6

[21] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I-900. IEEE, 2002. 4

[22] M. Lin, Q. Chen, and S. Yan. Network in network. CoRR, abs/1312.4400, 2013. 2

[23] D. G. Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150-1157. Ieee, 1999. 4

[24]D. Mishkin. Models accuracy on imagenet 2012 val. https://github.com/BVLC/caffe/wiki/ Models-accuracy-on-ImageNet-2012-val. Accessed: 2015-10-2. 3

[25] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In Computer vision, 1998. sixth international conference on, pages 555-562. IEEE, 1998. 4

[26] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. CoRR, abs/1412.3128, 2014. 5

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 5, 6, 7

[28] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, abs/1504.06066, 2015. 3, 7

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Image Net Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015. 3

[30] M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In Computer Vision-ECCV 2014, pages 65-79. Springer, 2014. 5, 6

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229, 2013. 4, 5

[32] Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. arXiv preprint arXiv:1409.6911, 2014. 7

[33]C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014. 2

[34]J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154-171, 2013. 4, 5

[35]P. Viola and M. Jones. Robust real-time object detection. International Journal of Computer Vision, 4:34-47, 2001. 4

[36]P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137-154, 2004. 5

[37]J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2497-2504. IEEE, 2014. 5, 6

[38]C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In Computer Vision-ECCV 2014, pages 391-405. Springer, 2014. 4

[39]AbdulMuhsinM,FarahF.Alkhalid,BashraKadhimOleiwi, “OnlineBlindAssistiveSystemusingObjectRecognition”,InternationalResearchJournalofInnovationsinEngineeringandTechnology(IRJIET),Volume3,Issue12, pp 47-51, December–2019.

[40]Aishwarya Sarkale, Kaiwant Shah, Anandji Chaudhary, Tatwadarshi P.N., "A Literature Survey: Neural Networks for Object Detection", VIVA Tech International Journal for Researchand Innovation Volume1,Issue1(2018)ISSN(Online):25817280ArticleNo.9 .

[41]Bhumika Gupta, Ashish Chaube, Ashish Negi, Umang Goel,"Studyon Object Detection using OpenCV Python", International Journal of Computer Applications Foundation of Computer Science (FCS), NY, USA, Volume 162,Number 8, 2017.

[42]Geethapriya.S,N.Duraimurugan,S.P.Chokkalingam,"Re al Time Object Detection with Yolo", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249–8958,Volume-8,Issue-3S, February2019.

[43]Karanbir Chahal1 and Kuntal Dey," A Survey of Modern Object Detection Literature using Deep Learning", International Research Journal of Engineering and Technology (IRJET),Volume8, Issue 9, 2018.

[44]Kartik Umesh Sharma and Nilesh singh Thakur, "A Review and an Approach for Object Detection in Images",International Journal of ComputationalVision and Robotics, Volume 7, Number1/2, 2017.

[45]Mukesh Tiwari, Dr. Rakesh Singhai,"A Review of Detection and Tracking of Object from Image and Video Sequences",InternationalJournalofComputationalIntellige nceResearch,Volume13, Number5 (2017).

[46]R.Sujeetha,VaibhavMishra,"ObjectDetectionandTracki ngusingTensorFlow",InternationalJournalofRecentTechno logyandEngineering(IJRTE)ISSN:2277-3878,Volume-8,Issue-1,May2020.

[47]RichardSocher,BrodyHuval,BharathBhat,Christopher D.Manning,AndrewY.Ng,"ConvolutionalRecursiveDeepLea rningfor3DObjectClassification",InternationalConferenceo nComputationalIntelligenceandCommunicationNetworks, 2018.

[48]RichardSocher,BrodyHuval,BharathBhat,Christopher D.Manning,AndrewY.Ng,"ConvolutionalRecursiveDeepLea rningfor3DObjectClassification",InternationalConferenceo nComputationalIntelligenceandCommunicationNetworks, 2018.

[49]YordankaKarayanevaandDianaHintea,"ObjectRecogni tioninPythonandMNISTDatasetModificationandRecognitio nwithFiveMachineLearningClassifiers",JournalofImageand Graphics,Volume6,Number1,June2018.

[50]MaheshPawaskar,SahilTalathi,Shraddha Shinde,Digvijay Singh Deora "YoloV4 Based Object Detection for Blind Stick"International Journal of Innovative Science and Research TechnologyVolume 8, Issue 5, May – 2023.

[51] Mais R. Kadhim, Bushra K. Oleiwi "Blind Assistive System Based on Real Time Object Recognition using Machine Learning"Engineering and Technology Journal 40 (01) (2022) 159-165.

[52]J. Liang, D. DeMenthon and D. Doermann Geometric Rectification of Camera-captured Document Images IEEE Transactions on PAMI, Vol. 30, No. 4, pp. 591-605, 2008.

[53] Aditya Raj, Manish Kannaujiya, Ajeet Bharti, Rahul Prasad, Namrata Singh, Ishan Bhardwaj " Model for Object Detection using Computer Vision and Machine Learning for Decision Making " International Journal of Computer Applications (0975 – 8887) Volume 181 – No. 43, March 2019.

[54] Selman TOSUN, Enis KARAARSLAN "Real-Time Object Detection Application for Visually Impaired People: Third Eye". IEEE Conferences 2018.

[55] "Jayshree R Pansare, Aditi Gaikwad, Vaishnavi Ankam, Priyanka Karne and Shikha Sharma "Real - Time Text Reader" International Journal of Computer Applications 182(34):42-45, December 2018.

[56] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang "EAST: An Efficient and Accurate Scene Text Detector". Cornell University, Jul 2017.

[57] T. Guo, J. Dong, H. Li, and Y. Gao, Simple convolution neural network on image classification, 2017 IEEE 2nd Int. Conf. Big Data Anal. ICBDA 2017, pp. 721 724, 2017, doi: 10.1109/ICBDA.2017.8078730.

[58]Wei Xiang Dong-Qing Zhang Heather Yu Vassilis Athitsos, Context-AwareSingle-ShotDetector, 2018 IEEE Winter Conference on Applications of Computer Vision, pp. 1784-1793,2018.

[59] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, J. M. Z. Maningo, "Object Detection Using Convolutional Neural Networks", TENCON 2018-2018 IEEE Region 10 Conference, 2018.

[60] A. Huete, J. Victores, S. Martinez, A. Gimenez, and C. Balaguer Personal autonomy rehabilitation in home environment by a portable assistive robot IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., no. 99, pp. 1–10, 2011.

[61] A. Beck and M. Teboulle Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems IEEE Trans. Image Process., vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[62] Hashino, S., Ghurchian, R A blind guidance system for street crossings based on ultrasonic sensors Information and Automation (ICIA), 2010 IEEE International Conference on, 2010, pp. 476 – 481.

[63] Mauricio Menegaz, Understanding YOLO Hacker Noon, Hackernoon. 2018

[64] D. Dakopoulos and N. G. Bourbakis Wearable obstacle avoidance electronic travel aids for blind: A survey IEEE Trans. Syst., Man, Cybern.,vol. 40, no. 1, pp. 25–35, Jan. 2010.

[65] Hesch A. J., and Roumeliotis, S.I An Indoor Localization Aid for the Visually Impaired IEEE Internet Conf. on Robotics and Automation, Roma, Italy, 2007, pp. 3545 – 3551.

[66] C. Yi and Y. Tian Assistive text reading from complex background for blind persons Proc. Int. Workshop Camera-Based Document Anal. Recognit., 2011, vol. LNCS-7139, pp. 15-28

[67] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi "You Only Look Once: Unified, Real-Time Object Detection". Cornell University, Jun 2015

[68] J. Du, Understanding of Object Detection Based on CNN Family and YOLO, J. Phys. Conf. Ser., vol. 1004, no. 1, 2018, doi: 10.1088/1742- 6596/1004/1/012029

[69] P. Poirson, P. Ammirato, C.-Y. Fu, J. Liu, Wei Koeck, A. C. Berg, "Fast single shot detection and pose estimation", International Conference on 3DVision(3DV), 2016.

BIOGRAPHIES

Anita is Currently Research Scholar of MTech in CSE department, RSR Rungta College of Engineering and Technology, Bhilai, Chhattisgarh, India. She has earned under Graduation Degree from Chhattisgarh Swami Vivekanand Technical University Bhilai, Chhattisgarh, India. She has regular reviewer of many publications (like International Journal of Science and Engineering Technology. She has one design patents and has published. She has good rate of citation in Googlescholar: **(**https://scholar.google.com/citations?user=Om6qkq4AAAAJ&hl=en**).** Her area of interested is Artificial Intelligence, Image Processing, Deep Learning, Natural Language Processing, Machine Learning and many more.

Rupali Chandrakar is currently serving as an Assistant Professor in the faculty of Engineering at RSR Rungta College of Engineering & Technology, under the aegis of Sanjay Rungta Group of Institutions, Bhilai, Chhattisgarh, having a teaching experience of seven years in reputed engineering organizations. She has received her postgraduate degree from MATS University, Aarang, Chhattisgarh. She has published several research papers in various UGC-CARE journals, national and international conferences and has two designs on her name, with an expertise and research interest in Deep Learning.

Dr. Shikha Pandey is currently working as Assistant Professor in CSE department, RSR Rungta College of Engineering and Technology, Bhilai, Chhattisgarh, India and taking responsibilities of overall PG and PHD courses as a PG co-coordinator. She has earned her Under Graduation Degree from Central University Bilaspur, Chhattisgarh. She earned Post Graduate degree, Master of Computer Technology (MTech) from NIT Raipur. She has done her PhD from Chhattisgarh Swami Vivekanand University Bhilai. She has 13+ years of overall experience in Post Graduate Teaching and Research area. She has one design patents and has published many research papers in Scopus, UGC Care – National and International Journals. She has regular reviewer of many publications (like Indian Journal of Science and Technology and Premier Publishers houses. She has good rate of citation in Googlescholar**(https://scholar.google.com/citations?hl=en&user=vCMmBscAAAAJ)and Scopus indexed journal (https://www.scopus.com/authid/detail.uri?authorId=57193486232).**
Her area of interested is Natural Language Processing, Machine Learning, Plagiarism Detection, Data Mining and many more.