[1,a] Aanchal Rani,

[1,b]Preeti Singh,

[1,c]Nidhi Garg,

[1,d]Charu Madhu,

[1,e]Preeti Gupta,

[1,f]Pardeep Kaur,

[2,g]Geeta Dalal

[3,h]Preeti Sharma

A Deep Learning-Based Structural Off-line Approach for the Automatic Recognition of Handwritten Punjabi Characters

Abstract: - Aim: This study employs the YOLOv5 deep learning algorithm to crop handwritten Punjabi Gurmukhi letters from scanned photos automatically. Background: The capacity of a computer to read and comprehend handwritten text is known as handwriting recognition technology. The digitization of society and the rising need for automation have led to considerable growth in this technology. Nearly all languages have handwriting recognition software, yet Punjabi is one language without much of it. Objective: By automating character cropping with YOLOv5, the paper's main goal is to reduce the pre-processing load during handwritten character recognition for Punjabi scripts. It also compares the suggested approach with previous work for the same dataset. Method: Using machine learning and deep learning algorithms, some recent research highlights the importance of automatically cropping letters, words, and sentences for handwriting identification. Many languages, except Punjabi, are available for this extensive undertaking. First, boundary boxes were made using a convolutional neural network (CNN) to segment the letters. Additionally, two optimizers—SGD and ADAMW—were used with YOLOv5 to recognize letters. Result: Punjabi alphabets without diacritics reached a maximum accuracy of 98.1% when implemented using YOLOv5 ADAMW, while Punjabi alphabets with diacritics were implemented using an automatic cropping technique and YOLOv5 SGD, yielding 97.4% accuracy. Conclusion: The study emphasized the significance of optimizer selection and how it affects the recognition of Punjabi characters. The paradigm being explained here provides an opportunity for scholars in this discipline.

Keywords: Handwriting in Punjabi, YOLOv5, Character Recognition, Deep Learning, Machine Learning and Scanned Documents.

I. INTRODUCTION

Notwithstanding the increasing prevalence of internet technology, handwritten documents still play a significant role in society in terms of production and distribution. The conversion of handwritten documents into machine-readable text is a crucial task for handwriting recognition systems, which enable effective information retrieval, analysis, and preservation [1]. Natural Language Processing (NLP) and Computer Vision approaches also commonly use scanned document identification to illustrate the effectiveness of the deep learning (DL) methodology [2]. The capacity to convert handwritten material into digital formats, whether from historical

[1] *Corresponding author: nidhi_garg@pu.ac.in

[1]Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh

[2]Pt. NRS Government College, Rohtak

[3]Department of Electronics and Communication Engineering, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

[a]Sharma0208aanchal@gmail.com

[b]Preeti_singh@pu.ac.in

[*c]nidhi_garg@pu.ac.in(Corresponding)

[d]Charu_uiet@pu.ac.in

[e]Preeti.uiet@gmail.com

[f]Pardeep.tur@gmail.com

[g]Geetadalal1@gmail.com

[h]Preeti.sharma@chitkara.edu.in

Copyright © JES 2024 on-line : journal.esrgroups.org

manuscripts or modern papers, has significant effects on accessibility and usability [3]. Pre-processing, segmentation, feature extraction, and classification are all included in text recognition. The accurate separation and extraction of individual handwritten fragments from a variety of document formats is one of the domain's most significant issues.

Punjabi, like many other Indic scripts, is written in the Gurmukhi script, which consists of 10 vowels and 35 consonants [4]. It has characters that can take on different forms depending on where they are in a word—at the start, middle, or finish. Furthermore, the application of matras, which are vowel modifiers that affix to consonants to change their appearance and sound. Because of this intricacy, segmenting individual characters and the matras that go with them is a difficult operation because algorithms need to precisely identify and separate these components for proper recognition. The identification of handwritten Punjabi letters has been the subject of continuing research. When approaches based on rules were first employed for recognition, their accuracy rate was lower. However, the accuracy and efficiency of recognition have increased with the development of machine learning algorithms. Character recognition in handwriting recognition refers to the methods of finding, classifying, and recognizing characters in an image. The identified text is then transformed into a machine- or digitally-encoded format. Online and offline systems are the two main types of character recognition systems. Real-time picture capturing is facilitated by equipment used in online optical character recognition (OCR) systems. On the other hand, static inputs like scanned copies of handwritten text or digitalized text documents are handled by offline handwritten character recognition systems [5].

The ability to divide specific pieces, such as words, lines, or characters, for simpler processing and analysis makes segmentation a crucial approach in handwriting recognition. Many applications use various segmentation techniques, such as Region of Interest (ROI), Word, Character, and Line segmentation. Finding regions of interest where specific kinds of data are present could be the segmentation process in more complex scenarios. It may be necessary to isolate handwritten portions or areas of interest for additional examination when identifying handwriting. Next, a feature extraction method supplies a set of pertinent characteristics to be used as the input for classifiers and other machine learning models. Depending on the elements that are taken out of the handwritten data, classification entails labeling various characters or symbols. Assessing whether a certain image contains the English letters "A," "B," or "C," for instance. In addition to deep learning-based schemes, conventional schemes such as region-based, textured-based, stroke-based, and sliding window-based systems can also be used for text detection [6]. As technology has progressed, greater outcomes in this domain have been reported for both deep learning and machine learning approaches. Employing multi-module machine learning models with gradient-based learning techniques is at the heart of the machine learning discipline [7]. Furthermore, a variety of pertinent methods for enhancing Punjabi character segmentation and recognition's precision and effectiveness are provided by developments in deep learning and computer vision. A common component of deep learning schemes is the object detection method. In the domain of segmentation and handwriting recognition, YOLOv5 (You Only Look Once version 5) is a revolutionary deep learning technique that dramatically reduces complexity and preparation time [8]. Due to the advancements in the inherent capabilities of YOLOv5, it is able to handle the variability and intricacies of handwritten text more effectively than conventional techniques. In this research work, the YOLOv5 algorithm is implemented and compared with pre-existing technique in terms of accuracy. Some of the issues faced by the researchers in recognition of Punjabi letters are mentioned below:

- Most research into handwriting recognition has been centered on widely used scripts like Latin, Chinese, English or Arabic. There is a significant gap in studies focusing specifically on the Gurmukhi script, leading to a lack of tailored solutions for Punjabi handwriting recognition.
- The Gurmukhi script, with its connected characters and specific strokes, poses a challenge for effective segmentation. Current algorithms, often struggle to accurately segment individual characters.
- Previous systems do not effectively crop the characters, leading to inaccuracies in character recognition.
- Several preprocessing stages are required for the effective segmentation of characters which leads to more time consumption to execute recognition process.
- Existing methods require manual intervention for segmenting handwritten texts into smaller manageable fragments.

The manual segmentation is time-consuming, labor-intensive, and prone to human error. Therefore, there is a need for an automated solution that can accurately crop handwritten Gurmukhi text fragments from larger documents or images, preparing them for efficient and accurate recognition. This paper mainly focuses on the recognition of offline handwritten Gurmukhi script which is scanned through a camera. The objective of this research are: a) To reduce pre-processing steps in handwritten character recognition techniques for Punjabi scripts by automating character cropping using YOLOv5. b) To improve the classification and recognition of the Punjabi handwritten characters. c) To analyze and compare the proposed algorithm with the existing work for the same dataset.

The study aims to contribute to the improvement of handwriting recognition systems by proposing and evaluating automatic cropping techniques using state-of-the-art object detection algorithms. The sections of this research paper are divided further as related work (section 2), materials and methods (section 3), results (section 4) and conclusion (section 5).

II. Related work

This section highlights a variety of segmentation and handwriting recognition techniques. Currently, deep learning algorithms are implemented in characters, words, sentence recognition of different languages [9-28] as well as in various other fields, like data mining [29], leaf disease detection [30-31], human disease diagnosis [32] etc.

Zhao and Liu [9], demonstrated the recognition of handwriting digits, where 0–9 digits are considered as a class. In this research work, the proposed framework includes features based on a CNN extracted from the MINST database as well as integrating algebraic multi-dimensional components with different sets of features adjusted by selected features used in the original feature set of CNN. Results display that segmentation reaches 98% of grading accuracy. **Ahlawat** et al**.** [10], created a hybrid model using the CNN and SVM for the classification of handwritten digits using MNIST databases. In this hybrid model, SVM serves as a dual phase and CNN serves as the default feature key. The MNIST database includes a variety of digitally altered manuscripts. These handwritten digits are automatically stripped of their most distinguishing characteristics by CNN's reception field. The test results demonstrate the usefulness of the suggested framework by outperforming MNIST handwritten digital datasets with a 98.88% testing recognition accuracy rate. **Ali** et al. [11], presented the characterization of the character from the handwritten pictures for the pattern recognition. This proposed effort aimed to provide high accuracy and quick calculations for handwritten numerals in order to offer a digital manner of transparency. In comparison to earlier proposed systems, the aforementioned technology successfully delivers 99.21% better accuracy. **Wang** et al. [12], presented the training model for digital recognition employing the well-known near-k neighbor technique. By choosing the value of K and using a similarity or calculation scale, it is possible to classify known digital images. However, the complexity and time considerably increases with the corresponding value and the search value of K as the amount of data employed exceeds a particular threshold. **Kamoona** et al. [13]**,** worked on the image enhancement algorithm using the enhanced cuckoo search optimization utilizing the random image dataset and obtained the promising outcomes with a new range of search space and did comparative analysis with the other bio-inspired optimization algorithms. The presented algorithm efficiently overcomes the limitations of the histogram equalization approach. **Albahli** et al. [14], introduced an advanced Faster-RCNN where DenseNet-41 was introduced to calculate in-depth features. Finally, the regressor layer and subdivision are used for local performance and the division of the digits into ten classes. The proposed method successfully combines digits from the inserted image and divides them into classes. **Hamid** et al. [15], introduced a method of digital recognition on the MNIST website using CNN, SVM and Multilayer preceptor. In this work, KNN and SVM accurately predicted the results in data sets but due to non-convex performance, the multilayer preceptor failed to detect digital 9. **Kavitha** et al. [16], introduced a handwriting recognition system using a Tamil alphabet data set introduced by HP Labs by using HTCR offline bench marking which showed a good recognition rate of 95.16%, which is significantly higher than that of conventional approaches. **Dai** et al. [17] achieves competitive accuracy but needs the most accessible box annotations. The proposed method, called "BoxSup", generates the outcomes of a tightly controlled box along with robust key lines that are completely covered in the same environment. BoxSup continues to generate the most recent results in PASCAL VOC 2012 and PASCAL-CONTEXT using a huge number of binding boxes. **Xu** et al. [18], proposed a cutting-edge network based on object acquisition. The first phase, Faster R-CNN, decreases the number of candidate areas to a minimum before moving on to the second phase. YOLOv3 necessitates processing a sizable number of candidate items in the image. Due to extreme class inequality, this technique processes information more quickly and decreases accuracy. **Mondal** et al. [19],

researchers worked on an object recognition model for recognizing handwriting to identify English handwritten text using YOLOv3 without a dictionary and obtaining sequence and identification. Preliminary annotations were initially created to identify an intriguing area than detailed features were calculated utilizing the cutting-edge Faster-RCNN, where DenseNet-41 was first introduced. The localization and categorization of the digits into ten classes is done lastly using the regressor layer and classification. **Khan** et al. [20], suggested about the digits of the expiration date and the model was successful tested. The model can be used in smart-expiry architecture and can remove the overhead barcode overhead days. An in-depth learning network, built in Python used with the Camera Library. **Md** et al. [21], demonstrated the digits and handwriting of Bangla recognition as a complex computer vision problem. The two popular data sets, BanglaLekha-Isolated and NumbtaDB were added to the CNN model using a conversion method that also performs many vowel tests, digits, and characters. Accuracy of 96.42% and 98.92% were received for BanglaLekha augmented and the NumtaDB database respectively. **Pareek** et al. [22], worked on Gujarati-language with an offline HCR system using artificial intelligence. 10,000 image sizes from a total of 250 persons, of different ages, of diverse activities, is a significant addition to data collection in this study. Multi-Layer Perceptron (MLP) and CNN model were implemented and a success rate of 64.48% and 97.21% were achieved respectively. **Islam** et al. [23], worked on extracting a region of interest (ROI) from natural scenes. Here, all the Bangla words are extracted from the sentence utilizing bounding box technology, connected component (CC) approach, and parsing. The accuracy of the proposed algorithm results in character extraction and the ROI is 93.23% and 92.70%, respectively. The average accuracy is 99.16% was achieved. **K. B. Neelimaa** et al. [24], developed a handwritten and machine printed alphabet recognition and classification method. The suggested model achieves 97.6% classification accuracy. The dataset was meticulously segmented manually, focusing on both word and character levels. This involved carefully cropping the images of the documents. **H. M. Balaha** et al. [25], presented an algorithm to detect and crop Arabic handwritten characters. The CNN system with two architectures (HMB1 and HMB2) is used to recognize the characters. **M. Kumar** et al. [26], worked on a method for recognizing handwritten texts in multiple languages, specifically targeting English, Hindi, and Punjabi. For script identification, this system is distinguished by its use of a unique combination of features. Special features include zoning, diagonal features, horizontal peak extent, and a novel approach involving intersections and open endpoints. To classify these features effectively, three different classifiers, like k-NN, Linear-SVM,and MLP were employed. Awful results were achieved with the recognition rate of 92.18% for English, 84.67% for Hindi, and 86.79% for Punjabi. **M. Kumar** et al. [27], employing power curve fitting-based features and achieved an accuracy rate of 97.14% and 94.29% for SVM with a linear kernel and SVM with polynomial and RBF kernels respectively. Interestingly, the research utilized a data split, 99% of the data for training and only 1% for testing (referred to as Strategy k). The research reached its utmost accuracy of 98.09% when applying an 85%-15% training-testing data split (Strategy h). **M. Kumar** et al. [28], applied a boundary extent based technique as a feature extraction method for Gurmukhi character recognition. Neural Network (NN) was used as the main classifier with SVM classifiers. The remarkable recognition accuracies with the CSA-based feature selection technique recorded for upper-zone as 88.3% , for the middle-zone was 95.2%, and 91.3% for lower-zone characters.

From the recent work done in this field, it is evident that various techniques and algorithms are extensively in use for the text detection and segmentation of handwritten text for various languages. Many researchers are working on the improvement of recognition rate by utilizing neural networks, machine learning, and hybrid models. English alphabets are recognized with an accuracy as high as 99.21% by using sophisticated algorithms like CNNs and databases like MNIST. However, handwriting recognition in Punjabi Gurmukhi script faces challenges due to the script's complexity and the variability in individual handwriting styles. Moreover, very few researchers are working on this language. Additionally, it has been observed that the previous work highlights the need for more effective automated techniques in Gurmukhi script recognition by addressing the problems faced in segmentation, cropping, and time-consuming pre-processing stages.

III. METHODOLOGY

The journey of deciphering handwritten text through computational means is a multifaceted challenge that encompasses various stages, from data collection and pre-processing to algorithm development and performance evaluation. Each of these stages requires specific tools and methodologies, carefully selected and applied to suit the unique challenges posed by the variability and complexity of handwritten text.

3.1 Dataset: Dataset includes handwritten Punjabi Gurmukhi letters images [33]. The dataset contains a total of 323 images of handwritten Punjabi Gurmukhi alphabets. Each image contains 35 alphabets and 12 vowel diacritic. So, total number of rectangles that are needed to crop is 15,181 scanned pictures from A4 sheets comprise of different sized rectangles which bound handwritten Punjabi Gurmukhi letters.

3.2 LabelImg tool: To segment the handwritten characters with the best results, segmentation by cropping of each handwritten character is carried out. This cropping process is automated using neural networks. Initially, the digitized images of a dataset are annotated with bounding boxes and converted into XML files. For every marked object in the picture, these XML files include bounding box information as well as other pertinent details. These annotated bounding boxes serve as ground truth data that the YOLOv5 model uses to learn how to predict bounding boxes for object detection during the training process. Thus, the images and these XML files are input to train the YOLOv5 model

3.3 Object Detection Algorithms

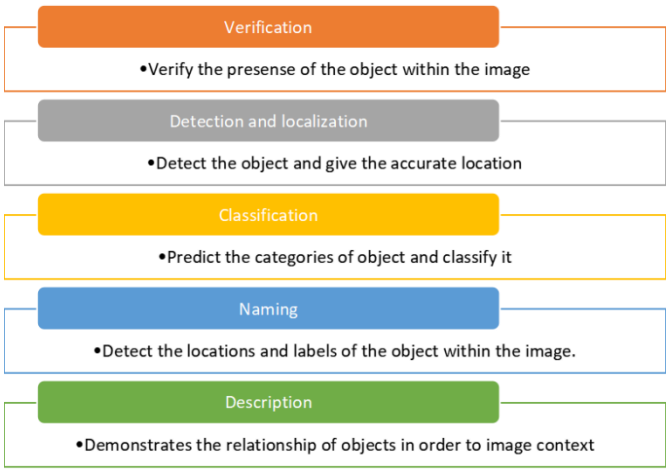


Figure 1: Phases of object detection in Image Recognition

The technique of identifying a class event object component and finding an physical entity's position by removing the bounding box around it is called object detection [34]. Single class object discovery is the process of extracting a single class event from an image, whereas multi-stage object discovery is the process of extracting the categories of every object contained in the image. A number of obstacles must be overcome during the acquisition of an object, including partial or complete closure, variable lighting conditions, placement, scale, etc. Detecting objects is an important stage in any visual activity, as illustrated in Figure 1.

Deep CNNs have been used comprehensively to find the object. CNN is a type of neural transmission network and operates with the goal of weight sharing. Combining layers does not change. Work maps are provided as input into integration layers. They work on each piece on a selected map.

3.3.1 Fast R-CNN

The Fast Region-Based Convolutional Network technique, often known as Fast R-CNN, is a training approach for object detection that was created in Python and C++ (Caffe). These algorithms primarily improve R-CNN speed and accuracy while fixing the shortcomings of these algorithms.

3.3.2 Faster R-CNN

Regional-based convolutional neural networks are the process of obtaining something from previous HOG [35-36] and SIFT methods. For R-CNN models, the most important features (usually about 2000 features) are extracted using selected features. The process of selecting the most important releases can be calculated with the help of a selected search algorithm that can achieve these most important regional suggestions.

3.3.3 Single Shot Detector (SSD)

The single-shot detector (SSD) is one of the quickest methods for gaining access to real-time computation workloads is the one-box multiple-box detector [36]. Although Faster R-CNN approaches can achieve high predicted accuracy, the entire process is time-consuming and necessitates real-time work to run at approximately 7 frames per second, which is not ideal. This issue is resolved by the SSD, which increases frames per second nearly five times over the Faster R-CNN model. The regional proposal network is no longer used; instead, multi-dimensional features and default boxes are used.

3.3.4 YOLO

YOLO trains with full images and directly enhances the performance of the findings [8,37]. The YOLOv5 framework is a single step, proposed to predict binding boxes and class opportunities simultaneously from the input image. This framework separates the formation of an input network input to predict 3 binding picture boxes in G × G cells that are used in the full 2D of each cell. Apart from this, if the cell contains an object, the chances of class C with conditions are limited. In this way, the network predicts intermediate links (x, y), width and length (w, h) and the verification score for each connection box. Unlike other versions of YOLO, YOLOv5 predicts the opposing points in each binding box using a method that separates multiple labels based on structure rather than soft-max for a better data model.

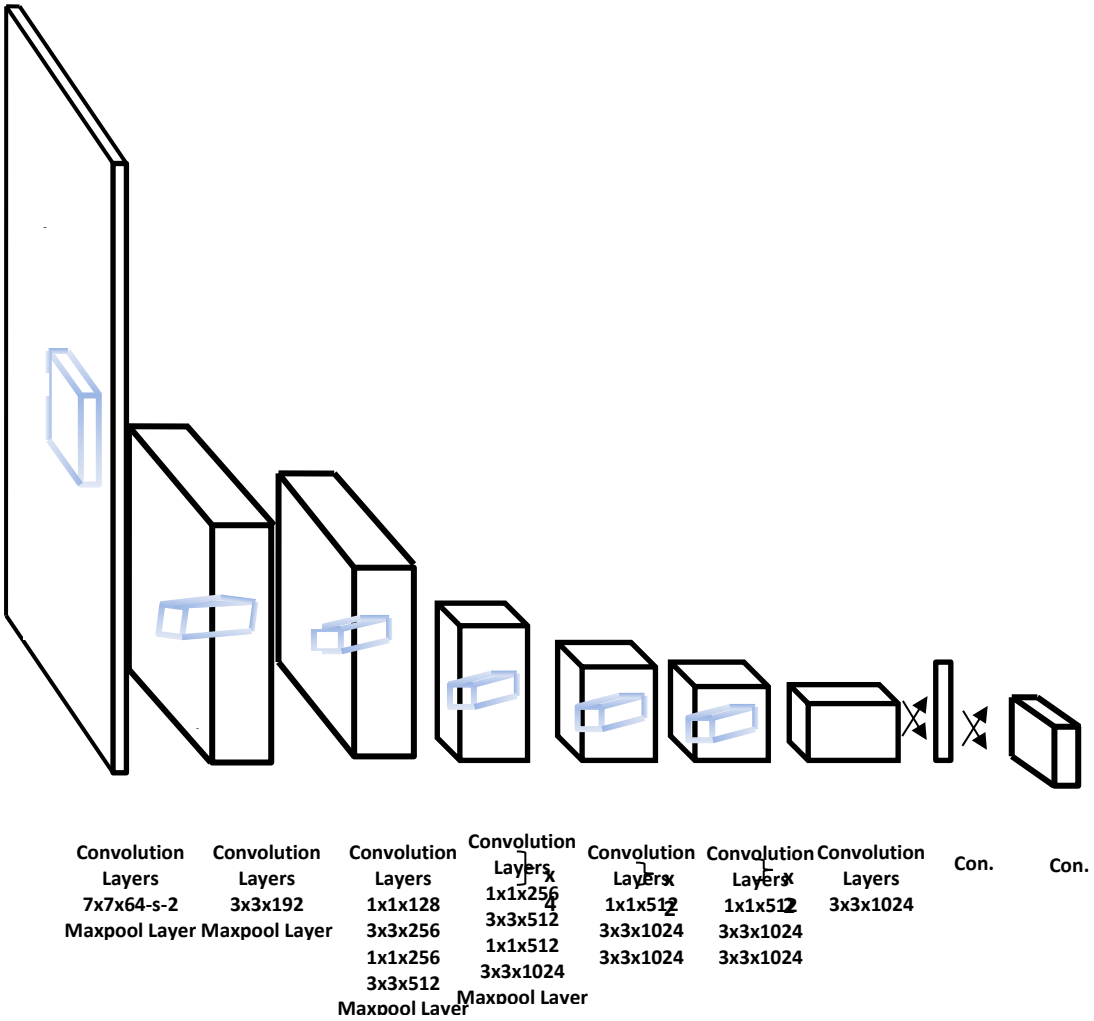


Figure 2: YOLO Architecture

3.3.5 Motivation choice of YOLO model

Yolo is open-source, publicly available, and accurate in real time, it was selected over other models. The entire image has been considered rather than just the region of interest in YOLOv5, unlike other state-of-the-art object recognition systems. As a result, background errors decreased. One of the existing best object recognition techniques is Fast R-CNN, which provides high-fidelity results but they are slow in processing as finding the right

bounding box region, classifying these regions, and then post-processing the result to improve it all took several steps.

3.3.6 Methodology

The research methodology involves several structured phases, each crucial to the development of a robust system capable of accurately identifying and processing handwritten characters and fragments. The proposed methodology framework is shown in Figure 3. Due to its fast processing rate, sound accuracy and open-source nature, YOLO is still the most popular algorithm for object recognition [54].

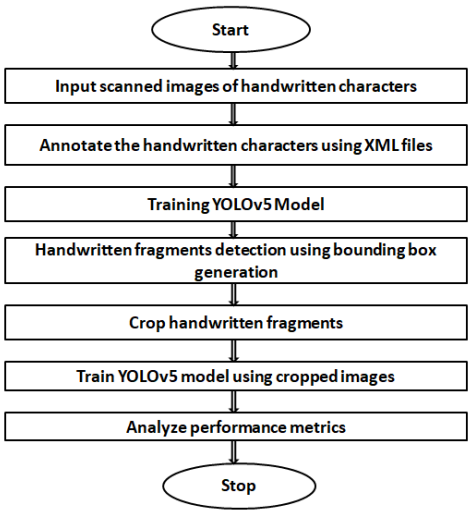


Figure 3: Flowchartof Research Methodology

After the image has been divided into a grid, each cell predicts the number of bounding boxes, which is represented by the variable. The object inside a fully enclosed grid cell must be detected by the grid cell. Confidence scores are predicted for every box, indicating the model's level of assurance in determining if the box actually borders an item and in identifying what the object is. The model architecture is similar to the previous versions but includes some changes like it is written in PyTorch framework so it can easily be accessed in Python. For example, it is now implemented in the PyTorch framework, making it easier to access in Python. A data loader is used to enrich data training for the YOLOv5 update, specifically for scaling, color space modifications, and mosaic augmentation.

The first step involves gathering a large dataset of scanned images containing handwritten characters. These must be very high-quality photos with a variety of handwriting styles. The characters are then manually tagged on each image using the LabelImg tool or a comparable annotation program. The annotations serve as the "ground truth" and are kept in XML files. After the annotated dataset is ready, handwritten character detection is trained into the YOLOv5 model. This entails providing the model with photos and the XML files that go with them, then letting the model pick up knowledge from the annotations of the ground truth. In order for the model to identify the different characteristics of handwritten characters, its layers and parameters are changed. After the model has been sufficiently trained, it can be applied to identify handwritten pieces in fresh, unexplored photos. For every handwriting piece that is detected, the model predicts bounding boxes. The original photos' handwritten fragments are cropped using the expected bounding boxes. In order to prepare them for additional processing or analysis, this stage separates the individual characters or words in the clipped text. Moreover, photos are employed to retrain or adjust the YOLOv5 model. By giving the model more precise data on the handwriting itself—possibly eliminating background noise or unnecessary data—this iterative process of training, detection, cropping, and retraining might increase the model's accuracy.

Ultimately, the YOLOv5 model was utilized for classification and evaluation of the handwritten fragments that had been discovered and cropped. The model's accuracy and dependability can be inferred from metrics like precision, recall, and F1 score. Where the model works well and where it might need further work is determined by the performance analysis. Using SGD and the ADAMW optimizer, the YOLOv5 model is trained. Based on the gradient of the loss function, the standard algorithm Stochastic Gradient Descent (SGD) updates model

parametric quantities. It's a simple, low-cost optimizer technique with a sluggish convergence rate that can be used to set linear classifiers and regressors. The ADAMW optimizer algorithm is an extension of SGD that modifies weights by combining adaptive learning rates with momentum. It is a combination of the RMSProp and gradient descent with momentum techniques, making it a computationally efficient algorithm with little memory requirements. They are also suitable for issues with sparse or noisy gradients.

IV. RESULTS AND DISCUSSIONS

In this section, the various evaluation parameters have been represented to demonstrate the efficiency of the proposed framework. Trained the model for the recognition of Punjabi characters in offline mode and further passes to the validation process where hyper-parameters need to tuned to achieve the desired outcomes.

4.1 Training and Validation Phase

As we are working with two optimizers- SGD and ADAMW, model VOLOv5 has been trained with both optimizer in two phases. In the first phase, hyper-parameters were tuned so that segmentation of characters is carried out [38]. Secondly, the model is trained with the output of the first phase for the classification process. Thus, the common arguments with which the training of model in phase (i) and (ii) are done include batch size (32 photos from each batch are pooled and saved in RAM once they have all been trained), epochs, weight (shorten training sessions by using a pre-trained weight), learning rate (model's weights are adjusted during training), momentum and the weight decay (prevent over-fitting). ADAMW optimizer was used to train the network with 80% training and 20% validation dataset over segmented letters (region of interest). The final output of phase (i) is shown in Figure 4.

Figure 4: Output of Phase (i)

The handwritten fragments are segmented and cropped with the help of bounding box prediction and selecting the same parameters to train the YOLOv5 model as discussed in our previous work [38].

4.2 Hyper-Parameters for Classification

For the classification, two optimizers Stochastic Gradient Descent (SGD) and ADAMW with different parameters and hyper-parameters groups were used. As shown in Table 1, two optimizers SGD and ADAMW with learning rate (lr) and momentum are fixed and then the other parameters are changed. Here, the learning rate is a hyper-parameter that controls the size of the steps the optimizer takes during the training process and the momentum value of 0.937 means that the optimizer uses a running average of 93.7% of the past gradients when updating the weights.

Table 1: Training Hyper-parameters for Phase (ii)

optimizer: SGD (lr=0.01, momentum=0.937) , epochs=200			
Parameter Group	Parameter Index	Parameter Type	Weight Decay
1	57	Weight	0.0
2	64	Weight	0.0005
3	63	Bias	0.0
optimizer: ADAMW (lr=0.01, momentum=0.937) , epochs=200			
Parameter Group	Parameter Index	Parameter Type	Weight Decay

1	57	Weight	0.0
2	64	Weight	0.0005
3	63	Bias	0.0

In Table 1, Parameter Group 1({'params': 57, 'weight_decay': 0.0}) specifies that the parameters with index 57 (likely a weight parameter) have a weight decay of 0.0. Weight decay is a regularization term that penalizes large weights to prevent over fitting. Parameter Group 2 ({'params': 64, 'weight_decay': 0.0005}) specifies that the parameters with index 64 (likely another weight parameter) have a weight decay of 0.0005. Parameter Group 3 ({'params': 63, 'weight_decay': 0.0}) specifies that the parameters with index 63 (likely a bias parameter) have a weight decay of '0.0'.These parameters are configured for tuning the model during the optimization process, with specific weight decay values applied to different sets of parameters.

4.3 Performance Analysis of Punjabi Handwritten Fragments

Performance parameters, in the context of machine learning and particularly in models like YOLOv5 used for object detection, are metrics used to evaluate and quantify the effectiveness of the model. These parameters provide insights into how well the model performs in various aspects, such as accuracy, reliability, and efficiency.

Here, in this proposed work, the two cases are considered for evaluation of performance matrices. The first case (Case I) where the dataset contains Punjabi handwritten fragments with diacritics and the second case (Case II) where the dataset contains Punjabi handwritten fragments without diacritics. For each case, the efficiency of two optimization algorithms, SGD and ADAMW, has been examined. Performance indicators, including F1 Score, Precision, Mean Average Precision (mAP), and Recall, are compared using graphical analyses for both optimizers.

4.3.1 Performance Evaluation for Alphabets with Diacritics (Case I)

The Precision-Confidence **(**PC) Curve depicts the relationship between precision and confidence for a classification model. Each plot in a curve represents a class or a set of predictions at different confidence levels.

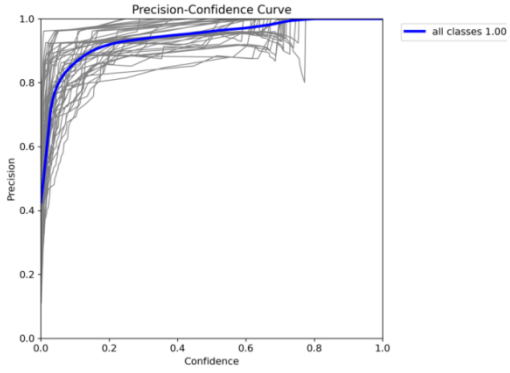


Figure 5: Precision-Confidence Curve for SGD Optimizer

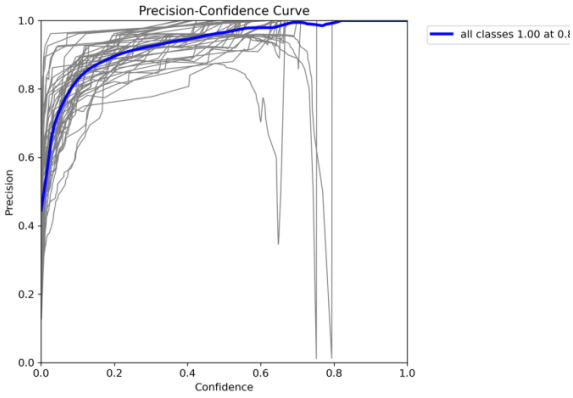


Figure 6: Precision-Confidence Curve for ADAMW Optimizer

When comparing Figure 5 and Figure 6, Figure 5 indicates a model that achieves a high precision of 1 (which means no false positives among the retrieved instances) with a lower confidence threshold of approximately 0.800. This could suggest that the model is able to be more certain about its predictions sooner. The curves are spread out across different confidence levels but tend to converge towards high precision as confidence increases. On the other hand, Figure 6, which requires a higher confidence threshold of 0.820 to achieve the same level of precision, might suggest that the model is slightly less reliable or that it needs more evidence to make a highly precise prediction. The curves in this appear to be more closely packed together until the confidence level of around 0.6, suggesting less variability in precision among classes at lower confidence levels.

The depicted Recall-Confidence Curve in Figure 7, showing identical performance for two different optimizers indicates that both are capable of training a model to achieve high recall from the onset without needing to increase the confidence threshold. A recall value equal to "1" means that the model is able to identify all relevant instances

correctly. The legend suggests that all classes achieve a recall of 0.99 at a confidence level of 0.000. This is particularly notable because it suggests that almost all positive instances are being captured by the model right from the lowest confidence threshold.

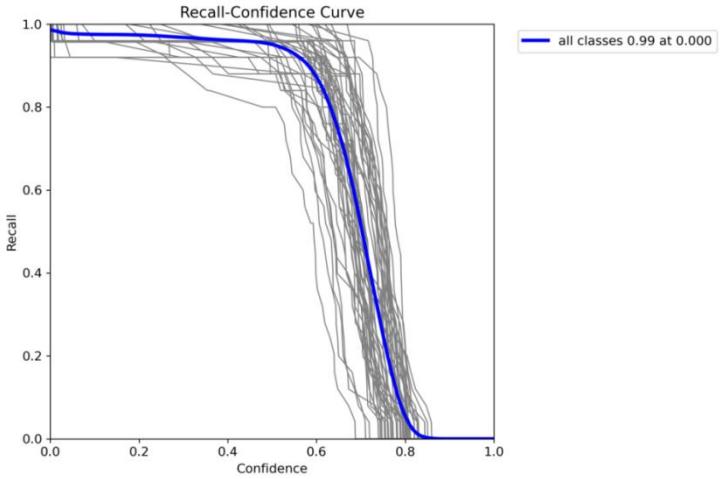


Figure 7: Recall-Confidence Curve for SGD Optimizer and ADAMW

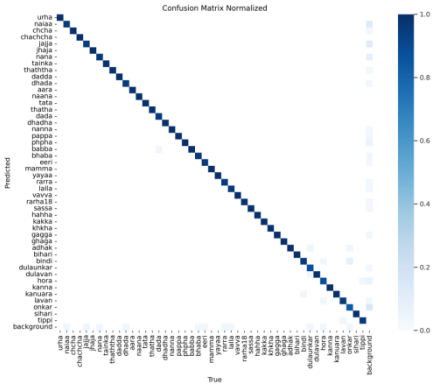


Figure 8: Confusion Matrix for SGD Optimizer (Case-I)

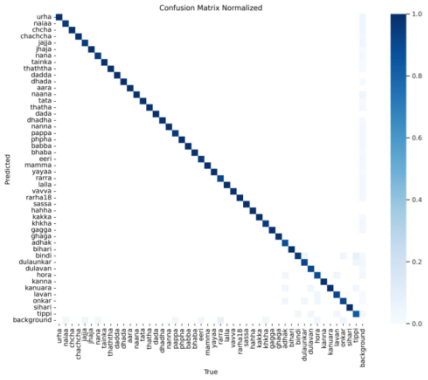


Figure 9: Confusion Matrix-ADAMW Optimizer (Case-II)

Figures 8 and 9 are normalized confusion matrix of SGD and ADAMW optimizers for Case-I. In the confusion matrix, the dark blue cells positioned along the diagonal, particularly for the 35 alphabets, signify high values, denoting correct predictions where the predicted and true classes align. Conversely, any cells deviating from this diagonal, showcased in light blue, indicate errors or misclassifications. Such 'off-diagonal' cells in both figures represent instances where diacritics are incorrectly classified. This suggests that the model tends to confuse diacritics with each other, resulting in a maximum accuracy of 97.04%

The main reason for the confusion in diacritics has been based on the inconsistency present in the dataset. From Figure 10, it is clearly evident from a few data samples of the dataset. In Figure 10 (a), the rectangle 40 and rectangle 47 contain the same diacritics i.e. “daulankar”, wherein rectangle 47 should contain “adhak” diacritic. Similarly, in Figure 10 (b), rectangle 46 and rectangle 39 should have different diacritics, whereas they are both written the same. Similarly, in Figure 10 (c), the diacritic “bihari” and “tippi” had been drawn in same manner; even an individual one can not differentiate the two diacritic by visualizing them in data sample.

So, from the samples depicted in Figure 10, it can be seen that the anomalies exist here in the dataset. Due to which, the model is getting confused about detecting the diacritics and only 97.04% or accuracy has been achieved, although the model is detecting the 35 alphabets very accurately.

So, the second case where the model is trained on only 35 alphabets is considered and performance matrices are evaluated for 35 alphabets, ignoring the diacritics which contain data inconsistency.

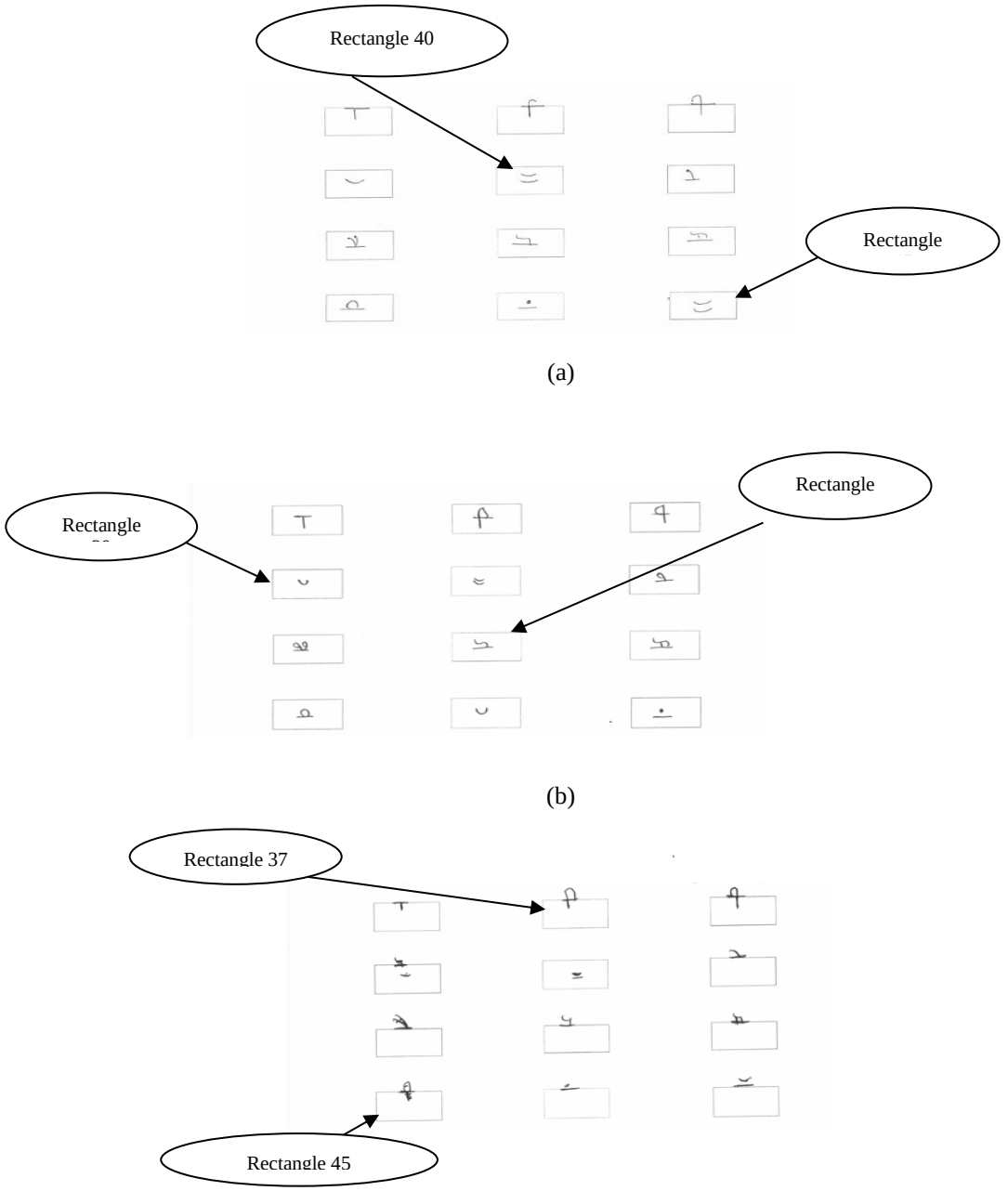


Figure 10: Inconsistencies in Dataset

4.3.2 Performance Evaluation for Alphabets without Diacritics (Case II)

The Precision-Confidence Curve is used to evaluate the performance of a classification or detection model at various confidence threshold levels for the Case-II.

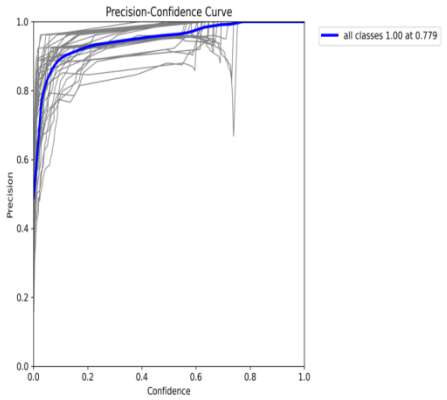


Figure 11: Precision-Confidence Curve: SGD

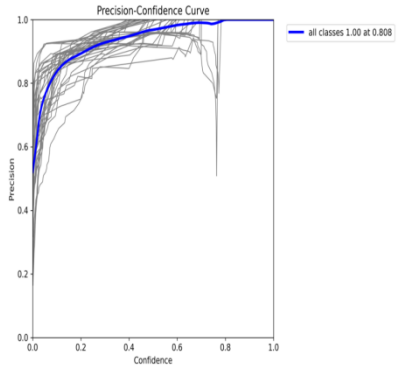
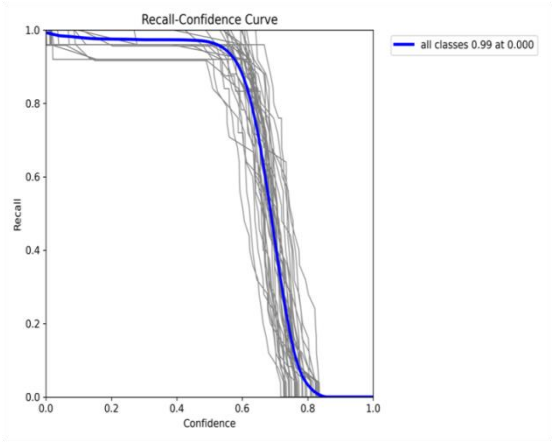
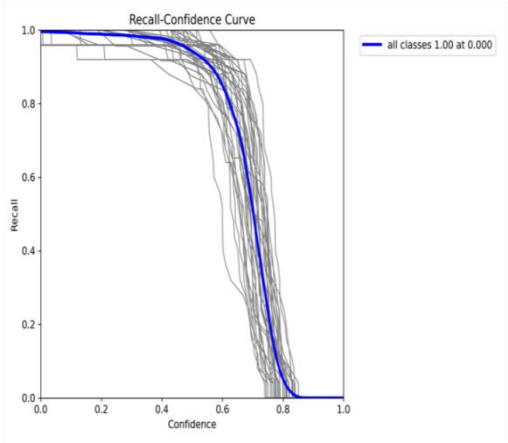


Figure 12: Precision-Confidence Curve: ADAMW Optimizer

Figure 11 and Figure 12 indicate that the model achieves a precision score of 1.00, which is perfect precision, when the confidence threshold is set to approximately 0.779 and 0.808 respectively.



(a)



(b)

Figure 13: Recall- Confidence Curve (a) SGD and (b) ADAMW Optimizer

The Recall-Confidence curve is shown in Figure 13. The recall values at different confidence thresholds, on average, achieve a recall score of 0.99 for SGD and 1.00 for ADAMW, which is extremely high as compared to SGD optimizer.

Further, in Figure 14 and Figure 15, the confusion matrices are demonstrated. Each confusion matrix consists of rows, columns and diagonal values. Here, each row of the matrix corresponds to the labels predicted by the model and each column represents the actual, true labels of the data.

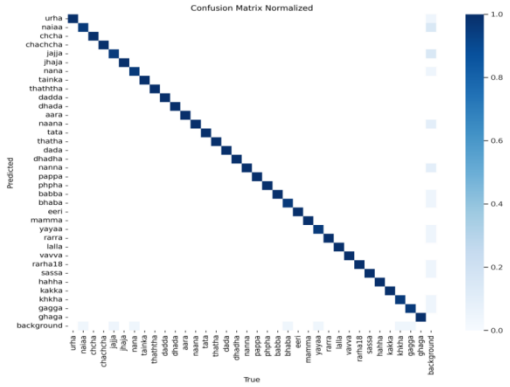


Figure 14: Confusion matrix: SGD Optimizer

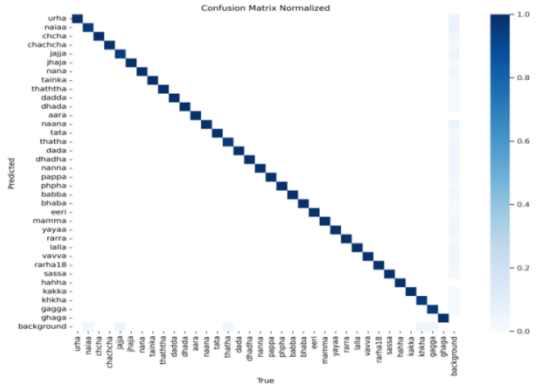


Figure 15: Confusion matrix: ADAMW Optimizer

The cells along the diagonal, from the top left to the bottom right, represent the proportion of correct predictions for each class which are in darker shades in both Figure 19 and Figure 20, indicating a high number of correct predictions.

Similarly,

Finally, Figure 16 and Figure 17 demonstrate the training and validation losses of SGD and ADAMW. In Figure 16, the first three left side boxes (i.e. train/box_loss,train/cls_loss, train/dfl_loss) it can be seen that the losses are decreasing with increasing epochs which shows good result with an accuracy of 97.08% whereas in case of val/box_loss and val/dfl_loss the losses are increasing with increasing epochs which is not a good sign.

In Figure 17, from the first three left side boxes (i.e. train/box_loss,train/cls_loss, train/dfl_loss), it can be seen that the losses are decreasing with increasing epochs also val/box_loss and val/dfl_loss are following decreasing trend and stabilizes at epochs=100 which indicates that the model's performance on the validation set is good and it is generalizing well with accuracy of 98.01%.

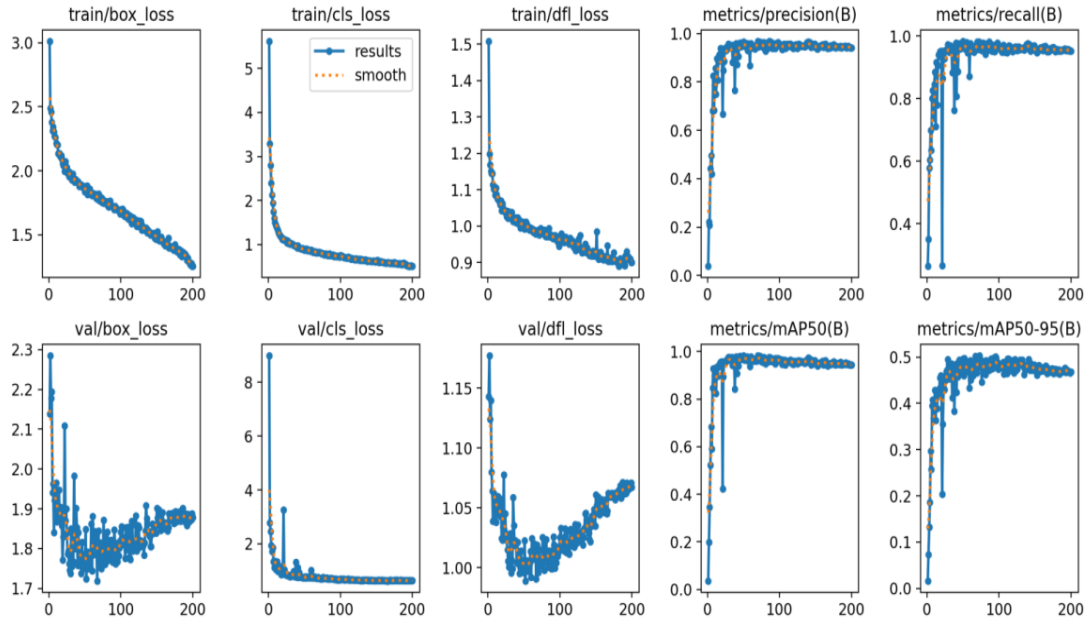


Figure 16: Overall analysis of SGD

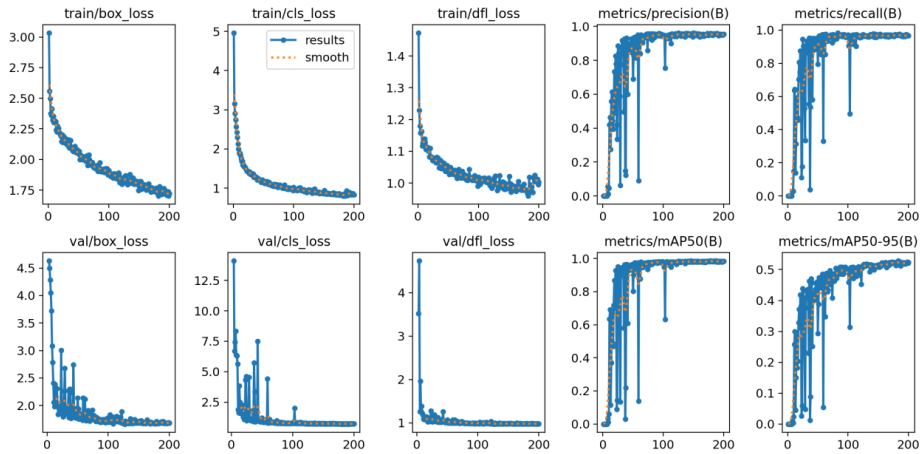


Figure 17: Overall analysis for ADAMW

4.4 Comparison Analysis of Performance Metrics

Figure 18 and Figure 19 show the overall comparison and measures performance in terms of Precision-Recall, Recall, Precision, and F1 Score. These metrics are crucial for understanding the balance between the true positives, false positives, and false negatives predicted by a classification model.

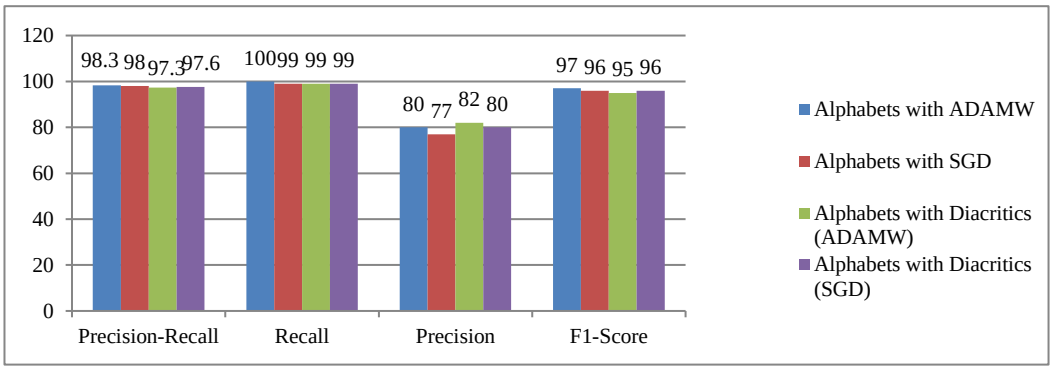


Figure 18: Comparison of Performance Analysis

Here two optimization algorithms have been compared i.e. ADAMW and SGD. Each algorithm is tested on two types of datasets, one with standard alphabets and another with alphabets that include diacritic marks.

For recall measure, both algorithms perform nearly perfectly on both datasets, with ADAMW showing a slight edge in the standard alphabet without diacritics. ADAMW consistently performs better than SGD across the standard alphabets dataset, as indicated by the higher precision-recall values. This suggests that ADAMW is better at balancing precision and recall. Overall, it can be concluded that the ADAMW performs better on standard alphabets without the diacritic marks dataset as compared to another combination of datasets and algorithms.

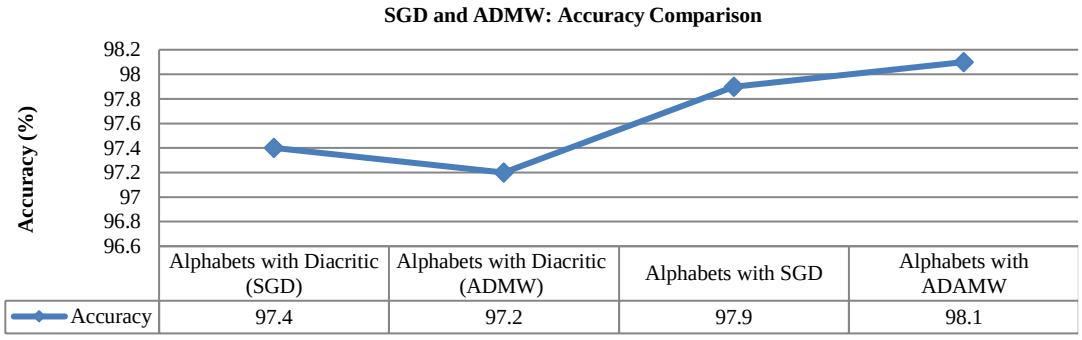


Figure 19: SGD and ADMW: Accuracy Comparison

The line chart in Figure 19 shows the accuracy of models trained with different optimization algorithms (SGD and ADAMW) on two types of datasets: one with alphabets that include diacritics (Case-I) and another with standard alphabets.For standard alphabets without diacritics (Case-II), the model using SGD performs significantly better at 97.9%, but the ADAMW optimizer achieves the highest accuracy overall at 98.1%.

Table 2: Comparison Analysis of proposed work with existing literature

References	Method	Accuracy
Gurpratap et. al.[33]	2 Stage DCNN	97.70%
Proposed work Case I	Punjabi alphabets with diacritic using automatic cropping and YOLOv5 SGD	97.40%
Proposed work Case II	Punjabi alphabets without diacritic using automatic cropping and YOLOV5 ADAMW	98.1%

The highest accuracy achieved is 98.1% by the model using the ADAMW optimizer on the standard alphabet dataset. From this graph, it can be inferred that diacritics add complexity to the task of alphabet classification due to dataset inconsistency, affecting the model's accuracy regardless of the optimizer used. The proposed study, which makes use of automatic cropping for segmentation and the YOLOv5 object detection algorithm, exhibits improved accuracy in comparison to state-of-the-art methods, as demonstrated in Table 2.

V. CONCLUSION

These methods focus on finding objects in an image and removing parts that are not needed. Once the handwritten Punjabi characters are detected, they are separated from the remainder of the picture and divided into different segments. Ultimately, a trained model is used to assign a label to each segment. First, a dataset of handwritten Punjabi characters is chosen in order to finish this process. There are 323 pictures of 35 alphabets and 12 diacritical marks in the dataset. Next, a model trained on YOLOv5 is used to automatically segment or crop these handwritten characters. The XML file of every sample image is also utilized in order to crop these characters. Each handwritten character is then recognized and categorized once more using YOLOv5.

Automatic cropping has been shown to be an effective method for reducing the amount of preprocessing needed for Punjabi handwritten fragment segmentation, mainly by using object detection techniques. The automatic cropping technique paired with the YOLOv5 model, particularly when optimized with ADAMW, showcased a slight edge in accuracy over previously art of state techniques. The F1-Confidence Curve revealed that both the SGD and ADAMW optimizers achieved high F1 scores, with ADAMW showing a slightly higher peak, indicating a refined balance between precision and recall at an optimal confidence threshold. Moreover, the analysis of the confusion matrices for both optimizers confirmed that the model is confusing diacritics marks with each other. The inconsistent nature of the dataset was the primary cause of the confusion. To mitigate this, the model was trained using only 35 alphabets, which resulted in an improved performance of higher accuracy of 98.1% whereas for identifying Punjabi characters with diacritics was still 97.04%. Further, losses plotted over epochs for both training and validation phases provided additional insights into the model's learning trajectory. From the result section, YOLOv5-ADAMW is giving better and more consistent results and qualifying the criterion in segmentation and recognition of Punjabi characters. In the future, researchers may explore the proposed framework further for the expansion of work in this field.

ACKNOWLEDGMENT

We would like to express our gratitude to Panjab University for providing the publication grant for this research work.

REFERENCES

[1] S. Jain and R. Chauhan, "Recognition of handwritten digits using DNN, CNN, and RNN," in *Communications in Computer and Information Science*, pp. 239-248, 2018.

[2] P. T. Krishnan and P. Balasubramanian, "Detection of Alphabets for Machine Translation of Sign Language Using Deep Neural Net," in *2019 International Conference on Data Science and Communication (IconDSC 2019)*, pp. 1-3, 2019.

[3] S. Samsuryadi, R. Kurniawan, and F. S. Mohamad, "Automated handwriting analysis based on Pattern Recognition: A survey," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 1, p. 196, 2021.

[4] G. Singh, S. Agrawal, and B. S. Sohi, "A robust methodology for creating large image datasets using a universal format," *Advances in Intelligent Systems and Computing*, pp. 279–288, July 2020.

[5] Y. Gurav, P. Bhagat, R. Jadhav, and S. Sinha, "Devanagari handwritten character recognition using convolutional neural networks," *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Jun. 2020. doi:10.1109/icecce49384.2020.9179193.

[6] A. G. Hochuli, L. S. Oliveira, A. S. Britto, and R. Sabourin, "Handwritten digit segmentation: Is it still necessary?," *Pattern Recognit.*, vol. 78, pp. 1-11, 2018.

[7] D. Xie, L. Zhang, and L. Bai, "Deep Learning in Visual Computing and Signal Processing," in *Appl. Comput. Intell. Soft Comput.*, vol. 2017, 2017.

[8] M. Chablani, "Yolo - you only look once, real time object detection explained," Medium, [Online]. Available: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006. Accessed Feb. 16, 2024.

[9] H. Zhao and H. Liu, "Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition," *Granular Computing*, vol. 5, no. 3, pp. 411–418, 2019.

[10] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM classifier for handwritten digit recognition," *Procedia Computer Science*, vol. 167, pp. 2554–2560, 2020.

[11] S. Ali et al., "An efficient and improved scheme for handwritten digit recognition based on convolutional neural network - discover Applied Sciences," SpringerLink, [Online]. Available: https://link.springer.com/article/10.1007/s42452-019-1161-5. Accessed Feb. 16, 2024.

[12] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K, "KNN Model-Based Approach in Classification" In: Meersman, R., Tari, Z., Schmidt, D.C. (eds) On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg.

[13] A.M. Kamoona and J.C. Patra, "A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images," *Applied Soft Computing*, 85, p. 105749, 2019.

[14] S. Albahli et al., "An improved faster-RCNN model for handwritten character recognition," *Arabian Journal for Science and Engineering*, 46(9), pp. 8509–8523, 2021.

[15] Handwritten recognition using SVM, KNN and neural network, [Online]. Available: https://www.researchgate.net/publication/313247443_Handwritten_Recognition_Using_SVM_KNN_and_Neural_Network. Accessed Feb. 16, 2024.

[16] B. R. Kavitha and C. Srimathi, "Benchmarking on offline handwritten Tamil character recognition using convolutional neural networks," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 4, pp. 1183–1190, Apr. 2022.

[17] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in Proc. IEEE Int. Conf. Comput. Vis., pp. 1635–1643, 2015.

[18] Y. Xu and J. Guo, "Low resolution handwritten digit string recognition based on Object Detection Network," in 2020 International Joint Conference on Neural Networks (IJCNN), Jul. 2020.

[19] R. Mondal, S. Malakar, E. H. Barney Smith, and R. Sarkar, "Handwritten English word recognition using a deep learning based object detection architecture," Multimedia Tools and Applications, vol. 81, no. 1, pp. 975–1000, Sep. 2021.

[20] T. Khan, "Expiry date digits recognition using deep learning," in 2019 IEEE National Aerospace and Electronics Conference (NAECON), Jul. 2019.

[21] H. Huda, M. Ariful Islam Fahad, M. Islam and A. K. Das, "Bangla Handwritten Character and Digit Recognition Using Deep Convolutional Neural Network on Augmented Dataset and Its Applications," 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, Republic of, 2022, pp. 1-7, doi: 10.1109/IMCOM53663.2022.9721634.

[22] J. Pareek, D. Singhania, R. R. Kumari, and S. Purohit, "Gujarati handwritten character recognition from text images," Procedia Computer Science, vol. 171, pp. 514–523, 2020.

[23] Rashedul Islam, Md. Rafiqul Islam, Kamrul Hasan Talukder, "An efficient ROI detection algorithm for Bangla text extraction and recognition from natural scene images", Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 8, Part B, 2022, Pages 6150-6164,34(4). DOI: 10.1016/j.jksuci.2022.02.001

[24] K. Bramara Neelima1, Dr.S. Arulselvi,"Classification of printed text and handwritten characters with Neural Networks," *Journal of critical reviews*, vol. 7, no. 02, Jan. 2020. DOI: http://dx.doi.org/10.31838/jcr.07.02.25

[25] H. M. Balaha, H. A. Ali, M. Saraya, and M. Badawy, "A new Arabic handwritten character recognition deep learning system (AHCR-DLS)," *Neural Computing and Applications*, vol. 33, no. 11, pp. 6325–6367, Oct. 2020.

[26] M. Kumar and S. R. Jindal, "A study on recognition of pre-segmented handwritten multi-lingual characters," *Archives of Computational Methods in Engineering*, vol. 27, no. 2, pp. 577–589, Mar. 2019.
[27] M. Kumar, R. K. Sharma, and M. K. Jindal, "Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition," *National Academy Science Letters*, vol. 37, no. 4, pp. 381–391, Jul. 2014.
[28] M. Kumar, M. K. Jindal, R. K. Sharma, and S. Rani Jindal, "Performance comparison of several feature selection techniques for offline handwritten character recognition," in *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, Aug. 2018.
[29] Verma, Kanupriya, et al. "Latest tools for data mining and machine learning." International Journal of Innovative Technology and Exploring Engineering 8.9 (2019): 18-23.
[30] R. Mudgil, N. Garg, P. Singh and C. Madhu, "Identification of Tomato Plant Diseases Using CNN- A Comparative Review," *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, Sonbhadra, India, 2022, pp. 174-181, doi: 10.1109/AIC55036.2022.9848931.
[31] Trivedi, Naresh K., et al. "Early detection and classification of tomato leaf disease using high-performance deep neural network." Sensors 21.23 (2021): 7987.
[32] G. Chutani, H. Bohra, D. Diwan and N. Garg, "Improved Alzheimer Detection using Image Enhancement Techniques and Transfer Learning," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-6, doi: 10.1109/INCET54531.2022.9824008.
[33] G. Singh, [Online]. Available: https://shodhganga.inflibnet.ac.in/handle/10603/482016. Accessed Feb. 16, 2021.
[34] Detection," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1706–1717, 2018.
[35] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. (2017). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE transactions on pattern analysis and machine intelligence, 39(6): 1137–49.
[36] A. G. Hochuli, L. S. Oliveira, A. S. Britto, and R. Sabourin, "Handwritten digit segmentation: Is it still necessary?," *Pattern Recognit.*, vol. 78, pp. 1–11, 2018.
[37] J.Solawetz, "What is Yolov5? A guide for beginners.," Roboflow Blog, [Online]. Available:https://blog.roboflow.com/yolov5-improvements-and-evaluation/. Accessed Feb. 16, 2024.
[38] Aanchal, Nidhi, Preeti, and Gurpratap, "Automatic cropping of handwritten scanned documents with object detection algorithm," published in *Procedia Computer Science*, vol. 218, pp. 1733–1741, 2023.