YOLO11-Flood Victim Detection and Rescue Alert System

Shribhakti S Vibhuti[1], Shruti Sutar[2], Bhoomika Marigoudar[3], Aishwarya Gopal[4], Sneha Varur[5], and Channabasappa Muttal[6]

School of Computer Science and Engineering,
KLE Technological University, Hubballi, India
01fe22bci048@kletech.ac.in,01fe22bci052@kletech.ac.in, 01fe22bci035@kletech.ac.in, 01fe22bci033@kletech.ac.in, sneha.varur@kletech.ac.in,channabasappa.muttal@kletech.ac.in

Abstract. Accurate detection of humans and animals is critical for enhancing the efficiency of flood rescue operations, enabling a quicker response and improving disaster management efforts. This study presents a model that identifies and counts humans and animals in flood-affected regions, while sending immediate email alerts to rescue teams for prompt action. The alert system enhances communication by providing real-time updates to rescue teams, enabling swift action. This not only boosts operational efficiency but also facilitates the optimal deployment of resources, enabling critical areas to be addressed effectively. The system in this study uses YOLO11 the most recent version in the You Only Look Once (YOLO) series of deep learning models. It is trained on a diverse dataset featuring humans and a variety of animal species, including dogs, cows, horses and goats. The model's performance is evaluated using key metrics such as precision, recall, F1 score and mean Average Precision (mAP). The model achieved a precision of 92%, demonstrating its suitability for real-time flood rescue.
Keywords–YOLO11, Alert, Email, Rescue, Floods, Disaster Management, Detection, Deep Learning.

1 Introduction

India's geographical landscape makes it a flood-prone country [1].In recent years, the occurrence of floods has significantly increased due to factors like changing weather patterns, characterized by rising temperature and irregular rainfall, rapid urbanization, inadequate drainage systems and unsustainable agricultural practices [2]. These recurrent floods have caused devastating impacts, including loss of human lives, economic instability and widespread damage to infrastructure and public utilities [3].

Recently, India has experienced a significant number of natural disasters due to climate change. In 2024, 109 people died in the Assam floods in June [4], followed by 49 people in the Gujarat floods in August [5] and 45 people in the NTR district of Andhra Pradesh in September [6].These are just the documented deaths of people, but many innocent animals also lost their lives [7]. Knowing that floods are inevitable natural catastrophes, this research focuses on technological inputs to reduce their impact by increasing the ease of rescue in good time.

Fig. 1. Aerial view of a flood-affected region in Assam [2024].[8]

The Fig.1 shows the submerged homes, roads and vegetation, emphasizing the severity of the disaster's impact.

The goal of this research is to develop a real-time system that combines quick victim identification with automated alert mechanisms for rescue operations. The proposed solution not only aims to improve the efficiency of rescue efforts but also paves the way for applying similar technologies to other natural disasters, such as earthquakes and hurricanes. By integrating technology, this approach has the potential to reduce the disaster impact. In this study, we designed and implemented a real-time system for identifying individuals and animals trapped in floods and sending alerts to rescue teams using the YOLO11 model [9], implemented with the PyTorch [10] and Python libraries for email notifications [11], [12]. In view of this, the paper focuses on the YOLO11 model for object detection during floods. Hence the study is titled 'YOLO11- Flood victim Detection and Rescue Alert System'.

Technological advancements in machine learning, deep learning[13] and computer vision offers promising opportunities for the precise detection of humans and animals in flooded areas. YOLO, a deep learning algorithm, provides efficient detection and classification of victims, even under challenging conditions [14]. In addition to progress in machine learning, improvements in communication systems ensure fast and effective dissemination of alerts to rescue teams.

The key objectives of this study are to explore the consequences of floods in various regions of India and assess their impact on communities. Additionally, the study aims to apprehend the role of AI technology in detecting people and animals trapped during floods, highlighting the potential of AI-driven solutions for disaster management. The study also concentrates on training an object detection model using a labeled dataset and evaluating its performance metrics across different classes of live entities. Furthermore, the research delves into integrating an email alert system, examining its functionality and effectiveness in real-time flood rescue scenarios.

This study is structured into various sections, starting with Section II, which provides insights into related literature and previous work. Section III focuses on the architecture of the YOLO11 model, explaining its design and key components. In Section IV, the implementation details and the algorithm used are discussed, providing a clear understanding of how the model was developed and integrated. Section V presents the results of the model, showcasing its performance with various metrics and visual graphs to illustrate its effectiveness. Additionally, this section includes a comparison between the performances of YOLOv8 [15] and YOLO11, emphasizing the selection of YOLO11 over other versions of YOLO and CNN models. Section VI concludes the study, summarizing the key findings, and Section VII looks ahead to future work, focusing on potential improvements that could enhance real-time rescue systems.

2 Literature work

Nehete et al.[16] identify significant issues in disaster management systems, emphasizing the computational complexity and adaptability issues of deep learning models like CNNs, R-CNNs, and GANs in real-time rescue operations, particularly in dynamic environments such as floods. Seyed Danial Jozi [17] highlights the limitations of UAV-based damage assessment systems, which are heavily reliant on favorable weather conditions and lack integration with real-time machine learning models. Haoqian Song et al.[18] stress the need for high-quality datasets and real-time integration to support impactful decision-making during rescue operations. Johan K Runtuk et al.[19] uncover critical gaps in flood disaster relief, including poor coordination and limited government involvement. Pradeep N Fale et al.[20] discuss the challenges of combining various technologies into a unified system for Android-based flood rescue applications. These systems encounter deployment issues, especially in fluctuating flood conditions, where quick changes and the requirement for real-time data processing reduce their efficiency. These studies emphasize the need to improve system adaptability, integrate real-time feedback, and leverage technologies like IoT, UAVs, and blockchain to enhance disaster management.

Earlier studies in flood detection have mainly concentrated on locating submerged buildings post disaster. However, these systems generally lacked the ability to perform real-time rescues, as no immediate actions were taken during the disaster itself. In contrast, our approach focuses on real-time detection of both humans and animals during floods. The model sends instant email alerts to rescue teams, using YOLO11 in combination with the SMTP[21] library which is inspired from the work of A. Amankossova and C. Turan[11].Their research demonstrates the utility of real-time alert-notification systems for monitoring compliance in the financial sector, focusing on leveraging automation to swiftly resolve critical data challenges. This integration enables timely and coordinated rescue efforts during active flood events.

3 Background Study

YOLO11, the latest in the YOLO series, combines advancements from earlier versions with new features for improved speed and precision. Known for real-time performance, it excels in applications like disaster response, autonomous systems and surveillance.

Architecture of YOLO11

YOLO architecture at its core is divided into three components, feature extractor, intermediate processing stage and the prediction mechanism. These help the YOLO model to perform various computer vision tasks efficiently[22]. The architecture of YOLO11 is represented in Fig. 2.

Feature Extractor: It can also be called as Backbone of the YOLO model in which it utilizes the conventional neural networks to transform the raw data and generate feature maps. YOLO11 model is structurally similar as its previous versions which mean's it uses conventional neural network layers and generate the feature maps of the images. The feature that separate's this with previous versions of YOLO is that the large single layered Convolution C2f (Convolution blocks used for splitting of feature map) is replaced with two convolution of smaller size block C3k2 extension of Cross Stage Partial (CSP) Bottleneck. Due to smaller kernel size ('k2' stands for small kernel) and two convolutions helps in fast processing. The addition of Cross Stage Partial with Spatial Attention (C2PSA) after Spatial Pyramid Pooling-Fast (SPPF)[23] enhances the accuracy of the YOLO11 model.

Intermediate processing stage : It acts as neck of YOLO model it enhances the feature representation by employing special layer. Here model combines the features to transmit them for next layer and captures multiscale information. The addition of C2PSA module in YOLO11 helps the model to concentrate more on key features of the image which enhances the accuracy of the smaller and congested objects.

The prediction mechanism: This layer takes the input of previous stage and makes the final prediction by putting the bounding boxes and class labels of the objects present in the image. Hence, referred as Head. Usage of multiple C3k2[24] blocks in YOLO11 models helps it to refine and processing the feature maps more efficiently. As these blocks helps in fast processing, smaller Kernel size and adaptability boosts the detection accuracy compare to other model. YOLO11 also uses CBS blocks which assure the transfer of feature maps for next layers efficiently.

Final Convolutional Layers and Detect Layer : The final detection layers refine the feature maps and produce the ultimate outputs. These include bounding box coordinates, which localize the objects percent in the image by defining spatial boundaries, Confidence score to estimate the likelihood of the object present in it along with it the class score is generated that identifies the class of the object. The Detect Layer consolidates these outputs, to ensure precise and dependable object detection results.

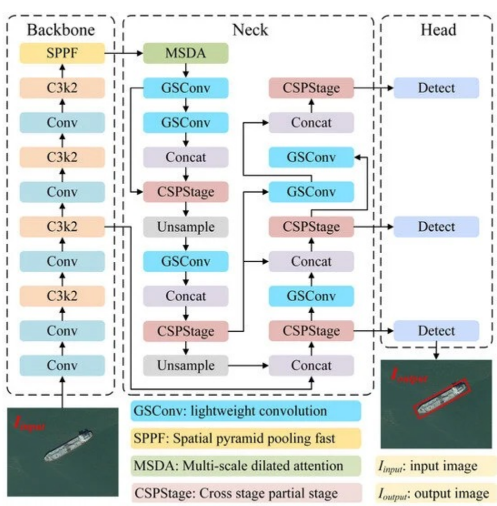

Fig. 2. YOLO11 architecture and its multi-task capabilities. [25]

4 Proposed Methodology

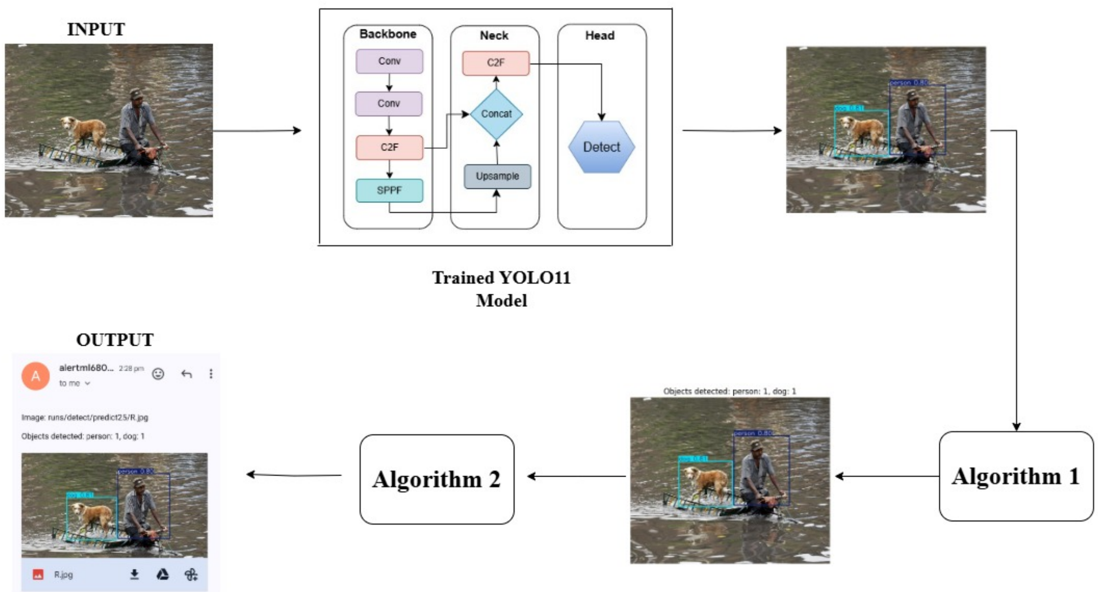

Fig. 3. AI-Powered Flood Rescue System Pipeline

The flood rescue system is designed to provide accurate object detection, efficient feature extraction and real-time alert generation. At the core of the system lies a robust, high-quality dataset sourced from Roboflow, which comprises 500 images. These images represent various classes of objects that are critical for flood rescue operations, including humans, dogs, cows, horses and goats. Roboflow is a powerful tool for creating, managing and deploying computer vision models. To ensure that the images are suitable for training, pre-processing steps are performed to maintain consistent dimensions and normalize pixel values, thus optimizing the model's performance and ensuring accurate object detection in diverse scenarios.

4.1 Model Training

The pre-processed dataset is utilized to train the YOLO11 object detection model. This model is designed to detect multiple objects within a single image, predict their respective bounding boxes and assign accurate class labels. YOLO11's speed and precision make it an ideal choice for real time applications, specially in dynamic and unpredictable environments such as flood zones. The training process is conducted using the PyTorch frame work, ensuring a flexible and efficient setup. During this phase, various hyperparameters, batch size and number of epochs, are tuned carefully to achieve the perfect balance between precision and recall, minimize the false positives (incorrectly identifying objects) and false negatives (failing to detect objects).

4.2 Alert Message Generation

Once the model has been successfully trained, it is tested on real-time images to generate annotated outputs. The system fetches these images, where the algorithm performs object detection by extracting relevant detection results, such as bounding boxes and class IDs. For each processed image, the algorithm counts the number of objects detected per class and generates an alert message that includes the object counts. The entire process, including object detection and alert generation, is outlined in Algorithm 1.

Algorithm 1 Alert Message Generation

Input: Annotated image as `img`
Output: Alert message of annotated image, sent via email
Initialisation:
1: Read the annotated image as `img`.
2: Extract detection results (bounding boxes and class IDs) from annotations of `img`.
Process:
3: **if** objects are detected in `img` then
4: **for** each detected object in `img` do
5: Count the objects by class.
6: **end for**
7: Generate alert message with class names and counts.
8: **else**
9: Generate alert message as "No objects detected."
10: **end if**
11: Send an email with the generated alert message and `img` attached.

The count of objects for each class c is calculated by the equation 1 and 2

$$n_c = \sum_{d \in D} \delta(c_d, c) \tag{1}$$

where:

$$\delta(c_d, c) = \begin{cases} 1 & \text{if } c_d = c, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In which,

- C: The set of unique class IDs in the detection results.
- n_c: The count of objects for each class $c \in C$.
- D: The set of detected objects, where each object d has a class ID c_d.

1. Input: A list of detected objects D in image `img` with their corresponding class IDs c_d.
2. Calculation: For each class c, the formula iterates over the detected objects and counts how many times the class ID c appears.
3. Indicator Function (δ):
   - If the object's class ID c_d matches the target class c, the function contributes 1 to the count n_c.
   - Otherwise, it contributes 0.

If no objects are detected, it generates a message stating "No objects detected".Then the generated message is sent to rescue team via email along with the annotated image,this is achieved by Alert System.

4.3 Alert System

The generated alert message,along with the annotated image is mailed to rescue team, so that they can analyze the severity of situation and plan the rescue, to ensure the timely response for critical detections.The algorithm of Alert system is shown in Algorithm 2.

5 Result and Analysis

The flood rescue system powered by YOLO11 is promising tool for disaster management in real time, with the overall precision of 92% and recall 53%, mean Average Precision (mAP) of 0.751 for IoU=0.5. width=

The overall mAP@50(mean Average Precision at IoU 0.5) is 0.751 tells us that the model performance is good when IoU is set at 0.5 and mAP@50-95 is 0.44 which implies model struggles for higher IoU.

Algorithm 2 Email Alert System

Input: Subject, body text (Generated Alert message), rescue team email, image path
Output: Email sent with Generated Alert message and image

1. Create a multipart email message.
2. Attach the generated alert message (body text) to the email.
3. Open the image file located at `image_path`.
4. Attach the image as a MIME object to the email.
5. Set up the SMTP server for sending the email (e.g., Gmail).
6. Log in to the SMTP server using the sender's credentials.
7. Send the email with the detection summary and image attachment.
8. Close the SMTP connection.

Table 1. Performance Metrics for Object Detection

Class	P	R	mAP@50	mAP@50-95
All	0.922	0.538	0.751	0.448
Cow	1.000	0.160	0.580	0.386
Dog	0.928	0.802	0.884	0.565
Goat	1.000	0.600	0.800	0.427
Horse	0.750	0.500	0.686	0.448
Person	0.933	0.628	0.805	0.413

From Fig. 4, Fig. 5, and Table 1, it is evident that a confidence threshold of 0.5 strikes a good stability between precision and recall across different classes. The precision remains consistently high, ensuring accurate detections. This threshold optimizes the trade off, providing reliable and actionable predictions for real-time rescue operations.

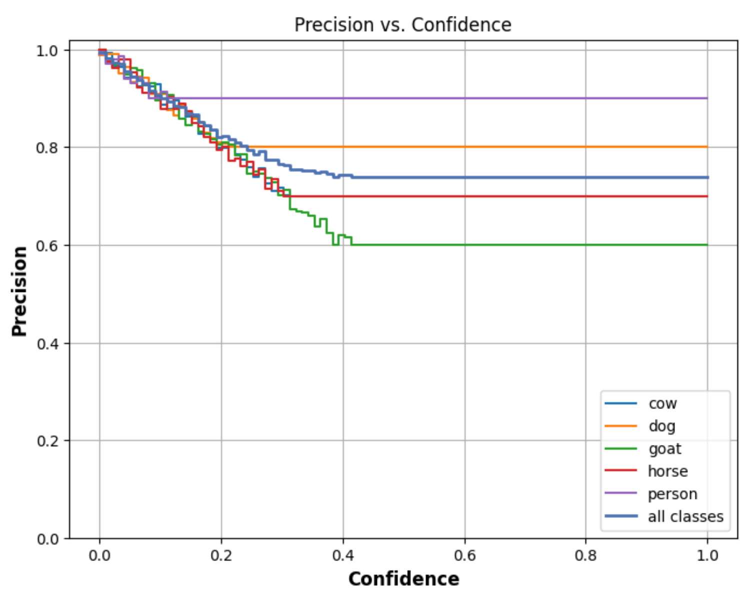

Fig. 4. Precision-Confidence Curve

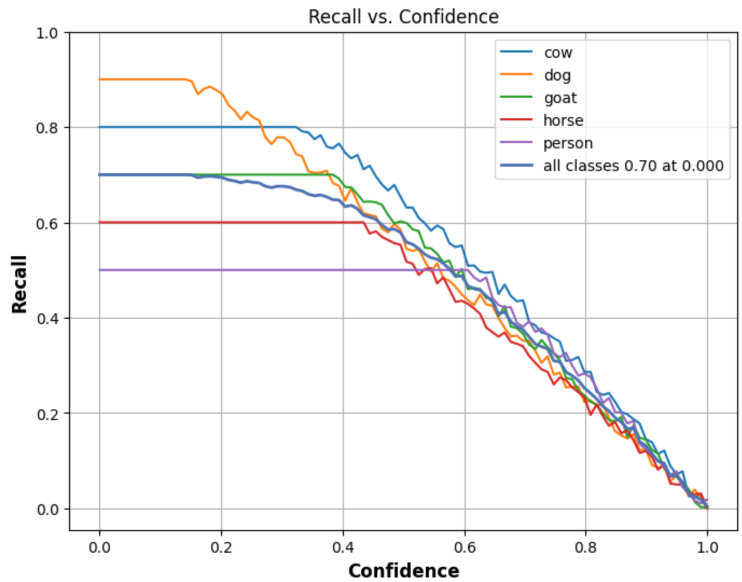

Fig. 5. Recall-Confidence Curve

Comaprision of YOLO11 and YOLOv8

In Fig. 6, YOLO11 detected 9 persons and 1 dog with complete bounding box annotations, whereas YOLOv8 identified only 7 persons and 1 dog, with the dog's bounding box missing. In Fig.7, YOLO11 demonstrates superior detection accuracy by correctly identifying 4 cows and 4 persons, while YOLOv8 only detected 2 cows and 1 person. Additionally, YOLOv8 misclassified one cow as a dog. This highlights YOLO11's robustness in handling partially occluded objects and smaller size objects, which are common in disaster scenarios and its enhanced ability to distinguish objects in cluttered or complex scenes, likely due to its improved architecture, such as smaller kernel sizes and better feature aggregation. Hence YOLO11 is more suitable for real-world and disaster scenarios.

The email alert sent to the rescue team is shown in Fig. 8.Alerts provided include the count of people, animals and images, which help the rescue team analyze the situation and plan the rescue so that they arrive at the site well prepared with all proactive measures.

Fig. 6. Comparison of results: (A) YOLO11 results, (B) YOLOv8 results.

Fig. 7. Comparison of results: (A) YOLO11 results, (B) YOLOv8 results.

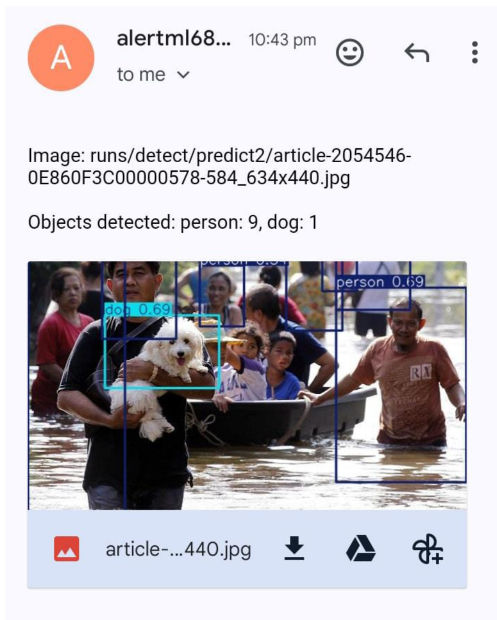

Fig. 8. The email alert sent to the rescue team

The recall(53%) of the model is lower than its precision (92%), which reflects a deliberate choice to prioritize reliability over coverage. This method reduces false positive, guaranteeing that the identified objects, essential for rescue missions, are precise and reliable. However, this trade-off could lead to missed detections in more complex scenarios. To address this, future efforts will focus on improving recall by expanding the dataset to include more diverse and challenging flood scenarios, and by using data augmentation techniques such as random cropping and brightness adjustments. Additionally, enhancements to the model's architecture, such as advanced attention mechanisms and multiscale feature detection, will be explored to better identify smaller and partially hidden objects. These enhancements aim to strike a higher stability among precision and recall, making sure more effective performance in real world scenario.

Overall, the results meet the goal of providing an efficient tool for real-time flood rescue operations. However, to maximize its practical utility in real-world scenarios, it is crucial to address its limitations through future enhancements.

6 Conclusion

The YOLO11 based flood rescue and alert system marks a significant step forward in utilizing deep learning technology to address challenges in disaster response. With its rapid processing capabilities and ability to operate effectively in diverse and challenging conditions, YOLO11 offers a practical solution for real-world disaster management applications. Its ability to detect and classify objects in real-time has demonstrated significant value in a variety of dynamic and unforeseen situations, allowing for the quick identification and counting of individuals and animals.The addition of an email alert feature strengthens the system by ensuring seamless communication with rescue teams, leading to improved planning and resource distribution. However, limitations such as occasional misclassification in low quality images highlights area for future enhancements. It sets the groundwork for further developments, such as improving accuracy, expanding the system's capabilities to cover other disaster types and enhancing its operational scope for greater impact in real-world applications.

7 Future Work

Future research will focus on expanding the dataset to include a wider range of disaster scenarios and object types, enhancing the model's ability to generalize effectively. Integration of multi-model data inputs, such as thermal or LiDAR imaging, could further improve detection accuracy under challenging conditions. Additionally, adding alternative communication methods like SMS or app-based notifications, along with adding location details to the alerts, can make the system more efficient for real-world deployments. While the current model prioritizes precision to ensure reliable alerts so, dataset of 500 annotated images was sufficient, future work will aim to improve recall by expanding the dataset and exploiting advanced feature extraction techniques. Regular testing in real-world disaster scenarios will provide valuable insights into the system's effectiveness, scalability and ability to balance recall and precision over time.

References

[1] S. C. Pal, I. Chowdhuri, B. Das, *et al.*, "Threats of climate change and land use patterns enhance the susceptibility of future floods in india," *Journal of Environmental Management*, vol. 305, p. 114 317, 2022.

[2] C. Mohan, V. Sireesha, M. Brijesh, and P. Kapoor, *Karnataka's climate crisis: Urban challenges and sustainable solutions for bangalore*, 2024.

[3] D. Singh, *Causes, impacts, risk and mitigation of urban flood management in india*, 2022.

[4] T. Hindu. "Assam flood situation improving." Accessed: 2024-12-06. (2024), [Online]. Available: `https://www.thehindu.com/news/national/assam/assam-flood-situation-improving/article68405617.ece`.

[5] T. I. Express. "Heavy rain claimed 49 lives in august last week, 37,000 people rescued across state." Accessed: 2024-12-06. (2024), [Online]. Available: `https://indianexpress.com/article/cities/ahmedabad/heavy-rain-claimed-49-lives-in-august-last-week-37000-people-rescued-across-state-9550772/`.

[6] T. N. Minute, "Death toll in andhra pradesh floods mounts to 45," 2024, Accessed: 2024-12-05. [Online]. Available: `https://www.thenewsminute.com/andhra-pradesh/death-toll-in-andhra-pradesh-floods-mounts-to-45`.

[7] A. D. P. Vieira and R. Anthony, "Reimagining human responsibility towards animals for disaster management in the anthropocene," *Animals in our midst: The challenges of co-existing with animals in the Anthropocene*, pp. 223–254, 2021.

[8] H. Ali, P. Modi, and V. Mishra, "Increased flood risk in indian sub-continent under the warming climate," *Weather and Climate Extremes*, vol. 25, p. 100 212, 2019.

[9] M. A. R. Alif, "Yolov11 for vehicle detection: Advancements, performance, and applications in intelligent transportation systems," *arXiv preprint arXiv:2410.22898*, 2024.

[10] V. R. Mishra, "Implementing various residual networks for cow stall object detection using pytorch,"

[11] A. A. Student and C. Turan, "Implementation of a real-time alert-notification system for data monitoring in the financial industry,"

[12] P. Atri, "Automating big query dependency management with email alerts: The big query dependency email trigger library," *European Journal of Advances in Engineering and Technology*, vol. 11, no. 1, pp. 52–55, 2024.

[13] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022.

[14] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical analysis of design aspects of various yolo-based deep learning models for object detection," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 126, 2023.

[15] P. Madnur, P. Shetty, G. Parashetti, S. Varur, and M. S. M, "Advancing in cricket analytics: Novel approaches for pitch and ball detection employing opencv and yolov8," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, 2024, pp. 1–8. DOI: 10.1109/I2CT61223.2024.10544224.

[16] P. U. Nehete, D. S. Dharrao, P. Pise, and A. Bongale, "Object detection and classification in human rescue operations: Deep learning strategies for flooded environments.," *International Journal of Safety & Security Engineering*, vol. 14, no. 2, 2024.

[17] S. D. Jozi, "Uav-based post-disaster damage assessment of buildings using image processing," 2024.

[18] H. Song, W. Song, L. Cheng, Y. Wei, and J. Cui, "Pdd: Post-disaster dataset for human detection and performance evaluation," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[19] J. K. Runtuk, A. L. Maukar, E. W. Puspitarini, *et al.*, "Flood disaster relief operation: A systematic literature review," *Jurnal Sistem Teknik Industri*, vol. 24, no. 2, pp. 203–220, 2022.

[20] P. N. Fale, P. Gajbhiye, T. Sondule, A. Barsagade, P. Jambhulkar, and N. Rahangdale, "A flood rescue system based on android application," *Available at SSRN 3850067*, 2021.

[21] M. K. SIRISHA, R. NANDINI, N. ASWINI, N. LAKSHMI, K. SHILPA, *et al.*, "Iot based theft detection using smtp protocol," *International Journal of Mechanical Engineering Research and Technology*, vol. 16, no. 2, pp. 229–237, 2024.

[22] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, "The effect of choosing optimizer algorithms to improve computer vision tasks: A comparative study," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16 591–16 633, 2023.

[23] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors," *arXiv preprint arXiv:2411.00201*, 2024.

[24] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[25] J. Huang, K. Wang, Y. Hou, and J. Wang, "Lw-yolo11: A lightweight arbitrary-oriented ship detection method based on improved yolo11," *Sensors*, vol. 25, no. 1, p. 65, 2024.