Lightweight Machine Learning Model for Hierarchical Bounding Boxes in Animal Tracking through Edge AI

YI-CHUN LO, GIAN ZIGNAGO, and TAI-CHANG ZHOU, University of California, Los Angeles, USA

In this report, we present a lightweight machine learning model that implements hierarchical bounding boxes for real-time tracking of tigers on edge computing devices. Utilizing the efficient MobileNets architecture and a two-tier bounding box strategy for whole animal and detailed body part detection, the model demonstrates high accuracy and computational efficiency in edge AI environments. Validated using the ATRW dataset, our approach surpasses baseline models in mean average precision (mAP). The findings underscore the potential of deploying scalable, efficient animal tracking systems on edge devices, with significant implications for wildlife monitoring, conservation efforts, and ethological studies.

Additional Key Words and Phrases: Edge Computing, Lightweight Machine Learning, Animal Tracking, Hierarchical Bounding Boxes, MobileNets, AI-driven Ethology, Real-time Object Detection, Computational Efficiency, Wildlife Monitoring

1 INTRODUCTION

In the rapidly evolving landscape of the Internet of Things (IoT), the fusion of artificial intelligence (AI) and edge computing brings the power of AI algorithms directly to devices at the network's edge. However, deploying sophisticated AI models on edge devices poses significant challenges, primarily due to their limited computational power and energy resources. This constraint necessitates the development of lightweight, efficient machine learning (ML) models that can operate within these limitations while still delivering accurate and reliable results. Such advancements have profound implications across various fields, including ethology, where AI-driven research into animal behavior requires the deployment of robust, real-time tracking and analysis systems in dynamic, unstructured environments.

1.1 Topic of Implementation

Within the domain of AI-driven ethology research, there exists a pressing need for efficient and accurate object detection and tracking systems capable of monitoring wildlife, particularly animals belonging to the Panthera lineage of the Felidae family, such as tigers (Panthera tigris) [1]. The ability to track these animals and their specific body parts (e.g., torso, head, limbs, tails) in real-time video feeds is crucial for behavioral analysis, conservation efforts, and the study of animal dynamics in natural habitats. However, the complexity of developing lightweight ML models that can perform these tasks on edge devices, coupled with the need for high accuracy and reliability in detection and tracking, poses significant technical challenges.

1.2 Research Objectives

This study aims to address these challenges by developing a lightweight machine learning model specifically designed for the detection and tracking of tigers, as well as their individual body parts, in video footage. The primary objectives of this research are twofold: firstly, to create and evaluate a model capable of tracking the entire animal with a single bounding box; and secondly, to extend this model to track multiple body parts with additional bounding boxes.

Authors' address: Yi-Chun Lo, yichunlo0919@g.ucla.edu; Gian Zignago, grz@cs.ucla.edu; Tai-Chang Zhou, tzhouam@g.ucla.edu, University of California, Los Angeles, 404 Westwood Plaza, Los Angeles, California, USA, 90095.

2 LITERATURE REVIEW

Recent advancements in AI for wildlife conservation have seen various machine learning algorithms being employed for tiger detection. M.N.S. Ohee and M. Asif utilized YOLOv3 to achieve an 80% accuracy rate in tiger detection [2]. Complementing this, R. Wei et al. developed a YOLO-mini-tiger detector, optimizing YOLOv3 for use on compact devices with limited computing power [3]. Further exploring the potential of convolutional neural networks, researchers have applied Faster R-CNN to the ATRW tiger detection dataset, marking a first in the field and showing its superior efficacy over other deep learning models [4] [5]. Another study harnesses the SSDlite model, achieving a 0.955 mAP, underscoring its speed and accuracy for the task [6].

3 REQUIREMENTS

Our project sets forth specific requirements aimed at developing a lightweight yet highly capable machine learning model. Central is the capacity to track tigers with precision and efficiency. The project's foundational requirement involves employing a single bounding box to accurately follow the entire tiger, necessitating a model that aligns with the low overhead constraints of edge devices. Expanding upon this foundation, our system must support more granular tracking by identifying and following distinct body parts of the tiger, including the head, tail, front legs, rear legs, and body trunk. This involves training a model to recognize and accurately track these individual components using separate bounding boxes. The implementation of this multi-level tracking system via hierarchical bounding boxes necessitates a robust evaluation mechanism to assess the accuracy of both the entire animal and its body parts tracking.

4 ISSUES

The development of our model faced significant challenges, notably the computational limitations of edge devices. These constraints demand a model capable of operating in resource-limited settings. Challenges are further compounded by the dynamic nature of tigers, including their varied postures and interactions within a scene, and by the variability in video quality, which affects tracking accuracy. Moreover, the detailed tracking of individual body parts requires a dual-model approach, a challenge given the computational restrictions of edge devices.

4.1 Our Approach to Addressing These Issues

We addressed the challenges of dynamic tiger postures and variable video quality using hierarchical bounding boxes and a preprocessing pipeline for data standardization, including dynamic range normalization and color correction. To manage the identification of individual body parts within the computational limits of edge devices, we implemented a two-tier part architecture: in Part 1, we detect the tiger, triggering Part 2, where we initiate detailed body part recognition upon successful initial detection.

5 TRAINING THE MODEL

5.1 Data Collection and Preprocessing

Our approach to training a comprehensive model for tiger detection and body part identification entailed the utilization of the Amur Tiger Re-identification in the Wild (ATRW) dataset, sourced through collaboration between the World Wildlife Foundation and MakerCollider. This dataset features 1080p-resolution images of 92 individual tigers across approximately 10 zoos. For Part 1 of our model, we used a subset of the ATRW dataset formatted in Pascal VOC, consisting of 2,485 training images and 277 validation images. The focus here was on the global detection of the tiger,

with annotations encompassing the entire body of the animal. Our training and validation sets were configured using the MediaPipe dataloader for the data handling and model feeding processes. For Part 2, the dataset's format switched to COCO, providing annotation detail for each tiger's head, torso, legs, and tail. This part included 2,187 training images and 342 validation images, allowing for the segmentation and distinct recognition of each body part.

5.2 Model Selection

In our model selection search, emphasis was placed on low-latency inference, small model size for ease of deployment, and low power consumption to accommodate the continuous operation in field conditions. We ultimately selected the MobileNet architecture within the TensorFlow ecosystem, which utilizes depthwise separable convolutions to significantly reduce the number of parameters without a decrease in performance [7]. In tandem with the MobileNet architecture, we leveraged the MediaPipe framework for its robust machine learning pipeline capabilities, enabling us to build custom models optimized for real-time on-device vision applications [8]. Its ability to process video streams with high performance facilitated the rapid iteration and testing of our models in a simulated real-world environment. OpenCV (CV2) allowed us to process and transform the image data required for the training of the model. It also provided the necessary functionality to augment our dataset through techniques such as image scaling, cropping, and rotation, enhancing the model's ability to generalize from diverse and challenging data inputs [9].

5.3 Training the Model

The training methodology for our tiger detection model is visually summarized in the system diagram shown in Figure 1. The initial phase, 'Extract the Whole Tiger,' involves preparing the dataset focused on identifying the tiger as a single entity. This first part of our model is trained using images with bounding boxes encompassing the entire tiger, ensuring the model can accurately detect the full presence of the animal in a frame.

Upon successful whole tiger detection, we proceed to 'Part 2' of our system, 'Extract Tiger's Body Part,' wherein the model is refined with the Pose Dataset to identify and segment individual body parts. Here, we implement a more granular model that delineates the tiger into its constituent components (the head, torso, and legs). Our approach ensures that the single model is capable of interpreting and learning from the pose variations to accurately recognize and track the specified body parts.

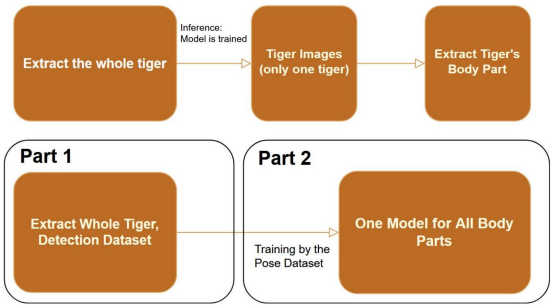

Fig. 1. Pipeline for data processing and model training

During the training sessions, we tuned the hyperparameters to cater to the dual demands of our model: identifying the whole tiger with high accuracy and segmenting its body parts with precision. For Part 1, we employed a learning rate of 0.01 with a minimal L2 weight decay of 0.00001, gradually refining this to a lower learning rate of 0.001 with an even smaller weight decay to prevent overfitting. In Part 2, the learning rate was initially set higher at 0.3 to quickly adapt to the nuances of detailed body parts, combined with a weight decay of 0.001 to maintain model generalization. As training progressed, these parameters were adjusted to lower rates to finely tune the model's precision.

6 RESULTS

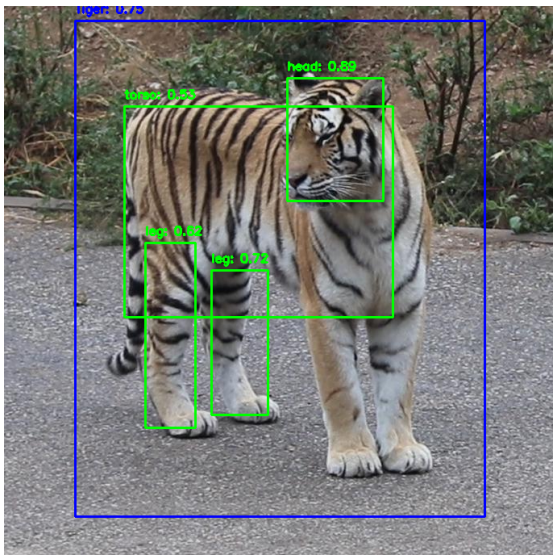

Fig. 2. Result of hierarchical bounding box placement using our model

In the results obtained from our model, the bounding boxes for tigers and their specific body parts were correctly identified and superimposed on the test images. In Figure 2, the bounding box encompassing the tiger's torso is appropriately scaled and placed, with a confidence score of 0.78, which demonstrates the model's ability to correctly identify and localize primary features of the tiger. The bounding box around the legs, with a confidence score of 0.58, also aligns well with the respective body part.

6.1 Negative Results

While our model successfully applied bounding boxes with high confidence to major body parts in the provided images, illustrating effective detection capabilities, we must acknowledge that the results were not uniform. In the cases illustrated in Figure 2 and Figure **??**, there were notable disparities in bounding box scaling, suggesting limitations in our model's spatial adaptability for certain poses and movements. These inconsistencies point towards the dataset's constraints, primarily its limited size and inherent biases, such as the orientation of the tigers and inconsistent image quality.

7 EVALUATION

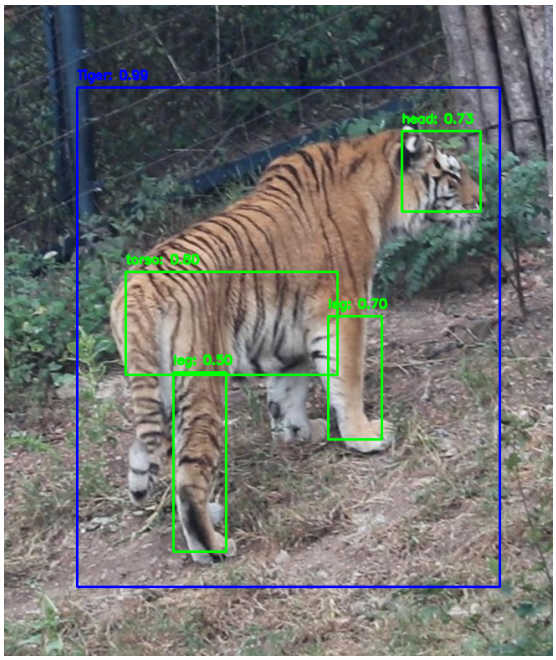

Fig. 3. Incorrect bounding box scaling around tiger torso section

Quantitative evaluation was conducted using the COCO metrics [10] on the validation subset of the dataset, and our full evaluation results can be seen in Table 1. The detection model achieved a mean Average Precision (mAP) of 0.269 across various Intersection over Union (IoU) thresholds, and a mean Average Recall (AR) of 0.366, indicating the model's ability to detect tigers with moderate accuracy. Despite these promising results, there was room for improvement, particularly in the accuracy of the part segmentation model.

Our model demonstrates reasonable data fit with a total loss of 0.9486, and its classification loss of 0.5232 indicates accurate image region classification. The low bounding box loss of 0.0065 suggests precise object localization. The difference between the model loss (0.8458) and total loss implies effective optimization of primary training objectives. COCO metrics reveal our model achieves an Average Precision (AP) of 26.9% across IoU thresholds

Table 1. Model evaluation metrics

Metric	Value	Description
Total Loss	0.9486	Combined classification and box regression loss
Classification Loss	0.5232	Accuracy of tiger classification
Bounding Box Loss	0.0065	Precision of bounding box localization
Model Loss	0.8458	Combined loss for model optimization
Average Precision (AP) at different IoU thresholds		
AP@[IoU=0.50:0.95]	0.269	Precision over multiple IoU thresholds
AP@[IoU=0.50]	0.580	Precision at 50% IoU threshold
AP@[IoU=0.75]	0.216	Precision at 75% IoU threshold
AP@[area=small]	0.001	Precision for small tigers
AP@[area=medium]	0.096	Precision for medium tigers
AP@[area=large]	0.391	Precision for large tigers
Average Recall (AR) for different numbers of detections		
AR@[maxDets=1]	0.295	Recall with max 1 detection per image
AR@[maxDets=10]	0.354	Recall with max 10 detections per image
AR@[maxDets=100]	0.366	Recall with max 100 detections per image
AR@[area=small]	0.024	Recall for small tigers
AR@[area=medium]	0.207	Recall for medium tigers
AR@[area=large]	0.478	Recall for large tigers

0.50 to 0.95, with better performance at a 50% overlap (AP50: 58.0%) and challenges at tighter IoU thresholds (AP75: 21.6%). The AP across object sizes—small (0.001), medium (0.096), and large (0.391)—highlights the model's efficiency in detecting larger targets and also reveals significant room for improvement in processing images of smaller tigers or those captured under less-than-ideal conditions.

Our quantitative evaluation using COCO metrics indicated that our model performs moderately well. However, the dataset's limitations – its restricted size and the quality variability of the images – have undoubtedly impacted these results. The biases present in the dataset and the challenges posed by images with extreme contrasts or dim lighting conditions have hindered the model's ability to generalize effectively.

8 ROADBLOCKS

The primary challenges impacting our results stemmed from the dataset itself. Despite the robust framework for detection, the relatively small size of the dataset proved to be a significant factor in the model's underperformance. Efforts to mitigate bias included the exclusion of images where tigers were not facing the camera, aiming to provide the model with the most informative views for detection and identification. However, the dataset exhibited inconsistencies in video quality, such as variable contrast levels, lighting conditions, and scale of the tigers within the images, further complicating the task of creating a universally robust model. Computational resource limitations were another significant hurdle during training, which we partially addressed through optimized training processes and resource management. Furthermore, our dual-model system's complexity caused synchronization issues, with dependencies causing tracking inaccuracies. We resolved this by ensuring frames were processed sequentially by the models, thereby improving tracking accuracy and system reliability.

9 FUTURE

Given additional time in the future, our primary focus for this project would be on refining and retraining our model to enhance its evaluation results and the accuracy of the generated bounding boxes. Additionally, a deeper dive into our dataset to identify and rectify any biases or gaps, along with an expansion of our training dataset to include more varied scenarios and tiger behaviors, could significantly contribute to the model's learning and generalization capabilities.

10 CONCLUSION

Our developed model achieves a promising balance between accuracy and computational efficiency. With a mean Average Precision (mAP) of 0.269 across varying Intersection over Union (IoU) thresholds and a mean Average Recall (AR) of 0.366, the model demonstrates moderate accuracy in detecting tigers in diverse conditions. Notably, the precision for detecting larger tigers is higher, indicating the model's usability in practical wildlife monitoring scenarios. These results affirm the model's potential to operate in ethological studies and conservation efforts within the computational constraints of edge environments.

10.1 Lessons and Insights

Through our work with MediaPipe and OpenCV, we learned the value of sequential processing for interdependent models and the critical nature of dataset quality and diversity, as careful calibration of spatial information is crucial for improving detection precision. We hope others can learn from the roadblocks we overcame, particularly in our attempts to address shortcomings and biases in our selected dataset.

11 SOURCE CODE

Our source code for this project is available and can be accessed here.

REFERENCES

[1] Z. Zhang, Z. He, G. Cao, and W. Cao, "Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification," *IEEE Trans. Multimedia*, vol. 18, pp. 2079–2092, Oct. 2016.

[2] M. N. S. Ohee and M. Asif, "real-time tiger detection using yolov3," *International Journal of Computer Applications*, vol. 975, p. 8887, 2020.

[3] R. Wei, N. He, and K. Lu, "Yolo-mini-tiger: Amur tiger detection," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 517–524, 2020.

[4] M. Z. Altobel and M. Sah, "Tiger detection using faster r-cnn for wildlife conservation," in *14th International Conference on Theory and Application of Fuzzy Systems and Soft Computing–ICAFS-2020 14*, pp. 572–579, Springer, 2021.

[5] B. Liu and Z. Qu, "Af-tigernet: A lightweight anchor-free network for real-time amur tiger (panthera tigris altaica) detection," *Wildlife Letters*, vol. 1, no. 1, pp. 32–41, 2023.

[6] S. B. Ghosh, K. Muddalkar, B. Mishra, and D. Garg, "Amur tiger detection for wildlife monitoring and security," in *Advanced Computing: 10th International Conference, IACC 2020, Panaji, Goa, India, December 5–6, 2020, Revised Selected Papers, Part II 10*, pp. 19–29, Springer, 2021.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017.

[8] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," June 2019.

[9] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, Y. Kwon, K. Michael, L. Changyu, J. Fang, P. Skalski, A. Hogan, *et al.*, "ultralytics/yolov5: v6. 0-yolov5n'nano'models, roboflow integration, tensorflow export, opencv dnn support," *Zenodo*, 2021.

[10] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, 2021.