ML-Based Road Asset Geolocation Using Object Detection and Camera Displacement

Victor Israel Anchieta de Medeiros*, Julio Cezar Soares Silva*,
Gabriely Barbosa*, Celice Alessandra Melato Argenta*,
Fabio Schmitz Ruver *, Thiago Meirelles Ventura†, Luciano Lourenço da Silva‡,
Rafael Rodrigues Vitale‡ and Raoni Florentino da Silva Teixeira§
*Nova Rota do Oeste, Mato Grosso, Brazil
Email: {victormedeiros,juliosilva,gabrielybarbosa,celiceargenta,fabioruver}@rotadooeste.com.br
†Instituto de Computação, Universidade Federal de Mato Grosso, Mato Grosso, Brazil
Email: thiago.ventura@ufmt.br
‡ANTT Brasília, Brazil
Email: {luciano.lourenco,rafael.vitale}@antt.gov.br
§Faculdade de Ciência e Tecnologia, Universidade Federal de Mato Grosso, Mato Grosso, Brazil
Email: raoni.teixeira@ufmt.br

***Abstract*—Geolocation methods identify objects in images and determine their geospatial locations. Current object geolocalization methods face challenges such as high hardware costs, limited object class coverage, difficulties with repeated object occurrences, and performance issues in dynamic environments. This paper introduces a machine learning approach for geolocalizing objects from low frame rate video using a single camera and image metadata, aiming to reduce costs and complexity compared to traditional methods. The method combines camera displacement data and object bounding boxes obtained from an object detection model to estimate geospatial locations. The approach was evaluated using diverse datasets that capture various driving environments and object types, demonstrating its capability to handle multiple scenarios.**

***Index Terms*—Geolocation, Road Management, Object Detection, Highway**

I. Introduction

Object geolocalization is the task of identifying objects within one or more images and determining their geospatial location, which is expressed as global positioning system (GPS) coordinates. This process has a wide range of applications, including land surveying, self-driving vehicles, and asset management [1], [2], [3]. Additionally, it benefits other fields that require automatic detection and geolocation of objects of interest [3], [4].

Automatically detecting GPS locations of objects from street images can be a cost-effective solution for road asset geolocalization. However, this approach presents several challenges, including GPS errors, the presence of multiple appearances of the same objects in different images or frames, and the variety of object types and sizes (e.g., road signs with multiple sub-classes). Objects may appear in one, two, or several images, necessitating an algorithm that can detect and consolidate these occurrences into a single prediction.

Existing methods are often limited to a single object class, rely on robust visual feature descriptions for effectiveness, and frequently require expensive equipment, such as drones, multiple cameras, or satellites [5].

In this paper, a multi-class machine learning approach for geolocalizing objects from low frame rate video was evaluated. This approach processes one or multiple frames and utilizes inexpensive hardware, relying solely on a single camera and the image's metadata. It enables the extraction of the camera's displacement relative to the previous frame, as well as the object bounding box obtained from an object detection model.

A public dataset capturing various driving environments was used to benchmark performance. It features diverse sign types and metadata that allow for organizing and sequencing images to provide information about camera displacement. This approach applies to geolocating road features, street markings, traffic lights, sidewalks, trees, buildings, and other elements. It is crucial for assessing the quality and maintenance of these objects, such as identifying damaged infrastructure, fading markings, or malfunctioning lights.

This paper is organized as follows. In Section II, the related work on object geolocalization is presented. Materials and methods are presented in section III. The results and discussions are elaborated in Section IV. Finally, the paper ends with the conclusions in Section V.

II. Related Work

Krylov et al. [6] proposed a triangulation-based method that uses a two-stage framework for object segmentation followed by geolocalization. This approach was further enhanced in [7] with drone point cloud footage to improve accuracy. However, these methods have limitations due to noisy segmented objects and assumptions of object sparsity, treating all objects within a distance threshold as a single entity.

Re-identification methods were introduced by Nassar et al. [8], where the model detects and geolocalizes objects in two images. They later proposed a graph-based method to handle multiple frames, which requires objects to appear in at least two frames and assumes proximity to the camera for easier detection [2].

Chaabane et al. [1] proposed a tracking-based method that utilizes a largely end-to-end trainable deep neural network to geolocalize traffic signs. Their approach required objects to appear in at least five frames and used six cameras, which faced hardware limitations.

Before the advent of deep learning, geolocalization commonly employed epipolar constraints [9] to reconstruct 3D points, such as for traffic lights [10] and traffic signs [11]. A related method [12] used a pipeline for telecom assets, leveraging HOG features and linear SVM [13]. However, these methods were limited by their reliance on handcrafted features.

Deep neural networks (DNNs) now dominate geolocalization by effectively capturing complex relationships, merging object detections from multiple images, and estimating depth in images obtained from omnidirectional cameras [14]. Various single-object tracking deep learning frameworks have been developed [15], [16], [17], including visual cue tracking [18], [19], filter-based methods [20], [21], siamese network-based methods [22], [23], and transformer-based methods [24], [25].

A recent study integrated bounding box regression data and motion constraints for filter-based single-object tracking in satellite videos [26]. The motion model could adaptively learn and predict the target's future trajectory based on its historical movement patterns using a long short-term memory (LSTM) network.

Despite these advancements, existing methods are often limited to a single object class, rely on robust visual feature descriptions for effectiveness, and frequently require expensive equipment, such as drones, multiple cameras, or satellites [5]. Wilson et al. [27] developed a two-stage technique that detects and geolocalizes dense multi-class objects, such as traffic signs (with nearly 200 sub-classes), using low frame rate videos recorded by a single camera.

This paper proposes a machine learning approach designed to reduce equipment costs, building on the work of Wilson et al. [27]. The method utilizes data from a single camera, GPS location, and image metadata. Additionally, it combines camera motion data with the pixel coordinates of bounding boxes for tracked objects. This combination makes the method suitable for simple supervised learning models to estimate the locations of various objects, such as traffic signs, license plates, road barriers, and trees.

III. Materials and Methods

A. System Overview

Figure 1 shows the flowchart of the proposed methodology for geolocation. The process begins with a set of images for which we compute the camera displacement. We need at least two images to calculate the camera displacement. As illustrated in the flowchart, the first component of our approach is to detect the object bounding box using the Yolov8 algorithm [28]. We train a custom Yolov8 object detector with object location and asset class manual annotations.

With the bounding box and camera displacement coordinates in hand, the geolocation model predicts the object location displacement $(\Delta lat_{obj}, \Delta lon_{obj})$. The final object coordinate is computed by equation 1, as follows:

$$\begin{aligned} lat_{obj} &= lat_{cam} + 10^{-4} \Delta lat_{obj} \\ lon_{obj} &= lon_{cam} + 10^{-4} \Delta lon_{obj}, \end{aligned} \tag{1}$$

where (lat_{obj}, lon_{obj}) and (lat_{cam}, lon_{cam}) are the final object and input camera GPS coordiantes.

Figure 2 depicts the predicted coordinates superimposed on the input image.

B. Data

The public datasets used in this study, detailed in Table I, cover a range of scenarios with different numbers of images, classes, and annotations.

Compared to other traffic recognition and geolocation datasets, ARTS is the largest in terms of the number of images and annotations. The public ARTS datasets contains high-quality images with a resolution of 1920 × 1080, available in various formats including video logs and individual annotations in a format similar to PASCAL VOC.

TABLE I: Description of the public datasets

Dataset	N. Images	N. Annotations
ARTS easy [29]	6.141	16.540
ARTS challenging [29]	19.908	35.970
ARTS V2 [27]	25.544	47.589

20% of the samples were used as test data. The dataset used in this study incorporates both computer vision features and geospatial features, as shown in Figure 1. The computer vision features were extracted by YOLOV8n. When evaluated on the ARTs Challenging dataset, using 20% of the dataset as test data, the more recent YOLOv8 model achieves an accuracy of 89%, surpassing the performance reported by [27]. The labels of the images contain their latitude and longitude. The object and camera geospatial features are obtained by combining two sequential frames.

Table II shows examples of samples with features used by the machine learning model in this study. This dataset is utilized to train a neural network model for estimating the geolocation of objects. Each row in the table represents a unique data sample, consisting of the following columns:

- ΔLat_{cam}: The change in latitude of the camera, indicating how much the camera's latitude has changed relative to the previous frame. To perform better, this tiny value should be normalized by arbitrary value (we used the factor 10000).
- ΔLon_{cam}: The change in longitude of the camera, showing how much the camera's longitude has changed relative to the previous frame. To perform better, this tiny value should be normalized by arbitrary value (we used the factor 10000).
- bbox obj(x_1, y_1, x_2, y_2): The bounding box of the object in the image, defined by the pixel coordinates of its top-left corner (x_1, y_1) and bottom-right corner (x_2, y_2) normalized by the image shape.
- ΔLat_{obj}: the difference between the object and camera latitude. To perform better, this tiny value should be scaled by some factor (we used the factor 10000).
- ΔLon_{obj}: the difference between the object and camera longitude. To perform better, this tiny value should be scaled by some factor (we used the factor 10000).

The camera displacement and the relative pixel coordinates of the bounding boxes' bottom-right and top-left corners provide the necessary input features for the model. The object coordinates distance (relative to the camera coordinates) serve as the output labels, enabling the model to learn the

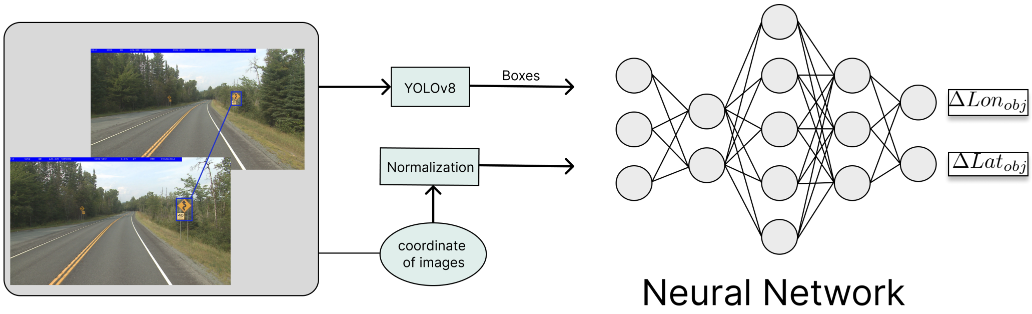


Fig. 1: Proposed system flowchart

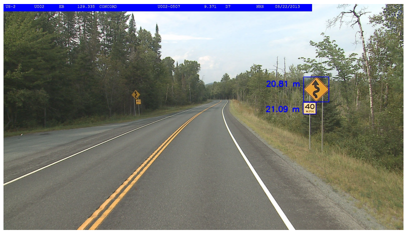


Fig. 2: Predicted object geolocation relative to the camera.

relationship between these variables.

TABLE II: Features used to train the geolocalization model

ΔLat_{cam}	ΔLon_{cam}	bbox obj norm(x_1, y_1, x_2, y_2)	ΔLat_{obj}	ΔLon_{obj}
0.6704	-0.2722	(0.7417, 0.4370, 0.7526, 0.4926)	-1.4987	-0.5925
0.6704	-0.2722	(0.6750, 0.3417, 0.7167, 0.4157)	-2.4578	-0.4316
0.7123	-0.2926	(0.9422, 0.4546, 0.9630, 0.5574)	-0.7864	-0.8851
0.7123	-0.2926	(0.7578, 0.3093, 0.8188, 0.4148)	-1.7455	-0.7242
0.6817	-0.2917	(0.4385, 0.4093, 0.4448, 0.4231)	-6.8924	4.2631

C. *Model for object geolocation*

The geolocation model was built using PyTorch and trained on a computer with an Intel Xeon W7-2495X processor, an NVIDIA GeForce RTX A5500 GPU, and 128GB of RAM. Table III outlines the architecture of the neural network used for object geolocation. This network consists of a series of linear layers designed to progressively transform the input data through various dimensions. It starts with a small number of input features and undergoes several transformations to capture complex patterns.

TABLE III: Architecture of the adopted Neural Network for object geolocation

Layer	**Type**	**Input Size**	**Output Size**	**Activation**
1	Linear	6	32	ReLU
2	Linear	32	64	ReLU
3	Linear	64	128	ReLU
4	Dropout	-	-	-
5	Linear	128	512	ReLU
6	Linear	512	1024	ReLU
7	Dropout	-	-	-
8	Linear	1024	1024	ReLU
9	Linear	1024	512	ReLU
10	Dropout	-	-	-
11	Linear	512	128	ReLU
12	Linear	128	64	ReLU
13	Linear	64	32	ReLU
14	Linear	32	2	None

Dropout layers were incorporated to enhance generalization and prevent overfitting, with dropout probabilities ranging from 0.1 to 0.3. The final layer produces a two-dimensional output, which aligns with the requirements for object geolocation tasks. The adopted loss function was the Mean Squared Error (MSE), defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2}$$

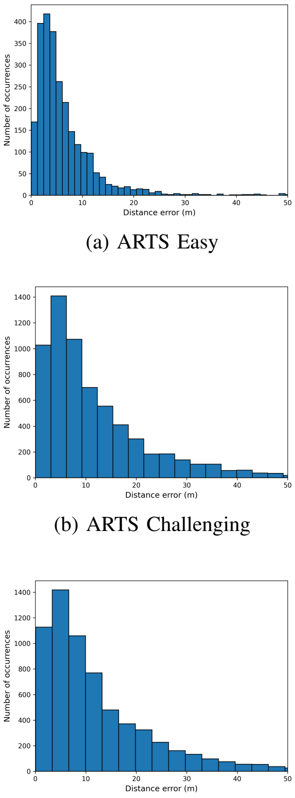
where y_i are the target values and $\hat{y}_i$ are the predicted values. The Adam optimizer was initialized with a learning rate of 0.001, and a scheduler with gamma factor of 0.5 and step size of 10 epochs.

IV. Results

Table IV presents summary statistics, including the mean, the mean confidence interval, and the median of distance errors for the considered datasets. For the ARTS Easy category, the model achieved

TABLE IV: Distance error summary statistics

Dataset	Mean (CI)	Median
Easy	6.32 [6.09, 6.57]	4.53
Challenging	12.26 [11.97, 12.57]	8.44
Arts V2	13.12 [12.80, 13.46]	8.73



(a) ARTS Easy

(b) ARTS Challenging

(c) ARTS V2

Fig. 3: Distance error distributions in the test subsets.

a mean error of 6.32 meters, a median of 4.53 meters. This suggests a relatively high accuracy in this simpler subset. In contrast, the results for bigger datasets indicated greater difficulty in accurately predicting object positions. In the ARTS Challenging the model presented a mean error of 12.26 meters and a median of 8.44 meters. Similarly, the ARTS V2 subset presented a mean error of 13.12 meters and a median of 8.73 meters.

The results illustrated in Figure 3 highlight the distribution of distance errors across various test subsets. Overall, the model's performance degrades as the datasets become more challenging, as evidenced by the increasing skewness and kurtosis, which point to a greater frequency and magnitude of large errors. The increasing mean and median distance errors from ARTS Easy to ARTS Challenging and ARTS V2 suggests that the model may struggle with more challenging environments where factors such as varied lighting conditions, occlusions, or diverse object types complicate accurate geolocation.

Although there is less precision in more complex datasets compared to the model proposed by Wilson [27], the former approach used a geolocation model with approximately 25M parameters, while ours contains 2M parameters. This significant reduction in the number of parameters results in a more efficient model with faster inference times and reduced computational requirements. Despite the trade-off in precision, our model remains competitive and offers practical advantages in scenarios where resources are limited or rapid processing is essential.

V. Conclusion

In this study, the challenge of object geolocalization, which involves identifying objects in images and determining their geospatial locations as GPS coordinates, was addressed. By providing geolocation predictions using accessible hardware, the approach offers a practical solution for evaluating and maintaining road infrastructure, enhancing the ability to identify issues such as damaged or missing road signs, faded markings, and malfunctioning traffic signals.

The results showed the model's effectiveness across varying levels of difficulty in the datasets. The ARTS Easy category shows the lowest error rates, indicating higher accuracy, while the ARTS Challenging and ARTS V2 categories demonstrate greater variability and error, likely due to more complex scenarios.

The observed increase in error across more challenging datasets suggests potential areas for improvement in the model. For instance, enhancing the model's ability to handle complex environmental factors, such as different terrain types or atmospheric conditions, could reduce errors. Moreover, refining the model's algorithms to better generalize across diverse conditions might lead to more consistent performance. Future work could focus on extending this system to handle real-time data, integrating additional low-cost sensors to further improve accuracy and on optimizing the balance between precision and efficiency to further enhance the model's performance.

Acknowledgment

The authors would like to thank the National Land Transport Agency (ANTT) for the resources provided under Resolution No. 483, of March 24, 2004, as well as for the infrastructure and coordination provided by Nova Rota do Oeste. The authors would also like to thank the Research Support Foundation of Mato Grosso – FAPEMAT (process FAPEMAT-PRO-2022/01047) for their support during the development of this work.

References

[1] M. Chaabane, L. Gueguen, A. Trabelsi, J. R. Beveridge, and S. O'Hara, "End-to-end learning improves static object geo-localization in monocular video," *CoRR*, vol. abs/2004.05232, 2020.

[2] A. S. Nassar, S. D'Aronco, S. Lefèvre, and J. D. Wegner, "Geograph: Graph-based multi-view object detection with geometric cues end-to-end," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII.* Berlin, Heidelberg: Springer-Verlag, 2020, p. 488–504.

[3] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman, "Shady dealings: Robust, long-term visual localisation using illumination invariance," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 901–906.

[4] N. Sünderhauf, S. Shirazi, A. Jacobson, E. Pepperell, F. Dayoub, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proceedings of Robotics: Science and Systems XII*, 07 2015.

[5] D. Wilson, X. Zhang, W. Sultani, and S. Wshah, "Image and object geo-localization," *International Journal of Computer Vision*, vol. 132, no. 4, p. 1350–1392, Nov. 2024.

[6] V. A. Krylov, E. Kenny, and R. Dahyot, "Automatic discovery and geotagging of objects from street view imagery," *Remote Sensing*, vol. 10, no. 5, 2018.

[7] V. A. Krylov and R. Dahyot, "Object geolocation using mrf based multi-sensor fusion," 10 2018, pp. 2745–2749.

[8] A. S. Nassar, S. Lefèvre, and J. D. Wegner, "Simultaneous multi-view instance detection with learned geometric soft-constraints," *CoRR*, vol. abs/1907.10892, 2019.

[9] R. Szeliski, *Computer Vision: Algorithms and Applications*, ser. Texts in Computer Science. Springer, 2011.

[10] N. Fairfield and C. Urmson, "Traffic light mapping and detection," 06 2011, pp. 5421 – 5426.

[11] B. Soheilian, N. Paparoditis, and B. Vallet, "Detection and 3d reconstruction of traffic signs from multiple view color images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 77, pp. 1–20, 03 2013.

[12] R. Hebbalaguppe, G. Garg, E. Hassan, H. Ghosh, and A. Verma, "Telecom inventory management via object recognition and localisation on google street view images," in *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 03 2017.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 07 2005, pp. 886–893.

[14] J. Bai, H. Qin, S. Lai, J. Guo, and Y. Guo, "Glpanodepth: Global-to-local panoramic depth estimation," *IEEE Transactions on Image Processing*, vol. 33, pp. 2936–2949, 2024.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[18] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 06 2019, pp. 7934–7943.

[19] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," 2019.

[20] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550.

[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[22] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 850–865.

[23] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4277–4286.

[24] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 428–10 437.

[25] Y. Cui, C. Jiang, L. Wang, and G. Wu, "Mixformer: End-to-end tracking with iterative mixed attention," in *2022*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 13 598–13 608.

[26] J. Fan and S. Ji, “Adaptive and anti-drift motion constraints for object tracking in satellite videos,” *Remote Sensing*, vol. 16, no. 8, 2024.

[27] D. Wilson, T. Alshaabi, C. Van Oort, X. Zhang, J. Nelson, and S. Wshah, “Object tracking and geo-localization from street images,” *Remote Sensing*, vol. 14, no. 11, 2022.

[28] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO,” Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[29] F. Almutairy, T. Alshaabi, J. Nelson, and S. Wshah, “Arts: Automotive repository of traffic signs for the united states,” *Trans. Intell. Transport. Syst.*, vol. 22, no. 1, p. 457–465, dec 2020.