




# An improved deep learning-based optimal object detection system from images

Satya Prakash Yadav<sup>1,2</sup> · Muskan Jindal<sup>3</sup> · Preeti Rani<sup>4</sup> ·  
Victor Hugo C. de Albuquerque<sup>5</sup> · Caio dos Santos Nascimento<sup>5</sup> · Manoj Kumar<sup>6,7</sup> 

Received: 22 May 2023 / Revised: 14 August 2023 / Accepted: 31 August 2023 /  
Published online: 15 September 2023  
© The Author(s) 2023

## Abstract

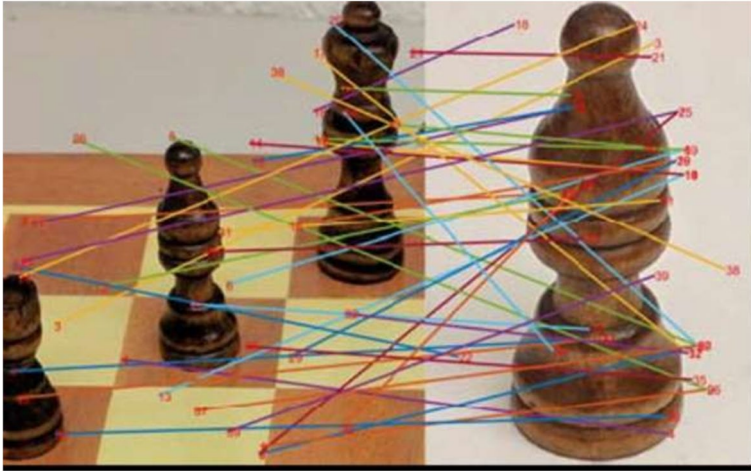
Computer vision technology for detecting objects in a complex environment often includes other key technologies, including pattern recognition, artificial intelligence, and digital image processing. It has been shown that Fast Convolutional Neural Networks (CNNs) with You Only Look Once (YOLO) is optimal for differentiating similar objects, constant motion, and low image quality. The proposed study aims to resolve these issues by implementing three different object detection algorithms—You Only Look Once (YOLO), Single Stage Detector (SSD), and Faster Region-Based Convolutional Neural Networks (R-CNN). This paper compares three different deep-learning object detection methods to find the best possible combination of feature and accuracy. The R-CNN object detection techniques are performed better than single-stage detectors like Yolo (You Only Look Once) and Single Shot Detector (SSD) in term of accuracy, recall, precision and loss.

**Keywords** Object Detection · Chess Piece Identification · You Only Look Once (YOLO) · Single Stage Detector (SSD) · Faster Region-Based Convolutional Neural Networks (R-CNN)

## 1 Introduction

Automation is not a significantly developing subject in the field of computer science or the domain of computer vision. Automating tasks has been the way for process optimization for many years. In primordial times, senior executives in organizations would make plans to perform a particular task. Once the workflow was all finalized, juniors would perform. Computer-based algorithms and systems perform automation to save time and human cognitive efforts. As a pervasive domain, computer vision uses automation, especially object detection, in its various applications: picture retrieval, security, observation, computerized vehicle systems and machine investigation.

Object detection facilitates automation in multiple ways like object identification, object classification, item tracking, and attribute identification like colour, features, and fine details of a particular object. While there are multiple use cases where object detection and



**Fig. 1** Visual Illustration into feature mapping of bishop with other fellow chess pieces

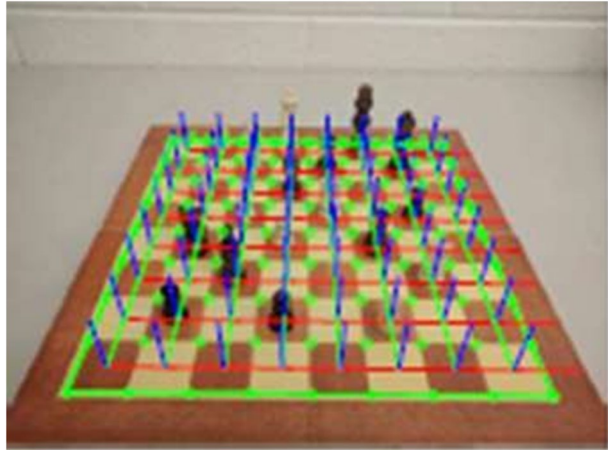
image segmentation techniques of computer vision have contributed towards automation, this research study focuses on chessboard mapping and chess piece identification- chess as a game is very complex, with multiple pieces, each with unique abilities but very similar features. The human eye might quickly identify these features but are abstruse an algorithm. To preview the problem set, Fig. 1 shows the feature mapping of the chess piece bishop [1] 2. It makes automation a comparatively simple task for algorithms like Artificial intelligence, image segmentation or object detection. Researchers in the past have encountered these challenges and, to address the same, implemented computer vision techniques like object detection- classification and image segmentations abstracts [3]. Thus, to automate a game of chess or provide players with a visual aid, some guided player assistance is cardinal to implement computer vision methodologies. It is necessary to map the chessboard and identify the discoverable paths for each position for chess player automation to succeed. The second is identifying and classifying individual chess pieces and meta-tagging their features, attributes, roles, and powers.

In the current application area- having input data from optical sensors for visual data, mapping discoverable paths on the chessboard and differentiating between various chess pieces are aspects of automating a chess game. Some sister techniques that could provide cutting-edge technology to aid automation in chessboard include- facial expression detection techniques [4].

Many researchers in the domain of computer vision have identified this parallel- requirement of visual data, real-time processing, identification and mapping of specific features. This parallel was leveraged to interchange techniques implemented in chessboard mapping and face recognition. Both areas face similar challenges- lack of high-resolution input data, the requirement for a robust framework to handle authentic images with noise, space-time complexity issues that occur in a real-time application, lighting issues in input images, feature similarity between different objects, the requirement of diverse data and extensive training [5, 6]. Thus, chessboard mapping becomes cardinal- where image recognition techniques are used to detect the on-board and vector areas as illustrated in Fig. 2.

A wide range of Convolutional Neural Network (CNN) algorithms are available for object detection, along with hybrid techniques. Several categorization frameworks,

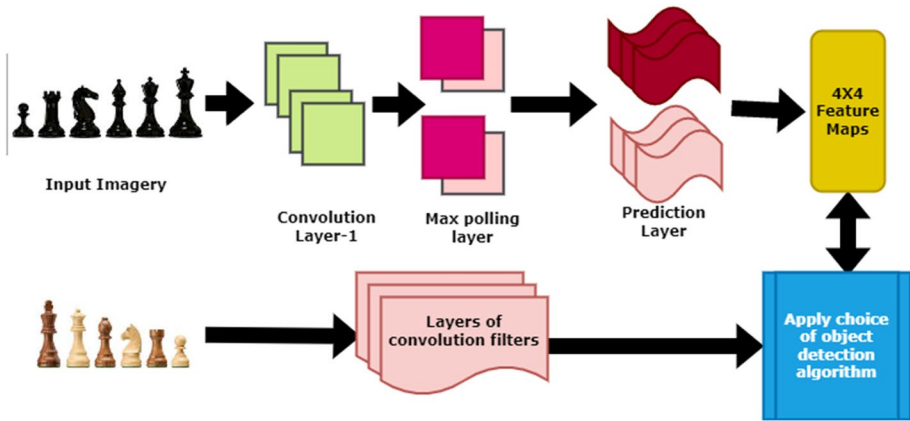
**Fig. 2** This is a classical pre-processing step in chessboard mapping where green indicates chessboard areas while blue indicates vector areas



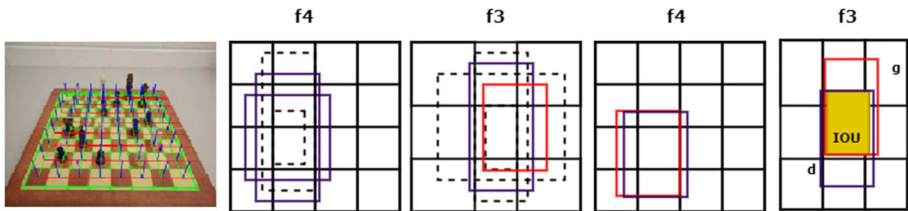
such as AlexNet and Inception, are effective for datasets containing extensive imagery data. Most studies have favoured Faster R-CNN over other object detection algorithms due to its superior performance, robust handling, and time–space complexity. In contrast to primordial object detection approaches like Histogram of Oriented Gradients (HOG), Region-based Convolutional Neural Networks (R-CNN), Spatial Pyramid Pooling (SPP-net) and Region-based Fully Convolutional Networks (R-FCN), Faster R-CNN uses insight gained from region proposal networks rather than primordial selection biased search. While Faster R-CNN might have an edge over other object detection frameworks due to its high similarity index, complex real-time detection, constant movement of chess pieces, human errors, difficulty tracking and detecting slight moments, and compromised pixel distribution, it suffers from significant performance degradation. Several competitive deep learning-based object detection frameworks, including SSD and YOLO, are capable of handling real-time object detection problems and have shown to be robust against human error and low-quality imagery input [7].

The most common research dilemma that often creates an impasse is a low threshold value might increase the recognition abilities of a wide range of objects, but the precision value drops exponentially. However, pushing a hard threshold value limits the framework’s recognition abilities, but whatever objects fall under the threshold are detected with precision [8]. A classical framework visualization of chess piece detection by applied object detection technique is shown in Fig. 3.

Another deliberate object detection technique is the Single Shot MultiBox Detector (SSD), which provides compelling performance when used to identify small objects and minor details like slight dislocation of an identified object [9]. Thus, its conditions and temperament are perfect for chess piece identification. Its implementation procedure is categorized into three gradations (1) Input imagery data undergoes segmentation to obtain multiple overlapped segments, (2) Each segment is then segregated and processed one at a time into the SSD network, (3) The output segments are merged into complete images in further two gradations, namely- sublayer segments are merged to obtain original image backdrop then identified objects are then merged box vice, refer to Fig. 4 for visual illustration.



**Fig. 3** Illustration of a classical framework visualization of chess piece detection by applied object detection technique



**Fig. 4** General architecture of the various procedures and sub-procedures implementation of Single Shot MultiBox Detector (SSD)

### 1.1 Objectives

This paper performs relatively better due to their deep learning capabilities, for the same Single Shot Detector (SSD) and You Only Look Once (YOLO) can provide effective results even when given noisy, a little blurred or degraded pixel distribution of imagery data. But for the same reason, i.e. the deep learning roots and attributes, training such models to reach a decent level of robust performance is challenging. A diverse, vast and variegated training set is required. The complexity of the training set example is directly proportional to the framework’s performance. Moreover, even with a decently diverse dataset, other issues like the requirement of enormous computation abilities and time–space complexity might not provide the prompt real-time input needed in real-world applications. The real-time object detection problem sets have predefined threshold parameters to differentiate between objects to be detected and objects not to be detected. It adds further complexities as recall and precision values are often hard to keep up under-explained circumstances. One must find a perfect balance between the threshold and precision-recall values by hit-and-trial at multiple values and weighted parameters, which is contingent on each framework.

1. In this paper, we compared the performance of the different types of object detection algorithms, i.e. Multiple Stage detectors like R-CNN, Faster R-CNN (Faster Region-

- Based CNN) and Single Stage Detectors like Yolo (You Only Look Once), Single Shot Detector (SSD).
2. The object detection algorithms are performed over publicly available real-world datasets.
  3. The performance of the object detection techniques is compared based on accuracy. The Faster R-CNN algorithm is performed over the other convolution neural network.

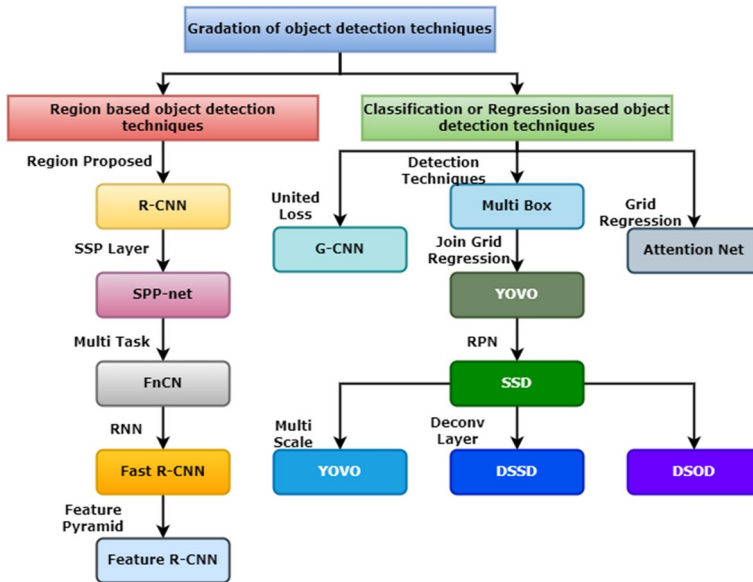
## 1.2 Paper organization

The rest of the paper is organized as follows- Section 2 discusses a detailed literature review of object detection techniques and their contingent areas of applications with publicly available real-world datasets via tabular other formats. Further, Section 3 shows the methodology of the implementation process and proposed framework presented. The numerical and empirical results are displayed and compared in tabular format in the Section 4. Finally the paper is conclude in the Section 5.

## 2 Literature review

Object detection is a vast domain with a variegated application area and multiple implementation techniques. This section of the proposed study aims to elaborate, assay, comprehend and perform a comparative analysis of various classical and contemporary object detection techniques and their contingent areas of applications with publicly available real-world datasets [8, 10].

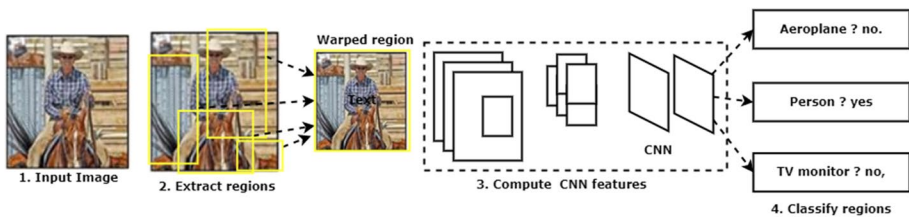
The process of identification of certain items, among others, then classifying them into predefined labels such that each object has its unique set of attributes- entices within the concept of object detection. Although there is a plethora of object detection, each with an individual set of merits and demerits contingent on their area of application, implementation details and algorithm of choice- this study has categorized classical object detection techniques into two significant gradations- (1) Region-based object detection techniques and (2) Classification/ Regression-based object detection techniques [11]. The former-Region-based object detection techniques follow a step-by-step procedure. Primarily each input image is segmented into regions or areas of interest and then later graded into predefined labels. In contrast, the latter- Classification or Regression-based techniques follow a more wholesome approach instead of a serial step-by-step procedure, where image classification and item recognition is done within the same framework- adopting a comparatively suitable technique to avoid in-process noise and issues. Object labelling and feature recognition in Region-based techniques is the last step, i.e. third gradation in serial order. In classification or Regression-based techniques, item recognition and feature mapping are performed within the unified workflow. Some techniques that are categorized are Region-based techniques are- Region-based Convolutional Neural Networks (R-CNN), Spatial Pyramid Pooling (SPP)-net, Fast Region-based Convolutional Neural Networks (R-CNN), Faster Region-based Convolutional Neural Networks (R-CNN), Region-based Fully Convolutional Network (R-FCN), Feature Pyramid Networks (FPN), Deep neural mapping support vector machine (DNMSVM) [12] and Mask Region-based Convolutional Neural Networks are among few examples [9, 13, 14]. According to Fig. 5, various object detection techniques and their algorithms are graduated.



**Fig. 5** A comprehensive gradation of various object detection techniques along with their various algorithms

Second gradation- Regression or classification-based object detection techniques include MultiBox, AttentionNet, G-CNN, YOLO, Single Shot MultiBox Detector, YOLOv2, Deconvolutional single shot detector (DSSD) and Deeply Supervised object detectors (DSOD). Recently Faster R-CNN was introduced, which amalgamates concepts from Region-based techniques and classification-based item detection techniques [15–17]. While multiple variations of R-CNN have been introduced and implemented over the years, a classical R-CNN framework can be categorized into the following steps- (1) Gathering Input images, (2) Extracting regions from input images, (3) Wrapping extracted features (4) Processing wrapped features via CNN filters (4) Categorize and classify gathered features into predefined labels as illustrated in Fig. 6 [7].

A step-by-step Region-based framework consists of a double gradation procedure- emulating the cognitive abilities of the human brain by identifying a region of interest (ROI). Some related networks include [18–20]. Overleaf [21] implements CNN among multiple models to gather insights about navigation-based object detection mapping features. Some



**Fig. 6** The procedure of object detection via implementing R-CNN is elaborated as- (1) Gathering Input images, (2) Extracting regions from input images, (3) Wrapping extracted features, (4) Processing wrapped features via CNN filters, (4) Categorize and classify gathered features into predefined labels



variations of R-CNN are elaborated. Table 1 presents a tabular comparison between Faster R-CNN and other object detection techniques, including Faster R-CNN, R-CNN, SSD, and YOLO.

Region-based Convolutional Neural Networks (R-CNN)- Multiple variations of CNN have displayed multiple implementation-based applications that teach deep learning concepts and their contingent architectures. But even though the performance of most of these techniques was decent, most could not capture high-definition attributes and features. To address this issue of scope and accuracy, R-CNN was introduced by [22] that gained much attention due to its robust performance and accuracy with credentials like mean average precision (mAP) of 53.3%, a 30% hike from classical CNN models on the PASCAL dataset. Some gradations and specifications of the R-CNN are mentioned below.

- a. **Region Identification or Proposal:** An essential preprocessing step in object detection is to segment the input image into regions of interest or region proposals. R-CNN implements an elective search strategy to obtain around 2000 segmented regions of interest for any input image. A classical bottom-up (BU) based technique is applied along with proxy areas to use the search space and processing abilities more optimally.
- b. **Deep Feature Extraction based on CNN:** This is the second gradation or step in the R-CNN anatomy, also known as the wrapping phase, where the region segments or region proposal obtained from the previous step are manipulated by wrapping, fixing or cropping to fit the requirements of CNN model- which aids in extracting multi-dimensional features for next step. High-resolution, robust and astute-performing attribute visualization is obtained after utilizing CNN models' multiple feature extraction filters.
- c. **Categorization and Localization:** The extracted features and regions obtained from the preceding steps are classified into predefined labels in this step. Support Vector Machines (SVMs) are generally implemented for Region-based labelled classification.

Although the previous decade has witnessed path-breaking improvements and developments in object detection and item feature recognition in computer vision, certain limitations are still considered open challenges [24]. This study has identified and collated such loopholes contingent on the research paper, area of application and implemented dataset in tabular form below as Table 2.

Researchers have very well recognized the issues elaborated above and have made multiple attempts to address these issues by proposing optimized techniques that amalgamate multiple concepts to provide hybrid methodologies. Some examples include- Geodesic object proposals [19] that can quickly solve the time complexity issue by replacing classical image segmentation techniques and graphs and performing geodesic-based graphing or region segmentation instead. Time-space complexity issues that pertain to multiple hierarchical segmentations such that each segment belongs to a different group with an individualistic set of features are solved by Multi-scale combinatorial grouping [29], which adopts edge boxing techniques replacing time-consuming visual boxing methods. Some sister techniques of edge boxing include DeepBox [28] and SharpMask [23].

Apart from the elaborated frameworks, techniques and methods, there are plenty of other object detection techniques, each with an individualistic proposal and loss function implemented on a unique platform and using a particular programming language. An overview of classical object detection and classification framework contingent on their proposal, loss function, implanted platform and language is provided in tabular format as Table 3.

**Table 1** A tabular comparison of Faster R-CNN with other techniques like Faster R-CNN, R-FCN, SSD and YOLO for detecting an object

Technique implemented	Merits	Limitations
Fast R-CNN [6]	Its accuracy is 25 times faster than the classical R-CNN approach, with a time complexity of fewer than 20 s due to its single-iteration approach	The application of a third-party region generator creates the implementation bottleneck
Faster R-CNN [22]	Its accuracy with real-time results makes it the right choice for real-time object detection techniques- 0.12 per image	The extensive computation abilities require parallel computing- making it unsuitable for actual time application
R-FCN [23]	Requires less time than R-CNN when processing test set	The mAp performance matrix is slightly lower than that of R-CNN
Mask R-CNN [7]	The segment location and positioning of the background are more accurate	Due to higher time complexity, this technique cannot be used in real-time applications
YOLO [9]	The precision and accuracy of the object detection algorithm within limited time complexity make it ideal for real-time applications and contingent frameworks	This lack of accuracy in detecting minute details and small objects
SSD [15]	Applying one iteration or single network makes it faster than classical techniques like R-CNN and its contingencies	The objects' detection accuracy is lower than the Fast-RCNN and Faster-RCNN methods



**Table 2** Limitations and open challenges in implementing CNN in object detection contingent on their research paper, area of application and implemented dataset

Authors	Research Objective	Problem Statement	Application Area	Dataset description	Dataset Link
[25]	In the deliberated network, the FC layer is introduced in the CNN framework about a requirement of input images of a particular size, $227 \times 227$ . It creates configuration issues in case of dataset acclimation, and the time complexity for the test set is also exponentially increased. If a validation set is introduced to increase accuracy or precision, then there is a lack of diverse and comprehensive datasets for extensive training	Previous approaches have often framed the problem as a contrast analysis problem	Salient Object Detection	PASCAL-S- Dataset used for classical salient item detection with 850 imagery sets, extracts validation sets from PASCAL VOC 2010 archive	<a href="https://ccvl.jhu.edu/datasets/">https://ccvl.jhu.edu/datasets/</a>
[26]	A multi-step complex training procedure is implemented for this particular R-CNN network. First convolution network is fine-tuned according to attributes or features of the dataset, and then classifier replacement occurs to fit the model. Classifier replacement is time-space-consuming, and implementing the complete procedure requires abstruse computation abilities that need parallel computing in the real world	A saliency map is further used for segmenting salient objects unsupervised	Region detection	The complex Scene Saliency (ECSSD) dataset consists of 1000 images with simple semantic attributes but complex structures	<a href="https://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/datas et.html">https://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/datas et.html</a>

**Table 2** (continued)

Authors	Research Objective	Problem Statement	Application Area	Dataset description	Dataset Link
[27]	Salient object detection is primarily a task that requires a higher level of sensitization towards features in the object, which is achieving my time-space extensive training. Since features are extracted individually from multiple regions in a serial manner, the time consumed exponentially increases. Moreover, each small region is trained independently with deep networks like VGG16, which hikes the increased time complexity	It cumulatively leads to the requirement of enormous memory slots for processing	Salient Object detection	HKU-IS- Used for robust saliency prediction algorithms and techniques with 4447 imagery sets, each complex due to contrast variations and more than one salient item to be detected	<a href="https://paperswithcode.com/datas-et/hku-is">https://paperswithcode.com/datas-et/hku-is</a>
[28]	The deliberated study implements selective search with the short network that makes feature extraction and region segmentation easier. Still, the serial implementation nature of selective search consumes a lot of time. Roughly 4 s are required to extract 2 units of regions	Sometimes, multiple iterations are required for extensive time-space complexity to receive high recalls	Object detection	SOD- Salient Objects Dataset based on Berkeley Segmentation Dataset (BSD) consisting of 300 imagery sets from 7 diverse areas	<a href="https://www.elderlab.yorku.ca/resources/salient-objects-datas-et-sod/">https://www.elderlab.yorku.ca/resources/salient-objects-datas-et-sod/</a>

**Table 3** An overview of classical object detection and classification framework contingent on their proposal, loss function, implanted platform and language

Method and reference	Proposal technique	Advantage & Disadvantage	Performance Metric
R-CNN [29]	Selective search algorithm	A current solution uses a data-driven approach to collect large-scale datasets with object instances under different conditions. It is tested only on the VOC dataset	Hinge loss classification and Bounding box regression
SPP-net [30]	Edge Boxed	It worked on deep learning-based object detection and worked on a commonly used dataset	Hinge loss classification and Bounding box regression
Faster R-CNN [23]	Selective search algorithm	Due to advances like SPPnet and Fast R-CNN, the computation of region proposals has become a bottleneck in these detection networks. Its work only hypothesizes object locations	Class log loss and Bounding box regression
Faster R-CNN [22]	RPN	Used the Region Proposal Network (RPN) for advancement. Its work only hypothesizes object locations	Class log loss and Bounding box regression
FCN [21]	RPN	An approach based on supervised learning is used for segmenting multi-level images. The contributions are used only two-fold	Class log loss and Bounding box regression
Mark R-CNN [27]	RPN	Finding salient vertices and hyperedges in a hypergraph becomes the problem of salient object detection	Class log loss, Bounding box regression and semantic sigmoid loss
FPN [21]	-	Just adopt the FPN, as with most previous methods	Class log loss and Bounding box regression
YOLO [7]	-	The YOLO framework allows for increased prediction scales and reduced network depths for salient object detection	Class sum error loss, Bounding box regression, object confidence and background confidence
SSD [4]	-	SSD cannot detect small objects, except for multi-objects with different scales simultaneously	Class softmax loss and Bounding box regression
YOLO V5 [9]	-	Based on the YOLOv5 model, four methods are proposed to improve the precision of small object detection	Class sum error loss, Bounding box regression, object confidence and background confidence

Other than techniques elaborated in tabular and textual format, some areas of continuous optimizations are still an open challenge for researchers, like- handling the massive number of annotated images with high pixel quality backgrounds without increasing time–space complexity within limited computation complexity. To handle this issue of process optimization with annotated images, [22] presented an algorithm for selection automation that reduces time and effort spent on model training, testing and feature extraction, known as an effective online algorithm (OHEM). Building on the same, the author [23] emulates the concept of selected deep convolutional frameworks for region-specific classification with networks like ResNet [27] and GoogLeNets.

Author [31] compared the seventeen-related models with the proposed model using six publicly available datasets. The experimental section shows it generates the best results by comparing the proposed model to other models based on area under the curve, recall, precision, and F-measure. The author's proposed approach includes videos from both a local and a global perspective [32]. Eratosthenes Sieve engagement significantly enhances clustering procedure performance. Qualitative and quantitative evaluations and complexity computations are carried out to compare the proposed model's performance with state-of-the-art models.

Author [33] proposes a local-alignment-based FASTA method for summarizing events in multi-view videos. A deep learning framework is used to extract the features to resolve variations in illumination, remove fine texture details, and detect objects in a frame. Local alignment of multiple video views is then used to capture interview dependencies among multiple views. The frames with low activity are then extracted using object tracking.

Author [34] developed a novel method for detecting emotion by examining lip structure over time. Analyzing the pattern with time using the recurrent neural network provided the classification of emotions into six categories. Both qualitative and quantitative evaluations are conducted to compare our proposed model with state-of-the-art models. The author [35] approaches the problem of the automatic fingerspelling recognition system by utilizing the concept that the brain changes some information to perceive things efficiently. A novel key-frame extraction technique is proposed to summarize the video lectures in real time so that readers can get critical information [36]. Author [36] is compared to state-of-the-art models both qualitatively and quantitatively.

The author [37] introduces a text-based query technique for summarizing and searching events in multi-view videos on the cloud. The features of moving objects in frames are extracted using a deep learning framework. A local alignment captures the inter-view dependencies among multiple views of the video. An important challenge in American Sign Language (ASL) is the recognition of dynamic hand gestures. 3-D CNNs, which can recognize patterns in volumetric data like videos, are employed to solve the challenges of dynamic ASL recognition [38].

Author [39] proposes a method for automatically extracting and recognizing a player's jersey number using optical character recognition (OCR). A hierarchical computation framework is presented to identify the player in an image that utilizes low- and high-level vision features. A study by the author [40] indicates that multimedia content over the cloud is better protected than existing approaches, where information cannot be unwrapped within a reasonable timeframe. Numerous cryptographic algorithms, including RSA, ElGamal, and Diffie Hellman, rely on unity. The author [41] proposes a technique for detecting anomalies in crowded scenes that is efficient and rapid. A deep learning framework (CNNs) is used to train our model using an advanced feature-learning technique. As the initial stage of our network, we first rescale the small frame, then use a more complex and deeper CNN to assess the remaining features of interest. The classical YOLO was

introduced and implemented, and its various versions, like versions 1, 2, 3, 4 and 5, were then created, each with its individualistic properties in Table 4.

The previous decade has seen variegated developments with the launching of various versions of YOLO, each with individualistic capabilities, merits and limitations. It's cardinal to provide insights to describe each version.

### 3 Proposed methodology

The proposed methodology is discussed in this section. The proposed methodology is considered three different types of improved deep-learning based object detection algorithms, i.e. Multiple stage detectors like R-CNN, Faster R-CNN (Faster Region-Based CNN) and Single Stage Detectors like Yolo (You Only Look Once), Single Shot Detector (SSD).

#### 3.1 YOLO algorithm

Most object detection algorithm tends to identify or map objects based on Region-based analysis, where the algorithm segments imagery data into multiple segments. Then, the framework tends to search only in areas where the probability of finding objects is maximum. In either case, the algorithm analyses the imagery data in multiple passes instead of identifying objects at once by viewing the whole image. Thus, You Look Only Once (YOLO) analyses the complete image simultaneously for object detection instead of taking multiple segments and iterations. YOLO is one of the fastest, simple and high-performing algorithms due to its one-step process and easy-to-no training- saving time and space complexity. Regression-based algorithms can be used on fresh images with processing capabilities of 45 frames/second [29]. Some researchers also consider it an optimized extension of Region-based convolutional neural networks. A classical neural network framework for You Look Only Once (YOLO) with around eight layers of the convolutional neural network is displayed in Fig. 7.

The detailed procedure of the You Look Only Once (YOLO) framework includes extracts from a Region-based neural network but is implemented optimally. To gather contextual information about classes and image details, YOLO analyses the image during limited training and test sets by processing the complete image simultaneously. Due to the lack of multiple iterations and robust procedures, its performance matrices are twice as effective as classical R-CNN-based frameworks with half as many background networks.

The Backbone of Yolo is a Feature Pyramid Network (FPN) followed by the Yolo Head consisting of 4 convolutional blocks, each concatenated with the backbone. Output from each bottleneck layer in Yolo's Head is passed through a convolutional layer for object detection.

A single-stage object detection algorithm, YOLO v5, has three main components, as shown in Fig. 8.

- a. **The backbone** is used to extract essential features from the given input image. In YOLO v5, the **CSP** (Cross Stage Partial Networks) are used as a backbone to extract informative feature maps from an input image.
- b. **A neck** that is mainly responsible for generating feature pyramids. Feature pyramids help models to generalize better on object scaling. Also, it helps to identify the same object with different sizes and scales.

**Table 4** Elucidating various versions of YOLO

YOLOv1 [42]	YOLOv2 [43]	YOLOv3 [44]	YOLOv4 [45]	YOLOv5 [46]
YOLO version 1 comprises 26 layers- 24 convolutional layers with 2 fully connected. It performance is better than the classical region based techniques for convolutional neural networks	This version includes batch normalization layer as part of 30-layer Network. The concept of Anchor boxes for optimal performance provides it with an edge over fellow algorithms	Version consists of 106 layers of neural network such that detection takes place in 3 phases-9 anchor boxes. Its performance is better than YOLO1 and YOLO 2 with optimal performance in small object identification	This version's neural network has many new features like- a modified Path aggregation network, an optimized spatial attention module, and spatial pyramid pooling to increase accuracy levels with limited training and look only once methodology	Concepts of dark net by Pytorch are implemented in YOLO-5 with contingencies with PAF- NET, augmentation and bounding box anchors

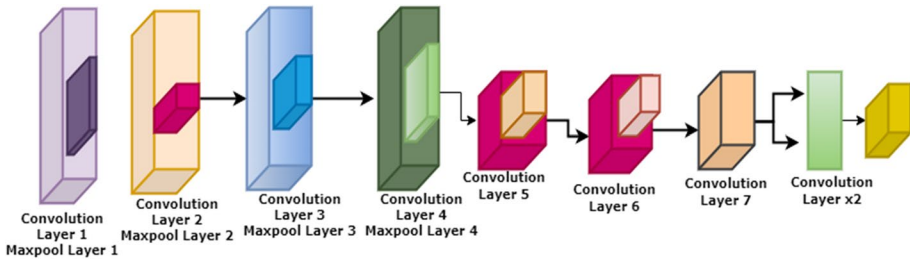


Fig. 7 A classical neural network framework for You Look Only Once (YOLO)

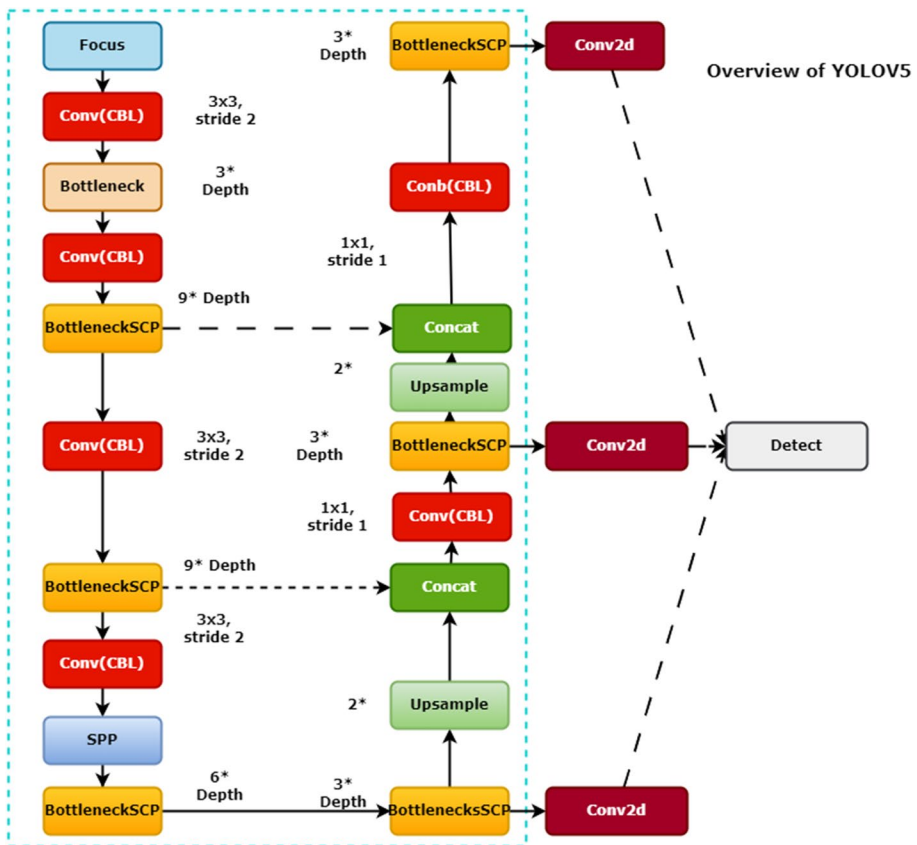


Fig. 8 Overview of Yolo v5 architecture

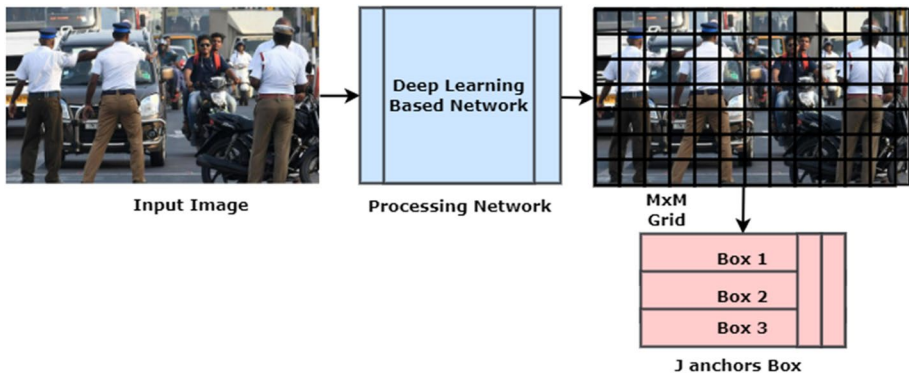
- c. **Model Head** that is used to perform the final detection part. It applies anchor boxes on features and generates final output vectors with class probabilities, object-ness scores, and bounding boxes.

Based on Fig. 8, input is fed into YOLOv5 Backbone via the focus layer. This layer down-scales the input resolution to match the rest of the model architecture and rearranges spatial



**Table 5** Architecture of Yolo with FPN backbone is used, along with layer specification

Layer Name	Channels	Specifications
Focus Layer	64	kernel size = 3 × 3
Convolutional Layer	128	Kernel size = 3 × 3, Stride = 2
Bottleneck CSP	128	Depth = 3
Convolutional Layer	256	Kernel size = 3 × 3, Stride = 2
Bottleneck CSP	256	Depth = 9
Convolutional Layer	512	Kernel size = 3 × 3, Stride = 2
Bottleneck CSP	512	Depth = 9
Convolutional Layer	1024	Kernel size = 3 × 3, Stride = 2
SPP	1024	Kernel Size = 5 × 5, 9 × 9, 13 × 13
Bottleneck CSP	1024	Depth = 6



**Fig. 9** Illustration of an input image processed into a deep learning network to obtain an MxM grid with five boxes/anchors

data blocks into depth. It is followed by a convolution layer, which matches the number of channels desired. After that, it is passed through the Bottleneck CSP layer, which is used to improve learning by passing on an unedited version of the feature map. Table 5 below gives the specification of each layer used in the backbone.

### 3.1.1 Box/anchor grid positioning of the input image

An input image is further segregated into grids and boxed as a preprocessing step in the object detection gradation step. Primarily, this gradation includes making an imagery grid of a size- M, making the complete grid’s size MxM and each grid’s size 'J' boxes or anchors. A deep learning network processes an input image into five boxes and anchors, as shown in Fig. 9.

Post creation of anchors and an MxM grid, the predicted height and width of the output image are obtained. The next step is calculating the confidence value for the output image via the IOU of boxes. Each grid predicts 'Pc' conditional loss probabilities Pr(Class|Object) is also obtained along with IOU. These values measure the model’s confidence in predicting that the obtained box contains the object. The formulae denote it.

$$\Pr(\text{Object}) * \text{IOU}$$

The confidence value is directly proportional to the probability of the existing object in the obtained box; the confidence value is zero if the object is not present in the obtained box, and it is 1 if the object is present in the respective box. This principle is known as bounding box prediction, gauged by Intersection Over Union (IOU) metric, as illustrated in Fig. 10 below [13].

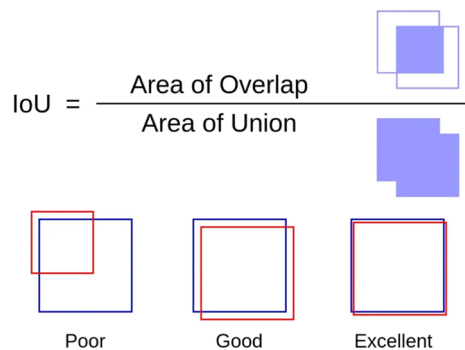
Apart from the implementation, as mentioned earlier, details of YOLO, another technique can exponentially increase YOLO's efficacy, i.e., Non-Max Suppression. This algorithm covers one of the most common loopholes in object detection algorithms – detecting the same object multiple times due to multiple iterations of detection schedules in multiple regions. Non-Max Suppression reduces this by decreasing the frequency of predicting boxes by finding the probability of most giant boxes and just deliberating it instead of all boxes. After selecting this box- all other boxes are gauged for their prediction values, and the box with the highest IOU value is selected. This process is repeated until all the boxes get selected or suppressed, and we get our final bounding boxes [27].

### 3.2 SSD (Single Shot Detector)

Classical and contemporary object detection algorithm families like R-CNN clan and YOLO, along with their various versions, provide higher values in accuracy matrices but lack robustness and are not suitable for actual time application due to their extensive time–space complexity requirement. On the contrary, Single Shot Detector (SSD) provides optimal accuracy with limited time–space complexity. This subsection elaborates upon the intricacies of SSD and its various versions [9].

The SSD framework consists of two primary components: the backbone model, a classical pre-trained convolutional neural network-based feature extractor, and a head responsible for precise object detection by implanting insights from the backbone model's extracted features. Many researchers in contingent models for object detection often manipulate the backbone layers by introducing some modifications like extra convolutional layers, drop layers, batch normalization layers, classification layers and fully connected layers to fine-tune and provide optimized performance values. But in most cases, no significant modifications are made to the convolutional layers in the end – SSD Head to avoid complexities. As the added layers move towards the SSD head, their size and number of convolutional elements diminish progressively, thus preserving spatial complexity. Every layer in the

**Fig. 10** Visual representation of Intersection over Union (IOU)



network has the capability and responsibility to detect the allotted number of features or objects and then pass the obtained results to the subsequent layer towards the SSD head via convolutional filters. This progressive decrease in convolutional neural layers across the network results in a steep decrease in feature map size and depth increase, segregating all layers into two categories- deep layers and initial convolutional layer filters. The unique characteristic of the deep layer is responsible for manipulating large receptive domains and producing abstract projections, making it optimal for robust large object detection problem sets.

In contrast, initial convolutional layers can provide insights for small object detection present in input imagery. This wholesome combination of deep and initial convolutional layers provides optimal performance by providing semantic insights while preserving the spatial structure of low-resolution images. The final findings from SSD are then interpreted by bounding boxes and objects from various classes. Confidence prediction and accuracy measurements are performed by multiple feature maps from multiple sizes that represent various scales [21]. A classical Single Shot Detector architecture is illustrated below in Fig. 11.

Most SSD techniques don't use sliding window-based techniques; instead, they implement a grid-based technique with each grid cell to detect objects in variegated regions of the input image. This methodology of the region-based techniques is very straightforward-when each grid is searched for pre-labelled objects, and the detecting objects are then classified into predefined classes or labels. If no particular object is detected in a particular grid, it is considered a background image by default. On the contrary, if multiple objects exist in a single grid, other techniques like anchor box and receptive field exist. One grid for each respective anchor or box is allotted, so assigned boxes or anchors are predefined [4, 5].

### 3.3 Faster R-CNN (Extension of fast R-CNN)

Region-based Convolutional Neural Network is a classical and well-explored algorithm that primarily segregates the image into multiple areas where in selective search algorithm is applied to each segregated area [27]. In Faster R-CNN, the training area and time-space complexity are reduced by implementing an ROI pooling layer within the

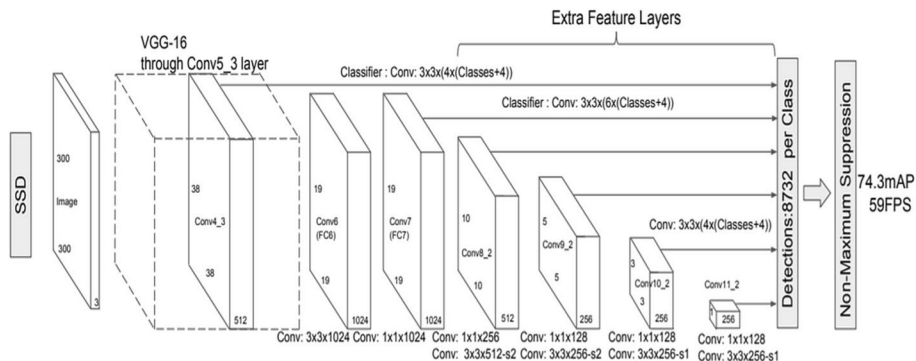


Fig. 11 A classical Single Shot Detector architecture

convolutional neural network layer, such that each iteration of the algorithm recreates about 2000 layers to reduce the total training time.

### 3.3.1 Classical framework of faster R-CNN algorithm

Thus, to reduce training time and space complexities, researchers and academics segregate the procedure of R-CNN into two gradations- primarily being a deep convolutional network coined as Region Proposal Network (RPM) with gradation two as Fast R-CNN detector that is implemented in deliberated regions to reduce training time and space- this provides it with an edge over fellow techniques when implemented in real-time applications. The Region Proposal Network (RPN) first deliberates the input image and provides a branched image with identified objects and background as output. Multiple convolutional network filters generate proposals for Region-based object and background identification. Numerous convolutional neural networks with a window of  $J \times J$  as introduced to extract features using the low dimensional feature extractor in the sliding CNN network layers. A similar network is then replicated via two fully connected layers- a classification layer and a box-regression layer. There are multiple sliding windows, so each window aids the segregation of numerous Region-based areas, where the maximum number of possible region proposals is depicted by 'k'. Next to this network is the regression layer-also aids in feature prediction and object identification with  $4 \times k$  output metric. Each output metric contains k boxes with a bounding box as centroid and output of  $2 \times k$  classification layers. Each convolutional feature map of a size  $W \times H$  has  $W \times H \times k$  anchors, as displayed in the network shown in Fig. 12 [19].

Although many budding researchers have implemented faster R-CNN networks and have provided convincing results bust, it still has many limitations—the training data and model used to train an ideal Faster R-CNN has predefined and tuned anchors, each with a size of 256. But actual data sets are not pre-tuned and correlated, creating the need for intensive dataset preprocessing [16].

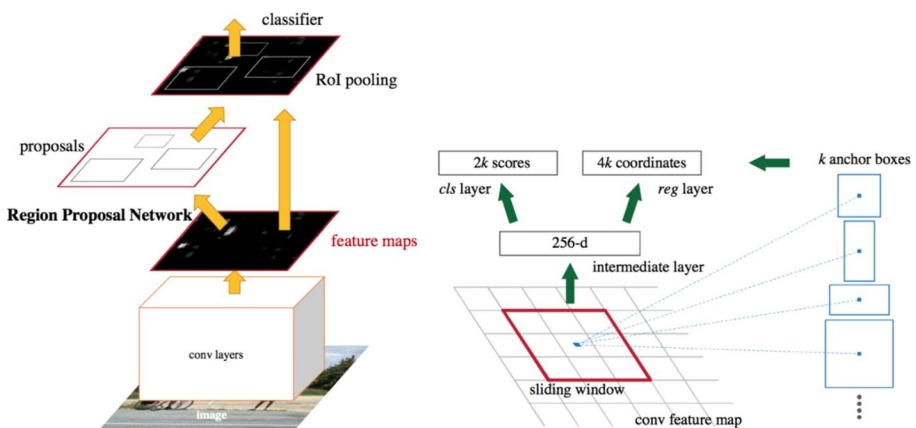


Fig. 12 A classical illustration of Faster R-CNN Architecture

## 4 Results analysis and discussion

This section discusses the results obtained from training our Object detection models. A comparative analysis of the object detection algorithms such as Multiple Stage detectors like R-CNN, Faster R-CNN (Faster Region-based CNN) and Single Stage Detectors like Yolo (You Only Look Once), Single Shot Detector (SSD). These models use images as input, detect objects in a single run, and can be deployed on a mobile device. In addition to performing multi-stage object detection, Faster R-CNN provides cutting-edge performance when it comes to accuracy.

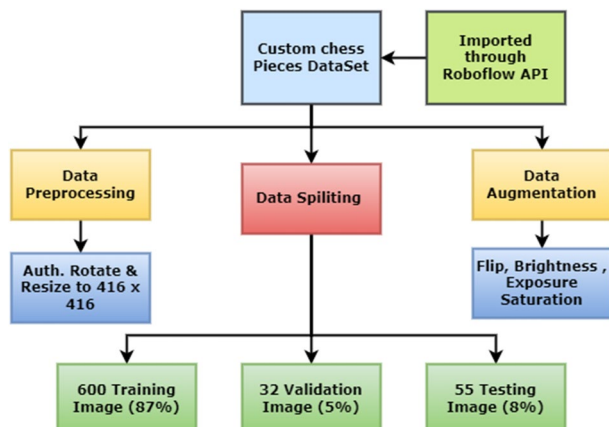
A comparative analysis of Faster R-CNN, Yolo, and SSD algorithms are used the Custom Chess Piece Dataset in Google Collaboratory Notebooks using Tensorflow Object Detection API and Roboflow. It utilizes Tensorboard, an interactive tool by Tensorflow, for visualizing the training and evaluation metrics. The graphical representation is designed on the MATLAB 2020a. The Window 11 based system with 16 Gb RAM and GTX GPU is consider for the execution.

### 4.1 Dataset

This paper modified the Chess Piece dataset from Roboflow Public Datasets according to our requirements. We have kept the number of images intact, that is, 693. They are RGB images with a size of  $416 \times 416$  pixels containing 12 classes at maximum in one image (six of white & six of black), as displayed in Fig. 13.

There were originally 289 images in the raw dataset. Still, they have done preprocessing on the raw dataset like resizing each image to  $416 \times 416$  auto orient and also applied augmentation techniques like Flip (Horizontal), Crop (15% Max Zoom), Shear ( $+6^\circ$  Vertical & Horizontal), Hue (between  $-15^\circ$  &  $15^\circ$ ), Saturation (between  $-15\%$  &  $15\%$ ), Brightness (between  $-10\%$  &  $10\%$ ), Exposure (between  $-10\%$  &  $10\%$ ). These augmentations are applied 3 times on the dataset making it 693 images. Then we split the dataset into training (87%), Testing (8%), and Validation (5%) sets.

**Fig. 13** A visual description of the dataset used to implement various algorithms in the proposed study

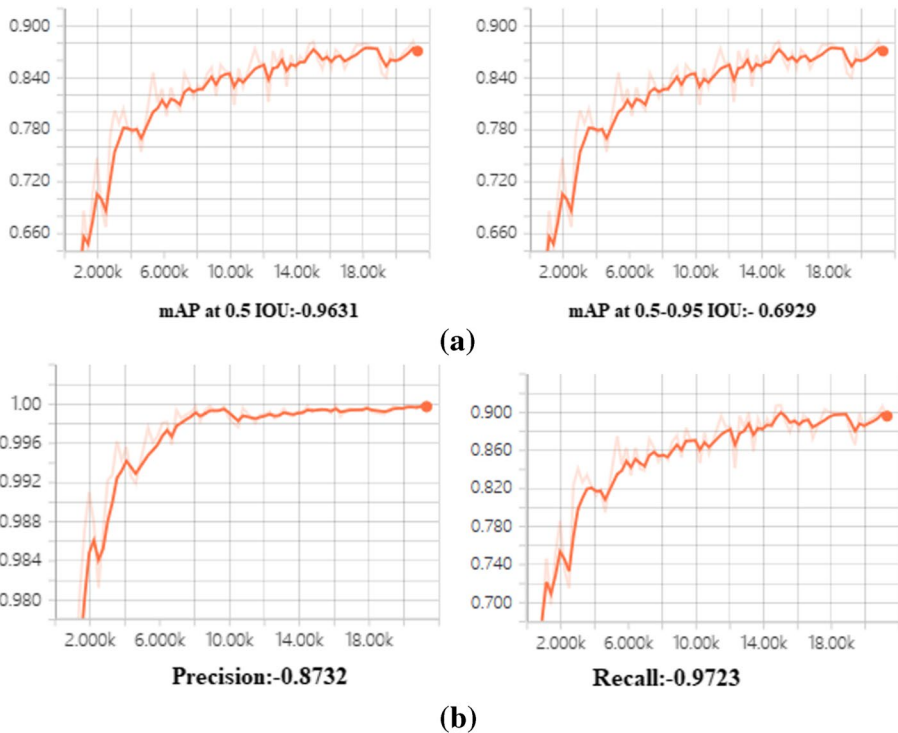


### 4.2 Experimental results

Object detection requires multiple steps, making it difficult to implement in real-world applications due to a lack of computation capabilities, in contrast to other techniques, which are more accurate but require fewer computations. In this section measure the performance of all three models based on their accuracy, precision, recall and classification loss.

The Yolo v5 Model performance elaborate with its specifications and Tenor board graphs with metrics like Object loss, mAP, Classification loss, Localization loss, RPN loss & Box loss, on which the performance of algorithms are evaluated for a better understanding. The proposed research venture and its contingent results in terms of graphical evaluations, visual comparisons and their respective value descriptions of quantitative grounds are described below in Fig. 14.

mAP (Mean Average Precision), in simple terms, we calculate it by taking the average of Average Precision. As we know, object detection Algorithms predict bounding boxes with their class labels. This metric is known as IOU, i.e. (Intersection Over Union). Therefore, we calculate the recall and precision metrics of the algorithms using an IOU Threshold. For example, mAP at 0.5 IOU means we have set 0.5 as the threshold value for IOU, so if the value of the predicted bounding box is more significant than 0.5, it will be considered as True Positive (TP) else it will be classified as False Positive (FP). Similarly, mAP



**Fig. 14** Results and performance matrices obtained for YOLO Algorithm with contingent optimal performance and parameters

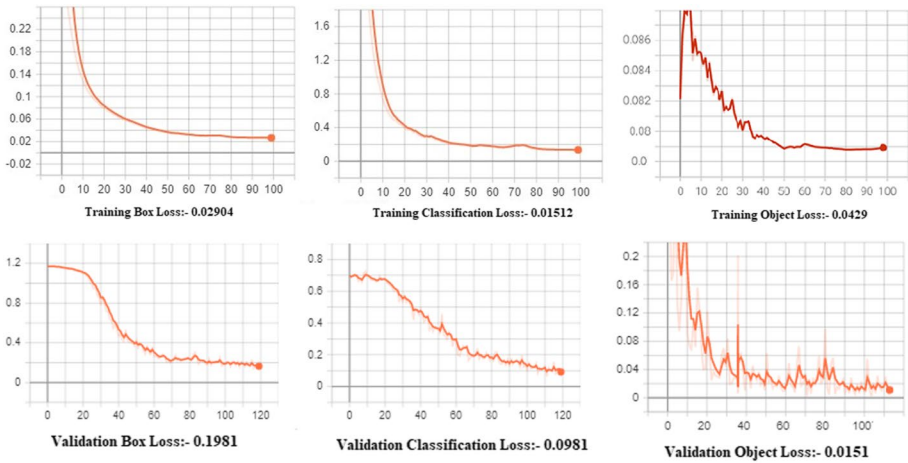


Fig. 15 Graphical description of YOLO loss matrices

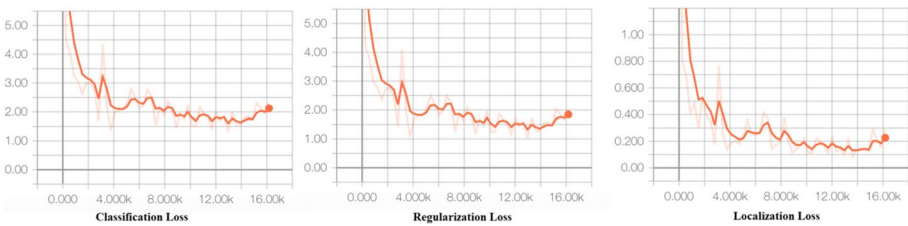


Fig. 16 SSD loss matrices for training and test sets with three types of loss description

[0.5 – 0.95] depicts mean average precision for different IOUs ranging from 0.5 – 0.95, varying at a step size of 0.05.

In the object detection domain, it is cardinal to evaluate the loss matrices and their respective values to gauge the accuracy of any technique. Loss matrices of the proposed YOLO algorithm are displayed below in Fig. 15. SSD (loss metrics) and its contingent graphical representations are depicted in Figs. 16, 17

The loss description of Faster R-CNN (Loss Metrics) for defined training, test and validation sets with three types of losses.

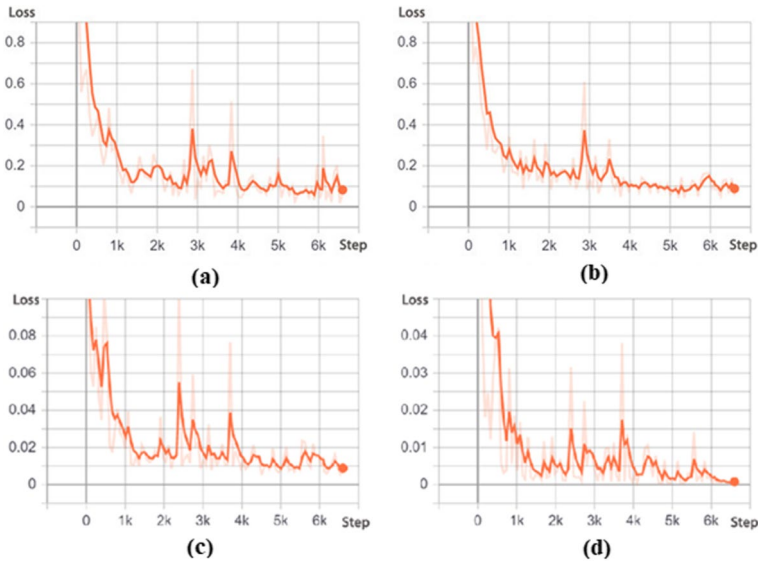
An average recall is also a numerical metric used to compare object detector performance that can be calculated as  $2 \times$  the area under the recall-IOU curve. It is calculated by averaging recall over all IOU ranging from [0.5 – 1.0].

Where  $o$  is IOU and  $\text{recall}(o)$  is the corresponding recall value. Classification Loss is a simple weighted soft-max loss of the 'N' number of classes. It tells how the model performs regarding a

$$AR = 2 \int_{0.5}^1 \text{recall}(o) do$$

classification task for each class. In our case, we need to classify chess pieces. Localization Loss, as the name suggests, is a loss calculated between the actual values and the predicted (correction) values of the coordinates of bounding boxes that specifies how good





**Fig. 17** Elucidating the graphical representation of loss matrices for Faster R-CNN (a) Box Classifier: Classification Loss, (b) Box Classifier: Localization Loss, (c) RPN Classifier: Localization Loss, (d) RPN Classifier: Object Loss

the algorithm is with the localization of objects. In the case of SSD, it is a smooth L1 loss. Object loss in Faster- RCNN We use RPN to generate region proposals. So, we use Object Loss, which tells us how good our model is when classifying whether the bounding box is an object of interest. The same is true for Yolo (but we do not use it for optimizing RPN as we do not have one).

### 4.3 Comparison of evaluation metrics of the proposed algorithms

Table 6 and Figs. 18, 19 below compares the algorithms we have implemented based on standard parameters described above, like Object loss, mAP, Classification loss etc. All of these metrics were analyzed and visualized in Tensor-board.

**Table 6** Comparison of standard evaluation metrics of the algorithms implemented

Metric	Faster R-CNN	YOLO V5	SSD
Localization Loss	Train0.1186	0.02904	0.0517
	Valid0.0925	0.02095	0.2198
Classification Loss	Train0.1037	0.01512	0.1951
	Valid0.1041	$9.1793 \times 10^{-3}$	0.2708
mAP @ 0.5 IOU	0.9908	0.9631	0.9526
Average recall	0.7464	0.9723	0.6693
Object Loss	Train $1.3798 \times 10^{-3}$	0.0429	NA
	Valid $9.3018 \times 10^{-4}$	0.02089	NA

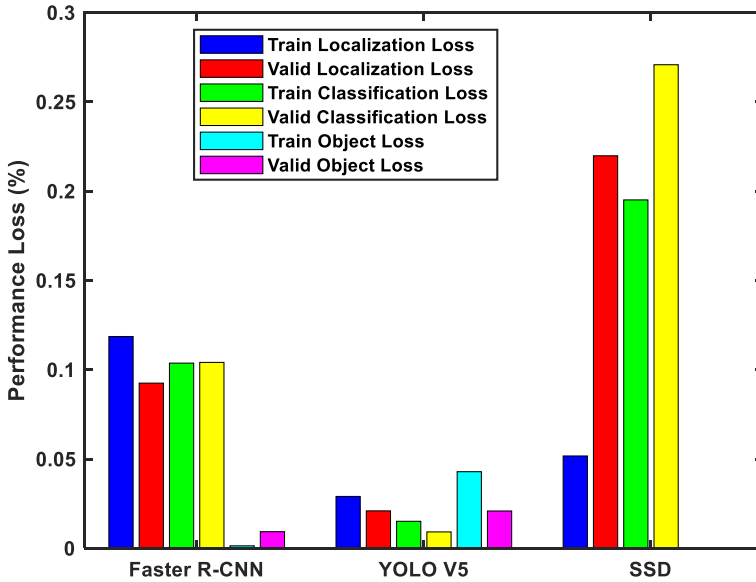


Fig. 18 A graphical representation of comparison of evaluation metrics of the proposed algorithms

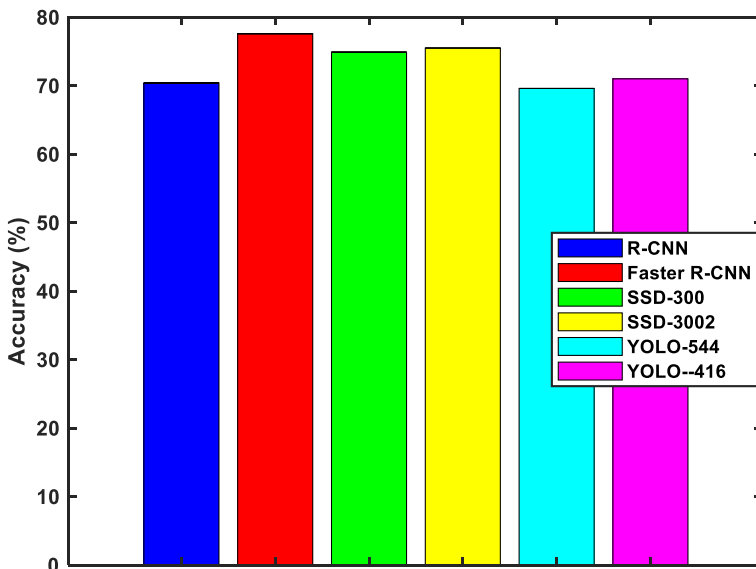


Fig. 19 A graphical representation of accuracy values for faster R-CNN and its fellow techniques

Therefore, presenting the comparative analysis of the proposed technique along with their metric values and numerical accuracy parameters obtained in validation, training and test set. The box localization loss values obtained represent the respective values for faster R-CNN, YOLO 5 and SSD, depicting the parameterized accuracy along with values of Classification Loss, mAP value for 0.5 IOU, average recall, and object loss values for

the proposed three techniques. SSD provides the optimal results with parametric values of 0.0517 for box location, 0.2708 for classification loss, 0.9526 and 0.6693 for mAP and average recall.

Figure-18 shows a graphical representation of accuracy values for faster R-CNNs with existing techniques.

The accuracy comparison of an algorithm is critical to determining its performance, as illustrated above in a graphic comparison between Faster R-CNN and some of its competitors. The above algorithms are illustrated with accuracy values based on predefined parameters.

## 5 Conclusion

This paper compares different types of object detection algorithms, such as Multiple Stage detectors like R-CNN, Faster R-CNN (Faster Region-based CNN), and Single Stage Detectors like Yolo (You Only Look Once), Single Shot Detector (SSD). The mobile device can work these models and detect the objects in a single run. The multi-stage task object detection techniques are performed better than single-stage detectors like Yolo (You Only Look Once) and Single Shot Detector (SSD) regarding accuracy, recall, precision and loss.

Faster R-CNN, Yolo, and SSD algorithms are compared using a Custom Chess Piece Dataset, training them in Google Collaboratory Notebooks. These algorithms are intended to correctly identify chess pieces and draw a bounding box near them using TensorFlow Object Detection API and Roboflow. A confidence interval indicates how confident the algorithm is about predicting their location on the board. The training and evaluation metrics are visualized with Tensorboard, an interactive tool provided by Tensorflow.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions

**Data availability** The data is available on request.

## Declarations

**Conflict of Interest** The authors have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.


## References

1. Fernández A, Salmerón A (2008) BayesChess: A computer chess program based on Bayesian networks. *Pattern Recognit Lett* 29(8) Art. no. 8, 2008
2. Villafaina S, Collado-Mateo D, Cano-Plasencia R, Gusi N, Fuentes JP (2019) Electroencephalographic response of chess players in decision-making processes under time pressure. *Physiol Behav* 198:140–143

3. Kumar A, Srivastava S (2020) Object detection system based on convolution neural networks using single shot multi-box detector. *Procedia Comput Sci* 171:2610–2617
4. Jang Y, Gunes H, Patras I (2019) Registration-free face-ssd: Single shot analysis of smiles, facial attributes, and affect in the wild. *Comput Vis Image Underst* 182:17–29
5. Yi C, Kaneko T (2021) Improving counterfactual regret minimization agents training in card game cheat using ordered abstraction. *Advances in Computer Games*. Springer International Publishing, Cham, pp 3–13
6. Sakai Y, Lu H, Tan J-K, Kim H (2019) Recognition of surrounding environment from electric wheelchair videos based on modified YOLOv2. *Future Gener Comput Syst* 92:157–161
7. Yuan J et al (2020) Gated CNN: Integrating multi-scale feature layers for object detection. *Pattern Recognit* 105:107131
8. Ahmed I, Ahmad M, Ahmad A, Jeon G (2021) IoT-based crowd monitoring system: Using SSD with transfer learning. *Comput Electr Eng* 93:107226
9. Pan H, Jiang J, Chen G (2020) TDFSSD: Top-down feature fusion single shot MultiBox detector. *Signal Process Image Commun* 89:115987
10. Rani P, Verma S, Yadav SP, Rai BK, Naruka MS, Kumar D (2022) Simulation of the Lightweight Blockchain Technique Based on Privacy and Security for Healthcare Data for the Cloud System. *Int J E-Health Med Commun* 13(4):1–15. <https://doi.org/10.4018/IJEHMC.309436>
11. Rani P, Sharma R (2023) Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. *Comput Electr Eng* 105:10854. <https://doi.org/10.1016/j.compeleceng.2022.108543>
12. Wang Q et al (2023) Deep convolutional cross-connected kernel mapping support vector machine based on SelectDropout. *Inf Sci* 626:694–709
13. Ding L, Xu X, Cao Y, Zhai G, Yang F, Qian L (2021) Detection and tracking of infrared small target by jointly using SSD and pipeline filter. *Digit Signal Process* 110:102949
14. Halim Z, Zouq A (2021) On identification of big-five personality traits through choice of images in a real-world setting. *Multimed Tools Appl* 80(24):33377–33408
15. Yundong LI et al (2020) Multi-block SSD based on small object detection for UAV railway scene surveillance. *Chin J Aeronaut* 33(6):1747–1755
16. Bennett S, Lasenby J (2014) ChESS–Quick and robust detection of chess-board figures. *Comput Vis Image Underst* 118:197–210
17. Rani P, Singh PN, Verma S, Ali N, Shukla PK, Alhassan M (2022) An Implementation of Modified Blowfish Technique with Honey Bee Behavior Optimization for Load Balancing in Cloud System Environment. *Wirel Commun Mob Comput* 2022:1–14. <https://doi.org/10.1155/2022/3365392>
18. Li C, Chen G (2020) Research on Chinese Chess Detection and Recognition Based on Convolutional Neural Network. In: *Recent Trends in Intelligent Computing, Communication and Devices: Proceedings of ICCD 2018*, Springer, pp. 467–473
19. Czyzewski MA, Laskowski A, Wasik S (2020) Chessboard and chess piece recognition with the support of neural networks. *Found Comput Decis Sci* 45(4), Art. no. 4
20. Yi J, Wu P, Metaxas DN (2019) ASSD: Attentive single shot multibox detector. *Comput Vis Image Underst* 189:102827
21. Adarsh P, Rathi P, Kumar M (2020) YOLO v3-Tiny: object detection and recognition using one stage improved model. In: *2020 6th international conference on advanced computing and communication systems (ICACCS)*. IEEE, pp 687–694
22. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *Advances in Neural Information Processing Systems*, Curran Associates, Inc.. Accessed: May 09, 2023. [Online]. Available: <https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html>
23. Preeti R, Sharma R (2022) An experimental study of IEEE 802.11 n Devices for Vehicular Networks with Various Propagation Loss Models." *International Conference on Signal Processing and Integrated Networks*. Singapore: Springer Nature Singapore, 2022.
24. Ansari G, Rani P, Kumar V (2023) A novel technique of mixed gas identification based on the group method of data handling (GMDH) on time-dependent MOX gas sensor data. In: Mahapatra RP, Peddoju SK, Roy S, Parwekar P (eds.) *Proceedings of International Conference on Recent Trends in Computing*. Lecture Notes in Networks and Systems, vol. 600. Springer Nature Singapore, Singapore, pp. 641–654. [https://doi.org/10.1007/978-981-19-8825-7\\_55](https://doi.org/10.1007/978-981-19-8825-7_55)
25. Li X, Li Y, Shen C, Dick A, Hengel AVD(2013) Contextual hypergraph modeling for salient object detection. In: *2013 IEEE International Conference on Computer Vision*, Sydney, pp. 3328–3335. <https://doi.org/10.1109/ICCV.2013.413>

26. Cheng M-M, Mitra NJ, Huang X, Torr PH, Hu S-M (2014) Global contrast based salient region detection. *IEEE Trans Pattern Anal Mach Intell* 37(3), Art. no. 3
27. Jiang H, Wang J, Yuan Z, Wu Y, Zheng N, Li S (2013) Salient object detection: a discriminative regional feature integration approach. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, pp 2083–2090. <https://doi.org/10.1109/CVPR.2013.271>
28. Hou Q, Cheng M-M, Hu X, Borji A, Tu Z, Torr PH (2017) Deeply supervised salient object detection with short connections. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3203–3212. <https://doi.org/10.48550/arXiv.1611.04849>
29. Wang X, Shrivastava A, Gupta A (2017) A-Fast-RCNN: hard positive generation via adversary for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, pp 3039–3048. <https://doi.org/10.1109/CVPR.2017.324>
30. Pathak AR, Pandey M, Rautaray S (2018) Application of deep learning for object detection. *Procedia Computer Science* 132:1706–1717. <https://doi.org/10.1016/j.procs.2018.05.144>
31. Kumar A, Singh N, Kumar P, Vijayvergia A, Kumar K (2017) A novel superpixel based color spatial feature for salient object detection. In: 2017 Conference on Information and Communication Technology (CICT). IEEE, Gwalior, pp. 1–5. <https://doi.org/10.1109/INFOCOMTECH.2017.8340630>
32. Kumar K, Shrimankar DD, Singh N (2018) Eratosthenes sieve based key-frame extraction technique for event summarization in videos. *Multimed Tools Appl* 77(6):7383–7404. <https://doi.org/10.1007/s11042-017-4642-9>
33. Kumar K, Shrimankar DD (2018) F-DES: Fast and Deep Event Summarization. *IEEE Trans Multimed* 20(2):323–334. <https://doi.org/10.1109/TMM.2017.2741423>
34. Sharma S, Kumar K, Singh N (2017) D-FES: Deep facial expression recognition system. In: 2017 Conference on Information and Communication Technology (CICT). IEEE, Gwalior, India, pp. 1–6. <https://doi.org/10.1109/INFOCOMTECH.2017.8340635>
35. Sharma S, Kumar K, Singh N (2022) Deep Eigen Space Based ASL Recognition System. *IETE J Res* 68(5):3798–3808. <https://doi.org/10.1080/03772063.2020.1780164>
36. Kumar K, Shrimankar DD, Singh N (2019) Key-Lectures: Keyframes Extraction in Video Lectures. In: Tanveer M, Pachori RB (eds.) *Machine Intelligence and Signal Analysis*. *Advances in Intelligent Systems and Computing*, vol. 748. Springer Singapore, Singapore, pp. 453–459. [https://doi.org/10.1007/978-981-13-0923-6\\_39](https://doi.org/10.1007/978-981-13-0923-6_39)
37. Kumar K (2021) Text query based summarized event searching interface system using deep learning over cloud. *Multimed Tools Appl* 80(7):11079–11094. <https://doi.org/10.1007/s11042-020-10157-4>
38. Sharma S, Kumar K (2021) ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimed Tools Appl* 80(17):26319–26331. <https://doi.org/10.1007/s11042-021-10768-5>
39. Abhay A, et al (2017) An automated hierarchical framework for player recognition in sports image. *Proceedings of the international conference on video and image processing*. <https://doi.org/10.1145/3177404.3177432>
40. Koppanati RK, Kumar K (2021) P-MEC: Polynomial Congruence-Based Multimedia Encryption Technique Over Cloud. *IEEE Consum Electron Mag* 10(5):41–46. <https://doi.org/10.1109/MCE.2020.3003127>
41. Kumar K, Kumar A, Bahuguna A (2017) D-CAD: Deep and crowded anomaly detection. *Proceedings of the 7th international conference on computer and communication technology*. <https://doi.org/10.1145/3154979.3154998>
42. Hu J, Shi C-JR, Zhang J (2021) Saliency-based YOLO for single target detection. *Knowl Inf Syst* 63(3):717–732. <https://doi.org/10.1007/s10115-020-01538-0>
43. Srivastava G, Srivastava R (2020) User-interactive salient object detection using YOLOv2, lazy snapping, and gabor filters. *Mach Vis Appl* 31(3):17. <https://doi.org/10.1007/s00138-020-01065-6>
44. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018). <https://doi.org/10.48550/ARXIV.1804.02767>
45. Cai Y et al (2021) YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving. *IEEE Trans Instrum Meas* 70:1–13. <https://doi.org/10.1109/TIM.2021.3065438>
46. Agyemang IO, et al. (2021) On salient concrete crack detection via improved Yolov5. In: 2021 18th International computer conference on wavelet active media technology and information processing (ICCWAMTIP). IEEE. <https://doi.org/10.1109/ICCWAMTIP53232.2021.9674153>

## Authors and Affiliations

Satya Prakash Yadav<sup>1,2</sup> · Muskan Jindal<sup>3</sup> · Preeti Rani<sup>4</sup> ·  
Victor Hugo C. de Albuquerque<sup>5</sup> · Caio dos Santos Nascimento<sup>5</sup> · Manoj Kumar<sup>6,7</sup> 

✉ Manoj Kumar  
wss.manojkumar@gmail.com

Satya Prakash Yadav  
prakashyadav.satya@gmail.com

Muskan Jindal  
muskanjindal2790@gmail.com

Preeti Rani  
preetiresearcher1@gmail.com

Victor Hugo C. de Albuquerque  
victor.albuquerque@ieee.org

Caio dos Santos Nascimento  
caio.santos@alu.ufc.br

<sup>1</sup> Department of Computer Science and Engineering, G.L. Bajaj Institute of Technology and Management (GLBITM), Greater Noida 201306, India

<sup>2</sup> Graduate Program in Telecommunications Engineering. (PPGET), Federal Institute of Education, Science, and Technology of Ceará (IFCE), Fortaleza, CE, Brazil

<sup>3</sup> Department of Computer Science and Engineering, Amity University, Noida 201313, India

<sup>4</sup> Department of Electronics & Communication Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, Delhi-Meerut Road, Modinagar, Ghaziabad, Uttar Pradesh, India

<sup>5</sup> Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, CE, Brazil

<sup>6</sup> School of Computer Science, FEIS, University of Wollongong in Dubai, Dubai Knowledge Park, Dubai, UAE

<sup>7</sup> MEU Research Unit, Middle East University, Amman 11831, Jordan