FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

# YOLO Based Deep Learning Model for Segmenting the Color Images

**D. Rasi[1], M. AntoBennet[2], P. N. Renjith[3], M. R. Arun[4] and D. Vanathi[5]**

*[1]Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, India. rasid@skcet.ac.in; priyamudanrasi@gmail.com.*

*[2]Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India, drmantobenet@veltech.edu.in*

*[3]Department of Computer Science, Vellore Institute of Technology, Chennai, India, renjith.pn@vit.ac.in*

*[4]Department of Electronics and Communication Engineering Veltech Rangarajan Dr. Sagunthala R&D institute of Science and Technology, Chennai, India, mrarunresearch@gmail.com*

*[5]Department of Computer Science and Engineering, Nandha Engineering College, Erode, India, vanathi.d@nandhaengg.org*

***Correspondence:** rasid@skcet.ac.in

**ABSTRACT-** The first stage is to extract fine details from a picture using Red Green Blue (RGB) colour space is colour image segmentation. Most grayscale and colour picture segmentation algorithms use original or updated fuzzy c-means (FCM) clustering. However, due to two factors, the majority of these methods are inefficient and fail to produce the acceptable segmentation results for colour photos. The inclusion of local spatial information often results in a high level of computational complexity due to the repetitive distance computation between clustering centres and pixels within a tiny adjacent window. The second reason is that a typical neighbouring window tends to mess up the local spatial structure of images. Color picture segmentation has been improved by introducing Deep Convolution Neural Networks (CNNs) for object detection, classification and semantic segmentation. This study seeks to build a light-weight for object detector that uses a depth and colour image from a publically available dataset to identify objects in a scene. It's likely to output in the depth way by expanding the YOLO network's network architecture. Using Taylor based Cat Salp Swarm algorithm (TCSSA), the weight of the suggested model is modified to improve the accuracy of region extraction findings. It is possible to test the detector's efficacy by comparing it to various datasets. Testing showed that the suggested model is capable of segmenting input into multiple metrics using bounding boxes. The results shows that the proposed model achieved 0.20 of Global Consistency Error (GCE) and 1.85 of Variation of Information (VOI) on BSDS500 dataset, where existing techniques achieved nearly 1.96 to 1.86 of VOI and 0.25 to 0.22 of GCE for the same dataset.

**Keywords:** Convolutional Neural Network; Color Image Segmentation; Computational Complexity; Fuzzy C-Means; Grayscale; Taylor based Cat Salp Swarm.

**ARTICLE INFORMATION**

**Author(s)**: D. Rasi, M. AntoBennet, P. N. Renjith, M. R. Arun and D. Vanathi;

**Received**: 17/12/2022; **Accepted**: 15/02/2023; **Published**: 30/05/2023;
**e-ISSN**: 2347-470X;
**Paper Id**: IJEER221201;
**Citation:** 10.37391/IJEER.110217
**Webpage-link**:
https://ijeer.forexjournal.co.in/archive/volume-11/ijeer-110217.html


**Publisher's Note:** FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

## 1. INTRODUCTION

Classical image processing topics include picture segmentation. Divides the original image rendering to the intensity, coloration, texture, and characteristics of the image. Image segmentation is commonly used as a pre-processing step for more advanced image analysis, object recognition. It is possible to break up an image into smaller, more manageable chunks using a technique called image segmentation [1-2]. Simplifying or transforming an image's representation into something more relevant and understandable is the goal. Image segmentation is utilised in a variety of computer vision applications, including [3], medical image investigation [4], and other monitoring requests [5]. It is possible to classify picture segmentation approaches [6-8]. There are a number of popular and useful clustering-based methods, such as k-means, which are straightforward to implement, fast, and produce decent results.

Color picture segmentation is a dangerous role in image investigation, interpretation, and computer vision [9-11]. As a result of the RGB colour space's, it highly linked components analysis are impossible for B space. There are only three components of the HSI colour space that aren't important (I). The use of hue information in colour image segmentation has increased since it is more accurate and human vision is closer to the HSI colour space than RGB [12-14]. Segmentation precision is therefore critical to the performance of a higher-class dispensation scheme, where segmentation methods include extraction of regions and feature clusters.

But in the last few years a new generation of image segmentation model has been developed by the use of deep learning, which has resulted in an entirely new paradigm shift in the field of segmentation [15-16]. Machine learning researchers have recently become interested in using deep

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

learning to automatically access the feature. By merging lower-level features, deep learning creates a higher-level feature. Computer vision uses deep learning algorithms to extract low-level expressions from the original image, such as edge detection and wavelet filtering for example. Then, the use of linear or nonlinear combination to get high-level expression from these low-level expressions using Auto Encoder Sparse Coding. For computer vision, CNN has proven to be a useful tool that can learn intermediate features and handle ambiguous ones effectively [17]. Compared to unrivalled artificial features, this is a pittance. Although CNNs accept low-resolution images to reduce the network's calculations and parameters.

It is therefore necessary to build an object that receives depth and colour information as inputs, then determines the item's position during the segmentation process. The result of the proposed model is a 3D box based on YOLO's construction. Even if the scene has occlusions that are difficult to distinguish using only 2D-based approaches and depth pictures, this can be expected to be robust. The OWM (optimal weighted-mean) for each object is applied to the bounding box to integrate the detection findings. For the sake of evaluating the model's efficacy based on various metrics (described in the following sections), experimental analysis is being conducted.

*Section 2* delivers a review of relevant literature, while *Section 3* offers a brief explanation of the suggested model. *Section 4* offerings the results of testing the projected model's various parameters against publicly available datasets using existing approaches. Finally, *Section 5* shows the research's scientific impact.

## 2. RELATED WORK

As a means of reducing the complexity of thresholding on circular histograms, Kang, C., [18] created the cumulative distribution function. The histogram is then linearized in either an anticlockwise or a clockwise orientation using the optimal entropy of the cumulative distribution function, as described in this study. Color image segmentation is done using a fuzzy entropy thresholding algorithm on a linearized histogram. According to the experimental results, compared to circular thresholding approach, the proposed method may raise the pixel accuracy index by 30.12 percent and the similarity index by 27.53 percent.

A multi-level thresholding with the most favourable maximising objective as Kapur, Otsu and deep learning model was used in this paper [19-20] to extract the precise threshold. Execution of non-parametric goal functions rises exponentially in computing time as segmentation level increases. Multiple investigations are being conducted to improve the speed of goal functions and various metaheuristics. The computational complexity was unlocked by switching to the most powerful, robust, and recently announced metaheuristic. Investors buy and sell shares in order to make a profit. In both stable and unpredictable market conditions, shareholders might use a strategic strategy to precisely investigate and utilise the market. Additionally, allowing investors with lower fitness levels to learn from those with higher and intermediate fitness levels is a notable time-saving strategy.

To compare thresholding segmentation approaches, Wang, S. [21] created an Otsu and Kapur's entropy. It is for this reason that they are referred to as the goal functions. There is an exponential rise in temporal complexity with each additional threshold. MALO, a modified antlion optimizer method, is developed to address this shortcoming and discover the optimum threshold values. Opposition-based learning improves both search accuracy and convergence performance. Eleven cutting-edge algorithms will be compared with the IEEE CEC 2017 benchmark functions. The algorithm's segmentation performance is evaluated through a number of trials. This includes the following metrics: fitness value, peak SNR, structural similarity index (SPI).

Wei, T. [22] important part of improving robustness in the FCM objective function is using Non-local information can be eliminated by using local variance templates. Using the denominator to reduce iterations and solve the problem of convergence early when membership has an outlier is a common practise in statistical analysis. The segmentation performance of colour images by allowing for a more flexible weighting for different dimensions. Using noisy grayscale and noisy colour images, this model outperforms previous fuzzy-based clustering methods in terms of performance.

Wu, C. [23] an initial polynomial kernel function is added, and the local neighbourhood information of the pixel is used to change the total Bregman divergence (TBD), which solves the drawbacks. Second, the algorithm's anti-noise performance is further enhanced by kernelized TBD kernelized TBD into the objective function of KWPFLICM. TKWPFLICM's average segmentation accuracy improvement ranges from 0.791 percent to 33.237 percent when compared to existing fuzzy clustering-related methods, according to the findings of our experiments. Because of this, TKWPFLICM's anti-noise resilience and segmentation accuracy are both improved.

M. Takahashi's [24] goal is to build a light-weight object detector that accepts depth and colour images from a stereo camera as input sources. It is likely to depth direction by extending the YOLOv3, object detectors that employ distance information as input for automatic driving are being actively investigated. Because of its extensive network structure, the conventional detector's real-time property is compromised. It is possible to test the detector's efficacy by comparing it to various datasets. Experiment results show that the suggested model is accomplished of producing 3D bounding boxes and identifying people whose bodies are partially obscured.

Small target detection can be enhanced by Xianbao, C., [25] proposing an improved technique based on YoloV3. Starting with the feature map acquisition network, we've made a few tweaks to it. The original network architecture's 2-step down sampling convolution network is replaced by an image, which increases the feature values of large and tiny objects. An additional module for recognising small objects in images is included in order to reduce the loss of morpheme features due to no-feature value filling. Thirdly, the recall rate rises from

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

84.6 to 91.3 percent, and the average accuracy increases from 95.55 percent to 97.3 percent when compared with YOLO V3. Magalhães SA et al., [26] Robotic tomato picking requires advances in edge artificial intelligence for in-situ and real-time visual tomato detection. Five deep learning models were chosen, trained, and benchmarked using this dataset to identify green and reddish greenhouse-grown tomatoes. Only the Single-Shot MultiBox Detector (SSD) and YOLO architectures were taken into account when designing our robotic platform. The outcomes demonstrated that the system is capable of identifying green and reddish tomatoes, even those hidden by leaves. SSD MobileNet v2 outperformed SSD Inception v2, SSD ResNet 50, SSD ResNet 101, and YOLOv4 Tiny in terms of performance, achieving an F1-score of 66.15%, a mAP of 51.46%, and an inference time of 16.44ms on an NVIDIA Tesla T4 machine running the Turing architecture.

Abas SM et al., [27] have demonstrated that breaking a task down into smaller ones would improve performance and accuracy. Additionally, the outcomes demonstrate that the (CAD3) successfully classified leukocytes with an accuracy of 94.3% and detected them with an average precision AP of up to 96%. Furthermore, the CAD3 report provides comprehensive details regarding the size and quantity of WBC for each kind in the input image. Finally, the outcomes demonstrated that the CAD3 system is more effective when combined with additional datasets like the Blood Cell Count Dataset and the Acute Lymphoblastic Leukemia - Data Base (ALL-DB1) (BCCD).

### 2.1 Problem Statement

Many image segmentation algorithms have been suggested, but none of them can provide a cohesive framework for attaining rapid and effective image segmentation. There are two reasons why image segmentation is challenging. It's important to remember that segmenting an image is a problem with many possible solutions, which means that a single image might have several different best segmentation outcomes. Because of noise, background, low SNR, and non-uniform intensity, an image is always complex. This means that a general segmentation framework is difficult to devise for complex image segmentation jobs because of this.

### 2.2. Contribution of the Research Study

Segmentation of a color image, identifying pixels from background images, is one of the most difficult tasks in image analysis, providing important information about the shapes and sizes of these objects. Various automated segmentation systems have been proposed by many researchers applying available techniques. Previous systems were built on traditional methods such as edge detection filters and mathematical methods. Subsequently, machine learning methods for extracting craft features have long become a mainstream technology. Designing and extracting these features is always a primary concern for the development of such a system, and the complexity of these methods is seen as a major limitation for their deployment. The promising capability of deep learning (DL) approaches has established it as a primary option for image segmentation, and traditional image segmentation techniques include intermediate level steps such as analysis of artifact features. DL, on the other hand, does not require such steps and analyzes features within the network. The performance of the compression is much better than the traditional imaging approach and can be improved by adjusting the parameters or by using different types of activation functions. Although training a DL model is computationally expensive, the number of hidden layers and batch sizes can be modified to overcome such problems. This motivates the development of this research study.

## 3. PROPOSED MODEL

Deep learning is briefly described here, while examples are presented in Section 4 of the document. Even if a CNN-based YOLO model structure can reach optimum object detection accuracy throughout the segmentation process. Even though it is substantially smaller than YOLO and YOLOv2, the tiny YOLO (60,5MB) is still too often used on embedded devices [28-30]. The goal of this research is to improve the CNN-based micro-YOLO model structure. Analyzing the time-consuming component of the CNN is used to optimise weight for the YOLO model, according to TCSSA [28]. Although the number of parameters, model dimension and detection speed are reduced, the accuracy of the detection is maintained at a low level.

### 3.1 Tiny YOLO network structure

Convolution, pooling, rectification, and normalisation are only a few of the building pieces that make up CNN. Convolution has the purpose of extracting characteristics. Pooling abstracts a feature and enhances the model's ability to generalise. To improve the model's ability to learn, it is necessary to make corrections. To speed up the training of a model, normalisation uses nonlinear processing. Tiny YOLO is no exception to the complexity of the CNN, which is mostly due to the vast number of parameters. Table 1 shows a network diagram of YOLO's little network

**Table 1: A table for Tiny YOLO network**

| Layer | Input | Name | Filters | Output |
|---|---|---|---|---|
| 0 | 416×416×3 | conv | 16 | 416×416×16 |
| 1 | 208×208×16 | max | 2×2/2 | |
| 2 | 208×208×16 | conv | 32 | 208×208×32 |
| 3 | 104×104×32 | max | 2×2/2 | |
| 4 | 104×104×32 | conv | 64 | 104×104×64 |
| 5 | 52×52×64 | max | 2×2/2 | |
| 6 | 52×52×64 | conv | 128 | 52×52×128 |
| 7 | 26×26×128 | max | 2×2/2 | |
| 8 | 26×26×128 | conv | 256 | 26×26×256 |
| 9 | 13×13×256 | max | 2×2/2 | |
| 10 | 13×13×256 | conv | 512 | 13×13×512 |
| 11 | 13×13×512 | max | 2×2/1 | |
| 12 | 13×13×512 | conv | 1024 | 13×13×1024 |
| 13 | 13×13×1024 | conv | 1024 | 13×13×1024 |
| 14 | 13×13×1024 | conv | 30 | 13×13×30 |
| 15 | – | detection | – | – |

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

According to *Table 1*, the YOLO network structure model has convolution kernel parameters. It's easy to see that the sum of kernels and parameters go hand in hand. Sum of feature map channels, size of convolution kernels, and convolution kernel count all affect how many convolution kernel parameters are needed in a given layer-by-layer convolution procedure. Following the input layer, the number of convolution kernel channels is consistent with that of the previous layer's kernels in following operation levels. As a result of the network's parameter size, the file saved by small YOLO contains 64MB of data. The program's function call relationship can be gleaned using the GNU profiler. *Figure 1* depicts the function diagram.

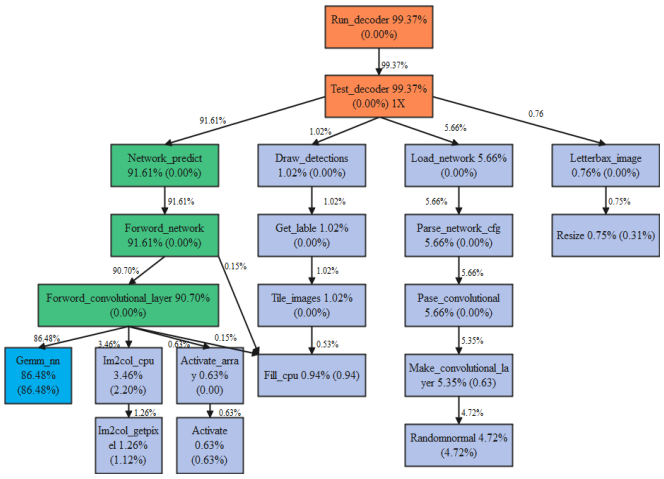



**Figure 1:** The function call relationship diagram of tiny YOLO in the test set

The gemmnn function, which performs multiplication, consumes 86% of the program's total runtime, as seen in *figure 1*. One for each of the network's nine convolution layers. As can be observed, the detection model for tiny YOLO uses a lot of time due to the high number of factors and subsequent high level of calculation. Using the optimization technique for weight parameters, the convolution kernel parameters can be reduced, resulting in a reduction in computing expenditure for the operation of the convolution layer for extracting features from the little YOLO network. The workflow of proposed model is provided in *figure 2*.

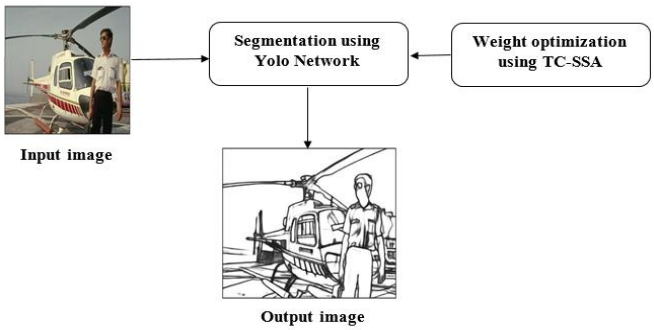



**Figure 2:** Workflow of proposed model

An input feature map is typically used in conjunction with a three-dimensional convolution kernel in conventional convolution operations. Convolution layer l's convolution kernel number is equal to $f^l \in R^{H\times W\times D^t}$ when XL's input tensor is $x^l \in R^{H^l\times W^l\times D^l}$. Below *table 2*, the sub-section explains how TCSSA is used to maximise the W in this case. The essentially all channels at this place in order to total the $HWD^l$ elements processed by a single convolution operation ($i.e.D^l$). Configuration is presented in *table 2* that presents a detailed breakdown of the proposed model.

**Table 2: Hyper-parametric configuration table**

| Interactive times | Input size | Weight decay | Learning rate | Training algorithm | Momentum | Batch |
|---|---|---|---|---|---|---|
| 42000 | 416×416×3 | 0.0005 | 0.001 | gradient descent | 0.9 | 64 |

## 3.2 Weight selection using Proposed Taylor C-SSA

Here, Taylor C-SSA is described that suggested a Taylor series and C-SSA integration that inherits the advantages of both techniques. Predicting the linear component is done using the Taylor series, which describes the stored historical data. When dealing with complex functions, Taylor series can be used as a straightforward and easy way to compute the solutions. The key advantages of the Taylor series are that it ensures an accurate assessment of the common functions and quickly converges. To that end, the C-SSA combines the advantages of the Cat Swarm Optimization (CSO) [31] and the SSA [31] algorithms, gaining the best of both worlds in the process of finding the optimal solution. One control parameter is changed adaptively, making the technique easier to implement. The C-SSA combines the rewards of both procedures to achieve the global optimal solution while maintaining a good balance between the exploitation and exploration phases. In addition to being simple to build, the C-SSA has a single control parameter that is constantly updated. As a result, the hybrid algorithm's primary goal is to address the drawbacks of the regular SSA algorithm by incorporating CSO. A global optimal solution can be found using the algorithm provided in this paper. *Table 3* shows the description of TC-SSA.

**Table 3: Configuration table for optimization model**

| | |
|---|---|
| Number of iterations | 200 |
| Population size | 416 |
| Upper boundary | 255 |
| Lower Boundary | 0 |
| Number of search agent | 20 |

For this reason it is intended that the suggested Taylor C-SSA algorithm will be able to overcome some of the C-SSA algorithm's drawbacks by using the Taylor series. Convergence is enhanced and an optimal solution is found by integrating the two strategies. The suggested Taylor CSSA algorithm is broken down into the following phases:

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

**Initialization**
The algorithm begins with the Salp population initialization step, which is represented as follows,

$$C = \{C_1, C_2, \dots, C_0, \dots C_u\} \quad (1)$$

Where, $C_0$ signifies the position of $o^{th}$ Salp, and u is the position of $u^{th}$ Salp.

**Evaluation of fitness function**
For the best solution, the fitness function must be evaluated next. Weight input from the convolutional layer generates the objective fitness. The derived fitness function is used to evaluate the fitness of each solution in a population. The ideal option is that which provides the greatest fitness return on investment.

**Computation of the novel solutions**
Based on the proposed algorithm, the solutions are updated in this stage. The CSO method is more searchable and allows for a better balance between exploitation and exploration in order to find the global best solution. The standard CSO equation is therefore utilised to give a more accurate result. Thus, the current iteration of the CSO algorithm uses the cat's current position and velocity to determine the update position. The cat's speed is adjusted to match the prey's speed as soon as it is known. Thus, the updated position of the cat o can be found by using the following formula,

$$C_{u+1}^0 = C_u^0 + v_{u+1}^0 \quad (2)$$

Where, $C_u^0$ signifies the cat position.
Once the cat has determined the prey's location, it decides to move, and thus, the cat's velocity is altered and the new velocity is used $v_{u+1}^0$ is characterized as,

$$v_{u+1}^0 = v_u^0 + v_1 \times \lambda_1(C^* - C_u^0) \quad (3)$$

A random number (*v*) determines the position of the cat *o* in this iteration; its velocity (*v u0*) determines its velocity (*C u0*); and *C** indicates the best solution (*C**).

In order to increase estimation accuracy and convergence, the cat's speed should be adjusted in accordance with the Taylor series. In addition, the Taylor series under complex functions is simple and straightforward to compute. Thus, the velocity equation derived from the Taylor series can be summarised as follows,

$$v_{u+1}^0 = 0.5v_u^0 + 1.3591v_{u-1}^0 - 1.3591v_{u-2}^0 + 0.6795v_{u-3}^0 - 0.2259v_{u-4}^0 + 0.555v_{u-5}^0 - 0.0104v_{u-6}^0 + 1.38e^{-3}v_{u-7}^0 - 9.92e^{-5}v_{u-8}^0 \quad (4)$$

As a result, the speed of the cat in this iteration is determined by,

$$v_u^0 = \frac{1}{0.5}\begin{bmatrix} v_{u+1}^0 - 1.3591v_{u-1}^0 + 1.359v_{u-2}^0 \\ 0.6795v_{u-3}^0 + 0.2259v_{u-4}^0 - 0.555v_{u-5}^0 \\ +0.0104v_{u-6}^0 - 1.38^{-3}v_{u-7}^0 + 9.92e^{-5}v_{u-8}^0 \end{bmatrix} \quad (5)$$

Using the Taylor series, the velocity is updated by substituting *Eq. (5)* in *Eq. (3)* below.

$$v_{u+1}^0 = \frac{1}{0.5}\begin{bmatrix} v_{u+1}^0 - 1.3591v_{u-1}^0 + 1.359v_{u-2}^0 - \\ 0.6795v_{u-3}^0 + 0.2259v_{u-4}^0 - 0.555v_{u-5}^0 \\ +0.0104v_{u-6}^0 - 1.38^{-3}v_{u-7}^0 + 9.92e^{-5}v_{u-8}^0 \end{bmatrix} + v_1 \times \lambda_1(C^* - C_u^0) \quad (6)$$

$$v_{u+1}^0 = \frac{1}{0.5}[v_{u+1}^0] - \frac{1}{0.5}\begin{bmatrix} 1.3591v_{u-1}^0 - 1.359v_{u-2}^0 + 0.6795v_{u-3}^0 \\ 0.2259v_{u-4}^0 + 0.555v_{u-5}^0 - 0.0104v_{u-6}^0 \\ 1.38^{-3}v_{u-7}^0 - 9.92e^{-5}v_{u-8}^0 \end{bmatrix} + v_1 \times \lambda_1(C^* - C_u^0) \quad (7)$$

$$v_{u+1}^0 = \frac{1}{0.5}\begin{bmatrix} 1.3591v_{u-1}^0 - 1.359v_{u-2}^0 + 0.6795v_{u-3}^0 \\ 0.2259v_{u-4}^0 + 0.555v_{u-5}^0 - 0.0104v_{u-6}^0 \\ 1.38^{-3}v_{u-7}^0 - 9.92e^{-5}v_{u-8}^0 \end{bmatrix} + v_1 \times \lambda_1(C^* - C_u^0) \quad (8)$$

Reorganizing the above reckoning, the resultant equation as,

$$v_{u+1}^0 = \begin{bmatrix} 2.7182v_{u-1}^0 - 2.718v_{u-2}^0 + 1.359v_{u-3}^0 - \\ 0.4518v_{u-4}^0 + 0.111v_{u-5}^0 - 0.0208v_{u-6}^0 + \\ 0.00276v_{u-7}^0 - 0.0001984v_{u-8}^0 \end{bmatrix} + v_1 \times \lambda_1(C^* - C_u^0) \quad (9)$$

$$v_{u+1}^0 - v_1 \times \lambda_1(C^* - C_u^0) = \begin{bmatrix} 2.7182v_{u-1}^0 - 2.718v_{u-2}^0 + 1.359v_{u-3}^0 - \\ 0.4518v_{u-4}^0 + 0.111v_{u-5}^0 - 0.0208v_{u-6}^0 + \\ 0.00276v_{u-7}^0 - 0.0001984v_{u-8}^0 \end{bmatrix} \quad (10)$$

The current iteration of the cat's velocity can be expressed using the equation above.

$$v_u^0 = \begin{bmatrix} 2.7182v_{u-1}^0 - 2.718v_{u-2}^0 + 1.359v_{u-3}^0 - \\ 0.4518v_{u-4}^0 + 0.111v_{u-5}^0 - 0.0208v_{u-6}^0 + \\ 0.00276v_{u-7}^0 - 0.0001984v_{u-8}^0 \end{bmatrix} \quad (11)$$

YOLO's weight optimization is more accurate and effective thanks to the C-SSA method. By merging the update rules of CSO and SSA, we arrive at the C-SSA update rule. With the C-SSA, the global optimal solution can be generated in the exploitation and exploration phases. Using C-SSA, the equation for updating is given by,

$$C_{u+1}^0 = \frac{1-v_1\times\lambda_1}{1-2v_1\times\lambda_1} \times \left[C_u^{0-1} - \frac{v_u^0 + v_1\times\lambda_1\times C^*}{1-v_1\times\lambda_1}\right] \quad (12)$$

After substituting *Eq. (11)* into *Eq. (12)*, the final Taylor C-SSA is represented by *Eq. (12)*,

$$C_{u+1}^0 = \frac{1-v_1\times\lambda_1}{1-2v_1\times\lambda_1} \times \left[C_u^{0-1} \frac{\begin{bmatrix} 2.7182v_{u-1}^0 - 2.718v_{u-2}^0 + 1.359v_{u-3}^0 - \\ 0.4518v_{u-4}^0 + 0.111v_{u-5}^0 - 0.0208v_{u-6}^0 + \\ 0.00276v_{u-7}^0 - 0.0001984v_{u-8}^0 \end{bmatrix} + v_1\times\lambda_1\times C^*}{1-v_1\times\lambda_1}\right] \quad (13)$$

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

*V* is a random integer, *C 1* denotes the optimal solution, and *u* is the velocity of cat *o* in the $u^{th}$ location, which is *C_** or *W*. The YOLO network effectively segmented the colour pictures by adjusting the weights [31-32]. Next, we'll look at the model's ability to be validated.

## 4. RESULTS AND DISCUSSION

Ubuntu 16.04 is the operating system used in this experiment. Intel (R) core (TM) i7-7700 CPU is used in the system. Titan x Pascal graphics cards with 64GB of server memory power this system. Tensor flow serves as the foundation for this application. In YOLO CSO, Every cat has its own position composed of M dimensions, velocities for each dimensions, a fitness value, which represents the accommodation of the cat to the fitness function, and a flag to identify whether the cat is in seeking mode or tracing mode. The final solution would be the best position in one of the cats due to CSO keeps the best solution till it reaches the end of iterations. Whereas in YOLO SSA, algorithm to solve some of the optimization problem that can be either single objective or multi-objective, and the algorithm will have to be specific to each.

### 4.1 Datasets Description

This technique was evaluated using two real-world datasets in the following section. BSDS500 [33-34], the Berkeley segmentation, provides the images for the first dataset to implement in python. Each of these photos has a resolution of 481 x 321 and is separated into two sets: one for testing and the other for training. An example of BSDS500 can be seen in *figure 3*, where four photos of this type are shown as an example. For each of the four to nine image variations, there is a set of ground truths obtained by a single human subject. There are 715 photos in the Stanford background dataset (SBD) [35], which represents outside images. This type of image is depicted in four different ways in *figure 4*. Objects with blurry foreground boundaries, many foreground objects, and detailed background regions are all present in these photos, which are used for ground truth segmentation. This complicates the evaluation of segmentation methods in these photos.

**Figure 3:** Examples of BSDS500 images

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

**Figure 4:** Examples of SBD images

## 4.2 Performance Metrics

Competitive segmentation approaches are evaluated using a set of performance indicators. PRI as probabilistic rand index, VI, GCE, and BDE stand for boundary displacement error, respectively. These measures are defined in the following paragraphs:

**Probabilistic Rand Index (*PRI*):** To classify pixels, it uses the similarity between labels to calculate the similarity.

$$PRI(S, S_g) = \frac{1}{T}\sum_{i<j}[c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})] \quad (14)$$

Where $c_{ij}$ and $p_{ij}$ signify the event that pixels i and j have the similar label and its likelihood.

**VOI:** depends on the difference in conditional entropy between comparison findings from two different clusters.

$$VOI(S, S_g) = H(S|S_g) + H(S_g|S) \quad (15)$$

Where $H(S|S_g)$ and $H(S|S_g)$ are the conditional entropies.

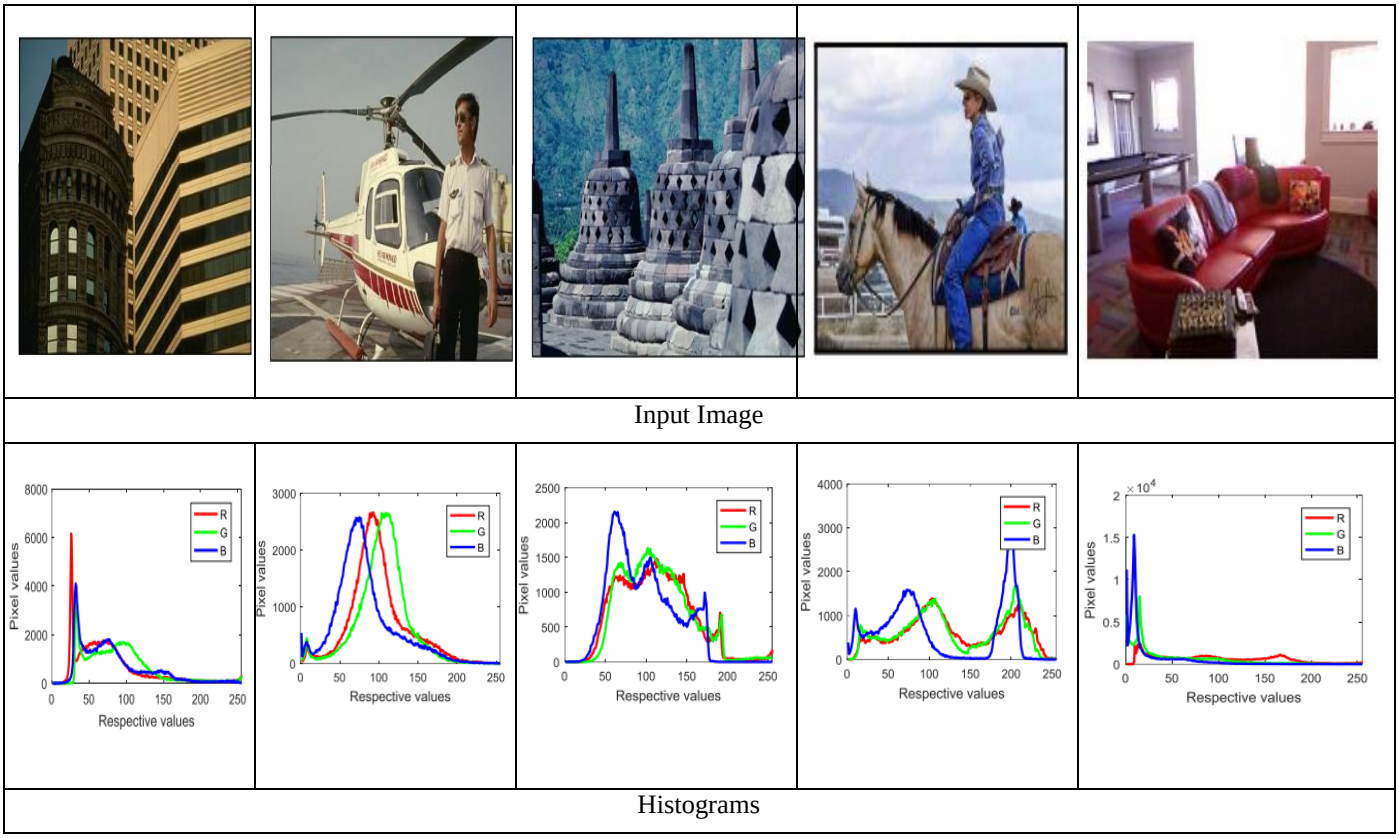
**GCE:** It assesses the overall inconsistency of two segmented pictures.

**Boundary Displacement Error (BDE):** Between two segmented images, it computes the average pixel displacement error.

It is generally accepted that the best method is one that delivers a higher PRI while maintaining low values for VOI, BDE, and GCE.

## 4.3 Segmentation Analysis

This section analysis how the proposed deep learning technique segmented the input color images. For instance, *figure 5* shows the three rows, where input image in first row, ground truth image in second row and finally, segmented images using proposed model in third row.



FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

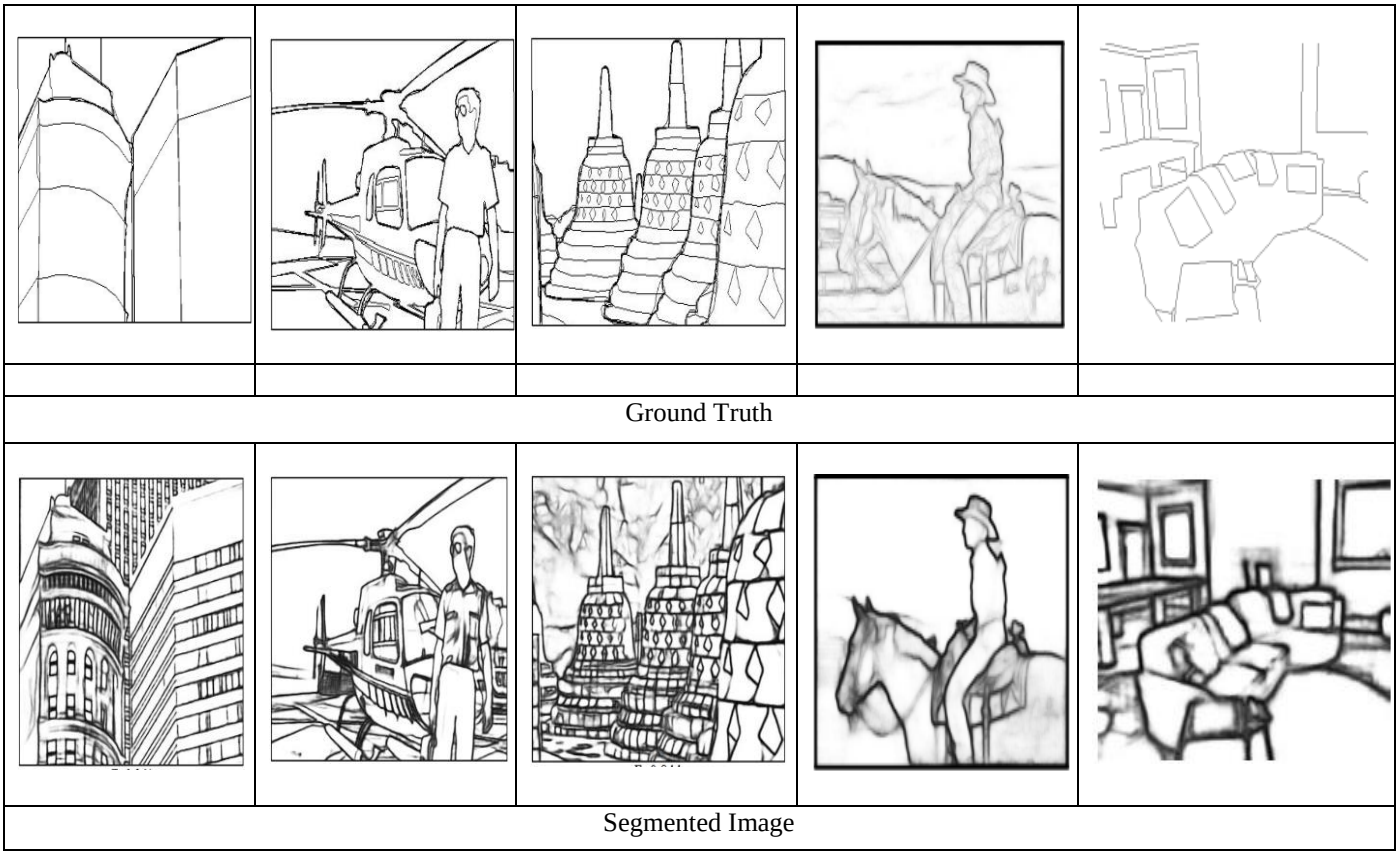



*Table 4* and *figure 6-7* shows the experimental analysis of proposed model in term of various metrics for different color images. For better graphical results, only some samples are taken and it is shown in *figure 6* and *7*.

**Table 4: Experimental Analysis of Proposed Model in terms of various metrics**

| Image | RI | VOI | GCE | BDE |
|---|---|---|---|---|
| 1 | 0.5522 | 5.5361 | 0.3414 | 10.4405 |
| 2 | 0.5776 | 5.9600 | 0.39050 | 10.2830 |
| 3 | 0.5057 | 5.0236 | 0.3619 | 10.9295 |
| 4 | 0.5196 | 5.5991 | 0.3163 | 10.9178 |
| 5 | 0.3997 | 5.0100 | 0.3498 | 11.5793 |
| 6 | 0.3951 | 5.7501 | 0.3268 | 11.4561 |
| 7 | 0.3353 | 5.8606 | 0.3398 | 11.9886 |
| 8 | 0.3676 | 5.7983 | 0.3489 | 11.4242 |
| 9 | 0.4976 | 6.6860 | 0.4370 | 11.8625 |
| 10 | 0.4703 | 6.9471 | 0.4816 | 11.9366 |
| 11 | 0.4569 | 6.6230 | 0.4893 | 11.7475 |
| 12 | 0.4651 | 6.9551 | 0.4586 | 11.2957 |
| 13 | 0.5802 | 5.8374 | 0.3805 | 9.2311 |
| 14 | 0.5012 | 5.1052 | 0.3544 | 9.9011 |
| 15 | 0.5364 | 5.9747 | 0.3084 | 9.7801 |
| 16 | 0.5788 | 5.6932 | 0.3285 | 9.2417 |
| 17 | 0.7774 | 3.2826 | 0.2586 | 8.3161 |
| 18 | 0.7077 | 3.6486 | 0.2418 | 8.0681 |
| 19 | 0.7632 | 3.4191 | 0.2357 | 8.6627 |
| 20 | 0.7908 | 3.9171 | 0.2936 | 8.7981 |
| 21 | 0.5466 | 5.1813 | 0.3148 | 10.1072 |
| 22 | 0.5200 | 5.8211 | 0.3356 | 10.9604 |
| 23 | 0.5087 | 5.9252 | 0.3455 | 10.5875 |
| 24 | 0.5251 | 5.3537 | 0.3049 | 10.8014 |
| 25 | 0.3880 | 5.0661 | 0.3872 | 11.9188 |
| 26 | 0.3531 | 5.8144 | 0.3388 | 11.6666 |
| 27 | 0.3988 | 5.9820 | 0.3256 | 11.2051 |
| 28 | 0.3707 | 5.6731 | 0.3458 | 11.6117 |

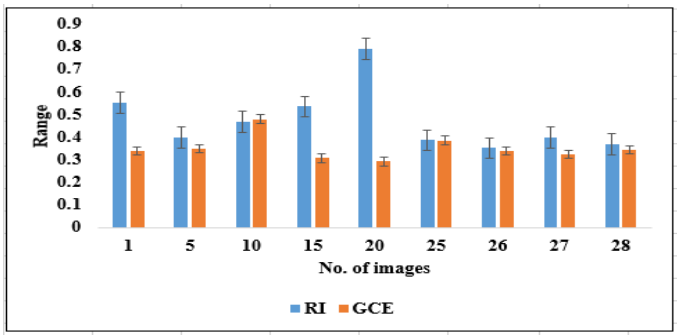



**Figure 6:** Graphical comparison of proposed model for various images

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

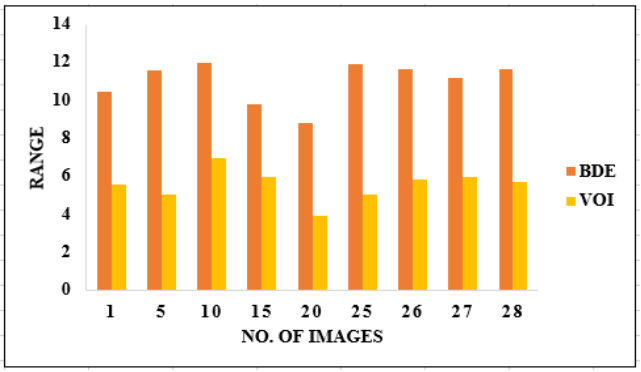



**Figure 7:** Graphical comparison of proposed model for various images

In above table 4, the different image performance is discussed for proposed model. In this experimentation analysis, the different images gives the different output results correspondingly. For various images such as 5 to 8, the model achieved nearly 0.33 to 0.39 of RI, nearly 0.31 to 0.39 of GCE for images 1 to 8. In the BDE analysis, the proposed model achieved nearly 8 to 11% for all images. From 1 to 28 images, the proposed model achieved 0.35 to 0.45 of GCE and achieved nearly 3 to 5% of VOI. Table 5 provides the experimental analysis of proposed model with existing techniques for first dataset. The existing techniques such as FCM [22, 27], RNN [39], CNN [41], GAN [42], YOLO [26, 27], CSO [31] and SSA [31] are all tested with these two datasets and results are mentioned in the following tables.

**Table 5. Results of competitive techniques for the BSDS500**

| Technique | RI | VOI | GCE | BDE |
|---|---|---|---|---|
| FCM | 0.8173 | 1.9702 | 0.2587 | 8.6165 |
| RNN | 0.8184 | 1.9679 | 0.2592 | 8.5876 |
| CNN | 0.8184 | 1.9686 | 0.2593 | 8.5914 |
| GAN | 0.7537 | 2.0523 | 0.2198 | 12.9771 |
| YOLO | 0.8276 | 1.8685 | 0.2223 | 8.9728 |
| YOLO-CSO | 0.8232 | 1.8743 | 0.2198 | 9.4888 |
| YOLO-SSA | 0.8296 | 1.8700 | 0.2264 | 9.0037 |
| YOLO- TC-SSA | **0.8361** | **1.8561** | **0.2077** | **8.3777** |

In above *table 5* represent that the results of competitive techniques for the BSDS500. In this comparative analysis, there are different technique were used as FCM, RNN, CNN, GAN, YOLO, YOLO-CSO, YOLO-SSA and YOLO-TC-SSA. Initially the FCM reaches the RI rate of 0.8173 and CNN achieved the GCE of 0.2593 respectively. YOLO network reaches the RI of 0.8276 and the VOI of 1.8685 respectively. Finally the YOLO- TC-SSA model reaches the RI of 0.8361, VOI of 1.8561, GCE of 0.2077 and BDE of 8.3777 respectively. Table 6 presents the comparative analysis of proposed model with existing techniques for second datasets. *Figure 8 and 9* shows the graphical comparison of two datasets using different techniques in terms of RI and VOI.

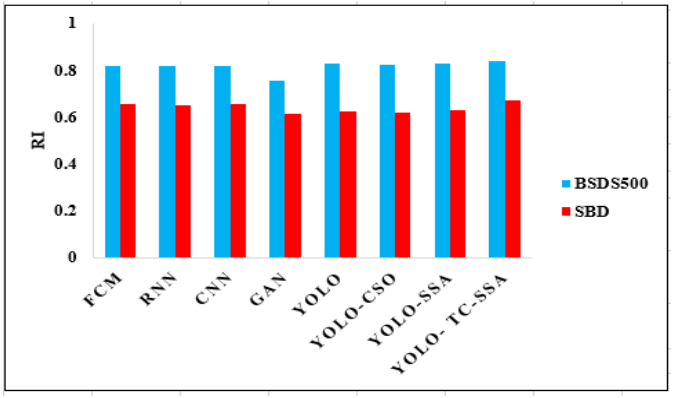



**Figure 8:** Comparison of proposed with existing techniques on two datasets in terms of RI

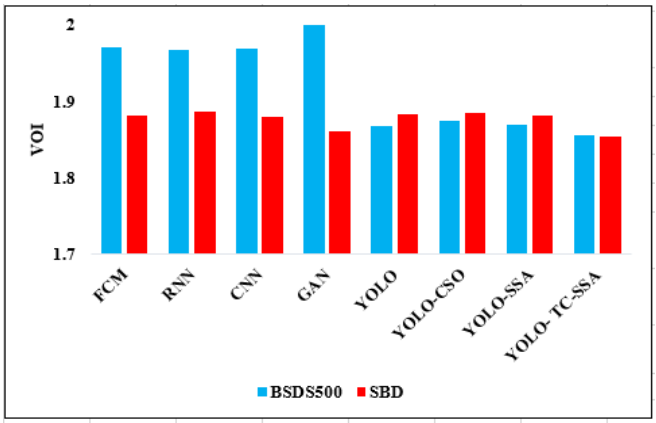



**Figure 9:** Comparison of proposed with existing techniques on two datasets in terms of VOI

**Table 6. Results of competitive techniques for the SBD**

| Technique | RI | VOI | GCE | BDE |
|---|---|---|---|---|
| FCM | 0.6540 | 1.8808 | 0.2447 | 15.6244 |
| RNN | 0.6517 | 1.8864 | 0.2443 | 15.8201 |
| CNN | 0.6545 | 1.8804 | **0.2049** | 15.6116 |
| GAN | 0.6142 | 1.8602 | 0.2068 | 17.7728 |
| YOLO | 0.6227 | 1.8825 | 0.2213 | 16.9126 |
| YOLO-CSO | 0.6179 | 1.8843 | 0.2188 | 17.1358 |
| YOLO-SSA | 0.6269 | 1.8819 | 0.2253 | 16.8736 |
| YOLO- TC-SSA | **0.6688** | **1.8541** | 0.2169 | **15.5535** |

In above *table 6*, the results of competitive techniques for the SBD is given. In this analysis, different technique were used for comparison that includes FCM, RNN, CNN, GAN, YOLO, YOLO-CSO, YOLO-SSA and YOLO-TC-SSA. FCM reaches the RI of 0.6540 and the VOI of 1.8808, GCE of 0.2447 and the BDE of 15.6244. Secondly the RNN reaches the RI of 0.6517, 1.8864 and VOI of 0.2443. YOLO achieved the RI of 0.6227

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

and VI of 1.8825. Finally the YOLO- TC-SSA model reaches the RI of 0.6688, VOI of 1.8541, GCE of 0.2169 and BDE of 15.5535 respectively. *Figure 10 and 11* provides the comparison of graphs for two publicly available dataset using proposed model in terms of GCE and BDE. *Figure 12* provides the ROC analysis of proposed model.

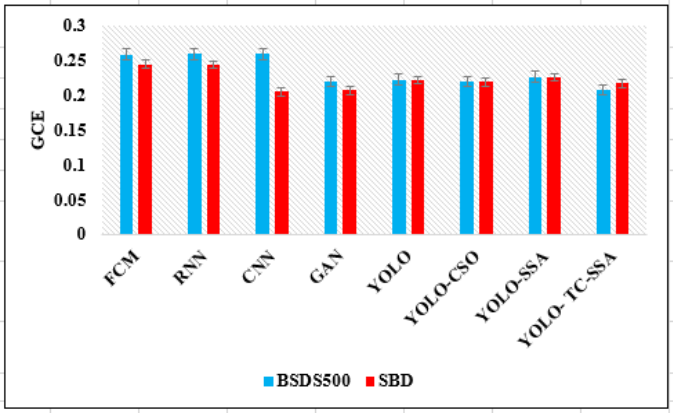



**Figure 10:** Comparison of proposed with existing techniques on two datasets in terms of GCE

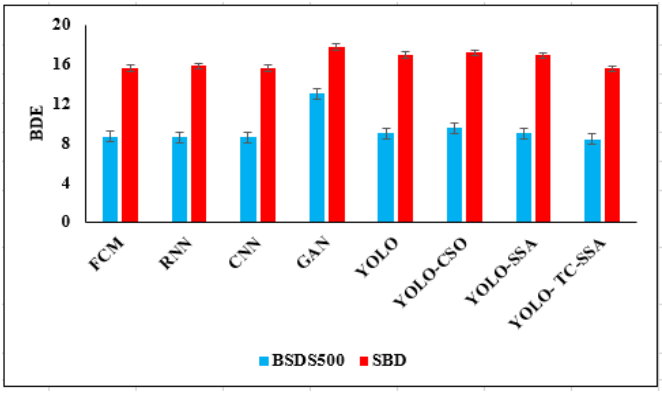



**Figure 11:** Comparison of proposed with existing techniques on two datasets in terms of BDE

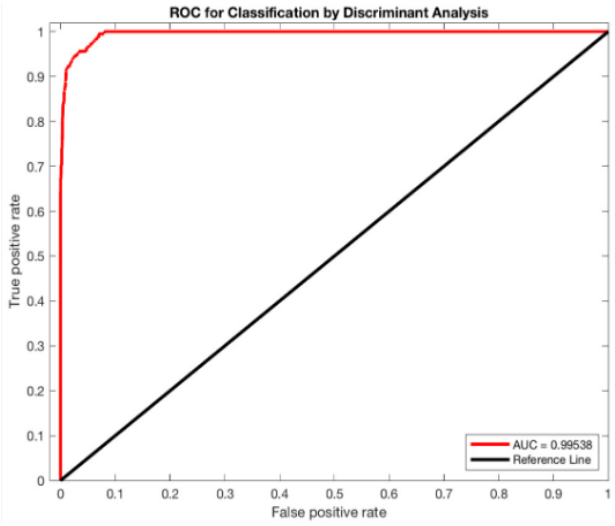



**Figure 12:** ROC Analysis

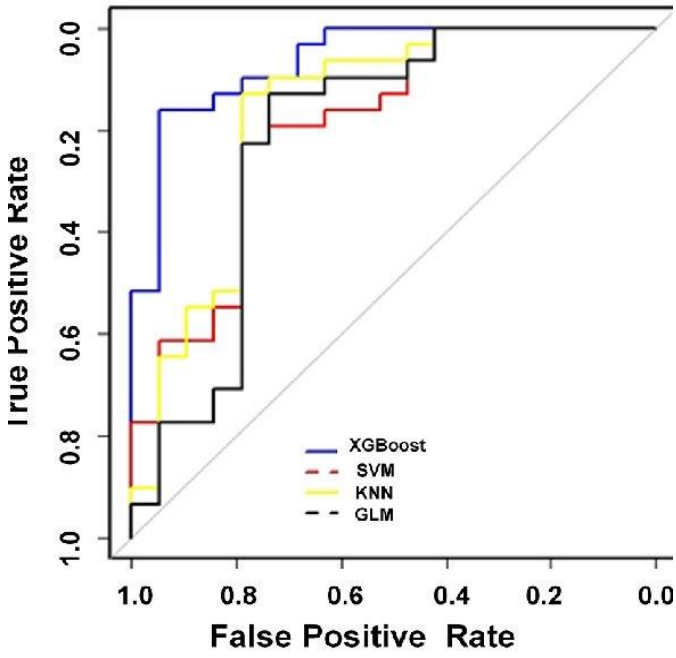



**Figure 13:** ROC Analysis with other algorithms

## 5. CONCLUSION

The improved version of YOLO network is introduced in this study for color image segmentation. To identify the objects in the scene, two publicly available datasets are used, where light-weight object detector is used in-depth for segmentation. In order to improve the region extraction technique's accuracy, the weight of the tiny YOLO network is optimized by TCSSA model. Testing showed that the suggested model is capable of segmenting input into multiple metrics using bounding boxes. In the experiment analysis, the proposed YOLO-TC-SSA model reaches the RI of 0.6688, VOI of 1.8541, GCE of 0.2169 and BDE of 15.5535 respectively. And also we evaluate the comparative analysis by using different techniques, but the proposed model reached the high performance than other technique.

## REFERENCES

[1] F. Yang, H. Fan, P. Chu, E. Blasch, H. Ling, Clustered object detection in aerial images, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8311–8320.

[2] M. Hashemzadeh, N. Farajzadeh, Combining keypoint-based and segment-based features for counting people in crowded scenes, Inform. Sci. 345 (2016) 199-216, http://dx.doi.org/10.1016/j.ins.2016.01.060, 2016/06/01/ 2016.

[3] M. Hashemzadeh, A. Zademehdi, Fire detection for video surveillance applications using ICA K-medoids-based color model and efficient spatio-temporal visual features, Expert Syst. Appl. 130 (2019) 60–78, http://dx.doi.org/10.1016/j.eswa.2019.04.019, 09/15/ 2019.

[4] M. Hashemzadeh, B. Adlpour Azar, Retinal blood vessel extraction employing effective image features and combination of supervised and unsupervised machine learning methods, Artif. Intell. Med. 95 (2019) 1–15, http://dx.doi.org/10.1016/j.artmed.2019.03.001, /04/01/ 2019.

[5] S. Wazarkar, B.N. Keshavamurthy, A survey on image data analysis through clustering techniques for real world applications, J. Vis. Commun. Image Represent. 55 (2018) 596–626, http://dx.doi.org/10.1016/j.jvcir.2018.07.009,/08/01/ 2018.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

[6] L. He, S. Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation, Neurocomputing 240 (2017) 152-174, http://dx.doi.org/10.1016/j.neucom.2017.02.040, /05/31/ 2017.

[7] S. Pare, A. Kumar, V. Bajaj, G.K. Singh, A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve, Appl. Soft Comput. 47 (2016) 76–102, http://dx.doi.org/10.1016/j.asoc.2016.05.040, /10/01/2016.

[8] L. Feng, H. Li, Y. Gao, Y. Zhang, A color image segmentation method based on region salient color and fuzzy C-means algorithm, Circuits Systems Signal Process. 39 (2) (2020) 586–610, http://dx.doi.org/10.1007/s00034-019-01126-w, 02/01 2020.

[9] Z. Zhou, X. Zhao, S. Zhu, K-harmonic means clustering algorithm using feature weighting for color image segmentation, Multimedia Tools Appl. 77 (12) (2018) 15139–15160, http://dx.doi.org/10.1007/s11042-017-5096-9.

[10] T.R. Farshi, J.H. Drake, E. Özcan, A multimodal particle swarm optimization- based approach for image segmentation, Expert Syst. Appl. 149 (2020) 113233, http://dx.doi.org/10.1016/j.eswa.2020.113233, 2020/07/01.

[11] F. Zhao, Z. Zeng, H. Liu, R. Lan, J. Fan, Semisupervised approach to surrogate-assisted multiobjective kernel intuitionistic fuzzy clustering algorithm for color image segmentation, IEEE Trans. Fuzzy Syst. 28 (6) (2020) 1023–1034, http://dx.doi.org/10.1109/TFUZZ.2020.2973121.

[12] A.G. Oskouei, M. Hashemzadeh, B. Asheghi, M.-A. Balafar, CGFFCM: CLuster-weight and group-local feature-weight learning in fuzzy C-means clustering algorithm for color image segmentation, Appl. Soft Comput. (2021) 108005, http://dx.doi.org/10.1016/j.asoc.2021.108005, 2021/10/26.

[13] Deepa, S.N., Rasi, D., Global biotic cross-pollination algorithm enhanced with evolutionary strategies for color image segmentation. Soft Computing A Fusion of Foundations, Methodologies and Applications, Springer 23(8), 2545–2559 (2019)

[14] Deepa, S.N., Rasi, D., Optimized deep learning neural network model for doubly fedinduction generator in wind energy conversion systems. Soft Computing A Fusion of Foundations, Methodologies and Applications, Springer 23(18), 8453-8470 (2019)

[15] Garcia-Lamont, F., Cervantes, J., López, A., Rodriguez, L.: Segmentation of images by color features: a survey. Neurocomputing 292(1), 1–27 (2018)

[16] Markchom, T., Lipikorn, R.: Thin cloud removal using local minimization and logarithm image transformation in HSI color space. In: The 4th International Conference on Front Signal Process, Sept, Poitiers, France. IEEE (2018)

[17] Okur, E.; Turkan, M. A survey on automated melanoma detection. Eng. Appl. Artif. Intell. 2018, 73, 50–67.

[18] Kang, C., Wu, C. and Fan, J., 2021. Entropy-based circular histogram thresholding for color image segmentation. Signal, Image and Video Processing, 15(1), pp.129-138.

[19] Sathya, P.D., Kalyani, R. and Sakthivel, V.P., 2021. Color image segmentation using Kapur, Otsu and minimum cross entropy functions based on exchange market algorithm. Expert Systems with Applications, 172, p.114636.

[20] Rasi, D., Deepa, S.N. Hybrid optimization enabled deep learning model for colour image segmentation and classification. Neural Comput & Applic 34, pp.21335–21352 (2022).

[21] Wang, S., Sun, K., Zhang, W. and Jia, H., 2021. Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation. Math. Biosci. Eng, 18(4), pp.3092-3143.

[22] Wei, T., Wang, X., Li, X. and Zhu, S., 2022. Fuzzy subspace clustering noisy image segmentation algorithm with adaptive local variance & non-local information and mean membership linking. Engineering Applications of Artificial Intelligence, 110, p.104672.

[23] Wu, C. and Zhang, X., 2022. Total Bregman divergence-driven possibilistic fuzzy clustering with kernel metric and local information for grayscale image segmentation. Pattern Recognition, 128, p.108686.

[24] Takahashi, M., Ji, Y., Umeda, K. and Moro, A., 2020, December. Expandable YOLO: 3D object detection from RGB-D images. In 2020 21st International Conference on Research and Education in Mechatronics (REM) (pp. 1-5). IEEE.

[25] Xianbao, C., Guihua, Q., Yu, J. and Zhaomin, Z., 2021. An improved small object detection method based on Yolo V3. Pattern Analysis and Applications, 24(3), pp.1347-1355.

[26] Magalhães SA, Castro L, Moreira G, Dos Santos FN, Cunha M, Dias J, Moreira AP. Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse. Sensors. 2021 May 20; 21(10):3569.

[27] Abas SM. A Yolo and convolutional neural network for the detection and classification of leukocytes in leukemia (Doctoral dissertation, Polytechnic University).

[28] Hurtik, P., Molek, V., Hula, J., Vajgl, M., Vlasanek, P. and Nejezchleba, T., 2022. Poly-YOLO: higher speed, more precise detection and instance segmentation for YOLOv3. Neural Computing and Applications, 34(10), pp.8275-8290.

[29] Su, Y., Liu, Q., Xie, W. and Hu, P., 2022. YOLO-LOGO: A transformer-based YOLO segmentation model for breast mass detection and segmentation in digital mammograms. Computer Methods and Programs in Biomedicine, p.106903.

[30] Jainulabudeen, A. and Surputheen, M.M., 2022. Novel Two-Level Randomized Sector-based Routing to Maintain Source Location Privacy in WSN for IoT. IJCSNS, 22(3), p.285.

[31] Vaiyapuri, T., Alaskar, H., Parvathi, R., Pattabiraman, V. and Hussain, A., 2022. Cat Swarm Optimization-Based Computer-Aided Diagnosis Model for Lung Cancer Classification in Computed Tomography Images. Applied Sciences, 12(11), p.5491.

[32] Zhang, H., Liu, T., Ye, X., Heidari, A.A., Liang, G., Chen, H. and Pan, Z., 2022. Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems. Engineering with Computers, pp.1-35.

[33] Yue, X., Wang, Q., He, L., Li, Y. and Tang, D., 2022. Research on Tiny Target Detection Technology of Fabric Defects Based on Improved YOLO. Applied Sciences, 12(13), p.6823.

[34] Hurtik, P., Molek, V., Hula, J., Vajgl, M., Vlasanek, P. and Nejezchleba, T., 2022. Poly-YOLO: higher speed, more precise detection and instance segmentation for YOLOv3. Neural Computing and Applications, 34(10), pp.8275-8290.

[35] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 530–549. doi:10.1109/TPAMI.2004.1273918.

[36] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in IEEE International Conference on Computer Vision, 2001, pp. 416–423.

[37] Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N. and Terzopoulos, D., 2021. Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence.

[38] Mohammdian-Khoshnoud, M., Soltanian, A.R., Dehghan, A. and Farhadian, M., 2022. Optimization of fuzzy c-means (FCM) clustering in cytology image segmentation using the gray wolf algorithm. BMC Molecular and Cell Biology, 23(1), pp.1-9.

[39] Sivapriya, G., Praveen, V., Gowri, P., Saranya, S., Sweetha, S. and Shekar, K., 2022. Segmentation of Hard exudates for the detection of Diabetic Retinopathy with RNN based sematic features using fundus images. Materials Today: Proceedings.

[40] Swetha S , S. Saranya and M. Devaraju, Skin Cancer Detection and Segmentation Using Convolutional Neural Network Models, Volume 10, Issue 4,2022 (pp. 984-987), IJEER.

[41] Moussaoui, H., Benslimane, M. and El Akkad, N., 2022. Image segmentation approach based on hybridization between K-means and mask R-CNN. In WITS 2020 (pp. 821-830). Springer, Singapore.

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 359-370 | e-ISSN: 2347-470X

[42] Chen, S., Zhou, Q. and Zou, H., 2022. A Novel Un-Supervised GAN for Fundus Image Enhancement with Classification Prior Loss. Electronics, 11(7), p.1000.

© 2023 by the D. Rasi, M. AntoBennet, P. N. Renjith, M. R. Arun and D. Vanathi. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).