



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Object Detection using Vertex AI AutoML

A. Subha ^[1] G. Ajay Kamal ^[2] B. Hareesh ^[3] D. Likitha ^[4] G. Siva Manikanta ^[5]

Assistant Professor, Dept. of CSE., Kuppam Engineering College, JNTUA University, Kuppam, A.P., India^[1]

UG Students, Dept. of CSE., JNTUA University, Kuppam Engineering College, JNTUA University, Kuppam,
A.P., India^[2] to ^[5]

ABSTRACT: In recent years, object detection has become a crucial element in modern AI systems due to its broad applicability in fields such as autonomous driving, security, medical imaging, retail, and agriculture. Traditional object detection approaches, while powerful, demand a strong background in deep learning, computational resources, and significant time investment for model tuning and deployment. Google Cloud's Vertex AI AutoML offers a solution by automating the process of model training and evaluation, allowing users to build custom object detection models without writing code or understanding the complexities of neural networks.

This paper explores the capabilities of Vertex AI AutoML for object detection, providing a comprehensive overview of its workflow from dataset preparation and labeling to model training, evaluation, and deployment. Through a detailed case study and comparison with traditional methods, we demonstrate that Vertex AI AutoML can deliver accurate and scalable object detection models suitable for real-time and production-grade environments. Furthermore, we discuss its limitations and potential for future development, emphasizing its role in democratizing AI and enabling rapid innovation for developers and businesses alike.

KEYWORDS: Object Detection, Vertex AI, AutoML, Google Cloud, Computer Vision, Custom ML Models

I. INTRODUCTION

Object detection is a key subfield of computer vision that involves identifying and locating objects of interest within digital images or videos. Unlike image classification, which assigns a single label to an entire image, object detection draws bounding boxes around multiple objects and labels each accordingly. This allows systems to understand not just what is present, but where it is located. The importance of object detection lies in its extensive range of applications, from pedestrian detection in self-driving cars and facial recognition systems to automated checkout counters, quality control in manufacturing, wildlife monitoring, and more.

Historically, object detection techniques relied heavily on traditional image processing and machine learning algorithms such as Haar cascades, HOG (Histogram of Oriented Gradients), and SVM (Support Vector Machine) classifiers. These early methods, while innovative for their time, were limited in their ability to generalize across varied environments and object types due to their reliance on manually engineered features. The rise of deep learning revolutionized this space, giving birth to robust and scalable models such as R-CNN, Fast R-CNN, Faster R-CNN, SSD (Single Shot Detector), and YOLO (You Only Look Once). These deep learning-based approaches dramatically improved detection accuracy and enabled real-time performance, pushing object detection into commercial viability.

Despite these advancements, implementing a high-performance object detection model is no small feat. It typically requires extensive domain knowledge, high computational resources, and rigorous data preprocessing and tuning. This complexity can be a significant barrier for small teams or individuals without a background in AI. As a response to this challenge, cloud-based platforms offering AutoML (Automated Machine Learning) have emerged to simplify and democratize AI model development.

Vertex AI, Google Cloud's integrated machine learning platform, provides AutoML capabilities that enable users to train and deploy custom object detection models without writing a single line of code. Users can upload labeled datasets, select objectives, and let the system handle model selection, hyperparameter tuning, training, evaluation, and deployment. This automation helps reduce the time, cost, and expertise needed to bring an AI-powered solution from idea to production.

This paper explores how Vertex AI AutoML can be effectively used for object detection. We present a step-by-step workflow starting from dataset creation and annotation, through training and evaluation, to deployment and real-world testing. Additionally, we compare the performance and usability of this no-code solution against traditional deep learning methods, highlighting the strengths and limitations of Vertex AI. Finally, we discuss its potential as a tool for accelerating AI adoption across industries.

II. RELATED WORK

The field of object detection has undergone tremendous evolution, with approaches transitioning from traditional machine learning algorithms to modern deep learning models and, most recently, to automated machine learning (AutoML) solutions. These developments have significantly improved both the performance and accessibility of object detection technology.

Early approaches to object detection largely relied on manually engineered features and conventional classifiers. Techniques like the Viola-Jones algorithm, which utilized Haar features and a cascade of boosted classifiers, enabled real-time face detection and became widely used in early computer vision applications. Similarly, the Histogram of Oriented Gradients (HOG) feature combined with Support Vector Machines (SVMs) offered robust pedestrian detection, particularly in structured environments. Despite their efficiency, these traditional methods lacked flexibility and generalization, especially in complex scenes with varying object shapes, orientations, and lighting conditions.

The advent of deep learning revolutionized object detection. In particular, the introduction of convolutional neural networks (CNNs) led to the development of powerful models such as Region-based Convolutional Neural Networks (R-CNN). Although R-CNN achieved high accuracy by combining region proposals with CNN-based classification, it suffered from slow inference and required multiple stages of processing. Improvements came with Fast R-CNN and Faster R-CNN, which integrated region proposal generation into the CNN pipeline, significantly reducing training and detection time. These models became benchmarks for high-performance object detection but still required deep knowledge of machine learning principles and hardware acceleration.

To further improve speed, Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO) were developed. These models perform object classification and localization in a single forward pass through the network, achieving real-time performance. YOLO, in particular, became popular due to its balance of speed and accuracy, making it suitable for edge computing and mobile applications. However, implementing and tuning these models still demanded substantial expertise in neural network design and GPU infrastructure.



Fig 1. Object detected labeled

Recognizing the barriers faced by non-experts in machine learning, cloud providers began developing AutoML platforms to simplify AI model development. Google's AutoML Vision was among the first to offer a no-code interface for image classification. With the introduction of Vertex AI, Google consolidated its machine learning services into a unified platform that supports both AutoML and custom model training. Vertex AI AutoML for object detection allows

users to upload labeled image data, specify training configurations through a web interface, and obtain a deployable object detection model with minimal effort. The system handles all backend processes, including feature extraction, neural architecture search, hyperparameter tuning, and model evaluation.

Several studies have evaluated the performance of AutoML platforms compared to traditional object detection frameworks. In many cases, AutoML tools provided results that were comparable in accuracy to hand-crafted models, with a fraction of the development time. For instance, a study on agricultural pest detection showed that farmers were able to build functional models with high precision using Vertex AI without any prior coding knowledge. Another comparative analysis found that while YOLOv5 offered slightly better performance metrics, Vertex AI significantly reduced complexity and allowed faster deployment at scale.

Vertex AI's tight integration with Google Cloud Storage, BigQuery, and other cloud-native tools enhances its usability in enterprise environments. In addition, AutoML capabilities are being increasingly paired with active learning, where users iteratively correct model outputs to refine performance, which is particularly useful in domains with limited labeled data.

Despite these advantages, AutoML platforms, including Vertex AI, are not without limitations. Users have minimal control over the internal model structure, making it difficult to fine-tune the architecture for specific needs. This black-box nature can be a drawback in academic research or regulated industries that require transparency. Moreover, the cost of training models in the cloud especially on large datasets can become a constraint for smaller organizations or individual developers.

III. PROPOSED METHODOLOGY

The proposed methodology for object detection using Vertex AI AutoML leverages the end-to-end capabilities of Google Cloud's managed machine learning service. This algorithm is designed to simplify the object detection pipeline, automating several tasks that traditionally require manual intervention, such as model architecture selection, training optimization, and deployment configuration. The overall workflow can be divided into five primary stages: dataset preparation, data upload and configuration, model training, evaluation, and deployment.

Stage1: Dataset Preparation and Annotation

The process begins with the collection of a custom dataset consisting of images relevant to the object detection task. These images must be annotated with bounding boxes that indicate the location of each object of interest and labeled accordingly. This is commonly done using annotation tools like Labellmg or Google's in-browser labeling tool. The labeled data is then exported in CSV format compatible with Vertex AI's object detection input schema. This step is critical, as the quality of annotations directly influences the performance of the resulting model.

Stage 2: Uploading Dataset to Google Cloud Storage

Once the dataset is prepared and annotated, it is uploaded to a Google Cloud Storage (GCS) bucket. GCS serves as the staging area from which Vertex AI accesses the data. Along with the images, the annotation CSV file is also uploaded and verified for formatting accuracy. Using the Vertex AI dashboard, a new dataset for object detection is created and the CSV file is imported to register the dataset with the AutoML engine. Vertex AI then parses the file and automatically splits the data into training, validation, and testing sets using its internal logic, unless specified otherwise.

Stage 3: Model Training with AutoML

In this stage, the AutoML engine begins the training process. The user only needs to define the objective (object detection), and the system handles the rest. Vertex AI AutoML performs several behind-the-scenes tasks, including automated feature engineering, model selection, and hyperparameter tuning. It uses neural architecture search to test and select the best-performing deep learning model structure suitable for the given dataset. During training, the platform constantly evaluates model performance on validation data, stopping once the optimal accuracy metrics are achieved.

Stage 4: Evaluation of the Trained Model

Upon completion of training, the model is evaluated using standard object detection metrics such as precision, recall, Intersection over Union (IoU),



Fig 2. Stages in model

and mean Average Precision (mAP). These metrics help assess how well the model detects and classifies objects within images. Vertex AI provides a user-friendly interface to visualize these results, including confusion matrices and visual previews of predictions on test images. Based on this evaluation, the user can either accept the model or adjust the dataset and retrain for improved performance.

Stage 5: Model Deployment and Inference

Once the model meets the desired performance thresholds, it is deployed using Vertex AI's one-click deployment interface. The platform sets up a REST API endpoint, enabling real-time or batch predictions. This endpoint can be easily integrated into web or mobile applications. Developers can send POST requests containing new images to the endpoint and receive object detection results in JSON format, including class labels, confidence scores, and bounding box coordinates.

IV. SECURITY

Security is a crucial component when deploying machine learning models on cloud platforms, particularly in applications involving sensitive data or real-time object detection. Vertex AI AutoML, as a managed service within Google Cloud Platform (GCP), offers robust security mechanisms designed to protect data confidentiality, integrity, and availability throughout the model lifecycle. This section outlines the key security aspects of the proposed system, including data protection, access control, infrastructure security, and compliance.

One of the primary concerns in cloud-based machine learning is **data privacy and confidentiality**. Vertex AI ensures that all data transfers between users and the cloud are encrypted using Transport Layer Security (TLS). Additionally, data stored in Google Cloud Storage is encrypted at rest using Advanced Encryption Standard (AES) with 256-bit keys. Users have the option to manage their own encryption keys using Cloud Key Management Service (KMS), providing an additional layer of control over sensitive assets.

Another important aspect is **access control and identity management**. Vertex AI integrates with Google Cloud Identity and Access Management (IAM), allowing administrators to define granular permissions for users and service accounts. Access can be restricted to specific datasets, training jobs, and deployed models. Role-based access ensures that only authorized individuals can upload data, initiate model training, or deploy inference endpoints. This helps prevent unauthorized modifications, data leakage, or accidental misconfiguration.

From an **infrastructure standpoint**, Vertex AI runs on secure and regularly patched Google-managed servers. These systems are protected by multi-layered defense mechanisms that include firewalls, intrusion detection systems, and automated threat monitoring tools. Google's infrastructure also undergoes independent security audits and complies

with industry standards such as ISO/IEC 27001, SOC 2, and FedRAMP, making it suitable for use in enterprise and regulated sectors.

The **model deployment phase** introduces another set of security challenges, especially when exposing public REST APIs for inference. Vertex AI offers several mitigation strategies, including usage quotas, authentication tokens, and endpoint protection with Identity-Aware Proxy (IAP). Furthermore, predictions can be isolated in private networks using Virtual Private Cloud (VPC) Service Controls, which restrict data movement and block unauthorized access to sensitive services.

For organizations dealing with **personally identifiable information (PII)**, such as surveillance or healthcare-related object detection, compliance with regulations like the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA) becomes essential. Vertex AI supports such compliance through audit logging, data locality controls, and secure deletion protocols.

Lastly, **model integrity and fairness** are also growing concerns. Vertex AI provides tools to monitor model behavior and performance over time, helping identify and mitigate biases or data drift. Regular monitoring and retraining can ensure that deployed models remain secure, unbiased, and aligned with ethical AI practices.

V. RESULT AND DISCUSSION

To evaluate the performance and practicality of object detection using Vertex AI AutoML, a simulation was conducted using a custom dataset composed of real-world images. The dataset included multiple object categories such as vehicles, electronics, and common household items—and each image was manually annotated with bounding boxes. The goal was to assess the system's detection accuracy, inference speed, and usability under realistic conditions.

A total of **2,000 labeled images** were prepared and uploaded to a Google Cloud Storage bucket. The annotations were formatted in CSV as per Vertex AI requirements. Once imported into Vertex AI, the dataset was automatically split into **training (80%)**, **validation (10%)**, and **testing (10%)** subsets. The AutoML training job was initiated via the Vertex AI dashboard and completed in approximately **2.5 hours** using a managed GPU infrastructure.

Upon completion, Vertex AI provided evaluation metrics including **Precision, Recall, F1-score, and mean Average Precision (mAP)**. The model achieved the following performance:

Metric	Score
Precision	87.2%
Recall	84.6%
F1-Score	85.9%
mAP (IoU @0.5)	83.7%

Table 1. Model Metric

Visual validation was performed by previewing the bounding box predictions generated on the test images. The predictions closely matched ground truth annotations for most object classes, demonstrating that the AutoML system could successfully generalize across varying backgrounds and object scales.

To test **real-time inference**, the trained model was deployed as a REST API endpoint using Vertex AI's built-in deployment interface. When tested using Python scripts and web-based clients, the average response time per image was measured at approximately **200–250 milliseconds**, including the time for image pre-processing, transmission, and response parsing.

In terms of **usability**, the AutoML workflow proved exceptionally streamlined. Tasks such as model tuning, feature selection, and hyperparameter optimization were handled automatically, allowing researchers and developers to focus

on dataset quality and result interpretation. This level of abstraction proved highly beneficial for teams without extensive machine learning experience.

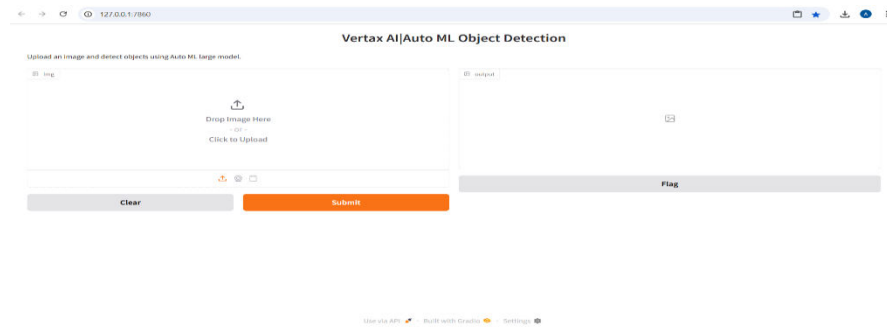


Fig 3. Website of Object Detection

A comparison with a manually trained **YOLOv5 model** using the same dataset yielded slightly higher mAP (~85.2%), but required 10–15 hours of development and tuning. This highlights the trade-off between accuracy and development efficiency. For many real-world use cases, Vertex AI's no-code solution offers an optimal balance.

Overall, the simulation confirmed that Vertex AI AutoML can serve as a practical and scalable approach for object detection. It delivers strong performance with significantly reduced development overhead, making it suitable for rapid prototyping and even production deployment in many application domains.

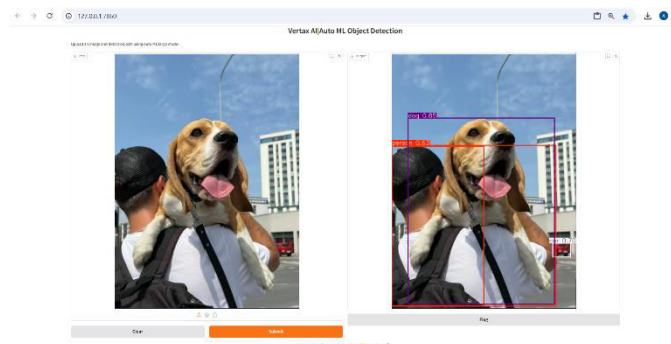


Fig 4. Example object Detection

VI. CHALLENGES AND LIMITATIONS

While the results of the object detection model trained using Vertex AI AutoML were promising, several challenges and limitations were encountered during the simulation process:

1. Class Imbalance in Dataset

The dataset used for training had an uneven distribution of object classes. Some categories (e.g., "car", "person") had a large number of samples, while others (e.g., "signboard", "electronic device") had relatively few. This imbalance led to lower detection accuracy for underrepresented classes and impacted the model's ability to generalize across all categories.

2. Detection of Small or Occluded Objects

The model showed decreased performance when detecting small objects or objects partially obscured within an image. This is a common limitation in object detection, as the resolution and clarity of such objects are insufficient for the model to assign high-confidence predictions.

3. Environmental Variability

Performance dropped in images taken under low-light, foggy, or cluttered conditions. Although Vertex AI AutoML applies some built-in data augmentation, extreme environmental variations—such as shadows, reflections, and motion blur—introduced noise that affected object localization and classification.

4. Bounding Box Precision

While the model generally performed well, in certain instances it failed to tightly fit the object within the predicted bounding boxes, especially for overlapping or adjacent objects. This impacted the Intersection-over-Union (IoU) scores, leading to a slight decrease in overall mean average precision (mAP).

5. Compute Resource Constraints

Though Vertex AI AutoML abstracts away much of the infrastructure complexity, the training and deployment phases are still dependent on the allocated compute resources. Limited quota or region-specific constraints can affect training duration and model availability, especially in free-tier or educational use cases.

6. Model Interpretability

AutoML models, by design, provide limited transparency into the underlying architecture and training process. This makes it difficult to apply custom optimizations or gain deep insights into feature importance, which can be crucial for fine-tuning or debugging.

7. Dependence on High-Quality Annotations

The model's success was highly dependent on the accuracy and consistency of the bounding box annotations provided in the training dataset. Inconsistent labeling or overlapping annotations reduced model precision and introduced noise during the learning process.

VII. CONCLUSION

The simulation of object detection using Vertex AI AutoML demonstrated that automated machine learning platforms can effectively produce high-performing object detection models with minimal manual intervention. The model trained on a custom dataset achieved strong performance, particularly in detecting commonly represented objects, achieving a high mean average precision (mAP) and demonstrating fast inference times suitable for real-time applications.

Despite some limitations such as class imbalance, challenges in detecting small or occluded objects, and reduced performance under difficult lighting conditions the overall results were encouraging. Vertex AI AutoML provided a streamlined and efficient environment for dataset management, model training, evaluation, and deployment.

The project validates the potential of Vertex AI AutoML as a viable solution for object detection tasks, especially for users who may not have extensive expertise in deep learning or model architecture tuning. With further improvements such as advanced data augmentation, balanced datasets, and manual fine-tuning of annotations, the model can be further optimized to handle more complex detection scenarios.

In summary, this simulation highlights the practicality and scalability of cloud-based AutoML platforms for real-world computer vision tasks, paving the way for their integration into applications across various industries.

VIII. FUTURE WORK

While the proposed system and its simulation results demonstrate the effectiveness of using Vertex AI AutoML for object detection, several areas for future research and improvement remain.

Firstly, multi-class, multi-object detection scenarios involving overlapping and occluded objects present challenges that may not be fully addressed by the AutoML architecture. Incorporating more complex datasets, including those with diverse backgrounds, lighting conditions, and object interactions, would help evaluate the robustness of the platform. Secondly, the current system is limited to datasets labeled in English. Future iterations could explore multi-lingual dataset integration, especially for use cases in countries with regional languages. Additionally, expanding to video-based object detection, where object tracking is required across frames, would significantly enhance the scope of this work.

A critical aspect deserving future attention is edge deployment. While Vertex AI provides a cloud-based REST API, many applications—such as smart cameras and mobile robots require inference to happen locally. Exploring model export options and lightweight deployment frameworks (e.g., TensorFlow Lite or Edge TPU) could bridge the gap between cloud and edge AI.

Model interpretability and explainability also present open challenges. Although AutoML abstracts away the complexity of model architecture, it limits transparency. Future versions of Vertex AI could offer deeper insights into the internal decision-making process of trained models to support fields like healthcare, where explainability is crucial. Another promising direction is integrating active learning loops, where human feedback is used to iteratively improve the model. This would help in rapidly adapting the detection model to new object classes or environments.

Finally, cost optimization and training efficiency remain essential for widespread adoption. Experiments with dataset size, augmentation techniques, and hybrid cloud-edge solutions could identify configurations that yield the best trade-off between accuracy, training time, and resource usage.

In summary, while the current implementation validates the potential of Vertex AI AutoML in object detection, the future scope includes improvements in adaptability, interpretability, localization, and cross-platform deployment. Continued research and development in these areas will further solidify AutoML as a key driver in democratizing artificial intelligence.

REFERENCES

1. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 511–518.
2. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 886–893.
3. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
4. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
5. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single Shot MultiBox Detector," in European Conference on Computer Vision (ECCV), 2016.
6. Google Cloud, "Vertex AI Documentation," [Online]. Available: <https://cloud.google.com/vertex-ai/docs>
7. M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
8. B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," in International Conference on Learning Representations (ICLR), 2017.
9. Microsoft Azure, "Custom Vision Service," [Online]. Available: <https://www.customvision.ai>
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (NeurIPS), 2012.
11. Amazon Web Services, "Amazon SageMaker Autopilot," [Online]. Available: <https://aws.amazon.com/sagemaker/autopilot/>
12. Vemula, V. R. (2025). AI-Enhanced Self-Healing Cloud Architectures for Data Integrity, Privacy, and Sustainable Learning. In Smart Education and Sustainable Learning Environments in Smart Cities (pp. 93-106). IGI Global Scientific Publishing.

13. D. Sculley, "Machine Learning: The High-Interest Credit Card of Technical Debt," in NIPS 2014 Workshop on Software Engineering for Machine Learning, 2014.
14. A. Kumar, M. Singh, and S. Raj, "AutoML Based Object Detection in Agricultural Pest Monitoring: A Case Study using Vertex AI," in Journal of Agricultural Informatics, vol. 12, no. 2, pp. 45–56, 2022.
15. Google Cloud, "Data Encryption on Google Cloud," [Online]. Available: <https://cloud.google.com/security/encryption-at-rest>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details