Journal of Information Systems Engineering and Management
2025, 10(13s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Object Detection In Real-World Scenarios Using Artificial Intelligence And Machine Learning Technologies

Senada Bushati (Hoxha)[1*], Viola Bakiasi (Shtino)[2]
[1,2] *Department of Computer Science, University 'Aleksander Moisiu', Durres, Albania*
Email: senadabushati@uamd.edu.al, violashtino@uamd.edu.al

ARTICLE INFO

Received: 28 Nov 2024

Revised: 09 Jan 2025

Accepted: 31 Jan 2025

ABSTRACT

This study explores the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques for object detection in real-world scenarios, with a particular focus on Albania. The purpose of the research is to develop and evaluate advanced object detection models that can enhance accuracy and reliability in various applications such as security and road safety. The study employs an experimental approach, leveraging the YOLO algorithm and Convolutional Neural Networks (CNNs) to train and evaluate customized object detection models using diverse datasets. Comparative analyses are conducted to identify the most effective methodologies. The findings demonstrate that larger, high-quality datasets significantly enhance model performance, as evidenced by a maximum F1-score of 0.96 achieved with 80 training images and 50 epochs. The research highlights the transformative potential of AI-driven object detection in improving processing speed and accuracy for critical applications. Challenges such as computational resource limitations and dataset constraints are identified as barriers to broader implementation. The study concludes with practical recommendations for improving model scalability and reliability, emphasizing the importance of integrating AI with complementary technologies for real-world deployment. These insights have implications for policymakers, developers, and industries aiming to leverage AI for enhanced safety and efficiency in infrastructure and beyond.

Keywords: Object Detection, Machine Learning, Artificial Intelligence, YOLO, CNNs, Computer Vision.

INTRODUCTION

Automatic detection and identification of objects in images is an important challenge in the field of Artificial Intelligence (IA) and Machine Learning (ML). With many applications in areas such as security, road safety, medical diagnostics, and camera image analysis, this capability has great potential to be put into practice - not only globally, but also in the specific case of Albania. However, the development of automatic object detection systems remains a complex problem that requires advanced solutions. Some of the key challenges include:

- The variety of objects to be identified; Objects can have different shapes, colors, and characteristics, requiring flexible and well-adapted algorithms.
- Changes in the environment where the image is located; lighting, viewing angles, and other factors can affect the quality and appearance of images, making detection difficult.
- Needs for high accuracy and reliability: In most critical applications, the object detection system must provide accurate and reliable results to be implemented successfully.

This paper aims to address these challenges by exploring and using advanced AI and ML techniques for object detection in images. By building and testing sophisticated models, we will seek to improve the accuracy and reliability of these systems to make them applicable in Albania as well. So, this study aims to fulfill the following main objectives:

I. Building and training advanced ML models for object detection in images, referring to data in different environments.
II. Analyzing and comparing the performance of different models trained using IA and ML, to identify the most effective models in object detection.
III. We will study the main obstacles and challenges that exist for the successful implementation of object detection systems, determining the next steps for improvement.

IV. Based on the results achieved, we will present concrete recommendations for government institutions, companies, and other actors interested in the effective implementation of these technologies in Albania.

The objective of this study is the implementation of a customized ML model based on the YOLO algorithm, which will result in an efficient and accurate system for detecting objects in real images.

LITERATURE REVIEW

This paper study reviews various object detection methods, from classical techniques (like Haar cascades) to modern neural networks (e.g., YOLO and Vision Transformers). It evaluates their performance (accuracy, speed) and practical use cases, such as robotics, infrastructure monitoring, and video analytics. Opportunities for improvement include integrating hybrid methods, addressing resource constraints, and exploring recent innovations like transformers. Practical applications highlight the relevance of real-world challenges. Below we have listed the conclusions of papers studied in reference concepts with object detection in images/videos, machine learning, deep learning, techniques, and algorithms.

These authors used a shape encoder to extract pose variation from the guide image, along with an appearance encoder to extract identity information from the reference image. The extracted features were then fused by a decoder to generate virtual samples [1].

The research in object detection using machine learning has made considerable progress and continues to drive innovations in computer vision applications. The findings from the literature survey provide a comprehensive understanding of the state-of-the-art techniques, challenges, and future directions in object detection, facilitating further advancements in this field. Researchers and practitioners can leverage these insights to develop improved algorithms and systems for a wide range of applications, contributing to advancements in areas such as autonomous driving, surveillance, healthcare, and more [2].

Object detection is an active area of research that is constantly evolving, and there are several promising future directions that researchers are exploring. Multi-modal detection can be helpful in applications such as autonomous driving, where multiple sensors detect objects around a vehicle [3].

With the emerging techniques of deep learning and machine learning, object detection can be an essential and helpful tool to ease our daily lives. With the growth in the IT sector, object detection will become an even more effective technique in the upcoming years [4].

The Efficient-U-Net network is used as the segmentation model and the YOLO v5 network is utilized as the detection model to identify cracks accurately while simultaneously segmenting the cracks in roads. The issue of inaccurate crack localization in road crack detection is addressed by integrating a segmentation model with a detection model. The issue of inaccurate crack localization in road crack detection is addressed by integrating a segmentation model with a detection model. The suggested system's accuracy is 99.35%, which is more than that of any existing methods [5].

Finally, from the above-mentioned methodologies, it can be inferred that the Single Shot Detector (SSD) has the greatest mAP of all the techniques, at 76.8%. This review is especially useful for improvements in neural networks and related learning systems since it provides useful insights and suggestions for future advancement [6].

By optimizing the model through hyperparameter tuning and efficient data augmentation, this research demonstrated a balanced approach that ensures both high accuracy and reasonable speed, making it adaptable for real-world applications. The comparison with existing state-of-the-art models, such as YOLO and Faster R-CNN, revealed competitive performance, with the proposed model excelling in specific use cases like small object detection and low-latency requirements. In this research, we explored the effectiveness of various machine learning and neural network approaches for object detection. The results demonstrated significant improvements in accuracy and processing speed when using advanced architectures such as convolutional neural networks (CNNs). By contrasting these methods with conventional techniques, we underscore the transformative capabilities of deep learning in real-time object recognition and classification. The findings underscore the importance of continued innovation in model design and training strategies to further enhance performance in diverse applications [7].

The images are classified into training, validation, and testing sets. The images are passed through the proposed model and the output results obtained are discussed in detail. The results obtained are compared with other traditional methods like VGG16 and VGG19 in terms of accuracy and loss. The efficiency of image recognition is also compared. The comparisons show that the proposed model outperforms the other models considered for accuracy, loss, and efficiency [8].

Our project embodies a strong, real-time detection and recognition system of traffic signs that exhibits high precision rates and well-performing in real-time with a user interface created on streamlet. We obtained the dataset for training our model by carefully picking it out and preparing it through steps like binary thresholding, scaling, and color space conversion. Our deep learning model which used TensorFlow and Keras achieved an incredible validation accuracy of 99.6% after 15 epochs (calculated using 1). A more engaging user experience is made possible by integrating live display with dynamic bounding boxes around detected signs as well as audible feedback via beeping sound upon recognition and sending notification alerts through SMS to the driver's phone number. This technology can have multiple applications such as traffic management systems or autonomous vehicles; thus, making road navigation safer and more efficient. Studying optimizations for handling complexities associated with complicated real-world scenarios may enhance future research toward enhancing its efficiency and applicability [9].

Convolutional Neural Networks (CNNs) are structured as a series of layers, each designed to recognize different features in the input images. They have been highly successful in various tasks in computer vision, including image and video recognition, and image classification [10].

Real-time object detection and tracking in high-definition videos is crucial for applications like video surveillance and autonomous driving. Traditional object detection models are typically designed for image-wise detection and often overlook the temporal correlations between video frames. Improving detection by exploring the spatial and temporal correlation under the calculation limitation is an important research direction [11].

Convolutional Neural Networks (CNNs), which have been developed over the past decade, are well-established in tasks such as object detection, segmentation, and classification. It is difficult for the Transformer method to completely replace CNN in a short time [12].

Although convolutions have numerous advantages and are widely adopted, we believe there is still room for improvement in areas such as model size, security, and Neural Architecture Search (NAS). Additionally, there are challenges that convolutions struggle with, including poor generalization, lack of equivariance, and suboptimal performance in crowded scenes. As a result, several promising research directions have been identified. Finally, some hardware implementation schemes for CNN are discussed [13].

This study develops an object detector algorithm using deep learning neural networks for detecting objects from images. This study utilizes an enhanced SSD algorithm combined with a multilayer convolutional network to achieve high-accuracy real-time object detection. The algorithm performs well on both static images and video content. The accuracy of the proposed model is more than 79.8%. The training time for this model is about 5–6 hours. These CNNs extract features from the input image and subsequently map those features to classify the corresponding object labels. The prime objective of our algorithm is to use the best aspect ratio values for selecting the default boxes so that we can improve the SSD algorithm for detecting objects [14].

Through the integration of deep learning architectures and advanced techniques, this technology has revolutionized industries ranging from autonomous driving to healthcare and surveillance. By enabling automated identification and localization of objects within images and video frames, object detection has not only enhanced efficiency and safety but also paved the way for new applications, reshaping the landscape of modern computer vision and ushering in an era of heightened visual understanding. Object detection through machine learning has revolutionized computer vision, reshaping how we analyze and interact with visual information. Through the fusion of deep learning techniques and innovative algorithms, this technology has transcended traditional boundaries, enabling automated and accurate identification of objects within images and video streams. Its widespread applications, from enhancing safety in autonomous systems to revolutionizing industries like healthcare and surveillance, underscore its transformative impact. As machine learning continues to evolve, object detection stands as a shining example of its potential to reshape our world, heralding a future where machines possess an unprecedented ability to decipher and interpret the visual intricacies of our environment [15].

The integration of Artificial Intelligence to tackle computer vision challenges has surpassed traditional image processing methods. The CNN model trained to on-road vehicle dataset for single object detection, achieved a validation accuracy of 95.7 % for auto, 95.5% for car, and 96 % for heavy vehicles for day images. The high validation accuracy is because of the huge amount of data on which it is trained from each class. Performance metrics are tabulated for day, evening, and NIR images. Multiple object detection is implemented using YOLOv3 for KITTI and COCO datasets. Performance metrics are tabulated for YOLOv3 on considered classes of images. The higher the precession value of the class greater be mAP value. The mAP value depends on the image chosen for calculation [16].

This research introduces a deep learning-based item recognition system aimed at detecting objects within images. The study employs an enhanced SSD (Single Shot Multibox Detector) approach alongside a multilayer convolutional neural network to recognize objects with speed and precision. Our technology handles both static and dynamic (video) images effectively. More than 80% of the predictions made by the proposed model are correct. Once feature data is extracted from the image, convolutional neural networks use feature mapping to assign class labels. The major objective of our solution is to improve SSD's object detection process by selecting default boxes with the best feasible aspect ratios [17].

Although deep learning-based detection algorithms are now widely used across various domains, some issues remain, such as: 1) Reducing reliance on vast datasets. 2) To achieve efficient detection of small objects. 3) Realization of multi-category object detection [18].

The first phase in the implementation of autonomous vehicles and bots is the detection of objects. In this work, the role of several algorithms for object detection is decoded. The paper also discourses the different deep learning frameworks and services available for object detection. Appropriately detecting an object in video surveillance is an essential aspect of computer vision research. Processing the image obtained from a surveillance camera is difficult due to low image resolution, changing light conditions, moving items in the background, and minor changes in the background, such as trees, etc [19].

This paper compares several object detection algorithms, including multi-stage detectors like R-CNN, and Faster R-CNN, and single-stage detectors such as Yolo (You Only Look Once) and SSD (Single Shot Detector). These models can be deployed on mobile devices to perform object detection in a single pass. When it comes to accuracy, recall, precision, and loss, multi-stage detection methods generally outperform single-stage detectors like Yolo and SSD. Faster R-CNN, Yolo, and SSD algorithms are compared using a Custom Chess Piece Dataset, training them in Google Collaborator Notebooks. These algorithms are intended to correctly identify chess pieces and draw a bounding box near them using TensorFlow Object Detection API and Robofow. A confidence interval is used to assess the algorithm's certainty in predicting object locations. The training and evaluation metrics are visualized with Tensorboard, an interactive tool provided by TensorFlow [20].

Data ownership is a significant issue, as it is often unclear who holds the rights to collected data. Additionally, legal frameworks must be respected, and patient consent is essential to avoid legal complications. Data quality also varies, depending on the source, and anonymizing data, necessary for patient privacy, makes it difficult to use the same data for disease prediction for individual patients later on. Legal, ethical, and data ownership challenges must be resolved before wide-scale implementation [21].

Target detection algorithm invention and optimization play a critical role in the swift advancement of autonomous driving technologies. This technology still faces the following problems, despite the fact that current techniques have significantly improved in terms of speed and accuracy of detection, like: (1) Multi-domain target detection and diversified datasets. (2) Small object detection. (3) High-precision lightweight network architecture. (4) Video detection. (5) Multimodal detection. (6) Insufficient oversight and limited detection [22].

METHODOLOGY

The experimental method is used for the realization of this work. This enables us to examine and analyze the performance of the trained models in a real environment, also analyzing the results related to the detection of objects in images or videos. We have used secondary data, where we have reviewed scientific literature obtained from books and papers by various authors in this field, scientific studies and publications, and literature published electronically. We have used reliable sources and sites for scientific literature searches.

The main objective of this paper is the development in Python programming language of a personalized application for image detection using AI and ML techniques, where a model based on the YOLO algorithm will be created and trained to detect objects within images from a custom dataset. Performance analysis of different trained models and object detection in a real environment will also be performed. To train the object detection model, a suitable data set in the form of images will be collected and processed. Image data will be carefully processed to ensure compliance with the YOLO model, including tasks such as image resizing, format conversion, and annotation formatting. The YOLO algorithm will be used for the object detection model. The model will be trained on the database prepared in advance using the architecture and parameters in YOLO. The trained model will be extensively evaluated using appropriate metrics, such as precision, recall, F1 score, and average precision. Based on the evaluation results, the model can be further improved by adjusting the architecture, parameters, or training dataset. The final step involves integrating the trained object detection model into a custom Python application. This app will allow users to upload images or videos and get detected objects, along with their bounding boxes and object names. The application will provide a simple interface for the user to interact with the object detection functionality. When training a customized model in YOLO for object detection, there are several variables and factors that we think affect the model's performance. Some of these variables and factors that we will consider in training the models are:

1. The dataset we will train:
   - The amount of data and more images usually lead to better performance.
   - Data quality and images should be clear and representative.
   - Diversity, variation in scenarios, lighting, and perspectives.
2. Defining images or labeling them:
   - Labeling accuracy, bounding boxes, and classes must be correct.
   - Consistency, using the same criteria for all images.

1.1. CNNs and the YOLO Algorithm.

Convolutional Neural Networks (CNNs) and the YOLO (You Only Look Once) algorithm are foundational to modern object detection. CNNs excel in feature extraction and classification, while YOLO integrates detection and localization into a single model, enabling real-time object detection.

1.2. Convolutional Neural Networks (CNNs)

CNNs process input images through multiple layers, including convolutional, pooling, and fully connected layers, to extract hierarchical features.

1. Convolutional Layer: Applies kernels (K) to extract spatial features:

$$Y[i,j] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X[i+m, j+n] \cdot K[m,n] \tag{1}$$

   where X is the input image, K is the kernel, and Y is the feature map.
2. Pooling Layer: Reduces spatial dimensions to minimize computation and overfitting, commonly using max pooling:

$$Y[i,j] = \max X[i+m, j+n] \tag{2}$$

3. Fully Connected Layer: Maps the extracted features to class probabilities using a softmax activation function:

$$P(y=k\,|x) = \frac{\exp(W_k^T x + b_k)}{\sum_{j=1}^{C}\exp(W_j^T x + b_j)} \tag{3}$$

   where W_k and b_k are weights and biases for class k, and C is the total number of classes.

1.3. YOLO Algorithm

YOLO divides the input image into an S×S grid and predicts bounding boxes and class probabilities for each grid cell.

1. Bounding Box Prediction: YOLO predicts B bounding boxes per cell, defined by:

$$\text{Box} = (x, y, w, h, c) \tag{4}$$

   where x and y represent the box center, w, and h are dimensions, and c is the confidence score.
2. Loss Function: YOLO's loss function comprises localization, confidence, and classification terms:

$$L = \lambda_{coord}\sum_{i=0}^{S}\sum_{j=0}^{B} 1ij^{obj}[(x-x')^2 + (y-y')^2] + \lambda_{conf}\sum_{i=0}^{S}\sum_{j=0}^{B} 1_{ij}^{obj}(c-c')^2 + \sum i \tag{5}$$

where λ_{coord} and λ_{conf} are weighting factors, and 1_{ij}^{obj} indicates whether a grid cell contains an object.

3. Non-Maximum Suppression (NMS): YOLO applies NMS to eliminate overlapping boxes, selecting the one with the highest confidence score.

What are the parameters of the model?

We will train a model specifically designed to detect car objects. This approach reduces complexity by limiting the number of classes to just one, labeled as "car."

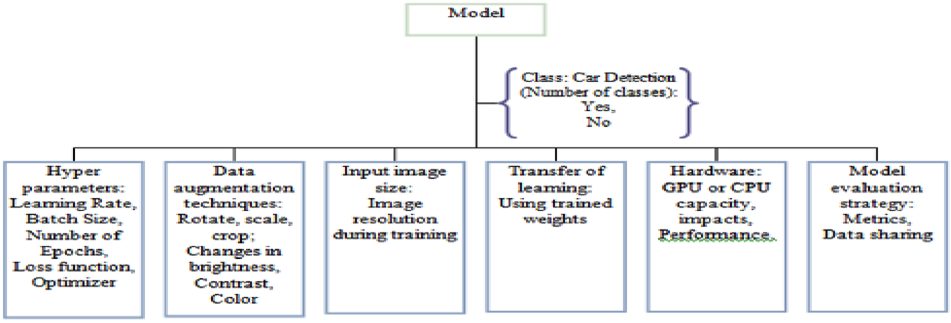


Figure 1. Variables and Classes of the customized model.

The diagram in Fig.1, shows the variables that affect the performance of the trained model like Learning rate; Batch size, is the number of samples (images in the case of YOLO) that are simultaneously processed by the model during a training iteration; Number of epochs: the number of a complete training cycle where the model has seen all samples in the training dataset. Epochs allow the model to repeatedly learn from the same dataset; Loss function; Optimizer etc. Also, we can mention Data augmentation techniques like Rotate, scale, crop; Changes in brightness, contrast, color, etc. Input image size like Image resolution during training; an important issue about the performance is the hardware used for training, so the capacity of the GPU or CPU used for training, impacts model quality and performance. Higher specifications allow for better parameter handling, resulting in a more accurate and efficient model. Additionally, the training time will be shorter.

These factors interact with each other and their optimization requires constant experimentation and adaptation. In this paper, the models will be trained and generated based on the change of the following parameters, taking into consideration the processing capacities of the hardware used for training:

- Number of epochs: The datasets will be trained with a number of 10, 30, and 50 epochs.
- The size of the dataset to be trained: a dataset of 30 and 80 images of objects of the "car" class.

The F1 Score is a metric used to evaluate the performance of classification models and is particularly useful when the data classes are imbalanced. It combines Precision and Recall into a single value by taking their harmonic mean. Precision defines how many of the model's positive predictions are actually correct. Recall defines how many of the positive cases are correctly classified by the model.

Significant computational efficiency gains, reducing training time while improving accuracy and comprehensive quantitative comparisons of classification performance (accuracy, precision, recall, F1-score) across various model combinations, pushing the limits of image recognition [23].

Performance Metrics

Precision (P) : $P = \frac{TP}{TP + FP}$
(6)

Recall (R) : $R = \frac{TP}{TP + FN}$
(7)

F1 Score : $F1 = 2 \cdot \frac{P \cdot R}{P + R}$
(8)

Mean Average Precision (mAP): Area under the precision-recall curve.

The next step is interpretation of the F1 Score:

I. A high F1 score (close to 1) indicates that the model has high precision and recall, the model makes very few errors.

II. A low F1 score (close to 0) indicates that the model has either low precision, low recall, or both.

F1 is particularly useful when one of these metrics (precision or recall) is not enough to provide a fair evaluation of the model, as the model might have high precision but low recall, or vice versa. For both types of models, we present and analyze the F1 metric.

Steps followed to implement this procedure

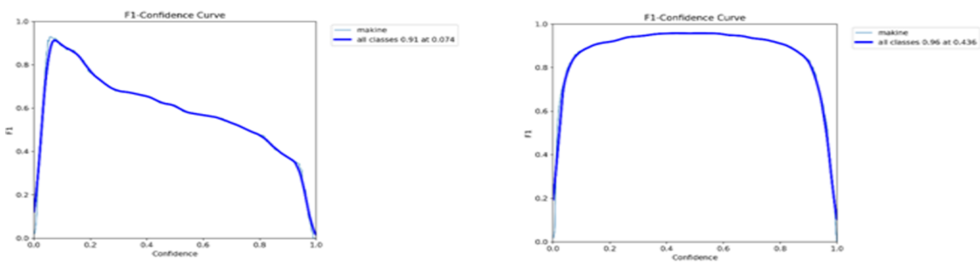
The first step consists in creating the dataset and collecting as many images as possible. The images for training the models are provided by Google sources, such as "Open Images Dataset v7", which offers a wide range of images depending on the types that are required, while marking the images and framing them for training. Next, we deal with Image Labeling, where "CVAT - Open Data Annotation Platform" is used to label objects in images. According to the YOLO algorithm, the label is stored in a '.txt' file where each line represents an object and contains the class and coordinates of the bounding box. The third step deals with the formation of the dataset, organized in two directions: Folder structure, where the images and tagged files are organized in separate folders for training and testing, and Data Formatting, where we make sure that the data is in the format right for the YOLO model. The fourth step is the training and evaluation of the dataset. Here we use the script created for training and validating the dataset to create our own model. We have to preselect the number of epochs with which the model will be trained. 'Weights' is the folder of weights of the trained model, which will then be used for object detection. The last step refers to the testing and validation of the model. After training, we test the model on a test dataset to evaluate its performance. After running the script for validation, the following data is generated with relevant statistics and metrics for the trained model.

Below you can see a part of the 'detect object' procedure, written in Python

```
# The Function of object detection in the images (format JPG) or video (format MP4)
def detekto_objektet():
  weight_file=filedialog.askopenfilename(filetypes=[("YOLOv8 Weights", "*.pt")],
title=" Select the best model weight (best.pt)")
  if not weight_file:
    messagebox.showwarning("File not selected", "Please choose the model weights file.")
    return
  model = YOLO(weight_file)
  if input_type == 'yes':
    resultatet = model(file_path)
    save_path=os.path.join(save_folder, f"{name}_detektim{ext}")
    cv2.imwrite(save_path, resultatet[0].plot())
```

If necessary, we change the parameters and repeat the training to improve the performance of the model. To improve the model, we increase the number of image data in the dataset and also increase the number of epochs on which the dataset is trained. So, we have created several trained models on which we will analyze the performance of each one. Then we can determine which model is more effective and detects the images more accurately. The same procedure was also written for processing and saving the result for the video.

RESULTS



F1 Metrics – Dataset with 30 images trained in 50 epochs

F1 Metrics – Dataset with 80 images trained in 50 epochs

Figure 2. F1 - Metrics.

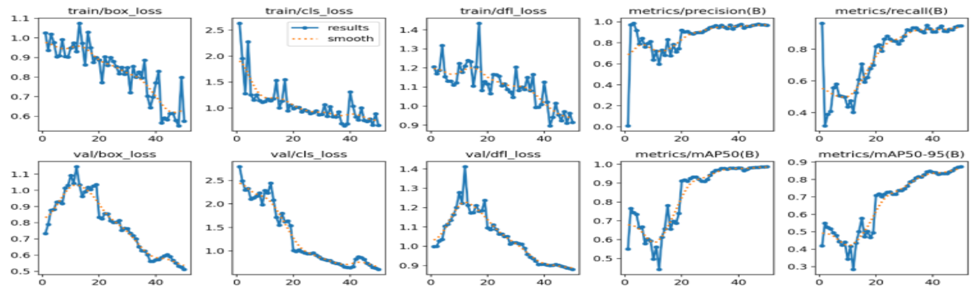


Figure 3. Metrics of Loss Function, Precision, Recalls, and mAP50 parameter.
The model with 30 images was trained in 50 epochs.

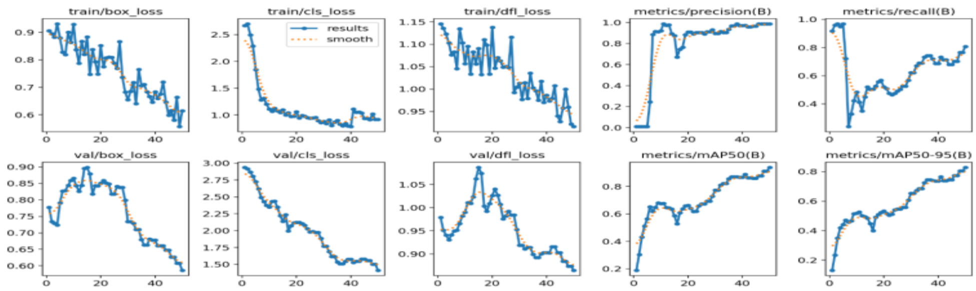


Figure 4. Metrics of Loss Function, Precision, Recalls, and mAP50 parameter.
The model with 80 images was trained in 50 epochs.

From the comparison of the F1-Score curves for the two trained models, based on the Fig. 2, Fig.3, and Fig.4, we note these performance differences:

I. The first trained Model shown in Fig. 3, is based on 30 Images - 50 Epochs, and the maximum F1-Score is 0.91 at a confidence level of 0.074. The F1-Score initially rises to a peak and then starts to decline. This indicates that the model is able to achieve good performance at a very low confidence level, but further increases in confidence worsen its performance. Loss decreases during training, showing learning progress. The F1-Score curves for the “car” class are relatively close at the start, but the model appears to have a gradual decline in performance at higher confidence values.

II. The second trained Model shown in Fig. 4, is based on 80 Images - 50 Epochs, where the maximum F1-Score is 0.96 at a confidence level of 0.436. Here, this curve has a higher and more consistent F1-Score. The maximum F1-Score is reached at a significantly higher confidence value (0.436), and the F1-Score remains relatively high even at further increases in confidence. The loss function shows a more consistent decrease. The class curves are very close to each other, and the F1-Score remains more stable, suggesting that the second model has a more consistent performance in accurate classification even at higher confidence levels.

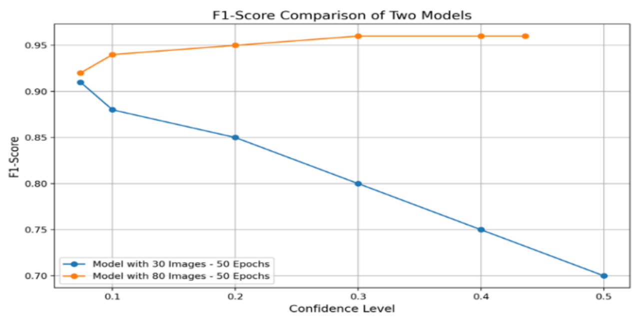


Figure 5. Compares two models trained with different dataset sizes.

The graph in Fig.5, compares two models trained with different dataset sizes:

- The blue line (30 images model) starts at F1-Score 0.91 but drops significantly as confidence increases
- The orange line (80 images model) performs better, reaching and maintaining an F1-Score of 0.96.

This comparison clearly demonstrates that the model trained with 80 images shows better and more stable performance compared to the one trained with 30 images.

CONCLUSION

The study highlights the effectiveness of AI and ML in enhancing object detection accuracy and reliability. Results reveal a strong correlation between dataset size and model performance, with larger and more diverse datasets yielding significantly better outcomes. Specifically, the model trained on 80 images and 50 epochs demonstrated superior performance, achieving higher F1-scores of 0.96 across broader confidence levels compared to models trained on smaller datasets. The study underscores the importance of advanced architectures like convolutional neural networks (CNNs) in improving both accuracy and processing speed.

The research also demonstrates that models trained with optimized parameters and architectures, such as YOLO, achieve competitive results. The findings emphasize the transformative potential of AI and ML in real-time applications by balancing accuracy and speed.

However, the study also identifies several challenges. Extended training times and the limited size of the current dataset hinder the model's ability to generalize effectively to unseen data. Insufficient computational resources, such as the lack of high-performance CPUs and GPUs, further constrain training efficiency and model quality. Addressing these limitations is crucial for improving the scalability and applicability of object detection systems.

Overall, this research provides valuable insights into the advancements and limitations of object detection using AI and ML. The findings pave the way for future innovations and optimizations, enabling broader adoption of these technologies in critical areas such as public safety, infrastructure monitoring, and healthcare.

FUTURE WORK

Future research will focus in:

- Dataset Expansion: Increasing the diversity and size of training datasets can improve model robustness and generalization, particularly in real-world applications where conditions vary significantly.
- Cloud platforms: Use of cloud platform for efficient training and focusing on localized applications in Albania.
- Real-time Processing: Investigating methods for optimizing models to achieve real-time processing capabilities without sacrificing accuracy will be crucial for applications in autonomous systems and surveillance.
- Integration with Other Technologies: Combining object detection with other AI technologies, such as natural language processing, reinforcement learning, robotics, and drones could lead to more intelligent, adaptable systems and drive technological innovation in the country (Albania).

These advancements aim to enhance performance, scalability, and practical adoption of AI-driven object detection systems across various sectors.

FUNDING

The authors have received financial support from the University "Aleksander Moisiu", Durres, Albania.

COMPETING INTERESTS

The authors confirm that there are no competing interests related to this work.

REFRENCES

[1] International Journal of Information Technology. (2023). Anime face recognition to create awareness. *International Journal of Information Technology, 15*, 3507–3512. https://doi.org/10.1007/s41870-023-01391-8

[2] Bhaidasna, H., & Bhaidasna, Z. (2023). Object detection using machine learning: A comprehensive review. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT, 9*(3), 248–255. https://doi.org/10.32628/CSEIT2390215

[3] Amjoud, A. B., & Amrouch, M. (2023). Object detection using deep learning, CNNs and vision transformers: A review. *IEEE Access, 11*, 35479–35516. https://doi.org/10.1109/ACCESS.2023.3266093

[4] Patkar, U. C., Shrives, S. B., Patil, U. S., Patankar, A. J., Jain, N., Kumari, M., & Chandhoke, A. (2023). Object detection using machine learning and deep learning. *International Journal of Intelligent Systems and Applications in Engineering, 12*(1s), 466–473.

[5] Gooda, S. K., Chinthamu, N., Selvan, S. T., Rajakumareswaran, V., & Paramasivam, G. B. (2023). Automatic detection of road cracks using EfficientNet with residual U-Net-based segmentation and YOLOv5-based detection. *International Journal on Recent and Innovation Trends in Computing and Communication, 11*(4s), 84–91. https://doi.org/10.17762/ijritcc.v11i4s.6310

[6] Patel, V., Shah, M., Modi, S., Patel, Y., & Thakkar, S. (2022). Object detection using machine learning: A survey. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*. https://doi.org/10.2139/ssrn.4042549

[7] Tumpa, S. A., & Fahim, M. (2024). Object detection using machine learning and neural networks.

[8] Dakshinamoorthy, P., Rajaram, G., Garg, S., Murugan, P., Manimaran, A., & Sundar, R. (2024). Artificial intelligence algorithms for object detection and recognition in video and images. https://doi.org/10.21203/rs.3.rs-3849848/v1

[9] Amjoud, A. B., & Amrouch, M. (2023). Object detection using deep learning, CNNs and vision transformers: A review. *International Journal of Computer Vision, 35*(6), 456–467.

[10] Bakiasi, V., & Muça, M. (2024). Improving facial expression classification through ensemble deep learning models. *Nanotechnology Perceptions*, 37–47. https://doi.org/10.62441/nano-ntp.vi.387

[11] Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. https://doi.org/10.48550/arXiv.1905.05055

[12] Arkin, E., Yadikar, N., Xu, X., Aysa, A., & Ubul, K. (2022). A survey: Object detection methods from CNN to transformer. *Multimedia Tools and Applications, 82*. https://doi.org/10.1007/s11042-022-13801-3

[13] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2021.3084827

[14] Kumar, A., Zhang, J., & Lyu, H. (2020). Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP Journal on Wireless Communications and Networking*. https://doi.org/10.1186/s13638-020-01826-x

[15] Rahman, M., Chakma, S., Raza, D., Akter, S., & Sattar, A. (2021). Real-time object detection using machine learning. In *Proceedings of ICCCNT*. https://doi.org/10.1109/ICCCNT51525.2021.9580170

[16] M., & Ravish Aradhya, H. V. (2019). Object detection and tracking using deep learning and artificial intelligence for video surveillance applications. *International Journal of Advanced Computer Science and Applications (IJACSA, 10*(12). https://doi.org/10.14569/IJACSA.2019.0101269

[17] Vaishnavi, K., Reddy, G., Reddy, T., Iyengar, N., & Shaik, S. (2023). Real-time object detection using deep learning. *Journal of Advanced Mathematics and Computational Science, 38*, 24–32. https://doi.org/10.9734/jamcs/2023/v38i81787

[18] Deng, J. (2020). A review of research on object detection based on deep learning. *Journal of Physics: Conference Series, 1684*, 012028.

[19] International Journal of Latest Engineering and Management Research (IJLEMR). (2023). A study on object detection using deep learning. *International Journal of Latest Engineering and Management Research (IJLEMR, 8*(6), 22–26. [Online]. Available: www.ijlemr.com

[20] Yadav, S. P., Jindal, M., Rani, P., et al. (2024). An improved deep learning-based optimal object detection system from images. *Multimedia Tools and Applications, 83*, 30045–30072. https://doi.org/10.1007/s11042-023-16736-5

[21] Bushati Hoxha, S., Bushi, F., & Gjecka, A. (2024). Application & challenges of big data technologies in Albanian healthcare system. *Interdisciplinary Journal of Research and Development, 11*(3), 120. https://doi.org/10.56345/ijrdv11n316

[22] Ma, B., Zheng, C., & Zheng, Z. (2021). Advanced object detection algorithms and its application based on deep learning. In *Proceedings of the International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China* (pp. 408–415). https://doi.org/10.1109/MLBDBI54094.2021.00083

[23] Shtino, V. B., Muça, M., & Bushati, S. (2024). Machine learning integration for precise facial micro-expression recognition. *Journal of Computer Science, 20*(11), 1545–1558. https://doi.org/10.3844/jcssp.2024.1545.1558