Received Date: 8th December, 2025
Revision Date: 19th December, 2025
Accepted Date: 22nd January, 2026

Traditional Nepali Object Detection using Fine-Tuned YOLOV8

Bhumika Magar[1*], Jessica Dangol[2], Neharika Shrestha[3], Nistha Rajbhandari[4], Sharad Chandra Joshi[5]

[1]Dept. of Computer Engineering, Kathmandu Engineering College, Nepal. Email: bhumikamagar13@gmail.com
[2]Dept. of Computer Engineering, Kathmandu Engineering College, Nepal. Email: jcka.dangol@gmail.com
[3]Dept. of Computer Engineering, Kathmandu Engineering College, Nepal. Email: neharikashrestha2003@gmail.com
[4]Dept. of Computer Engineering, Kathmandu Engineering College, Nepal. Email: rnistha22@gmail.com
[5]Assoc. Professor, Dept. of Computer Engineering, Kathmandu Engineering College, Nepal. Email: sharad.joshi@kecktm.com.np

Abstract — Visual learning supports comprehension by using images and videos to present information in an intuitive way. This study develops and optimizes an object detection model for integration into Drishti Kosh, a mobile visual dictionary designed to identify and translate traditional Nepali objects. A dataset containing 3,591 images across twelve culturally significant classes was used to fine-tune the YOLOv8n model. Three optimization algorithms—Stochastic Gradient Descent, Adaptive Moment Estimation, and AdamW—were evaluated to determine the most effective configuration for real-time use. Among them, the model trained with the AdamW optimizer demonstrated superior performance, achieving the highest recall value of 0.8753 and a mean Average Precision at 50 percent of 0.6545. It also produced the fastest inference time of 2.15 milliseconds and the lowest post-processing time of 0.98 milliseconds, along with a low training loss of 0.00087, indicating stable convergence. The results show that a lightweight and culturally aware object detection model can support real-time mobile deployment and enhance user interaction in multilingual environments. This contributes to improved accessibility for learners, tourists, and general users engaging with traditional Nepali objects in everyday contexts.

Keywords —- YOLOv8n, Visual Learning, Object Detection, AdamW, Nepali objects

I. INTRODUCTION

Advancements in computer vision have enabled the development of real-time object detection systems capable of identifying and localizing objects within images and video streams. When combined with multilingual dictionary support and text recognition, such systems can provide practical and immediate assistance in areas such as education, tourism, and accessibility. These technologies can facilitate dynamic interaction with the environment, making them useful for both native and non-native users navigating multilingual or unfamiliar contexts.

Early work on visual dictionaries demonstrated the value of combining images with textual definitions to improve comprehension and retention. Gesing et al. introduced VisDict, a manually curated visual dictionary for computational science concepts, which later incorporated automation and community feedback for expansion and validation as shown in [1]. Building on this pedagogical foundation, Anggraeni et al. conducted a controlled study comparing visual dictionaries to general dictionaries in eighth-grade classrooms, finding that the visual dictionary group achieved significantly higher vocabulary gains (mean score 81.98 vs. 74.24) as shown in [2].

However, the application of such systems in Nepal remains limited. Existing object detection models are predominantly trained on datasets featuring Western objects and environments, leading to poor recognition performance for region-specific items. Furthermore, current applications often focus solely on visual identification, without providing accompanying linguistic or contextual information. This limits their utility for users who require a deeper understanding of what they encounter in their surroundings. In domains like tourism and education, the lack of culturally aware, AI-powered tools results in a fragmented and often inaccessible user experience.

The advent of deep learning revolutionized object detection, shifting from region-based approaches such as R-CNN and Faster R-CNN to single-stage detectors. The YOLO (You Only Look Once) family exemplifies this progression, with each version improving speed and accuracy. Terven and Cordova-Esparza provide an extensive review of YOLOv1 through YOLOv8 and YOLO-NAS, highlighting key innovations—anchor-free detection, enhanced backbones (e.g., CSPDarknet), decoupled heads, and advanced augmentation strategies—that make YOLOv8 both highly accurate and efficient for real-time applications as shown in [3]. For evaluating detection performance, robust metrics are needed. Padilla, Netto, and da Silva surveyed popular evaluation measures—including variants of Average Precision and interpolation techniques—and proposed a standardized implementation to ensure consistency across datasets and bounding-box formats [4].

The absence of localized real-time object detection solutions that also incorporate language support presents a significant

* Corresponding Author

gap in Nepal's technological landscape. To address this, we developed Drishti-Kosh, a mobile application that integrates object detection, text recognition, and multilingual dictionary features to create a culturally aware visual dictionary. The system leverages a fine-tuned YOLOv8n model, trained on a curated dataset comprising 3,591 images across twelve classes of culturally significant Nepali objects. Once an object is detected, the system provides its name and meaning through a multilingual dictionary interface.

While the full application integrates additional features, this paper focuses specifically on the development, training, and performance evaluation of the object detection module. Our goal is to create a lightweight and efficient model suitable for deployment on mobile devices, supporting a practical and scalable solution for enhancing information accessibility in multilingual and culturally rich environments.

II. MATERIALS AND METHODS

Data Collection and Preparation

The dataset used in this study was specifically curated to train a YOLOv8-based object detection model for recognizing twelve traditional Nepali objects: Aankhi Jhyal, Doko, Karuwa, Khukuri, Lakhey Mukut, Nanglo, Nepali Topi, Nyaphu Sikha, Panas, Stupa, Tapari, and Vajra. These culturally significant classes were selected to ensure the model's relevance and utility in identifying objects unique to Nepal.

The dataset initially consisted of 3,591 images, which were divided into 2,700 training images (75%), 565 validation images (16%), and 326 testing images (9%). To improve the model's ability to generalize, a comprehensive data augmentation process was applied, resulting in a final augmented dataset of 8,991 images, with the training set expanded to 8,100 images. All images were auto-oriented to ensure proper alignment and resized to fit within a 640×640 pixel square while maintaining the original aspect ratio, with white padding used where necessary.

The augmentation process included generating three variations for each original training image using transformations that simulate real-world variations. These included horizontal and vertical flipping to represent mirrored and inverted views of objects, brightness adjustments within a ±20% range to simulate different lighting conditions, Gaussian blur up to 1.7 pixels to reflect varying focus levels, and the addition of random noise affecting up to 0.93% of pixels to simulate image artifacts. Furthermore, bounding boxes were randomly rotated within a range of ±20 degrees to help the model learn to detect objects from different angles.

However, certain object classes remain underrepresented, potentially limiting the model's detection accuracy for those categories.

YOLOv8 Fine-tuning

YOLO (You Only Look Once) is a real-time object detection system that has revolutionized the field by introducing a unified approach to object localization and classification. The version, YOLOv8, further refines the architecture by integrating new features such as an anchor-free detection head, better backbone designs, improved loss functions, and support for transfer learning.

- *Backbone (CSPDarknet):* Extracts features efficiently, balancing performance and computational cost for both low- and high-level object detection.
- *Neck (FPN + PAN):* Feature Pyramid Network (FPN) enables multi-scale detection, while Path Aggregation Network (PAN) improves localization and feature aggregation.
- *Head (Decoupled Detection Head):* Separates classification and bounding box tasks, predicting bounding box coordinates, confidence scores, and class probabilities.

Training YOLOv8 involves refining the model's weights using labeled image data to improve object detection accuracy. This process is guided by key hyperparameters such as learning rate, batch size, epochs, image size, and weight decay, which influence learning speed, generalization, and computational efficiency. The model learns hierarchical features through its convolutional layers—starting from low-level patterns (e.g., edges) to high-level object representations. Each image is divided into a grid, and the model predicts bounding boxes, confidence scores, and class probabilities for each grid cell. Bounding boxes localize objects, confidence scores estimate object presence, and class probabilities determine object categories.

Post-processing in YOLOv8 involves applying Non-Maximum Suppression (NMS**)** to eliminate redundant detections. After the model predicts multiple bounding boxes, NMS calculates the Intersection over Union (IoU) between pairs of boxes and retains only the one with the highest confidence score when overlaps occur.

Fine Tuning YOLOv8: To adapt the pre-trained YOLOv8n model for domain-specific object detection, three different versions were fine-tuned on the custom dataset. While all models were trained on the same dataset, variations in optimizers and hyperparameters influenced their behavior and outcomes. The first model employed Stochastic Gradient Descent (SGD) and was trained over a longer duration to achieve stability. In contrast, the second model used the Adam optimizer, which generally achieves convergence in fewer epochs due to its adaptive learning capabilities. The third model leveraged the AdamW optimizer with a learning rate of 0.0005, benefiting from decoupled weight

decay to promote better generalization. Throughout training, validation was performed after each epoch to monitor model performance and prevent overfitting. Other hyperparameters, including augmentation strategies, image scaling, and tracking options, were kept at their default values as specified by the YOLOv8 framework. Performance metrics and comparative analysis of the three models are detailed below:

Table I

Hyperparameters for fine-tuning yolov8n model

Hyperparameter	**First Model (SGD Optimizer)**	**Second Model (Adam Optimizer)**	**Third Model (AdamW Optimizer)**
Epochs	100	85	100
Batch Size	16	16	16
Input Image Size	640	640	640
Initial Learning Rate (lr0)	0.001	0.0005	0.0005
Momentum	0.9	0.9	0.9
Weight Decay	0.0005	0.0007	0.0001
Patience	10	10	10
Optimizer	SGD	Adam	AdamW
cos_lr	False	False	True
Max Detections	100	100	100
Confidence Threshold	0.5	0.5	0.5
IoU Threshold	0.7	0.7	0.7
Box Loss	7.5	7.5	7.5
Class Loss	0.5	0.5	0.5
Objectness Loss	1.0	1.0	1.0

Evaluation Metrics: To evaluate the effectiveness of the YOLOv8 model in detecting traditional Nepali objects, we employed standard object detection metrics, including Precision, Recall, F1-score, and Mean Average Precision (mAP), as presented in Table II.

Precision quantifies the proportion of correctly predicted object instances out of all instances predicted as positive by the model. It reflects the model's ability to minimize false positives. Recall, on the other hand, measures the proportion of actual object instances that were correctly identified, indicating the model's sensitivity to relevant detections. The F1-score provides a harmonic mean of precision and recall, offering a balanced view of model performance.

To evaluate detection across various localization thresholds, we compute Average Precision (AP) at different Intersection over Union (IoU) levels. The Mean Average Precision (mAP) is then derived by averaging the AP values across all object classes, serving as a comprehensive measure of overall model accuracy.

TABLE II

YOLOv8 PERFORMANCE METRICS

Metrics	**Formula**
Precision (P)	$P = \frac{TP}{TP + FP}$
Recall (R)	$R = \frac{TP}{TP+FN}$
F1-Score	$F1 = \frac{2 * P * R}{P+R}$
mAP	$mAP = \frac{\sum_{k=1}^{n} APk}{n}$

The confusion matrix terminology used are as follows:

- True Positives (TP): Cases where the model correctly detects and classifies a traditional object.
- False Positives (FP): Cases where the model predicts an object that is not present or misclassifies it.
- False Negatives (FN): Cases where the model fails to detect an object that is present.
- True Negatives (TN): Cases where the model correctly identifies the absence of any target object.

III. RESULTS

The YOLOv8n model underwent training using a custom authentic dataset of traditional Nepali objects to detect and classify the objects. The deployed fine-tuned YOLOv8 object detection model was trained on a dataset consisting of 8991 images across twelve distinct object classes. *Table III* presents comparison of different models' performances along with the different optimizers.

TABLE III

VALIDATION METRICS OF APPLIED YOLOv8 FINE TUNED MODEL

Optimizer	**Precision**	**Recall**	**mAP @ 0.5**	**mAP @ 0.5-0.95**
SGD	0.915	0.865	0.914	0.639
Adam	0.906	0.869	0.910	0.646
AdamW	0.904	0.875	0.915	0.655

TABLE IV

EVALUATION METRICS PER CLASS FOR FINE-TUNED YOLOv8N WITH SGD OPTIMIZER

Class	**Precision**	**Recall**	**mAP @ 0.5**	**mAP @ 0.5-0.95**
All	0.929	0.847	0.92	0.649
Aankhi Jhyaal	0.814	0.841	0.886	0.671
Doko	0.918	0.593	0.795	0.443
Karuwa	0.972	0.861	0.978	0.753

Khukuri	0.949	0.949	0.977	0.734
Lakhey Mukut	0.996	1	0.995	0.799
Nanglo	0.887	0.876	0.924	0.696
Nepali Topi	0.896	0.831	0.889	0.704
Nyaphu Sikha	1	0.958	0.995	0.739
Panas	0.98	0.87	0.919	0.577
Stupa	0.935	0.693	0.892	0.518
Tapari	0.876	0.898	0.919	0.589
Vajra	0.93	0.795	0.869	0.568

The performance metrics for each object class in the fine-tuned YOLOv8n model with the SGD optimizer are summarized in *Table IV*. The model demonstrates a precision of 0.929 and recall of 0.847 for all classes. The mAP50 of 0.919 shows a strong performance at the 50% IOU threshold, but the mAP50-95 of 0.649 indicates that performance drops when evaluated across a range of IOU thresholds. In terms of speed, the model had a preprocessing time of approximately 0.83 seconds, an inference time of 4.14 seconds per image, and a post-processing time of about 1.23 seconds.

Table V

Evaluation metrics per class for fine-tuned yolov8n with adam optimizer

Class	Precision	Recall	mAP @ 0.5	mAP @ 0.5-0.95
All	0.909	0.854	0.916	0.651
Aankhi Jhyaal	0.82	0.829	0.891	0.688
Doko	0.93	0.56	0.776	0.463
Karuwa	0.94	0.914	0.97	0.739
Khukuri	0.938	0.949	0.98	0.664
Lakhey Mukut	1	0.894	0.986	0.771
Nanglo	0.853	0.888	0.922	0.705
Nepali Topi	0.876	0.86	0.88	0.681
Nyaphu Sikha	0.923	1	0.995	0.738
Panas	0.966	0.833	0.902	0.591
Stupa	0.893	0.827	0.905	0.538
Tapari	0.898	0.897	0.928	0.64
Vajra	0.876	0.803	0.858	0.597

In *Table V*, these values show that the Adam-optimized model has a precision of 0.909 and a recall of 0.854 for all classes, indicating competitive performance. The mAP50 score is 0.916, just slightly lower than the SGD model, while the mAP50-95 value of 0.651 is also similar, suggesting a comparable drop-off in performance at stricter IOU thresholds. In terms of speed, the Adam-optimized model showed a preprocessing time of 1.84 seconds, inference time was 4.22 seconds and post-processing time was 1.48 seconds, slightly slower than SGD-optimized model.

Table VI

Evaluation metrics per class for fine-tuned yolov8n with adamw optimizer

Class	Precision	Recall	mAP @ 0.5	mAP @ 0.5-0.95
All	0.904	0.875	0.915	0.655
Aankhi Jhyaal	0.851	0.906	0.923	0.716
Doko	0.884	0.669	0.782	0.477
Karuwa	0.973	0.886	0.983	0.753
Khukuri	0.936	0.915	0.970	0.667
Lakhey Mukut	0.994	1.000	0.995	0.816
Nanglo	0.834	0.888	0.910	0.710
Nepali Topi	0.844	0.873	0.872	0.699
Nyaphu Sikha	0.918	1.000	0.994	0.734
Panas	0.908	0.884	0.897	0.579
Stupa	0.874	0.834	0.912	0.550
Tapari	0.908	0.852	0.902	0.608
Vajra	0.921	0.795	0.842	0.545

In *Table VI*, the values indicate that the model demonstrates competitive performance with a precision of 0.904 and recall of 0.875, showing strong detection accuracy with relatively few false positives. The mAP50 score of 0.915 indicates solid performance at the 50% IOU threshold, while the mAP50-95 value of 0.655 suggests a comparable drop-off in performance at stricter IOU thresholds. In terms of speed, the model showed a preprocessing time of 0.8ms, an inference time of 2.2ms per image, and a post-processing time of 1.0ms, indicating good speed for real-time applications.

IV. Discussion

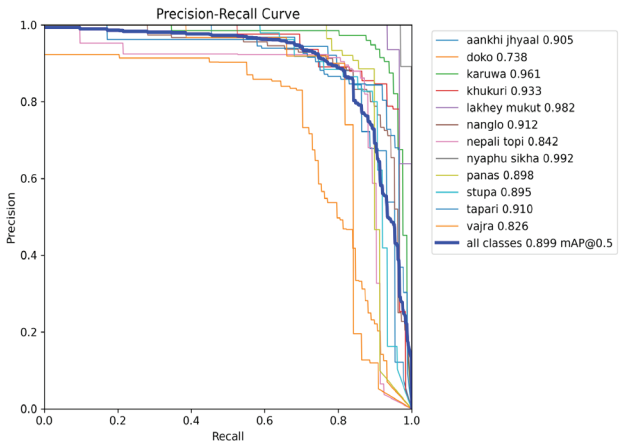

Fig. 2 (a). Precision-Recall Curve for Fine-Tuned YOLOv8n with SGD Optimizer

In *Fig. 2 (a)*, the presence of lower precision at higher recall levels suggests that when the model tries to capture more true positives, it also increases false positives. Evaluating on the validation set confirms that the model generalizes well. These results indicate the model performs well on unseen data but requires refinement to improve recall without increasing false positives.

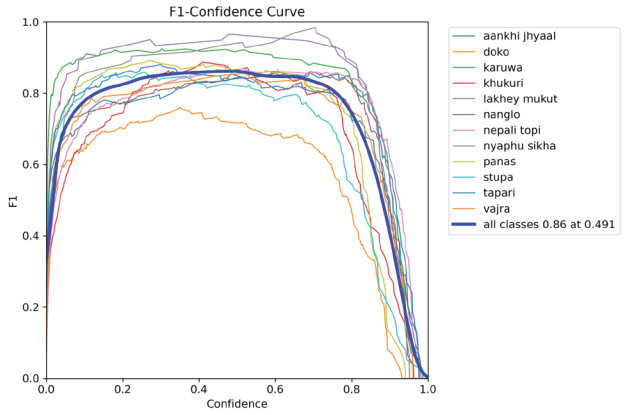

Fig. 2 (b). F1 Curve for Fine-Tuned YOLOv8n with SGD Optimizer

In *Fig. 2 (b)*, the overall F1-score reaches 0.86 at a confidence threshold of 0.491, meaning that predictions made with confidence greater than 0.49 yield the best balance between precision and recall. The majority of classes achieve high F1-scores, with slight variations.

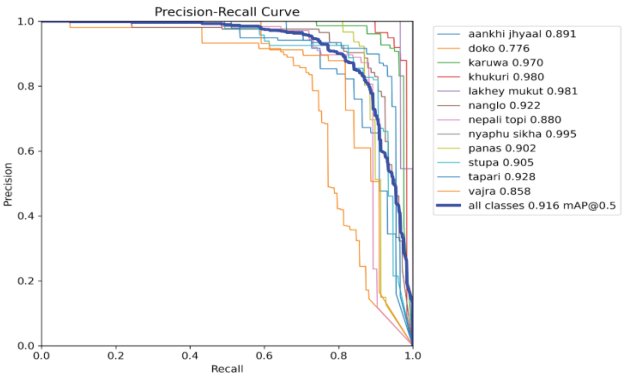

Fig. 3 (a). Precision-Recall Curve for Fine-Tuned YOLOv8n with Adam Optimizer

The precision-recall curve *(Fig. 3 (a))* demonstrates similar overall performance. The lower confidence threshold compared to SGD-optimized suggests that this model detects objects at a slightly earlier stage but may be prone to more false positives.

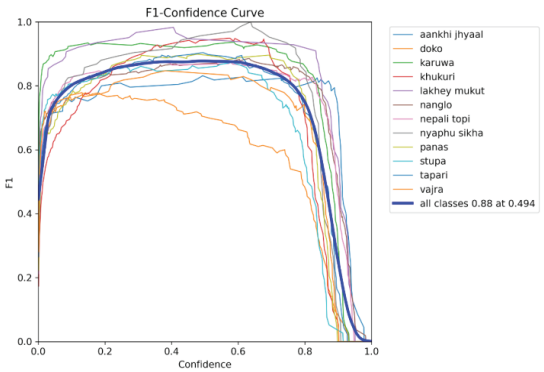

Fig. 3 (b). F1 Curve for Fine-Tuned YOLOv8n with Adam Optimizer

The model's F1-confidence curve at *Fig. 3 (b)* peaks at 0.494 confidence, indicating that the model achieves its best performance at a lower detection confidence compared to the previous model. While this suggests that the model is slightly more recall-oriented, allowing it to detect more objects at lower confidence levels, it also means that it may be more prone to false positives.

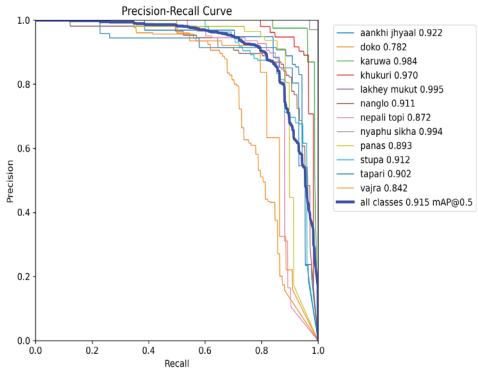

Fig. 4 (a). Precision-Recall Curve for Fine-Tuned YOLOv8n with AdamW Optimizer

In *Fig. 4 (a)*, the PR curve demonstrates the model's strong overall performance, indicating high detection accuracy across classes. The presence of a steeper decline in precision at higher recall levels for these lower-performing classes suggests that as the model captures more true positives, it also misclassified more instances, leading to reduced precision.

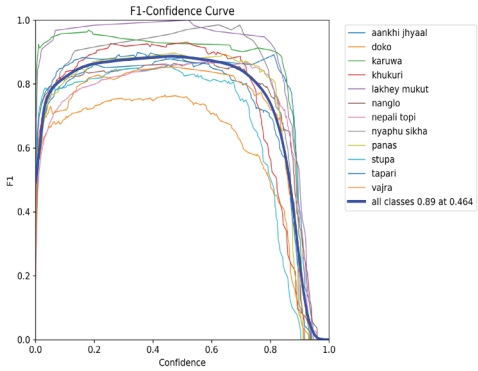

Fig. 4 (b). F1 Curve for Fine-Tuned YOLOv8n with AdamW Optimizer

In *Fig. 4 (b)*, the overall F1-score peaks at 0.89 at a confidence threshold of 0.464, indicating that predictions made above this confidence level optimize precision-recall balance. The sharp drop in F1-score at lower confidence thresholds indicates an increase in misclassifications when the model makes uncertain predictions. The validation set results confirm that the model generalizes well.

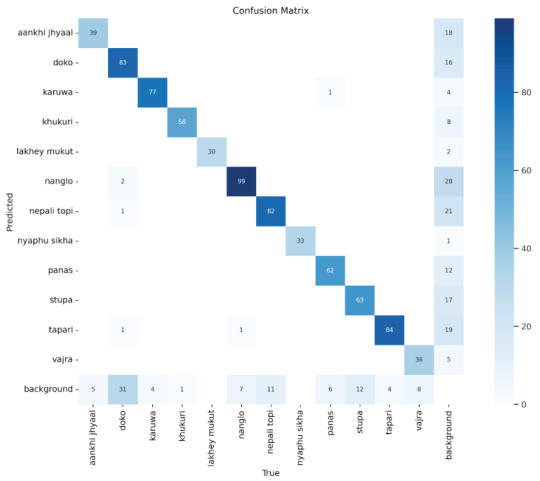

Fig. 5. Confusion Matrix (SGD Fine-Tuned YOLOv8n)

In *Fig. 5*, the model performs well on classes such as, ‘karuwa’, ‘nyaphu sikha’ and ‘lakhey mukut’ with minimal errors, whereas ‘doko’, ‘nepali topi’ and ‘stupa’ frequently get misclassified as background, suggesting difficulty in distinguishing these objects from their surroundings. Some classes, such as ‘tapari’ and ‘nanglo’ show confusion with each other, likely due to their similar circular shape, leading to overlapping feature recognition.

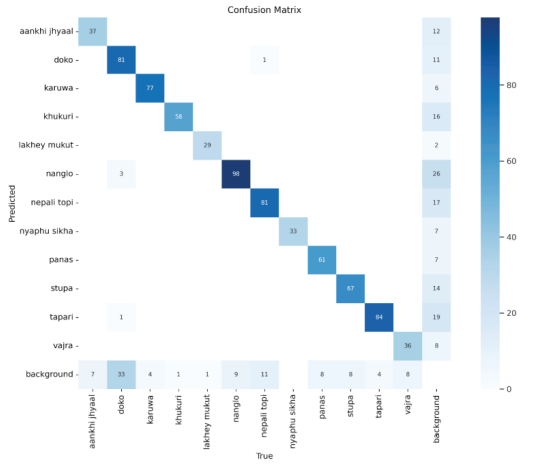

Fig. 6. Confusion Matrix (Adam Fine-Tuned YOLOv8n)

In *Fig. 6*, the model demonstrates strong performance in recognizing 'nyaphu sikha', with very few misclassifications. However, 'doko' and 'nepali topi' are often incorrectly identified as background, indicating difficulty in differentiating them from their surroundings. Additionally, the background is frequently mistaken for 'nanglo' and 'tapari', suggesting visual similarities. A notable pattern of confusion is observed between 'doko' and 'nanglo', likely due to their comparable textures.

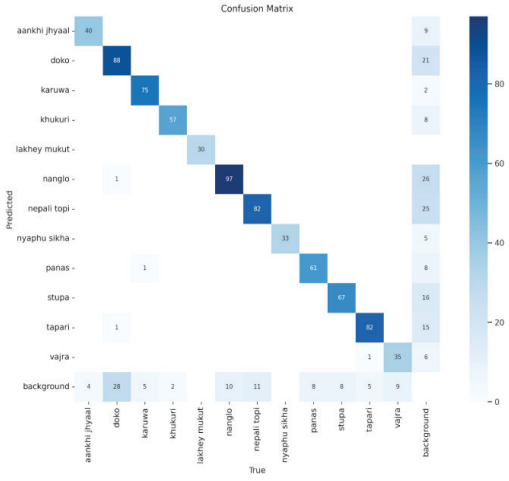

Fig. 7. Confusion Matrix (AdamW Fine-Tuned YOLOv8n)

Most diagonal values are high, indicating strong overall accuracy in *Fig. 7.* High-performing classes like ‘karuwa’, ‘nyaphu sikha’, and ‘lakhey mukut’ show minimal errors, demonstrating the model's strength in recognizing these objects. However, misclassifications between ‘tapari’ and ‘nanglo’ suggest that their similar shapes contribute to feature confusion. The background itself is occasionally misclassified, potentially due to dataset noise or feature overlaps. While the model performs well overall, refining feature extraction for ‘doko’, ‘nanglo’, and ‘vajra’ could improve classification accuracy.

From the validation metrics for all models *(Table III)*, mAP (Mean Average Precision) is considered the most important metric among other parameters in YOLO object detection, because it provides a comprehensive evaluation of both precision and recall across various levels of object detection confidence thresholds. Notably, the YOLOv8n model achieved the highest mAP of 0.915 for Adam W optimizer fine tuned model. In terms of detection metrics, the AdamW model demonstrates the highest recall (0.881), ensuring fewer missed detections, while maintaining competitive precision (0.902). Additionally, it achieves the best mAP50-95 score (0.648), indicating strong performance across varying IoU thresholds.

V. Conclusion

This research presents the development and optimization of a YOLOv8n object detection model for integration into Drishti Kosh, a mobile visual dictionary designed to identify and translate traditional Nepali objects. To determine the most effective configuration for real time deployment, three optimization algorithms—SGD, Adam, and AdamW—were evaluated. Among them, the model trained with the AdamW optimizer achieved the best overall performance. It recorded the highest recall value of 0.8753 and a mAP50 95 of 0.6545, while also exhibiting the fastest inference time of 2.15 milliseconds and the lowest postprocessing time of 0.98 milliseconds. Furthermore, it achieved a low training

loss of 0.00087, indicating stable and efficient convergence.

These characteristics make the AdamW optimized model highly suitable for mobile environments where computational resources are limited and inference speed is critical. Despite constraints such as a relatively small dataset and hardware limitations, the model demonstrated reliable detection performance, particularly for frequently encountered objects.

Acknowledgement

We would like to express our sincere gratitude towards the faculty members of the Department of Computer Engineering, Kathmandu Engineering College, for their constant support and guidance while conducting this research.

References

[1] S. Gesing et al., "*VisDict: Improving Communication Via a Visual Dictionary in a Science Gateway*," Computing in Science & Engineering, vol. 25, no. 2, pp. 7-11, 2023.

[2] R. Anggraeni, A. Ngafif, and Z. Chasanah, "*The Effectiveness of Using Visual Dictionary to Teach Vocabulary at the Grade Students*," Scripta: English Department Journal, vol. 8, pp. 11-19, 2021. https://doi.org/10.37729/scripta.v8i2.1196.

[3] J. Terven and D. Cordova-Esparza, "*A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond,*" 2023. [Online]. Available: https://arxiv.org/pdf/2304.00501.

[4] R. Padilla, S. Netto, and E. A. B. da Silva, "*A Survey on Performance Metrics for Object-Detection Algorithms*," 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil, 2020, pp. 237-242, doi: 10.1109/IWSSIP48289.2020.9145130. Available: https://www.researchgate.net/publication/343194514_A_Survey_on_Performance_Metrics_for_Object-Detection_Algorithms.