International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-024-02324-y

ORIGINAL ARTICLE

TL-LFF Net: transfer learning based lighter, faster, and frozen network for the detection of multi-scale mixed intracranial hemorrhages through genetic optimization algorithm

Lakshmi Prasanna Kothala[1] · **Sitaramanjaneya Reddy Guntur**[2]

Received: 2 November 2023 / Accepted: 11 August 2024

Abstract
Computed tomography (CT) is the most commonly used imaging method in intracranial hemorrhage (ICH). Although deep learning (DL) models are well suited for detecting and segmenting multi-class hemorrhages, localizing multi-scale mixed hemorrhages with limited resources such as bounding boxes is difficult. To address this issue, the current study proposes a novel transfer learning-based TL-LFF Network. To detect multi-scale mixed hemorrhages, the proposed model employs a backbone module that extracts in-depth features from the input images, and a spatial pyramid pooling faster layer that performs the pooling operation at various levels. In the neck section, a path aggregated network (PANet) is used to store spatial information. Furthermore, to achieve a lightweight nature, the proposed backbone and neck modules were frozen during the backpropagation stage, resulting in a decrease in detection accuracy. To improve detection capability while remaining lightweight, a concept known as transfer learning is used. This strategy significantly improves the accuracy of the proposed model. In addition, the Genetic Algorithm (GA) concept is used to optimize the hyperparameters, where the mutation is used to develop new offspring based on previous generations. The brain hemorrhage extended dataset was used to train and validate the proposed model. In terms of detection metrics and lightweight criteria, the experimental results showed that the proposed model performed better when compared to other existing models. As a result, we can use the proposed model in the clinical implementation stage to reduce the radiologist's CT scan read time.

Keywords Deep learning · Frozen YOLO · Genetic algorithm · Hyperparameters optimization · Mixed intracranial hemorrhage · Transfer learning

1 Introduction

Stroke is one of the leading causes of death in India, and ICH accounts for 20% of all strokes. Blood loss inside the skull, or ICH, is a serious medical problem that necessitates immediate medical attention. Because ICH has a high mortality rate of 40–50%, it is critical to make a diagnosis as soon as the disease manifests itself [1–3]. One technique for diagnosing ICH that can provide a three-dimensional tomography image of the individual non-invasively and with a lower radiation dose than others is CT of the brain. It is a faster and more accurate diagnostic technique. Detecting ICH requires extreme precision and accuracy in a very short period. ICH is classified into five subtypes based on the location of the bleeding: intraparenchymal hemorrhage (IPH), epidural hemorrhage (EDH), intraventricular hemorrhage (IVH), subarachnoid hemorrhage (SAH), and subdural hemorrhage (SDH) [4, 5]. As illustrated in Fig. 1, a single patient may experience multiple hemorrhages as a result of multiple brain fractures Many classification strategies focus on developing a DL-based convolutional neural network (CNN) model that has been trained to classify hemorrhages as well as their type [6–9]. CNN is more useful than machine learning techniques because it extracts target features automatically. The concept of DL-based automatic cerebral bleeding

✉ Sitaramanjaneya Reddy Guntur
drgsr_bme@vignan.ac.in

Lakshmi Prasanna Kothala
klp_ecep@vignan.ac.in

[1] Department of Electronics and Communication Engineering, Vignan's Foundation for Science, Technology, and Research, Guntur, Andhra Pradesh 522213, India

[2] Department of Biomedical Engineering, Vignan's Foundation for Science, Technology, and Research, Guntur, Andhra Pradesh 522213, India

Published online: 23 August 2024

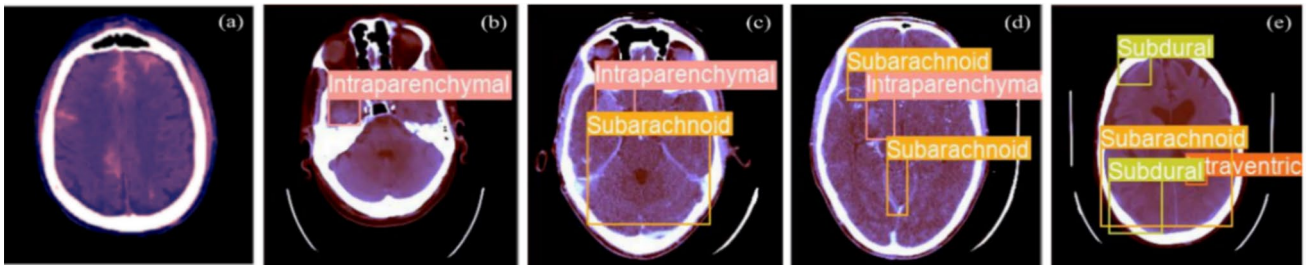

Fig. 1 A sample view of CT input images containing varying numbers of multiple hemorrhages, as well as a normal scan

classification and detection from a collection of CT scan images is proposed in this study [10–12]. The primary goal of this research is to reduce the role of humans in detecting ICH, to increase efficiency, and to save time and manpower.

In recent years, DL techniques, particularly CNN, have been used as the cutting edge for image processing. CNN has become well-known for object detection and classification problems due to its ability to automatically extract features from annotated images [13–16]. Despite CNN's superior categorization and detection abilities, the process is time-consuming because it must focus on multiple areas at the same time. The region-based CNN (RCNN) was created to address this issue. RCNN divides a picture into regions, which are then combined based on their attribute. The final step in this strategy is to exclude areas where the object is unlikely to be present. The slowness of RCNN, on the other hand, is critical in ICH detection applications [17]. Many models have been proposed to improve detection speed. "You Only Look Once" (YOLO) is the most likely used algorithm [18].

The main difference between RCNN and YOLO is that YOLO deletes areas with the lowest likelihood of having entities. YOLO has recently gained popularity due to its simplicity and adaptability to a wide range of jobs. YOLO will typically generate a P×P grid on the input image and calculate a separate output value based on the following inputs: [BBx, BBy, BBh, BBw, Pc, C1-C5]. Pc represents hemorrhage class probability, BBx, BBy, BBh, and BBw represent bounding box dimensions, and C1-C5 represents confidence levels for the five different hemorrhage class labels. YOLO was released in several versions to improve detection accuracy and reduce loss value, including YOLOv2, YOLOv3, YOLOv4, and YOLOv5 [19–21].

2 Related work

DL exhibits exceptional accuracy in classifying and identifying medical tasks such as brain tumors and hemorrhages from raw images [22–24]. The majority of researchers in the literature have attempted to identify ICH using two-class detection or multi-class classification, which attempts to identify different ICH types. Yuh et al. provided a threshold-based approach for binary ICH detection. Their method had a 98% recall and a 59% specificity for detecting ICH [25]. Li et al. used the Bayesian decision procedure to achieve 100% testing recall and 92% specificity [26]. In classifying the presence of hemorrhage in a patient, Pong et al. obtained a 0.997 F1 score from LeNet [27]. Thay et al. proposed a random forest classifier with 99% and 98.8% precision and recall, respectively [28]. Two transfer learning-based models were developed by Zhou et al. and obtained a recall score of 0.874 for the Resnet architecture and 0.802 for the DenseNet-121 simulation [29]. Before transfer learning, Vrbancic et al. tuned the hyperparameters using the grey wolf optimization technique, resulting in recall, precision, and an F1 score of 93%, 90%, and 91% [30]. Three models were developed by Chen et al. based on the concept of transfer learning and the results were very effective in identifying whether a patient was having an ICH or something else [31]. Despite having a better classification accuracy, these approaches could only categorize hemorrhages into two categories: either positive or negative.

Majumdar et al. classified different types of ICH with a high specificity of 98% and recall of 81%. [32]. Grewal et al. used DenseNet architecture for slice-level predictions and a recurrent neural network layer for incorporating 3-D context. When the results were compared to those of three radiologists, they discovered that their method was superior to those of two radiologists in terms of hemorrhage prediction accuracy of 81.82% [33]. Chilamkurthy et al. used ResNet18 and random forest algorithms to classify the ICH subtype, and they predicted test results with relatively lower sensitivity and 70% specificity [34]. Two similar ResNet50 models with different preprocessing methods were utilized by Sage et al. Numerous features were produced by combining the feature maps from the two modules and the F1 score was 88.2% [35]. The findings of the EfficientNet-B3 and SE-ResNetXt50 deep network architectures were combined by the author He to identify cerebral bleeding and its subtypes on head CT scans [36]. Although these models can detect bleeding, they cannot pinpoint the location of

ICH. In addition to detection, some researchers attempted to segment ICH.

Farzaneh et al. presented a method for segmenting the SDH class using a traditional feature extraction algorithm and a three-bagger classifier [37]. Remedios et al. created the U-Net with transferred weight as a multisite learning model [38]. Although previous research has shown great success in segmenting various hemorrhagic lesions, segmenting multiple hemorrhages on a single CT scan remains a challenge. Kuo et al. proposed using a PatchFCN model to detect acute cerebral bleeding to address the multi-class segmentation issue [39]. Even though the PatchFCN provided pixel-level monitoring as well as evaluation metrics for classification, the quantitative assessment of various lesions was ignored. Chang et al. used a modified quick mask RCNN method to identify and segment bleeding [40]. Despite producing excellent segmentation results, the model only predicted the volume of IPH. Sharrock et al. made the Deep Bleed public source code available for the ICH segmentation of ICH lesions as well as the detection of IVH and SDH [41]. Object detection (OD) methods are difficult to apply in mixed ICH detection due to the very small bleeding region in comparison to the overall size of the image.

Nevertheless, some authors prefer OD algorithms because they have advantages over both conventional and DL-based approaches. A 2D faster R-CNN network was proposed by Ferlin et al. with recall, precision, and F1-scores of 92.6%, 89.7%, and 90.8%, respectively [42]. With two proposed architectures, Le THY et al. found that the R-FCN model had a greater recall of 82.6%, a precision of 90.5%, and an F1 score of 86.45 [43]. By combining YOLO and 3D CNN, Al-masni et al. achieved an F1 score and precision of 77.6%, and 67.2, respectively [44]. Zhang et al. worked on segmenting ICH using a two-stage Mask RCNN [45]. A ResNet-101 architecture is used as the primary framework, followed by a thresholding-based algorithm for localizing the bleeding area and a 3D visualization technique for modelling the clot area. The operation of Mask R-CNN requires two steps, which increases the execution time. Li et al. proposed a feature improvement strategy and applied it separately to the current SSD-512 and SSD-300 to obtain optimized results while using a pre-trained VGG as a backbone network [46]. Even though the SSD is a single-stage model, Myung et al. proposed a YOLOv2 network based on a pre-trained ResNet-50 backbone due to technological advances [47].

Finally, Ertugrul et al. used the YOLOv4 architecture to improve detection results by constructing a bounding box around the bleeding zone [48]. The head was YOLOv3, the neck was SPP and PANet, and the backbone was cross-stage partial (CSPDarknet53) and these three modules comprise the YOLOv4 architecture. By training each unique kind of hemorrhage separately, they were able to acquire overall values of 93.8%, 92.8%, 90.6%, and 91.8%, for accuracy, F1 score, mAP, and recall, respectively. When all types were considered together, these figures fell to 92%, 86%, 79.6%, and 81% respectively. These findings imply that model implementation can be enhanced in the context of multi-scale mixed ICH. However, after analyzing the findings, we concluded that there is still a chance to improve detection performance when mixed hemorrhage is present. Another drawback of current models is that they require more system resources to perform the dense prediction task. To address both issues, a unique TL-LFF Network was proposed, which will greatly enhance accuracy even with low resources.

The primary motivation of this research is to create a model that can detect multi-scale mixed hemorrhages with fewer resources. To accomplish this, we presented a new architecture called TL-LFF Net, which was built using transfer learning and the frozen layer concept. The frozen concept was used to reduce the complexity of the recommended model, while transfer learning was used to localize multi-scale mixed ICH. Furthermore, hyperparameter optimization via GA was used to improve detection accuracy. The following are the main contributions of the proposed model:

1. Proposed a lighter, faster network for detecting multi-scale mixed hemorrhages without sacrificing detection accuracy.
2. To localize multi-scale mixed hemorrhages upon detection with higher accuracy the proposed model uses a more streamlined architecture compared to YOLOv4. To achieve this the proposed model backbone employs a spatial pyramid pooling faster (SPPF) layer that performs the pooling operation at five different levels. In the neck part, a path-aggregated network is used to store spatial information.
3. To achieve a lightweight nature the backbone and neck sections were frozen and the transfer learning concept was used to improve efficiency.
4. To improve detection accuracy furthermore, a genetic algorithm is used to optimize hyperparameters. To carry out GA, the mutation is used to generate new child classes with an 80% probability and a 0.04 variance based on a mixture of the best parents from all previous generations.
5. The proposed model's performance is compared to other existing models in terms of detection accuracy and computational complexity. To assess detection accuracy, four metrics were considered such as recall, precision, F1 score, and mAP. Several parameters, including frames per second (FPS), iterations per second (IPS), number of floating-point operations (FLOPS), execution time, GPU utilization percentage, etc., were used to assess computational complexity.

This research article is divided into the sections listed below. Section 1 includes a detailed survey of the literature on standard techniques as well as the most recent deep-learning models, as well as a basic introduction to ICH. Section 2 describes the benchmark methodology for detecting multi-scale mixed ICH with limited resources. Section 3 presents the experimental results as well as the dataset used. Section 4 provides a comparative analysis with a related discussion to choose the best model. Finally, the proposed model's advantages, limitations, and future scope were discussed.

3 Proposed model

3.1 Employed methodology

The primary goal of this study is to develop a lighter and faster model for categorizing multi-scale mixed hemorrhages in a specific CT without sacrificing accuracy. To achieve lightweight, a portion of the initial weights are frozen, while the remaining weights are used to compute loss and are updated by the SGD optimizer. As a result, the proposed model consumes fewer resources than traditional training and enables shorter training times, resulting in lower final training accuracy. The transfer learning concept is used to improve accuracy. It is a useful technique for quickly retraining the proposed model without retraining the entire network. Furthermore, the hyperparameters were tuned to their optimal values using a Genetic Algorithm. Figure 2 depicts the entire flow of the recommended method. The following sections discuss each of the steps shown in Fig. 2 in a detailed manner.

3.2 Data pre-processing and augmentation

The collected input data is in DICOM format, and the pixel information is extracted and converted to grayscale. Windowing is used to create three types of input images: brain, bone, and subdural window images. This technique is used in the proposed model because a particular window image will highlight a specific part of the total image. The generated images were layered all together to get a single 3-channel RGB image. Different types of augmentation techniques were employed in the proposed model to increase detection accuracy and to avoid overfitting issues. The mosaic method combines four training images into a single image with a fixed ratio and the mosaic has the advantage of continuing the training process even if the input images are of varying scales [49–51]. The mix-up method is useful for regularisation because it creates new training images by combining two separate class objects from the input dataset. A random region of the input image is covered with black square pixels in the cutout process. Neurons in a contiguous section of a feature map are dropped collectively in a drop block. A sample of the training images after the augmentation process is shown in Fig. 3.

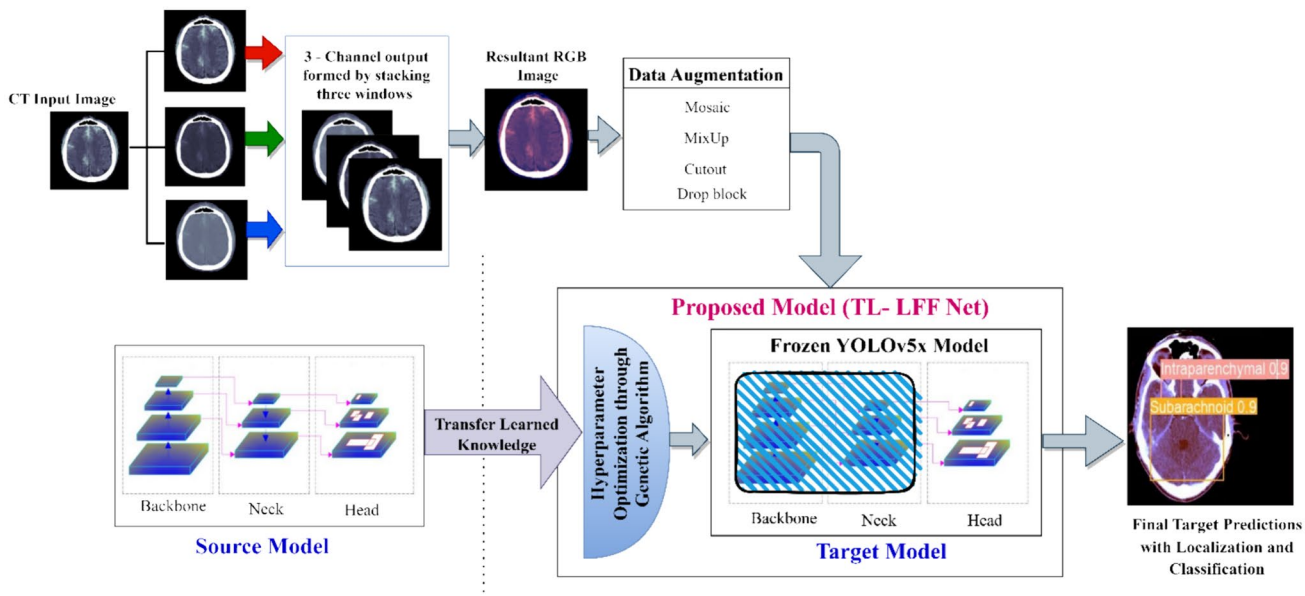

Fig. 2 A block diagram of the overall process for developing a lightweight model to detect mixed ICH in a given CT with improved accuracy

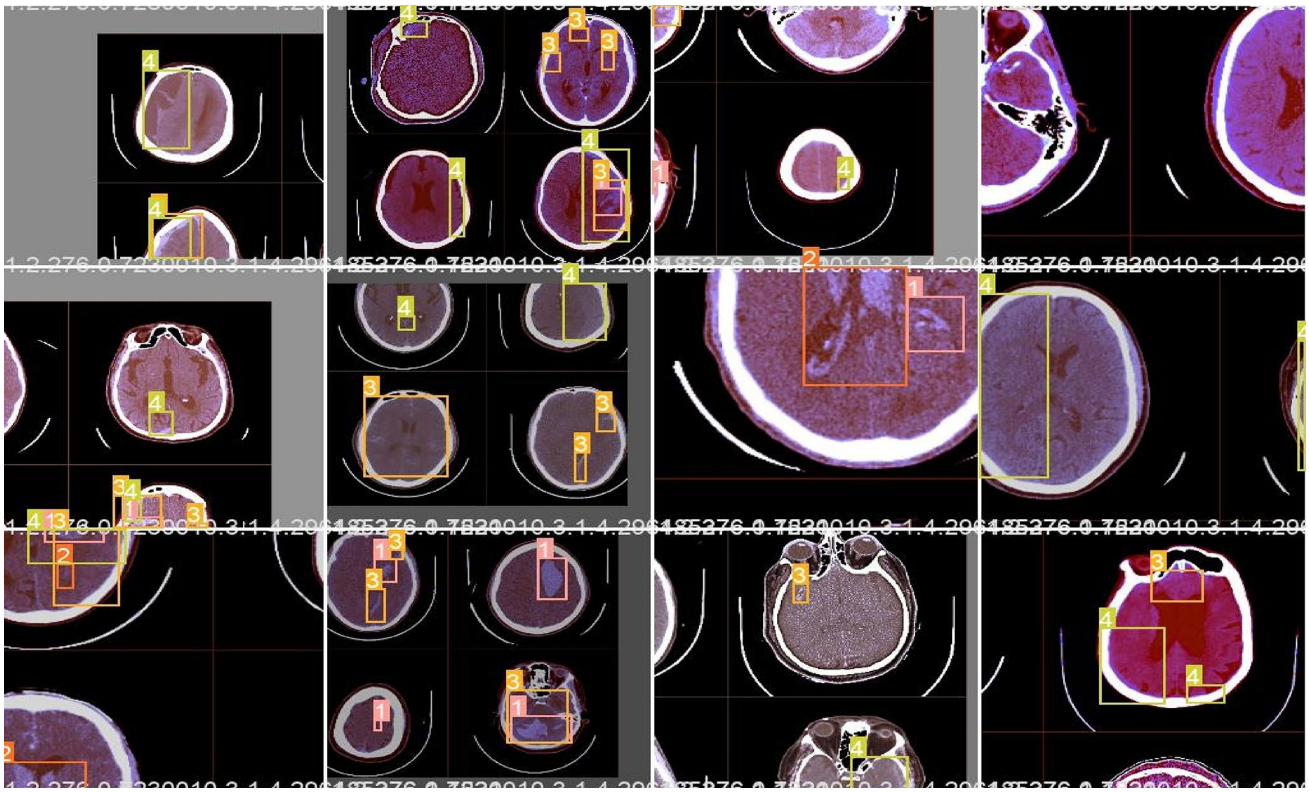

Fig. 3 An image that displays a collection of twelve training photos that were created while using different data augmentation techniques to train the proposed model

3.3 Proposed architecture (TL-LFF Net)

The proposed model is a single-stage architecture, directly predicting bounding boxes and class probabilities. It uses a backbone network, such as CSPDarknet53, for feature extraction. The neck and head structure for feature aggregation and detection. It predicts bounding box coordinates, confidence scores, and class probabilities simultaneously to achieve real-time performance.

3.3.1 Source model

The proposed model architecture is pre-trained on the ImageNet database and consists primarily of three modules referred to as backbone, neck, and head. Figure 4 depicts the overall architecture of the proposed source model. The backbone is a component that extracts the image features. It consists of four C3 layers (bottleneck cross-stage partial with three convolutional layers), four convolutional layers, and one SPPF layer [52]. The cross-stage partial (CSP) network is used by the backbone to extract feature maps from the input image while minimizing computations. The proposed model incorporates a focus layer to reduce the number of layers, parameters, FLOPS, and CUDA memory while increasing forward and backward performance. The focus layer employs the slicing mechanism and concat operation as shown in Fig. 4. The striding approach is used for convolution, followed by batch normalization and mish activation. C3 is used to reduce complexity and eliminate the vanishing gradient problem by dividing the base layer input into two parts. Four C3 modules and convolutions were used in the design of the backbone. Finally, SPPF is employed to carry out the pooling operation at five distinct sizes [53].

The purpose of the neck block is to collect all of the feature maps from various stages. In this model, PANet serves as the neck and it goes through three processes while reducing the number of feature maps. To shorten the information path, bottom-up path augmentation is used [54, 55]. An adaptive feature pooling module is used to aggregate both high and low-level feature maps. Finally, fully connected layers predicted the location of the hemorrhage in the given image. The head block is used in the final stage of the detection model. It generates final outputs such as detection probability scores, bounding boxes, and class names. Our proposed model recognizes small-size hemorrhages with a scale of 13×13, medium-size with a scale of 26×26, and large-size with a scale of 52×52.

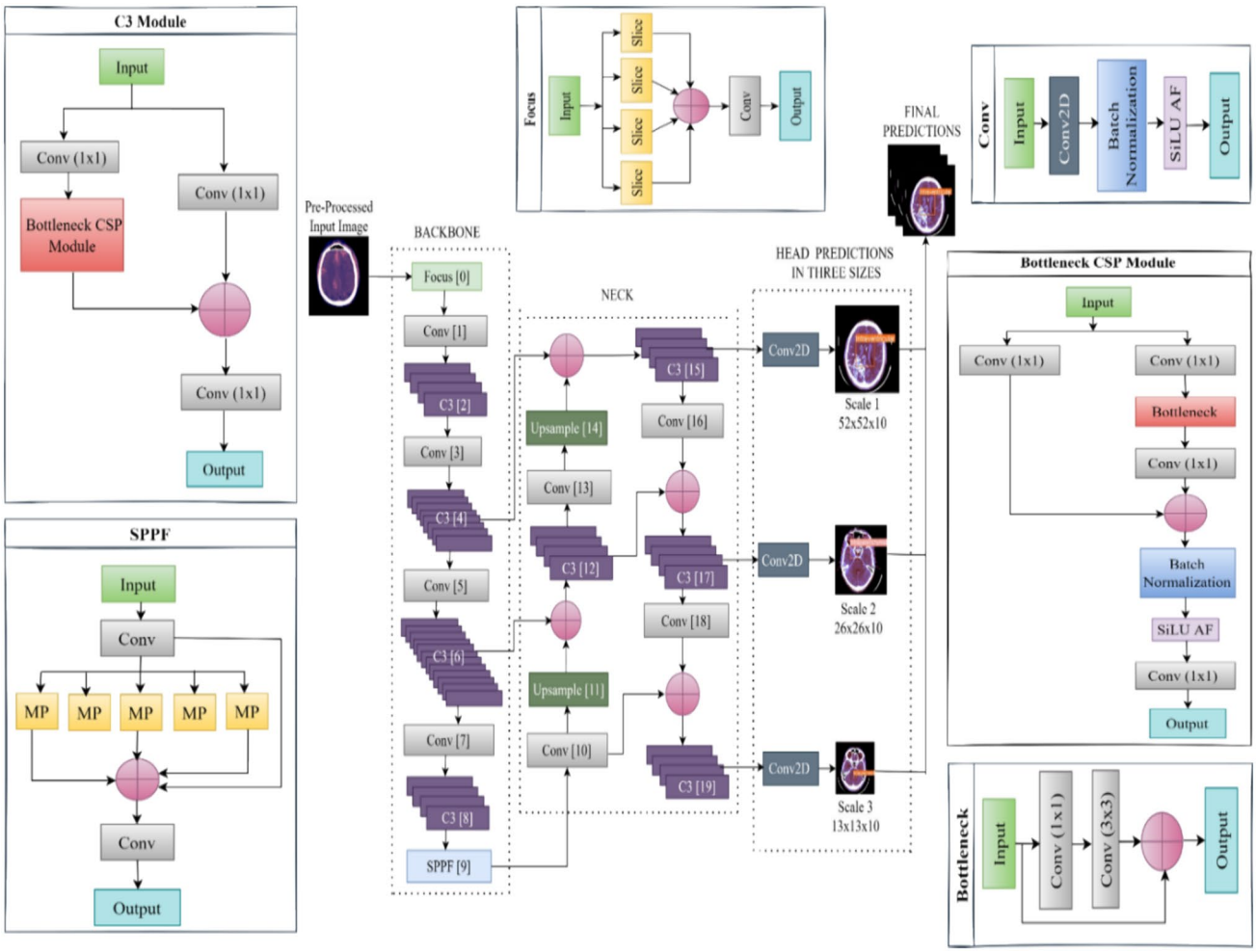

Fig. 4 The source model architecture, including all of its submodules

3.3.2 Target model

3.3.2.1 Hyperparameter optimization Hyperparameter optimization is a critical component in the field of DL. As a result, we used the hyperparametric optimization model during the training process. Before training the model, the hyperparameters, which are different from model parameters, must be set. In contrast to model parameters, hyperparameters typically exhibit unpredictable behavior. Since the proposed model performance on training and validation samples must be evaluated empirically, hyperparameter adjustment is usually required. Many aspects of training are governed by hyperparameters and determining the best settings for them can be difficult. Grid searches, for example, can quickly become problematic due to (1) the large dimensional search field, (2) unknown relationships between dimensions, and (3) the time-consuming process of assessing fitness at each stage, making GA an excellent choice for hyperparameter optimization. To optimize YOLOv5 using a genetic algorithm, the following steps are required.

1. Define the Genetic Algorithm Parameters: Set parameters such as population size, mutation rate, crossover rate, and termination criteria.
2. Initialize Population: Generate an initial population of YOLOv5 models with random hyperparameters. Each individual in the population represents a potential solution.
3. Evaluate Fitness: Evaluate the fitness of each individual in the population. Fitness can be measured by metrics such as mean average precision (mAP) on a validation dataset. The better the performance of the YOLOv5 model, the higher its fitness score.
4. Selection: Select individuals from the current population to create the next generation. This selection process can be based on fitness scores, where individuals with higher fitness have a higher chance of being selected.
5. Crossover: Apply crossover to selected individuals to create offspring for the next generation. This involves

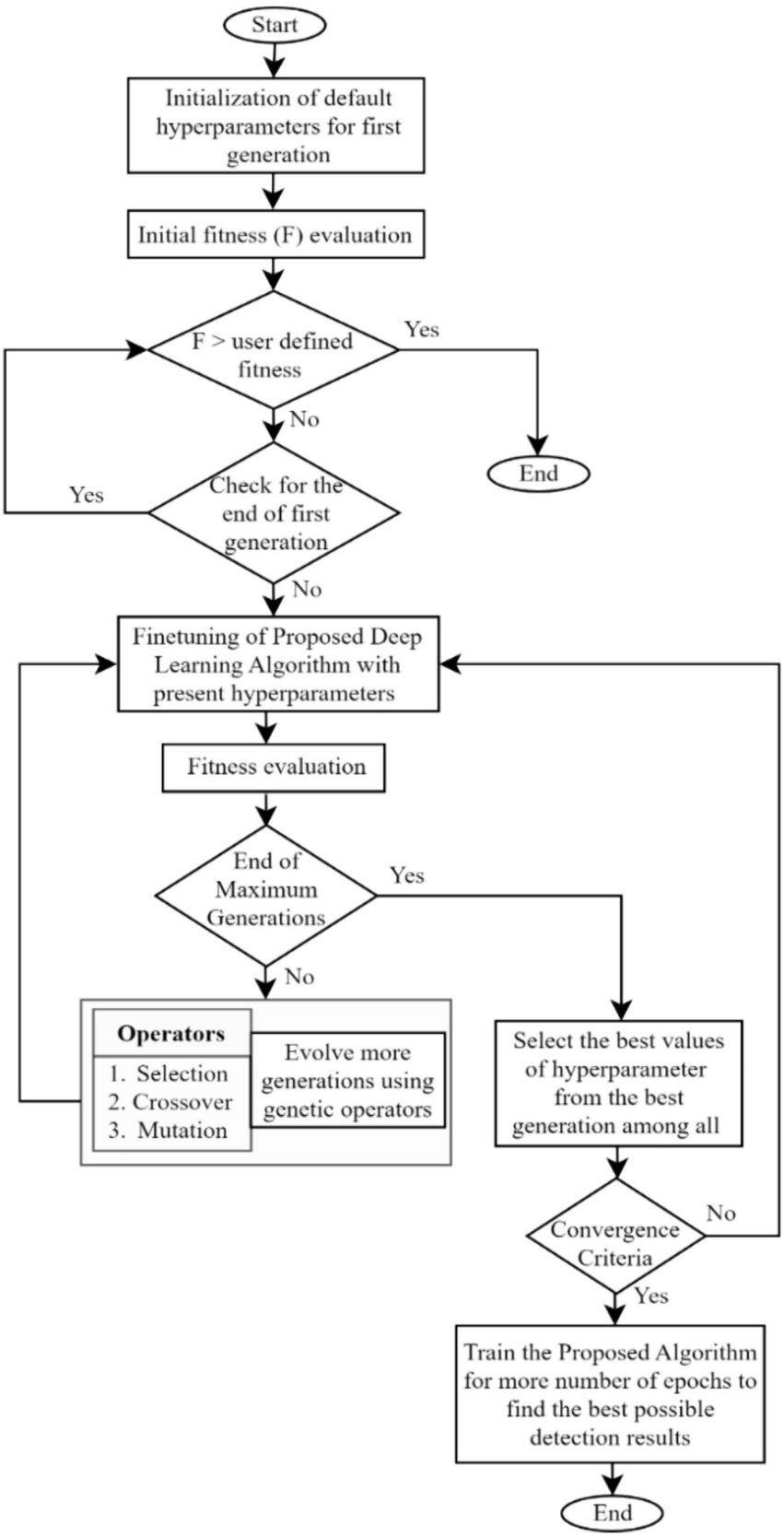

Fig. 5 Flow of the genetic algorithm working to find the best hyperparameter values

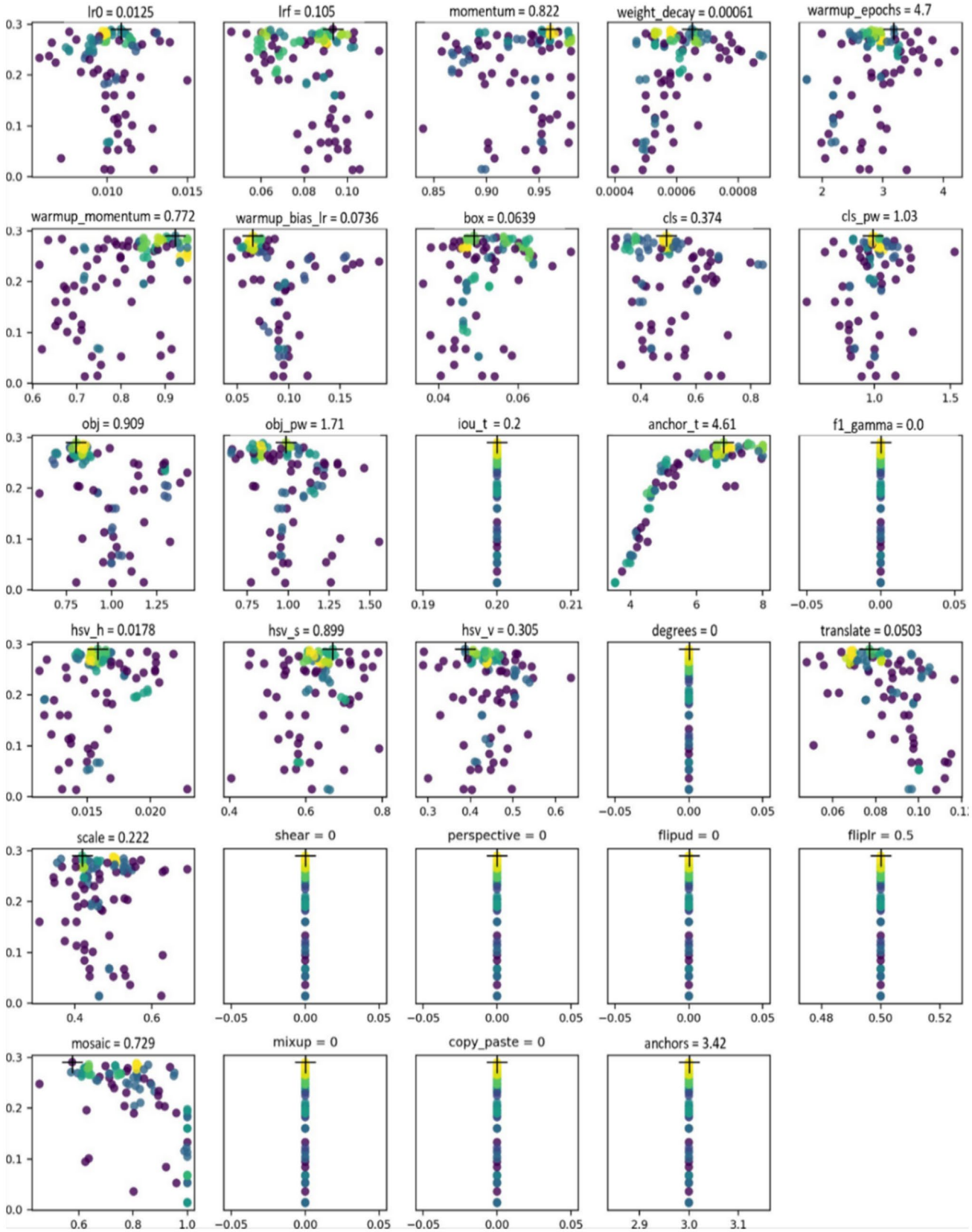

Fig. 6 The hyperparameter values including learning rate, momentum, anchor, weight, warm-up, and so on obtained using GA

Table 1 Hyperparameter values that show the variation before and after the application of the GA

S. no	Name of the hyperparameter	Scratch value (Before GA)	Resultant finetune value
1	Initial Learning rate value	$LR_{initial}$: 0.01	$LR_{initial}$: 0.0125
2	Final learning rate value	LR_{final}: 0.1	LR_{final}: 0.105
3	Momentum value in SGD optimizer	0.937	0.822
4	Weight factor decay value in SGD optimizer	0.0005	0.00061
5	Warmup epochs	3	4.7
6	Initial warmup momentum value	0.8	0.772
7	Initial warmup bias learning rate value	0.1	0.0736
8	Gain of bounding box loss	0.05	0.0639
9	Gain of classification loss	0.5	0.374
10	Classification BCELoss positive weight factor	1.0	1.03
11	Gain of object loss	1.0	0.909
12	Object BCELoss positive weight factor	1.0	1.71
13	The threshold for IOU during training	0.2	0.2
14	The threshold for anchor-multiple during training	4.0	4.61
15	Gamma value in focal loss calculation	0	0.0
16	HSV-hue augmentation value	0.015	0.0178
17	HSV-saturation augmentation value	0.7	0.899
18	HSV-value augmentation value	0.4	0.305
19	Rotation factor for augmentation	0	0.0
20	Translation factor for augmentation	0.1	0.0503
21	Scale factor for augmentation	0.5	0.222
22	Shear factor for augmentation	0.0	0.0
23	Perspective transformation value	0.0	0.0
24	Image up-down flip probability value	0.0	0.0
25	Image left–right flip probability value	0.5	0.5
26	Image mosaic probability for augmentation	1.0	0.729
27	Image mix-up probability for augmentation	0.0	0.0
28	Image segment copy-paste probability	0.0	0.0
29	Number of anchors per output layer	3	3.42
30	Anchor values	[10,13,16,30,33,23] [30,61,62,45,59,119] [116,90,156,198.373,326]	[14,16,24,24,16,51] [29,42,47,51,35,79] [77,62,53,118,116,134]

exchanging genetic information between two parent individuals to produce new solutions.

6. Mutation: Introduce random changes to the offspring's hyperparameters to maintain diversity in the population and explore new regions of the search space.
7. Evaluate Fitness of Offspring: Evaluate the fitness of the offspring generated through crossover and mutation.
8. Replacement: Replace the least fit individuals in the current population with the offspring to form the next generation.
9. Termination: Check if termination criteria are met. This could be a maximum number of generations reached or achieving a desired level of performance.
10. Repeat: If termination criteria are not met, repeat steps 3–9 until the termination criteria are satisfied.

During training, YOLOv5 employs nearly 30 hyperparameters. To boost the model's performance, we aim to find the ideal combination of these 30 hyperparameters [56]. During the first generation of GA hyperparameter tuning, all parameters will be assigned default values. Our goal in the GA is to either maximize or decrease the fitness function. In YOLOv5, we created a weighted mixture of metrics as the default fitness function, with mAP@0.5 accounting for 10% of the weight and mAP@0.5:0.95 accounting for the remaining 90%. The GA is repeated 100 times for the base scenario, for a total of 100 generations. Selection, crossover, and mutation are the three most important genetic operators. The mutation is used to produce new children from a mix of the best parents from all previous generations, with an 80% likelihood and a 0.04 variance. Figure 5 shows the overall working flow of GA. Once the convergence criteria are

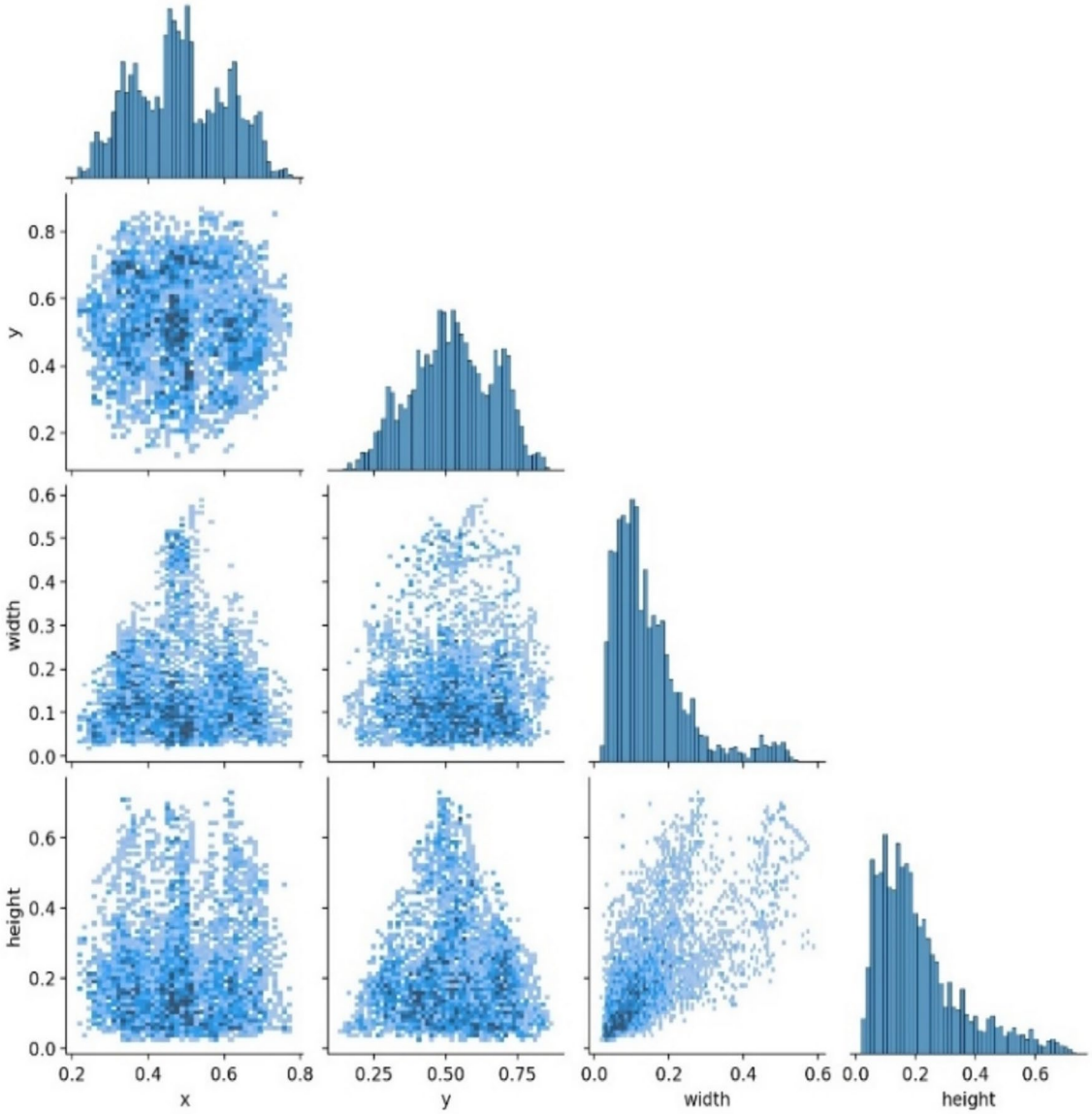

Fig. 7 The statistical distribution plot of training image labels correlograms and histograms

reached, the GA returns the best hyperparameter values. The final hyperparameter values are depicted in Fig. 6, which include learning rate, momentum, anchor, weight, warm-up, and so on. Table 1 shows the default and optimized hyperparameter values used in the proposed model training. An auto anchor mechanism is activated to regenerate a new set of anchors when the initial size of the anchors is not suitable, and it is implemented using k-means clustering [57, 58]. The k-means clustering algorithm groups samples that are similar to each other by assessing their similarity.

3.3.2.2 Froze network and transfer learning To build the target model, initially, we froze the backbone and neck modules of the source model, and then applied the transfer learning concept. The concept of freezing was used in the target model for two reasons. First, because we are freezing 24 layers (backbone and neck), backpropagation cannot occur through those layers so they are not being trained [59]. As a result, the time required for training iterations will be reduced. Second, even though we temporarily froze a few blocks, the medium model is still capable of making predictions on par with a fully trained model. Transfer learning is a method of training a new model on similar data by using a previously trained model on a large dataset. This technique accelerates the training process, limiting the deep learning model ability to be created from scratch. The detailed outline of how transfer learning is implemented is given below:

1. Pre-trained Model Selection: We choose a pre-trained YOLOv5 model as the starting point for transfer learning. Pre-trained models are typically trained on large-scale datasets like COCO or ImageNet.
2. Dataset Preparation: The target dataset is prepared for fine-tuning which involves organizing the dataset into YOLO accepted format and splitting it into training, and validation.
3. Network Modification: Modified the pre-trained model to adapt it to the target task depending on the specific requirements. Here the pretrained model is frozen during fine-tuning. Typically, the early layers such as the backbone and neck are frozen to retain the learned low-level features, while the later head layer is fine-tuned to adapt to the target task.
4. Fine-tuning Process: It starts by feeding batches of images and their corresponding annotations into the proposed network. During training, the weights of the unfrozen layers are updated. Finally, the training process is tracked by using metrics such as accuracy, precision, and mAP on the validation set [60, 61].

4 Experiments

This section initially outlines the evaluation dataset and the criteria that were used to verify both detection and lightweight capacity. Afterward, the proposed model's implementation details are presented. Then, ablation tests were performed on the proposed enhancements and compared them to state-of-the-art approaches. Finally, the advantages, limitations, and potential future application of our proposed model were discussed.

4.1 Dataset for experimentation

The proposed model was trained and validated with 491 patients data from a publicly accessible brain hemorrhage extended (BHX) dataset [62]. BHX is an addition to the qure.ai CQ500 dataset, in which, 205 were found to be positive and the remaining were found to be normal. For both training and validation, only data from positive patients were used. So, a total of 21,132 slices were taken from 205 individuals, including all of the different types of ICH as well as a few slices with mixed hemorrhages. The count of each class in the employed data is EDH of 587, IPH of 7244, IVH of 2432, SAH of 9951, and SDH of 7494. The annotations for all of the scans were created by three professional radiologists, and the percentage of agreement among these readers was determined by two measures known as Cohen's Kappa and Fleiss' Kappa coefficients [63]. Figure 7 shows the statistical distribution of the correlogram graph, which is used to determine the correlation and randomness between two parameters of the training data. The main limitation of the employed data is it is unbalanced, and has a skewed distribution for all the different types of ICH classes. So, to prevent the proposed model from being biased towards the more prevalent hemorrhage types, techniques like data augmentation, resampling, or weighting can be used.

4.2 Evaluation criteria

The proposed model is assessed in terms of detection accuracy and computational complexity. This paper used a variety of metrics, which are defined in the sections below.

4.2.1 Criteria to check detection accuracy

Different measures are used to assess detection accuracy, including the confusion matrix, precision, recall, PR curve, F1 score, mAP at various thresholds, and confidence score.

4.2.1.1 Confusion matrix It is an N×N square matrix with the elements of TP, FP, TN, and FN. Where N represents the number of classes in the employed dataset. It displays the comparison between actual and predicted values. This matrix is used to track the effectiveness of object classification.

4.2.1.2 Precision It is defined as the proportion of positively recognized samples to the total number of positive samples. The precision value at distinct confidence levels is considered while plotting the precision curve. The precision is calculated using Eq. 1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

4.2.1.3 Recall The recall is calculated as the percentage of positive samples among positive samples that were accurately classified as positive. Recall measures the model's ability to recognize positive samples. The recall value at distinct confidence levels is considered while plotting the recall curve. The recall is calculated using Eq. 2.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

4.2.1.4 Precision-recall curve It is a trade-off between recall and precision that is determined by taking into consideration of confidence values about the bounding boxes generated by a detector. An object detector is regarded as good if it has high precision and recall values.

4.2.1.5 F1 score It is a weighted average of precision and recall that equals one when both are 1. The F1 score is calculated using Eq. 3.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

4.2.1.6 mean average precision (mAP) It is determined as an addition to the average precision (AP). The precision of each ICH class is calculated separately using AP, while the total model precision is calculated using mAP. The mAP is the sum of all individual classes because the proposed model calculates hemorrhages in three sizes: small, medium, and large. Equation 4 can be used to calculate the value of mAP, where K is the number of classes. The area under the precision and recall curve (AP), is given by Eq. 5,

$$mAP = \frac{1}{K}\sum_{i=1}^{K}(Average Precision)_i \tag{4}$$

$$AP = \int_0^1 p(r)dr \tag{5}$$

where r represents the recall rate and p represents the precision rate. mAP@0.5:0.95 is calculated by taking the average of ten different IOU criteria that fall between [0.5, 0.95], with each value increased by 0.05.

4.2.1.7 Confidence score The confidence score reflects the likelihood of the box containing an object of interest as well as the classifier's level of confidence. If there is no object in that box, the confidence score should be zero. It can be calculated using Eq. 6

$$CS_P^B = P_{B,P} \times IOU_{Groundtruth}^{Predicted} \tag{6}$$

where CS is the confidence score of the B^{th} bounding box in the P^{th} grid cell and $P_{B,P}$ is the bleeding probability. IOU denotes the amount of overlap between the predicted and actual bounding boxes.

4.2.2 Criteria to validate lightweight capability

The lightweight capability is determined in terms of computational complexity with the help of measures such as FPS, IPS, total execution time, percentage of CPU and GPU utilization, GPU power usage, process memory in use, GPU time spent accessing memory, and GPU memory allocated.

4.2.2.1 Frames per second (FPS) It is a measurement of the number of images processed per second during the learning process. A large number indicates that the training period will be shorter. It is one of the most important factors to consider when using object detection algorithms.

4.2.2.2 Iterations per second (IPS) It is used to represent the number of iterations analyzed per second during the learning process. The training session will be short if the IPS is large.

4.2.2.3 Total execution time It is the time required to complete the entire training process.

4.2.2.4 Percentage of CPU and GPU utilization These two parameters describe how much work a CPU or GPU can handle swhile the proposed model is being executed.

4.2.2.5 GPU power usage It denotes the amount of processing power required by a GPU to complete the training process.

4.2.2.6 GPU time spent accessing memory It is used to compute the percentage of time GPU memory was read or write during the previous sample period.

4.2.2.7 GPU memory allocated percentage This indicates how much GPU memory has been used.

4.3 Implementation details

Once the proposed architecture is designed, the total dataset is divided in the ratio of 80:20 for training and validation, yielding 16,905 images for training and 4227 images for validation. This process is repeated using a k-fold validation (k = 3). To begin the training process, all the hyperparameters are initially set to their default values as shown in Table 1. The training procedure generated 1057 iterations for each epoch. Finally, a Tesla P100 GPU and PyCharm framework were used to run the recommended model on the Google Colab Pro platform for 50 epochs.

4.4 Model training and loss function

To begin the source model training, pre-trained COCO weights were used. These weights, however, were not appropriate to our data. Equation 7 was used to change the weight values for each iteration,

$$Weight_{Updated} = Weight_{Previous} - LR\frac{\partial Loss}{\partial Weight_{Previous}} \tag{7}$$

$$LOSS(Proposed\_Model) = LOSS_{Box} + LOSS_{confidence} + LOSS_{Class} \tag{8}$$

where $Weight_{updated}$ stands for the weight values after training, $Weight_{Previous}$ for the weight values before training, LR

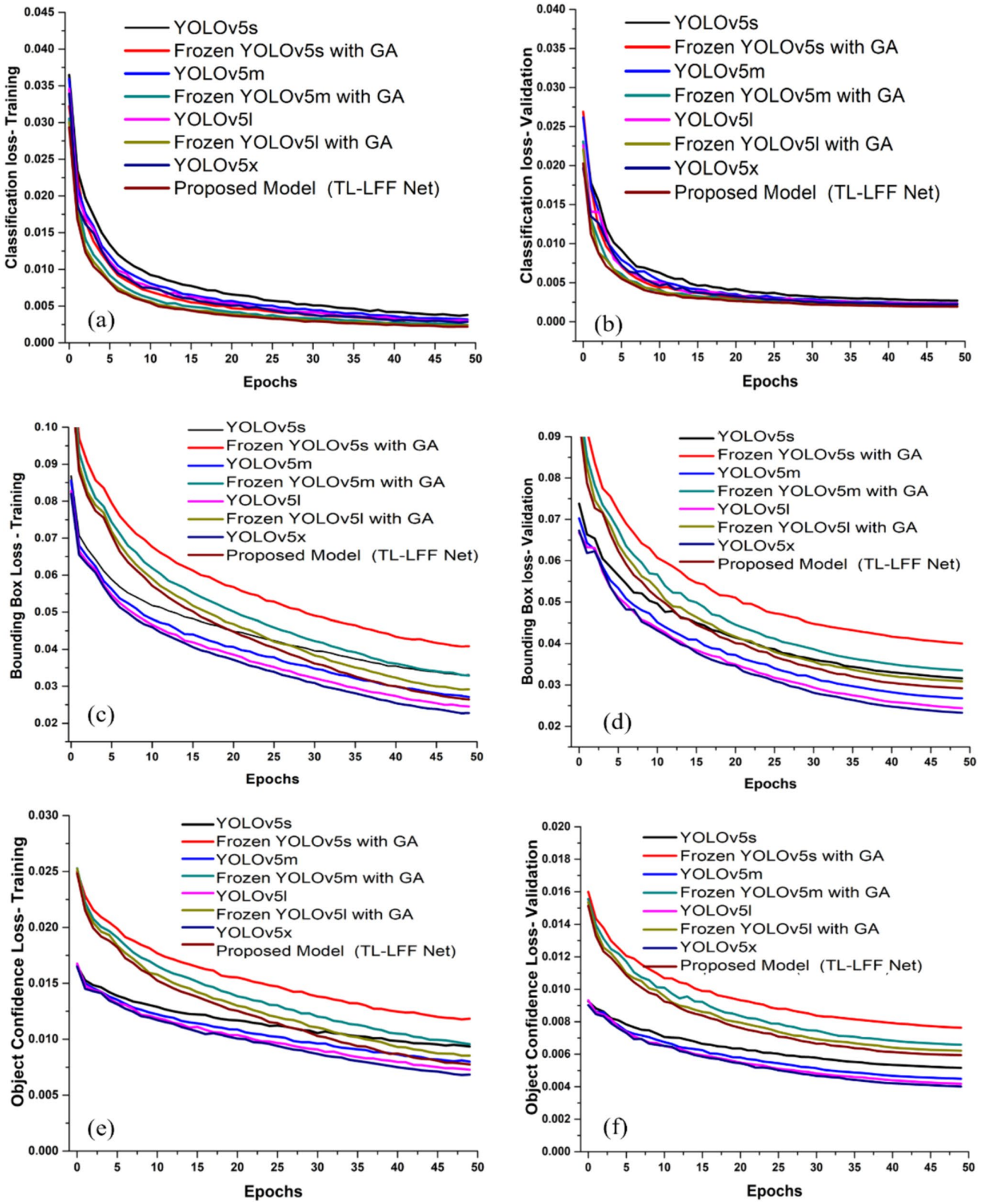

Fig. 8 Three individual losses for a given 50 epochs during training and validation

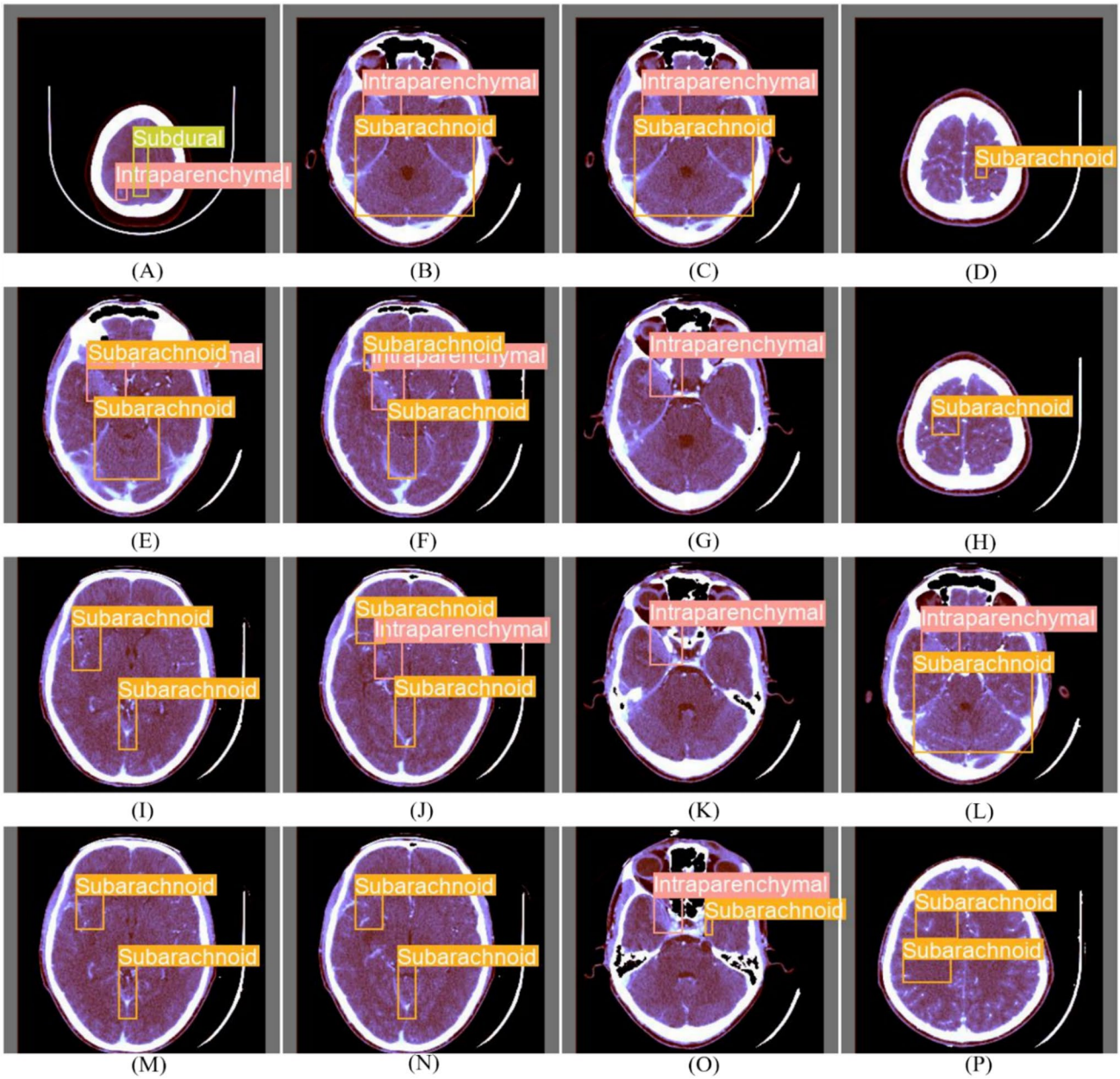

Fig. 9 View of ground truth images along with particular hemorrhage class names

stands for learning rate, and Loss is the sum of three separate functions as shown in Eq. 8. When the loss value decreased to its lowest point, the training procedure was finished.

Let a specific box labeled "b" experience bounding box loss due to either the box dimensions (width, height) or its location (x_pos, y_pos). Therefore, as stated in Eq. 9, box loss is a combination of two separate terms. The box actual dimensions are represented by the coordinates (x_pos_b, y_pos_b, $width_b$, $height_b$), whereas its estimated dimensions of are represented by ($\hat{x\_pos}_b$, $\hat{y\_pos}_b$, $\hat{width}_b$, $\hat{height}_b$).

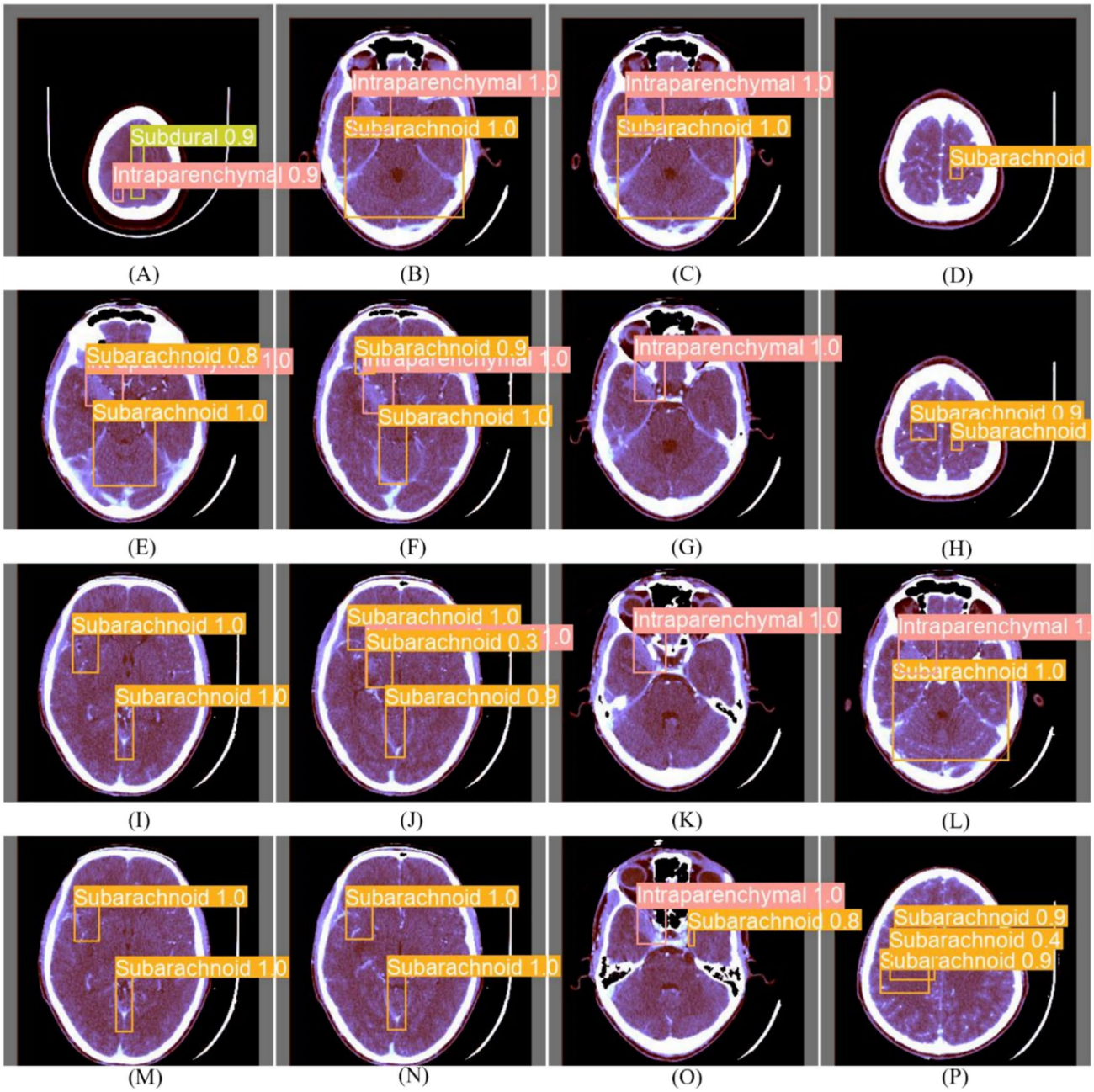

Fig. 10 View of the proposed TL-LFF Network recognition results, along with different hemorrhage class names and confidence score values

$$
\begin{aligned}
LOSS_{Box} = & \theta_{Pos}\left\{\sum_{i=0}^{P^2}\sum_{j=0}^{B}\pi_{ij}^{ICH}\left[\left(x\_pos_b - \hat{x\_pos}_b\right)^2 + \left(y\_pos_b - \hat{y\_pos}_b\right)^2\right]\right\} \\
& + \theta_{Pos}\left\{\sum_{i=0}^{P^2}\sum_{j=0}^{B}\pi_{ij}^{ICH}\left[\left(\sqrt{Width_b} - \sqrt{\hat{Width}_b}\right)^2 + \left(\sqrt{height_b} - \sqrt{\hat{height}_b}\right)^2\right]\right\}
\end{aligned} \tag{9}
$$

where, θ_{Pos} is used to punish incorrect positions and Π_{ij}^{ICH} states that the jth bounding box in an ith grid cell is responsible for prediction and its value will be "1" if the presence of hemorrhage is true, otherwise "0". In this case, a square root was utilized for the height and width values to penalize

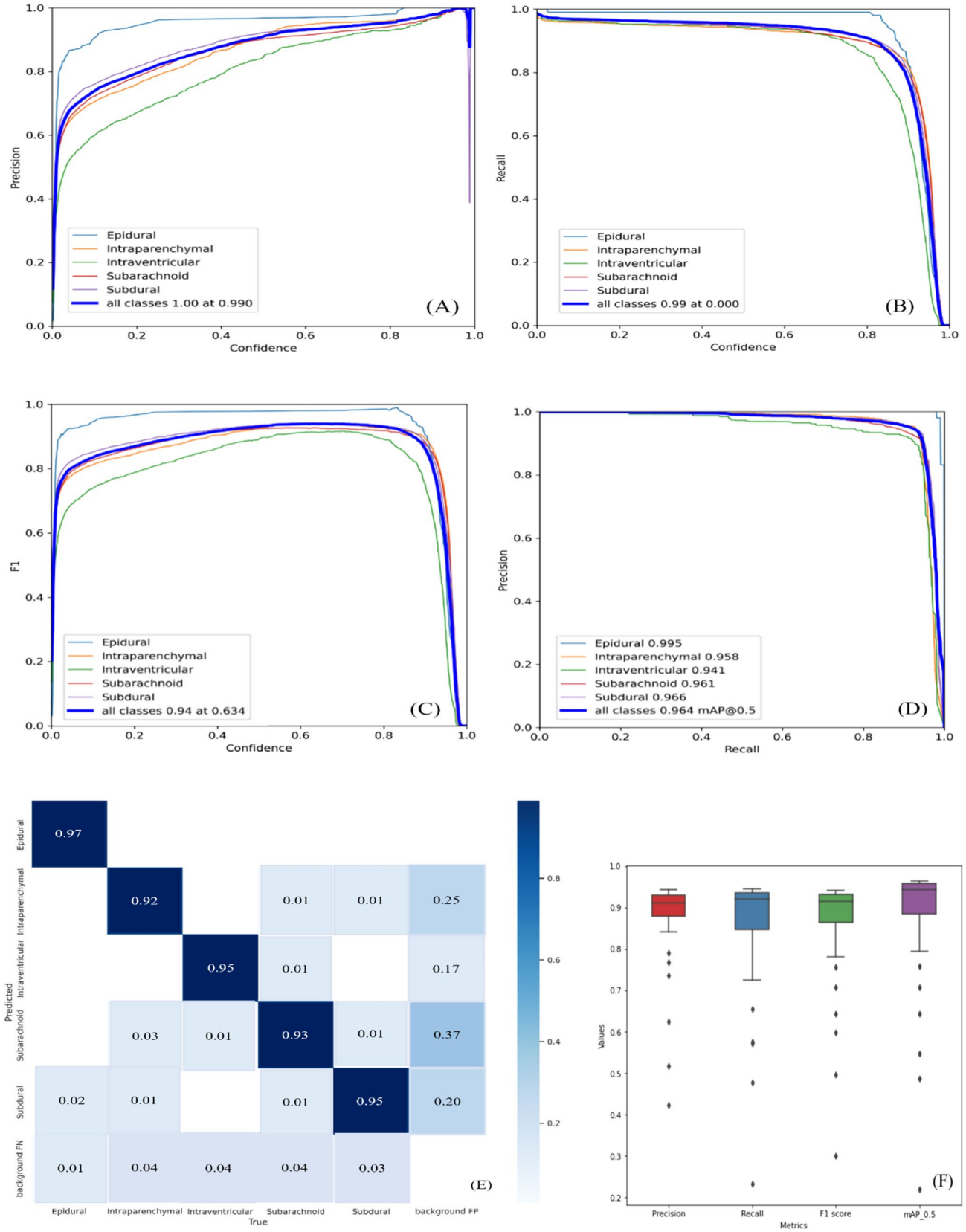

Fig. 11 **A** Precision, **B** Recall, **C** F1 Score, **D** Precision-Recall Plots of the proposed model in terms of confidence score, **E** Confusion Matrix, and **F** Box plot illustrating the proposed model's four metric values

Table 2 Performance metrics such as Precision, Recall, F1-Score, and mAP at different thresholds of the existing and proposed models w.r.t each hemorrhage class

Class name	Precision			Recall			F1-Score			mAP@.5			mAP@.5:.95		
	Existing model	Source model	Proposed model	Existing model	Source model	Proposed model	Existingmodel	Source model	Proposed model	Existing model	Source model	Proposed model	Existing model	Source model	Proposed model
EDH	1.00	0.965	0.971	1.00	0.99	0.990	1.00	0.978	0.980	1.00	0.992	0.995	–	0.722	0.766
IPH	0.97	0.944	0.951	0.86	0.923	0.928	0.91	0.933	0.939	0.85	0.942	0.958	–	0.704	0.744
IVH	0.68	0.903	0.896	0.65	0.91	0.932	0.67	0.906	0.914	0.63	0.925	0.941	–	0.589	0.631
SAH	0.97	0.903	0.919	0.74	0.91	0.935	0.84	0.906	0.927	0.71	0.938	0.961	–	0.708	0.754
SDH	0.97	0.927	0.937	0.80	0.921	0.938	0.88	0.624	0.937	0.79	0.942	0.966	–	0.695	0.747
Overall	0.92	0.928	0.935	0.81	0.931	0.945	0.86	0.929	0.940	0.796	0.948	0.964	–	0.663	0.728

*Existing Model (YOLOv4-TT), Source Model (YOLOv5x), and Proposed Model (TL-LFF Net)

errors in big bounding boxes rather than errors in tiny boxes. The confidence score loss is represented by Eq. 10.

$$LOSS_{Confidence} = \sum_{i=0}^{P^2} \sum_{j=0}^{B} \pi_{ij}^{ICH} \left[\left(conf_i - \overline{conf_i} \right)^2 \right] + \partial_{no-ICH} \sum_{i=0}^{P^2} \sum_{j=0}^{B} \pi_{ij}^{no-ICH} \left[\left(conf_i - \overline{conf_i} \right)^2 \right] \tag{10}$$

where, the value of Π_{ij}^{no-ICH} is the inverse of Π_{ij}^{ICH} , while $\partial_{\text{no-ICH}}$ is equal to θ_{Pos}. The terms conf_i and $\overline{conf_i}$ indicate the actual and estimated confidence scores, respectively. The classification loss was comparable to the usual binary classification loss, as shown in Eq. 11.

$$LOSS_{Class} = \sum_{i=0}^{P^2} \pi_i^{ICH} \sum_{C \in class} \left[Ground_i(c) - Pred_i(\overline{c}) \right]^2 \tag{11}$$

where, the value of Π_i^{ICH} is "1" if there is bleeding in the i^{th} grid cell otherwise it is "0". The predicted output was represented by Predi(c), whereas the ground truth output was represented by $\text{Ground}_i(c)$. After calculating these three losses, the total loss value for each epoch was calculated. The three individual losses during training and validation are generated by Eqs. 9, 10, and 11. Figure 8 depicts the training and validation loss curves as a function of epoch count for the proposed and other comparative models. A stochastic gradient descent (SGD) optimizer is used after each epoch to reduce the loss value to the minimum position.

5 Results and discussion

5.1 Experimental qualitative results

In this study, a lighter architecture based on Transfer Learning (TL-LFF Net) was used to categorize different scales of mixed hemorrhages in a given CT slice. Instead of a traditional segmentation mask, the proposed model used the bounding box technique to select a bleeding zone. The BHX dataset was used to train the proposed architecture, and the accuracy for both training and validation was evaluated using four metrics such as precision, recall, F1 score, and mAP in terms of confidence score. Finally, the performance of the proposed model was evaluated over 50 epochs. Figure 9 depicts the use of 16 images along with their ground truth labels as a batch for predicting the mixed ICH. Out of 16 images, some of the images contain a single hemorrhage class, while others contain multiple ICH during the training phase. Figures 9(D), 9(G), 9(H), and 9(K), for example, show a single hemorrhage, whereas Figs. 9(E), 9(F), and 9(J) show three types of hemorrhages with varying scales, and the remaining images show two hemorrhages.

Table 3 Performance metrics such as Accuracy, TNR, FPR, FNR, and MCC of proposed model

ICH Type	TP	TN	FP	FN	Accuracy	Specificity	FPR	FNR	MCC
EDH	0.97	3.85	0.02	0	0.995	0.994	0.005	0	0.987
IPH	0.92	3.86	0.04	0.02	0.987	0.989	0.010	0.02	0.960
IVH	0.95	3.87	0.01	0.01	0.995	0.997	0.002	0.01	0.987
SAH	0.93	3.89	0.03	0.05	0.983	0.992	0.007	0.05	0.948
SDH	0.95	3.89	0.02	0.04	0.977	0.994	0.005	0.04	0.961
Average	**0.944**	**3.872**	**0.024**	**0.024**	**0.987**	**0.993**	**0.005**	**0.02**	**0.968**

Table 4 Results of threefold cross-validation of the proposed model

TL-LFF Net	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Fold 1	0.928	0.807	0.881	0.721
Fold 2	0.921	0.810	0.875	0.735
Fold 3	0.917	0.805	0.867	0.729
Average	0.922	0.807	0.874	0.728

Figure 10 shows the final predictions of the proposed model, along with a confidence score value and class name. A confidence score of 1.0 indicates that the proposed model can predict the specific type of hemorrhage with 100% accuracy. Despite the different scales, each hemorrhage in Fig. 10 is accurately predicted, along with its label names and bounding boxes. Based on the results, the proposed model has a high detection probability for multi-scale and mixed hemorrhages. As a result, the proposed model achieves the best recognition performance with perfect detection. The only disadvantage is that a few images 9(H), 9(J), and 9(P) are predicted with a false positive value. In any case, the confidence score of false positive predictions was much lower in value when compared to the actual predictions, so we can ignore those predictions to achieve perfect detections.

5.2 Experimental quantitative results

Figure 11 illustrates the proposed model confidence score as a function of performance measures such as precision (Fig. 11A), recall (Fig. 11B), F1-Score (Fig. 11C), and a PR curve (Fig. 11D) for each ICH class independently during the training stage. The EDH class has the best detection results across all metrics, and the PR curve shows that the proposed model has the best precision at a tolerable recall for all classes throughout training. Despite having fewer input samples than any other type of hemorrhage, EDH had the highest precision, while intraventricular hemorrhages had the lowest. This is due to the fact that the EDH class is relatively easy to detect, whereas the IVH class is surrounded by various brain tissues, making the model difficult to detect. Figure 11(E) shows the confusion matrix for five different classes using a heatmap. When compared to true predictions, the percentage of background false positive and false negative predictions is very small. The boxplot

Table 5 Comparison of hardware requirements, number of calculations used in conventional and proposed models, and other parameters used to determine the best model

Name of the model used	Layers	Params (Million)	FLOPS	Frames per second (FPS)	Iterations per second (IPS)	GPU Memory used (GB)	Avg. time of an epoch (Min)	Total execution time (hrs)
YOLOv5s	270	7.033	15.9	187	11.65	0.751	1.30	1.490
YOLOv5s- GC	453	3.695	8.10	165	10.27	0.629	1.42	1.540
Frozen YOLOv5s	270	7.033	15.9	242	15.14	0.415	1.11	1.311
YOLOv5m	369	20.88	48.1	94	5.86	1.400	3.00	2.851
YOLOv5m—GC	695	8.542	18.4	123	7.67	1.170	2.15	2.140
Frozen YOLOv5m	369	20.88	48.1	223	13.92	0.614	1.16	1.435
YOLOv5l	468	46.15	108.0	60	3.77	2.240	4.41	4.320
YOLOv5l -GC	937	15.61	33.3	95	5.92	1.770	2.56	2.650
Frozen YOLOv5l	468	46.15	108.0	194	12.14	0.966	1.25	1.611
YOLOv5x	**567**	**86.24**	**204.3**	**33**	**2.10**	**3.780**	**8.22**	**7.752**
YOLOv5x -GC	**1179**	**25.08**	**53.3**	**59**	**3.68**	**2.350**	**4.47**	**3.320**
Proposed model	**567**	**86.24**	**204.3**	**179**	**11.20**	**1.490**	**1.36**	**1.923**

GC Ghost convolution

Table 6 Comparison of precision, recall, F1-Score, and mAP values for conventional, YOLO, and proposed models

Model name	F=0	F=10	F=24	GA	M	GC	mAP@0.5	mAP@0.5:0.95	FPS	IPS
YOLOv5s	√	×	×	×	×	×	0.585	0.262	208	13.01
	√	×	×	×	√	×	0.909	0.550	187	11.65
	√	×	×	×	√	√	0.873	0.510	165	10.27
	×	√	×	×	√	×	0.712	0.370	204	12.77
	×	√	×	√	√	×	0.928	0.594	204	12.77
	×	×	√	×	√	×	0.910	0.556	242	15.14
	×	×	√	√	√	×	0.938	0.597	242	15.14
YOLOv5m	√	×	×	×	×	×	0.633	0.291	165	10.31
	√	×	×	×	√	×	0.920	0.617	94	5.86
	√	×	×	×	√	√	0.915	0.572	123	7.67
	×	√	×	×	√	×	0.822	0.465	151	9.44
	×	√	×	√	√	×	0.949	0.667	151	9.44
	×	×	√	×	√	×	0.936	0.627	223	13.92
	×	×	√	√	√	×	0.957	0.670	223	13.92
YOLOv5l	√	×	×	×	×	×	0.693	0.336	125	7.81
	√	×	×	×	√	×	0.934	0.638	60	3.77
	√	×	×	×	√	√	0.929	0.608	95	5.92
	×	√	×	×	√	×	0.854	0.491	134	8.36
	×	√	×	√	√	×	0.961	0.699	134	8.36
	×	×	√	×	√	×	0.951	0.665	194	12.14
	×	×	√	√	√	×	0.962	0.709	194	12.14
YOLOv5x	√	×	×	×	×	×	0.721	0.349	76	4.75
	√	×	×	×	√	×	0.948	0.663	33	2.10
	√	×	×	×	√	√	0.931	0.624	59	3.68
	×	√	×	×	√	×	0.877	0.529	126	7.85
	×	√	×	√	√	×	0.962	0.716	126	7.85
	×	×	√	×	√	×	0.956	0.681	179	11.20
TL-LFF Net	×	×	√	√	√	×	**0.964**	**0.728**	**179**	**11.20**

*F=0 (No Freeze), F=10 (Backbone Freeze), F=24 (Backbone and Neck Freeze), GA (Genetic Algorithm), M (Mosaic), and GC (Ghost Convolution)

representation of the final detection results after 50 epochs of the proposed model as shown in Fig. 11(F).

Table 2 compares the proposed model, the source model, and the existing YOLOv4 model on a class-by-class basis. The performance metrics for each of these three models were provided separately. When compared to the existing YOLO4, the sources model achieves a precision of 0.928, a recall of 0.931, and mAP@0.5 of 0.948, which is relatively higher. Later, the concept of freezing was added to the proposed model to achieve the lightweight nature. However, the detection capability was reduced with these modifications. As a result, the concept of GA with transfer learning was used in the proposed TL-LFF network to improve the results even further. As a result, the detection results improved to 0.935, 0.945, and 0.964 for precision, recall and mAP, respectively. The BHX dataset contains five different types of hemorrhages, so results for each class are presented independently in Table 2. Finally, the proposed model class-wise mAP scores 0.995, 0.958, 0.941, 0.961, and 0.966, for EDH, IPH, IVH, SAH, and SDH, respectively were obtained.

In addition to the standard metrics, we calculated other metrics such as Accuracy, Specificity or True negative rate (TNR), False positive rate (FPR), false negative rate (FNR), and Matthew's correlation coefficient (MCC) with the help of confusion matrix. For a perfect classification model, the values of accuracy, specificity, and MCC should be "1". The values of FPR and FNR should be "0". The formulae to calculate the above metrics are given in Eqs. (12–16) and the results are summarized in Table 3.

Table 7 Comparison of precision, recall, F1-Score, and mAP values for conventional, YOLO, and proposed models

Name of the model	Precision	Recall	F1-Score	mAP@.5	mAP@.5:.95
DenseNet121 [9]	0.949	0.814	0.872		
MTANS [64]	0.679	0.795			
RFDCR [65]	0.840	0.910			
SSD (512)-FE [46]	0.797	–	0.845	–	–
2D Faster RCNN [42]	0.897	–	0.908	–	–
Faster R-CNN [43]	0.857	0.844	0.85	–	–
R-FCN [43]	0.905	0.826	0.864	–	–
3D-CNN + YOLO: HR [44]	0.619	–	0.747	–	–
3D-CNN + YOLO: LR [44]	0.672	–	0.776	–	–
YOLOv2 + single label [47]	0.609	–	0.695	–	–
YOLOv2 + double labels [47]	0.627	–	0.611	–	–
YOLOv2 + CSF filtering [47]	0.797	–	0.727	–	–
Faster RCNN [66]	0.765	0.691			
YOLOv2 with attention [66]	0.473	0.620			
YOLOv3 [67]	0.770	0.405	0.500	0.527	
YOLOv4 -ST [48]	0.938	0.918	0.928	0.906	–
YOLOv4 -TT [48]	0.920	0.810	0.860	0.796	–
YOLOv5x -TT [63]	0.928	0.931	0.930	0.948	0.663
YOLOv5x-GCB -TT [63]	0.921	0.889	0.900	0.931	0.624
YOLOv5s-CAM [68]	0.935	0.908	0.921	0.943	0.650
Frozen YOLOv5s without GA	0.893	0.860	0.876	0.910	0.556
Frozen YOLOv5s with GA	0.917	0.896	0.906	0.938	0.597
Frozen YOLOv5m without GA	0.919	0.890	0.904	0.936	0.627
Frozen YOLOv5m with GA	0.934	0.917	0.925	0.957	0.670
Frozen YOLOv5l without GA	0.922	0.912	0.916	0.951	0.665
Frozen YOLOv5l with GA	0.932	0.934	0.933	0.960	0.709
Frozen YOLOv5x without GA	0.931	0.924	0.927	0.956	0.681
Proposed Model with GA (TL-LFF Net)	0.935	0.945	0.940	0.964	0.728

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{12}$$

$$Specificity = \frac{TN}{TN + FP} \tag{13}$$

$$FPR = \frac{FP}{TN + FP} \tag{14}$$

$$FNR = \frac{FN}{TP + FN} \tag{15}$$

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{16}$$

To comprehensively evaluate the performance of the proposed model, we employed a k-fold cross-validation approach on the BHX dataset. This method helps to mitigate the risk of overfitting by randomly dividing the dataset into k folds, where k = 3. Each fold represented a distinct training and validation subset, ensuring robust evaluation across different data splits. We then assessed the performance metrics for each fold, providing a more thorough understanding of the models' performance. The k-fold cross-validation results are summarized in Table 4.

Table 5 compares the proposed model lightweight capability compared to the other methods. Various versions such as small, medium, large, and x-large of YOLOv5 are executed by changing the depth and width multiples. The number of layers used in the first experiment, YOLOv5s is 270, resulting in an FPS of 187, IPS of 11.65, GPU memory of 0.75 GB, and a total execution time of 1.49 h. To improve these values, YOLOv5 was implemented with ghost convolution (GC) in the following experiment, resulting in a decrease in detection accuracy [63]. As a result, the proposed model ignored the concept of GC. Following that, the concepts of froze and GA with TL were incorporated to achieve both lightweight capability and improved detection accuracy. FPS was raised to 242 while IPS reached to 15.14 with this technique. Total execution time and GPU memory were reduced to 1.311 h and 0.415 GB, respectively. Based on these findings, the proposed model

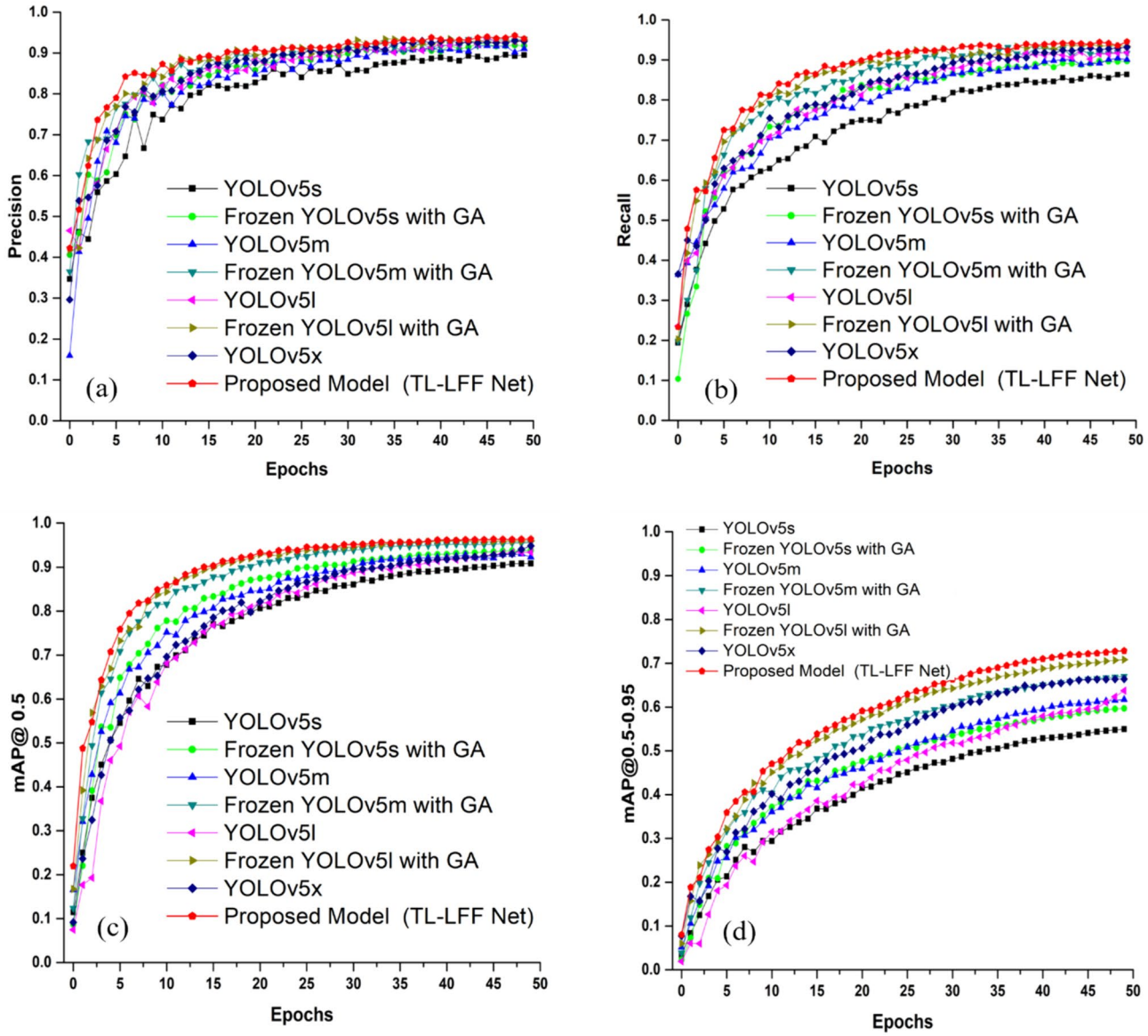

Fig. 12 Detection metrics of proposed and other ablation experimental results for 50 epochs

achieves FPS of 179 from 33, IPS of 11.20 from 2.10, GPU memory of 1.49 GB from 3.78 GB, and total execution time of 1.92 h from 7.75 h without sacrificing detection accuracy.

5.3 Ablation results

Table 6 shows the results of a series of ablation experiments performed on the BHX dataset.

5.3.1 Ablation of different freeze levels

The performances of various pre-trained YOLOv5 models and three levels of freeze settings were compared in this section. The comparison includes four YOLOv5 models such as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. We distinguished the results of models that had the backbone frozen (F = 10), backbone and neck frozen (F = 24), and those that had no layers frozen (F = 0). The performance of various models was evaluated based on mAP, FPS, and IPS, as shown in Table 5. If the F = 0, the YOLOv5x will produce a mAP value of 0.948 at the 0.5 threshold and 0.663 by taking the average of ten different threshold values ranging from 0.5 to 0.95 at a step size of 0.5. However, the FPS and IPS values remained at 33 and 2.10, respectively. However, our main motivation is to develop a lightweight model to detect multi-scale mixed ICH. When we tested with F = 10, the mAP@0.5 decreased to 0.87, but the FPS and IPS values increased to 126 and 7.85, respectively. We experimented

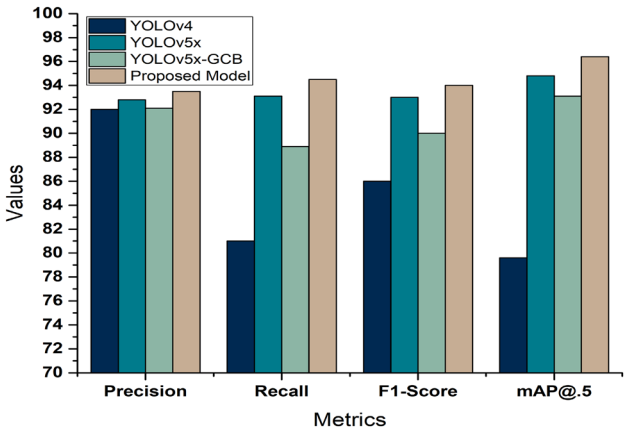

Fig. 13 Comparison among existing model (YOLOv4), Source Model (YOLOv5x), Source Model with GC (YOLOv5x-GCB), and Proposed Model (TL-LFF Net)

with F = 24 to improve these values even further, and they became 179 and 11.20. The disadvantage is observed in the level of freezing; as it is increased, the detection capability is reduced; however, the lightweight model is achieved. To overcome this limitation, we used the GA concept with transfer learning to perform ablation.

5.3.2 Ablation of GA

The GA concept is used to optimize the hyperparameters, and the TL concept is used to improve detection accuracy. As a result, we experimented with various YOLOv5 models, and the results are shown in Table 6. In addition to F = 10, if the GA is used, the mAP values are increased to 0.962 and 0.716, respectively, while the FPS and IPS remain unchanged. So, we also tried F = 24 and got higher mAP values like 0.964 and 0.728 while keeping the FPS and IPS at 179 and 11.20, indicating that we got a lightweight model with improved detection capability.

5.3.3 Ablation of mosaic augmentation

One of the most noticeable aspects of the YOLO model is mosaic. As a result, we conducted a number of experiments to demonstrate the significance of mosaic. If the mosaic concept is not used, the basic YOLOv5x performance drops to 0.721 from 0.948. As a result, we used mosaic augmentation throughout all of the ablations.

5.3.4 Ablation of ghost convolution (GC)

The main aim of this research is to develop a lightweight model; however, we experimented with the GC concept in the YOLOv5 model. The basic idea behind GC is to reduce the number of feature maps by removing unnecessary data. The disadvantage of GC is that detection accuracy is reduced to 0.931 from 0.948. As a result, we did not include it in our proposed model.

5.4 Comparative results

Table 7 compares the proposed model to current state-of-the-art techniques. Initially, the proposed model is compared with the classical models such as DenseNet, MTANS, and RFDCR. These models achieved higher classification and segmentation accuracy but they can't localize the mixed ICH cases. Later, we compared with the two stage detection models such as faster RCNN, RFCN etc. But the detection accuracy, and the inference speed is low with these models. Finally, the proposed model is compared with the single stage detection models such as YOLO series. The findings are mostly compared to the existing YOLOv4 model because the other papers used relatively a lesser number of images for training. The proposed model outperformed YOLOv4 in terms of precision, recall, F1 score, and mAP values, which was trained on the same publicly available BHX dataset. We can conclude from this comparison that the proposed model outperformed all single-stage and two-stage models. Furthermore, reaching the optimal values by taking very little time.

Figure 12 compares the performance characteristics of the proposed model to those of the other YOLOv5 versions. Precision (Fig. 12a), recall (Fig. 12b), mAP@0.5 (Fig. 12c), and mAP@0.5–0.9 (Fig. 12d) all increased over time for all models. When compared to other models, the proposed model achieved the highest precision, recall, and mAP at various thresholds, as illustrated in Fig. 12. Another advantage is that the proposed model was able to provide the best predictions even after a small number of epochs, indicating that the model had quickly reached its optimal convergent state. Finally, Fig. 13 compares the overall results of the existing model (YOLOv4), the Source Model (YOLOv5x), the Source Model with GC (YOLOv5x-GCB), and the Proposed Model (TL-LFF Net). This graph clearly shows that the proposed model achieved the highest detection accuracy for each detection metric.

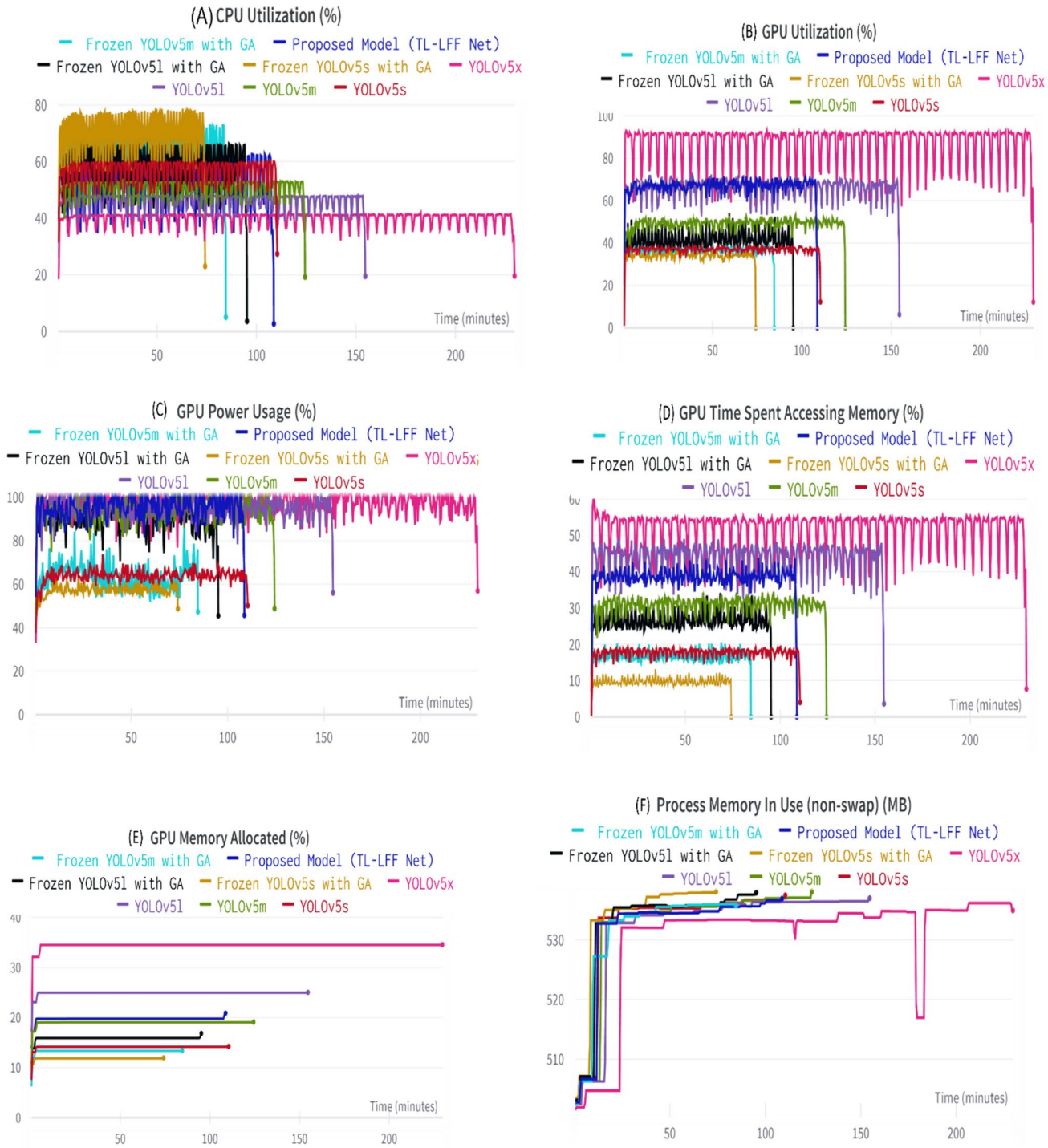

Fig. 14 View of the system utilization parameters of the proposed model using the weight and biases package

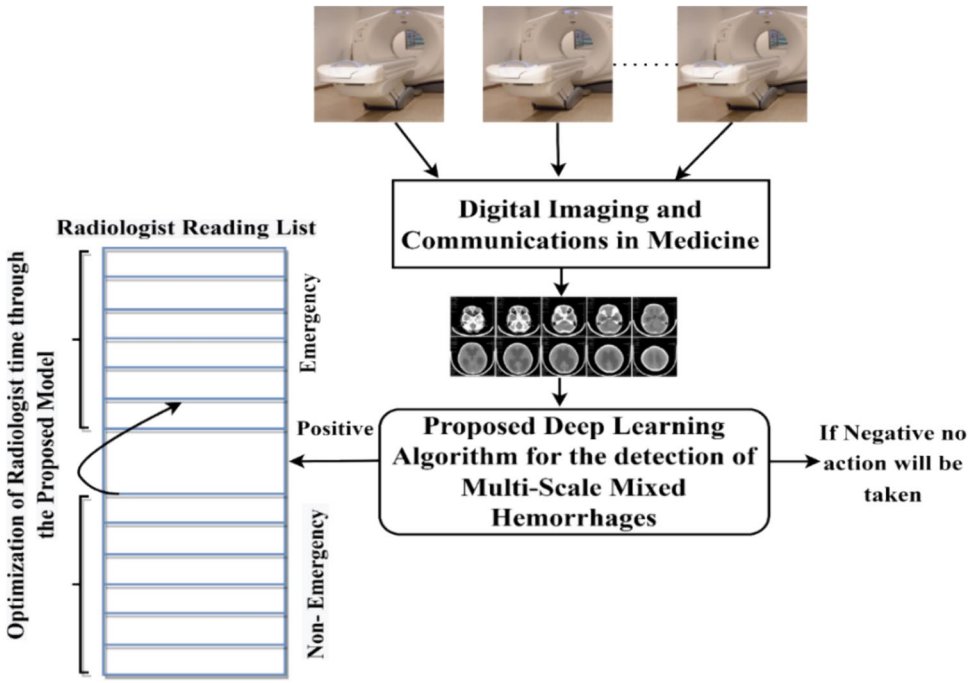

Fig. 15 Total methodology for clinically implementing the proposed deep learning model

5.5 System utilization results

We used the Python programming language's weight and biases library to track and visualize system utilization parameters. Figure 14 compares the four versions of YOLO models such as small, medium, large, and x-large, with their corresponding frozen models. Figure 14A and B show that the amount of CPU and GPU utilization time has been halved for any one of the proposed frozen YOLO series. As a result, the amount of GPU power used to execute the proposed model is also reduced by half during the training process as shown in Fig. 14(C). Similarly, when compared to standard models, the GPU memory, GPU time spent accessing memory, and the amount of process memory in use were all reduced during the execution of the proposed model, as shown in Fig. 14D–H and F Based on these findings, we can confidently say that the proposed model achieved the lightweight without sacrificing detection accuracy.

The following are the suggested model's primary benefits:

1) It can locate multi-scale mixed ICH upon detection.
2) Since less memory is needed, the proposed model could be used for cloud-based real-time clinical diagnostics.
3) It is easier than pixel-wise semantic segmentation and forms a bounding box around each ICH.
4) The proposed model requires less time to execute since it has a higher FPS and IPS.

The primary limitations of the proposed model are given below:

1) The dataset employed to train the proposed model is imbalanced and the number of mixed hemorrhage slices is less in comparison to the total dataset size.
2) The lightweight nature has reduced complexity compared to larger models. While this leads to faster inference times and lower computational requirements, it can limit the model's ability to capture complex patterns, resulting in a potential decrease in detection accuracy.
3) While transfer learning can help improve performance, there is a risk of overfitting, especially when the pre-trained model is significantly different from the target task. In such cases, freezing too many layers might delay the model's ability to adapt to the new task.
4) The drawback of using GA is its computational complexity, especially for large-scale optimization problems. The time required to converge to an optimal solution is long, particularly if the evaluation of each solution is time-consuming.
5) GA itself has several parameters that need to be tuned, such as population size, crossover probability, mutation probability, selection mechanism, etc. Selecting appropriate values for these parameters can be challenging and require additional optimization. The quality of solutions found by GA is influenced by the initial population of individuals. If the initial population is not diverse GA may struggle to explore promising regions effectively.
6) The performance of the model was notably limited in the presence of extremely small hemorrhages, especially those that are below the resolution of the imaging modality.

The proposed model can be extended in the real-time clinical stage by deploying it in the cloud service, where it typically works behind the scenes to optimize the radiologist's assessment time. Radiologists typically place CT scans at the top of their reading list and they can be sorted using a rule-based engine by giving priority to emergency examinations. A data pipeline system transports CT scans to the computer server containing the proposed algorithm, enabling the proposed method to be implemented in real time. The program analyses a particular CT study and produces a binary result (either positive or negative ICH). If the results were positive, the study's priority was upgraded to "emergency" and the radiologist's reading list was promptly revised. However, if the results were not urgent, the study's priority remained unchanged. Figure 15 depicts the overall methodology for implementing the proposed model in the clinical section.

6 Conclusion

The existing deep-learning models were primarily focused on segmentation and classification of multi-class hemorrhages, but they were unable to localize the multi-scale mixed hemorrhages with the highest detection accuracy and lowest system parameters. To achieve this objective, this research proposed a transfer learning-based TL-LFF Network. To achieve the highest ICH detection accuracy, it primarily uses the knowledge obtained from the source model, which is designed with the help of backbone, neck, and head modules. Later, to achieve a lightweight nature, the backbone and neck modules were frozen, while a GA is used to optimize the hyperparameters, resulting in improved detection accuracy. We assessed the performance of proposed model using precision, recall, and mAP at various thresholds on the publicly available BHX dataset. These metrics demonstrate that our model is a good fit for localizing the multi-scale mixed hemorrhage. Later, several system parameters such as GPU and CPU utilization percentages, GPU memory, and so on were used to observe the lightweight nature. These results demonstrated that the proposed model uses fewer resources than existing models. Finally, we conducted several ablation studies to validate the proposed model fine convergence under various conditions, demonstrating that the proposed model outperforms all other state-of-the-art models. The main limitations of the proposed model are the dataset used to train is unbalanced, implementation of GA requires tuning of several hyperparameters, and freezing of layers may leads to missing of some important features from the input images. In future the proposed model can be used in real-time clinical diagnosis by deploying in cloud services because it has a higher detection accuracy in a shorter converge time.

Author contribution All authors contributed to the study conception and design. Material preparation, Methodology, data collection and formal analysis and investigation were performed by [Lakshmi Prasanna Kothala], and [Sitaramanjaneya Reddy Guntur]. The first draft of the manuscript was written by [Lakshmi Prasanna Kothala] Writing – review, editing, Supervision by [Sitaramanjaneya Reddy Guntur], All authors read and approved the final manuscript.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability The data supporting this study's findings is taken from Brain Hemorrhage EXtended (BHX): Bounding box extrapolation from thick to thin slice CT images. (version 1), PhysioNet (2020) (website: https://physionet.org/content/bhx-brain-bounding-box/1.1/).

Declarations

Conflict of interest The author(s) declared that they have no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Qureshi AI, Mendelow AD, Daniel FH (2009) Intracerebral haemorrhage. The Lancet 373(9675):1632–1644
2. Parikh S, Marcella K, Narayan RK (2007) Traumatic brain injury. Int Anesthesiol Clin 45(3):119–135
3. Heit JJ, Iv M, Wintermark M (2017) Imaging of intracranial hemorrhage. J. Stroke 19(1):11–27
4. Chan T (2007) Computer aided detection of small acute intracranial hemorrhage on computer tomography of brain. Comput Med Imaging Graph 31(4–5):285–298
5. Balasooriya U, Perera MS (2012) Intelligent brain hemorrhages diagnosis using artificial neural networks. Business Engineering and Industrial Applications Colloquium (BEIAC) 128–133
6. Hemphil JC, Greenberg SM, Anderson CS, Becker K, Bendok BR, Cushman M (2015) Guidelines for the management of spontaneous intracerebral hemorrhage: a guideline for healthcare professionals from the american heart association/American stroke association. Stroke 46(7):2032–2060
7. Asch CJ, Luitse MJ, Rinkel GJ, Tweel I, Algra A, Klijn CJ (2010) Incidence, case fatality, and functional outcome of intracerebral haemorrhage over time, according to age, sex, and ethnic origin: a systematic review and meta-analysis. Lancet Neurol 9(2):167–176
8. Kothala LP, Guntur SR (2024) An efficient stacked bidirectional GRU-LSTM network for intracranial hemorrhage detection. Int J Imaging Syst Technol 34(1):e22958
9. Kothala LP, Guntur SR (2022) Multi-class classification of intracranial hemorrhages in a 3-channel CT image by using a transfer learning based DenseNet121 model. International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). IEEE Xplore 978–1–6654–5499–5: 1–5
10. Magadza T, Viriri S (2021) Deep learning for brain tumor segmentation: a survey of state-of-the-art. J Imaging 7(2):19
11. Ren L, Heidari AA, Cai Z, Shao Q, Liang G, Chen HL, Pan Z (2022) Gaussian kernel probability-driven slime mould algorithm with new movement mechanism for multi-level image segmentation. Measurement 192:110884

12. Litjens G, Kooi T, Bejnordi BE, Setio AA, Ciompi F, Ghafoorian M, Laak JWM, Ginneken B, Sánchez CI et al (2017) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88
13. Santosh K, GhoshRoy D, Nakarmi SA (2023) Systematic Review on deep structured learning for COVID-19 screening using chest CT from 2020 to 2022. Healthcare 11:2388
14. Piccialli F, Sommaa VD, Giampaoloa F, Cuomoa S, Fortino G (2021) A survey on deep learning in medicine: why, how and when? Inf. Fusion 66:111–137
15. Amicizia D, Piazza MF, Marchini F, Astengo M, Grammatico F, Battaglini A, Schenone I, Sticchi C, Lavieri R, Di Silverio B, Andreoli GB, Ansaldi F (2023) Systematic review of lung cancer screening: advancements and strategies for implementation. Healthcare (Basel) 11(14):2085
16. Xia J, Zhang H, Li R, Wang Z (2022) Adaptive barebones salp swarm algorithm with quasi-oppositional learning for medical diagnosis systems: A comprehensive analysis. J Bionic Eng 19(1):240–256
17. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition
18. Joseph R, Santosh D, Girshick R, Ali F (2016) You only look once: Unified, real-time object detection, IEEE conference on computer vision and pattern recognition 779–788
19. Joseph R, Ali F (2017) YOLO9000: better, faster, stronger, In: Proceedings of the IEEE conference on computer vision and pattern recognition 6517–6525
20. Joseph R, Ali F (2018) Yolov3: An incremental improvement. arXiv abs/1804.02767: 1804. 02767
21. Alexa B, Wang CY, Liao HM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv:2004: 10934
22. Ibrahim MR, Youssef SM, Fathalla KM (2023) Abnormality detection and intelligent severity assessment of human chest computed tomography scans using deep learning: a case study on SARS-COV-2 assessment. J Ambient Intell Human Comput 14:5665–5688
23. Xie P, Zhao X, He X (2023) Improve the performance of CT-based pneumonia classification via source data reweighting. Sci Rep 13:9401
24. Tan W, Liu P, Li X, Liu Y, Zhou Q, Chen C, Gong Z, Yin X, Zhang Y (2021) Classification of COVID-19 pneumonia from chest CT images based on reconstructed super-resolution images and VGG neural network. Health Inf Sci Syst 9(1):10
25. Yuh EL, Gean AD, Manley GT, Callen AL, Wintermark M (2008) Computer-aided assessment of head computed tomography (CT) studies in patients with suspected traumatic brain injury. J Neurotrauma 25(10):1163–1172
26. Li YH, Zhang L, Hu QM, Li HW, Jia FC, Wu JH (2012) Automatic subarachnoid space segmentation and hemorrhage detection in clinical head CT scans. Int J Comput Assisted Radiol Surg 7(4):507–516
27. Phong TD, Duong HN, Nguyen HT, Trong NT, Nguyen VH, Hoa TV, Snasel V (2017) Brain hemorrhage diagnosis by using deep learning. Proceedings of the 2017 International Conference on Machine Learning and Soft Computing 34–39.
28. Thay S, Aimmanee P, Uyyanavara B, Rukskul P (2018) Fast hemorrhage detection in brain CT scan slices using projection profile based decision tree. In: International conference. on Intelligent Information Technology 18–21
29. Zhou Q, Zhu W, Li F, Yuan M, Zheng L, Liu X (2022) Transfer learning of the ResNet-18 and DenseNet-121 model used to diagnose intracranial hemorrhage in CT scanning. Curr Pharm Des 28(4):287–295
30. Vrbancic G, Zorman, Podgorelec V (2019) Transfer learning tuning utilizing grey wolf optimizer for identification of brain hemorrhage from head CT images. In Proc. 6th Student Comput Sci Res Conf 61–66
31. Chen YT, Chen YL, Chen YY, Huang YT, Wong HF, Wang JL, Wang JJ (2022) Deep learning–based brain computed tomography image classification with hyperparameter optimization through transfer learning for stroke. Diagnostics 12(4):807
32. Majumdar A, Brattain L, Telfer B, Farris C, Scalera J (2018) Detecting intracranial hemorrhage with deep learning. 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 583–587
33. Grewal M, Srivastava MM, Kumar P, Varadarajan S. (2018) Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in CT scans. IEEE 15th International Symposium on Biomedical Imaging 281–284
34. Chilamkurthy S, Ghosh R, Tanamala S, Biviji M, Campeau NG, Venugopal VK, Mahajan V, Rao P, Warier P (2018) Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study. The Lancet 392(10162):2388–2396
35. Sage A, Pawel B (2020) Intracranial hemorrhage detection in head CT using double-branch convolutional neural network, support vector machine, and random forest. Appl Sci 10(21):7577
36. He J (2020) Automated detection of intracranial hemorrhage on head computed tomography with deep learning. In: Proceedings of the 2020 10th international conference on biomedical engineering and technology 117–121
37. Farzaneh N, Soroushmehr SR, Williamson CA, Jiang C, Srinivasan A, Bapuraj JR, Ward KR, Korley FK, Najarian K (2017) Automated subdural hematoma segmentation for traumatic brain injured (TBI) patients. In: 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC) 32: 3069–3072
38. Remedios SW, Roy S, Bermudez C, Patel MB, Butman JA, Landman BA, Pham DL (2020) Distributed deep learning across multisite datasets for generalized CT hemorrhage segmentation. Med Phys 47(1):89–98
39. Kuo W, Hane C, Mukherjee P, Malik J, Yuh EL (2019) Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning. Proc Natl Acad Sci 116(45):22737–22745
40. Chang PD, Kuoy E, Grinband J, Weinberg BD, Thompson M, Homo R, Chen J, Abcede H, Shafie M, Sugrue L (2018) Hybrid 3d/2d convolutional neural network for hemorrhage evaluation on head CT. Am J Neuroradiol 39(9):1609–1616
41. Sharrock MF, Mould WA, Ali H, Hildreth M, Awad IA, Hanley DF, Muschelli J (2020) 3D deep neural network segmentation of intracerebral hemorrhage: development and validation for clinical trials. Neuroinformatics 19:403–415
42. Ferlin MA, Michał G, Arkadiusz K, Agnieszka M, Edyta S, Małgorzata G, Sabis A (2021) A comprehensive analysis of deep neural-based cerebral microbleeds detection system. Electronics 10(18):2208
43. Le THY, Phan AC, Cao HP, Phan TC (2019) Automatic identification of intracranial hemorrhage on CT/MRI image using meta-architectures improved from region-based CNN. World Congress on Global Optimization. Springer, Cham, pp 740–750
44. Al-masni M, Kim WR, Kim EY, Noh Y, Kim DH (2020) Automated detection of cerebral microbleeds in MR images: a two-stage deep learning approach. Neuroimage Clin 28:102464
45. Zhang T, Song Z, Yang J, Zhang X, Wei J (2021) Cerebral hemorrhage recognition based on mask R-CNN network. Sens Imag. https://doi.org/10.1007/s11220-020-00322-2
46. Li T, Zou Y, Bai P, Li S, Wang H, Chen X, Meng Z, Kang Z, Zhou G (2021) Detecting cerebral microbleeds via deep learning with

features enhancement by reusing ground truth. Comput Methods Programs Biomed 204:106051
47. Myung MJ, Lee KM, Kim HG, Oh J, Lee JY, Shin I, Kim EJ, Lee JS (2021) Novel approaches to detection of cerebral microbleeds: Single deep learning model to achieve a balanced performance. J Stroke Cerebrovasc Dis 30(9):105886
48. Ertugrul OF, Akıl MF (2022) Detecting hemorrhage types and bounding box of hemorrhage by deep learning. Biomed Signal Process Control 71:103085
49. Mason D (2011) Pydicom: an open source DICOM library. Med Phys 38(10):3493–3493
50. Zhao B, Wu Y, Guan X, Gao L, Zhang B, Boya Z (2021) An improved aggregated-mosaic method for the sparse object detection of remote sensing imagery. Remote Sensing 13(13):2602
51. Kothala LP, Guntur SR (2023) An Improved Mosaic Method for the Localization of Intracranial Hemorrhages Through Bounding Box. IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA) 226–230
52. Mohiyuddin A, Basharat A, Ghani U, Peter V, Abbas S, Naeem OB, Rizwan M (2022) Breast tumor detection and classification in mammogram images using modified YOLOv5 Network. Comput Math Methods Med 1–17:1–16
53. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
54. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. IEEE conf. comput. Vis. Pattern Recognit 8759–8768.
55. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. IEEE Conf. Comput. Vis. Pattern Recognit 2117–2125
56. G. Jocher (2021) "yolov5," https://github.com/ultralytics/yolov5
57. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press
58. Yang XS (2021) Chapter 6 - genetic algorithms," in Nature-inspired optimization algorithms (Second Edition), 2nd edition., Academic Press 91–100
59. Liau YY, Kwangyeol R (2021) Status recognition using pre-trained YOLOv5 for sustainable human-robot collaboration (HRC) system in mold assembly. Sustainability 13(21):12044
60. Liu W, Quijano K, Crawford MM (2022) YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning. IEEE J Sel Top Appl Earth Obs Remote Sens 15:8085–8094
61. Mantau AJ, Widayat IW, Leu JS, Köppen M (2022) A human-detection method based on YOLOv5 and transfer learning using thermal image data from UAV perspective for surveillance system. Drones 6(10):290
62. Reis EP, Nascimento F, Aranha M, Saco FM, Machado B, Felix M, Stein A, Amaro E (2020) Brain Hemorrhage EXtended (BHX): Bounding box extrapolation from thick to thin slice CT images. PhysioNet 215–220.
63. Kothala LP, Jonnala P, Guntur SR (2023) Localization of mixed intracranial hemorrhages by using a ghost convolution-based YOLO network. Biomed Signal Process Control 80(2):104378
64. Chen G, Ru J, Zhou Y, Rekik I, Pan Z, Liu X, Lin Y, Lu B, Shi J (2021) MTANS: multi-scale mean teacher combined adversarial network with shape-aware embedding for semi-supervised brain lesion segmentation. Neuroimage 244:118568
65. Chen G, Li Q, Shi F, Rekik I, Pan Z (2020) RFDCR: Automated brain lesion segmentation using cascaded random forests with dense conditional random fields. Neuroimage 211:116620
66. Nemček J, Jakubíček R (2021) Analysis of circulatory system pathologies in head CT data–hemorrhage localization. Lékař a Tech-Clin Technol 51(1–4):66–70
67. Abdesselam F, Benierbah S, Ferdi Y (2023) YOLOv3-based intracranial hemorrhage localization from CT images. 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE)
68. Vidhya V, Raghavendra U, Gudigar A, Basak S, Mallappa S, Hegde A, Menon GR, Datta P (2023) YOLOv5s-CAM: A deep learning model for automated detection and classification for types of intracranial hematoma in CT Images. IEEE Access 11:141309–141328