Received: 22 April 2023 | *Revised: 4 June 2023* | *Accepted: 28 June 2023*
DOI: 10.37917/ijeee.20.1.15 *Vol. 20 | Issue 1 | June 2024*

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

IJEEE University of Basrah College of Engineering

Multiple Object Detection-Based Machine Learning Techniques

Athraa S. Hasan[1,2], Jianjun Yi [2*], Haider M. AlSabbagh[3], Liwei Chen [1]
[1]Department of Mechanical Engineering, East China University of Science and Technology ECUST-Shanghai 200237, China
[2]Department of Electromechanical, University of Technology, Baghdad, Iraq
[3]Department of Electrical Engineering, College of Engineering, University of Basrah, Basrah, Iraq

Correspondance
*Jianjun Yi
Department of Mechanical Engineering,
East China University of Science and Technology ECUST-Shanghai 200237, China
Email: jjyi@ecust.edu.cn

Abstract
Object detection has become faster and more precise due to improved computer vision systems. Many successful object detections have dramatically improved owing to the introduction of machine learning methods. This study incorporated cutting- edge methods for object detection to obtain high-quality results in a competitive timeframe comparable to human perception. Object-detecting systems often face poor performance issues. Therefore, this study proposed a comprehensive method to resolve the problem faced by the object detection method using six distinct machine learning approaches: stochastic gradient descent, logistic regression, random forest, decision trees, k-nearest neighbor, and naive Bayes. The system was trained using Common Objects in Context (COCO), the most challenging publicly available dataset. Notably, a yearly object detection challenge is held using COCO. The resulting technology is quick and precise, making it ideal for applications requiring an object detection accuracy of 97%.

Keywords
Object Detections, Computer Vision, Machine Learning, K-Nearest Neighbors, Naïve Bayes, Stochastic Gradient Descent, Decision Tree, Linear Regression, Random Forest Learning.

I. Introduction

Naked-eye object detection is simpler because humans can easily detect various parameters of objects, such as orientation, color, texture, and opacity. A computer requires significant time to recognize and identify objects in an image. In computer vision, "object detection" refers to searching and detecting an object in an image or a video. Object detection entails three main processes: feature extraction, feature processing, and object classification [1]. Conventional object detection methods can be divided into four categories: bottom feature extraction, feature coding, feature aggregation, and classification. Most of these methods have achieved good results. Feature extraction is crucial for identifying and classifying objects. Point-of- interest detection is expected to outperform its predecessors owing to increasing potentially useful redundant data [2, 3]. Machine learning (ML) is a data analysis approach that can automate analytical model development. It is a subdivision of artificial intelligence (AI) because of its foundational idea: a computer program can automatically learn new information, see patterns, and make decisions with minimal human guidance [4]. ML algorithms can be divided into two major categories: unsupervised and supervised learning. After teaching a classifier using a training set, efficacy of the classifier can be measured using the testing set. However, unsupervised learning involves exploring potential connections between different entities. The documents can be categorized using supervised learning algorithms, which can be used to train a classifier on a given collection of documents using a ML algorithm. Subsequently, the trained classifier is used to categorize documents of a testing set [5].

Programs incorporating ML can use training data and past experience to improve some performance metrics. A learning algorithm modifies the model parameters by analyzing the training data or previous experience. Defining a model involves assigning parameters, and learning involves using them to increase the model accuracy. The model can be either descriptive (used to learn from the past) or predictive (used to predict the future). Statistical theory is used to develop mathematical models because making inferences from samples is central to ML [6]. First, training requires efficient algorithms for storing and processing large amounts of data and solving the optimization problem. Second, after a model is trained, its representation and algorithmic solution must be effective for making optimal inferences. The predictive accuracy and algorithm efficiency (regarding space and time complexity) may be equally significant in some scenarios [7].
Classification is categorized as a supervised learning technique in ML owing to its dual nature: a predictive modeling problem and a class label for an input sample. Mathematically, a classification task transforms the input variables (X) into output variables $(Y, Z, L,\ and\ C)$. It can anticipate the type of data bits given by either structured or unstructured data. Various classification methods have been proposed for ML and data science. This study used the following six ML methods considering an integrative approach to resolving object detection problems: stochastic gradient descent (SGD), naïve Bayes (NB), decision trees (DT), random forest (RF), k-nearest neighbor (KNN), and logistic regression (LR). The outline of this paper is as follows: Related works are discussed in Section II. Section III presents the blueprint for the proposed procedure in its basic form. Section IV outlines the requirements for the experimental work. The Section V discusses the performance evaluation of the classification algorithm. Section VI discusses the results and analysis. Finally, Section VII concludes the study.

II. Helpful Literature Review

Several recent studies have focused on resolving the problem of object detection. In-depth features can be quickly identified and investigated using modern and sophisticated learning technologies. This study compiled data on the numerous object identification methods and algorithms employed by various researchers to thoroughly compare and derive actionable results. Rasheed et al. [7] advocated employing a wide support vector machine (SVM) to combine data from various resolutions, which increased the representational quality of urban objects. An SVM network may enhance performance; however, it can incur a heavy computational burden, and simplify the processes without compromising the effectiveness. Feature maps from a convolutional neural network can be used with an SVM network. Ren et al. [8] provided a region proposal network (RPN) that allows approximately free region proposals by sharing full-image convolutional features with the detection network. An RPN is a fully convolutional network that simultaneously predicts both the boundedness and object lessness of an object at each place. End-to-end training of the RPN produces high-quality region recommendations that Faster Region-based Convolutional Neural Networks (Faster R-CNN) can employ to perform detections. Burde and Budihal [3] proposed a structure to create learning-based algorithms for object detection and tracking. Furthermore, they analyzed video- and image-based security and surveillance systems. The You Only Look Once (YOLO) V3 model was trained to detect objects on the Common Objects in Context (COCO) dataset. Items like humans, bottles, drilling machines, and air- powered saws have been detected. The model identifies objects within the field of view of the camera (i.e., in plain sight). In the preprocessed form, the suggested model provides an average accuracy of approximately 96% across various test cases. Srivastava et al. [9] proposed following three key image processing algorithms that are the quickest and most effective: Single Shot Detection, Faster R-CNN, and YOLO. Using the Microsoft COCO dataset, the similarities and differences of the three algorithms were studied regarding various metrics, such as accuracy, precision, and F1-score. Wenze et al. [10] evaluated Faster R-CNN in depth and analyzed its efficacy using various pre- training models. Three datasets were used to determine the times and accuracies of R-CNN, fast R-CNN, and faster R- detection CNN. Furthermore, Zhang et al. [11] proposed a learning-to- match (LTM) approach to avoid the limitation on object- anchor matching. LTM uses Maximum Likelihood Estimation (MLE) detector training to transit from manual anchor assignment to ”free” anchor matching. LTM transforms the detection likelihood into easily implementable anchor-matching loss function during training. Learning and feature selection for object classification and localization are performed by minimizing the matching loss functions. The generalizability of a learnable object-feature matching mechanism for visual object detection was verified by extending the LTM from anchor-based to anchor-free detectors. LTM detectors significantly outperform alternative detectors in experiments involving the MS COCO dataset. Moreover, LTM does not require additional architecture or parameters, making training and inference computationally inexpensive. Li et al. [12] investigated human-object interactivity or interactiveness. Knowledge of interactiveness can be acquired across Human-Object Interaction (HOI) datasets and used to connect otherwise dissimilar HOI category settings. Before making inferences from HOI classification, we used an Interactiveness Network to learn generic interactiveness from different HOI datasets. Subsequently, we suppressed non-interactions using a recurrent neural network. The generalizability makes the Interactiveness Network a knowledge learner, which can be used along with any HOI detection model. We can interact with hierarchical paradigms by studying human instances and body features. A consistent task can guide the learning to obtain more in- depth interactive visual clues. We put the method through its pace on HICO-DET, V-COCO, and HAKE-HOI. The superiority of our trained interactivity was verified by comparing it with the best HOI detection methods.

The inference process for feature-pyramid-based object detectors is sped up by the novel query mechanism proposed by Yang et al [13]. The pipeline makes approximate predictions regarding the locations of small objects based on low-resolution features. After that, it uses high-resolution features sparsely guided by approximate predictions to compute accurate detection results. Therefore, detailed feature maps can be leveraged while avoiding unnecessary background computations. On the COCO dataset, the proposed method increased the detection mAP, mAPsmall, and high-resolution inference speed by$1.0, 2.0,$ *and* 3.0,respectively. We achieved a new state-of-the-art high- resolution acceleration on the COCO dataset (2.3) times that of the VisDrone dataset) because the number of small objects was higher in the VisDrone dataset. Li et al. [14] proposed a full-stack approach to object detection using DT and deep neural networks. In neural networks, soft DT decouples decision options from the prediction values. We propose randomized decision routing with node-selective and associative losses to enhance feature-representative learning and network decision-making. After that, we designed a decision head for object detection with narrow branches to generate routing probabilities and masks for divergent node decisions. This strategy can be described by randomized decision routing for object detection (RDR) (Det 2). Detection using the MS-COCO dataset was enhanced by R (Det) 2. The current generation of detectors boosts AP performance by 1.4–3.6$\%$. Saito et al. [15] proposed Learning to Detect Everything (LDET), a straightforward data augmentation and training scheme with surprisingly potent results. Our new data augmentation method, called BackErase, places annotated objects on top of a background image taken from a small region of the original image to prevent suppressing unseen objects. We propose a multi domain training approach to aid the transfer of the model to real-world images. LDET performed better than baseline on the UVO, Objects365, Cityscapes cross-dataset evaluations, and the COCO cross-category generalization task.

III. Methodology

This section of the text is dedicated to discussing in detail the proposed model presented in Fig. 1. The model comprises various image-processing procedures that are followed by extraction-based principal component analysis for feature extraction. Additionally, six machine learning algorithms are discussed within the context of this proposed model. The initial stage of this model involves a set of image-processing procedures aimed at enhancing and optimizing input images for subsequent analysis. This crucial step primarily consists of pre-processing tasks such as noise reduction, contrast adjustment, and image normalization to enhance the overall quality and optimize them further for ML algorithms and feature extractions. The proposed model employs the extraction-based principal component analysis to extract relevant features from pre-processed images. PCA is a statistical technique that transforms high-dimensional datasets into lower dimensions while maintaining critical information. It identifies principal components by projecting image data onto an orthogonal coordinate system and captures significant variations in the dataset. The extracted features are represented by these principal components, embodying essential image characteristics. The model includes six distinct machine-learning algorithms that are designed to analyze the extracted features and detect various objects. The selection of these algorithms is based on their unique capabilities in dealing with specific characteristics and demands inherent in the problem domain. This article explores each algorithm's properties, advantages, and limitations within the context of this proposed model thoroughly. Additionally, detailed illustrations demonstrate how every method optimizes object detection performance for improved overall system functionality.

A. Image Processing:

Every image must undergo five stages before sending it to the model. The operational process of the five stages are:

1) Data Set:

The COCO dataset [16] used in this study contained 526394 pictures. The 92 classes were divided between the training and testing phases with a ratio of 70:30. Hence, the total number of photos for testing and training were 157,918.2 and 368,475.8, respectively. Fig. 2 Displays the original images

2) Convert RGB Images to Grayscale Images

Grayscale is the most elementary system because it uses luminance to characterize colors. A number between 0 (black) and 255 (white) defines brightness. However, the information content of grayscale photographs is lower than that of their RGB counterparts. Although uncommon in other fields, grayscale images are widely used in image processing because of their space and processing time benefits. By taking a weighted average of the features from different parts, the contribution of green features will increase while that of blue features will decrease in the final value. Equation (1) [17] was derived after several experiments and additional in-depth studies. Fig. 3

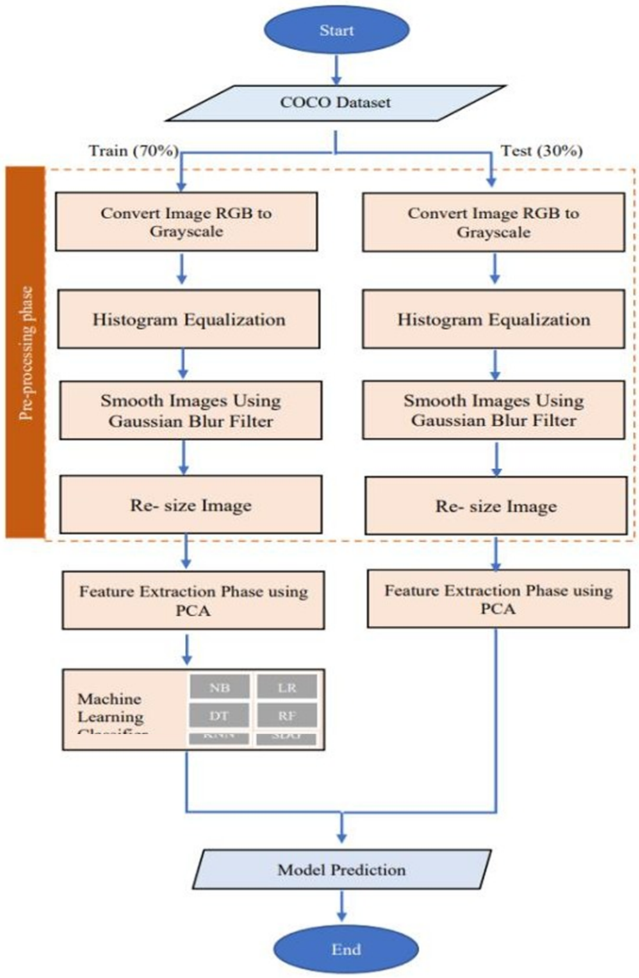

Fig. 1. Block Diagram of Proposed Work

Displays the resulting photos produced by this process.

$$New\ grayscale\ image = ((0.3*R)+(0.59*G)+(0.11*B)) \quad (1)$$

3) *Histogram Equalization Method:*

Histogram equalization is a well-known method to improve the appearance of images. The function is analogous to histogram stretch. However, it typically yields more aesthetically acceptable outcomes for various photos. The histogram of an image can be made as flat as required using histogram equalization. However, histogram stretching would retain the general shape of a histogram. Spreading or flattening the histogram causes the dark pixels to look darker and the light pixels to look lighter (the operative word being "appear"); nonetheless, the black pixels in a photograph cannot appear darker than the original. However, the black pixels may appear darker if the slightly lighter pixels become substantially lighter [18, 19]. The histogram equalization procedure for digital images involves four steps:

Fig. 2. Original Images Dataset

Fig. 3. Transform Images from RGB to Grayscale

Step 1: Calculate the cumulative histogram values.

Step 2: Divide the values from Step 1 by the total number of pixels to normalize them.

Step 3: Multiply the numbers obtained in Step 2 by the highest possible gray level and round off the result to the nearest integer.

Step 4: Perform a one-to-one mapping of grayscale values to the output of the Step 3 correspondence.

Histogram equalization was calculated for all photos in the dataset. The histogram was constructed using equation (2).

$$h[i] = \sum_{x=1}^{N} \sum_{y=1}^{M} \begin{Bmatrix} 0 & f[x.y] = i \\ 1 & otherwise \end{Bmatrix} \quad (2)$$

Where $h[i]$ denotes the resulting value of the histogram equalization ; M and N correspond to the dimension of the grayscale image [20]. Fig. 4 Shows image transformations caused by the histogram

4) *Gaussian-blur*

Gaussian blur is used to improve multiscale image architectures. Applying a Gaussian blur with a filter size of $(5*5)$ to a picture is mathematically equivalent to convolving the image with a Gaussian function applied to each pixel. The one-dimensional Gaussian function is given by:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

Fig. 4. The Effect of Histogram on the Grayscale Images.

Fig. 5. Gaussian Distribution

Where x corresponds to the horizontal distance from the center and is the standard deviation of the Gaussian distribution. The effect of (3) can be seen in Fig. 5.

5) *Re-size Images*

The bilinear interpolation process is often used to enlarge or reduce image size. An interpolated value is produced using a weighted average of the values of four adjacent pixels, resulting in a smoother image with fewer jagged edges than the original image. This method yields superior results compared to nearest-neighbor interpolation [21]. An intermediate value can be calculated using linear interpolation. Methods for determining the value of a function defined by a linear equation using the law of proportionality are incorporated in the field of linear interpolation. The interpolar bilinear approach demonstrates that interpolar outcomes are attained when three values converge from top to bottom, right to left, and left to right [22]. Pixels x_0, x_1, y_0, and y_1 are shown in the bilinear interpolar as:

$$y = y_0\left(1 - \frac{1 - x_0}{x_1 - x_0}\right) + y_1\left(1 - \frac{x_1 - x}{x_1 - x_0}\right) \tag{4}$$

Bilinear interpolation provides a robust means of adjusting image size while maintaining visual fidelity. This method utilizes linear interpolation techniques and the weighted average of neighboring pixels to produce smoother transitions and minimized jagged edges compared with alternative forms of interpolation. Additionally, this approach is useful across various areas such as thermal imaging cameras, where it assists in generating color palettes based on interpolar bilinear coefficients. Numerous studies have highlighted the efficacy and wide applicability of this technique for accurate image.

In this case, the values of the pixels involved in developing interpolar objects are adjusted accordingly. The value between pixels x_0 and x_1 is considered the interpolated value of y, and similarly for pixels y_0 and y_1. Notably, the y values

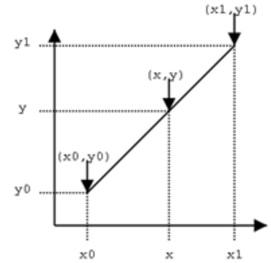

Fig. 6. The 4-Dimensional Surrounding Area of a Point 'p' In a 2-Dimensional Picture Space.

cannot exceed the x_0, x_1, y_0, and y_1 values. Fig. (6) shows that the interpolar bilinear coefficient was calculated based on the horizontal and vertical distance between nearby pixels. The result showed that the image comprised white and black color patterns separated by grayscale transitions [23].

B. Features Extraction: Principal Component Analysis

PCA is a typical statistical method that discovers patterns in high-dimensional data through a holistic approach. The objective of PCA is based on an information theory. PCA involves breaking images down into small sets of distinguishing feature images known as eigen objects, which are then used to represent both new and existing objects. The statistical data recently published in object recognition technology shows the significance of applying the PCA approach to identify and validate object features. The two- dimensional object image matrices must be translated into a one-dimensional vector before incorporating the PCA method. A one-dimensional vector can be either a row vector or a column vector. In phase one, PCA feature extraction for object images is illustrated in Algorithm (1) [24, 25].

Estimators of the mean and covariance matrices based on the maximum likelihood method were calculated for both the positive and negative datasets. Furthermore $(\overrightarrow{\mu_{pos}} - \Sigma_{pos} - \overrightarrow{\mu_{neg}} - \Sigma_{neg})$ can be computed by mapping the training data into the constrained domain. Subsequently, the Gaussian classifier can be used to label each sub-image [26] (see (5)):

$$\begin{aligned} \log(P(c_i|\overrightarrow{x})) = &-\frac{1}{2}(\overrightarrow{x} - \overrightarrow{\mu_i})\Sigma_i(\overrightarrow{x} - \overrightarrow{\mu_i}) \\ &-\frac{1}{2}(\log(|\Sigma_i|) + k \end{aligned} \tag{5}$$

where c_i denotes class i, $\overrightarrow{x} - \overrightarrow{\mu_i}$ corresponds to the mean image of class i, Σ_i denotes the covariance matrix of class i ,and $\overrightarrow{x}$ represents the classified new image segment. The constant k is class-independent. Hence, it is identical in all classes c_i . New images were assigned to the class for which their logarithmic likelihood.

$\log(p(c_{pos}|\overrightarrow{x})), p(c_{neg}|\overrightarrow{x}))$ was largest-either the positive or the negative class.

IV. Machine Learning (ML) Algorithms

ML technique that may be used to automate analytic model development. The idea that computers can learn independently from data, identify patterns, and make decisions with minimal human intervention has led to their classifying ML as a subset of artificial intelligence [7]. This corresponds to the features (input value) [27]. This study used six algorithms—NB, SGD, LR, KNN, DT, and RF. The following sections provide details of these algorithms:

1) Naive Bayes

It is an algorithm for learning under supervision. The data frequency and collection sizes are used to derive a set of probabilities. The posterior probability of document (d) belonging to class (c) is derived based on the NL method as [7, 28].

$$P(c|d) = P(d|c)P(c)P(d) \tag{6}$$

$$P(c|d)P(w_1, w_2, w_3, ..., w_n|c)P(c)P(d) \tag{7}$$

where $P(d|c)$ represents the likelihood, which is the probability of the predictor given class. $P(w_i|c)$ Represents the qualified probability of term w_i in document d of class c . $P(w_i|c)$ Corresponds to measuring how much w_i contributes, and c denotes the accurate class. Furthermore, $(w_1, w_2, w_3, ..., w_n)$ represents the tokens in document d and are part of the vocabulary used for classification; n is the number of tokens in document d.

2) Stochastic Gradient Descent

This model is a powerful learning algorithm for linear classifiers. Replacing the true gradient with an estimate based on a smaller portion of the information is sufficient. A stochastic (or "operational") gradient descent algorithm provides a gradient to each learning element. Some alterations were made to the parameters considering the projected gradients. The model parameters were re-estimated after introducing a new learning object. The SGD method for large datasets is noticeably faster than the traditional method [29]. SGD is an effective form of facilitation. Easily digestible SGD is updated as shown in equation (8):

$$\theta^{(t+1)} = \theta^t - \alpha_t \nabla l_i(\theta^t) \tag{8}$$

where t represents the iteration and $l_¸$ denotes the parameter updates. The value of index i was picked randomly before each iteration. [30].

3) Logistic Regression

LR belongs to a subset of log-linear or exponential classifiers. NB favors LR for most parts. A log-linear classifier was used to extract a set of weighted highlights from the data input, and then the logs were taken and joined linearly. The *LR* is as follows:

$$\log_y = \frac{1}{1 + e^{-(b_0 + b_1.x_1 + b_2.x_2)}} \tag{9}$$

where b_0 , b_1, and b_2 are the coefficients; x_1 and x_2 correspond to the features (input value) [27].

4) K-Nearest Neighbor

The KNN algorithmic pattern is a mild and simple supervised learning technique. It is as easy as settling on a measurement for the gap between two known samples. KNN assigns a label to an input based on its similarity to k-nearest training set prototypes [30]. Equation (10) represents the standard human conceptualization of distance in the physical world. It is used to represent the Euclidean distance [31].

$$D_{euclidean}(x, y) = \sqrt{\sum_{i=0}^{m} (x_i - y_i)^2} \tag{10}$$

where m corresponds to the total number of distinct words in the set of documents, x_i represents the term i in document x, and yi denotes its importance in document y.

5) Decision Tree

Models in the shape of a tree can be efficiently generated using the conventional DT technique. Using a DT, a dataset was divided into narrower categories for analysis. An accompanying DT was also constructed in parallel. The output of DT is a tree containing decision and leaf nodes. Both quantitative and categorized information can be processed using DT [32]. The rule for the DT is as follows:

$$Entropy(t) = -\sum_{i=0} p(i|t) \log_2 p(i|t)c - 1 \tag{11}$$

where c corresponds to the number of classes, $p(i|t)$ denotes the fraction of records that belong to class i at a given node t.

6) Random Forest

RFs are used in the learning process known as the ensemble tree-based technique. An assortment of DT, each selected randomly from the training data, constitutes the RF classifier. RF considers the votes from all distinct DT when the ultimate class of the test object is determined. RF algorithm has several benefits (1) it is useful for various classification and regression problems, (2) it can handle missing values and maintain accuracy for missing data, and (3) it can work with a large dataset of a higher dimensionality [33].

V. Experimental Work

Researchers tested the suggested methods for object detection and compared the results to determine the effectiveness of the methods. The data were split into 70% training and 30% test sets. Managing object detection in the current digital world is a serious obstacle. Multiple classification strategies have been used to address the issue of object detection systems. Object detection involves a multistep process. Each dataset was used to train a model and evaluate a classification classifier. The ML classifier is the basis of the intended model, which has been trained to work with various techniques and massive datasets. Furthermore, a model written in Python with the help of the ML library was developed.

This study aimed to calculate the classification accuracy of the model. The accuracy of the classifier was tested by determining the effectiveness of correspondence between expected class labels and the actual class labels. Calculating the number of class examples correctly recognized (true positives), the number of well-recognized examples not class-relevant (true negatives), and examples either incorrectly classified (false positives) or not classified (false negatives) are all ways to evaluate the accuracy of classification. The following are the measures taken for quantitative analysis [34, 35]:

A. Precision:

It represents the number of true positives that can be distinguished from the total number of true and false positives. False positives are instances correctly labeled by the model as positive but negative, as illustrated in equation (12). TP and FP stand for true positives and false positives, respectively

$$Pricision = \frac{TP}{TP+FP} \tag{12}$$

B. Recall:

The accuracy of the model depends on the number of data pieces that are claimed to be significant and relevant by the model. In this case, TP and FN refer to true positives and false negatives, respectively

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

C. F1-score:

The F_1 metric assesses swiftness with which precision and recall are synchronized:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{2TP}{2TP+FP+FN} \tag{14}$$

If F_1 is high, the system is generally running smoothly.

VI. Results And Discussion

This section illustrates the results achieved while applying the model and the other algorithms discussed in the previous sections. A better result with a more exact output was obtained when elements, such as feature extraction, feature processing, and object classification, were included. Additionally, the DT and RF methods are the most accurate of all the algorithms employed. Researchers obtained a precision of 98% using DT and RF techniques with additional attributes, as described in Table (1) and Fig. 7. Precision computation benefits the most from the terms that appear most frequently in the dataset collection, which can be determined using the DT approach. Figures 7 and 9 demonstrate a more in-depth comparison. The NB algorithm demonstrated satisfactory performance with a precision score of 0.58, implying that 58% of the detected positive objects were true positives. A recall score of 0.52 indicated that only 52% of actual positive objects were correctly identified by the algorithm. The F1-score was computed to be 0.48, signifying moderate effectiveness in object detection but room for improvement exists. In comparison to NB's results, SGD yielded lower scores: a precision value of merely 0.44 suggested poor identification rates as only 44% of classified positives indeed turned out to be true positives. These measures translate into an overall substandard F1-score performance level close to just around "21%".

The findings imply that the SGD algorithm might necessitate further fine-tuning or is unsuitable for this designated task. In comparison to other assessed algorithms, LR exhibited lackluster performance with a meager precision score of 0.21, indicating only 21 % of positive object classifications were genuine positives. Moreover, an unexceptional recall score of 0.14 shows the algorithm could accurately identify merely 14 % of actual positive objects while its F1-score at a dismal value of 0.15 confirms substandard results in object detection by LR's implementation herein and suggests alternative approaches may be more effective than using it for this specific task.

The KNN algorithm outperformed SGD and LR with a precision score of 0.51, indicating that it accurately classified 51% of positive objects. A recall score of 0.23 suggests successful detection of only 23% actual positives resulting in an average F1-score at 0.27; however, the DT algorithm demonstrated high scores on all evaluation metrics: precise (0.97), Recall (0.97) and F1-score(0..97), which indicated accurate detection rate above 95%. Although some improvement was observed with the implementation of KNN over SGD or LR, there is still room for further enhancement to achieve better performance results similar to those obtained from using the decision tree classifier. The DT algorithm has high accuracy in detecting positive objects with few false positives or negatives.

Additionally, the RF algorithm showed similarly high scores

Algorithm (1): Principal Component Analysis

Input: Input image

Output: Feature Vectors

BEGIN

Step1: Read images.

Step2: Make a training set of total M images to use the in computing the Average Mean as shown in the equation .

$$Avarage = \frac{1}{M}\sum_{n=1}^{M} Training\ images\ (n)$$

Step3: Subtract the Original image from the Average Mean as shown in the equation:

Sub= Training images - Average

Step4: Calculate the Covariance Matrix as shown in the equation :

$$Covariance = \sum_{n=1}^{M} sub(n)\ sub^{T}(n)$$

Where M: is the Training set of total images

μ: represent the average Mean

Sub: represent the subtracted image from the average μ

Step5: Calculate the Eigenobject of the Covariance Matrix.

Step6: Sort and choose the best Eigenobject. The highest Eigenobject that belong to a group of Eigenvectors are chosen; these M Eigenvectors describe the Eigenobject. Given that new faces are encountered, the Eigenobject can be updated or recalculated accordingly.

Step7: Project the training samples onto Eigenobject and attain feature space.

END

across all evaluation metrics, achieving a precision, recall and F1-score of 0.97 - evidencing an ability to detect true positives at a rate of 97%. These results demonstrate that both algorithms perform well for object detection tasks with comparable levels of effectiveness.

The ML technique performed reasonably well, considering its precision, recall, and F1-score while constructing the model. The outcomes of the proposed methodologies are encouraging. However, a more in-depth analysis reveals some interesting and significant statistics that may be used to make better decisions while choosing correct ML algorithms and test the effectiveness of such choices. Insightful analysis and helpful details are presented below:

(1) Compared to other ML techniques, the proposed model was found to have the best balance of accuracy, precision, recall, and F1-score to construct the model metrics

(2) The DT and RF algorithms performed the best among a pool of six ML algorithms: Bayesian modeling techniques, SGD, adaptive boosting, DT, LR, and KNN. We note that using ML methods, the chosen ML performance is quite effective for object detection in computer vision

(3) This study focused on just three metrics—precision, recall, and F1-score—to assess the effectiveness of ML algorithms. However, the accuracy and time required to create the model are crucial. All employed ML techniques performed well in terms of the chosen metrics.

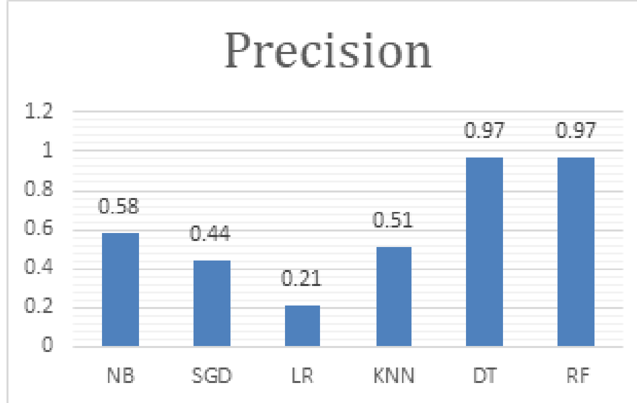

Fig. 7. Performance Evaluation of All Models Based on Classifier Precision Measure

TABLE I.
PERFORMANCE EVALUATION OF DIFFERENT CLASSIFIERS

Algorithms		Measures		
		Precision	Recall	F1-score
Proposed	NB	0.58	0.52	0.48
	SGD	0.44	0.22	0.21
	LR	0.21	0.14	0.15
	KNN	0.51	0.23	0.27
	DT	0.97	0.97	0.97
	RF	0.97	0.97	0.97
Srivastava et al. [15]	SSD	0.424	0.577	NA
	YOLO	0.568	0.623	NA
	FRCNN	0.873	0.893	NA

(4) Using a soft set proved successful with all the given ML measures. However, applying extra parameters and ML methods is still beneficial.

(5) This study used the soft-set strategy, which is used in selection and decision-making. Researchers have shown that DT and RF algorithms are successful when a soft set is employed. Despite this, this strategy must be employed to face various challenges related to object detection in computer vision. ML is a challenging algorithmic field. It has diversified uses within the field of object detection. The ratios of the performance compression for all the algorithms are shown in Fig. 11.

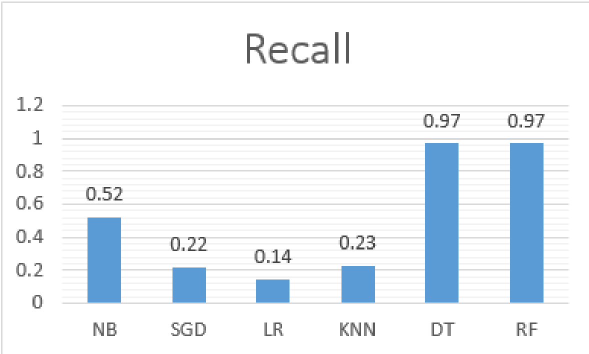

Fig. 8. Performance Evaluation of All Models Based on Classifier Recall Measure

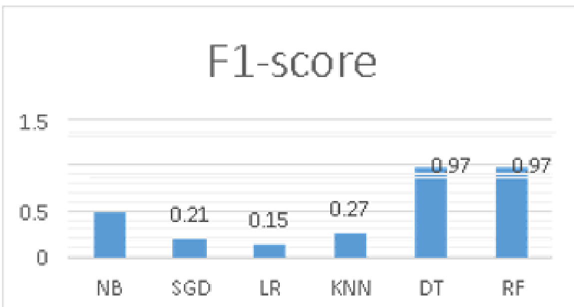

Fig. 9. Performance Evaluation of All Models Based on Classifiers Precision, Recall and F1-score Measure

VII. Conclusion

Data mining methods are essential for the efficient categorization of large datasets. This study covers six distinct ML techniques for object detection. Results of comparing DT and RF showed that both of them may be equivalent. However, the NB algorithm outperformed the SGD in terms of accuracy. Additionally, despite its subpar performance, the KNN method is the most effective technique while performing linear regression. The proposed model demonstrated 97 % precision in successfully predicting object recognition using the COCO dataset. Different ML approaches provide widely varying outcomes, suggesting that model accuracy may be improved. The difference between the score on the recall and the F1 test lends credence to this assertion. We expect to broaden the prospect of the study in the future to create a hybrid detector that can detect items.

Conflict of Interest

The authors have declared no conflict-of-interest statements.

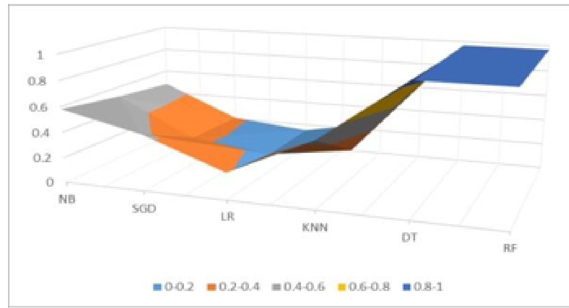

Fig. 10. Performance Evaluation of All Models Based on Classifiers Precision, Recall and F1-score Measure

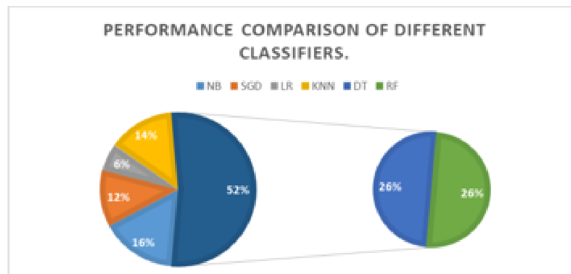

Fig. 11. The Total Ratios of Performance Compression

References

[1] C.-J. Du, H.-J. He, and D.-W. Sun, "Object classification methods," in *Computer vision technology for food quality evaluation*, pp. 87–110, Elsevier, 2016.

[2] E. K. Wang, Y. Li, Z. Nie, J. Yu, Z. Liang, X. Zhang, and S. M. Yiu, "Deep fusion feature based object detection method for high resolution optical remote sensing images," *Applied Sciences*, vol. 9, no. 6, p. 1130, 2019.

[3] D. Racheed, R. Muin, and A. Jaylan, "Object detection and object classification using machine learning algorithms," *International Journal of Information Technology and Applied Sciences (IJITAS)*, vol. 2, no. 3, pp. 21–29, 2020.

[4] L. Li, Y. Zhang, and Y. Zhao, "k-nearest neighbors for automated classification of celestial objects," *Science in China Series G: Physics, Mechanics and Astronomy*, vol. 51, no. 7, pp. 916–922, 2008.

[5] I. S. Abuhaiba and H. M. Dawoud, "Combining different approaches to improve arabic text documents classification," *International Journal of Intelligent Systems and Applications*, vol. 9, no. 4, p. 39, 2017.

[6] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN computer science*, vol. 2, no. 3, p. 160, 2021.

[7] H. M. Fadhil, N. Q. Makhool, M. M. Hummady, and Z. O. Dawood, "Machine learning-based information security model for botnet detection," *Journal of Cybersecurity and Information Management (JCIM) Vol*, vol. 9, no. 01, pp. 68–79, 2022.

[8] M. F. Alotaibi, M. Omri, S. Abdel-Khalek, E. Khalil, and R. F. Mansour, "Computational intelligence-based harmony search algorithm for real-time object detection and tracking in video surveillance systems," *Mathematics*, vol. 10, no. 5, p. 733, 2022.

[9] H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "A framework for predicting airfare prices using machine learning," *Iraqi J. Comput. Commun. Control Syst. Eng*, vol. 22, pp. 81–96, 2022.

[10] Q. Li, C. Tai, and E. Weinan, "Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1474–1520, 2019.

[11] H. R. Pourghasemi, A. Gayen, S. Park, C.-W. Lee, and S. Lee, "Assessment of landslide-prone areas and their zonation using logistic regression, logitboost, and naïvebayes machine-learning algorithms," *Sustainability*, vol. 10, no. 10, p. 3697, 2018.

[12] M. A. Abuzneid and A. Mahmood, "Enhanced human face recognition using lbph descriptor, multi-knn, and back-propagation neural network," *IEEE access*, vol. 6, pp. 20641–20651, 2018.

[13] M. Vasumathi and M. Kamarasan, "A comparative study on traditional data mining and big data mining classification algorithms," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 4, pp. 1523–1527, 2019.

[14] S. Burde and S. V. Budihal, "Multiple object detection and tracking using deep learning," in *Proceedings of International Conference on Communication, Circuits, and Systems: IC3S 2020*, pp. 257–263, Springer, 2021.

[15] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *Journal of Big data*, vol. 8, no. 1, pp. 1–27, 2021.

[16] W. Li, "Analysis of object detection performance based on faster r-cnn," in *Journal of Physics: Conference Series*, vol. 1827, p. 012085, IOP Publishing, 2021.

[17] X. Zhang, F. Wan, C. Liu, X. Ji, and Q. Ye, "Learning to match anchors for visual object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3096–3109, 2021.

[18] Y.-L. Li, S. Zhou, X. Huang, L. Xu, Z. Ma, H.-S. Fang, Y. Wang, and C. Lu, "Transferable interactiveness knowledge for human-object interaction detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3585–3594, 2019.

[19] C. Yang, Z. Huang, and N. Wang, "Querydet: Cascaded sparse query for accelerating high-resolution small object detection," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 13668–13677, 2022.

[20] Y. Li and S. Wang, "R (det) 2: Randomized decision routing for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4825–4834, 2022.

[21] V. K. Gupta, A. Gupta, D. Kumar, and A. Sardana, "Prediction of covid-19 confirmed, death, and cured cases in india using random forest model," *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 116–123, 2021.

[22] R. D. Haamied, B. Q. Al-Abudi, and R. N. Hassan, "Classification coco dataset using machine learning algorithms," *Turkish Journal of Physiotherapy and Rehabilitation*, vol. 32, p. 3.

[23] S. Chun, W. Kim, S. Park, M. Chang, and S. J. Oh, "Eccv caption: Correcting false negatives by collecting machine-and-human-verified image-caption associations for ms-coco," in *European Conference on Computer Vision*, pp. 1–19, Springer, 2022.

[24] K. Saito, P. Hu, T. Darrell, and K. Saenko, "Learning to detect every thing in an open world," in *European Conference on Computer Vision*, pp. 268–284, Springer, 2022.

[25] T.-Y. Lin, G. Patterson, M. R. Ronchi, Y. Cui, M. Maire, S. Belongie, L. Bourdev, R. Girshik, J. Hays, P. Perona, *et al.*, "Coco dataset," *COCO Consortium*, 2017.

[26] S. A. H. Alrubaie and A. H. Hameed, "Dynamic weights equations for converting grayscale image to rgb image," *Journal of University of Babylon for Pure and Applied Sciences*, vol. 26, no. 8, pp. 122–129, 2018.

[27] P. Parsania, P. V. Virparia, *et al.*, "A review: Image interpolation techniques for image scaling," *International*

Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 12, pp. 7409–7414, 2014.

[28] B. Oktavianto and T. W. Purboyo, "A study of histogram equalization techniques for image enhancement," *International Journal of Applied Engineering Research*, vol. 13, no. 2, pp. 1165–1170, 2018.

[29] W. A. Mustafa and M. M. M. Abdul Kader, "A review of histogram equalization techniques in image enhancement application," in *Journal of Physics: Conference Series*, vol. 1019, p. 012026, IOP Publishing, 2018.

[30] W. Wang, X. Wu, X. Yuan, and Z. Gao, "An experiment-based review of low-light image enhancement methods," *Ieee Access*, vol. 8, pp. 87884–87917, 2020.

[31] I. Adiyasa, A. Prasetyono, A. Yudianto, P. Begawan, and D. Sultantyo, "Bilinear interpolation method on 8x8 pixel thermal camera for temperature instrument of combustion engine," in *Journal of Physics: Conference Series*, vol. 1700, p. 012076, IOP Publishing, 2020.

[32] D. Khaledyan, A. Amirany, K. Jafari, M. H. Moaiyeri, A. Z. Khuzani, and N. Mashhadi, "Low-cost implementation of bilinear and bicubic image interpolation for real-time image super-resolution," in *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–5, IEEE, 2020.

[33] D. Khaledyan, A. Amirany, K. Jafari, M. H. Moaiyeri, A. Z. Khuzani, and N. Mashhadi, "Low-cost implementation of bilinear and bicubic image interpolation for real-time image super-resolution," in *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–5, IEEE, 2020.

[34] M. Z. Nasution, "Face recognition based feature extraction using principal component analysis (pca)," *Journal of Informatics and Telecommunication Engineering*, vol. 3, no. 2, pp. 182–191, 2020.

[35] L. R. Fleah and S. A. Al-Aubi, "A face recognition system based on principal component analysis-wavelet and support vector machines," *Cihan University-Erbil Scientific Journal*, vol. 3, no. 2, pp. 14–20, 2019.