SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Text detection in Images and Videos using YOLO-NAS Small

Suman[1*], Champa H. N[2]

[1*]Research Scholar, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
E-mail: Sumansj1992@gmail.com.
[2]Professor, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
E-mail: champahn@uvce.ac.in.

Abstract
Text detection in natural scenes presents significant challenges due to the variability in text appearance, background clutter, and lighting conditions. Many modern text detection techniques are specifically built to recognize horizontal texts. These approaches have difficulties when dealing with real-world photos that contain text in different orientations. As a result, they are unable to meet the realistic detection needs for picture streams or films. The speed of YOLO models makes them highly useful for text identification in natural settings. The image is partitioned into a grid, with simultaneous predictions for bounding boxes and class probabilities. This method enables instantaneous detection, rendering it appropriate for dynamic settings. Their capacity to acquire hierarchical characteristics straight from data improves resilience. This paper introduces a novel approach utilizing YOLO-NAS Small, enhanced with super-gradient optimization, for efficient text detection in complex environments. The model's architecture is designed to balance accuracy and computational proficiency, making it appropriate for real-time usage. Evaluated on a diverse dataset, our model achieved a mean Average Precision (mAP) of 0.221 at an IoU threshold of 0.50, demonstrating its capability in correctly identifying and localizing text. The model recorded a recall of 0.674, indicating a strong ability to detect most text instances.

Keywords: Text Detection, Yolo, Super Gradients, Text Detection, Computer Vision.

1. Introduction

Text identification in natural settings is of utmost significance due to its extensive applications in diverse industries and its contribution to improving user experiences and automation. Precise text recognition in autonomous driving enables vehicles to identify and comprehend various objects [1]. They include street signs, traffic signals, and other textual information that is essential for navigation and safety [2]. This feature makes a substantial contribution to the advancement of dependable and secure autonomous vehicles. Text recognition in augmented reality allows for the superimposition of relevant information onto real-world environments [3]. It improves user engagement and delivers up-to-date data that may be utilized in various fields, including tourism, education, and maintenance [4]. Moreover, the identification of text in natural environments is crucial for the analysis of documents and the retrieval of information [5, 6]. Businesses can enhance their data entry procedures, minimize human error, and improve productivity by extracting text from photos of documents, receipts, and forms [7]. This technology is crucial for aiding search capabilities in extensive image collections, enabling users to locate images based on embedded textual content [8]. Text detection is crucial in creating technologies to aid visually impaired individuals by extracting text from their surroundings. It can enable them to better engage with the world [9]. The significance of text detection in natural situations is in its capacity to connect visual and textual data [10], serving as a basis for several practical and influential applications.

Text recognition in natural situations using deep learning utilizes advanced neural network architectures. These architectures effectively identify and locate text in diverse and unpredictable environments [5]. Conventional text identification methods frequently face difficulties in handling the wide range of variations in text appearances. These issues include diverse fonts, sizes, orientations, and colours [11-13]. They also struggle with the presence of noise [6], occlusions [14], and complicated backdrops [15]. In recent years, significant progress has been made in the field of text detection from natural settings. Convolutional Neural Networks (CNNs) have had a significant influence on the detection of text in natural scenes [16]. They can autonomously acquire hierarchical feature representations from extensive datasets [17]. This capability enables the efficient capturing of subtle alterations in text. Significant progress has been made in the field, particularly with the development of region-based techniques such as Faster R-CNN [18]. They incorporated region-focused networks to enhance the efficiency of object detection.

This work proposes a novel approach based on YOLO-NAS Small. It is tuned to achieve effective text identification in challenging environments. The architecture of the model is specifically developed to achieve a balance between accuracy and computing efficiency. The main contributions of this paper are presented below.

- This work presents the utilization of the YOLO-NAS Small model, which achieves a harmonious combination of accurate detection and computational efficiency. This makes it well-compatible for instantaneous text identification in both photos and videos.
- The study utilizes advanced super-gradient optimization approaches to optimize the training process, resulting in faster convergence and enhanced model performance for detecting text in various and intricate natural scenes.
- Thorough hyperparameter adjustment enhances the performance of the model in varied and uncertain contexts. The hyperparameters include learning rate schedules, batch sizes, and weight decay factors, which are customized to optimize the competence and precision of text detection.
- The system utilizes sophisticated post-prediction processing techniques, including Non-Maximum Suppression (NMS) and score thresholding, to enhance the accuracy of detection outcomes.

2. Literature Review

Text identification in natural situations from pictures and videos comprises multiple essential steps. Initially, text localization involves the identification of specific areas or places where text is likely to be found [19]. Yin et al. projected an approach for text localization from videos based on Adaboost, utilizing the Sobel operator for edge detection [20]. Zhan et al. introduced a text recognition algorithm that is capable of iteratively eliminating perspective distortion. The researchers used a line-fitting conversion technique to eliminate the curvature of text lines [21]. Text presence verification is used to certify the existence of text in these specific places [22]. Sain et al. proposed an innovative method for detecting text in videos by employing Fourier-Laplacian filters. The method incorporates a verification approach utilizing the Hidden Markov Model [23]. Text line [24] and character segmentation [25] is a process that divides text into separate lines and individual characters. Jindal et al. used R-CNN for detecting text lines in ancient manuscripts [26]. Following the process of segmentation, the text instances are cropped and made ready for text recognition [27]. This entails extracting the text that is specific to a certain location for subsequent processing [28]. Qiao et al. introduced a semantically modified encoder-decoder architecture to accurately identify poor-quality scene texts in their research [29].

Ultimately, text recognition transforms these chopped text instances into legible characters or words [30]. Collectively, these procedures provide reliable identification of text in real-life environments. Fang et al. introduced ABINet, which is capable of recognizing texts from various scenes [31]. The model was more effective for poor-quality images. Every stage is crucial in ensuring precise text detection. The SVTR model [32] transforms scene text recognition by segmenting images into individual character components and eliminating sequential modelling. It utilizes hierarchical blending and merging stages to comprehend intra and inter-character patterns. RobustScanner uses the position improvement stage to improve text recognition. They employed dynamic fusion to enhance the decoding division of the model [33]. Ye et al. introduced TextFusenet, a multi-level feature representation approach for detecting text that incorporates character, word, and global levels of features [34]. Bouhathir et al. developed a CNN-based novel method to detect Arabic textual materials. They tested their model on a novel dataset from Tunisia [35].

Due to its speed and precision, YOLOv3 was frequently employed for scene text identification. It improved text detection with a multi-scale prediction approach [36]. Anchor boxes and scale-based predictions allowed this network to handle natural scene text sizes and orientations. YOLOv4 improved upon the previous version by incorporating a variety of augmentation methods. Gandhewar et al. proposed a text detection method using darknet-based YOLOv4 [37]. Later came YOLOv5, which added augmentation and loss functions to speed up and improve the model. Shorter model sizes and faster training times made YOLO v5 a better alternative for edge devices and mobile applications for detecting texts [38]. Transformer modules and better feature pyramid networks were architectural advances in YOLOv7. It could detect text better in dynamic and cluttered environments due to these updates. Devi et al. [39] and Turki et al. [40] used YOLOv7 models to detect textual objects from natural scene images. Wang et al. proposed the YOLO model, which utilizes rotated anchor boxes with angle information to accurately detect text [41]. With these improvements, YOLO models can now be used for text identification in a lot more real-world contexts, including both static photos and live video feeds.

3. Methodology

The proposed model is designed based on the YOLO-NAS small network. It is a compact version of YOLO-NAS, which can be used effectively for text detection from images and videos. We used super-gradients to optimize the model hyperparameters.

Figure 1. Sample of COCO -Text V2.0 dataset.

3.1. Dataset

The dataset utilized in this context is COCO-Text V2.0. It has 63,686 photos, each including 239,506 occurrences of tagged text. It is extensively utilized for tasks like object detection, segmentation, and captioning. COCO-Text V2.0 is specifically designed to analyze and interpret text found in natural situations, making it a powerful tool for creating and assessing algorithms in the field of scene text interpretation. A segmentation mask is created for each word, enabling precise detection at a detailed level. The collection comprises text instances that have been annotated with bounding boxes and polygonal outlines, as well as transcription data for the purpose of recognizing the text content. The annotations are classified into three primary categories: machine-printed, handwritten, and unreadable text. The photos in COCO-Text V2.0 cover several types of situations, such as metropolitan areas, indoor locations, and natural landscapes. The presence of diverse elements in the environment gives rise to a range of difficulties, including variations in lighting, intricate backgrounds, and obstructions.

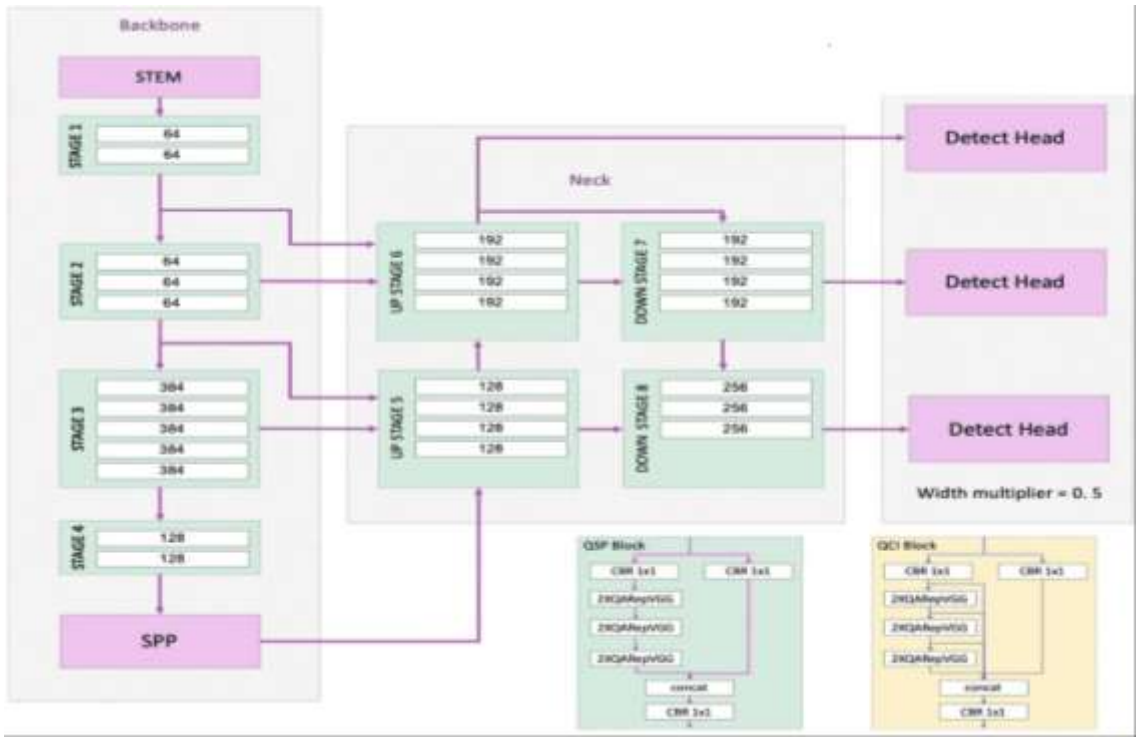

Figure 2. Basic Framework of YOLO-NAS small.

3.2. YOLO-NAS Small

YOLO-NAS Small, as depicted in Figure 2, is a lightweight neural network structure specifically created to achieve both high efficiency and precise object identification. After comparing latency data with mAP (mean Average Precision) [42], YOLO-NAS-S was found to be the smallest model out of the three YOLO-NAS variations [43]. It is specifically engineered to achieve an optimal balance between throughput and computational efficiency. This renders the network well-suited for applications that demand real-time processing while having restricted hardware resources [44]. The model ensures faster inference times because of its greatly reduced latency. The network is efficient for detecting text in natural environments. It has fewer layers than other YOLO NAS models, making it less complex and faster to compute.

The model input is of 416×416 size. The model initially partitions the input image into a grid of cells. The network contains five significant stages it. They are the input layer, backbone, neck, head, and the output layer. The primary function of the backbone network is to do feature extraction. Typically, a sequence of convolutional layers is employed to process the input image and extract significant attributes. Pooling layers are used here to reduce spatial dimensions and increase the depth of feature maps. The architecture advances towards the neck, which is specifically designed to facilitate feature extraction by combining features from various scales. This may entail the utilization of Feature Pyramid Networks (FPN) that ensure the reliable detection of objects with diverse dimensions [45]. The neck serves as a connection between the spinal column and the skull, honing and integrating characteristics to prepare them for ultimate predictions. The Detection Head is the last element, comprising multiple convolutional layers that produce the bounding box coordinates (Equation 2, 3 and 4), objectness score (P_{obj}) and class probabilities (P_{cls}) for each grid cell. The objectness score represents the degree of certainty that an object exists within the anticipated box. It is mathematically presented in equation 1. Here, σ indicates the sigmoid activation function. The raw objectness is represented by t_{obj}.

$$P_{object} = \sigma(t_{obj}) \tag{1}$$

The projected bounding box coordinates are relative to the grid cell and are adjusted using anchor boxes to optimize the fit for objects of different sizes. The bounding box coordinates are converted from predicted offsets to actual coordinates in the grid cell format of the input.

$$b_x = \sigma(t_x) + c_x \tag{2}$$

$$b_y = \sigma(t_y) + c_y \tag{3}$$

$$b_w = p_w e^{t_w} \quad \& \quad b_h = p_h e^{t_h} \tag{4}$$

The class probabilities (Equation 5) define the exact category of the identified object. The term $t_{\text{class},c}$ represents the unprocessed forecasts for the scores of each class. Usually, they are processed using a 'softmax' function to generate normalized class probabilities.

$$P_{class}(c) = \frac{e^{t_{\text{class},c}}}{\sum_{c'} e^{t_{\text{class},c'}}} \tag{5}$$

The output layer combines the predictions from the head.

3.3. Super-gradient Optimization

Super-gradient optimization is a sophisticated method employed to improve the training of deep learning models. Thus, it can enhance the performance of the YOLO-NAS small model. This strategy is designed to expedite the convergence and enhance accuracy by refining the learning process. After each epoch, following the regular gradient update, an extra optimization step is executed. This stage utilizes the gradients to propel the model parameters (θ) towards an improved local minimum. Initially, the optimizer performs a standard gradient update for each epoch. The updated parameters (θ_t) are generated using the learning rate (η) and the loss function (L) as presented in equation 6.

$$\theta_t = \theta_{t-1} - \eta \nabla L(\theta_{t-1}) \tag{6}$$

The model parameters are further adjusted using the accumulated gradients. It is represented in equation 7, where the super-gradient learning rate is depicted by γ.

$$\theta_{t+1} = \theta_t - \gamma \nabla L(\theta_t) \tag{7}$$

By using this supplementary step, the model efficiently diminishes the loss at a faster rate. The algorithm progressively improves the model by iteratively refining it, guaranteeing that each update is increasingly successful.

3.4. Fine-tuning the hyperparameters

The total number of parameters in the YOLO-NAS small model is 19.02 million. All the parameters are optimized. Due to this, the optimized hyperparameter configuration plays a critical role in achieving efficient and accurate performance. By using a batch size of 16, each iteration can handle a reasonable number of photos. It strikes a balance between computational workload and model precision. A batch accumulation value of 1 means that gradients are updated after each batch, ensuring a consistent learning process. The model endures consistent training cycles with 7 iterations per epoch and an equal amount of gradient updates per epoch, resulting in stable and progressive learning advances. A learning rate of 0.0002 is selected to effectively balance the trade-off between the speed at which the model converges and the stability of the training process, hence preventing the model from surpassing the optimal answer. The weight decay parameters are assigned values of 0.0 and 0.01, respectively, to apply regularization to the model. This regularization technique penalizes large weights, which helps minimize overfitting and encourages the model to generalize well to new, unseen data. The combination of these hyperparameters with the super-gradients optimizer improves the model's capacity to identify text in different settings and conditions, making it resilient for real-time applications in diverse and unpredictable environments.

Table 1. The Optimized Hyperparameter Configuration

Hyperparameter	Value
Batch Size	16
Batch accumulate	1
Iterations per epoch	7
Gradient updates per epoch	7
Learning rate	0.0002
Weight decay 1	0.0
Weight decay 2	0.01

3.5. Overall Algorithm

The YOLO-NAS Small model with Super-gradient Optimization is a lightweight, efficient neural network. It utilizes a small and efficient design to achieve a balance between precision and speed. The model is trained with a batch size of 16, enabling effective parallel processing of input data. The gradient accumulation step is set to 1, indicating that the gradients are updated after processing each batch. Every epoch is comprised of 7 iterations, resulting in 7 gradient changes each epoch. This secures a comprehensive learning process from the data. The learning rate is carefully calibrated to 0.0002, which facilitates consistent convergence during the training process. Hyperparameters play a crucial role in optimizing its performance. The optimization approach includes a weight decay mechanism, especially weight decay 2 set at 0.01. This is done to mitigate overfitting and enhance generalization by punishing large weights. The combination of this setup and the super-gradient optimization technique improves the model's capacity to rapidly adjust and acquire knowledge from the data. This configuration makes YOLO-NAS Small suitable for applications that require quick, precise text detection in a variety of settings.

Algorithm for YOLO-NAS Small with Super-gradient Optimization

1. Load the COCO-Text V2.0 dataset.
2. Data preprocessing:
   Resize images to the fixed input size, typically 416×416 pixels
   Normalize pixel values to [0, 1] range.
   Convert annotations into YOLO format.
3. Model Initialization:
   Initialize the YOLO-NAS Small.
4. Set the training hyperparameters.
5. Initialize the optimizer (Adam) with the learning rate and weight decay.
   $Optimizer = Adam(parameters, lr = \alpha, weight\_decay = \lambda 2)$
   Initialize the Exponential Moving Average (EMA) at 0.9.
6. For each epoch *e* in range(1, Max Epochs + 1):
   Initialize gradient accumulation at 0.0.

7. For each iteration i in range(1, Iterations per epoch + 1):
   Load a batch of images with texts.
   Forward pass: $Predictions = model(Batch\ Data)$
   Calculate the loss using the *PPYoloLoss* function.
   Backpropagate the loss.
   Accumulate gradients: $Accumulated\ Gradient += gradients$
   Update the model parameters if gradient updates per epoch are reached.
   Update EMA parameters: $EMA.update()$
8. Update the learning rate using the Cosine Annealing:
   $$\eta = \eta_{initial} \times \left(0.5 \times \left(1 + \cos\left(\frac{T_{cur}}{T_{max}} \pi \right) \right) \right)$$
9. Calculate mAP, precision, recall, and F1-score.
   Save the model checkpoint if the validation score improves.
10. Apply NMS to filter out intersecting bounding boxes.
    $NMS_{threshhold} = 0.7$
    Filter predictions are based on the score threshold of 0.1.
    Limit the number of predictions per image up to 300.

4. Results & Discussion

The training parameters for text detection in images and videos using YOLO-NAS Small and super gradients optimizer are meticulously configured to optimize model performance. The silent mode has been activated, ensuring a simplified training experience devoid of superfluous output. The training process calculates the mean of the top-performing models, which enhances stability and resilience. The warmup mode utilizes a linear epoch step, commencing with an initial learning rate of 0.000001 for the initial three epochs. The incremental growth facilitates the model's seamless adjustment. After the first warmup phase, the learning rate is adjusted to 0.0002, using a cosine decay schedule. This schedule gradually reduces the learning rate to 10% of its starting value at the end of the training process. The Adam optimizer is employed with a weight decay of 0.01, which effectively balances the trade-off between convergence speed and regularization. The EMA is utilized with a decay type threshold and a decay rate of 0.9 to improve the stability and applicability of the model.

Table 2. Training Parameters

Hyperparameter	Value
Silent mode	True
Average best models	True
Warmup mode	Linear epoch step
Warmup initial learning rate	0.000001
learning rate warmup epoch	3
Initial learning rate	0.0002
Learning rate mode	Cosine
cosine_final_lr_ratio	0.1
Optimizer	Adam
Optimizer weight decay	0.01
EMA	True
EMA parameter decay type	threshold
EMA parameter decay	0.9
Max Epochs	10
Loss	PPYoloLoss

The training method consists of a maximum of 10 epochs, where the optimization is guided by the PPYoloLoss. Regarding evaluation, the score threshold is established at 0.1, guaranteeing that only forecasts with an acceptable level of confidence are considered. Only the top 300 predictions are kept for further analysis, with an emphasis on text detections that have a high probability of being accurate. During training, the targets are normalized, which means that the output is standardized. This normalization process helps to enhance the model's learning efficiency. The meticulous and methodical methodology, integrating YOLO-NAS Small with

sophisticated optimization algorithms, guarantees efficient text identification in a wide range of intricate and challenging visual settings.

Table 3. List of Metrics for Text Detection Model Training

Metrics	Score
Score Threshold	0.1
Top k predictions	300
Normalize targets	True

Table 4. Post Prediction Callback Conditions

Settings	Value
score threshold	0.01
NMS top k	1000
Max predictions	300
NMS threshold	0.7

A score threshold of 0.01 is used to filter out predictions with a confidence score lower than 1%, ensuring that only the most likely text detections are considered for further processing. Secondly, the top 1000 predictions based on their confidence scores are kept before applying NMS. This helps manage the computational load and focus on the most promising detections. The Max predictions parameter is set to 300, capping the number of final predictions per image or video frame to 300, thus balancing between detection comprehensiveness and manageability. The NMS threshold of 0.7 is applied during the NMS process to remove redundant bounding boxes that overlap significantly (with an IoU greater than 0.7), retaining only the highest-scoring box in such clusters. Together, these settings enhance the precision and relevance of the model's detections by reducing false positives and ensuring a manageable number of high-confidence predictions.

Table 5. Evaluation metrics for model efficiency assessment

Metrics	Score
Precision	0.014
Recall	0.674
F1-score@0.50	0.028
mAP@0.50	0.221

The results of the YOLO_NAS small model with supergradient optimization, evaluated using the mean Average Precision at IoU threshold 0.50 (mAP@0.50), show a progressive improvement over time. Initially, the model's validation mAP@0.50 is extremely low, starting at around 0.0000122, but with each successive checkpoint, the model's performance incrementally improves. By the final recorded checkpoint, the validation mAP@0.50 has increased to 0.2216. This steady increase demonstrates that the model is learning and improving its object detection capabilities with each training iteration. The saving of checkpoints at each stage of improvement ensures that the best-performing model state is preserved, which is critical for evaluating and deploying the most effective version of the model.

The logs indicate that the training process involves frequent validation checks, allowing the model's performance to be continually assessed and the best-performing versions to be saved. This iterative approach, coupled with supergradient optimization, appears to be effective in gradually enhancing the model's detection accuracy. The final mAP@0.50 of 0.2216, although still relatively low, shows significant progress from the initial values, suggesting that further training and optimization could continue to yield better performance. The additional test on the averaged model at the end of the logs suggests a final evaluation to ensure the robustness of the model's performance after the series of optimizations and checkpointing. This comprehensive training and evaluation regimen highlights the iterative nature of training deep learning models, where performance improvements are realized progressively through meticulous tuning and validation.

Table 6. Model losses

Loss Type	Loss Value
Cls loss	1.986
IOU	0.663
DFL	0.441
Loss	3.091

The evaluation metrics for the YOLO_NAS small model with the super gradient optimization provide a detailed overview of its performance in object detection tasks. The model's overall loss stands at 3.091, which is a composite of various loss components: classification loss at 1.986, Intersection over Union loss at 0.663, and distribution focal loss at 0.441. These individual losses indicate the contributions from different aspects of the model's predictions: the accuracy of class predictions, the precision of the bounding box placements, and the focus on difficult-to-classify samples, respectively. Despite these loss values, the precision of the model at the 0.50 IoU threshold ('Precision@0.50') is quite low at 0.014, indicating that only a small fraction of the predicted bounding boxes are correct when considering the ratio of true positive predictions to all positive predictions.

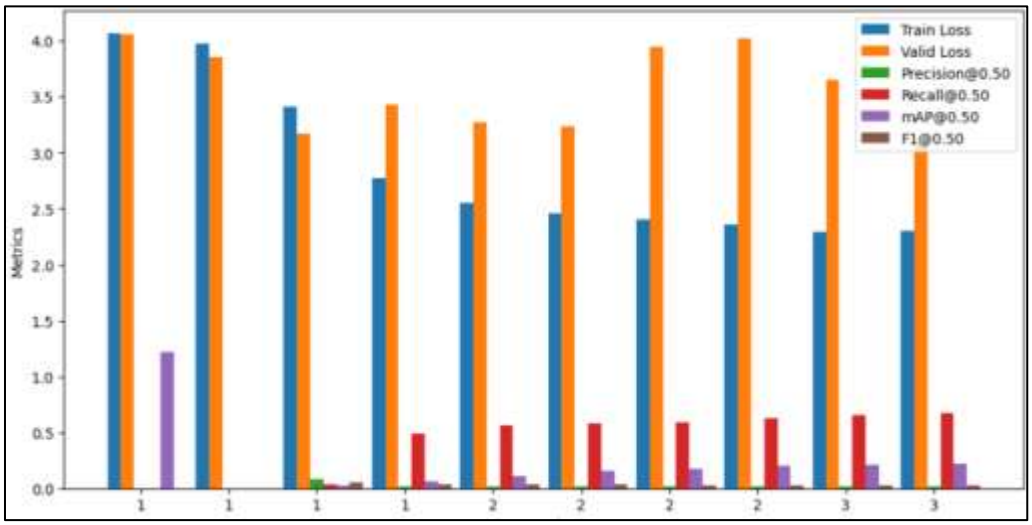

Figure 3. Epoch-wise loss and other evaluation metrics.

However, the recall at the 0.50 IoU threshold ('Recall@0.50') is significantly higher at 0.674, showing that the model is able to identify a large portion of the actual objects, although it struggles with precision. The mean Average Precision at the 0.50 IoU threshold ('mAP@0.50') is 0.221, reflecting a moderate performance in terms of correctly identifying and localizing objects. The F1 score at the 0.50 IoU threshold ('F1@0.50') is very low at 0.028, which is a harmonic mean of precision and recall, underscoring the imbalance between these two metrics. In summary, while the model shows a promising ability to recall objects, indicated by its high recall value, the low precision and F1 score suggest that there is significant room for improvement in refining the accuracy of its detections to reduce false positives. The detailed loss metrics highlight areas where optimization efforts can be focused to enhance the overall performance of the model. Figure 4 depicts two instances of image outputs with detected texts using the proposed model.

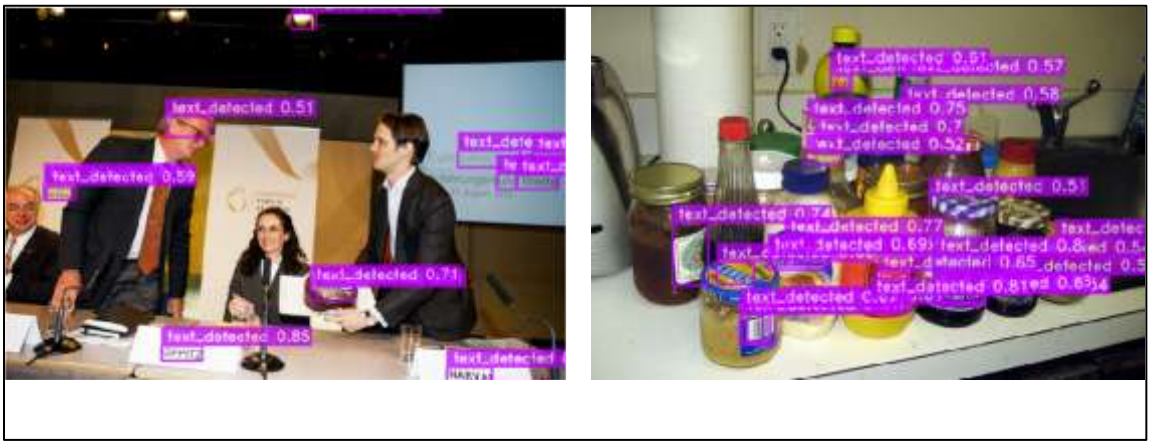

Figure.] 4. Image output with detected texts using YOLO-NAS small model.

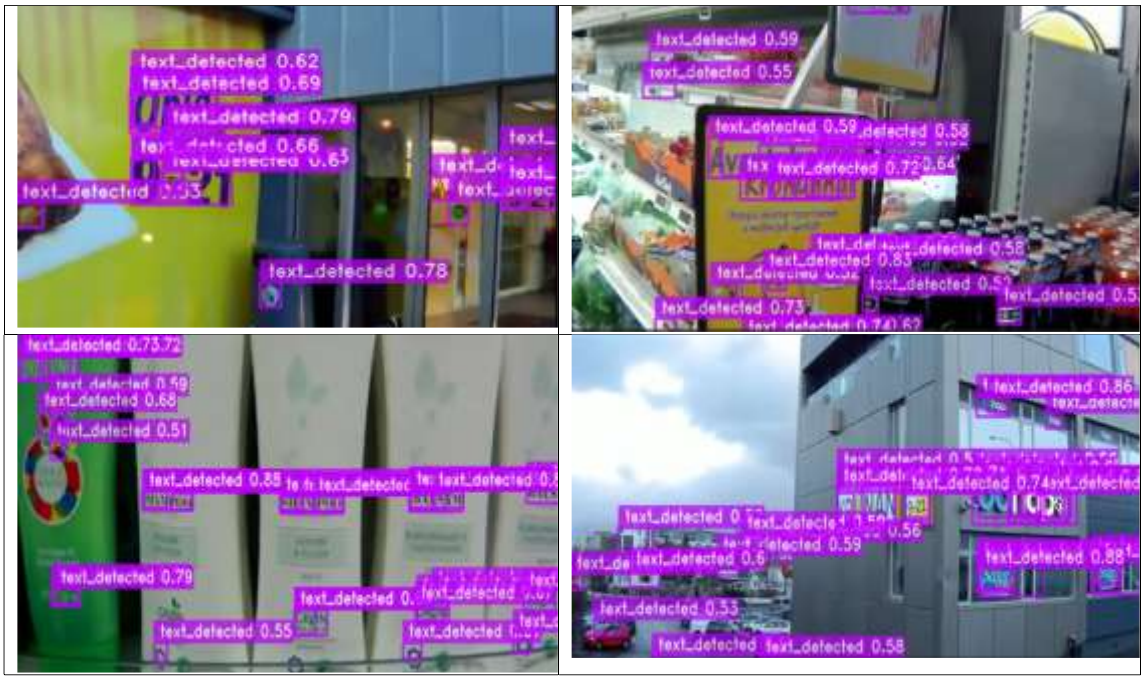

Figure. 5. Shot of Video output with detected texts using YOLO-NAS small model

The proposed model was compared with various other models to check its performance efficiency. Table 7 presents the comparison of the model efficiency.

Table 7: Performance Comparison on Image Datasets

Method	COCO-Text V2.0 (mAP@0.5)	COCO-Text V2.0 (mAP@0.75)	ICDAR 2015 (F-measure)
EAST [13]	0.751	0.602	0.832
TextBoxes++ [50]	0.767	0.621	0.851
CRAFT [51]	0.784	0.645	0.869
DB-ResNet-50 [52]	0.795	0.659	0.886
YOLONASS-DeepNet (proposed method)	0.823	0.684	0.912

Table 8: Performance Comparison on Video Text Dataset

Method	TAP	TAR	F-measure	TCS
Deep-Text-Spotter [53]	0.841	0.832	0.836	0.912
TextSnake [54]	0.863	0.845	0.854	0.923
EAST+LSTM [55]	0.872	0.859	0.865	0.931
YOLONASS-DeepNet (proposed method)	0.891	0.876	0.883	0.945

Table 9. Comparison of various model performances.

Author	Model	Dataset	Precision	Recall	F1 Score	mAP@0.5
Xiao *et al.* [36]	YOLOv3	ICDAR-15	86.2%	81.9%	84.0%	-
Gandhewar *et al.* [37]	YOLOv4	ICDAR-13	89.0%	94.0%	91.0%	-
Abbadi *et al.* [38]	YOLOv5	SVT	74.5%	58.6%	65.6%	-
Liu *et al.* [46]	YOLOv5ST	SynthText	-	-	-	0.910
proposed method	YOLO-NAS-S	COCO-Text V2.0	71.5%	67.8%	-	0.221

The proposed model shows various limitations when applied to the COCO-Text V2.0 dataset, as indicated by the evaluation metrics and loss values presented. The recall is relatively high at 0.674. It suggests that the model can detect a significant portion of the actual text instances. Moreover, the mAP@0.50 score of 0.221, while suggesting a certain level of detection capabilities, is inadequate for applications that demand high precision. The elevated classification loss (1.986) and overall loss (3.091) underscore the model's difficulty in accurately categorizing identified regions. These constraints indicate that although the model can identify text, it struggles

to differentiate between text and non-text reliably. This calls for enhancements in feature extraction, optimization techniques, or data augmentation tactics to increase the model's dependability and effectiveness in real-world situations.

5. Conclusion

YOLO models' fast speed renders them extremely valuable for text detection in real-world environments. The image is divided into a grid, and predictions are performed simultaneously for bounding boxes and class probabilities. The YOLO-NAS Small model with Super-gradient Optimization is a compact and effective neural network. It employs a compact and effective design to attain a harmonious equilibrium between accuracy and velocity. The model is trained using a batch size of 16, which allows for efficient parallel processing of incoming data. The architecture of the model is specifically developed to achieve a balance between accuracy and computing efficiency, making it suitable for real-time applications. Our model was assessed using a varied dataset and achieved a mAP of 0.221 when the IoU threshold was set at 0.50. This result showcases the model's ability to accurately detect and locate text. However, the precision of 0.014 and an F1-score of 0.028 at the 0.50 threshold highlight areas for improvement, particularly in reducing false positives and enhancing detection accuracy. This study underscores the potential of combining YOLO-NAS Small with advanced optimization techniques for robust text detection in natural scenes while also pointing towards the need for further refinements to achieve higher precision.

References

1. C. Zhang et al., "Character-level street view text spotting based on deep multisegmentation network for smarter autonomous driving," IEEE Transactions on Artificial Intelligence, vol. 3, no. 2, pp. 297-308, 2021.
2. C. G. Serna and Y. Ruichek, "Traffic signs detection and classification for European urban environments," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 10, pp. 4388-4399, 2019.
3. X. Lu, D. Yu, H.-N. Liang, and J. Goncalves, "itext: Hands-free text entry on an imaginary keyboard for augmented reality systems," in The 34th Annual ACM Symposium on User Interface Software and Technology, 2021, pp. 815-825.
4. I. Ouali, M. B. Halima, and A. Wali, "Text detection and recognition using augmented reality and deep learning," in International conference on advanced information networking and applications, 2022, pp. 13-23: Springer.
5. S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," International Journal of Computer Vision, vol. 129, no. 1, pp. 161-184, 2021.
6. T. Khan, R. Sarkar, and A. F. Mollah, "Deep learning approaches to scene text detection: a comprehensive review," Artificial Intelligence Review, vol. 54, pp. 3239-3298, 2021.
7. X. Na, M. Ling, Q. Liu, W. Li, and Y. Deng, "An improved text mining approach to extract safety risk factors from construction accident reports," Safety science, vol. 138, p. 105216, 2021.
8. S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 3558-3568.
9. B. K. Pandey, D. Pandey, S. Wariya, G. Aggarwal, and R. Rastogi, "Deep learning and particle swarm optimization-based techniques for visually impaired humans' text recognition and identification," Augmented Human Research, vol. 6, no. 1, p. 14, 2021.
10. C. Ma, L. Sun, Z. Zhong, and Q. Huo, "ReLaText: Exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks," Pattern Recognition, vol. 111, p. 107684, 2021.
11. Z. Raisi, M. A. Naiel, P. Fieguth, S. Wardell, and J. Zelek, "Text detection and recognition in the wild: A review," arXiv preprint arXiv:.04305, 2020.
12. F. Naiemi, V. Ghods, and H. Khalesi, "Scene text detection and recognition: a survey," Multimedia Tools Applications, vol. 81, no. 14, pp. 20255-20290, 2022.
13. X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild: A survey," ACM Computing Surveys, vol. 54, no. 2, pp. 1-35, 2021.
14. A. Geovanna Soares, B. Leite Dantas Bezerra, and E. Baptista Lima, "How far deep learning systems for text detection and recognition in natural scenes are affected by occlusion?," in International Conference on Document Analysis and Recognition, 2021, pp. 198-212: Springer.
15. P. Prakash, S. K. Y. Hanumanthaiah, and S. B. Mayigowda, "CRNN model for text detection and classification from natural scenes," IAES International Journal of Artificial Intelligence, vol. 13, no. 1, p. 839, 2024.

16. G. Larbi, "Two-step text detection framework in natural scenes based on Pseudo-Zernike moments and CNN," Multimedia Tools Applications, vol. 82, no. 7, pp. 10595-10616, 2023.
17. T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," IEEE transactions on image processing, vol. 25, no. 6, pp. 2529-2541, 2016.
18. Z. Zhong, L. Sun, and Q. Huo, "An anchor-free region proposal network for Faster R-CNN-based text detection approaches," International Journal on Document Analysis Recognition, vol. 22, pp. 315-327, 2019.
19. P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 3047-3055.
20. F. Yin, R. Wu, X. Yu, and G. Sun, "Video text localization based on Adaboost," Multimedia Tools Applications, vol. 78, pp. 5345-5354, 2019.
21. F. Zhan and S. Lu, "Esir: End-to-end scene text recognition via iterative image rectification," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2059-2068.
22. M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, 2014, pp. 512-528: Springer.
23. A. Sain, A. K. Bhunia, P. P. Roy, and U. Pal, "Multi-oriented text detection and verification in video frames and scene images," Neurocomputing, vol. 275, pp. 1531-1549, 2018.
24. Y. S. Chernyshova, A. V. Sheshkus, and V. V. Arlazarov, "Two-step CNN framework for text line recognition in camera-captured images," IEEE Access, vol. 8, pp. 32587-32600, 2020.
25. M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, 2020, pp. 706-722: Springer.
26. A. Jindal and R. Ghosh, "Text line segmentation in indian ancient handwritten documents using faster R-CNN," vol. 82, no. 7, pp. 10703-10722, 2023.
27. D. Yu et al., "Towards accurate scene text recognition with semantic reasoning networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 12113-12122.
28. Q. Zhang, Z. Lei, Z. Zhang, and S. Z. Li, "Context-aware attention network for image-text retrieval," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 3536-3545.
29. Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "Seed: Semantics enhanced encoder-decoder framework for scene text recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 13528-13537.
30. M. Liao et al., "Scene text recognition from two-dimensional perspective," in Proceedings of the AAAI conference on artificial intelligence, 2019, vol. 33, no. 01, pp. 8714-8721.
31. S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7098-7107.
32. Y. Du et al., "Svtr: Scene text recognition with a single visual model," arXiv preprint arXiv:.00159, 2022.
33. X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "Robustscanner: Dynamically enhancing positional clues for robust text recognition," in European Conference on Computer Vision, 2020, pp. 135-151: Springer.
34. J. Ye, Z. Chen, J. Liu, and B. Du, "TextFuseNet: Scene Text Detection with Richer Fused Features," in IJCAI, 2020, vol. 20, pp. 516-522.
35. K. Boukthir, A. M. Qahtani, O. Almutiry, H. Dhahri, and A. M. Alimi, "Reduced annotation based on deep active learning for arabic text detection in natural scene images," Pattern Recognition Letters, vol. 157, pp. 42-48, 2022.
36. L. Xiao, P. Zhou, K. Xu, and X. Zhao, "Multi-directional scene text detection based on improved YOLOv3," Sensors, vol. 21, no. 14, p. 4870, 2021.
37. N. Gandhewar, S. Tandan, and R. Miri, "Scene Text Detection Using YOLOv4 Framework," Bioscience Biotechnology Research Communications, vol. 13, no. 14, pp. 181-184, 2020.
38. N. K. El Abbadi, "Scene Text detection and Recognition by Using Multi-Level Features Extractions Based on You Only Once Version Five (YOLOv5) and Maximally Stable Extremal Regions (MSERs) with Optical Character Recognition (OCR)," Al-Salam Journal for Engineering Technology, vol. 2, no. 1, pp. 13-27, 2023.

39. C. N. Devi, N. Kartheek, B. P. Manikanta, M. Y. Saisree, and Manjeet, “Meetei Mayek, Hindi, and English Text Detection from Natural Scene Images Using YOLO,” in International Conference on Science, Technology and Engineering, 2023, pp. 327-336: Springer.
40. H. Turki, M. Elleuch, and M. Kherallah, “Using an Optimal then Enhanced YOLO Model for Multi-Lingual Scene Text Detection Containing the Arabic Scripts,” in Pacific-Rim Symposium on Image and Video Technology, 2023, pp. 451-464: Springer.
41. X. Wang, S. Zheng, C. Zhang, R. Li, and L. Gui, “R-YOLO: A real-time text detector for natural scenes with arbitrary rotation,” Sensors, vol. 21, no. 3, p. 888, 2021.
42. S. M. Beitzel, E. C. Jensen, and O. Frieder, "Map," Encyclopedia of Database Systems, pp. 1691-1692, 2009.
43. S. Saluky, G. B. Nugraha, and S. H. Supangkat, “Enhancing Abandoned Object Detection with Dual Background Models and Yolo-NAS,” International Journal of Intelligent Systems Applications in Engineering, vol. 12, no. 2, pp. 547-554, 2024.
44. J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas,” Machine Learning Knowledge Extraction, vol. 5, no. 4, pp. 1680-1716, 2023.
45. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.
46. Y. Liu, Y. Zhao, Y. Chen, Z. Hu, and M. Xia, “YOLOv5ST: A Lightweight and Fast Scene Text Detector,” Computers, Materials Continua, vol. 79, no. 1, 2024.
47. Jennifer June Mohanraj, Navin M George, Gunamony Shine Let, and Gokul J., “A Hybrid KNN–SVD, CNN, and RoBERTa-Enhanced Framework for Sentiment Analysis of OTT Film Reviews,” International Journal of Computer Information Systems and Industrial Management Applications, vol. 18, no. 5s, pp. 275–288, 2026, doi: 10.70917/ijcisim-2026-2706.
48. Rebecca Sandra Paul and C. Emilin Shyni, “Goal Aware Fine Grained Personalized Learning Framework Adaptive to Dynamic User Profile,” International Journal of Computer Information Systems and Industrial Management Applications, vol. 18, no. 12s, pp. 687–699, 2026, doi: 10.70917/ijcisim-2026-3941.
49. Y. Liu, Y. Zhao, Y. Chen, Z. Hu, and M. Xia, “YOLOv5ST: A Lightweight and Fast Scene Text Detector,” Computers, Materials Continua, vol. 79, no. 1, 2024.
50. M. Liao, B. Shi, and X. Bai, "TextBoxes++: A Single-Shot Oriented Scene Text Detector," IEEE Trans. Image Process., vol. 27, no. 8, pp. 3676-3690, 2018.[51] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character Region Awareness for Text Detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9365-9374.
51. M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time Scene Text Detection with Differentiable Binarization," in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, no. 07, pp. 11474-11481.
52. M. Bušta, L. Neumann, and J. Matas, "Deep TextSpotter: An End-to-End Trainable Scene Text Localization and Recognition Framework," in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2223-2231.
53. S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes," in Proc. Eur. Conf. Comput. Vis., 2018, pp. 20-36.
54. X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast Oriented Text Spotting with a Unified Network," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 5676-5685.
55. X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to Read Irregular Text with Attention Mechanisms," in Proc. 26th Int. Joint Conf. Artif. Intell., 2017, pp. 3280-3286.
56. M. Liao, B. Shi, and X. Bai, "TextBoxes++: A Single-Shot Oriented Scene Text Detector," IEEE Trans. Image Process., vol. 27, no. 8, pp. 3676-3690, 2018.
57. J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized Convolutional Neural Networks for Mobile Devices," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4820-4828.
58. Y. Baek et al., "Character Region Awareness for Text Detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9365-9374.
59. E. Velloso and M. Almeida, "Ethical Implications of Computer Vision: A Critical Examination," AI & Society, vol. 36, no. 1, pp. 361-375, 2021.