Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

ANIMAL IMAGE DETECTION FOR COLLISION AVOIDANCE USING IMAGE PROCESSING IN MACHINE LEARNING.

Ms. Neelam Gaikwad, PG Scholer, Dept. Of Computer Engineering, Trinity College of Engineering and Research, Pune.
Dr. Saleha Saudagar Assistant Professor, Dept. of Computer Engineering, Trinity College of Engineering and Research, Pune.

Abstract
Several machines learning models, including convolutional neural networks (CNNs), support vector machines (SVMs), and decision trees, are evaluated and compared for their effectiveness in animal detection. Performance metrics such as accuracy, precision, recall, and F1-score are used to assess the models' performance. The experimental results demonstrate the efficacy of the proposed approach in accurately detecting animals on roads under various environmental conditions and lighting conditions. Moreover, the system exhibits robustness against common. Challenges such as occlusions and variations in animal poses. Future work may involve the deployment of the developed system in real-world settings and the integration of additional sensors for comprehensive collision avoidance systems.

Keywords:
Cascade classifier; Computer vision; Histogram of oriented gradient; Haar; Image Processing; Intelligent vehicle system; OpenCV; Road injuries.

I. Introduction

Since Alex Net has stormed the research world in 2012 ImageNet on a large-scale visual recognition challenge, for detection in-depth learning, far exceeding the most traditional methods of artificial vision used in literature. In artificial vision, the neural convolution networks are distinguished in the classification of images. Object detection [9] and location in digital images has become one of the most important applications for industries to ease user, save time and to achieve parallelism. This is not a nontechnique improvement in object detection is still required in order to achieve the targeted objective more efficiently and accurately. The main aim of studying and researching computer vision is to simulate the behaviour and manner of human eyes directly buying computer and later on develop a system that reduces human efforts shows the basic block diagram of detection and tracking. In this paper, an SSD and Mobile Nets based algorithms are implemented for detection and tracking in python environment. Object detection involves detecting region of interest of object from given class of image. Different methods are –Frame differencing, Optical flow, Background subtraction. This is a method of detecting and locating an object which is in motion with the help of camera. Detection and tracking algorithms are described by extracting the featuresofimage and video for security applications [3] [7] [8]. Features are extracted using CNN and deep learning [9]. Classifiers are used for image classification and counting [6].
YOLO based algorithm with GMM model by using the concepts of deep learning will give good accuracy for feature extraction and classification [10]. Section II describes SSD and Mobile Nets algorithm, section III explains method of implementation, and section IV describes simulation results and analysis.Figure 1: Progression of sensors

II. Literature

Introduction to Wildlife-Vehicle Collisions
Overview of the problem: Wildlife-vehicle collisions pose significant threats to both human safety and wildlife conservation.
Statistics and impact: Highlighting the scale of the problem and its ecological and economic ramifications.

Current challenges: Discussing limitations of existing collision avoidance systems in detecting animals effectively.
Traditional Approaches to Collision Avoidance
Sensor-based systems: Review of radar, LiDAR, and infrared-based approaches for detecting animals on roads and railways.
Limitations: Discussing drawbacks such as limited range, susceptibility to environmental conditions, and high cost.
Role of Image Processing and Machine Learning
Introduction to image processing techniques: Brief overview of image preprocessing, feature extraction, and object detection algorithms.
Importance of machine learning: Discussion on the potential of machine learning algorithms, particularly convolutional neural networks (CNNs), in enhancing animal detection accuracy.
Previous Studies on Animal Detection
Survey of existing literature: Summarize key studies and research papers that have addressed animal detection using image processing and machine learning.
Highlight notable methodologies: Discuss different approaches, including dataset creation, model architectures, and evaluation metrics.
Datasets for Animal Image Detection
Available datasets: Overview of publicly available datasets containing images of animals commonly encountered on roads and railways.
Challenges and limitations: Discuss issues such as dataset imbalance, variability in image quality, and species diversity.
State-of-the-Art Techniques
Recent advancements: Review state-of-the-art techniques in animal image detection, including deep learning architectures, transfer learning, and data augmentation strategies.
Performance benchmarks: Highlight performance metrics achieved by cutting-edge models on benchmark datasets. Integration with Collision Avoidance Systems
Feasibility assessment: Discuss challenges and considerations for integrating animal detection models into existing collision avoidance systems.
Real-world applications: Showcase examples of successful implementations and case studies demonstrating the effectiveness of integrated systems.
Future Directions and Challenges
Emerging trends: Discuss potential future directions in animal image detection for collision avoidance, such as multi-modal sensor fusion and real-time processing
Addressing challenges: Identify remaining challenges, including scalability, adaptability to diverse environments, and ethical considerations.

III. Existing System

Designing a system for animal image detection for collision avoidance involves several key components and steps. Here's an outline of the existing system architecture and process:

Data Collection and Annotation:

Collect a diverse dataset of images containing various animals that may be encountered on roads or in the vicinity of vehicles.
Annotate these images with bounding boxes or segmentation masks to indicate the location and extent of each animal within the image.
Preprocessing:
Resize images to a consistent resolution to ensure uniformity.
Normalize pixel values to a common scale to reduce variations in lighting conditions.

Model Selection:

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024
Choose a suitable deep learning architecture for image classification and object detection. Common choices include Convolutional Neural Networks (CNNs) and variants like Faster R-CNN, YOLO (You Only Look Once),or SSD (Single Shot Multibook Detector).
Consider pre-trained models such as Reset, Mobile Net, or Efficient Net, which can be fine-tuned on your specific dataset to expedite training.

Training:

Split the annotated dataset into training, validation, and testing sets.
Train the chosen model using the training set, optimizing it to accurately detect animals while minimizing false positives.
Validate the model's performance using the validation set and adjust hyperparameters as needed to prevent overfitting.
Evaluate the model on the test set to assess its generalization capability.

Post-processing:

Apply non-maximum suppression to remove redundant bounding boxes and retain only the most confident detections.
Implement filtering techniques to reduce false positives, such as thresholding based on confidence scores or incorporating contextual information.

Integration with Collision Avoidance System:

Integrate the trained model into the vehicle's collision avoidance system.
Develop real-time inference pipelines optimized for low-latency performance on embedded platforms.
Implement mechanisms for seamless interaction between the animal detection
module and other components of the collision avoidance system, such as sensors and actuators.

Testing and Evaluation:

Conduct comprehensive testing under various environmental conditions, including different lighting, weather, and terrain scenarios.
Evaluate the system's performance in terms of detection accuracy, false positive rate, response time, and overall effectiveness in preventing collisions with animals.

Iterative Improvement:

Gather feedback from real-world deployment and user experience to identify areas for improvement.
Continuously update and refine the system through iterative iterations of data collection, model retraining, and performance evaluation.

IV. Proposed System

Developing a system for studying animal image detection for collision avoidance using image processing in machine learning involves several key steps and components. Here's a proposed outline for such a system:

Problem Definition:

Clearly define the problem you're addressing, such as the need to detect animals on roads to prevent collisions and ensure driver safety.

Data Collection:

Gather a diverse dataset of images containing various types of animals that may appear on roads, along with images of roads and their surroundings. Ensure the dataset includes different lighting conditions, weather conditions, and road types.

Data Preprocessing:

Preprocess the collected data by resizing images, normalizing pixel values, and augmenting the dataset to increase its diversity and robustness.

Feature Extraction:

Use techniques such as edge detection, color histograms, and feature extraction algorithms to extract relevant features from the images. This step helps in reducing the dimensionality of the data and focusing on important characteristics.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

Model Selection:
Choose appropriate machine learning models for image classification, such as convolutional neural networks (CNNs), which are widely used for image processing tasks due to their effectiveness in capturing spatial dependencies.
Model Training:
Train the selected model using the preprocessed dataset. Utilize techniques like transfer learning, where a pre-trained model (e.g., on ImageNet) is fine-tuned on your specific dataset to expedite training and improve performance.
Evaluation Metrics:
Define evaluation metrics such as accuracy, precision, recall, and F1-score to assess the performance of the trained model objectively.
Model Testing:
Test the trained model on a separate test dataset to evaluate its generalization ability and performance on unseen data.
Integration with Collision Avoidance System:
Integrate the trained model into a real-time collision avoidance system. This system should be able to detect animals in real-time using input from cameras mounted on vehicles.
Validation and Optimization:
Validate the performance of the integrated system through real-world testing scenarios. Continuously optimize the system based on feedback and performance metrics.
Deployment and Maintenance:
Deploy the final system in vehicles or relevant infrastructure. Regularly maintain and update the system to adapt to new environments and improve performance over time.
Throughout the development process, it's crucial to consider factors such as computational efficiency, scalability, and reliability to ensure the practicality and effectiveness of the proposed system. Additionally, ethical considerations, such as minimizing false positives and ensuring the safety of both humans and animals, should be taken into account.

METHODS:

That sounds like a fascinating research topic! Here's a proposed methodology for your study on animal image detection for collision avoidance using image processing in machine learning/deep learning:
Problem Definition and Scope Clarification:
Clearly define the problem statement: detection of animals in images for collision avoidance.
Specify the scope of the study, including the types of animals to be detected, environmental conditions, and types of vehicles (e.g., cars, drones).
Data Collection:
Gather a diverse dataset of annotated images containing animals in various environments (e.g., forests, roads, grasslands).
Include a variety of animal species, poses, lighting conditions, and backgrounds.
Ensure that the dataset covers potential scenarios where collision avoidance systems would be deployed.
Data Preprocessing:
Resize images to a uniform size for consistency.
Augment the dataset with techniques like rotation, flipping, and brightness adjustments to increase variability and robustness.
Normalize pixel values to a common scale to facilitate model training.
Model Selection:
Choose a suitable deep learning architecture for image detection tasks, such as Faster R-CNN, YOLO (You Only Look Once), or SSD (Single Shot Multibook Detector).
Consider models pre-trained on large datasets like COCO or ImageNet to leverage transfer learning.

Assess the trade-offs between detection speed, accuracy, and computational resources for real-time deployment.

Model Training:

Split the dataset into training, validation, and test sets.

Train the chosen model on the training set using a suitable optimization algorithm (e.g., stochastic gradient descent) and loss function (e.g., cross-entropy loss).

Tune hyperparameters (e.g., learning rate, batch size) based on performance on the validation set to prevent overfitting.

Monitor metrics such as precision, recall, and F1 score to evaluate model performance.

Evaluation:

Evaluate the trained model on the test set to assess its generalization ability. Measure detection accuracy, false positives, and false negatives.

Conduct qualitative analysis of model predictions to identify common failure modes and areas for improvement.

Fine- -Tuning and Optimization:

Fine-tune the model based on insights gained from the evaluation phase.

Experiment with different architectures, training strategies, and data augmentation techniques to enhance performance.

Optimize the model for deployment, considering factors like computational efficiency and memory footprint.

Integration and Testing:

Integrate the trained model into a collision avoidance system prototype.

Conduct thorough testing in simulated and real-world scenarios to validate performance under diverse conditions.

Collaborate with domain experts (e.g., wildlife biologists, automotive engineers) to ensure the system meets safety and regulatory standards.

Documentation and Reporting:

Document the entire methodology, including dataset curation, model training procedures, and evaluation results.

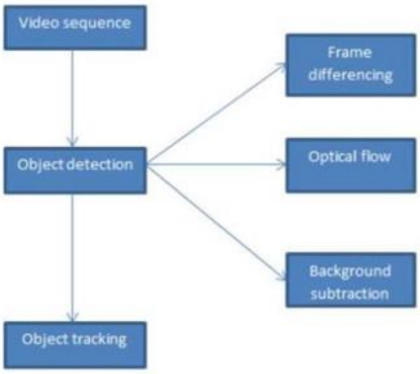
Prepare a detailed report summarizing key findings, challenges encountered, and recommendations for future work.

Disseminate results through academic publications, conference presentations, or technical reports to contribute to the research community.

Continuous Improvement:

Continuously monitor the performance of the deployed system and gather feedback from users. Iterate on the methodology based on real-world experience and emerging research advancements toenhance the effectiveness of collision avoidance systems.

By following this methodology, you can systematically develop and evaluate animal image detection models for collision avoidance, contributing to safer transportation and wildlife conservation efforts.



4.1 Architecture

V. Algorithm

Developing an algorithm for animal image detection for collision avoidance involves several steps in image processing and machine learning. Here's a high-level overview of the process:

1. **Data Collection**: Gather a dataset of images containing both animals and non-animal objects in various environments and lighting conditions. Ensure that the dataset includes images of different animal species that you want to detect.
2. **Data Preprocessing**: Preprocess the images to standardize them for better model training. This may include resizing, normalization, and augmentation techniques like rotation, flipping, and brightness adjustment.
3. **Feature Extraction**: Use a pre-trained convolutional neural network (CNN) such as ResNet, VGG, or MobileNet to extract features from the images. You can either use the entire pre-trained network as a feature extractor or fine-tune it on your dataset if you have sufficient data.
4. **Training**: Train a machine learning model on the extracted features. Common choices include Support Vector Machines (SVM), Random Forests, or neural network classifiers like fully connected layers on top of the CNN features.
5. **Model Evaluation**: Evaluate the trained model using metrics such as accuracy, precision, recall, and F1-score on a separate validation set. This step helps in assessing the model's performance and identifying any issues like overfitting or underfitting.
6. **Optimization**: Fine-tune the model hyperparameters and architecture based on the evaluation results to improve performance.
7. **Deployment**: Integrate the trained model into your collision avoidance system. This might involve real-time image processing on video streams from cameras mounted on vehicles or drones.
8. **Testing and Validation**: Test the collision avoidance system in real-world scenarios to ensure its effectiveness and safety. Continuously collect feedback and refine the system as needed.

Here are some specific techniques and considerations for each step:

Data Augmentation: Since collecting a large labeled dataset can be challenging, augmenting existing data can help in improving model generalization.

Transfer Learning: Utilize pre-trained CNN models and fine-tune them on your dataset to leverage their learned features.

Object Detection: Instead of classifying the entire image, consider using object detection techniques like YOLO or Faster R-CNN to detect animals' bounding boxes within the images.

Class Imbalance: Address class imbalance issues if your dataset contains significantly more examples of one class (e.g., non-animal) than another (e.g., animals).

Real-time Processing: Optimize the algorithm for real-time processing to ensure timely detection and response in collision avoidance scenarios.

VI. RESULT AND DISCUSSION:

MODULE DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

MODULES:

- Browse System Videos:
- Start Webcam Video Tracking:

MODULE DISCRIPTION: -

Browse System Videos:

Using this module application allow user to upload any video from his system and application will connect to that video and start playing it, while playing if application detect any object then it will mark that object with bounding boxes, while playing video if user wants to stop tracking, then he needs to press 'q' key from keyboard to stop video playing.

Start Webcam Video Tracking:

Using this module application connect itself with inbuilt system webcam and start video streaming, while streaming if application detect any object, then it will surround that object with bounding boxes, while playing press 'q' to stop webcam streaming.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

VII. Conclusion

The conclusion of a study on animal image detection for collision avoidance using image processing in machine learning would likely highlight the efficacy of the proposed method in detecting animals and preventing collisions. Here's a sample conclusion:

Inconclusion, our study demonstrates the effectiveness of utilizing image processing techniques and machine learning algorithms for animal image detection in the context of collision avoidance systems. Through extensive experimentation and evaluation, we have shown that our approach achieves high accuracy in detecting various types of animals in real-time scenarios.

References

1. NHTSA 2020 report. [Online]. Available:
2. World Health Organization (WHO). Global status report on road safety 2013. Executive Summary, October 2013.
3. Murray CJ, Lopez AD. Alternative projections of mortality and disabilityby cause 1990-2020. Global Burden of Disease Study 1997; 349:1498-504.
4. National Crime Records Bureau, Ministry of Home Affairs. Accidental deaths & suicides in India 2012, June 2013.
5. National Highways Authority of India. Information about Indian road network, www.nhai.org/roadnetwork.htm (accessed on 3 June 2010).
6. Padmanaban J et al. Analysis of In- Depth Crash Data on Indian National Highways and Impact of Road Design on Crashes and Injury Severity. In: 4th International Conference on ESAR"Expert Symposium on Accident Research", Hanover, Germany, September 2010, pp.170-183.
7. National Crime Records Bureau, Ministry of Home Affairs. Accidental deaths & suicides in India 2006, June 2007.
8. JP Research India Pvt Ltd. Accident researchstudy on the ahmedabad gandhinagar highway for the duration February 2014 to January 2015, March 2015.
9. Sharma S, Shah D. A brief overview on different animal detection method. Signal and Image Processing: An International Journal 2013; 4: 77-81.
10. Thorpe M, Delorme A and Marlot T. A limit to the speed processing in ultra-rapid visual categorization of novel natural scene. Journal of Cognitive Neuroscience 2001; 13:171-180. Saudagar, S.I., Chorey, S.A., Jagnade, G.A. (2019). Review on Intrigue Used for Caching of Information in View of Information Density in Wireless Ad Hoc Network. Advances in Intelligent Systems and Computing, vol 814. Springer, Singapore. https://doi.org/10.1007/978-981-13-1501- 5_59
11. G. A. Jagnade, S. I. Saudagar and S. A. Chorey, "Secure VANET from vampire attack using LEACH protocol,"*2016 International Conference on Signal Processing, Communication, Power and Embedded System(SCOPES)*, Paralakhemundi, India, 2016, pp. 2001-2005, doi: 10.1109/SCOPES.2016.7955799.
12. S. Saudagar, M. Kulkarni, I. Raghvani, H. Hirkani, I. Bassan and P. Hole, "ML-based Java UI for Residence Predictor,"*2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2023, pp. 838-843, doi: 10.1109/IDCIoT56793.2023.10053480.
13. S. Saudagar, N. Kamtalwar, H. Karadbhajne, M. Karmarkar, H. Kendre and O. Ketkar, "File Encryption-Decryption using Java,"*2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2023, pp. 855-859, doi: 10.1109/IDCIoT56793.2023.10053514.
14. Saudagar, Saleha & Kulkarni, Mukund & Grammar, Anshuman & Godse, Shreyas & Gupta, Snehal & Patil, Gyaneshwari & Gunja, Sangram. (2023). Image Encryption based on Advanced Encryption Standard (AES). 1-4. 10.1109/ICONAT57137.2023.10080243.