AI-Based Fake Media Detection Using Machine Learning and Natural Language Processing

CHOWTIPALLI BABY KRISHNA USHARANI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

V.Sarala

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Object detection is a fundamental task in computer vision that involves identifying and localizing objects within an image. With the advancement of deep learning, models like YOLO (You Only Look Once) have significantly improved the speed and accuracy of object detection systems. This project presents a YOLOv8-based object detection system integrated into a Django web framework, enabling users to upload images and receive real-time detection results.The system provides a complete user management module, including user registration, login, and role-based access control for users and administrators. Authenticated users can upload images through a web interface, which are then processed using the YOLOv8 model from the Ultralytics library. The model detects multiple objects within the image and generates bounding boxes along with confidence scores.

Once the image is processed, the system stores both the original and annotated images. It also records detection details such as detected object names, object counts, confidence scores, and a descriptive summary. The results are displayed to the user along with visual annotations, enhancing interpretability.The system maintains a history of all detection activities, allowing users to revisit previous results. Additionally, an admin dashboard provides insights into user activity, including total users, detection counts, and individual user statistics.The YOLOv8 model is chosen for its efficiency and real-time detection capability. It processes images in a single pass, making it significantly faster than traditional region-based methods. The integration with Django ensures scalability, security, and ease of deployment.This project demonstrates the practical application of deep learning in building intelligent web-based systems. It can be extended to real-time video processing, surveillance systems, and industrial automation. The combination of deep learning and web technologies makes the system highly effective for modern computer vision applications.

Keywords: Object Detection, YOLOv8, Deep Learning, Computer Vision, Django Web Application, Image Processing, Artificial Intelligence, Bounding Box Detection, Smart Surveillance, Neural Networks

I. INTRODUCTION

In recent years, computer vision has emerged as one of the most impactful fields in artificial intelligence, with applications in security, healthcare, autonomous vehicles, and smart cities. Object detection, a key task in computer vision, involves identifying objects within an image and determining their locations using bounding boxes.Traditional object detection methods relied on manual feature extraction and sliding window techniques, which were computationally expensive and inefficient. With the introduction of deep learning, particularly convolutional neural networks (CNNs), object detection has seen significant improvements in both speed and accuracy.YOLO (You Only Look Once) is one of the most popular object detection algorithms due to its real-time performance. Unlike region-based approaches such as R-CNN, YOLO processes the entire image in a single forward pass, making it highly efficient. The latest version, YOLOv8, provides improved accuracy, faster inference, and better generalization.

This project aims to develop a web-based object detection system using YOLOv8 integrated with the Django framework. The system allows users to upload images and receive detection results instantly. It also includes user authentication features to ensure secure access and personalized experience.The backend of the system handles image processing, model inference, and data storage. When a user uploads an image, the YOLO model processes it and returns detected objects along with confidence scores. The results are stored in a database and displayed through a user-friendly interface.An important feature of the system is the detection history, which enables users to track their previous uploads and results. The admin dashboard provides an overview of system usage, helping administrators monitor activity and manage users effectively.The proposed system has wide-ranging applications, including surveillance, traffic monitoring, retail analytics, and industrial inspection. It demonstrates how deep learning models can be deployed in real-world applications using web technologies.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Object detection has evolved significantly over the years, transitioning from traditional computer vision techniques to deep learning-based approaches. Early methods such as Haar cascades and Histogram of Oriented Gradients (HOG) relied on handcrafted features and were limited in accuracy and scalability.Region-based Convolutional Neural Networks (R-CNN) marked a major breakthrough by introducing deep learning into object detection. R-CNN and its variants, Fast R-CNN and Faster R-CNN, improved detection accuracy by generating region proposals and classifying them using CNNs. However, these methods were computationally expensive and not suitable for real-time applications.Single-shot detectors such as SSD (Single Shot MultiBox Detector) and YOLO addressed these limitations by performing detection in a single pass. YOLO divides the image into grids and predicts bounding boxes and class probabilities simultaneously. This significantly improves speed while maintaining high accuracy.YOLO has undergone several improvements, from YOLOv1 to YOLOv8. The latest version, YOLOv8, introduces architectural enhancements, improved loss functions, and better training techniques. It achieves state-of-the-art performance in object detection

tasks and is widely used in real-world applications.Recent research has focused on improving detection accuracy in challenging conditions such as low-light environments, occlusions, and small object detection. Techniques such as attention mechanisms, transformer-based models, and hybrid architectures have been proposed.In addition, web-based deployment of deep learning models has gained popularity. Frameworks like Django and Flask are commonly used to build user-friendly interfaces for AI applications. Integration with cloud services and APIs further enhances scalability and accessibility.

The proposed system leverages YOLOv8 for object detection and Django for web development. It combines the strengths of deep learning and web technologies to create an efficient and scalable solution for real-world applications.

III. EXISTING SYSTEM

Existing object detection systems primarily rely on traditional machine learning or earlier deep learning models. Many of these systems are either offline or lack user-friendly interfaces, limiting their practical usability.Traditional systems use feature-based approaches such as edge detection, color histograms, and texture analysis. These methods are not robust and fail in complex environments with varying lighting conditions and object orientations.Earlier deep learning models like R-CNN and Fast R-CNN improved detection accuracy but were computationally intensive and slow. They required multiple stages, including region proposal generation and classification, making them unsuitable for real-time applications.Some modern systems use SSD or earlier versions of YOLO, which provide better speed but may not achieve the same level of accuracy as newer models like YOLOv8. Additionally, many existing systems do not include features such as user authentication, history tracking, and administrative control.

Another limitation is the lack of integration with web platforms. Many systems operate as standalone applications, making them difficult to access and use. They also lack scalability and cannot handle multiple users efficiently.Furthermore, existing systems often do not provide detailed insights such as confidence scores, object counts, or descriptive summaries. Visualization and result interpretation are also limited.The proposed system addresses these limitations by using YOLOv8 for high-speed and accurate detection, along with a Django-based web interface for accessibility, scalability, and user management.

IV. PROPOSED METHOD

The proposed system is a web-based object detection platform that leverages the power of the YOLOv8 deep learning model integrated with the Django framework. The system is designed to provide accurate, fast, and user-friendly object detection capabilities through an interactive web interface.The system allows users to register, log in, and upload images for object detection. Once an image is uploaded, the YOLOv8 model processes it and identifies objects present in the image. The model generates bounding boxes, class labels, and confidence scores for each detected object. YOLOv8 is selected due to its real-time performance and high accuracy, achieved through its advanced

architecture and anchor-free detection mechanism .The system also stores detection results, including original images, processed images, detected object names, and confidence scores. A descriptive summary is generated automatically to improve interpretability. Users can view their previous detection history, making the system more interactive and useful.An admin module is included to monitor system usage, manage users, and analyze detection statistics. The admin dashboard displays insights such as total users, total detections, and user-specific activity.

The system is designed to be scalable and can be extended to support real-time video detection, IoT integration, and cloud deployment. It addresses the limitations of traditional systems by providing a complete end-to-end solution combining deep learning and web technologies.

V. IMPLEMENTATION

The implementation of the system is carried out using Python, Django, and the YOLOv8 model from the Ultralytics library. The system is divided into multiple modules, including authentication, image processing, detection, and data management.The frontend is developed using HTML, CSS, and Django templates, providing an intuitive interface for users to interact with the system. Users can sign up, log in, upload images, and view detection results.

The backend is implemented using Django views and models. User authentication is handled using Django's built-in authentication system. Additional user information, such as phone number, is stored using a UserProfile model.When a user uploads an image, it is stored in the media directory, and a record is created in the DetectionHistory model. The system then loads the YOLOv8 model and processes the uploaded image using OpenCV.

The YOLOv8 model performs object detection in a single forward pass, making it highly efficient for real-time applications. It detects multiple objects and returns bounding boxes along with confidence scores. Research shows that YOLOv8 achieves high detection accuracy even with relatively small datasets .The detected objects and their confidence scores are extracted and stored in JSON format. An annotated image is generated with bounding boxes and saved for visualization. A descriptive summary is created based on detected objects and their counts.

The system also handles errors gracefully. If the YOLO library is not installed, a warning message is displayed. Any runtime errors during detection are captured and stored in the database.The prediction results are displayed on a separate page, showing the annotated image, detected objects, confidence scores, and description. Users can also access their detection history, which is stored in the database.The admin dashboard provides system-level insights, including total users and detection statistics. It allows administrators to monitor activity and manage users effectively.

VI. ALGORITHMS

The system primarily uses the YOLOv8 algorithm for object detection, supported by deep learning techniques.

YOLOv8 Algorithm

YOLOv8 (You Only Look Once version 8) is a state-of-the-art object detection algorithm that performs detection in a single pass through the neural network. Unlike traditional multi-stage detectors, YOLOv8 divides the image into grids and predicts bounding boxes and class probabilities simultaneously.

YOLOv8 uses an anchor-free approach, which simplifies the detection process and improves performance. It incorporates advanced architectural components such as feature pyramid networks and improved backbone structures to enhance feature extraction .

The algorithm outputs:

- Bounding box coordinates
- Object class labels
- Confidence scores

Convolutional Neural Networks (CNN)

YOLOv8 is based on CNN architecture, which is effective for image processing tasks. CNN layers extract spatial features from images, enabling accurate object recognition.

Non-Maximum Suppression (NMS)

After detection, multiple overlapping bounding boxes may be generated. NMS is applied to remove redundant boxes and retain only the most relevant detections.

Recent research indicates that YOLOv8 provides high precision and robustness in various real-world applications, including traffic monitoring and surveillance

VII. SYSTEM DESIGN

The system follows a multi-layered architecture consisting of presentation, application, and data layers.

1. Presentation Layer

This layer includes the user interface built using Django templates. It allows users to:

- Register and log in
- Upload images

- View detection results
- Access detection history

The interface is designed to be user-friendly and responsive.

2. Application Layer

The application layer is the core of the system, implemented using Django views. It handles:

- User authentication
- Image upload and storage
- YOLO model execution
- Result processing and formatting

Key components include:

- **Authentication Module**: Handles login, signup, and logout
- **Detection Module**: Processes images using YOLOv8
- **History Module**: Stores and retrieves detection records
- **Admin Module**: Provides system analytics

3. Data Layer

The data layer includes:

- **Database (SQLite/MySQL)** for storing users and detection history
- **Media Storage** for images
- **JSON Data** for storing detection results

Workflow

1. User registers/logs in
2. Uploads an image
3. Image is saved in the server
4. YOLOv8 processes the image
5. Objects are detected with confidence scores
6. Results are stored in the database
7. Output is displayed to the user

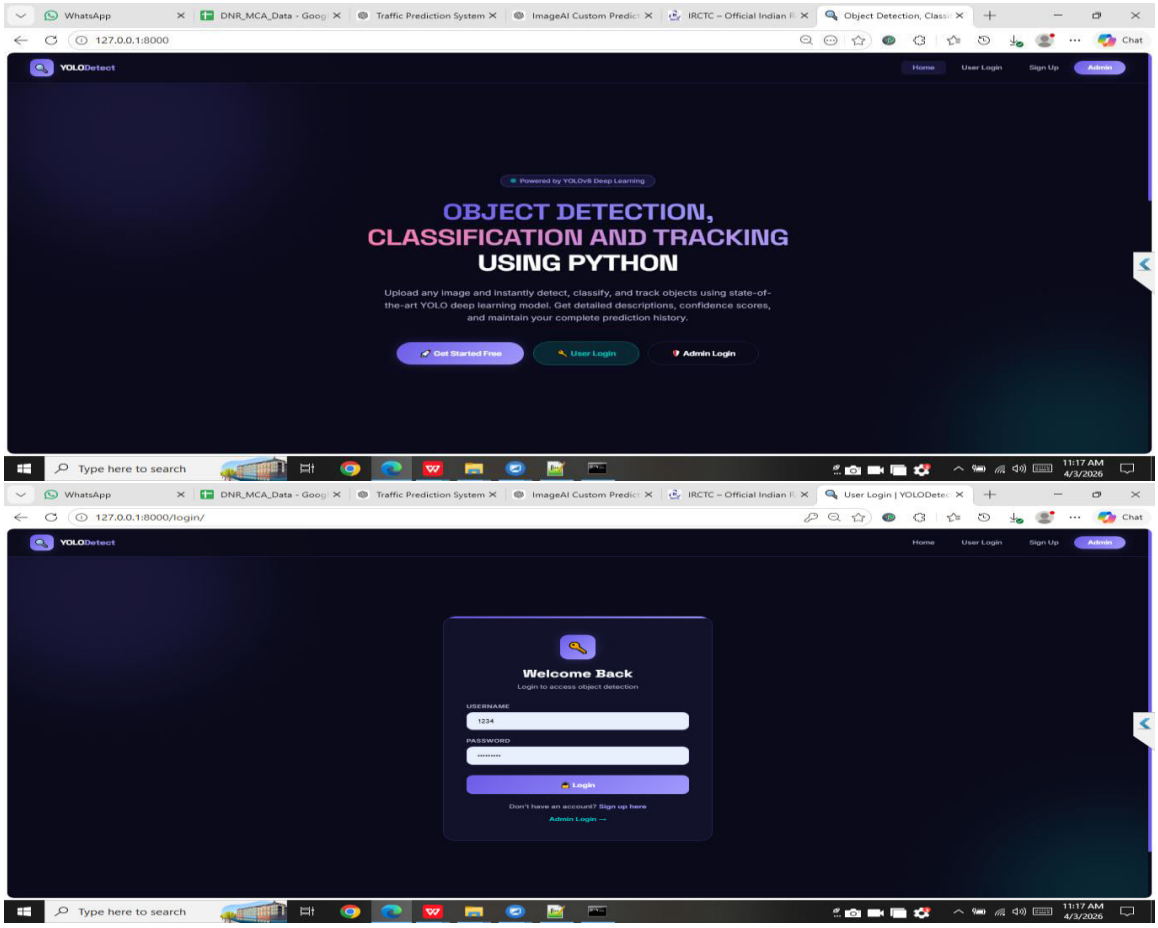
Design Features

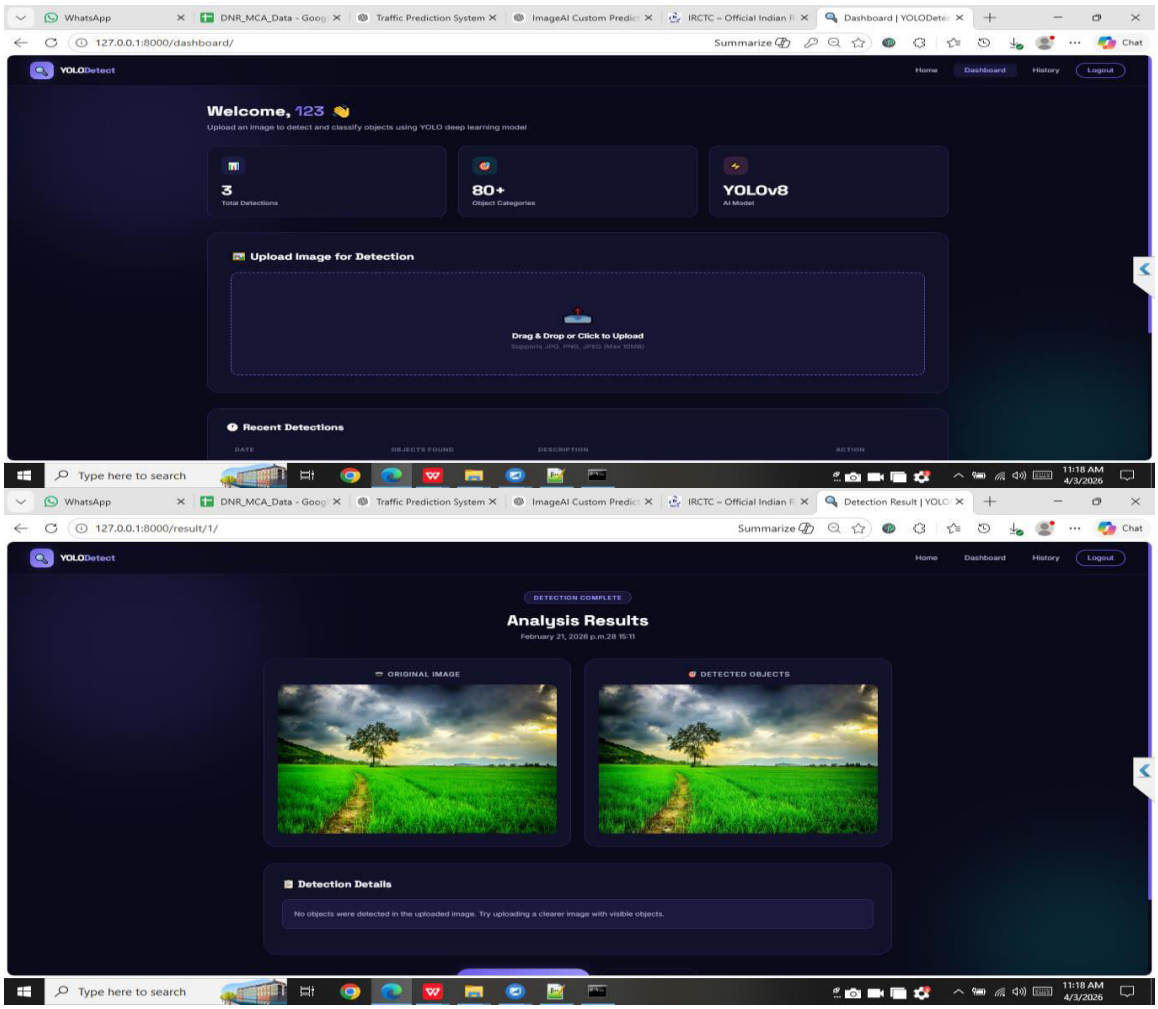
- Modular architecture
- Scalable design

- Secure authentication
- Efficient data storage

Modern systems enhance YOLOv8 with additional modules to improve detection in complex scenarios such as small or occluded objects .

SYSTEM DESIGN IMAGES





VIII. CONCLUSION

This project presents a YOLOv8-based object detection system integrated with the Django web framework. The system successfully combines deep learning and web technologies to provide an efficient and user-friendly solution for object detection.The use of YOLOv8 ensures high accuracy and real-time performance, making the system suitable for practical applications such as surveillance, traffic monitoring, and industrial automation. The integration with Django enables scalability, security, and ease of use.

The system provides additional features such as user authentication, detection history, and admin dashboard, enhancing its usability and functionality. The ability to store and retrieve detection results allows users to analyze past data effectively.One of the key strengths of the system is its modular design, which makes it easy to extend and improve. Future enhancements may include real-time video detection, integration with IoT devices, and deployment on cloud platforms.Despite its advantages, the system may face challenges such as dependency on model availability and computational resources. However, these limitations can be addressed through optimization and hardware acceleration.

Overall, the project demonstrates the effectiveness of YOLOv8 in solving real-world object detection problems and highlights the potential of combining artificial intelligence with web technologies.

REFERENCES

1. · YOLOv8 Behavioral Research Paper
2. · SRE-YOLOv8 UAV Detection Model (2024)
3. · Recurrent YOLOv8 Framework (2025)
4. · Varghese & Sambath, "YOLOv8: Enhanced Performance and Robustness," 2024
5. · Efficient YOLOv8 Small Object Detection (2024)
6. · YOLOv8 Rice Disease Detection (2025)
7. · Z-YOLOv8 Traffic Detection Model (2024)
8. · YOLOv8 Architecture Study
9. · YOLOv8 Conference Paper (ADICS 2024)
10. · SOD-YOLOv8 Small Object Detection
11. · YOLOv8 RF Signal Detection Study
12. · YOLOv8 Forensic Detection Study
13. · Frontiers in Neuroscience YOLOv8 Study, 2025
14. · MDPI Sensors YOLOv8 Study, 2024
15. · Scientific Reports YOLOv8 Enhancement, 2025