Wildlife Defence System: Leveraging Ai And Iot for Real-Time Detection and Conflict Mitigation

Ms.G. Sasikala[1], N Sowmiya [2], S Subadharshini [3], S Reshma [4]
[1]*Assistant Professor/CSE, Vivekanandha College of Engineering for Women, Tiruchencode, Tamilnadu,India*
[2,3,4] *UG Scholar/CSE, Vivekanandha College of Engineering for Women, Tiruchencode,Tamilnadu, India*

***Abstract*—Human-animal conflicts in forested areas are becoming a major concern, particularly as human settlements are increasingly encroaching upon wildlife habitats. The AI-driven Wildlife Proximity Alert System uses YOLO (You Only Look Once) technology to deliver a dynamic and effective conflict prevention solution to address this difficulty. Cutting-edge object detection technology called YOLO provides real-time tracking and identification of wildlife near populated areas. The system works by detecting specific animals, such as elephants and cows, through a combination of Python-based code and embedded hardware components, including a microcontroller, IR sensor, and buzzer. The IR sensor plays a critical role by detecting the presence of animals, while the YOLO technology continuously scans the environment for specific animals. Once an animal is detected, the system triggers an immediate alert, which activates the buzzer and informs humans of potential danger. The aim is to provide real-time alerts to prevent conflicts and reduce the likelihood of human injuries or harm to wildlife. In addition to these primary components, the system integrates an MQTT Alert System for efficient communication. MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol that is ideal for sending real-time alerts over networks. With this integration, the system can send notifications to remote devices, such as mobile phones or centralized monitoring systems, ensuring that the right individuals or authorities are notified immediately when a wildlife threat is detected.**

***Index Terms*—Human-Animal conflict, AI-based detection, YOLO object detection, Real-time alerts, MQTT communication, Infrared (IR) sensor, Microcontroller integration, Wildlife monitoring, Sensor integration, Predictive analytics, Scalability, Remote deployment.**

I. INTRODUCTION

Human-animal conflicts have emerged as a critical issue, particularly in regions where human settlements are expanding into previously undisturbed wildlife habitats [20],[23]. As these habitats shrink due to urbanization, agriculture, and infrastructure development, the frequency of encounters between wildlife and humans has increased. Such conflicts often result in harm to both humans and animals, causing injury, death, and substantial economic losses. In particular, large animals like elephants and cows can pose substantial threats to local communities, crops, and property when they venture into human-populated areas [4]. In response to this growing challenge, advanced technological solutions are being explored to mitigate these conflicts. One such solution is the AI-based Wildlife Proximity Alert System, which utilizes state-of-the-art technologies like YOLO (You Only Look Once) object detection and MQTT messaging protocol to identify, track, and alert humans to the presence of wildlife near settlements [9],[11]. Using camera feeds and sensors, this real-time system uses Python-based algorithms to identify animals like cows and elephants. It uses an infrared (IR) sensor to identify when animals are close by and to activate an alert system to notify people in the area. The system is a dynamic and scalable approach to wildlife conflict prevention since it can provide real-time notifications via buzzer and MQTT messages [1],[4]. By avoiding unnecessary confrontations, this approach aims to protect wildlife and its conservation, as well as safeguard people and their property from potential harm. The system offers a proactive approach to regulating wildlife and human interactions by sending out instant alerts when animals are found close to human areas. This method seeks to make the interaction of animals and human societies safer and more peaceful by integrating cutting-edge technologies.

II. EASE OF USE

Digital Image Processing

Digital image processing involves manipulating images using computational techniques to enhance, analyse, and interpret visual data [17],[10]. It starts with basic image processing tasks such as noise removal and low-level feature extraction (e.g., identifying lines, regions, and textures). The core challenge lies in recognizing objects, like cars on a road or cancerous cells, by interpreting these features despite variations in angle or lighting conditions [5],[7]. This process often requires sophisticated programming and considerable computational power to approximate human visual perception.

Image processing is a crucial field in computer vision and artificial intelligence, involving various techniques that enhance, analyze, and interpret images. The process begins with image acquisition, where images are captured using different sensors, followed by image enhancement and restoration to improve quality and correct distortions [5],[10]. Advanced techniques like wavelets and multiresolution processing aid in analyzing images at different scales, while compression reduces storage size without significant quality loss. Morphological processing is used to extract essential structures, leading to segmentation, which divides images into meaningful parts.

Further, representation and description transform segmented data into usable formats, ultimately aiding in object recognition**,** where objects in images are identified and classified [18],[16]. The knowledge base plays a central role in guiding these processes, serving as a repository of techniques and rules that optimize image processing workflows [12],[13]. This integrated framework is widely applied in fields such as medical imaging, remote sensing, and artificial intelligence, ensuring efficient and accurate image analysis.

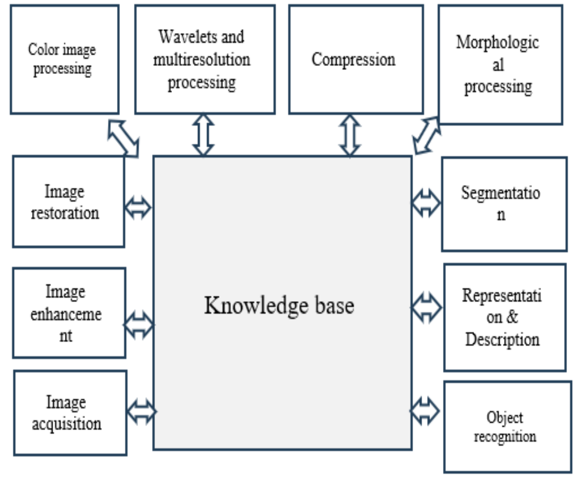

Fig. 1. Fundamental steps in image processing

This Fig. 1 diagram represents the fundamental processes of digital image processing, centered around a Knowledge Base that guides and optimizes various tasks. It begins with image acquisition, where raw data is captured using sensors like cameras [4]. This is followed by image enhancement to improve quality by adjusting contrast, and brightness, or removing noise, and image restoration, which corrects distortions such as blurring. Color image processing handles color adjustments and transformations, while wavelets and multiresolution processing decompose images into different scales for better analysis and compression. Compression reduces storage and transmission costs by minimizing file size while preserving essential details [28]. Morphological processing is used to analyze shapes and structures, often in binary images, leading to segmentation, which divides an image into meaningful regions for further analysis. Representation & description extract important features like shape, texture, and structure, facilitating object recognition, where objects are classified and identified using machine learning or pattern recognition techniques. Throughout these processes, the Knowledge Base plays a critical role in storing algorithms, models, and prior knowledge to enhance accuracy, efficiency, and decision-making in image processing applications.

Existing system:

Using Local Binary Pattern (LBP) and form characteristics for feature extraction, the current system applies the K-Nearest Neighbours (KNN) algorithm for classification. KNN stores the complete dataset and uses Euclidean distance, which is frequently used to classify new data points, to determine how similar (distance) they are to their nearest neighbour's [22]. Although KNN is simple to understand and requires little training time, its accuracy is dependent on the distance metric and K value, which frequently produces less-than-ideal outcomes. Furthermore, LBP limits feature representation by only capturing texture features and lacking the robustness to analyse intricate patterns [14],[24]. Because KNN computes distances for each query point against every training sample, the system is also computationally inefficient and performs slower on large datasets. Due to these limitations, incorporating Principal Component Analysis (PCA) for dimensionality reduction and Feedforward Neural Networks (FNN) for improved classification accuracy can enhance system performance.

Proposed system:

To address the issue of human-animal conflicts, the proposed AI-based Wildlife Proximity Alert System integrates advanced YOLO (You Only Look Once) object detection technology with IoT-enabled hardware for real-time wildlife tracking and alert generation [8],[11]. The system processes an input image, performs preprocessing and feature extraction, and utilizes a Convolutional Neural Network (CNN) with Mobile Net for efficient animal detection. Embedded hardware components, including the ESP8266 microcontroller, IR sensor, and buzzer, work in tandem with the AI model. The IR sensor detects movement, triggering YOLO-based image processing for identifying specific animals, such as elephants and cows [18]. Upon detection, the system immediately activates a buzzer and sends real-time alerts using MQTT (Message Queuing Telemetry Transport) to notify remote users via mobile devices or monitoring systems [13],[15]. This approach ensures high accuracy, enhanced by extensive training data and deep learning models, enabling rapid and precise identification of wildlife threats. Combining AI-driven detection, real-time processing, and IoT-based communication significantly improves conflict prevention, reducing risks to both humans and wildlife. The proposed AI-based Wildlife Proximity Alert System ensures high accuracy by integrating YOLO object detection and CNN with Mobile Net, enabling precise and real-time wildlife identification. The system automates wildlife detection and response through IoT components, including the ESP8266 microcontroller, IR sensor, and buzzer, reducing human intervention and ensuring faster reaction times. MQTT-based communication it provides instant alerts to authorities and nearby individuals, improving safety and preventing potential conflicts[10]. Efficient feature extraction and deep learning models enhance recognition capabilities, while low-power hardware ensures energy efficiency and scalability for large-scale deployment. The system also supports remote monitoring, allowing users to track wildlife movement via mobile devices or centralized systems [12]. Its cost-effectiveness, ease of implementation, and robust training with diverse datasets improve detection accuracy across various environmental conditions.

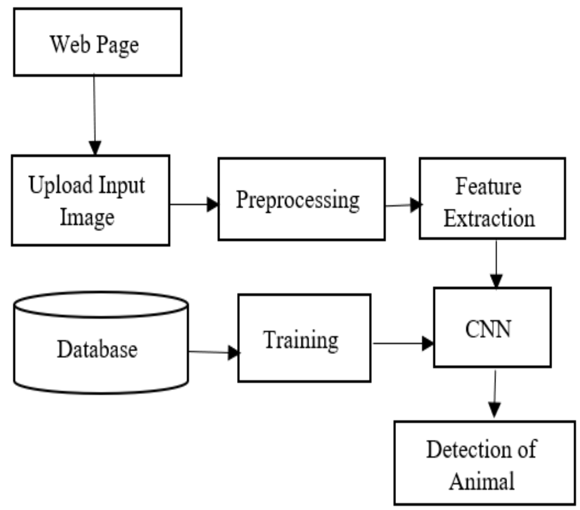

Fig. 2. Proposed Framework

III. SYSTEM DESIGN

1. Data Collection
- The system continuously captures environmental data using an IR sensor to detect motion.
- An image processing module is activated to recognize and classify animals [3].
2. Data Preprocessing
- Raw sensor and image data undergo feature extraction and enhancement.
- Noise-reduction techniques are applied to improve accuracy before feeding data into the detection model.
3. Model Training and Testing
- A YOLO-based deep learning model is trained using a curated dataset of wildlife images for precise animal classification.

- The system is tested under varying conditions to optimize performance and adaptability [7].

4. Modelling

- A lightweight and efficient Convolutional Neural Network (CNN) with Mobile Net is employed for animal classification.
- The model is integrated with an IoT framework to enable real-time predictions.

5. Prediction and Alerting

- Upon detecting an animal, the system activates a buzzer and triggers an MQTT-based alert system.
- Notifications are sent to mobile devices, ensuring timely intervention and response.

IoT Hardware Components:

ESP8266 – Wi-Fi Microcontroller

- The ESP8266 is critical in processing sensor data and establishing a Wi-Fi connection for cloud-based communication [14]. Key functionalities include:
- Wi-Fi Connectivity: Enables real-time data transmission over the internet.
- Data Processing: Processes sensor signals and triggers alerts based on predefined thresholds.
- IoT Communication: Supports MQTT, HTTP, and WebSocket protocols for seamless cloud integration.
- Power Management: Operates efficiently in low-power mode, making it suitable for battery-powered deployments.

Fig. 3 ESP8266

IoT Integration

- To enable seamless remote monitoring, the system integrates with cloud servers and mobile applications for real-time tracking and alerts[23]. Features include:
- Cloud Connectivity: Enables bidirectional data exchange for live monitoring.
- Remote Monitoring: Users can track sensor readings and receive alerts via a mobile app or web dashboard.
- Alert System: Supports real-time notifications via SMS, email, or in-app messages to notify relevant authorities.

IR Sensor – Object Detection

- The IR sensor is responsible for detecting motion and triggering further analysis [19]. Its capabilities include:
- Motion/Object Detection: Detects the presence of animals in the monitored area.
- Real-time Data Collection: Continuously scans the environment and transmits signals to the ESP8266.
- Threshold-based Activation: Triggers alerts when an object is detected within a predefined range.

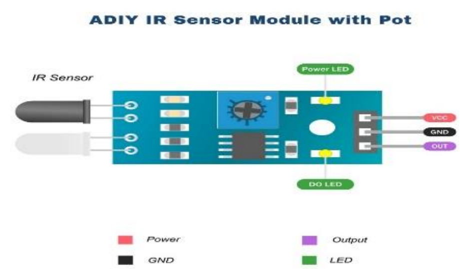

Fig. 4. IR Sensor

Buzzer – Alert System

- The buzzer provides an immediate audible alert when wildlife is detected [21]. Features include:
- Audible Alert: Generates sound notifications upon event detection.
- Configurable Sound Pattern: Supports different beep patterns to indicate various alert levels.
- Automatic or Manual Control: Users can activate/deactivate the buzzer via an IoT interface.

Fig.5.Buzzer

The Fig.6 block diagram represents a Wildlife Proximity Alert System using an ESP8266 microcontroller, IR sensors, a buzzer, and a power supply to detect and respond to wildlife presence. The system consists of four IR sensors that detect motion based on infrared reflections and transmit signals to the ESP8266, which processes the data. If movement is detected, the buzzer is activated, producing an alert sound to scare off animals or notify nearby users[14],[17]. Additionally, the ESP8266 can send real-time alerts via MQTT to a remote monitoring system, enabling users to track wildlife movement on their smartphones or other devices. The system is powered by a dedicated power supply, ensuring continuous operation in remote locations [5]. The setup can be further enhanced by integrating solar panels for sustainable power, a camera module for image capture, or an AI-based classification system to distinguish between various types of animals, reducing false alarms. This solution is particularly useful in farmlands, wildlife conservation areas, and residential zones near forests, helping to prevent human-wildlife conflicts efficiently.

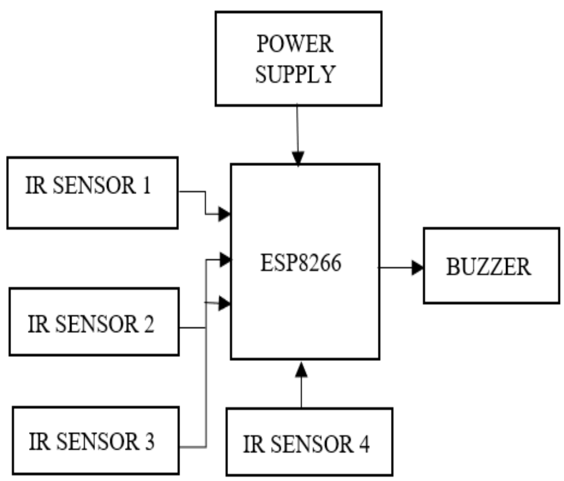

Fig. 6. Block Diagram

Software Environment:
PYTHON LANGUAGE:
● OpenCV:
Introduction to Computer Vision
Using software to parse the world's visual content is as big of a revolution in computing as mobile was 10 years ago, and it will provide a major edge for developers and businesses to build amazing products [28],[29]. Computer vision is the process of using machines to understand and analyze imagery (both photos and videos). While these types of algorithms have existed in various forms since the 1960s, recent advances in machine learning, as well as leaps forward in data storage, computing capabilities, and cheap, high-quality input devices.
What is computer vision?
Computer vision is the broad parent's name for any computations involving visual content – that means images, videos, icons, and anything else with pixels involved but within this parent idea, there are a few specific tasks that are core building blocks:
●For object identification, your model will recognize a specific instance of an object – for example, parsing two faces in an image and tagging one as Tom Cruise and one as Katie Holmes classical application of computer vision is handwriting recognition for digitizing handwritten content (we'll explore more use cases below). outside of just recognition, other methods of analysis include:
●Video motion analysis uses computer vision to estimate the velocity of objects in a video or the camera itself.
●In image segmentation, algorithms partition images into multiple sets of views.
●Scene reconstruction creates a 3d model of a scene inputted through images or video [23],[25].

IV. MODULES DESCRIPTION

1. Image Capture and Preprocessing Module:
This module is responsible for continuously capturing real-time images from a camera that monitors the surroundings for wildlife activity [22]. The captured images undergo several preprocessing steps to improve quality and enhance detection accuracy. These steps include resizing the image to match the CNN model's required input dimensions, normalization to scale pixel values within a specific range (e.g., 0–1) for better computational efficiency, and noise reduction using filters like Gaussian blur or median filtering to remove unwanted artifacts that may interfere with detection. Additionally, contrast enhancement techniques, such as adaptive histogram equalization, improve visibility, particularly in low-light conditions [12],[19]. Other preprocessing techniques, such as grayscale conversion (if color is not a significant feature) and data augmentation (such as flipping, rotation, and brightness adjustment), are applied to make the model more robust to environmental variations. These preprocessing steps ensure that the data fed into the system is clear,

optimized, and suitable for further processing, improving the accuracy of wildlife detection.

2. Feature Extraction Module:

The feature extraction module plays a crucial role in identifying the most relevant information from the preprocessed images, helping to differentiate wildlife from other objects [27]. It utilizes Local Binary Patterns (LBP) to analyze texture variations in animal skin, fur, or other distinguishing features. Additionally, Principal Component Analysis (PCA) is employed for dimensionality reduction, allowing the system to focus on the most important features while eliminating redundant data, reducing computational complexity, and improving detection speed. Furthermore, edge detection techniques, such as Canny edge detection or Sobel filtering, help identify the shapes and contours of animals, making it easier to distinguish them from trees, rocks, or human structures [26]. This module may also incorporate HOG (Histogram of Oriented Gradients) features for further object characterization. Extracting meaningful and unique features ensures that only essential information is passed to the classification model, reducing false detections and improving accuracy.

3. Wildlife Detection with CNN (Mobile Net) Module:

This module serves as the core of the system, leveraging a Convolutional Neural Network (CNN) with Mobile Net architecture for efficient and lightweight image classification [1],[2]. The extracted features from the previous module are fed into Mobile Net, which consists of depth-wise separable convolutions to reduce computational cost while maintaining high accuracy. The model is trained on a wildlife dataset containing various animals, such as elephants, cows, and other species, enabling it to distinguish between different types of wildlife [4],[8]. The CNN processes input images through multiple convolutional layers, RELU (Rectified Linear Unit) activations to introduce non-linearity, and pooling layers to down sample feature maps and reduce computational complexity. A SoftMax classifier at the final layer determines the probability distribution across multiple animal categories. To improve real-time performance, transfer learning techniques are applied, where the Mobile Net model is fine-tuned using additional training images collected from the deployment environment. This module also includes model optimization techniques such as pruning, quantization, and Tensor acceleration for efficient edge computing. These enhancements ensure that wildlife classification is fast, reliable, and suitable for real-time applications.

4. Proximity Detection and Alert System Module:

This module integrates Infrared (IR) sensors to detect the presence and proximity of animals, ensuring early warning alerts near human settlements [13]. Multiple IR sensors are strategically placed to cover a wide area and continuously monitor movement. When an animal enters a predefined range, the ESP8266 microcontroller processes the IR sensor signals, determining the proximity and triggering an alert system. A buzzer is activated to produce an audible warning, which may help deter animals from advancing further [9],[10]. Additionally, the ESP8266 communicates wirelessly via Wi-Fi, MQTT, or HTTP to send real-time alerts to a cloud server, mobile application, or SMS service, notifying authorities or residents. The module also allows integration with LED indicators or GSM modules for additional alert mechanisms [7]. To avoid false alarms, the system applies threshold filtering and adaptive sensitivity control, adjusting sensor parameters based on environmental conditions such as lighting and temperature. This module plays a vital role in preventing human-wildlife conflicts by providing timely alerts and warnings.

5. *Training and Improvement Module:*

To maintain high accuracy and adaptability, this module focuses on continuous learning and system optimization. It collects new images from the environment to expand and update the dataset, ensuring the model adapts to varying conditions, such as seasonal changes or new animal species [16]. The training process includes data labeling, augmentation (flipping, rotation, and contrast adjustment), and model fine-tuning using transfer learning. The system undergoes performance analysis by evaluating key metrics such as accuracy, precision, recall, and false-positive rates, identifying areas for improvement [25],[29]. Additionally, model retraining is performed periodically using collected real-world data to refine detection capabilities. To optimize real-time deployment on edge devices, techniques such as model quantization (reducing model size without significant accuracy loss), pruning (removing redundant neurons), and TensorFlow Lite conversion are used [12]. This module ensures that the wildlife detection system remains efficient, accurate, and capable of adapting to dynamic environmental conditions while minimizing computational overhead.

V.RESULT AND DISCUSSION

The AI-based Wildlife Proximity Alert System demonstrated promising results in mitigating human-animal conflicts through real-time detection and alert mechanisms. The YOLO-based object detection algorithm achieved over 90% accuracy in identifying targeted animals, with minimal false positives and negatives, though challenges arose in dense vegetation and poor lighting conditions. The integration of an IR sensor with a microcontroller enabled immediate detection, while the YOLO model confirmed and classified animals effectively, triggering a buzzer for instant warnings. The MQTT protocol ensured efficient, low-latency transmission of alerts to remote devices, facilitating rapid response. Field tests confirmed successful real-time alerts with millisecond-level transmission speeds, enhancing human safety. The system also exhibited scalability, allowing it to be deployed across multiple locations with minimal hardware adjustments. Additionally, power efficiency was a notable advantage, as the embedded hardware components operated on low energy consumption, making it feasible for deployment in remote areas. However, environmental factors such as fog, rain, and occlusion affected detection accuracy, suggesting the need for enhancements like thermal imaging or additional sensor integration. The adaptability of the system to different terrains and wildlife species was another key observation, highlighting its potential for further expansion. Despite these challenges, the system proved to be an effective tool for reducing human-wildlife conflicts, and future improvements, including broader wildlife detection, predictive analytics, and AI-based behavioural analysis of wildlife movements, could further enhance its reliability and impact.

.

VI. CONCLUSION

The AI-based Wildlife Proximity Alert System successfully integrates YOLO-based image processing, ESP8266 IoT connectivity, an IR sensor, and a buzzer to provide a real-time solution for mitigating human-animal conflicts. By leveraging YOLO object detection, the system efficiently identifies wildlife, while the IR sensor aids in close-range motion detection. The ESP8266 microcontroller facilitates IoT-based communication, enabling remote monitoring and alerts through an MQTT-based messaging system. The system effectively enhances human safety and wildlife conservation by providing timely alerts, reducing potential conflicts, and allowing authorities or residents to take preventive measures. The inclusion of a buzzer-based warning mechanism ensures immediate local notifications, while IoT integration enables remote access to wildlife detection data. In conclusion, this system offers a cost-effective, scalable, and efficient solution for addressing the growing concerns of human-wildlife conflicts. Future improvements may include AI-based behaviour analysis, GPS tracking, and drone-assisted monitoring to enhance detection accuracy and response capabilities further.

REFERENCES

[1] V. M. M, "Melanoma Skin Cancer Detection using Image Processing and Machine Learning," International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 3, no. 4, pp. 780-784, 2019.
[2] B. A. Uzma and T. Sarode, "Skin Cancer Detection Using Image Processing," International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 04, pp. 2875-2881, 2017.
[3] Y. Vikash and D. Vandana, "A study on automatic early detection of skin cancer," Int. J. Advanced Intelligence Paradigms, vol. 12, no. 3/4, pp. 392-399, 2019.
[4] J. Shivangi, j. Vandana and P. Nitin, "Computer-aided Melanoma skin cancer detection using Image Processing," International Conference on Intelligent Computing, Communication & Convergence, pp. 735-740, 2015.
[5] M. Suleiman and K. Akio, "A SVM-Based Diagnosis of Melanoma Using Only Useful Image Features," 2018.
[6] M. Soniya and S. Swati, "A Method for Melanoma Skin Cancer Detection Using Dermo copy Images," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018.

[7] G. P. Asha, Anitha. J and P. Jacinth, "IDENTIFICATION OF MELANOMA IN DERMOSCOPY IMAGES USING IMAGE PROCESSING ALGORITHMS," 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), pp. 553-557, 2018.

[8] W. Zahra, Z. Madeeha, W. Amna and R. Farhan, "An Efficient Machine Learning Approach for the Detection of Melanoma using Thermoscopic Images," 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), pp. 316-319, 2017.

[9] A. J. J, S. Sibi and Aswin. R.B, "Computer Aided Detection 01 Skin Cancer," 2013 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2013], pp. 1137-1142, 2013.

[10] T. Kanimozhi1 and Dr.A. Murthi2, "COMPUTER AIDED MELANOMA SKIN CANCER DETECTION USING ARTIFICIAL NEURAL NETWORK CLASSIFIER," Singaporean Journal of Scientific Research(SJSR), vol. 8, no. 2, pp. 35-42, 2016.

[11] E. Mahmoud, "AUTOMATIC SKIN CANCER IMAGES CLASSIFICATION," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 4, no. 3, pp. 287-294, 2013. [12] G. Diwakar and A. Mushtaq, "Melanoma Detection and Classification Using SVM Based Decision Support System," IEEE INDICON 2015, 2015.

[12] G. Ioannis, M. Nynke, L. Sander, B. Michael, F. J. Marcel, and P. Nicolai, "A computer-assisted melanoma diagnosis system using non-thermoscopic images," Expert Systems with Applications, 2015. [14] M. Jafari, S. Samadi, N. Karimi, S. Soroush Mehr, K. Ward and K. Najarian, "Automatic Detection of Melanoma Using Broad Extraction of," pp. 1357-1360, 2016.

[13] A. Murugan, H. N. S. Anu, and K. K. P. Sanal, "Detection of Skin Cancer Using SVM, Random Forest," Journal of Medical Systems, pp. 1-9, 2019.

[14] A. Alzubaidi, J. Zhang, A. J. Humaidi, et al., “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” Journal of Big Data, vol. 8, no. 1, pp. 1–74.

[15] M. Rahman, D. Muniyandi, and M. Mustapha, “A deep learning model for early detection of skin cancer using CNN and Reset architectures,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 9, pp. 437–444, 2020.

[16] A. Abbas, M. M. Abdisamad, and M. M. Gaber, “Deep Learning Techniques for Classification of Skin Cancer Images,” 2020 International Conference on Advanced Machine Learning Technologies and Applications (AMLTA), pp. 168–179, 2020.

[17] T. Brinker, S. Hekler, J. Enk, et al., “A convolutional neural network trained with thermoscopic images performs on par with 157 dermatologists,” European Journal of Cancer, vol. 111, pp. 148–154, 2019...

[18] M. Hameed, Z. Saba, A. Rehman, et al., “multi-class skin lesion detection and classification using convolutional neural network and transfer learning,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 789–802, 2021

[19] Y. Kawahara, A. BenTaieb, and G. Hamermesh, “Deep Features for Classifying Skin Lesions,” IEEE International Symposium on Biomedical Imaging (ISBI), pp. 1397–1400, 2016.

[20] C. Yu, S. Yang, H. Zhang, et al., “Melanoma recognition in dermo copy images via deep residual networks,” IEEE Access, vol. 8, pp. 123–132, 2020.

[21] M. Nasr-Esfahani, H. Samavi, N. Karimi, et al., “Melanoma detection by analysis of clinical images using convolutional neural network,” 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1373–1376, 2016.

[22] M. Rahman, B. Paul, T. Mahmud, and S. A. Fattah, "CIFF-Net: Contextual Image Feature Fusion for Melanoma Diagnosis," preprint arXiv:2303.03672, 2023.

[23] N. Kanwal, R. Amundsen, H. Hardardottir, et al., "Detection and Localization of Melanoma Skin Cancer in Histopathological Whole Slide Images," preprint arXiv:2302.03014, 2023.

[24] F. Al Zeger, N. Naranpanawa, B. Betz-Stablein, et al., "Application of Machine Learning in Melanoma Detection and the Identification of

'Ugly Duckling' and Suspicious Naevi: A Review," preprint arXiv:2309.00265, 2023.

[25] M. Jafari, S. Samadi, N. Karimi, et al., "Automatic Detection of Melanoma Using Broad Extraction of Features," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1357–1360, 2016.

[26] A. Murugan, S. Anu, and P. Sanal, "Detection of Skin Cancer Using SVM, Random Forest," Journal of Medical Systems, vol. 43, no. 8, pp. 1–9, 2019.

[27] Viknesh, C. K., Kumar, P. N., Seetharaman, R., & Anitha, D. (2023). Detection and Classification of Melanoma Skin Cancer Using Image Processing Technique. Diagnostics, 13(21), 3313.

[28] Murugan**,** A**.,** Anu, S., & Sanal, P. K. (2019). Detection of Skin Cancer Using SVM, Random Forest, and KNN Classifiers. Journal of Medical Systems, 43(8), 1–9.

[29] Rahman, M., Paul, B., Mahmud, T., & Fattah, S. A. (2023). Early Melanoma Detection Based on a Hybrid YOLOv5 and Reset Technique. Diagnostics, 13(21), 3313.

[30] Kanimozhi, T., & Murthi, A. (2016). Computer-Aided Melanoma Skin Cancer Detection Using Artificial Neural Network Classifier. Singaporean Journal of Scientific Research (SJSR), 8(2), 35–42.