Jellyfish Chicken Swarm Optimizer based Convolutional Neural Network with transfer learning for Recognizing Text in Complex Images

[1]Thuraka Gnana Prakash, [2] B. Sujatha,[3]L.Sumalatha
[1]Research scholar, Dept of CSE,JNTUK, Kakinada, Eastgodavarti, AP.INDIA.ptlprakash@gmail.com
[2] Professor, Dept of CSE, Godavari Institute of Engineering and Technology, Rajahmundry, Eastgodavari AP, INDIA. birudusujatha@gmail.com
[3] Professor, Dept of CSE, UCOE, JNTUK, Kakinada, Eastgodavarti, AP.INDIA.sumalatha.lingamgunta@gmail.com

KEYWORDS
Differentiable Binarization Net++ (DBNet++), Fuzzy Local Information C-Means (FLICM), LeNet-5, Jellyfish Search Optimizer (JSO), Chicken Swarm Optimization (CSO).

ABSTRACT:
Text acquired in images contains diverse information and thus it is widely utilized in several applications for understanding image scenarios. Also, it is employed for retrieving visual information. Semantic information in complex image is highly important for the humans to recognize the complete environment. Even though, text present in images reveal the flexible form in unconstrained state that makes identification of text as well as character recognition process as very challenging task. Here, Jellyfish Chicken Swarm Optimizer based Convolutional Neural Network with transfer learning (JCSO_CNN with TL) is presented for text recognition in complex images. The input image obtained from database performs text foreground extraction employing GrabCut method. Thereafter, text detection is carried out by Differentiable Binarization Net++ (DBNet++). Afterwards, character segmentation is accomplished utilizingFuzzy Local Information C-Means (FLICM). Finally, character recognition is conducted using CNN with TL in which CNN is employed with hyperparameters from the trained models namely LeNet-5. Moreover, CNN with TL is tuned by designed JCSO that is formed by incorporating Jellyfish Search Optimizer (JSO) with Chicken Swarm Optimization (CSO). Furthermore, JCSO_CNN with TL attained 90.8% of precision, 94.7% of recall and 92.7% of f-measure.

1. Introduction

The visual scene understanding is a vital probe region for computer vision communities. It requires vast research in an area of computer vision as well as its subareas [2]. The images are detected as easily obtainable and extremely valuable media that performs enormous quantity of information. The pixels and helpful information comprised in an image are extracted with regard to requirements of application by the computer vision [9]. Text acquired in images and videos has accurate semantic information. Likewise, skill concerning text present in images is employed by several image cognition as well as understanding applications such as video or image retrieval, book digitization, multilingual translation and so on [10] [7]. The recognition of text has a vital part, particularly for the people to read or else to know about a text present in building name, objects and so forth [26]. Optical character recognition (OCR) specifies to utilization of machine for converting manuscripts or printed texts in image to the format that is capable for computers to process directly. OCR engages text recognition on orientation, geometric, location as well as symmetric information like circular, vertical, elliptic and horizontal symmetry in the detection. a the significant branch of computer vision, general OCR application is to process the information input utilizing image text recognition [4]. With a growth of OCR system, people are fulfilled with text recognition in a book or document and concentrates on recognition of text in normal scene termed Scene Text Recognition (STR) [8].

An appearance in the natural scenes show more alterations in scripts, fonts as well as scale that includes complex and curved shapes, other than traditional scanned images. Additionally, text forms in scenes are referred to incidental naturally. The text content recognition in scene images need a robustness feature extraction method for capturing appearance of text in every form and extracted features encoding in sequential manner for further assessment to allocate character labels [6]. Text recognition is the

challenging task in machine learning (ML) and computer vision that tries to design computer application to read the texts from images in automatic manner. It is also useful in automated recognition of the traffic signs, license plates and independent robot navigation [3]. In present state, an application of an automated text extraction procedure pursued by the text recognition is gaining enormous demand. Recently, various probe processes are performed for extracting text from the scanned document images and this is detected as a segment of OCR [11] [12]. Moreover, OCR is appropriate only for texts that have clear backgrounds as it is not capable to extract texts from the complex backgrounds [13]. Owing to this, newer text extraction techniques that execute better on complicated background images are growing currently [7]. Furthermore, image is termed as complex image while one amongst the subsequentparameters is existed in an image. The factors include text is consisted on graphical background, text as well as background are same and an image captured includes surroundings or environments [26].

Text recognition comprises of two portions namely identification of text and recognizing text. The detection of text is a procedure of positioning text areas in image. It is performed by techniques namely connected component assessment, sliding windows, edge detection and so on. The text recognition refers to aprocedurefor identifying individual word or character in an individual text area. It is carried out by approaches like neural networks (NNs), feature extraction, template matching and so on [4]. The tremendous amount of conventional methods includes researches dealing with recognizing texts. Many techniques are concentrated on digitized paper-enabled documents or document images [14]. Though, reading a text in an image is non-trivial chore as the images are vastly changeable in layout of characters, appearance and a subject to diverse degradations [3]. With advancement in deep learning (DL) probe, several current tasks on text recognition have been utilized deep neural network (DNN)-based techniques to resolve the problems faced during text recognition [15] [16] [6]. During past decades, many current probes generally stem from DL structures and gained better outcomes in recognizing texts [17] [18]. These structures have numerous layers of depictions and attributes to be learned must be incorporated to feature extraction approach and then utilized for classifying objects. The CNN is proven as most successful technique in various computer vision chores. However, in spite of vital progression, text recognition remains as the challenging chore in the computer vision probe owing to increasing alterations in appearances of text viewed in random orientations, fancy font styles, curved shapes, size alterations and so forth [6].

The leading objective is to introduce JCSO_CNN with TL for recognizing text in complex images.Understanding of visual scene includes text as well as image processing. It is a very hard chore to recognize the scenes and read a text written in an image. There is an important necessity to design approaches for extracting as well as recognizing text in the images. Here, input image obtained to carry out text recognition in complex images is obtained from particular dataset. Then, text foreground extraction is carried outbyGrabCuttechnique for separating foreground text from complex background. Afterwards, text detection is done in text foreground extracted image by DBNet++. Thereafter, character segmentation is accomplished, wherein characters are segmented employing FLICM. Lastly, CNN with TL is employed for character recognition. However, CNN is used with hyperparameters from LeNet-5 andCNN with TL is tuned employing JCSO that is devised by incorporating JSO and CSO.

- **ProposedJCSO_CNN with TL for recognizing text in complex images:**Recognition of text process is recently utilized to gain a better understanding about image contents that are very helpful in vast range of real-time applications. In this work, character recognition is accomplished by CNN with TL. Moreover, CNN with TL is tuned utilizing JCSO that is an amalgamation of JSO and CSO.

The arrangement of sections is: Existing techniques are indicated in section 2, methodology ofJCSO_CNN with TL is revealed in section 3, JCSO_CNN with TL outcomes are explained in section 4 whereas section 5 explains conclusion of JCSO_CNN with TL.

2. Motivation

The text recognition from an image requires unique features as a character consist in image may vary in color, aspect ratio, size, orientation, shape, writing style, image quality owing to diverse lighting states as well as blurred and complicated background, which are considered as several challenges. Considering these challenges, a new scheme is developed for recognizing text in complex images. In this portion, reviewed methods in accordance to text recognition in complex images and its challenges are interpreted.

2.1 Literature Survey

Wu, Y.,*et al.* [1] developed context-Aware and embedded text detector (CE-text) for recognizing text in natural view images. This method not only acquired precise detection of text, but also ran with faster performance in the embedded systems, though it was led to less storage space and low performance.Kantipudi, M.V.V.,*et al.* [2] presented Bidirectional Long-Short-Term Memory (BiLSTM)+CNN for text recognition. It was capable to detect a text from diverse backgrounds, blurred images, indistinct text, several orientation and various font sizes. However, it detected a text partially in the images that were captured from longer distances or diverse image orientations.Harizi, R.,*et al*. [3] designed CNN for text reading. It obtained much lesser computational complexity, but it failed to locate characters in the natural images.Liu, Y.,*et al*. [4] devised Convolutional Recurrent Neural Network (CRNN) to recognize text that was highly robust to diverse scales and could manage challenging text detection conditions. Though, it utilized only two measures to assess the model that was not enough.

Lin, Q.,*et al*. [5] introduced Sequential Transformation Attention-based Network (STAN) for recognizing text. This approach was more flexible with respect to rectifying scene images with uneven text in them. It failed to recognize text under specific states like noisy backgrounds, severe curved text and low resolution images.Hassan, E. and VL, L.,[6] designed deep encoder–decoder network to recognize text. It was highly capable to capture broad spatial text content and generated best outcomes even with lesser aggregation of instance images. Though, it did not incorporate text rectification scheme for addressing complex criterions of the irregular text appearances.Pandey, D.,*et al.* [7] devised weighted Naive Bayes Classifier (WNBC) for recognizing the character from scene images. An extraction of text was performed accurately, but still characters might get lost or same character could be extracted many times. Luan, X.,*et al*. [8] developed Lightweight Scene Text Recognition (LSTR) for text recognition in scene images. It maintained smaller count of parameters and quick inference speed, but had high computational complexity.

2.2 Challenges

The limitations experienced by classical methods based upon text recognition in complex images are elucidatedbelow.

- CE-text designed in [1] for text recognition performed better on both embedded systems and workstations that proved the robustness and efficacy for text localization chore. Even though, the performance was sometimes dropped and special consideration was required to keep a balance among running time performance and storage sizes.
- In [2], Bi-LSTM+CNN presented for text recognition was proven as most promising approach to recognize text in images, but still it failed to deal with heavier background variations and diverse three-dimensional reflections.
- To recognize text in scene images, CNN developed in [3] failed to extend training set with the scene character images having several orientations as well as realistic distortions for enhancing recognition performance.
- The model in [4] was capable to recognize and segment text efficiently in several orientations and backgrounds, though it did not utilize knowledge distillation for compressing a model by decreasing count of parameters to enhance the model.
- Despite considerable development, the current STR techniques are inadequate, owing to performance necessities and complicated design phases. Also, most of the conventional techniques strongly depend on efficacy of attention mechanism in decoding phase.

3. ProposedJCSO_CNN with TL for recognizing text in complex images

Recognizing text in images has been an effective probe area in pattern recognition and computer vision. Text recognition remains to be challenging due to various factors like several fonts, defective imaging states and complicated backgrounds. Here,JCSO_CNN with TL is designed for recognizing text in complex images. Initially, considered input image performs text foreground extraction utilizingGrabcut technique. Thereafter, text detection is accomplished in extracted image by DBNet++. Then, character segmentation is executed in detected image employing FLICM. At last, character recognition is done by CNN with TL, where CNN is used with hyperparameters from LeNet-5. The tuning of CNN with TL is executed by JCSO that is combined form of JSO with CSO. Figure 1mentions diagrammatical depiction of JCSO_CNN with TL for recognizing text in complex images.

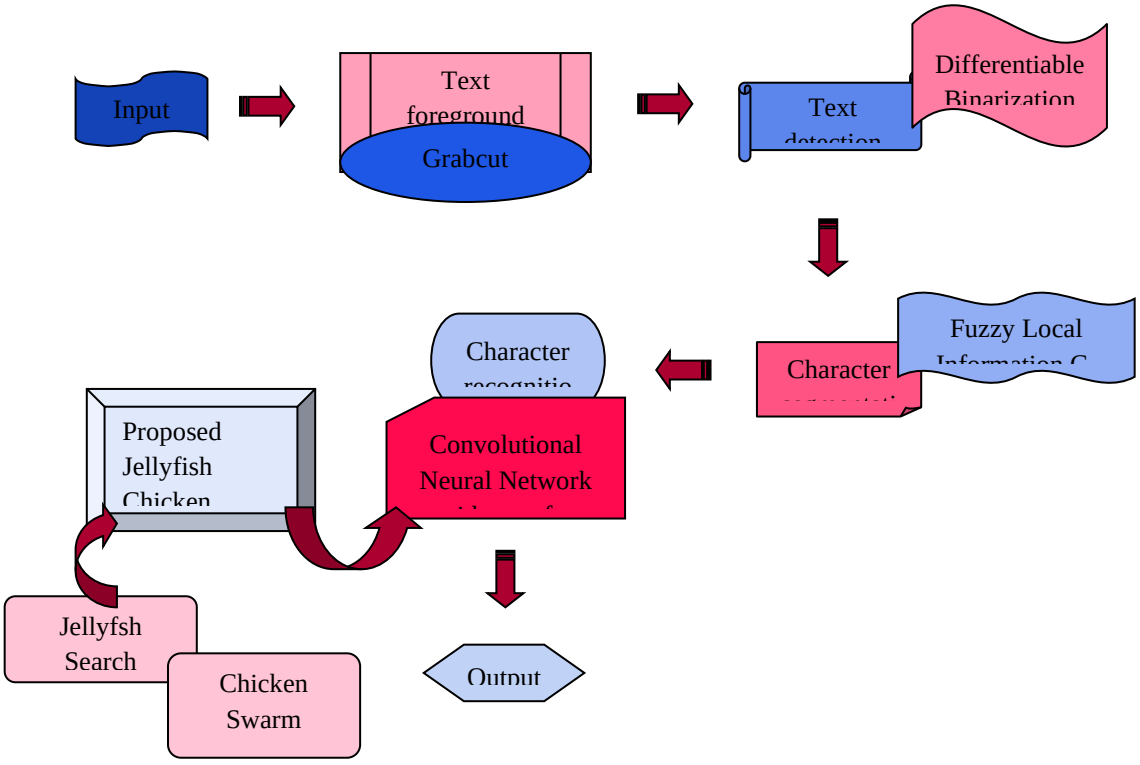


Figure 1. Diagrammatic depiction of JCSO_CNN with TL for recognizing text in complex images

3.1 Acquisition of input image

To recognize text in complex images, an input image is taken from dataset specified in [25], which is formulated as,

$$I = \{I_1, I_2, ...I_r ...I_o\} (1)$$

Where, r^{th} input image in database I is represented by I_r whereas I_o symbolizesoverall images consisted in database.

3.2Text foreground extraction utilizing GrabCut algorithm

Foreground extraction is an approach that permits to extract foreground from images for following processes. Designing a general model to extract foreground text information form complex image remains the challenging issue owing to its higher complexity and unpredictability. Here, text foreground extraction is done by GrabCut algorithm considering I_r as input.

GrabCut algorithm [27] refers to an iterative image segmentation approach on basis of Graph Cut method. This algorithm widens GraphCut to the color images as well as to imperfect trimaps. An user interaction is simplified for drawing the rectangle across desirable foreground, pursued by less quantity of the corrective editing. An incorporation of colour details in GraphCut approach and iterative learning technique increases the robustness. Hence, GrabCut is highly advantageous for extracting text foreground. The GrabCut algorithm steps are explicated beneath.

Step 1:The user develops initial trimaps by choosing rectangle. The pixels within rectangle are then marked as unidentified. The pixels exterior to rectangle is specified as recognized background.

Step 2:The computer generates initial image segmentation, wherein every pixels are placed uncertainly in foreground class as well as every recognized background pixels is positioned in background class.

Step 3:Gaussian Mixture Models (GMMs) are developed for the initial background and foreground classes.

Step 4: Individual pixel in foreground classes are allocated to most probably Gaussian element in foreground GMM. Likewise, individual pixel in background class is allocated to most certainly background Gaussian element.

Step 5: GMM is thrown out and newer GMM is learned from pixel groups produced in prior group.

Step 6:A graph is constructed and GraphCut is processed to identify newer tentative background as well as foreground pixel classification.

Step 7: Aforementioned steps from 4 to 6 are continued up to classification is met.

The text foreground extracted image using GrabCut algorithm is illustrated as G_r that is fed as input to carry out text detection.

3.3 Text detection utilizing DBNet++

Text detection is an essential chore for consistent performance of the character recognition. Though, detection of text in an image has various complexities as it does not offer homogeneous state such as brightness. Also, these states cause several issues like shadow and light, complex background as well as reflection in accordance to medium quality, whereupon characters are positioned, weakening accuracy of text detection.Here, text detection is conducted utilizing DBNet++ with consideration of G_r as input.

3.3.1 Structure of DBNet++

DBNet++ [19] module incorporates a most significant phase in post-processing operation termed binarization process into segmentation network. It executes more precisely due to uncomplicated and effective DB algorithm.

Initially, an input image is subjected to feature-pyramid back-bone. Then, pyramid features are up-sampled to similar scale and fed to an Adaptive Scale Fusion (ASF) part to generate context feature X. Afterwards, feature X is utilized for predicting probability map (A) and threshold map (H). Thereafter, an approximate binary map (Y) is computed by A and X. In tuning stage, supervision is applied on A, H and Y, wherein A and Y shares similar supervision. In an inference stage, bounding boxes are acquired effortlessly from Y or A utilizing box formation procedure.

(i) ASF

ASF is devised to combine the features of various scales dynamically.The features of diverse scales are scaled to equal resolution previously passing to ASF part. Consider, input feature maps comprises of O feature maps $Z \in \mathbb{R}^{O\times \mathrm{I}\times \mathrm{M}\times \mathrm{K}} = \{Z_j\}_{j=0}^{O-1}$, where O is fixed to 4. Initially, scaled input features Z are concatenated and thereafter, 3×3 convolutional (conv) layer is pursued to acquire the intermediary feature $S \in \mathbb{R}^{\mathrm{I}\times \mathrm{M}\times \mathrm{K}}$. Next, attention weights $\mathcal{N} \in \mathbb{R}^{O\times \mathrm{M}\times \mathrm{K}}$ can be computed through applying the spatial attention module to a feature S. Next, attention weights $\mathcal{N}$ is divided into O segments through channel size as well

as weighted multiply with related scaled feature to obtained combined feature $Z \in \mathbb{R}^{O\times I\times M\times K}$. In this manner, a scale attention can be modelled by,

$$\left.\begin{aligned} S &= Conv\left(concat\left(\left[Z_0, Z_1, ..., Z_{O-1}\right]\right)\right) \\ N &= Spatial\_Attention\left(S\right) \\ X &= concat\left(\left[E_0 Z_0, E_1 Z_1, ..., E_{O-1} Z_{O-1}\right]\right) \end{aligned}\right\} \quad (2)$$

Here, $concat$ represents concatenation operator and $Spatial\_Attention$ signifies spatial attention part whereas $Conv$ implies 3×3 conv operator.

(ii) Binarization

The binarization process is of two type namely standard binarization and differentiable binarization.

(a) Standard binarization: For a given probability map $A \in \mathbb{R}^{M\times K}$ generated by segmentation network, wherein M and K represents height as well as width of map and it is significant to convert this to a binary map $Y \in \mathbb{R}^{M\times K}$, where the pixels having value 1 are concerned as true text regions. Generally, binarization procedure is computed by,

$$Y_{j,l} = \begin{cases} 1 & if\ A_{j,l} \geq n, \\ 0 & otherwise \end{cases} \quad (3)$$

Here, n denotes pre-defined threshold whereas (j,l) implies coordinate points in a map.

(b) Differntiable binarization: A standard binarization formulated in Eq. (3) is not differentiable. Hence, this binaization cannot be optimized together with a segmentation network in tuning stage. To resolve this issue, binarization is performed with approximate step operation as follows.

$$\hat{Y}_{j,l} = \frac{1}{1+e^{k\left(A_{j,l}-H_{j,l}\right)}} \quad (4)$$

Here, $\hat{Y}$ indicates approximate binary map, k reveals amplifying factor and H denotes adaptive threshold map that is learned from a network. A differentiable binarization having adaptive thresholds cannot only assist discriminate text areas, but also separates text instance that are combined closely.

(iii) Box formation

In an inference stage, approximate binary map or else probability map is utilized for generating text bounding boxes that generates about similar outcomes. This comprises of three steps:

Step 1:Firstly, probability map or an approximate binary map is binarized with constant threshold for obtaining binary map

Step 2: The associated areas or shrunk text areas are attained from a binary map.

Step 3: Shrunk areas are dilated with an offset D', aVatti clipping approach and it is computed by,

$$D' = \frac{A'\times r'}{L'} \quad (5)$$

Here, A' indicates area of shrunk polygon, r' is fixed to 1.5 and L' specifies perimeter of shrunk polygon.

The text detected output utilizing DBNet++ is implied by B_r and model of DBNet++ is delineated in figure 2.

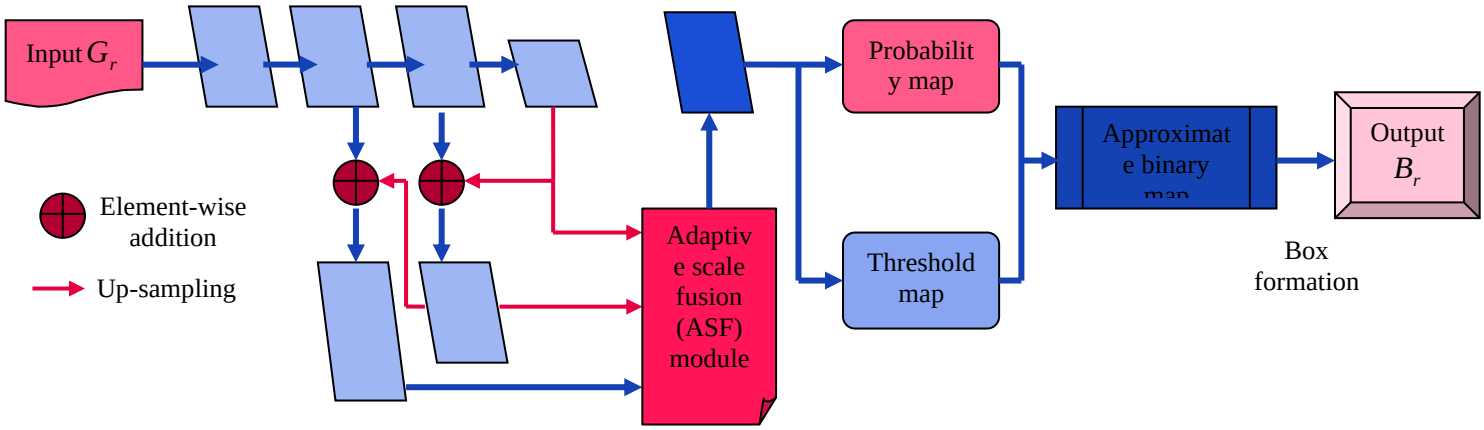


Figure 2. Structure of DBNet++

3.4 Character segmentation using FLICM

After every text is detected, the character segmentation is utilized to further examine each of a word into individual character. Here,text detected output B_r is considered as input to perform character segmentation using FLICM.

FLICM [28] integrates local spatial as well as gray-level information to its objective function that can be formulated as,

$$F_r = Z_g = \sum_{z=1}^{J}\sum_{m=1}^{x}\left[s_{mz}^{g}\left\|k_z - \alpha_m\right\|^2 + \mathrm{E}_{mz}\right] \tag{6}$$

The required criterions for F_r to be at local minimum extreme regarding s_{mz} and α_m can be illustrated by.

$$s_{mz} = \frac{1}{\sum_{w=1}^{x}\left(\frac{\left\|k_z - \alpha_m\right\|^2 + \mathrm{E}_{mz}}{\left\|k_z - \alpha_w\right\|^2 + \mathrm{E}_{wz}}\right)^{1/g-1}} \tag{7}$$

$$\alpha_m = \frac{\sum_{z=1}^{J} s_{mz}^{g} k_z}{\sum_{z=1}^{J} s_{mz}^{g}} \tag{8}$$

Here, m indicates reference cluster whereas w^{th} pixel belonging to group of neighbours, which falls into the window across z^{th} pixel (J_z), s_m denotes degree of membership, g implies weighting exponent of individual fuzzy membership and α_m symbolizes prototype of centre of a cluster m.

Hence, FLICM approach steps are elucidated beneath.

Step 1: Set a number x of cluster prototypes, stopping criterion η and fuzzification parameter g.

Step 2: Arbitrarily initialize fuzzy partition matrix.

Step 3: Fix loop counter $\ell = 0$.

Step 4: Compute cluster prototypes by Eq. (8).

Step 5: Evaluate membership values by Eq. (7).

Step 6: $\max\left\{O^{(\ell)} - O^{(\ell+1)}\right\} < \eta$ thereafter stop, or else, fix $\ell = \ell + 1$ and return to step 4.

The character segmented output utilizing FLICM is symbolized as F_r, which is fed to next phase to perform character recognition.

3.5 Character recognition utilizing CNN with TL

Character recognition refers to machine simulation of the human reading that is also called optical character recognition. The character recognition is utilized for recognizing characters in a document comprising handwritten as well as machine printed text, videos, graphics and so on. It is then converted to digitized format into the machine readable. The character recognition is accomplishedby CNN with TL. Moreover CNN is used with hyperparameters from LeNet-5. However, CNN with TL is tuned by JCSO that is an assimilation of JSO with CSO.

A character segmented output F_r is passed to train LeNet-5. From training process of LeNet-5, the hyperparameters are fetched and subjected to CNN. By training CNN, considering character segmented output and fetched hyperparameters, recognized output is obtained. Figure 3 reveals process of CNN with TL. The structure of LeNet-5 and CNN are explained in beneath subsections.

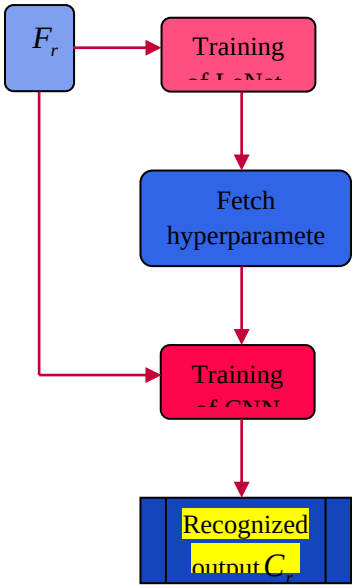


Figure 3. Process of CNN with TL

3.5.1 Architecture of LeNet-5

LeNet-5 [22] is the gradient-enabled learning CNN architecture and comprises of convolutional (conv) layers, pooling layers as well as Fully connected (FC) layer in addition to input layer and an output layer.

(i) Conv layer

The conv layer is mostly adopted for performing a process of feature extraction. An individual layer has count of conv kernels. An input matrix is convolved with conv kernel in conv layer. Consider an input matrix $Q = \left\{q_{b,p} | b = 1,2,\ldots,\mathrm{B},\ p = 1,2,\ldots,\mathrm{P}\right\}$.A conv kernel is symbolized by $K = \left\{\kappa_{u,v} | u = 0,1,\ldots,\mathrm{H}-1, v = 0,1,\ldots,\mathrm{H}-1\right\}$, wherein H signifies conv kernel dimension. The mathematical formulation of conv layer is expressed as,

$$\lambda_{b,p} = \left\{ f\left(\sum_{u=0}^{\mathrm{H}-1} \sum_{v=0}^{\mathrm{H}-1} \kappa_{u,v} q_{b+u,p+v} + \beta \right) \right\}_{b=1,2,\ldots,\mathrm{B},\ p=1,2,\ldots,\mathrm{P}} \tag{9}$$

Here, $\lambda_{b,p}$ specifies output from conv layer, β denotes offset term for an individual convolution whereas $f(.)$ implies activation function.

(ii) Activation function

Normally, Sigmoid, ReLU, $Tanh$, Gaussian and Softplus are most expansively utilized activation functions. $Tanh$, Gaussian and Sigmoid are referred to saturating non-linear functions. These functions are mostly selected as an activation function in conventional CNNs. The $Tanh$, Sigmoid and Gaussian activation functions are computed beneath.

$$f(q)=\frac{1}{1+e^{-q}} \tag{10}$$

$$f(q)=\frac{e^{q}-e^{-q}}{e^{q}+e^{-q}} \tag{11}$$

$$f(q)=-e^{-q^{2}} \tag{12}$$

Recently, unsaturated non-linearity operations are regularlyexploited activation functions in the CNN architectures. ReLU and Softplusactivations are generally utilized functions that are illustrated below.

$$f(q)=\max(0,q) \tag{13}$$

$$f(q)=\ln(1+e^{q}) \tag{14}$$

(iii) Pooling layer

Pooling layer intends to execute a feature selection procedure for decreasing image sizes when conserving the major attributes of image. Max-pooling, random pooling and mean-pooling are commonly utilized functions that extract all the points with high value, random and mean values. The pooling layer can be modeled as shown beneath.

$$\lambda_{v}^{i}=pool\left(\lambda_{v}^{i-1}\right) \tag{15}$$

Where, $pool(.)$ reveals maximal pooling function, λ_{v}^{i} implies i^{th} layer output and λ_{v}^{i-1} mentions previous layer output.

(iv) FC layer

FC layer is actually the ultimate layer, wherein individual neuron utilizes ReLU activation operation that is entirely connected to the neurons of former layer. FC layer can include local data that have the capacity to discriminate classes and neuron output is fed to an output layer. An output from FC layer is computed below.

$$\lambda_{v}^{i}=f\left(\kappa^{i}.\lambda_{v}^{i-1}+\beta^{i}\right) \tag{16}$$

Where, κ^{i} symbolizes conv kernel whereas β^{i} signifies offset term.

(v) Output layer

An output layer is also known as a softmax layer. It is mostly utilized in numerous classifications and recognition processes that maps FC layer output to $(0,1)$. An individual output relates to classification probability. At last, maximal probability classification is chosen as output and softmax function process can be evaluated by,

$$L_{r}=\lambda_{v}^{Y}=soft\max\left(\kappa^{Y}.\lambda_{v}^{Y-1}+\beta^{Y}\right) \tag{17}$$

The character recognized output from LeNet-5 is denoted as L_{r} and architecture of LeNet-5 is expounded in figure 4.

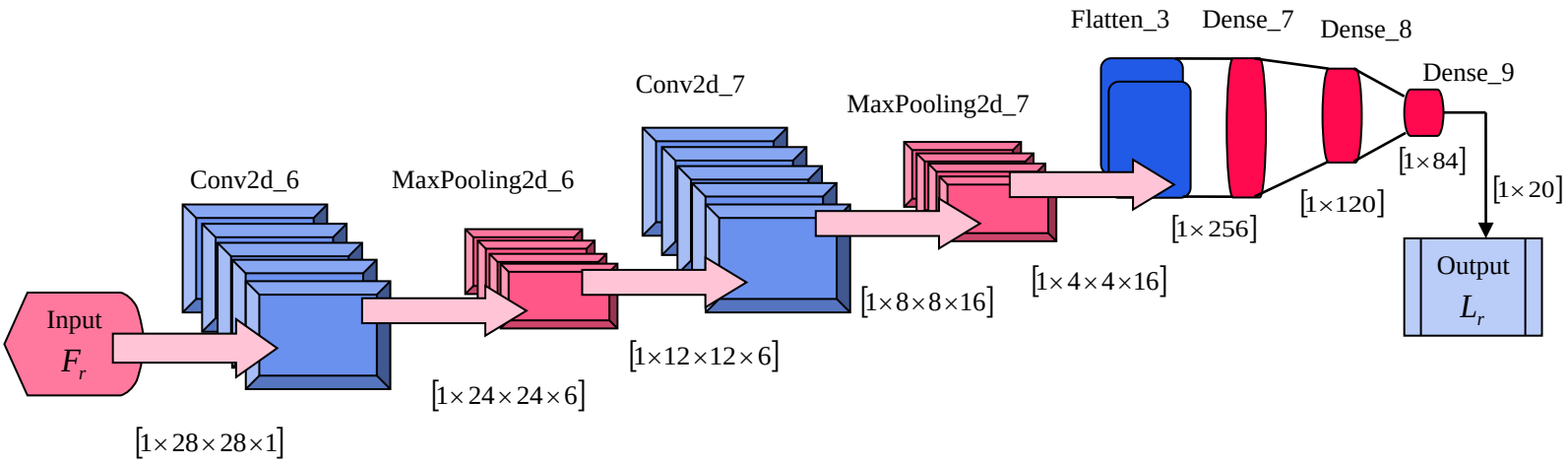


Figure 4. Structure of LeNet-5

3.5.2Architecture of CNN

CNN [20] [21] refers to the function for mapping an input image to output and it is normally comprises of input layer, conv layers, sum or max-pooling layer, activation layer such as sigmoid activation layer or Rectified Linear Unit (ReLU), FC layer and output layer.An input given to CNN is signified as P_r, such that,

$$P_r = \{\mathrm{N}_r, L_r\} \tag{18}$$

Here, N_r refers to fetched hyperparameters from LeNet-5.

(i) Input layer

An input layer is referred to layer with images that is subjected as the input to a network. An input layer may be colour or gray-scale image. Depending on a kind of input image, an input layer can have size $\zeta \times \omega \times \rho$, where ζ indicates width, ω represents height and ρ signifies depth of an image.

(ii) Conv layer

This layer is a building block of whole network as majority of computation work is carried out in conv layer. It comprises of group of parameters termed learnable filters. At this state, all filters are served as square matrix, wherein individual matrix is spatially tiny with respect to height and width, though it is able to extend itself across entire depth of known input volume.

An activation map offers the respond of filter at all spatial location. It is developed as an outcome of convolution of feature or filter over input image sizes. The complete image is convoluted by filters and it is specified as activation maps.

A layer obtains an input having dimension of $\zeta \times \rho \times \omega$ and utilizing two hyperparameters namely stride (d) and filter size (h), creates an input for other layer with sizes $\zeta 1 \times \omega 1 \times \rho 1$, wherein $\zeta 1$ and $\omega 1$ are formulated in beneath equations. The depth $\rho 1$ remains same like ρ. Also, it presents newer column and row of zeros on individual side of an image.

$$\zeta 1 = (\zeta - h + 2S)/d + 1 \tag{19}$$

$$\omega 1 = (\omega - h + 2S)/d + 1 \tag{20}$$

Here, S implies padding.

(iii) Pooling layer

This layer pursues conv layer in a network and its vital intention is to slowly lessen the spatial dimension of depiction. Additionally, it attempts to decrease computation quantity and network parameters, thus

handling overfitting task. Pooling layer operates independently over all input depth slices and resizes spatially. The most often utilized function is max-pooling and other filters namely L2 norm, averaging and min-pooling can also be utilized in this layer.

(iv) FC layer

The neurons present in FC layer have complete associations of all activation in prior layer as in ordinary NNs. The activations are calculated with matrix multiplication that is pursued by bias offset. Numerous FC layers can be present depending on application structure. The final comprises of neurons similar to count of classes in issue area.

The character recognized output from CNN can be illustrated by C_r and CNN structure is demonstrated in figure 5.

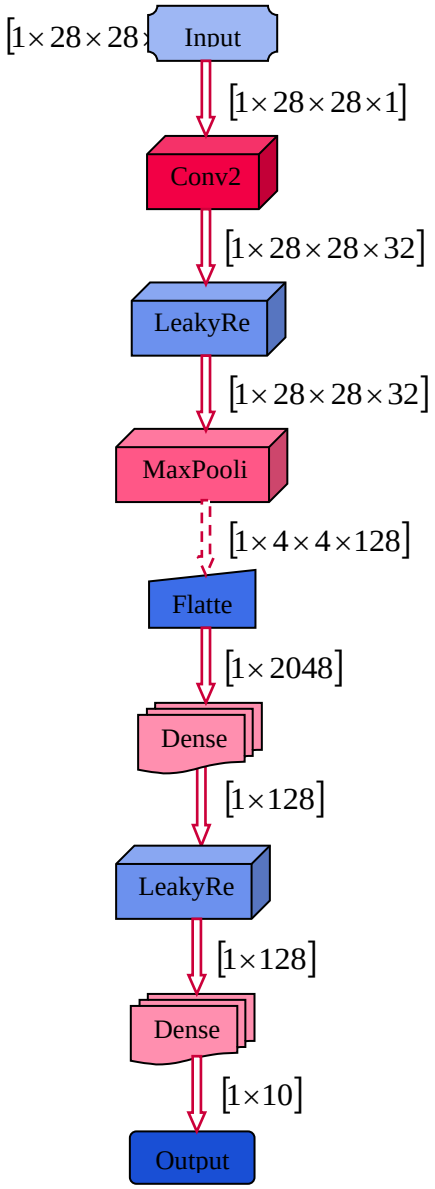


Figure 5. CNN structure

3.5.3 Training of CNN with TL by JCSO

JSO [23] is a bio-inspired metaheuristic approach that is designed on basis of food detecting characteristic of jellyfish on the sea. It has been revealed to be effective in resolving variousreal-time unconstrained and constrained applications. Also, this algorithm has better population utilization rate as well as it maintains good balance amongst local and global searches. CSO [24]is designed for optimization applications that mimic hierarchal structure of chicken swarm and its attributes. It has self-adaptive capability for solving optimization issues. Herein, JSO and CSO are merged to design JCSO that is highly acceptable to train CNN with TL owing to the advantages of JSO and CSO.

Jellyfish search solution encoding

CNN with ς learning parameteris trainedcontinuously in $\aleph$ search space for attaining excellent solution, in such a manner it can be given as $\aleph = [1 \times \varsigma]$.

Fitness function

Acomputation of fitness function is executed by identifyingvariationamid targetand CNN with TL output,which is represented as,

$$\Re = \frac{1}{o}\sum_{r=1}^{o}\left[\mathrm{A}_r - C_r\right]^2 \tag{21}$$

Here, o symbolizes overall imageswhereas C_r denotes output from CNN with TL and A_r signifiestargetoutput.

JCSOpursues below mentioned steps for obtaining good solution.

Step 1: Initializing solution

Firstly, population of JSO algorithm is normally initialized at random manner in $\aleph$ search space and it can be illustrated by,

$$T = \{T_1, T_2, ..., T_c, ..., T_e\} \tag{22}$$

Where, T_e signifiesoverall variables, T_c reveals c^{th} solution and T indicates population.

The chaotic map termed logistic map is utilized by JSO that provides more varied initial populations beside random selection and it outcomes with lesser premature convergence probability that is modeled by,

$$T_{c+1} = \vartheta T_c (1 - T_c), \quad 0 \le \mathcal{T}_c \le 1 \tag{23}$$

Here, c^{th} jellyfish location is indicated as T_{c+1}, logistic chaotic value of the c^{th} jellyfish location is implied by $\mathcal{T}_c$ whereas ϑ symbolizes parameter andit is equal to 4.

Step 2: Estimation of objective function

The objective function estimation is conducted by calculatingvariationbetweenCNN with TLas well as target outputs utilizing Eq. (21).

Step 3: Pursuing of the ocean currents

Ocean currentscontainmore quantity of nutrients and thus jellyfish iscaptivated against it. $\left(\vec{D}\right)$signifies ocean present direction that is mathematically expressed as follows.

$$\vec{D} = T^* - \vartheta \times n(0,1) \times \tau \tag{24}$$

Where, T^* signifiesbest location in swarm wherein jellyfish is currently present τ reveals mean location of all the jellyfish, rand is denoted by n and ϑ mentions distribution coefficient. Therefore, newer location of particular jellyfish isformulatedby,

$$T_c(\gamma + 1) = T_c(\gamma) + n(0,1) \times \vec{D} \tag{25}$$

$$T_c(\gamma+1)=T_c(\gamma)+n(0,1)\times T^*-\vartheta\times n(0,1)\times\tau \quad (26)$$

The standard equation of CSO can be interpreted by,

$$T_{c,a}^{\gamma+1}=T_{c,a}^{\gamma}+UV*\left(T_{\varepsilon,a}^{\gamma}-T_{c,a}^{\gamma}\right) \quad (27)$$

Let us consider,

$$T_{c,a}^{\gamma+1}=T_c(\gamma+1) \quad (28)$$

$$T_{c,a}^{\gamma}=T_c(\gamma) \quad (29)$$

$$T_{\varepsilon,a}^{\gamma}=T_{\varepsilon}(\gamma) \quad (30)$$

Applying above considerations in Eq. (27), the equation becomes,

$$T_c(\gamma+1)=T_c(\gamma)+UV*(T_{\varepsilon}(\gamma)-T_c(\gamma)) \quad (31)$$

$$T_c(\gamma+1)=T_c(\gamma)[1-UV]+UV*T_{\varepsilon}(\gamma) \quad (32)$$

$$T_c(\gamma)=\frac{T_c(\gamma+1)-UV*T_{\varepsilon}(\gamma)}{1-UV} \quad (33)$$

Substitute Eq. (33) in Eq. (26),

$$T_c(\gamma+1)=\frac{T_c(\gamma+1)-UV*T_{\varepsilon}(\gamma)}{1-UV}+n(0,1)\times T^*-\vartheta\times n(0,1)\times\tau \quad (34)$$

$$T_c(\gamma+1)-\frac{T_c(\gamma+1)}{1-UV}=\frac{\left(n(0,1)\times T^*-\vartheta\times n(0,1)\times\tau\right)(1-UV)-UV*T_{\varepsilon}(\gamma)}{1-UV} \quad (35)$$

$$\frac{(1-UV-1)T_c(\gamma+1)}{1-UV}=\frac{\left(n(0,1)\times T^*-\vartheta\times n(0,1)\times\tau\right)(1-UV)-UV*T_{\varepsilon}(\gamma)}{1-UV} \quad (36)$$

An update equation of JCSO can be manifested by,

$$T_c(\gamma+1)=\frac{UV*T_{\varepsilon}(\gamma)+\left(n(0,1)\times T^*-\vartheta\times n(0,1)\times\tau\right)(UV-1)-}{UV} \quad (37)$$

Where, $T_c(\gamma)$ represents jellyfish's current positionand $T_c(\gamma+1)$ symbolizesjellyfish's newer location, UV implies parameter that ranges among $[0,2]$, $T_{\varepsilon}(\gamma)$ denotes location of chick's mother $(\varepsilon\in[1,M])$ and τ specifies mean location.

Step 4: Swarm of the jellyfish

In a swarm, the jellyfish reveals active as well as passive kinds of moves. Initially, several jellyfish shows passive kind ofmovementwhilst swarm is formed currently. After the period of time, jellyfishsteadilyreveal active kind of movement.

In a passive kind ofmovement, jellyfish motion is over their ownpositions. The relevant updated position of every jellyfish is given as,

$$T_c(\gamma+1)=T_c(\gamma)+\xi\times n(0,1)\times(P-W) \quad (38)$$

Where, P and W signifies upper as well as lower bounds and ξ denotes motion coefficient.

For assuming active movement, jellyfish (y) other than the one amongstthem is chosenin random manner and vector from ajellyfish of interest specified as (c) to elected jellyfish (q) is employedto identify moving direction. The direction of motion and the updated place of jellyfish are mathematically expressed by,

$$\vec{E}=n(0,1)\times\vec{N} \quad (39)$$

$$\vec{N} = \begin{cases} T_y(\gamma) - T_c(\gamma), & if\ \Re(T_c) \geq \Re(T_y) \\ T_c(\gamma) - T_y(\gamma), & if\ \Re(T_c) < \Re(T_y) \end{cases} \quad (40)$$

Hence,

$$T_c(\gamma+1) = T_c(\gamma) + \vec{E} \quad (41)$$

Where, $\Re$ indicates objective function of the location T and E implies step whereas N symbolizes direction.

Step 5: Time control system

To adjust jellyfish movement amidsucceeding ocean currentsand moving ina jellyfish swarm, a time control systemcomprises time control operationspecified by $t(\gamma)$as well asthe constant indicated as t_0 . A time control operation is arbitrarily value that altersamongst 0 and 1. This mechanism is computed by,

$$t(\gamma) = \left| \left(1 - \frac{\gamma}{\not\gamma}\right) \times (2 \times n(0,1) - 1) \right| \quad (42)$$

Where, γ implies time that is elucidated by count of iteration and $\not\gamma$ symbolizesoverallitercount.

Step 6: Termination

JCSO attains finest solution by repetitively accomplishingaboveexplicatedphases. Algorithm 1 mentions pseudo code of JCSO.

Algorithm 1. Pseudo code of JCSO

SL. No	Pseudo code of JCSO
1	**Input:** $\not\gamma, P, W, \Re(T)$
2	**Output:** $T_c(\gamma+1)$
3	**Begin**
4	Logistic chaotic map is utilized to initialize population of jellyfish
5	Evaluate amount of food at individual T_c as $\Re(T_c)$
6	Find the jellyfish at location presently having abundant food (T^*)
7	***for*** $\gamma = 1$ to $\not\gamma$ ***do***
8	***for*** $c = 1$ to e ***do***
9	Estimatemechanism of time control $t(\gamma)$utilizing Eq. (42)
10	***if*** $t(\gamma) \geq t_0$
11	Jellyfish chases ocean currents
12	Find ocean current using Eq. (24)
13	Jellyfish newer location is calculatedutilizing Eq. (37)
14	***else***
15	Jellyfish move against a swarm
16	***if*** $n(0,1) > (1 - t(\gamma))$
17	Passive type movement is revealed by jellyfish utilizing Eq. (38)
18	***else***

19	Active type motion is revealed by jellyfish
20	Direction of jellyfish is determined by Eq. (40), Eq. (39) and Eq. (41)
21	***end if***
22	***end for***
23	Confirm whether boundary states are fulfilled and evaluate amount of at newer location
24	Estimate amount of food at newestlocation $\Re(T_c)$
25	***end for***
26	**Terminate**

4. Results and discussion

JCSO_CNN with TL designed for recognizing text in complex images acquired good outcomes and the results obtained by this method are revealed in this segment.

4.1 Experimentation setup

JCSO_CNN with TL devised to recognize text in complex images is conducted in PYTHON tool.

4.2 Description of dataset

ICDAR 2013 Dataset [25] concentrates on extraction of text contents from born-digital images like those utilized in online and e-mail. This dataset contains about 462 images that include 229 images in training set and 233 images intesting set.

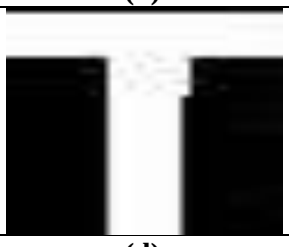
4.3 Experimentation outcomes

Figure 6 mentions experimentally acquired outcomes by JCSO_CNN with TL. Input image, foreground extracted image, text detected image, character segmented image-1 and character segmented image-2 are specified in figure 6 a), b), c), d), e).

(a)

(b)

(c)



(d)

(e)

Figure 6. Experimentation outcomes, a) Input image, b) Foreground extracted image, c) Text detected image, d) Character segmented image-1, e) Character segmented image-2

4.4 Performance measures

Precision, f-measure and recall are the performance measures taken into consideration to evaluateJCSO_CNN with TL for text recognition in complex images.

4.4.1 Precision

The ratio amongst true positive (TP) and total detections is called precision that can be expressed as mentioned beneath.

$$\mu = \frac{\hbar}{\hbar + \hat{\lambda}} \quad (43)$$

Where, $\hbar$ indicates TP and $\hat{\lambda}$ specifies false positive (FP).

4.4.2 Recall

A ratio among recognized actual text and overall TP is referred to recall, which is computed as follows.

$$\nu = \frac{\hbar}{\hbar + \varpi} \quad (44)$$

Here, ϖ represents false negative (FN).

4.4.3 F-measure

This is the measure applied to determine an effectiveness of model, wherein recall and precision metrics are considered to estimate f-measure. F-measure can be calculated utilizing beneath equation.

$$\chi = 2 \times \frac{\mu \times \nu}{\mu + \nu} \quad (45)$$

Where, μ and ν implies precision and recall.

4.5 Performance evaluation

The evaluation to prove the performance of JCSO_CNN with TL is conducted by altering training data and K fold.

4.5.1 Assessment regarding training data

Assessment of JCSO_CNN with TL with consideration of measures by changing training data for different iterations is illustrated in figure 7. For training data is 90%, the values obtained by JCSO_CNN with TL are elucidated. Figure 7 a) delineates evaluation of JCSO_CNN with TL based on precision. Precision achieved by JCSO_CNN with TL is 0.826, 0.837, 0.848, 0.858 and 0.898 with iteration=10, 20, 30, 40 and 50. Estimation of JCSO_CNN with TL regarding recall is shown in figure 7 b). JCSO_CNN with TL acquired recall of0.887 with iteration=10, 0.899 with iteration=20, 0.908 with iteration=30, 0.918 with iteration=40 and 0.937 with iteration=50. Figure 7 c) demonstrates analysis of JCSO_CNN with TL considering f-measure. The f-measure value acquired by JCSO_CNN with TL with iteration=10, 20, 30, 40 and 50 are0.855, 0.867, 0.877, 0.887 and 0.917.

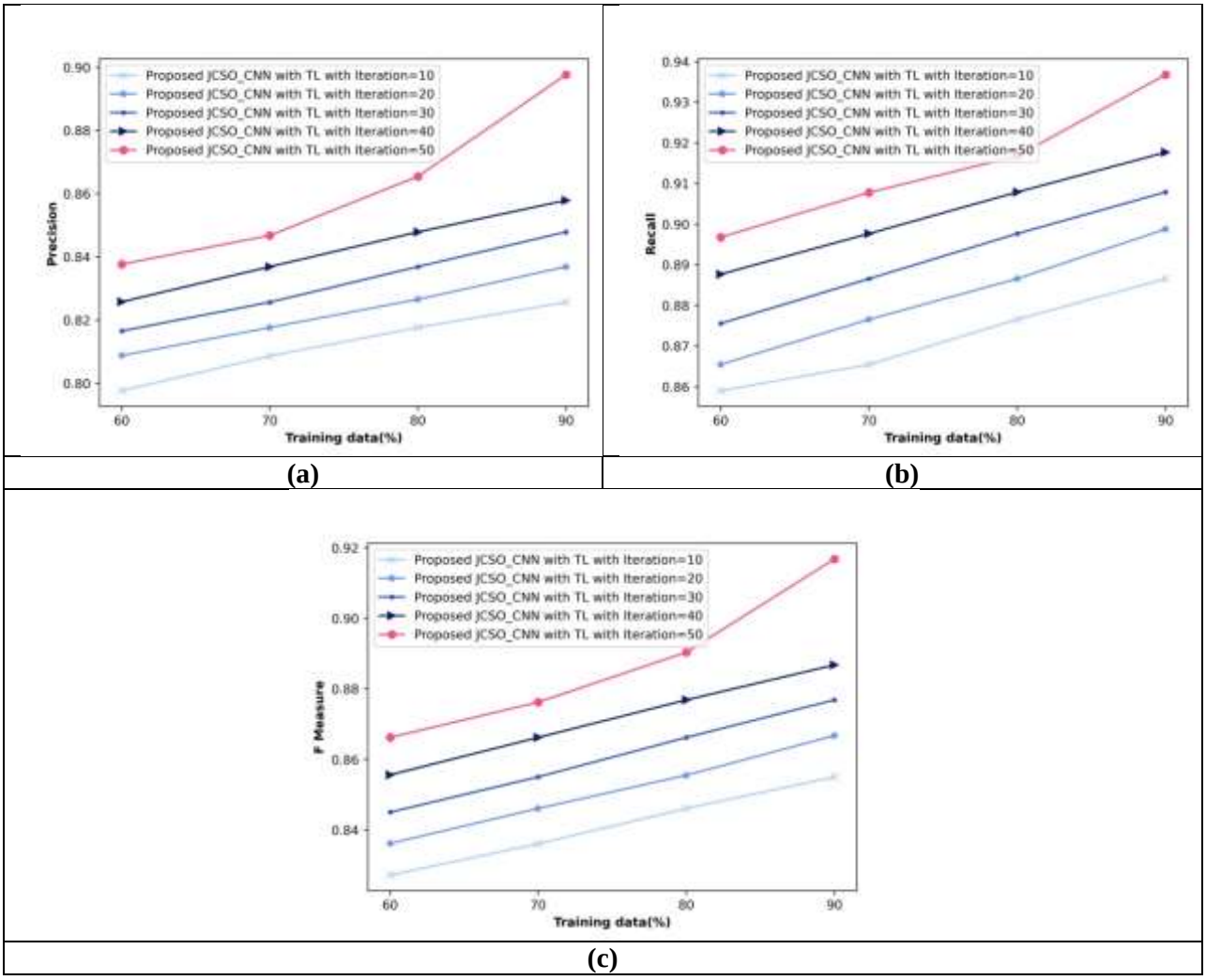


Figure 7. Performance analysis concerning training data, a) Precision, b) Recall, c) F-measure

4.5.2 Assessment regarding K fold

Figure 8 explicates estimation of JCSO_CNN with TL with regard to metrics by altering K fold for several iterations. The performance evaluation values obtained by JCSO_CNN with TL for K fold=8 are interpreted. Assessment of JCSO_CNN with TL by means of precision is expounded in figure 8 a). JCSO_CNN with TL obtained precision value of0.838 with iteration=10, 0.848 with iteration=20, 0.858 with iteration=30, 0.865 with iteration=40 and 0.908 with iteration=50. Figure 8 b) displays analysis of JCSO_CNN with TL based upon recall. Recall value attained by JCSO_CNN with TL are0.898, 0.908, 0.918, 0.937 and 0.946 with iteration=10, 20, 30, 40 and 50. Figure 8 c) illustrates evaluation of JCSO_CNN with TL concerning f-measure. F-measure acquired by JCSO_CNN with TL is 0.867 with iteration=10, 0.877 with iteration=20, 0.887 with iteration=30, 0.900 with iteration=40 and 0.926 with iteration=50.

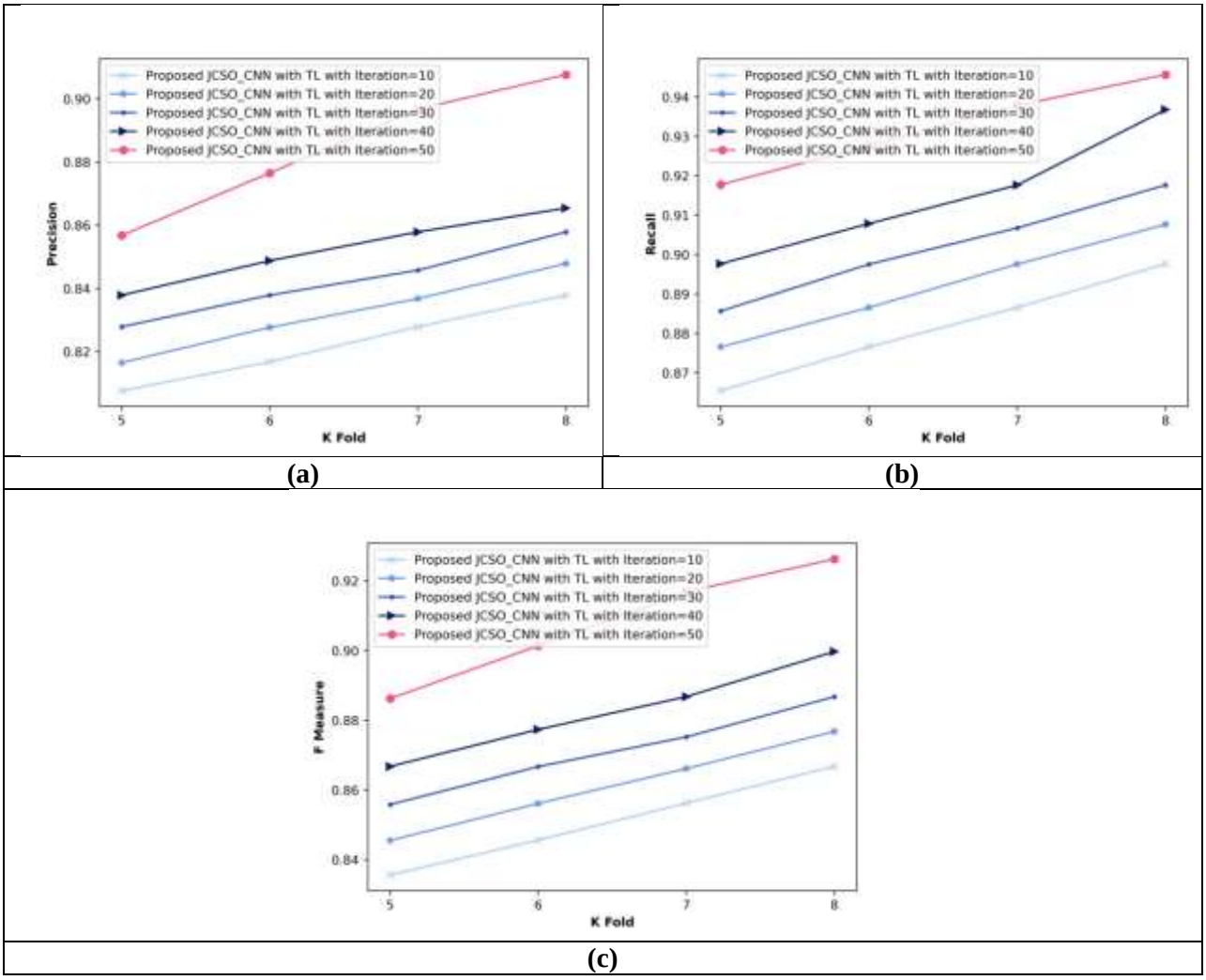


Figure 8. Performance analysis regarding K fold, a) Precision, b) Recall, c) F-measure

4.6 Comparative techniques

CE-Text [1], Bi-LSTM+CNN [2], CNN [3] and CRNN [4] are considered classical approaches to assess with JCSO_CNN with TL to prove its efficacy to recognize text in complex images.

4.7 Comparative evaluation

The evaluation of comparison concerning performance metrics are carried out in terms of training data and K fold.

4.7.1 Assessment regarding training data

Assessment of JCSO_CNN with TL by varying training data is explained in figure 9. For considered training data=90%, obtained values by JCSO_CNN with TL and classical techniques are manifested. Figure 9 a) mentionsestimation of designed JCSO_CNN with TL in relation to precision. JCSO_CNN with TL acquired precision of 0.898 whereas CE-Text, Bi-LSTM+CNN, CNN and CRNN obtained0.798, 0.808, 0.816 and 0.848 showing performance enhancement of JCSO_CNN with TL about11.140%, 10.023%, 9.135% and 5.542%. Estimation of JCSO_CNN with TL in regard to recall is described in figure 9 b). Recall attained by JCSO_CNN with TL is 0.937 while recall obtained by CE-Text is 0.847, Bi-LSTM+CNN is 0.865, CNN is 0.887 and CRNN and 0.898. It illustrates improvement of performance about9.606%, 7.615%, 5.339% and 4.175%. Analysis of JCSO_CNN with TL with relation to f-measure

is mentioned in figure 9 c). F-measure obtained by JCSO_CNN with TL is 0.917 whereas CE-Text, Bi-LSTM+CNN, CNN and CRNN achieved0.821, 0.836, 0.850 and 0.872. It represents the performance enhancing of JCSO_CNN with TL by10.396%, 8.861%, 7.316% and 4.878%.

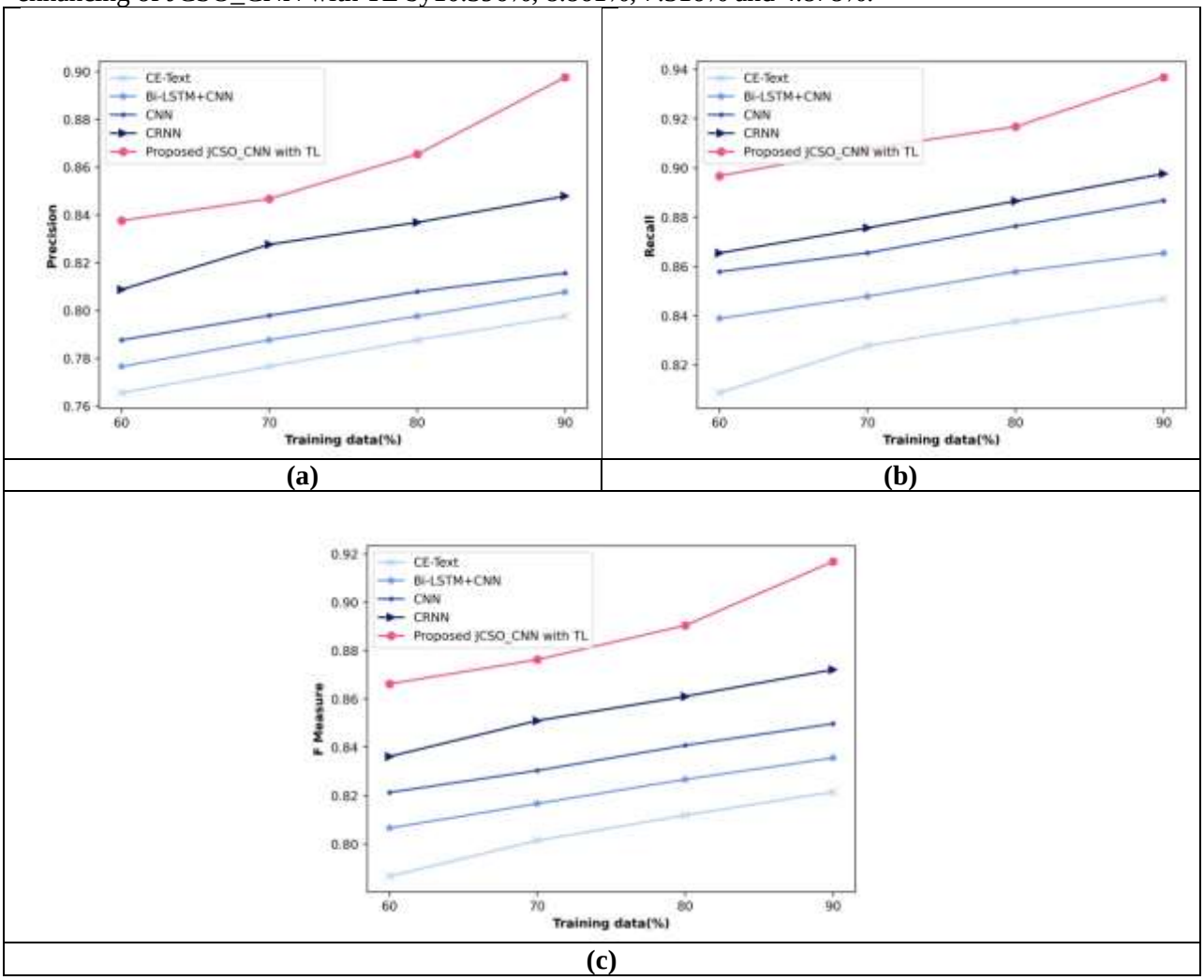


Figure 9. Comparative evaluation regarding training data, a) Precision, b) Recall, c) F-measure

4.7.2 Assessment regarding K fold

Figure 10 displays evaluation of JCSO_CNN with TL by altering K fold. The obtained values by JCSO_CNN with TL and conventionalapproaches for K fold=8 are explicated. Analysis of JCSO_CNN with TL considering precision is specified in figure 10 a).Precision value obtained by JCSO_CNN with TL is 0.908 while CE-Text, Bi-LSTM+CNN, CNN and CRNN achieved precision of0.837, 0.848, 0.857 and 0.876. This signifies performance enhancement of JCSO_CNN with TL by7.819%, 6.606%, 5.627% and 3.549%. Figure 10 b) indicates analysis of JCSO_CNN with TL in respect to recall. JCSO_CNN with TL acquired recall of 0.947 whereasrecall achieved by CE-Text, Bi-LSTM+CNN, CNN and CRNN are0.877, 0.898, 0.909 and 0.918that revealsenhancing of performance by JCSO_CNN with TL about7.417%, 5.187%, 4.015% and 3.075%. Assessment of JCSO_CNN with TL consideringf-measure is interpreted in figure 10c). F-measureobtained by JCSO_CNN with TL is 0.927 while the value acquired by CE-Text is 0.856, Bi-LSTM+CNN is 0.872, CNN is 0.882 and CRNN and 0.896. It

explainsenhancement of performance by JCSO_CNN with TLabout7.623%, 5.917%, 4.845% and 3.317%.

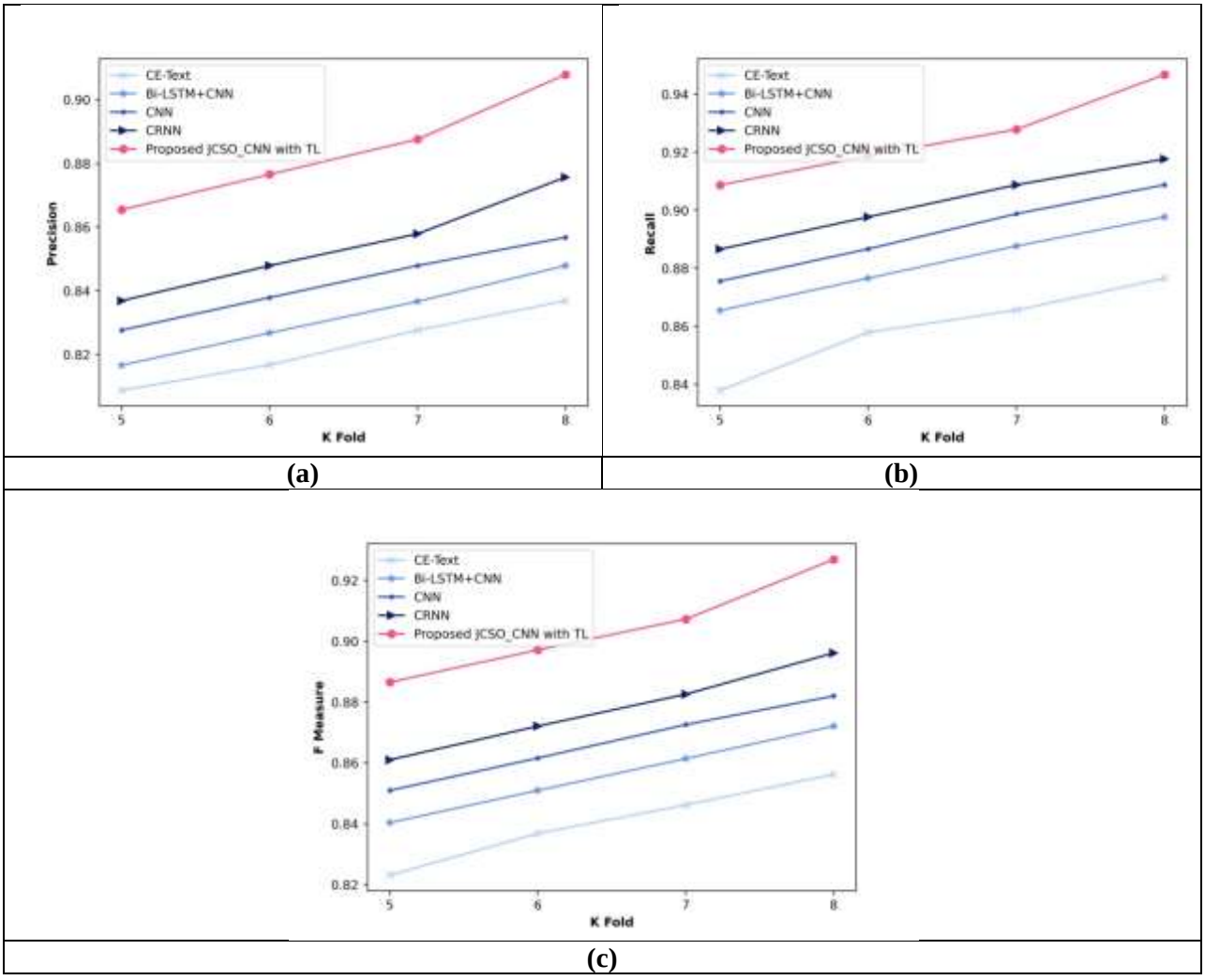


Figure 10. Comparative analysis concerning K fold, a) Precision, b) Recall, c) F-measure

4.8 Comparative discussion

JCSO_CNN with TL achieved excellent outcomes when comparing with CE-Text, Bi-LSTM+CNN, CNN and CRNN. The values obtained by the methods for evaluations are elucidated in table 1. It can be acknowledged that JCSO_CNN with TL acquired high values of precision, recall and f-measure of 90.8%, 94.7% and 92.7% when K fold is 8. Therefore, JCSO_CNN with TL is revealed as best method for recognizing text in complicated images.

Table 1. Comparative discussion of JCSO_CNN with TL

Analysis based upon	Metrics/Methods	CE-Text	Bi-LSTM+ CNN	CNN	CRNN	Proposed JCSO_CNN with TL
Training data=90%	*Precision*	79.8%	80.8%	81.6%	84.8%	89.8%
	Recall	84.7%	86.5%	88.7%	89.8%	93.7%

	F-measure	82.1%	83.6%	85.0%	87.2%	91.7%
K fold=8	*Precision*	83.7%	84.8%	85.7%	87.6%	**90.8%**
	Recall	87.7%	89.8%	90.9%	91.8%	**94.7%**
	F-measure	85.6%	87.2%	88.2%	89.6%	**92.7%**

5. Conclusion

With considerable strength of DL structures, researchers have been made great progression on efficiency and effectiveness of text recognition in last decades. However, owing to lack consideration of distinctive attributes of text elements, directly utilizing DL methods to execute text recognition chore is liable to outcome in less accuracy, particularly generating FP detection outcomes. In this work, JCSO_CNN with TL is designed to recognize text in complex images. Firstly, considered input image executes text foreground extraction using GrabCut algorithm. Then, text detection is conducted in text foreground extracted image employing DBNet++. Thereafter, character is segmented utilizing FLICM and lastly, character is recognized by CNN with TL, wherein CNN is utilized with hyperparameters from LeNet-5. The tuning of CNN with TL is performedby JCSO, which is introduced by joining JSO and CSO. In addition JCSO_CNN with TL attained maximum precision about 90.8%, maximum recall about 94.7% and maximum f-measure about 92.7% while considered K fold is 8. As a future chore, multi-orientation as well as multi-language recognition and detection will be carried out utilizing designed approach.

References

[1] Wu, Y., Zhang, W. and Wan, S., “CE-text: A context-Aware and embedded text detector in natural scene images”, Pattern Recognition Letters, vol.159, pp.77-83, 2022.
[2] Kantipudi, M.V.V., Kumar, S. and Kumar Jha, A., “Scene text recognition based on bidirectional LSTM and deep neural network”, Computational Intelligence and Neuroscience, 2021.
[3] Harizi, R., Walha, R., Drira, F. and Zaied, M., “Convolutional neural network with joint stepwise character/word modeling based system for scene text recognition”, Multimedia Tools and Applications, pp.1-16, 2022.
[4] Liu, Y., Wang, Y. and Shi, H., “A Convolutional Recurrent Neural-Network-Based Machine Learning for Scene Text Recognition Application”, Symmetry, vol.15, no.4, pp.849, 2023.
[5] Lin, Q., Luo, C., Jin, L. and Lai, S., “STAN: A sequential transformation attention-based network for scene text recognition”, Pattern Recognition, vol.111, pp.107692, 2021.
[6] Hassan, E. and VL, L., “Attention Guided Feature Encoding for Scene Text Recognition”, Journal of Imaging, vol.8, no.10, pp.276, 2022.
[7] Pandey, D., Pandey, B.K. and Wairya, S., “Hybrid deep neural network with adaptive galactic swarm optimization for text extraction from scene images”, Soft Computing, vol.25, pp.1563-1580, 2021.
[8] Luan, X., Zhang, J., Xu, M., Silamu, W. and Li, Y., “Lightweight Scene Text Recognition Based on Transformer”, Sensors, vol.23, no.9, pp.4490, 2023.
[9] Faustina Joan, S.P. and Valli, S., “A survey on text information extraction from born-digital and scene text images”, In Proceedings of the National Academy of Sciences, India Section A: Physical Sciences, vol.89, pp.77-101, 2019.
[10] Wang, Y., Shi, C., Xiao, B., Wang, C. and Qi, C., “CRF based text detection for natural scene images using convolutional neural network and context information”, Neurocomputing, vol.295, pp.46-58, 2018.
[11] Paul, S., Saha, S., Basu, S., Saha, P.K. and Nasipuri, M., “Text localization in camera captured images using fuzzy distance transform based adaptive stroke filter, Multimedia Tools and Applications, vol.78, pp.18017-18036, 2019.
[12] Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y. and Xue, X., “Arbitrary-oriented scene text detection via rotation proposals”, IEEE transactions on multimedia, vol.20, no.11, pp.3111-3122, 2018.

[13] Ghai, D. and Jain, N., "Comparative analysis of multi-scale wavelet decomposition and k-means clustering based text extraction", Wireless Personal Communications, vol.109, pp.455-490, 2019.
[14] El Bahi, H. and Zatni, A., "Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network", Multimedia tools and applications, vol.78, no.18, pp.26453-26481, 2019.
[15] Yue, X., Kuang, Z., Lin, C., Sun, H. and Zhang, W., "Robustscanner: Dynamically enhancing positional clues for robust text recognition", In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX, Cham: Springer International Publishing, pp. 135-151, November 2020.
[16] Huang, Y., Gu, C., Wang, S., Huang, Z., Chen, K. and Region, H.A., "Spatial Aggregation for Scene Text Recognition", In Proceedings of the 32nd British Machine Vision Conference, 2021.
[17] Arafat, S.Y. and Iqbal, M.J., "Urdu-text detection and recognition in natural scene images using deep learning", IEEE Access, vol.8, pp.96787-96803, 2020.
[18] Chen, X., Wang, T., Zhu, Y., Jin, L. and Luo, C., "Adaptive embedding gate for attention-based scene text recognition", Neurocomputing, vol.381, pp.261-271, 2020.
[19] Liao, M., Zou, Z., Wan, Z., Yao, C. and Bai, X., "Real-time scene text detection with differentiable binarization and adaptive scale fusion", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.45, no.1, pp.919-931, 2022.
[20] Geetha, R., Thilagam, T. and Padmavathy, T., "Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks", Neural Computing and Applications, pp.1-12, 2021.
[21] Shi, Z., Hao, H., Zhao, M., Feng, Y., He, L., Wang, Y. and Suzuki, K., "A deep CNN based transfer learning method for false positive reduction", Multimedia Tools and Applications, vol.78, pp.1017-1033, 2019.
[22] Wei, G., Li, G., Zhao, J. and He, A., "Development of a LeNet-5 gas identification CNN structure for electronic noses", Sensors, vol.19, no.1, pp.217, 2019.
[23] Chou, J.S. and Molla, A., "Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems", Scientific Reports, vol.12, no.1, pp.19157, 2022.
[24] Meng, X., Liu, Y., Gao, X. and Zhang, H., "A new bio-inspired algorithm: chicken swarm optimization", In Proceedings of Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, Proceedings, Part I 5, Springer International Publishing, pp. 86-94, October 17-20, 2014.
[25] ICDAR 2013 Dataset is taken from "https://datasets.activeloop.ai/docs/ml/datasets/icdar-2013-dataset/", accessed on August 2023.
[26] Thuraka, G.P., Sravani, V., Sujatha, B. and Sumalatha, L., "Deep Learning Model for Recognizing Text in Complex Images. In Machine Learning Technologies and Applications: Proceedings of ICACECS 2020, Singapore: Springer Singapore, pp.299-309, 2021.
[27] Talbot, J.F. and Xu, X., "Implementing grabcut", Brigham Young University, vol.3, 2006.
[28] Krinidis, S. and Chatzis, V., "A robust fuzzy local information C-means clustering algorithm", IEEE transactions on image processing, vol.19, no.5, pp.1328-1337, 2010.