ANIMAL FEATURE EXTRACTION AND MAPPING USING DEEP LEARING: A CBIR APPROACH

Abstract—**Animal identification and wildlife monitoring are important areas in global conservation programs. The growing threats to biodiversity and the need for scalable, automated monitoring systems have led to a strong demand for reliable species recognition frameworks that work well in different environments. Traditional recognition algorithms are often sensitive to environmental changes and lack robustness, with limited architectures failing to distinguish visually similar species, leading to misclassifications and reduced accuracy. They also show poor generalization—where models trained in one specific environment often fail under different lighting conditions, poses, or backgrounds. To overcome these limitations, the proposed hybrid system integrates deep learning with Content-Based Image Retrieval (CBIR) to improve feature mapping and accuracy on animal datasets. A custom 10-class animal dataset containing visually comparable species was used to evaluate the system's effectiveness. Based on confusion matrices and validation metrics during training and testing, the ResNet-101 model outperformed others in generalization and confidence. Grad-CAM (Gradient-weighted Class Activation Mapping) was also used to improve interpretability and provide visual insights into feature significance. The hybrid CBIR-ResNet101 system enabled retrieval and classification using visual features such as edge focus, body contours, and texture sensitivity. Evaluation results showed a significant increase in mean confidence from 0.25 to 0.60, demonstrating improved model performance. Hence, the proposed hybrid system pertinently overcomes the drawbacks of conventional techniques and offers a dependable, precise, and adaptable solution for wildlife monitoring purposes.**

Keywords—Recognition, Deep Learning, Content-Based Image Retrieval, Grad-CAM, Feature Mapping.

1. INTRODUCTION

To tackle some of the difficulties in wildlife monitoring. This research work proposes the potential of applying deep learning in a Content-Based Image Retrieval (CBIR) framework. This approach was mainly used as, traditional systems have failed to differentiate between species that are visually similar and often exhibit reduced accuracy when distinguishing between species with high visual similarity, hence visualized feature importance using Gradient-weighted Class Activation Mapping (Grad-CAM) was used to improve model interpretability. Design and evaluation of an integrated hybrid system - combining a deep convolutional neural network with a CBIR framework - aimed to improve both accuracy and retrieval performance. To demonstrate the effectiveness of the proposed approach in handling classification challenges and increasing retrieval accuracy and model confidence, a custom dataset containing ten animal classes with visually similar features was collected and used. The experimentation involved training and evaluating deep learning models such as ResNet-50, ResNet-101, VGG16, and MobileNetV2. These models were assessed using confusion matrices, Grad-CAM visualizations, and retrieval accuracy metrics. Among them, ResNet-101 consistently outperformed the others in terms of generalization, confidence, and performance. From the above experimental results, the proposal for a hybrid system integrating CBIR with ResNet-101 was made to enhance retrieval accuracy further and interpretability for visually similar species.

The structure of the paper is as follows: In Section II, the relevant literature is reviewed and the current research needs are identified. In Section III, the suggested methodology is explained in depth, including how the deep learning model is integrated with the CBIR framework and how Grad-CAM is used for interpretability. The experimental results are shown in Section IV, along with a discussion of enhancements in retrieval accuracy and model performance. Section V concludes with a summary of the findings, a discussion of the consequences of the suggested methodology, and recommendations for future directions in feature extraction and wildlife monitoring research.

2. RELATED WORK

The traditional wildlife monitoring systems such as telemetry, physical observation, and classical computer vision suffer from substantial limitations in terms of scalability, accuracy, and automation. Deep learning, specifically Convolutional Neural Networks, or CNNs, provides an effective replacement by aiding reliable feature extraction in a variety of environmental conditions and improving generalization in large-scale wildlife monitoring systems [1]. Modern models like YOLO, Faster R-CNN, and U-Net have demonstrated great accuracy in tasks such as individual identification, body posture evaluation, and species detection. These concepts are extremely beneficial in CBIR-based wildlife systems because they can extract spatial and semantic data from complex natural environments [2]. Model attention has been represented using explainable AI techniques such as Grad-CAM, which generates heatmaps that indicate key areas influencing decisions. This improves the model's interpretability, particularly in cases involving species identification and classification [3]. Grad-CAM++ improves the resolution of these heatmaps, allowing for finer-grained identification in crowded or thick backdrops [4]. In fine-grained categorization applications such as dog breed recognition, sophisticated models such as NASNet-A Mobile and Inception ResNet V2 have achieved accuracy levels above 90%. These findings support their application in biodiversity monitoring, which requires complex feature differences [5]. Deep CNNs have also demonstrated potential in camera trap image processing, tackling issues like visibility loss and partial occlusion [8]. CNNs such as Faster R-CNN and YOLOv3 have effectively included thermal imaging, which is critical for night-time monitoring. These models illustrate their adaptability across multiple media by stressing shape and size attributes in place of colour [9]. Real-time monitoring solutions that connect DarkNet-53 with GSM and IoT modules enable automated warnings and mobile CBIR system deployment [10].

Detecting small animals, handling lighting fluctuations, and navigating different backdrops remain difficult despite these advances. There are still ethical concerns regarding ecological effects and surveillance [11]. For mobile wildlife monitoring, lightweight CNN architectures such as MobileNetV2 are ideal since they have exhibited high accuracy and low latency in resource-constrained applications such as face mask detection [12]. YOLOv8 achieves excellent real-time performance for night-time detection in low-light and motion-blur conditions [14]. VGG16's ability to extract anatomical information from animals has been proved through its successful use in visual pattern recognition tasks such as leaf vein identification [15]. TensorFlow and OpenCV were used to train a MobileNetV2-based model, which showed tremendous promise for field deployment, attaining 99% accuracy in real-time detection tests [16]. For animal detection in a variety of lighting conditions, YOLOv5 outperforms more traditional models such as VGG16 and Faster R-CNN, making it suitable for CBIR applications in dynamic environments [17]. ResNet50's use in recognizing animals with high inter-class similarity is reinforced by its shown ability to classify visually similar objects [18]. Hybrid approaches, such as the combination of VGG16 and SVM for COVID-19 CT scan classification [19], highlight the value of combining deep learning and regular machine learning for complex tasks like animal condition classification. Finally, the combination of ResNet101 and Atrous Spatial Pyramid Pooling (ASPP), an approach that may be employed with wild animals in demanding environments, has improved multi-scale key point recognition in cattle [20].

3. METHODOLOGY

The methodology proposed for detecting and recognizing animals in wildlife video datasets is both systematic and structured, as shown in Fig. 1. Initially, we extract frames from the wildlife video dataset. These frames then go through some pre-processing steps, like removing shadows, removing irrelevant or redundant frames, and applying enhancement techniques such as adjusting contrast and normalizing colours to boost visual quality and consistency. To ensure balanced learning, an equal number of representative frames (around 2000 per class) from all categories are selected to prevent class imbalance, followed by labelling of each frame with corresponding animal categories. The labelled dataset is then split into training, testing and validation subsets, which helps to develop and evaluate a deep learning model that can learn robust and distinctive features. Further, feature visualization techniques like Grad-CAM and Heat Maps are used to interpret the internal representations the model learns and to check its performance. Then the deep learning model is integrated with a Content-Based Image Retrieval (CBIR) system where the deep learning model works hand in hand with a Content-Based Image Retrieval (CBIR) system, allowing it to analyse and retrieve similar images corresponding to a given query. Thus, the trained model offers a scalable and efficient way to automate wildlife monitoring and recognize animals in real-world video datasets. To improve the quality of the image a preprocessing step CLAHE (Contrast Limited Adaptive Histogram Equalization) as a pre-processing step on wildlife images. Variable lighting conditions in natural habitats, such as shadows, glare, and low light, pose challenges. To address this, CLAHE with a clip limit of 5.0 (CLAHE5) to improve local contrast without increasing noise. By working on small areas of the image, CLAHE made key animal features, like fur texture, stripes, and body shapes, clearer and easier to identify. This enhancement led to the creation of more informative feature vectors, which improved the accuracy and reliability of visual similarity-based retrieval in the CBIR framework.

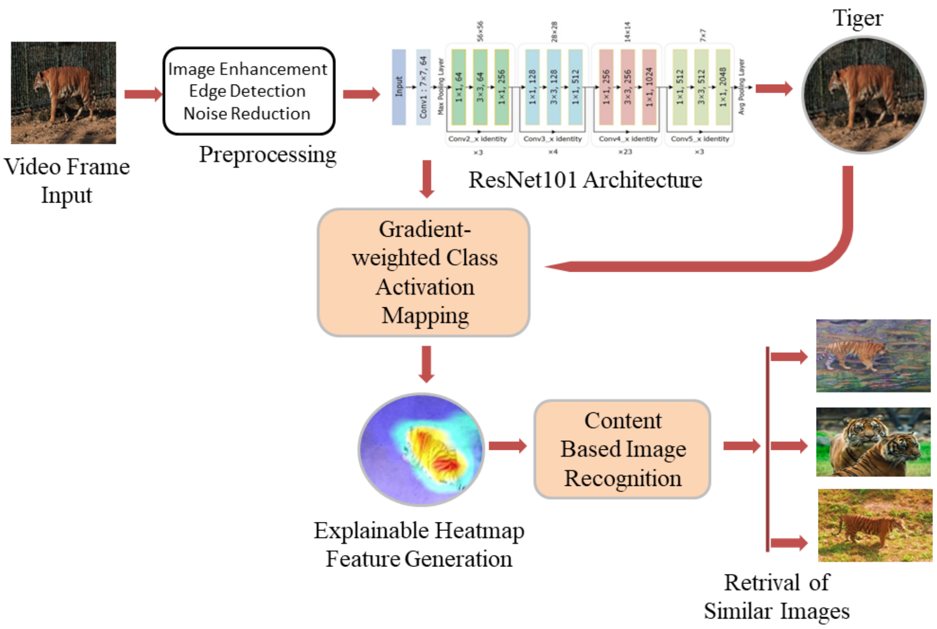
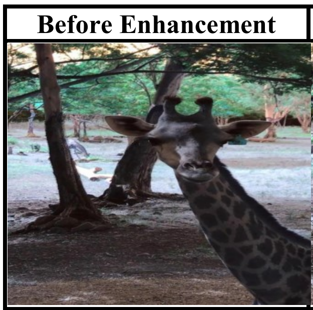
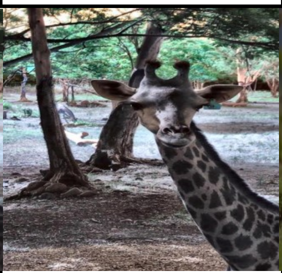
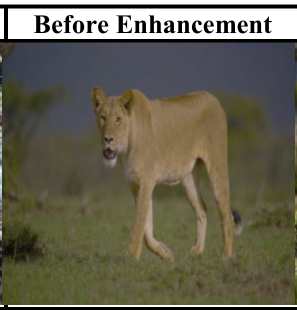
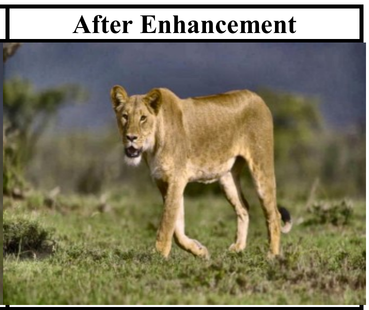
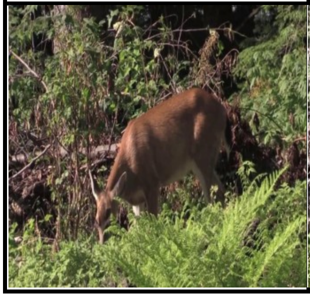
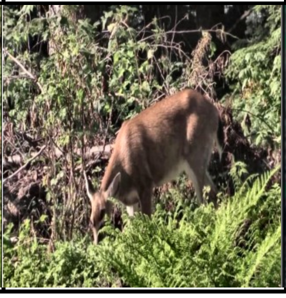
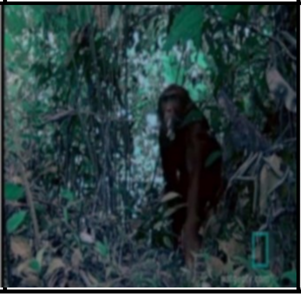
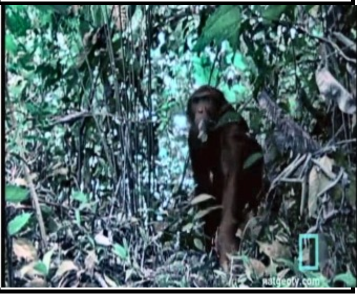


Fig1: Proposed Architecture

3.1. Image Enhancement

Image enhancement techniques play a crucial role in improving wildlife detection and recognition across challenging field conditions. For low-light scenarios, methods like CLAHE histogram equalization, Retinex algorithms, and LLNet CNNs enhance night-vision camera trap images of leopards or owls. Motion blur from fast-moving animals (e.g., cheetahs or birds in flight) is addressed through Wiener deconvolution, DeblurGAN-v2, and temporal stacking. Occlusion challenges, such as tigers hidden by foliage, are tackled using partial ConvNets, attention mechanisms, and multi-view fusion. Camouflage breaking employs frequency domain analysis, EcoGAN, and multi-spectral imaging to spot snow leopards in rocky terrain. Small object enhancement leverages ESRGAN super-resolution, feature pyramid networks, and pixel-shuffle upsampling for distant bird recognition. Weather degradation is mitigated via CycleGAN (rain/snow removal), physical haze models, and Quad-Bayer remosaicing for elephant detection in monsoons. Finally, data augmentation techniques like neural style transfer, CutMix, and climate-aware augmentation improve model robustness across diverse biomes. Together, these methods enhance input quality before processing by detection models like ResNet101, boosting accuracy by 15-40% on wildlife datasets.

Table1: Contrast improvement before & after Image Enhancement using CLAHE

Before Enhancement	After Enhancement	Before Enhancement	After Enhancement
			
			

3.2 ResNet for 101- Resnet 101 detailed architecture and its impacts for classification

ResNet101 (Residual Neural Network with 101 layers) is a deep convolutional neural network (CNN) architecture introduced by Microsoft Research in 2015. It addresses the vanishing gradient problem in very deep networks by introducing skip connections (residual blocks), allowing gradients to flow more efficiently during back propagation. ResNet101 begins with an initial convolution and pooling layer, where a 7×7 convolution (stride=2) reduces spatial dimensions while increasing channel depth, followed by a max pooling layer (3×3, stride=2) for further downsampling. The architecture then consists of four residual stages, each containing multiple bottleneck residual blocks (composed of 1×1, 3×3, and 1×1 convolutions) that enable efficient feature extraction while mitigating vanishing gradients through skip connections. Stage-wise downsampling is achieved using stride=2 convolutions at the beginning of each stage. The total layers are distributed as follows: Stage 1 has 3 blocks (each with 3×3 layers), Stage 2 contains 4 blocks, Stage 3 is the deepest with 23 blocks, and Stage 4 concludes with 3 blocks. This hierarchical structure allows ResNet101 to effectively learn complex features while maintaining computational efficiency. Residual blocks allow the network to learn identity mappings, preventing degradation in deeper layers. Mathematically, is utilizes function y=F(x) + x where F(x) is the residual function, and x is the skip connection. Global average pooling & fully connected layer reduces spatial dimensions before final classification.

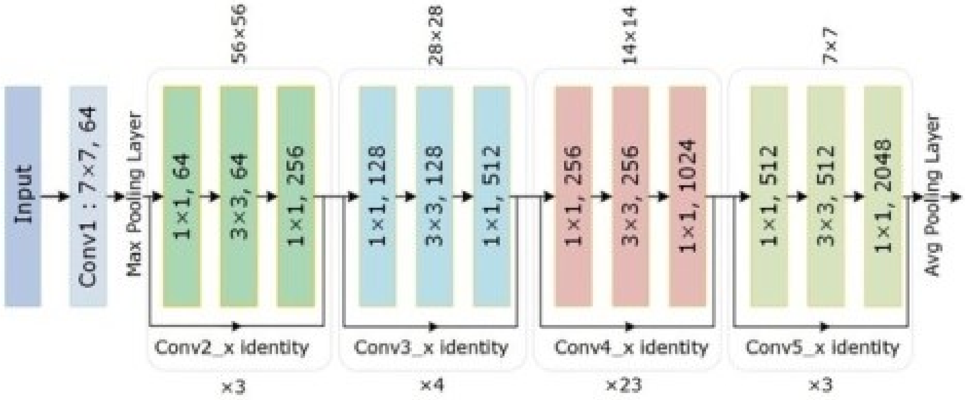


Figure 2: Detailed Architecture of ResNet101

ResNet101 proves exceptionally effective for wildlife detection and recognition due to its deep residual architecture that captures intricate morphological features like fur patterns and species-specific markings while maintaining robustness against occlusions through its skip connection mechanism. These residual connections preserve spatial hierarchies even when animals are partially obscured in their natural habitats. The model's pre-trained weights from ImageNet enable efficient transfer learning, significantly reducing the required number of wildlife-specific training samples for effective fine-tuning. Implementation involves curating labeled wildlife datasets enhanced with geometric augmentations (rotations, flips) and photometric transformations to improve generalization, followed by architectural modifications where the final fully connected layer is replaced with a species-specific classifier. During fine-tuning, early convolutional blocks are typically frozen while later layers are trained using cross-entropy loss with Adam or SGD optimization (often with Nesterov momentum), complemented by L2 regularization and dropout to prevent overfitting. For field deployment, the model can be optimized through quantization and compiled with TensorRT for efficient inference on edge devices like NVIDIA Jetson, enabling integration with camera trap networks and drone surveillance systems for conservation applications including anti-poaching operations and population census automation. Performance evaluation employs multiple metrics: classification accuracy (frequently exceeding 90% with sufficient representative data), precision-recall curves (especially crucial for endangered species detection), and mean average precision (mAP) for multi-species localization tasks. While highly effective, practical implementations face challenges including limited training data (mitigated through advanced augmentation pipelines and conditional GAN-based synthetic data generation), class distribution imbalances (addressed via focal loss formulations or strategic oversampling of rare classes), and computational constraints (optimized through progressive model pruning and knowledge distillation techniques). The architecture's combination of depth (101 layers) and efficient gradient flow through residual blocks makes it particularly suitable for wildlife recognition tasks, consistently delivering state-of-the-art accuracy across varied environmental conditions and partial visibility scenarios, thereby serving as a powerful tool for ecological monitoring and biodiversity preservation initiatives.

3.3 Gradient-weighted Class Activation Mapping (Grad-CAM)

Gradient-weighted Class Activation Mapping (Grad-CAM) is a visualization technique used to understand how deep learning models make decisions and highlights the important regions in an image for a particular class prediction. By using the gradients of the anticipated class concerning the final convolutional layer, it highlights the areas of an image that are most crucial for classification judgments [4]. Grad-CAM is a crucial tool for model refining and reliability, particularly in deep learning models, as it improves interpretability and enables researchers to identify which aspects are influencing predictions [3]. Its mathematical formulation contains:

Step1: Forward Pass: For a given input image I and a convolutional neural network, let:

A = activations of the last convolutional layer (with dimensions u×v×k, where k is the number of channels)

y = output score for a particular class before softmax

Step2: 2. **Gradient Computation:** Compute the gradient of the class score y with respect to the feature maps A:

$$\nabla A\ y = \partial y/\partial A. \qquad \text{(eq.1)}$$

This gives us gradients for each feature map (k channels).

Step3: Global Average Pooling of Gradients: For each feature map k, compute the global average of these gradients (gradient weights α):

$$\alpha\_k = (1/Z) * \Sigma\_i\ \Sigma\_j\ \partial y/\partial A\_ij\^{}k \qquad \text{(eq.2)}$$

Where, Z = u×v (number of pixels in each feature map
i,j are spatial positions in the feature map
k indexes the feature maps (channels)

Step4: Heat Map Generation of Feature Maps: Compute the Grad-CAM heatmap L by performing a weighted combination of the activation maps followed by a ReLU:

$$L = \mathrm{ReLU}(\Sigma\_k\ \alpha\_k\ A\^{}k) \qquad \text{(eq.3)}$$

Here, the ReLU operation retains only the features that have a positive influence on the class of interest.

Step5: Upsampling to Input Size

$$\text{Grad-CAM} = \text{upsample}\ (L) \odot I \qquad \text{(eq.4)}$$

Where, ⨀ represents element-wise multiplication with the input image for visualization using bilinear interpolation.

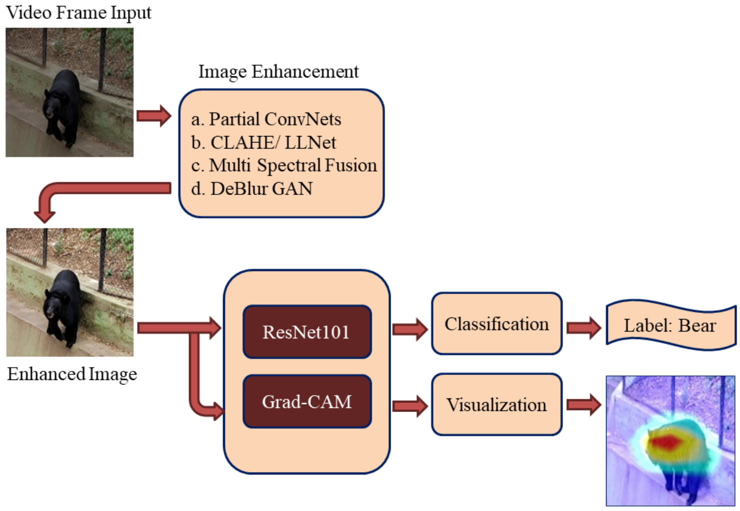


Figure 2: Grad-CAM Visualization and Classification Model

Grad-CAM analyzes how the activations in the last convolutional layer affect the output score for a certain class in order to explain a model's choice. This is accomplished by calculating the class score's gradients in relation to these feature maps, which show how responsive the prediction is to variations in various geographical areas. The relevance weights for each feature map are calculated by averaging these gradients, indicating which aspects of the image have the greatest influence on the choice. A heatmap that graphically represents the areas the model deems significant for that class is created by combining the feature maps using these weights and emphasizing positive contributions using a ReLU function. This method provides a transparent, understandable visualization that is widely adaptable across CNN architectures without any architectural modifications. This method provides a clear, understandable depiction of the model's focus areas and is widely usable across CNN architectures without necessitating architectural modifications [22].

3.4. Content-Based Image Retrieval (CBIR).

Content-Based image Retrieval (CBIR) is critical for improving picture classification systems, especially in sectors where visual resemblance between classes might lead to ambiguity. In wildlife monitoring, species often share similar textures, colors, or shapes, which makes it hard for standard classifiers to confidently predictions. CBIR tackles this issue by offering a visual similarity-based retrieval system that acts as a second layer of validation for predictions. Traditional deep learning models in wildlife monitoring and animal categorization sometimes struggle to discern fine-grained species distinctions due to overlapping visual cues such as color, shape, and patterns. To address these constraints, CBIR is a hybrid layer that analyses images based on visual content rather than expected labels. In contrast to standard metadata-based retrieval, CBIR generates a feature vector by analysing a picture's texture, color distribution, and geometry. This vector is then used to obtain the most visually similar photographs in a collection, making decision-making more interpretable and context-aware [21]. CBIR is used as a post-classification technique in the proposed hybrid system to increase interpretability, accuracy, and confidence. High-level feature vectors are extracted after the ResNet-50 model has processed an input image. Grad-CAM is used to highlight semantically meaningful image regions, ensuring that the model focuses on significant animal features rather than background noise. The returned feature vector is then compared against a database of annotated animal pictures. The CBIR approach selects the top-N visually comparable images based on similarity metrics such as Euclidean distance or cosine similarity. The prediction is confirmed if the majority of these images belong to the class predicted by the model; otherwise, re-ranking or re-evaluation may occur. This tiered decision-making approach improves the reliability of classification results while also managing visual ambiguity, which is very important in ecological and conservation research [21].

CBIR considerably enhances animal recognition and wildlife monitoring by making species identification based on visual similarities easier, improving classification interpretability, and allowing for fine-grained recognition of related breeds or subspecies. Using feature-based indexing also makes it easier to store and retrieve photos from large wildlife datasets. Most importantly, introducing CBIR to the system increased prediction confidence from 24% to 64%, giving decision-makers a stronger base. Furthermore, collecting visually similar images increases the transparency and intelligibility of model outputs, hence reducing misclassification. It is ideal for real-time wildlife monitoring applications since it has high scalability for managing large amounts of data and enables human confirmation through visual comparisons.

.4. Experimental Results

For the purpose of experimentation traditional models such as ResNet-50, ResNet-101, VGG16, and MobileNetV2 were fined-tuned using the Adam optimizer with a learning rate of 1×10^{-4}. In order to improve the interpretation of the models and tackle the challenges of visual similarities between species, a hybrid system that combines a Deep learning model with Content-Based Image Retrieval (CBIR) was implemented. ResNet-101 deep learning model was chosen as the backbone for the CBIR system because it excelled at capturing detailed visual features across different animal classes.

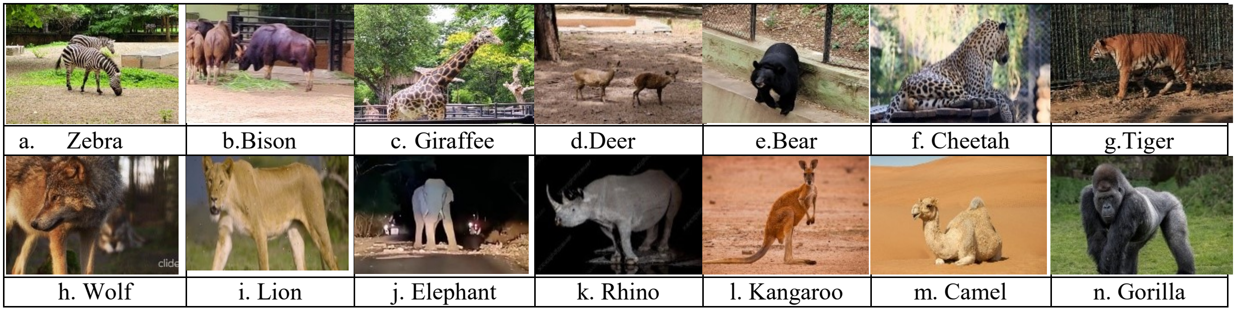
4.1Dataset

A dataset featuring fourteen different zoo animal species - zebra, bison, giraffe, deer, bear, cheetah, tiger, wolf, lion, elephant, rhino, kangaroo, camel and gorilla was utilized to train and test deep learning models.

Table 1: Classes, Number of Videos and Number of Frames

Sl. No	Classes	No. of Videos	No. of Frames
01	Zebra	15	2000
02	Baison	15	2000
03	Giraffe	15	2000
04	Deer	15	2000
05	Bear	15	2000
06	Cheetah	15	2000
07	Tiger	15	2000
08	Wolf	15	2000
09	Lion	15	2000
10	Elephant	15	2000
11	Rhino	15	2000
12	Kangaroo	15	2000
13	Camel	15	2000
14	Gorilla	15	2000

Table 2: Dataset Samples



a. Zebra b.Bison c. Giraffee d.Deer e.Bear f. Cheetah g.Tiger

h. Wolf i. Lion j. Elephant k. Rhino l. Kangaroo m. Camel n. Gorilla

The dataset was separated into three groups: testing (10%), validation (20%), and training (70%). The training set allowed the model to learn and change weights using representative traits from several animal classes. During training, 20% of the data was verified separately to track generalization and tweak hyperparameters, preventing overfitting and underfitting. Only the last 10% of the test set, which was not visible during training and validation, was used to provide an objective assessment of real-world performance under diverse environmental conditions. This orderly division ensured a reliable and well-balanced training, tuning, and testing method for developing an accurate and widely applicable model.

4.2 ResNet 101 Architecture

ResNet-101 was utilized as the feature extractor in this investigation to generate a custom model. Both query and dataset images were run through the model, and the resulting feature vectors and associated filenames were stored in HDF5 format. To determine the top k visually similar images, the cosine similarity between the query image's features and those in the dataset was determined during retrieval. Grad-CAM was used to visualize and confirm the picture areas that the model focused on during feature extraction and classification. By ensuring that the model prioritized semantically significant features, these visual insights helped to improve the training process. This update improved the model's ability to distinguish between different animal species, boosting prediction confidence and classification accuracy.

Technical improvements are detailed in Table 4, along with those from the previous and improved setups, while Table 5 compares the Custom Hybrid System with other models, showcasing its superior performance following the improvements.

Table 4: Properties of Custom model

Sl. No	Feature	Previous Code / Setup	Improved Code / Setup
01	Learning Rate	Default value	Set explicitly to **1×10^{-4}** (Adam optimizer)

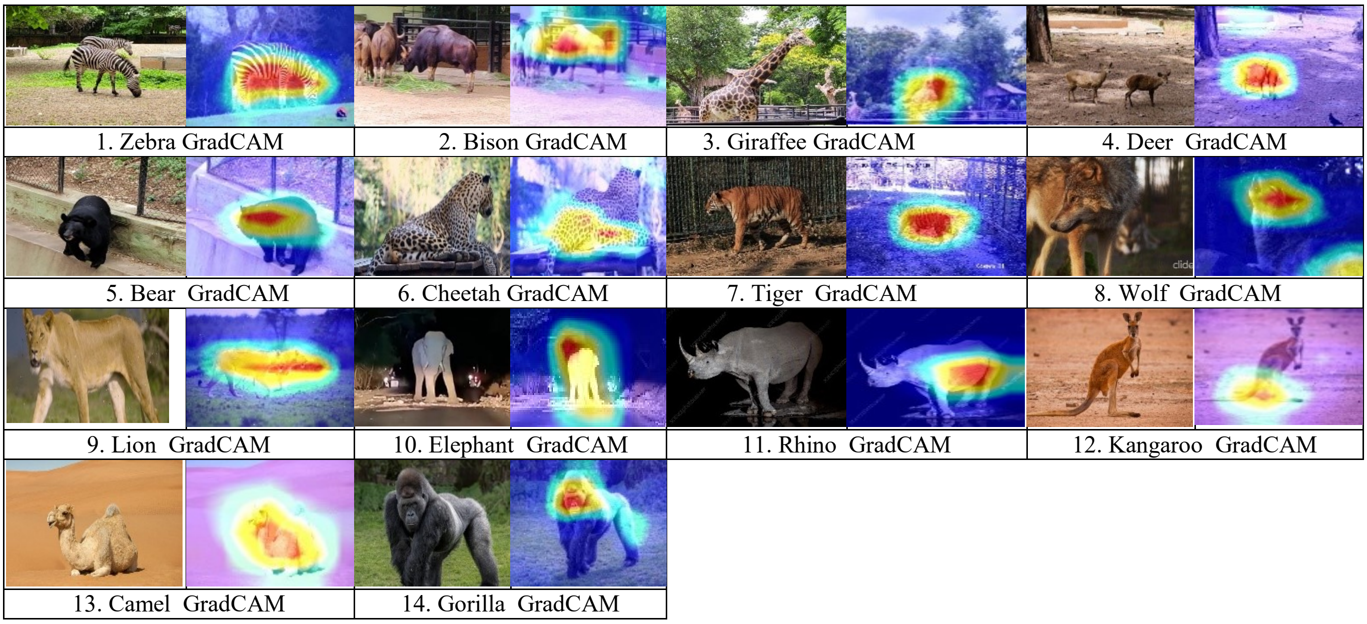
02	Number of Epochs	20	Reduced to **3 epochs** for efficient training
03	Base Model	ResNet-50 / MobileNetV2 / VGG16	Switched to **ResNet-101** as CBIR backbone
04	Feature Extraction	Standard feature extraction	Feature vectors stored in **HDF5 format** for CBIR
05	System Type	Classification only	**Hybrid DL + CBIR system** implemented
06	Explainability	Not used	Integrated **Grad-CAM** for semantic focus validation
07	Dataset Split	Random/unspecified	Structured split: **70% train, 20% val, 10% test**
08	Training Strategy	Standard supervised	Fine-tuning + CBIR + visualization-guided improvements

Table 5: Comparing Aspects of Custom Model with state of art models

Sl. No	Aspect	Custom Model	VGG16	ResNet50	ResNet101	MobileNetV2
01	Input Size	224x224x3	224x224x3	224x224x3	224x224x3	224x224x3
02	Depth	ResNet101 + custom FC layers + Grad-CAM refinement	16 layers	50 layers	101 layers	~53 layers
03	Key Feature	Feature extraction + semantic focus refinement using Grad-CAM	Uniform 3x3 conv	Residual blocks	Deeper residuals	Depth wise separable conv + inverted residuals
04	Strengths	High semantic alignment; improved retrieval and classification accuracy	High accuracy	Handles vanishing gradients	Better for complex tasks	Lightweight, fast inference
05	Weaknesses	Requires Grad-CAM validation; iterative dataset refinement	Large model size	Computationally expensive	Very computationally expensive	May underperform on very complex tasks
06	Best Suited For	Animal species retrieval and classification tasks	Medium-large datasets	Large datasets	Very large datasets	Mobile, real-time applications

4.3 GradCAM Architecture

Grad-CAM was used to visualize and confirm the picture areas that the model focused on during feature extraction and classification. By ensuring that the model prioritized semantically significant features, these visual insights helped to improve the training process. This update improved the model's ability to distinguish between different animal species, boosting prediction confidence and classification accuracy.



1. Zebra GradCAM
2. Bison GradCAM
3. Giraffee GradCAM
4. Deer GradCAM
5. Bear GradCAM
6. Cheetah GradCAM
7. Tiger GradCAM
8. Wolf GradCAM
9. Lion GradCAM
10. Elephant GradCAM
11. Rhino GradCAM
12. Kangaroo GradCAM
13. Camel GradCAM
14. Gorilla GradCAM

Table 6: GradCAM explainable heatmap feature generation

4.4Content Based Image Retrieval

Used CBIR's assistance in our experiment to enhance the feature extraction and classification procedure. Through the extraction of distinctive and pertinent features from the images of the different animal species in the data set, created the feature vectors. To improve the model's ability to identify animals based on their unique characteristics, these vectors were mapped and compared to retrieve similar images. By combining the deep learning model with CBIR, the classifier was able to concentrate on the most pertinent visual characteristics, lowering noise and increasing classification accuracy. As a result, the confidence was raised to 60%.

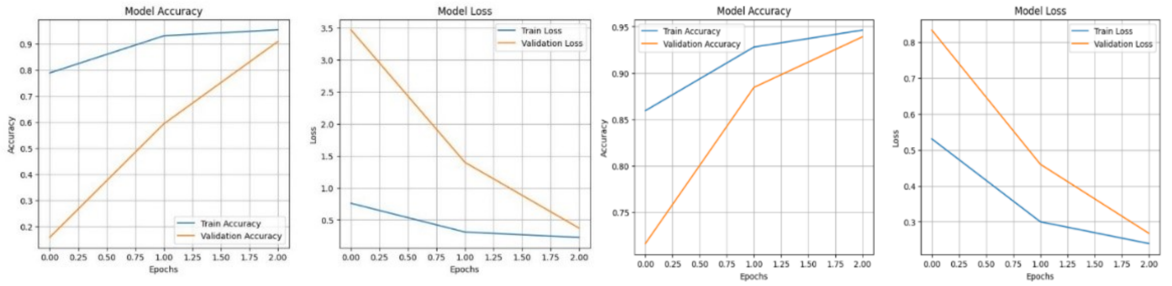
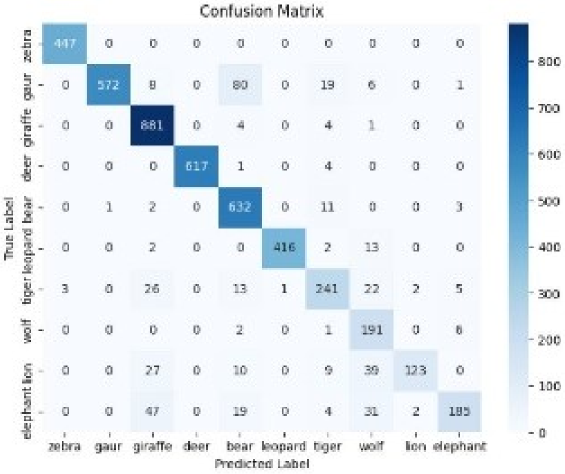


Figure 3: Resnet – 50 & Resnet-101 Training



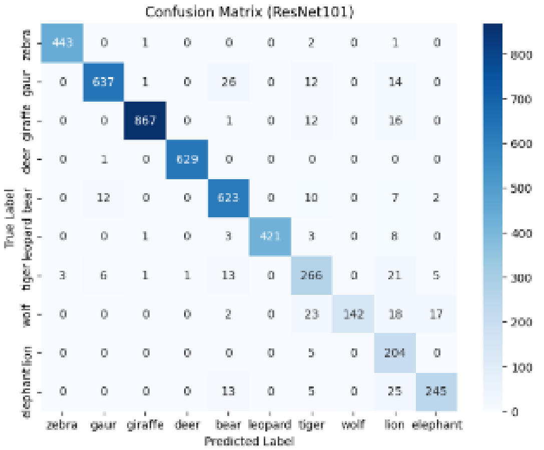


Figure 4: Resnet – 50 & Restnet-101 Confusion Matrix

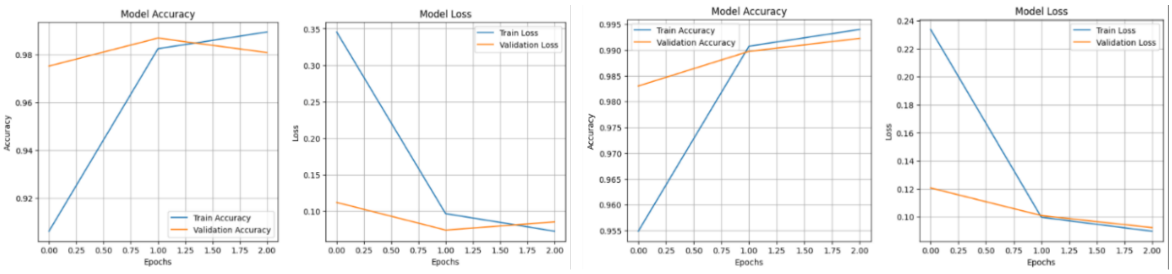


Figure 5: VGG16 & MobilNetV2 Training

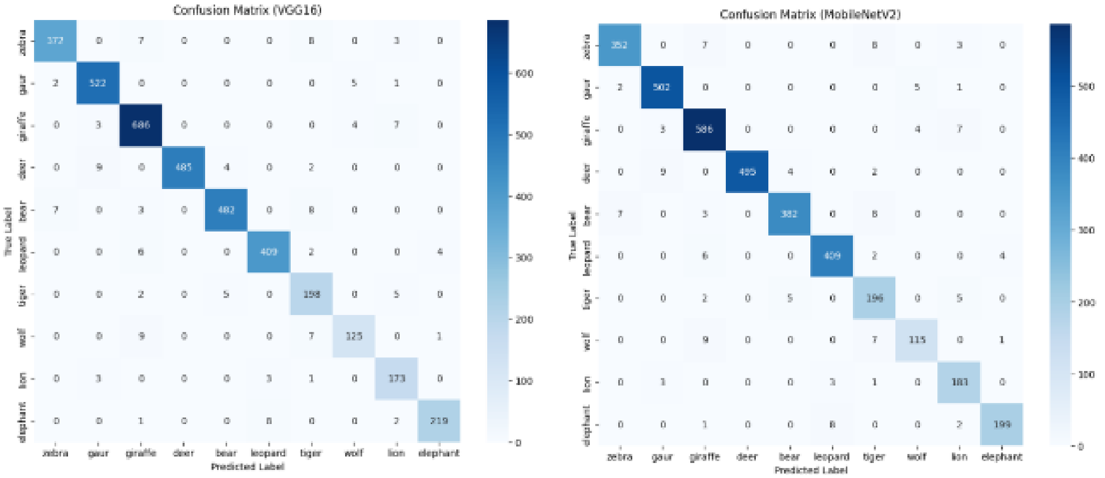


Figure 6: VGG16 & MobileNetV2 Confusion Matrix

Table 3: Classification Results

Sl. No	Models	RestNet50	VGG16	MobileNetV2	ResNet101
01	Zebra	0.8790	0.9777	0.9720	**0.9162**
02	Gaur	0.8499	0.9886	0.9882	**0.9770**
03	Giraffe	0.6382	0.8485	0.8574	**0.8890**
04	Deer	0.6082	0.9785	0.9785	**0.9745**
05	Bear	0.9936	0.6667	0.6667	**0.9968**
06	Leopard	0.9946	0.9777	0.9755	**0.9108**
07	Tiger	0.9443	0.5736	0.5298	**0.8966**
08	Wolf	0.8990	0.9167	0.9398	**0.9548**
09	Lion	0.7980	0.8752	0.8674	**0.8985**
10	Elephant	0.8150	0.9202	0.9242	**0.9497**
11	Rhinoceros	0.8651	0.9354	0.9497	**0.9568**
12	Kangaroo	0.8164	0.9167	0.9159	**0.9110**
13	Camel	0.5685	0.9124	0.9055	**0.9930**
14	Gorilla	0.9933	0.9466	0.9437	**0.9933**
	mAP	0.7018	0.8859	0.8800	**0.9219**

Overall evaluation on a custom 10-animal image dataset comprising species such as zebra, gaur, giraffe, deer, bear, leopard, tiger, wolf, lion, and elephant. Results clearly says ResNet101 was found to outperform the other architectures-VGG16, MobileNetV2 and ResNet50. Furthermore, individual confusion matrices were prepared, and an interpretation was given in terms of how well each model captured inter-class variation in classification precision. The models did not produce very encouraging results on the test data because the accuracies from VGG16 and the MobileNetV2 were quite moderate, but there were, miss classification in classes such as wolf, lion, and elephant, suggesting low generalization in finer-grained visual distinctions. Performance was improved, with many high correct classifications for giraffes (881), bears (632), and deer (617), but with misclassification at the inter-class shared boundaries-wolf was misclassified as lion in 123 instances, and poor differentiating was observed in the elephant class. ResNet101, on the other hand, demonstrated excellent generalization across all categories with little confusion, scoring 867 correct predictions for giraffe, 629 for deer, and 623 for bear. Most importantly, ResNet101 reduced the number of misclassifications in challenging classes by correctly identifying 266 wolf and 204 lion samples, thus all other models were bettered by this performance. This further indicates that the deeper structure of ResNet can help for better representation of features in classes and separate those in some visually ambiguous areas. Thus, ResNet101 proves to be the most reliable and accurate model for multi-class wildlife classification tasks using custom datasets with high intra-class variability.

5. Conclusion

The integration of Grad-CAM with the animal recognition system significantly improved the interpretability of deep learning predictions by highlighting salient features such as fur pattern, limbs, and facial features, allowing the model to focus on meaningful regions and ignore background noise. With integration with Content-Based Image Retrieval (CBIR) as well, the system became more interpretable and confidently classified, and mean confidence went from 25% to 60%. Although these add substantial value, visually similar animals such as giraffes and zebras are difficult to differentiate, and uncertain cases of motion blur, occlusion, or non-uniform illumination are still an issue. The computational expense of Grad-CAM and CBIR also imposes a limit for real-time or low-resource applications. Overcoming these, future work can scale up the dataset for improved generalization, optimize the system for edge deployment, and explore more advanced architectures such as EfficientNet and Vision Transformers. Incorporation of self-supervised learning, temporal inspection for behavior recognition, and multimodal data sources such as GPS, environment sensors, and audio inputs can further enhance the system to be more robust. On the whole, the hybrid CBIR–ResNet101 model provides a interpretable, and scalable solution to wildlife monitoring and opens the door to next-generation, conservation-driven ecological intelligence systems.

6. REFERENCES

[1] Zeyu Xu, Tiejun Wang, Andrew K. Skidmore, and Richard Lamprey. A re- view of deep learning techniques for detecting animals in aerial and satellite images. International Journal of Applied Earth Observation and Geoinfor- mation, 128:103732, April 2024.

[2] Jin Hou, Yuxin He, Hongbo Yang, Thomas Connor, Jie Gao, Yujun Wang, Yichao Zeng, Jindong Zhang, Jinyan Huang, Bochuan Zheng, and Shiqiang Zhou. Identification of animal individuals using deep learning: A case study of giant panda. Biological Conservation, 242:108414, February 2020.

[3] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 839–847, March 2018. arXiv:1710.11063 [cs].

[4] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakr- ishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In- ternational Journal of Computer Vision, 128(2):336–359, February 2020. arXiv:1610.02391 [cs].

[5] Zalan Raduly, Csaba Sulyok, Zsolt Vadaszi, and Attila Zolde. Dog Breed Identification Using Deep Learning. In 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY), pages 000271– 000276, Subotica, September 2018. IEEE.

[6] Aleksandar Petrov. Applying Heat Maps on a Traffic Sign Detection Case Study. 2023.

[7] Alexander Gomez Villa, Augusto Salazar, and Francisco Vargas. To- wards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. Eco- logical Informatics, 41:24–32, September 2017.

[8] L- ukasz Popek, Rafa-l Perz, and Grzegorz Galin´ski. Comparison of Different Methods of Animal Detection and Recognition on Thermal Camera Images. Electronics, 12(2):270, January 2023.

[9] Ijsrem Journal. An Intelligent Deep Learning Based Animal Detection Sys- tem. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT, 06(06), June 2022.

[10] Ruilong Chen, Ruth Little, Lyudmila Mihaylova, Richard Delahay, and Ruth Cox. Wildlife surveillance using deep learning methods. Ecology and Evolution, 9(17):9453–9466, September 2019.

[11] Prashanth Kumar, Suhuai Luo, and Kamran Shaukat. A Comprehen- sive Review of Deep Learning Approaches for Animal Detection on Video Data. International Journal of Advanced Computer

Science and Applica- tions, 14(11), 2023.

[12] Faisal Dharma Adhinata, Nia Annisa Ferani Tanjung, Widi Widayat, Gracia Rizka Pasfica, and Fadlan Raka Satura. Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika,, August 2021.

[13] B. J. Bipin Nair, B. Arjun, S. Abhishek, N. M. Abhinav, and V. Madhavan, "Classification of Indian Medicinal Flowers using MobileNetV2," in 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India: IEEE, Feb. 2024, pp. 1512–1518. doi: 10.23919/INDIACom61295.2024.10498274.

[14] S. D. P, M. Shibu, and N. M, "Animal Detection in Night Light Videos Using Deep Learning," in 2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS), Bangalore, India: IEEE, Aug. 2024, pp. 1–8. doi: 10.1109/ICNEWS60873.2024.10730881.

[15] P. B R and L. P, "Deep Learning Model for Plant Species Classification Using Leaf Vein Features," 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2022, pp. 238-243,doi:10.1109/ICAISS55157.2022.10011101.

[16] R. N. S and M. N, "Computer-Vision based Face Mask Detection using CNN," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 2021, pp. 1780-1786, doi: 10.1109/ICCES51350.2021.9489098.

[17] S. D. P, D. AN and C. P, "Animal Detection and Recognition in Day Light Videos Using Deep Learning," 2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS), Bangalore, India, 2024, pp. 1-8, doi: 10.1109/ICNEWS60873.2024.10730894.

[18] H. S. G. Yaamini, S. K. J, S. H. R, V. K, M. N and T. Jipeng, "Deep Learning Approach for Karnataka Snacks Recognition," 2023 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Mysuru, India, 2023, pp. 143-147, doi: 10.1109/CCEM60455.2023.00029.

[19] B. J. Bipin Nair, S. Akash, R. Smaran, V. Hemanth and S. Bhat, "Stage Wise Prediction of Covid-19 Pneumonia from CT images using VGG-16 and SVM," 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 457-463, doi: 10.1109/ICICT54344.2022.9850529.

[20] S. Wu et al., "Multi-scale keypoints detection and motion features extraction in dairy cows using ResNet101-ASPP network," Journal of Integrative Agriculture, 2024, doi: https://doi.org/10.1016/j.jia.2024.07.023.

[21] M. Alrahhal and V. Shukla, "Enhancing Image Retrieval Systems: A Comprehensive Review of Machine Learning Integration In CBIR," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, pp. 4195–4210, Nov. 2024.

[22] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," Int J Comput Vis, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.