INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1811-1817

WILD ANIMAL ALERT DETECTION SYSTEM USING DEEP LEARNING

Mrs. V Vanaja[1], K Divyani[2], Veldurthi Geethika[3], Rapolu Varun[4]

[1]Assistant Professor, Information Technology, ACE Engineering College, India.

[2,3,4]Information Technology, ACE Engineering College, India.

ABSTRACT

Animal attacks pose a growing threat to rural populations and forestry workers. Surveillance cameras and drones are commonly used to monitor wild animal movements, but there is a need for an efficient system that can identify animal types and provide location data.This information can then trigger alerts to ensure the safety of people and foresters. Although computer visionand machine learning techniques are often employed foranimal detection, they tend to be costly and complex, leading to suboptimal results. We introduced a Hybrid Visual Geometry Group (VGG)−19 and BidirectionalLong Short-Term Memory (Bi-LSTM) network designed to detect animals and generate alerts. These alerts are sent via email to the local forest office, facilitating immediate action. The proposed model significantly enhances performance, offering a dependable solution for delivering precise animal information and safeguarding human lives.

1. INTRODUCTION

In this work, a dataset was created by gathering authentic CCTV footage of both common and uncommon animal behaviors, along with wildlife videos from YouTube. Optimize inference latency and streamline video classification tasks to promote smart cities' sustainability. For action identification and video classification, the findings demonstrate that Image Classifiers are equivalent to the current SOTA 3D networks in their ability to recognize and discriminate between various animal activities.

The models used were Deep Convolutional Neural Networks (DCNN or CNN), Visual Geometry Group (VGG), and Alex Net. A detection model was developed that can detect animal activities with generalizability on the form and dynamic features of animals. The model's generalization capability is enhancedby emphasizing animal elements and dynamic characteristics obtained from the adjacent frames.

From theinput feature map, spatial properties may be extracted using a 3D CNN framework. The datasets used are Real- Life Wildlife Situations (RLWS) and Wildlife Flow. Thesedatasets test the generalization capacity of the HD Net using contrast tests. This generalization capacity is confirmed by comparing it to other classical animal detection algorithm.

Alert messaging systems designed for wild animal detection represent a paradigm shift in how we approach human-wildlife conflicts. These systems harness the powerof modern technology, particularly artificial intelligence and sensor networks, to provide real-time alerts and warnings to individuals or communities at risk.

and other sources, these systems can detect the presence of wildlife and promptly notify relevant stakeholders, enabling timely and informed decision-making. An increasing area of land surface has been transformed by human action, altering wildlife populations, habitat, and behavior. More seriously, many wild species on Earth have been driven to extinction, and many species are introduced into new areas where they can disrupt both natural and human systems. Monitoring wild animals, therefore, is essential as it provides researchers with evidence to inform conservation and management decisions tomaintain diverse, balanced, and sustainable ecosystems in the face of those changes.

2. OBJECTIVES

The objective encompasses developing an algorithm not only for automated animal detection and classification but also for integrating an alert mechanism via email.

This multifaceted objective addresses the challenges posed by the diverse array of animal species, labor-intensive monitoring methods, and the pressing need for timely response to potential human-wildlife conflicts and wildlife-related crimes. The algorithm is designed to accurately identify and categorize animals in images capturedby camera traps or surveillance systems, leveraging deep learning techniques for robust performance. Simultaneously, theincorporation of email notification functionality adds a crucial layer of real-time communication, enabling relevant stakeholders to receive immediate alerts based on predefined criteria.

By seamlessly integrating email alerts into the algorithm, the objective aims to enhance the efficiency andeffectiveness of wildlife monitoring efforts, empowering conservationists, park rangers, and other stakeholders to respond promptly and proactively to emerging wildlife situations.

3. METHODOLOGY

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1811-1817

To address the growing threat of animal attacks on rural populations and forestry workers, we developed a Hybrid Visual Geometry Group (VGG)−19 and Bidirectional Long Short-Term Memory (Bi-LSTM) network to detect animals and generate alerts. Our methodology involves collecting authentic CCTV footage and wildlife videos, preprocessing this data into frames, and labeling them for different animal types and behaviors. The VGG-19 network, pre-trained on ImageNet, extracts spatial features from these frames, which are then fed into a Bi-LSTM network to capture temporal dependencies and movement patterns. The combined model is trained on a dataset using data augmentation, with an adaptive optimizer and early stopping to prevent overfitting. Upon detecting an animal, the system generates an alert containing the animal type, behavior, sand location, which is sent via email to the local forest office. Performance is evaluated using accuracy, precision, recall, and F1-score, and the model's effectiveness is validated against baseline models. The deployed system is

By leveraging data from cameras drones, acoustic sensors, continuously monitored and updated to ensure reliable detection and alerting, thereby enhancing the safety of rural communities and forestry personnel.

4. LITERATURE SURVEY

Fang, Y., et al. discussed a technique to move animaldetection by taking benefit of global patterns of pixel motion. In the dataset, where animals make obvious movement against the background, motion vectors of every pixel were estimated by applying optical flow techniques. Acoarse segmentation then eliminates most parts of the background via applying a pixel velocity threshold. Using the segmented regions, another threshold was used to filter out negative candidates, which could belong to the background.

Parham, J., et al. proposed a 5-component detection pipeline to utilize in a computer vision-based animal recognition system. The result of this approach was a collection of novel annotations of interest (AoI) with species and viewpoint labels. The concept of this approach was to increase the reliability and automation of animal censusing studies and to offer better ecological information to conservationists.

Gupta, P., & Verma, G. K. proposed a technique for detection of visual wild animals in images by dictionary learning. Discriminative Feature-oriented DictionaryLearning was utilized for learning discriminative features ofpositive images, that have animals present in positive class, in addition to of negative images that do not have animals present in that class. The system was created dictionaries that were class-specific and was capable of automatic feature extraction by example training image samples. The proposed approach was learned these dictionaries through positive (animal class and negative background class) sparse representation of image samples.

5. PROPOSED SYSTEM

The proposed Hybrid VGG-19+Bi-LSTM model representsa cutting-edge advancement in the field of deep learning, particularly in the domain of image recognition and classification. By integrating two powerful neural network architectures, VGG-19 and Bi-LSTM, this model leverages the strengths of both approaches to achieve superior accuracy in identifying and classifying objects within images. The meticulous fine-tuning of hyperparameters ensures that the model operates at peak performance, maximizing its ability to discern subtle features and patterns within the data. This optimization process is crucial for enhancing recognition accuracy and overall model effectiveness.

6. HARDWARE AND SOFTWARE REQUIREMENTS

6.1 HARDWARE REQUIREMENTS

- System : Intel(R) Core(TM) i3-7020UCPU @2.30GHz
- Hard Disk : 1 TB.
- Input Devices : Keyboard, Mouse
  - Ram : 4 GB
  - SOFTWARE REQUIREMENTS
  - Operating system :Windows XP/7/10.
  - Coding Language :python
  - Tool :VS Studio

7. PACKAGES USED

os: Operating system interaction.

cv2: Computer vision and image processing.

numpy: Numerical computing and array operations.

sklearn: Machine learning algorithms and tools.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1811-1817

Keras: High-level neural networks API.

Matplotlib.pyplot: Data visualization.

TensorFlow. keras: Deep learning framework.

seaborn: Statistical data visualization.

8. TECHNOLOGY DESCRIPTION

Anaconda is an open-source distribution of the Python and R programming languages, designed specifically for scientific computing, data science, and machine learning. It simplifies the setup and management of Python environments by including pre-installed packages and libraries commonly used in these domains. Python, a high-level, integrated programming language, emphasizes an object-oriented approach to help programmers write clear and logical code for projects of all sizes. Flask is a lightweight and flexible web framework for Python, enabling developers to quickly create web applications and APIs with minimal boilerplate code. Jupyter, an open-source web application, allows users to create and share documents containing live code, equations, visualizations, and narrative text, supporting various programming languages such as Python, R, and Julia, and is widely used for data exploration and visualization.

Visual Studio Code (VSCode), a free and open-source code editor developed by Microsoft, offers built-in support for various programming languages, debugging tools, version control integration, and an extensive library of extensions, making it a popular choice among developers. MySQL, an open-source relational database management system (RDBMS), uses SQL (Structured Query Language) for querying and managing data. It is known for its reliability, scalability, and performance, making it widely used in web development for storing and retrieving structured data. Lastly, HTML (Hypertext Markup Language), CSS(Cascading Style Sheets), and JavaScript are core technologies for building websites and web applications. HTML creates the structure and content of web pages, CSS handles the styling and formatting, and JavaScript adds interactivity and dynamic behavior, forming the foundation of modern web development.

9. SOURCE CODE

```
import os import cv2
import numpy as np
from sklearn.model_selection import train_test_splitfrom tensorflow import keras
from keras.utils import to_categoricalfrom keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten,MaxPooling2D
from keras.layers import Convolution2D as Conv2Dimport cv2
import numpy as np train_path = 'dataset/train'test_path = 'dataset/test'
activities = os.listdir(train_path)num_classes = len(activities) train_frames = []
train_labels = []
for i, activity in enumerate(activities): activity_path = os.path.join(train_path, activity) for video_file in os.listdir(activity_path):
video_path = os.path.join(activity_path, video_file)cap = cv2.VideoCapture(video_path)
while True:
ret, frame = cap.read()if not ret:
break
resized_frame = cv2.resize(frame, (64, 64))train_frames.append(resized_frame) train_labels.append(i)
cap.release()
X_train = np.array(train_frames)y_train = np.array(train_labels) test_frames = []
test_labels = []
for i, activity in enumerate(activities): activity_path = os.path.join(test_path, activity) for video_file in os.listdir(activity_path):
video_path = os.path.join(activity_path, video_file)cap = cv2.VideoCapture(video_path)
while True:
ret, frame = cap.read()if not ret:
break
```

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1811-1817

```
resized_frame = cv2.resize(frame, (64, 64))test_frames.append(resized_frame) test_labels.append(i)
cap.release()print("done")
X_test = np.array(test_frames)y_test = np.array(test_labels) # Normalize pixel values X_train = X_train / 255.0 X_test = X_test / 255.0
# Convert labels to one-hot vectors
y_train = to_categorical(y_train, num_classes)y_test = to_categorical(y_test, num_classes)
# Step 2: Define the model architecturemodel = Sequential()
model.add(Conv2D(32, kernel_size=(3, activation='relu',input_shape=(64, 64, 3)))
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, kernel_size=(3, activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Flatten())
model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(num_classes, activation='softmax')) # Step 3: Compile and train the model model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32,validation_data=(X_test, y_test))
# Step 4: Save the trained modelmodel.save('ani.h5')
from sklearn.metrics import accuracy_score, precision_score,recall_score, f1_score
from keras.models import load_modelmodel = load_model('ani.h5') y_train_pred = model.predict(X_train)
y_train_pred = np.argmax(y_train_pred, axis=1)y_train_true = np.argmax(y_train, axis=1)
# Calculate accuracy
train_accuracy = accuracy_score(y_train_true, y_train_pred) train_precision = precision_score(y_train_true, y_train_pred,average='macro')
train_recall = recall_score(y_train_true, y_train_pred,average='macro')
train_f1 = f1_score(y_train_true, y_train_pred,average='macro')
print(f'Training Accuracy: {train_accuracy * 100:.2f}%') print(f'Training Precision: {train_precision:.2f}') print(f'Training Recall: {train_recall:.2f}') print(f'Training F1-score: {train_f1:.2f}')
from sklearn.metrics import classification_report# Load the trained model
model = load_model('a.h5') y_train_pred = model.predict(X_train)
y_train_pred = np.argmax(y_train_pred, axis=1)y_train_true = np.argmax(y_train, axis=1)
train_report = classification_report(y_train_true, y_train_pred,target_names=activities)
print("Training Classification Report:")print(train_report)
from sklearn.metrics import confusion_matriximport seaborn as sns
import matplotlib.pyplot as plt
from tensorflow.keras.models import load_modelimport numpy as np
model = load_model('animal.h5') y_train_pred = model.predict(X_train)
y_train_pred = np.argmax(y_train_pred, axis=1)y_train_true = np.argmax(y_train, axis=1)
cm = confusion_matrix(y_train_true, y_train_pred)class_labels = ['bear','elephant','lion','tiger'] plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')plt.xlabel('Predicted Labels')
plt.ylabel('True Labels') plt.xticks(np.arange(len(class_labels)), class_labels,rotation=45)
plt.yticks(np.arange(len(class_labels)), class_labels,rotation=0)
plt.title('Confusion Matrix')plt.show()
#Testing import cv2
import numpy as np
from keras.models import load_model model = load_model('animal.h5') activities = ['bear','elephant','lion','tiger'] test_video_path = 'p (4).mp4'
 cap = cv2.VideoCapture(test_video_path)frame_width = int(cap.get(3)) frame_height = int(cap.get(4))
fps = cap.get(cv2.CAP_PROP_FPS)
output_video_path = 'Lion1.mp4' # Replace with thedesired output video path
fourcc = cv2.VideoWriter_fourcc(*'MJPG')
```

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1811-1817

out = cv2.VideoWriter(output_video_path, fourcc, fps,(frame_width, frame_height))
while True:
ret, frame = cap.read()if not ret:
break cap.release() out.release() **Sendmail.py** import smtplib
def sendmail(email,msg,loc):TO = email
SUBJECT = 'Prediction:'
TEXT ='Message:'+msg+'location:'+locprint(TEXT)
gmail_sender = "animaldetection98@gmail.com"gmail_passwd = "wlfuwbipnfmehpmr"
server = smtplib.SMTP('smtp.gmail.com', 587)server.ehlo()
server.starttls()
server.login(gmail_sender, gmail_passwd)BODY = '\r\n'.join(['To: %s' % TO,
'From: %s' % gmail_sender,'Subject: %s' % SUBJECT, '', TEXT])
rver.sendmail(gmail_sender, [TO], BODY)print ('email sent')
except: print ('error sending mail')server.quit()

10. OUTPUT

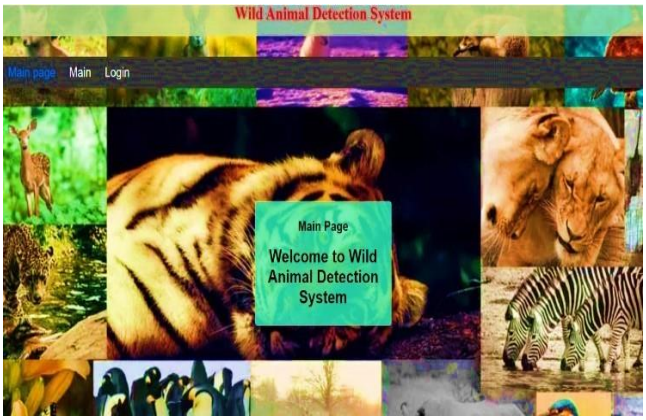

Figure: 10.1 Web Application Main Page

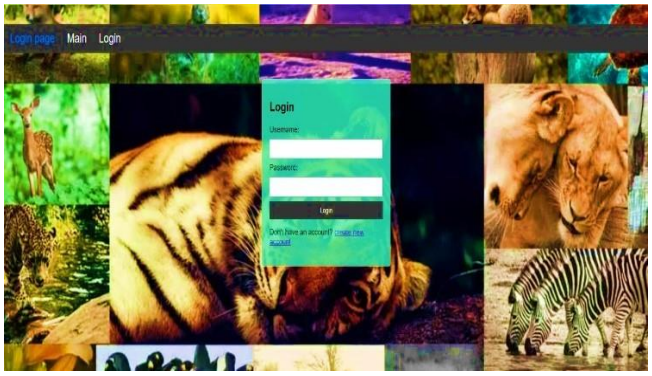

Figure: 10.2 Login Page

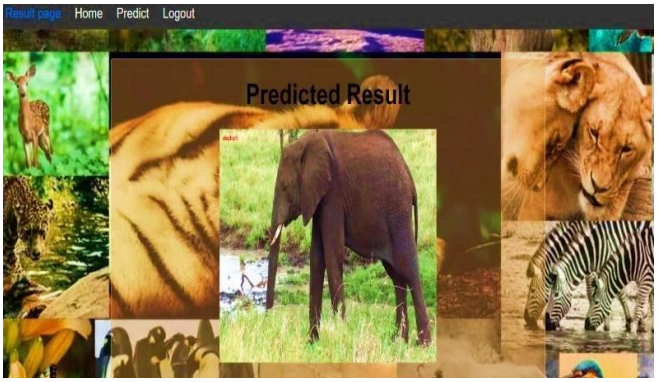

Figure: 10.3 File Uploading Page

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1811-1817

Figure: 10.4 Animal Detecting Page

Figure: 10.5 Alert Message

11. CONCLUSION

In conclusion, the proposed automated animal detection and classification algorithm leveraging deep learning techniques represents a significant advancement in wildlife monitoring technology. By addressing the limitations of existing manual observation and camera trap methods, such as labor intensity, limited coverage, and human error, the proposed system offers improved efficiency, accuracy, and coverage. However, while the algorithm itself presents substantial benefits, the absence of the email alert integration feature in this project's scope is acknowledged. Despite this, the core functionality of the algorithm lays a strong foundation for future enhancements, including the integration of an alert system to provide timely notifications to stakeholders about significant wildlife events. Overall, this project lays the groundwork for a comprehensive and proactive approach to wildlife monitoring, contributing to better conservation practices and human-wildlife conflict mitigation efforts.

This project presents promising avenues for expansion, notably through the integration of an alert system to provide timely notifications about significant wildlife events. This additionwould enhance the algorithm's utility, enabling proactive conservation measures and human-wildlife conflict mitigation. Moreover, further advancements could include real-time monitoring capabilities, multi-species detection, behavioral analysis functionalities, and cross-domain applications. Collaborative efforts with wildlife conservation organizations and research institutions would facilitate field testing and refinement of the algorithm in diverse environmental conditions, ultimately contributing to more effective wildlife conservation practices and fostering harmonious coexistence between humans and wildlife.

12. FUTURE SCOPE

Future research and development efforts in continued research and development efforts can lead to further improvements in the accuracy and precision of deep learning models for wild animal detection. Advanced techniques such as attention mechanisms, ensemble learning, and multi-modal fusion could be explored to enhance detection performance. Future research trends we identified are video-based detection, very high- resolution satellite image-based detection, multiple species detection, new annotation methods, and the development of specialized network structures and large foundation models. The need for real-time detection and future systems could be designed to detect multiple species of wild animals simultaneously,allowing for more comprehensive monitoring and conservation efforts. This would require the development of robust models capable of accurately identifying and localizing diverse species within complex natural environments. In addition to detecting the presence of wild animals, future systems could analyze animal behavior based on captured images or video footage. This could provide valuable insights into wildlife ecology, including movement patterns, habitat preferences, and interactions with humans and other species. Advances in edge

computing and Internet of Things (IoT) technology could enable the deployment of distributed detection systems capable of processing and analyzing data directlyat the point of capture.

This would reduce latency, bandwidth requirements, and dependence on centralized infrastructure, making it feasible to deploy detection systems in remote or resource-constrained environments. Continuous monitoring of wild animal populations over extended periods could enable trend analysis and predictive modeling of wildlife dynamics. By analyzing historical data and detecting long-term trends, conservationists could identify emerging threats, track population trends, and inform adaptive management strategies.

13. REFERENCES

[1] Animal Detection using Faster R-CNN for Wildlife Monitoring by S. Kumar and R. Singh, published in the Journal of Advanced Research in Dynamical and Control Systems in 2018.

[2] Terven, J., & Cordova-Esparza, D. (2023). Acomprehensive review of YOLO: From YOLOv1 toYOLOv8 and beyond. arXiv preprint arXiv:2304.00501.

[3] Kupyn, O., & Pranchuk, D. (2019). Fast and efficientmodel for real-time tiger detection in the wild. In Proceedings of the IEEE/CVF International Conferenceon Computer Vision Workshops (pp. 0-0).

[4] Tan, M., Chao, W., Cheng, J. K., Zhou, M., Ma, Y., Jiang, X., ... & Feng, L. (2022). Animal detection and classification from camera trap images using differentmainstream object detection architectures. Animals, 12(15), 1976.

[5] Liu, B., & Qu, Z. (2023). AF-TigerNet: A lightweight anchor-free network for real-time Amur tiger (Panthera tigris altaica) detection. Wildlife Letters, 1(1), 32-41.

[6] "Real-Time Wildlife Detection using YOLOv3" by N. Nair et al., published in the International Conference on Machine Learning and Data Science in 2021.