A Project report on

SAFEGUARDING AGRICULTURE: USING DEEP LEARNING TECHNIQUES TO PREVENT ANIMAL INVASION

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

K. MOULIKA	**(204G1A0559)**
C. BHAVANA	**(204G1A0523)**
U. ANUSHA	**(204G1A0514)**
T. CHANDANA	**(204G1A0526)**

Under the Guidance of

Mr. C. Lakshminatha Reddy **M.Tech., (Ph.D)**

Assistant Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701

2023-2024

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the Project report entitled SAFEGUARDING AGRICULTURE: USING DEEP LEARNING TECHNIQUES TO PREVENT ANIMAL INVASION is the bonafide work carried out by **K. Moulika, C. Bhavana, U. Anusha, T. Chandana** bearing Roll Number **204G1A0559, 204G1A0523, 204G1A0514, 204G1A0526** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023 - 2024.

Project Guide

Mr. C. Lakshminatha Reddy M.Tech., (Ph.D)

Assistant Professor

Head of the Department

Mr. P. Veera Prakash M.Tech., (Ph.D)

Assistant Professor

Date:

External Examiner

Place: Rotarypuram

DECLARATION

We, Ms. K. Moulika bearing reg no: 204G1A0559, Ms. C. Bhavana bearing reg no: 204G1A0523, Ms. U. Anusha bearing reg no: 204G1A0514, Ms. T. Chandana bearing reg no: 204G1A0526 student of **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, Rotarypuram, hereby declare that the dissertation entitled **"SAFEGUARDING AGRICULTURE: USING DEEP LEARNING TECHNIQUES TO PREVENT ANIMAL INVASION"** embodies the report of my project work carried out by me during IV year Bachelor of Technology under the guidance of **Mr. C. Lakshminatha Reddy,** Assistant Professor, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

K. MOULIKA	Reg no: 204G1A0559
C. BHAVANA	Reg no: 204G1A0523
U. ANUSHA	Reg no: 204G1A0514
T. CHANDANA	Reg no: 204G1A0526

VISION & MISSION OF THE INSTITUTION

Vision:

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

Mission:

- Continually enhance the quality of physical infrastructure and human resources to evolve in to a center of excellence in engineering education.
- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.
- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever changing requirements of the industry.
- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

VISION & MISSION OF THE DEPARTMENT OF CSE

Vision:

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

Mission:

DM 1: Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that I would like to express my in debted gratitude to my Guide **Mr. C. Lakshminatha Reddy, Assistant Professor, Computer Science & Engineering,** who has guided me a lot and encouraged me in every step of the project work. I thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

I express my deep felt gratitude to **Mr. C. Lakshminatha Reddy, Assistant Professor and Mr. M. Narasimhulu, Assistant Professor, Project Coordinators** for their valuable guidance and unstinting encouragement enabled me to accomplish my project successfully in time.

I am very much thankful to **Mr. P. Veera Prakash, Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

I wish to convey my special thanks to **Dr. G. Bala Krishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing my project work. Not to forget, I thank all other faculty and non- teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

I also express my sincere thanks to the Management for providing excellent facilities**.**

Finally, I wish to convey my gratitude to my family who fostered all the requirements and facilities that I need.

Project Associates

204G1A0559

204G1A0523

204G1A0514

204G1A0526

ABSTRACT

Crop damage caused by animal attacks is one of the major threats in reducing the crop yield. Due to the expansion of cultivated land into previous wildlife habitat, crop raiding is becoming one of the most antagonizing human-wildlife conflicts. Farmers in India face serious threats from pests, natural calamities & damage by animals resulting in lower yields. Traditional methods followed by farmers are not that effective and it is not feasible to hire guards to keep an eye on crops and prevent wild animals. Since safety of both human and animal is equally vital, it is important to protect the crops from damage caused by animal as well as divert the animal without any harm. Thus, in order to overcome above problems and to reach our aim, we use machine learning to detect animals, entering into our farm by using deep neural network concept, a division in computer vision.

This project describes a novel, end to end design of a Deep Learning based, distributed, cost-effective and scalable models tracking system for animal intrusion detection, proposed to be installed along the borders of vulnerable areas. In this proposed work, we provide the software by using deep learning models CNN, VGG-16, we detect the entry of animals and send the mail to the users regarding the animal information.

Keywords: *Deep Learning, CNN, VGG-16, VGG19.*

CONTENTS

Page No.

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolution Neural Network
VGG	Visual Geometric Group
CM	Confusion Matrix
ACC	Accuracy Machine Learning
ReLU	Rectified Linear Unit
CV	Computer Vision

CHAPTER 1

INTRODUCTION

In agriculture, one of the major social problems that is existing in the present is the damaging of the crops by the wild animals. Wild animal intrusion has always been a persisting problem to the farmers. Some of the animals that act as a threat to the crops are deer, wild boar, moles, elephants, monkeys and others. These animals may feed on crops and also run around the field in the absence of farmer and thus cause damage to those crops. This may in turn results in significant loss in the yield and will cause additional financial protection in order to deal with the aftermath of the damage.

Every farmer, while utilizing his production, should also be aware of the fact that animals are also existing in the same place and that they need to be protected from any probable suffering. This problem must be attended immediately and an effective solution must be created and accomplished. Thus, this project aims to address this problem.

The conflicts between human and animal become a major problem in the agriculture field and in the forest zone which leads to human life in danger and also losing a huge quantity of resources.

Deforestation, lack of natural prey and habitat loss has forced wild animals to live closely with human settlement prey on domestic livestock. So animals are started attacking humans for theirs food. Compared with other tiger, elephant entry are in extreme form of conflict and responsible for killing human's lives in India. Due to human [19] and animal conflict and increase in population humans started demolishing the forest for their existence these leads to harm animals and its habitats. Losing subsistence and dryness makes them raving mad and pretend to attack crops, livestock, sometimes human and farming lands. Usually, farmers use the electrical fence to protect the field from animals which cause electrocution with cramp makes them behave unusual manner. The safety of animal and human is identically important. vision is a machine learning software library called OpenCV is available for free use.

In this VGG-16 algorithm, we can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained

network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

A Safeguarding Agriculture using Deep Learning techniques to prevent animal invasion is Proposed of having features which may reduce cost, robustness, reliable, easy access by farmers, and remote monitor with lower energy consumption.

1.1 Problem Statement

Wildlife conservation faces a significant challenge in preventing human-animal conflicts and safeguarding both wildlife and human interests. Incidents of animals intruding into human settlements or agricultural areas can lead to property damage, crop destruction, and pose risks to both humans and animals. Traditional methods of monitoring and preventing such intrusions often fall short in terms of efficiency and accuracy.

This project aims to address these challenges by developing an Animal Intrusion Detection System (AIDS) utilizing deep learning techniques. The primary goal is to create a robust and reliable system that can accurately identify and classify instances of animals entering human-inhabited or restricted areas.

1.2 Objectives

To accomplish the project's purpose, the following particular objectives have been established.

i. Create and train deep learning models capable of analyzing images or video feeds for the detection of animal intrusions
ii. Integrate the developed system with existing alert systems to enable timely notifications to relevant authorities or stakeholders when an intrusion is detected

1.3 Scope of the Project

The following are the boundaries that have established in the proposed system which defines scope.

i. Crop damage will be reduced.
ii. An User can get the email notification when the animal is entered into the farm.
iii. There will be no harm to both the animals and humans.

CHAPTER 2
LITERATURE SURVEY

[1] Hardiki Patil and Namrata Ansari, animals straying into human settlements in search for food cause conflicts resulting in injury to humans, animals, or both. Such human animal conflicts can be largely reduced through a fully automated monitoring system to detect animal transgressions. This paper describes a novel, end to end design of a Computer Vision based distributed, cross camera tracking system (a Digital Border) for animal intrusion detection based on deep learning networks. Apart from generating alerts on detection of animal intrusion, the system also reports valuable information such as the approximate location, last known direction of movement and the type and count of the intruders. Animal re-identification is used to track an individual across multiple cameras. The MobileNetv2- SSD model is used to detect presence of animals. The prototype is tested using three animal species – tigers, jaguars and elephants, and achieves detection accuracy of 80%, 89.47% and 92.56% respectively, while operating at 2-3 frames per second. The salient features of the proposed design include high accuracy of animal detection, cost-effectiveness, efficient animal tracking through re- identification and scalability to cover large installations.

[2] DR.R.S. Sabeenian, N. Deivanai, B. Mythili, have clearly explained that due to the expansion of cultivated land into previous wildlife habitat, crop raiding is becoming one of the most antagonizing human-wildlife conflicts. Farmers in India face serious threats from pests, natural calamities & damage by animals resulting in lower yields. Traditional methods followed by farmers are not that effective and it is not feasible to hire guards to keep an eye on crops and prevent wild animals. Since safety of both human and animal is equally vital, it is important to protect the crops from damage caused by animal as well as divert the animal without any harm. Thus, in order to overcome above problems and to reach our aim, we use machine learning to detect animals, entering into our farm by using deep neural network concept, a division in computer vision. In this project, we will monitor the entire farm at regular intervals through a camera which will be recording the surrounding throughout the day. With the

help of a machine learning model, we detect the entry of animals and we play appropriate sounds to drive the animal away. This report specifies various libraries and concepts of convolutional neural networks used to create the model.

[3] K. Bala Krishna, **Fazil Mohammad, C. R. Ullas, C. M. Hema and S. K. Sonakshi** have proposed model presents the development of the Internet of Things and Machine learning technique-based solutions to overcome this problem. Raspberry Pi runs the machine algorithm, which is interfaced with the ESP8266 Wireless Fidelity module, Pi Camera, Buzzer, and LED. Machine learning algorithms like Region-based Convolutional Neural Network and Single Shot Detection technology plays an important role to detect the object in the images and classify the animals. The experimentation reveals that the Single Shot Detection algorithm outperforms than Region-based Convolutional Neural Network algorithm. Finally, the Twilio API interfaced software decimates the information to the farmers to take decisive action in their farm field.

[4] Vidhyalatha T, Y Sreeraman, E Purushotham Employing gatekeepers to keep an eye on crops and deter wild animals is not a practicable solution, contrary to conventional wisdom used by ranchers. Since the safety of both humans and animals is crucial, it is essential to protect the crops from animal-caused damage and to reroute the animal with little chance of mischief. In order to overcome these problems and get to our point, we employ artificial intelligence (AI) to detect animals as they enter our ranch using a division of computer vision known as a deep brain network. This paper promotes the ability to identify organisms in the natural world. Since there are so many different types of species, it might be difficult to physically discern between them. According on the amount of activated vibration sensors, a camera trap takes a picture, and Latest API is used to determine whether a monster is around, according to Suganthi et al. Sailesh and colleagues utilize a camera attached to a Raspberry Pi device that continually takes pictures and looks to see if any of them have elephants in them.

[5] Faseeha M. M, and Ambily Jacob [5] On account of farmlands or rural terrains reconnaissance is vital to keep unapproved individuals from accessing the region as well as to shield the region from animals. The proposed system uses YOLOv3 algorithms to detect real-time object. YOLOv3 is a real-time object discovery algorithm that distinguishes explicit objects in recordings, live feeds, or pictures. Consequences be damned uses highlights learned by a deep convolutional neural network to recognize an object. YOLOv3 is a superior form of YOLO and YOLOv2. YOLO is carried out utilizing the Keras or OpenCV deep learning libraries. For developing this system they have to collect some dataset. Here our datasets are images of wild animals of 8 different classes. After collecting the dataset it will go through an image pre-processing step, it is also called as annotation and we will get our final dataset for training. While testing the system it will check an image and look for matching class and it will predict an output which have an accuracy near to 1. After that it will send these predicted class name to firebase which is a real-time database, further the class name will be notified to the user's device

CHAPTER 3

METHODOLOGY

As you can see, each image are presented in the matrix formats, which are made up of rows and columns. The pixel is an image's fundamental building block. A group of pixels make up an image. These are all little squares. We may build the entire image by arranging them side by side. The smallest amount of information that can be present in an image is a single pixel. Every image has pixels with values ranging from 0 to 255.

Each pixel is composed of Three values are R, G, and B, which are the basic colours red, green, and blue. The combination of these three basic colours will create all these colours here in the image so we conclude that a single pixel has three channels, one channel for each one of the basic colours.

3.1 Deep Learning

Deep learning is the branch of machine learning which is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data. The reason deep learning made such a splash is the very fact that it allows us to phrase several previously impossible learning problems as empirical loss minimization via gradient descent, a conceptually super simple thing.

Basic Terminology:

- Dataset: A set of data examples, which contain features important to solving the problem.
- Features: Important pieces of data that help us understand a problem. These are fed into a Deep Learning algorithm to help it learn.
- Model: The representation (internal model) of a phenomenon that a Deep Learning algorithm has learnt. It learns this from the data it is shown during training. The model is the output you get after training an algorithm. For example, a decision tree algorithm would be trained and produce a decision tree model.

Types of Deep Learning:

There are multiple forms of Deep Learning. Three main types of deep learning methods that are popularly used: Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN). Each form of Deep Learning has differing approaches, but they all follow the same underlying process and theory.

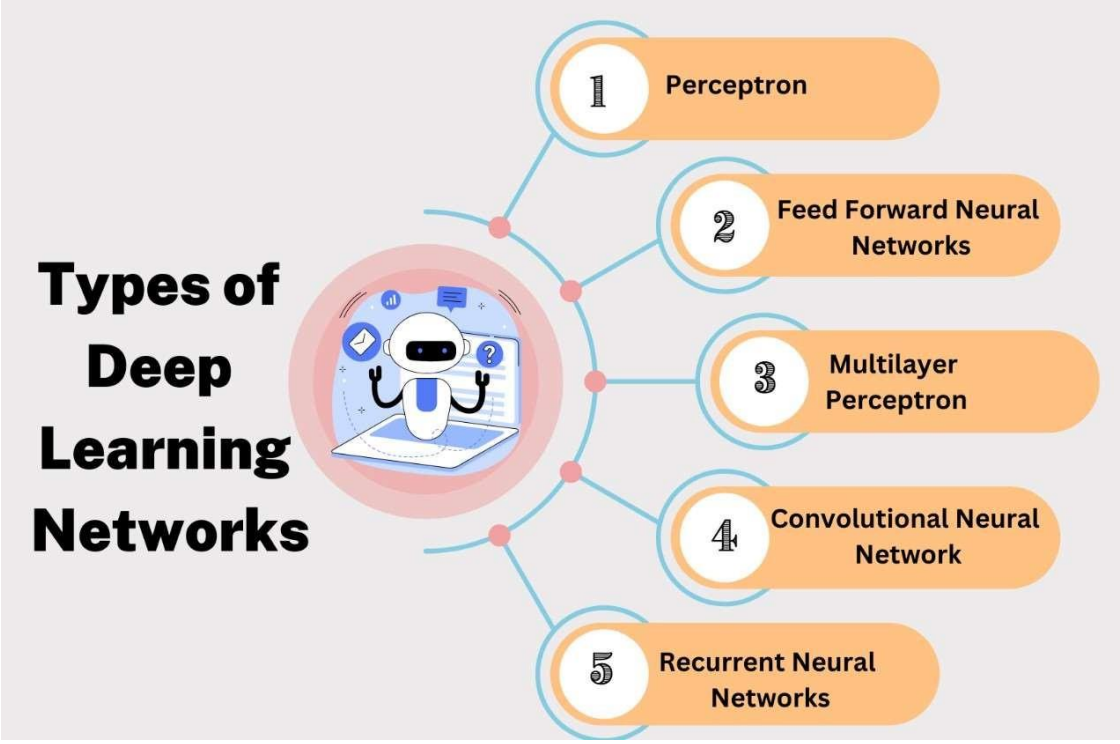


Fig 3.1: Types of Deep Learning

Multi-Layer Perceptron:

A multilayer perceptron (MLP) is a class of a feedforward artificial neural network (ANN). MLPs models are the most basic deep neural network, which is composed of a series of fully connected layers. Today, MLP machine learning methods can be used to overcome the requirement of high computing power required by modern deep learning architectures. Each new layer is a set of nonlinear functions of a weighted sum of all outputs (fully connected) from the prior one.

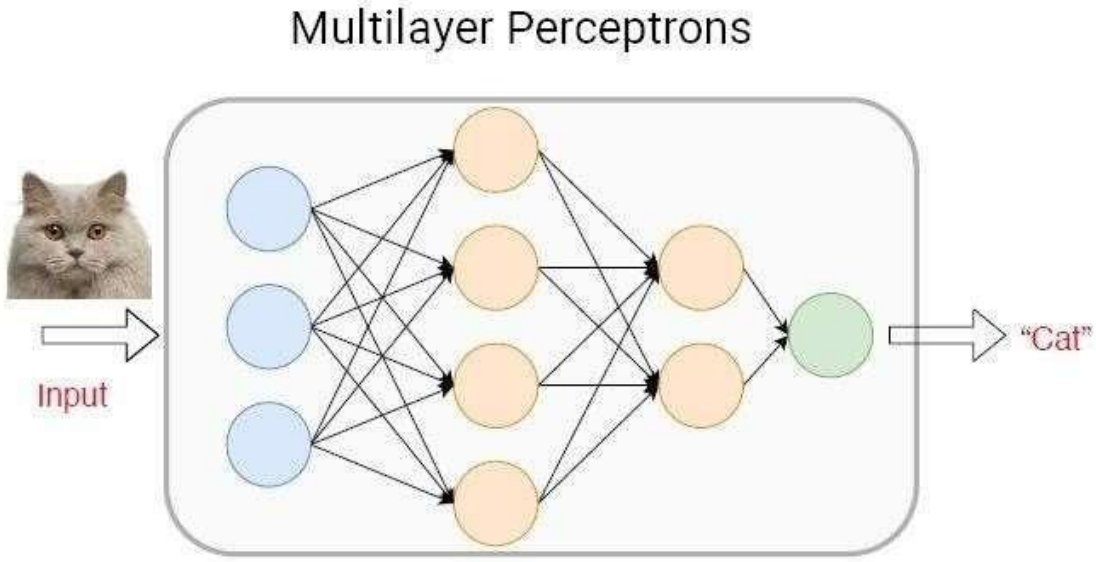


Fig 3.2: Process of Multilayer Perceptron

Convolutional Neural Networks:

A convolutional neural network (CNN, or ConvNet) is another class of deep neural networks. CNNs are particularly useful for finding patterns in images to recognize objects, classes, and categories. CNNs are most commonly employed in computer vision. Given a series of images or videos from the real world, with the utilization of CNN, the AI system learns to automatically extract the features of these inputs to complete a specific task, e.g., image classification, face authentication, and image semantic segmentation. Different from fully connected layers in MLPs, in CNN models, one or multiple convolution layers extract the simple features from input by executing convolution operations. Each layer is a set of nonlinear functions of weighted sums at different coordinates of spatially nearby subsets of outputs from the prior layer, which allows the weights to be reused.

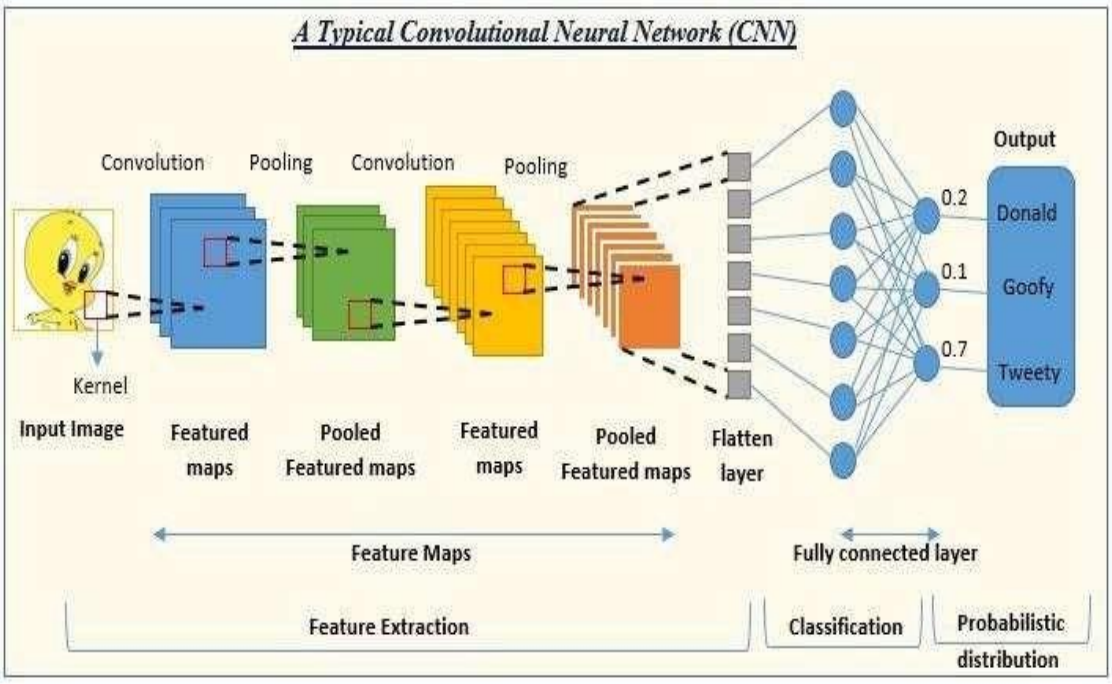


Fig 3.3: Convolution Neural Networks

Recurrent Neural Networks:

A recurrent neural network (RNN) is another class of artificial neural networks that use sequential data feeding. RNNs have been developed to address the time-series problem of sequential input data. The input of RNN consists of the current input and the previous samples. Furthermore, each neuron in an RNN owns an internal memory that keeps the information of the computation from the previous samples.

In RNNs, each subsequent layer is a collection of nonlinear functions of weighted sums of outputs and the previous state. Thus, the basic unit of RNN is called "cell", and each cell consists of layers and a series of cells that enables the sequential processing of recurrent neural network models.

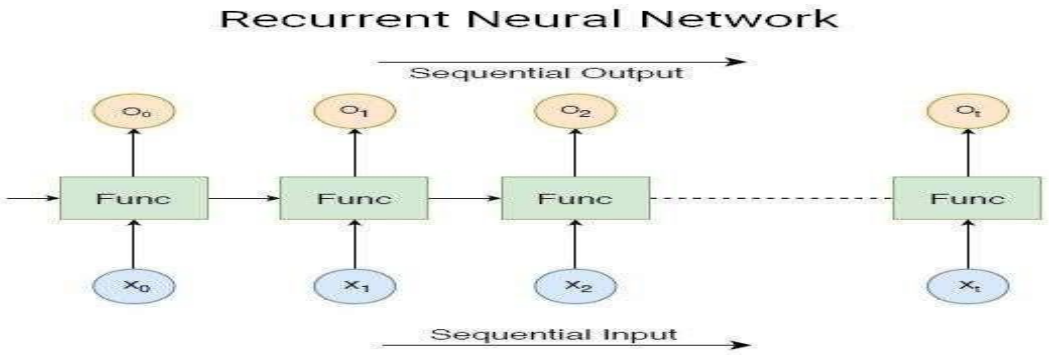


Fig 3.4: Recurrent Neural Network

3.2 Algorithms Used

CNN (Convolutional Neural Networks)

Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Researchers and enthusiasts alike, work on numerous aspects of the field to make amazing things happen. One of many such areas is the domain of Computer Vision. The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm CNN

VGG-16 (Visual Geometry Group):

VGG-16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU'

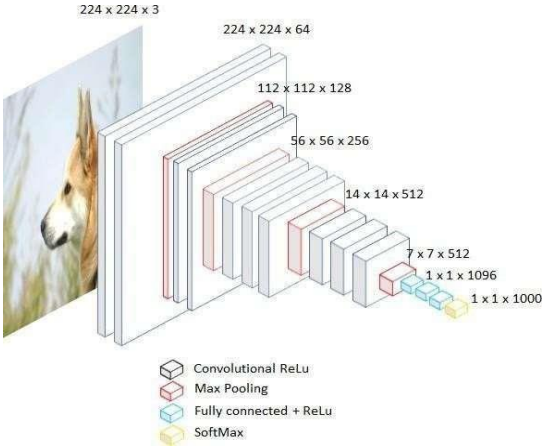


Fig 3.5: Visual Geometry Group - 16

CHAPTER 4

SYSTEM REQUIREMENTS SPECIFICATIONS

Software Specification

The languages of programming for logical formulation (Information science, AI applications, large-scale information preparation, advanced examinations, and so on) are dispersed free and open source, which implies working to bundle managers and organizations. This means working together.

The Anaconda bundle and virtual climate leader accompany Boa constrictor dispersion in excess of 1500 bundles. It also features a graphical alternative in contrast to the command line interface as a GUI called conda Navigator (CLI).

The great difference between Anaconda and the pipe-bundle administrator is in the control of bundle conditions, which for Python information science is very difficult and the Conda explanation exists. Pip introduces all Python bundle requirements, whether or whether you have introduced those bundles in advance.

So, when you pip a different bundle that demands an alternative version of the Numpy library, your functionality of, for example, Google Tensorflow, might abruptly cease operating. Everything could in any event look all the more traitorously, while you currently obtain different findings from your information studies or, in light of the fact you haven't introduced a comparable request, you cannot reproduce similar results elsewhere.

Conda breaks down the existing climate and any variants (for example, you only require tensorflow >= 2.0) that you have added and figures out ways to create feasible circumstances. Or it will let you know again that you don't have anything that you need. In contrast, Pip will only introduce what you required and circumstances, irrespective of whether other things are broken. Conda Cloud (anaconda.org) or your own archive or mirror can only provide open-source bundles, using the order entered in conda. Boa Constrictor Inc. collects and produces each bundle in the Anaconda Store itself and provides Windows 32/64, Linux 64 and MacOS 64-bit duplicate. Equally, you can insert anything on PyPI into a piping Conda climate, and Conda understands what the piping introduced. You may use the Anaconda order to make custom bundles and transmit them to others through the transfer to conda Cloud, PyPI or other repository.

Python 2.7 and conda3 are included in the basic setup of conda2. In any instance, new criteria may be established which include any kind of Python with conda.

4.1 Functional Requirements

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications.

Functional software requirements help you to capture the intended behaviour of the system.

Benefits of functional requirements:

- Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application
- A functional requirement document helps you to define the functionality of a system or one of its subsystems.
- Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
- Errors caught in the Functional requirement gathering stage are the cheapest to fix.
- Support user goals, tasks, or activities

4.2 Basic Requirements

1. Data collection and Annotation: A diverse and representative dataset of images or video clips containing instances of animal intrusions is required. The dataset needs to be properly annotated, indicating the presence of animals and, if possible, specifying the species.

2. Data Preprocessing: Preprocessing techniques to clean and prepare the dataset for training, including resizing images, normalization, and augmentation to enhance the model's robustness and generalization.

3. Model training: After a data scientist has preprocessed the collected data and split it into train and test can proceed with a model training. This process entails ―feeding the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data an answer you want to get a predictive analysis. The purpose of model training is to develop a model. We trained our model using the random forest algorithm. On training the model it predicts the yield on giving the other attributes of the dataset as input.

4. Model evaluation and testing: The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That‘s the optimization of model parameters to achieve an algorithm‘s best performance.

5. Training and Model updating: Procedures for initial model training and subsequent updates to adapt to changes in the environment, seasonal variations, or new animal behaviors.

4.3 Non-Functional Requirements

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional requirements are critical aspects of a project that define how the system should perform and behave rather than specifying specific features. For a project aiming to safeguard agriculture using deep learning techniques to prevent animal invasion, the non-functional requirements include:

1. Reliability:

The system should have high reliability, ensuring consistent and accurate detection of animal invasions to minimize false negatives and positives, especially in critical agricultural areas**.**

2. Performance:

The system should demonstrate high-performance capabilities, processing real-time data efficiently and maintaining low latency in detecting and responding to animal invasions.

3. Scalability:

The solution must be scalable to accommodate varying scales of agriculture, from small farms to large plantations, and should handle an increasing number of surveillance cameras and sensors without compromising performance**.**

4. Availability:

The system should maintain high availability, ensuring that it operates continuously to protect crops against animal invasions. Any scheduled maintenance or downtime should be minimal and strategically planned.

5. Adapatability to Environmental Conditions:

The system should be designed to adapt to diverse environmental factors such as different terrains, weather conditions, and lighting variations commonly found in agricultural settings.

6. Usability:

The user interface should be intuitive, user-friendly, and easily navigable for farmers or agricultural workers. Training requirements for end-users should be minimal, and the system should be accessible to users with varying levels of technical expertise.

7. Security:

Implement robust security measures to protect the system from unauthorized access, data breaches, or tampering. Ensure that sensitive agricultural data and surveillance information are securely stored and transmitted.

8. Compliance with Regulations:

Adhere to relevant agricultural and privacy regulations to ensure that the system complies with legal standards and ethical considerations. This includes data protection, surveillance, and environmental regulations.

9. Cost-effectiveness:

Strive for a cost-effective solution, considering both initial deployment costs and ongoing operational expenses. The benefits gained from preventing animal invasions should justify the investment in the system.

10. Maintainability:

Design the system with ease of maintenance in mind, allowing for straightforward updates, patches, and system enhancements. Provide documentation and support mechanisms for system administrators and maintenance personnel.

4.4 Python Libraries:

Normally, a library is a collection of books or is a room or place where many books are stored to be used later. Similarly, in the programming world, a library is a collection of precompiled codes that can be used later on in a program for some specific well-defined operations. Other than pre-compiled codes, a library may contain documentation, configuration data, message templates, classes, and values, etc.

A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning and Deep Learning, Data Science, Data Visualization, etc.

Working of Python Library

As is stated above, a Python library is simply a collection of codes or modules of codes that we can use in a program for specific operations. We use libraries so that we don't need to write the code again in our program that is already available. But how it works. Actually, in the MS Windows environment, the library files have a DLL extension (Dynamic Load Libraries). When we link a library with our program and run that program, the linker automatically searches for that library. It extracts the functionalities of that library and interprets the program accordingly. That's how we use the methods of a library in our program. We will see further, how we bring in the libraries in our Python programs.

Packages used

The Python Standard Library contains the exact syntax, semantics, and tokens of Python. It contains built-in modules that provide access to basic system functionality like I/O and some other core modules. Most of the Python Libraries are written in the C programming language. The Python standard library consists of more than 200 core modules. All these work together to make Python a high-level programming language. Python Standard Library plays a very important role. Without it, the programmers can't have access to the functionalities of Python. But other than this, there are several other libraries in Python that make a programmer's life easier. Let's have a look at some of the commonly used libraries:

1. **OS:** The 'os' module provides a way to interact with the operating system. It

facilitates operations related to file and directory management, such as creating, deleting, and navigating through directories. Working with file paths, managing directories, checking file existence, and executing system commands

2. Pandas: Pandas is a powerful open-source data analysis and manipulation library for Python. It is built on top of the NumPy library and provides easy-to-use data structures and data analysis tools for cleaning, reshaping, aggregating, and visualizing structured data. The core data structure in Pandas is the Data Frame. It is a two-dimensional, labeled data structure with columns that can be of different data types. It resembles a spreadsheet or SQL table, making it efficient for analyzing and manipulating structured data.

3. NumPy: 'NumPy' refers to Numerical Python. NumPy is a fundamental open-source library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays efficiently. NumPy is a foundational library for data science, machine learning, and scientific computing**.** NumPy arrays are more efficient for numerical operations compared to Python lists.

4. OpenCV: OpenCV is an open-source software library for computer vision and machine learning. The OpenCV full form is Open-Source Computer Vision Library. It was created to provide a shared infrastructure for applications for computer vision and to speed up the use of machine perception in consumer products. OpenCV, as a BSD-licensed software, makes it simple for companies to use and change the code. There are some predefined packages and libraries that make our life simple and OpenCV is one of them.

5. Random: The random module is part of Python standard library and provides functions for generating pseudo-random numbers. These functions are used for various applications such as simulations, games statistical sampling and cryptography.

6. Matplotlib: Matplotlib is a plotting library for python programming language and its numerical mathematics extension Numpy. Matplotlib is a data visualization library that's used for making plots and graphs. Matplotlib can produce high-quality and publish ready diagrams, graphs, plots, histograms, error charts, scatter plots and bar charts. It is easy to use and also offers GUI toolkit support, including wxPython, Tkinter, and Qt.

7. **Scikit-learn:** Scikit-learn package also known as sklearn. Scikit-learn is a free Python library that's often considered a direct extension of SciPy. It was specifically designed for data modeling and developing machine learning algorithms, both supervised and unsupervised.

8. **Tensor-Flow:** Tensor Flow is a free and open-source python library that offers a collection of tools and resources that help make building Deep learning and Machine learning models. TensorFlow architecture and framework are flexible and allow it to run on several computational platforms such as CPU and GPU.

9. **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn produces the most visually appealing and attractive graphs and plots, making it perfect for use in publications and marketing. It allows to create extensive graphs with little code and simple commands so it can help save time and effort.

4.5 Hardware Requirements

The hardware requirements include the requirements specification of the physical computer resources for a system to work efficiently. The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. The Hardware Requirements are listed below:

System Processor	:	Intel I3
Hard Disk	:	500 GB
Ram	:	4 GB

4.6 Software Requirements

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

Operating system	:	Windows 10
Coding Language	:	Python
Libraries Used	:	Pandas, Numpy
Development Tool	:	Google Colab(GPU)

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

Systems development is a systematic process which includes phases such as planning, analysis, design, deployment, and maintenance. System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

System Design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently. System Design focuses on how to accomplish the objective of the system.

5.1 BLOCK DIAGRAM:

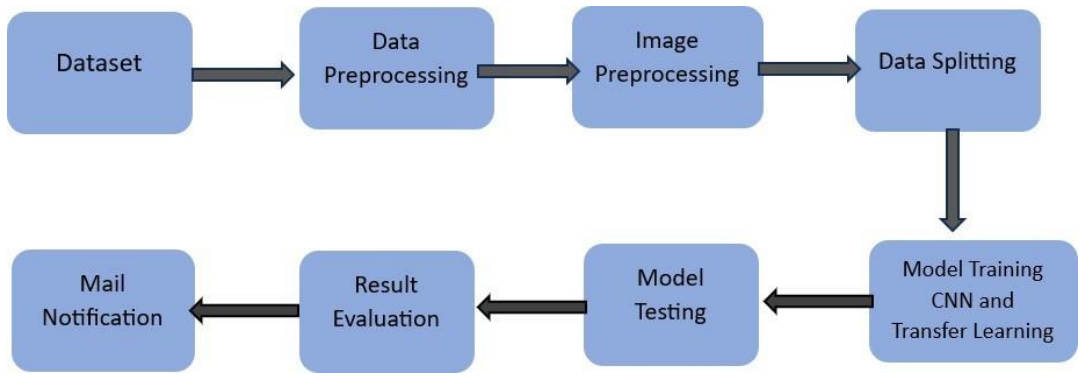


Fig 5.1: Block Diagram

In case build CNN, VGG-16, VGG-19 and Mobile net model. We train the model on train data and test it. Based on results we finalize the model and use that final model to deploy the project and based on that model only we send mail notification to the famer whenever any animal came into their fields.

5.1.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

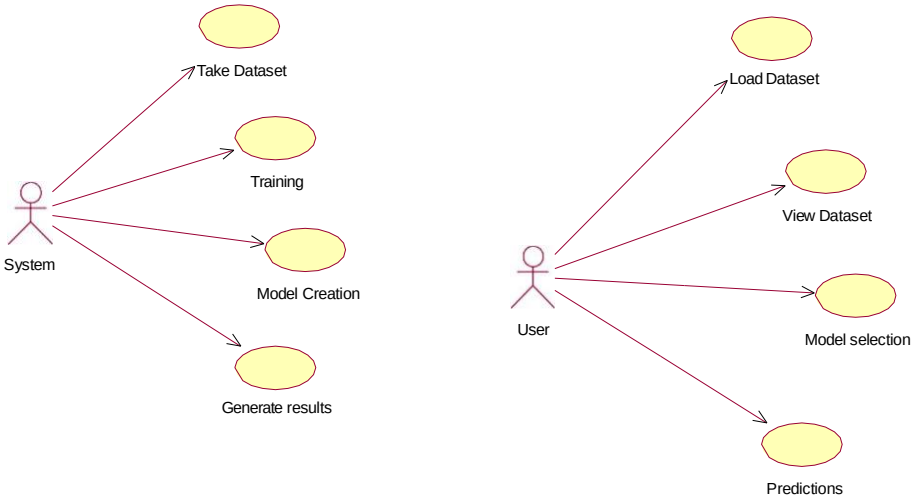


Fig 5.2: Usecase Diagram

5.2 System Architecture

Architecture diagrams can help system designers and developers visualize the high-level, overall structure of their system or application for the purpose of ensuring the system meets their users' needs. They can also be used to describe patterns that are used throughout the design. It's somewhat like a blueprint that can be used as a guide for the convenience of discussing, improving, and following among a team. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

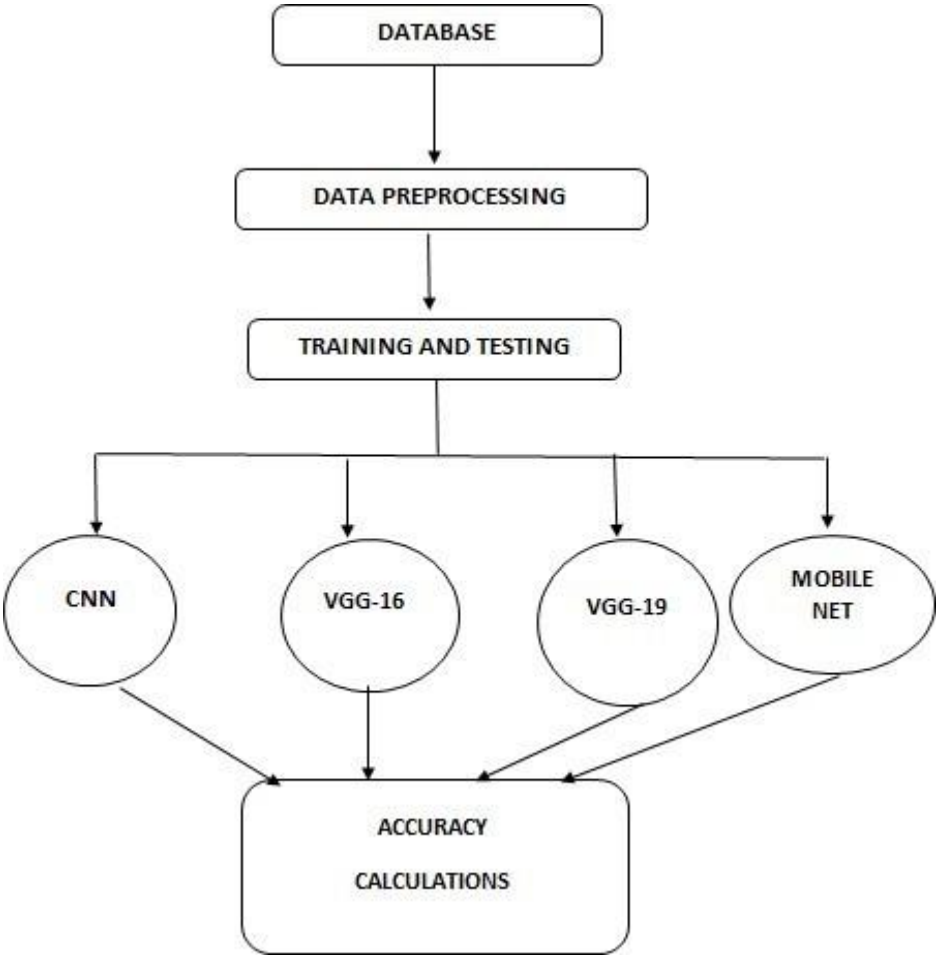


Fig 5.3: System Architecture

Fetching Data:

Fetch is the retrieval of data by a software program, Script, or a hardware device. After being retrieved, the data is moved to an alternate location or displayed on a screen.

Data Set:

A data set is a collection of related, discrete items of related data that may accessed individually or in combination or managed as a whole entity. A data set is organized into some type of data structure.

Data Cleaning:

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted . But, as we mentioned above, it isn't as simple as organizing some rows or erasing information to make space for new data. Data cleaning is a lot of muscle work. There's a reason data cleaning is the most important step if you want to create a data-culture, let alone make airtight predictions. It involves:

- Fixing spelling and syntax errors
- Standardizing data sets
- Correcting mistakes such as empty fields
- Identifying duplicate data points

Data Pre-processing:

Most machine learning algorithms require data to be formatted in a very specific way, so datasets generally require some amount of preparation before they can yield useful insights. Some datasets have values that are missing, invalid, or otherwise difficult for an algorithm to process.

If data is missing, the algorithm can't use it. If data is invalid, the algorithm produces less accurate or even misleading outcomes. Some datasets are relatively clean but need to be shaped (e.g., aggregated or pivoted) and many datasets are just lacking useful business context (e.g., poorly defined ID values), hence the need for feature enrichment. Good data preparation produces clean and well-curated data which leads to more practical, accurate model outcomes.

Model Training:

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model This iterative process is called "model fitting". The accuracy of the training dataset or the validation dataset is critical for the precision of the model. Model training in machine language is the process of feeding an ML algorithm with data to help identify.

CHAPTER 6

IMPLEMENTATION

In this project we are working with four different modules:

- Data Pre-processing
- Normalizing Numeric data
- Partitioning Dataset
- Algorithms

6.1 Data-Preprocessing:

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way.

Includes the following steps:

1. Getting the dataset
2. Importing libraries
3. Importing datasets
4. Finding Missing Data
5. Encoding Categorical Data
6. Splitting dataset into training and test set
7. Feature scaling

1. Getting the dataset:

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset.

The benefits of user feedback for safe and effective medication usage are now being demonstrated. Consumer perceptions of their ailments and previously used medications are included in this data collection. Companies like 1mg may find this product useful line providing detailed ratings of the product's side effects

their website

2. Importing Libraries:

First step is usually importing the libraries that will be needed in the program. A library is essentially a collection of modules that can be called and used.

3. Importing datasets:

A lot of datasets come in CSV formats. We will need to locate the directory of the CSV file at first and read it using a method called read.csv.

```
Images = []
import os
for dirname, _, filenames in os.walk(data):
    for filename in filenames:
        img = os.path.join(dirname, filename)
        Images.append(img)
```

Fig 6.1 Datasets

4. Finding missing data:

The next step of data pre-processing is to handle missing data in the datasets. If our data set contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

There are mainly two ways to handle missing data, which are:

By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

5. Encoding categorical data:

Whenever we have a text data need do apply text processing and clean it. In this text preprocessing first step punctuation symbols removal. First step want to remove some punctuation removal there is no using this symbol and get create some high

dimensionality. Second step remove the stop words define or import the stop words from NLTK tool and remove the all stop from the each data point and then apply the tokenization. In this step split the sentence into words and apply stemming. Stemming is nothing but convert the word into base form for example beautiful, beauty, be stain the base form is beauty. By using stemming concept we can reduce the dimensionally also. By doing the all the text preprocessing steps we will step preprocessed text. Apply text featurization concept on preprocessed.

6. Splitting the dataset into training set and testing set:

Now we need to split our dataset into two sets a Training set and a Test set. We will train our machine learning models on our training set and then we will test the models on our test set to check how accurately it can predict.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these data sets as:

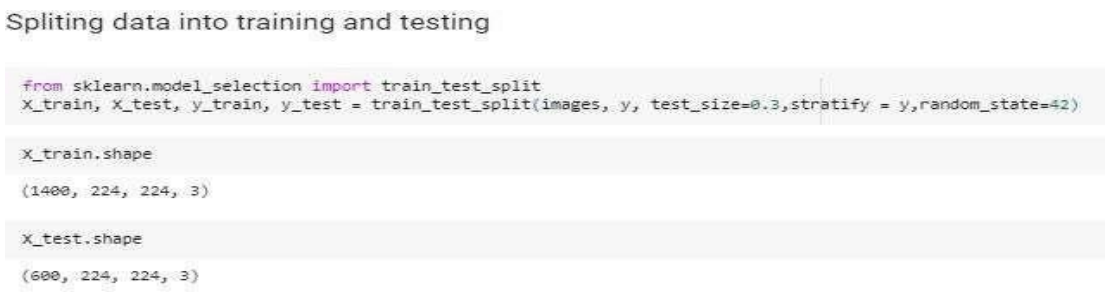


Fig 6.2: Splitting data into train and test

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test Set: A subset of data set to test the machine learning model, and by using the test set, model predicts the output.

7. Feature scaling:

Feature scaling is the final step of data pre-processing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no any variable dominate the other variable. The goal is to

bring all features to a similar scale, mitigating the impact of different units or magnitudes.

6.2 Normalizing Numeric Data:

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization. It is required only when features have different ranges.

Normalization in machine learning is the process of translating data into the range [0, 1] (or any other range) or simply transforming data onto the unit sphere. Some machine learning algorithms benefit from normalization and standardization, particularly when Euclidean distance is used.

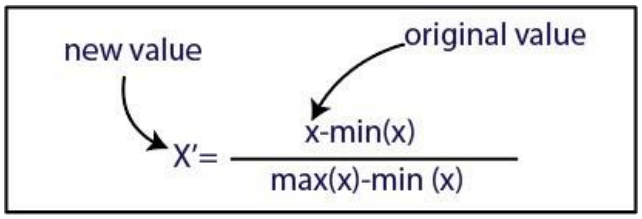


Fig 6.3: Euclidean distance

6.3 Data Set:

Image Net is a dataset of over 15 million labelled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labelled by human labellers using Amazon's Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. At all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. Image Net consists of variable- resolution images. Therefore, the images have been down-sampled to a fixed resolution of 256×256. Given a rectangular image, the image is rescaled and cropped out the central 256×256 patch from the resulting image.

Due to its depth and number of fully-connected nodes, VGG16 is over 533MB. This makes deploying VGG a tiresome task. VGG-16 is used in many deep learning image classification problems; however, smaller network architectures are often more desirable (such as Squeeze Net, Google Net, etc.). But it is a great building block for learning purpose as it is easy to implement.

6.4 Algorithms

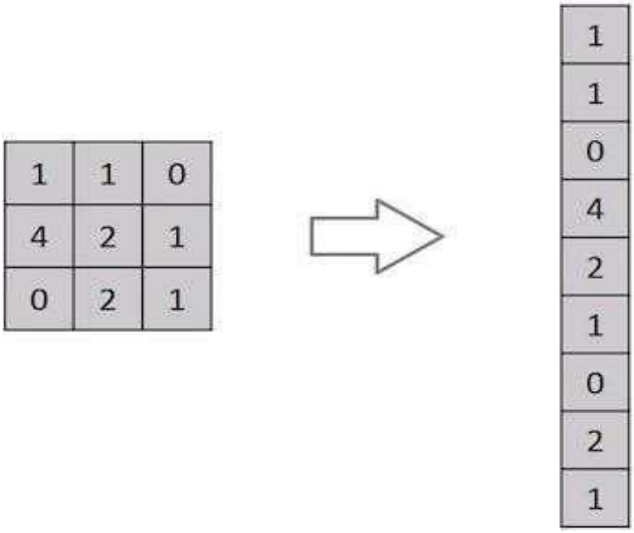
6.4.1 CNN (Convolutional Neural Networks)

Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. One of many such areas is the domain of Computer Vision. The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm CNN.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand- engineered, with enough training, ConvNets have the ability to these filters/characteristic

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

Why Conv Nets over Feed-Forward Neural Nets?



Flattening of a 3x3 image matrix into a 9x1 vector

An image is nothing but a matrix of pixel values. So we just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes

In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A Conv Net is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Input Image:

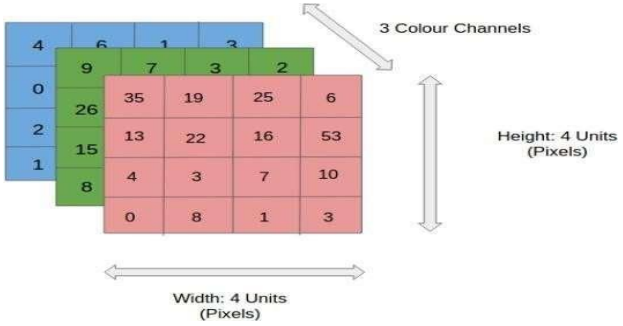


Fig.6.4 4x4x3 RGB Image

In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc. You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

Convolution Layer — The Kernel:

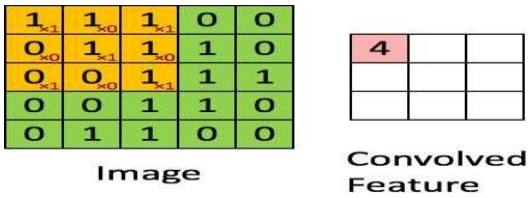


Fig.6.5: Convolutional layer image

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, Ex: RGB). In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix

Pooling Layer:

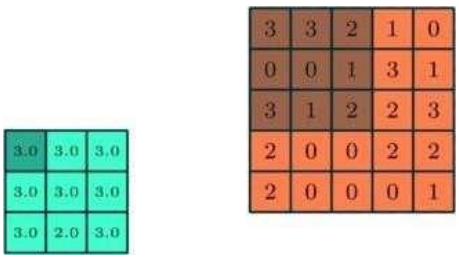


Fig.6.6 Pooling layer

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

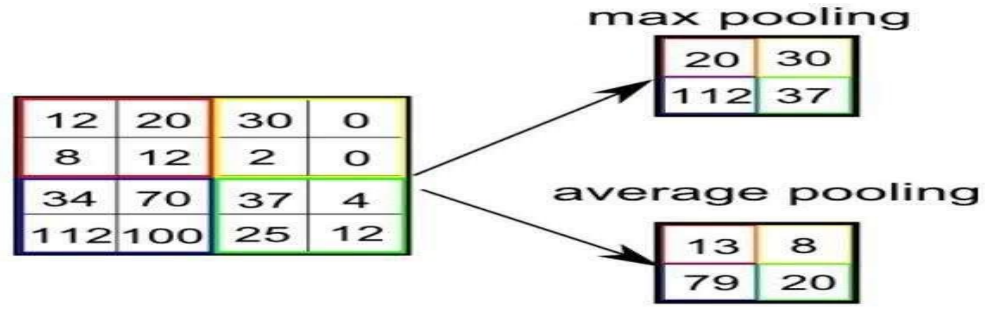


Fig 6.7: Average and Max Pooling

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

Classification — Fully Connected Layer (FC Layer):

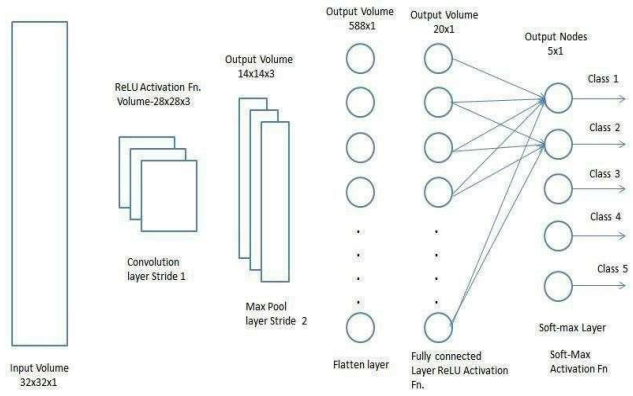


Fig.6.8: Classification Fully connected layer

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully- Connected layer is learning a possibly non-linear function in that space.

Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed- forward neural network and back propagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the **Soft max Classification** technique.

CNN Model

```
# Set the CNN model

batch_size = None

model = Sequential()

model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
                 activation ='relu', batch_input_shape = (batch_size,224, 224, 3)))


model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
                 activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.2))


model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',
                 activation ='relu'))
model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'same',
                 activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(filters = 128, kernel_size = (3,3),padding = 'Same',
                 activation ='relu'))
model.add(Conv2D(filters = 128, kernel_size = (3,3),padding = 'Same',
                 activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.4))


model.add(GlobalMaxPooling2D())
model.add(Dense(256, activation = "relu"))
model.add(Dropout(0.5))
model.add(Dense(10, activation = "softmax"))
model.summary()
```

Fig 6.9: CNN model

6.4.2 VGG-16:

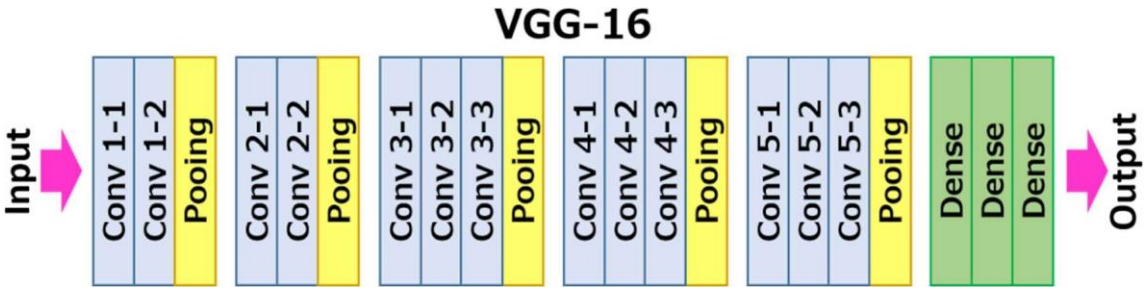


Fig 6.10: Layers of VGG

VGG-16 is a convolutional neural network (CNN) architecture designed for image classification. It was introduced by the Visual Geometry Group (VGG) at the University of Oxford. The "16" in VGG-16 represents the number of layers in the network. VGG-16 is characterized by its simplicity and uniform architecture, where each convolutional layer has 3x3 filters and is followed by a max-pooling layer. The architecture has proven effective for various computer vision tasks, including image recognition.

Fig 6.11: VGG-16 Model

6.4.3. VGG-19

VGG-19 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 19 layers deep and can classify images into 1000 object categories, such as a keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images.

Fixed size of (224 * 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3).

- The only preprocessing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set.
- Used kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image.
- spatial padding was used to preserve the spatial resolution of the image.
- max pooling was performed over a 2 * 2 pixel windows with sride 2.
- this was followed by Rectified linear unit (ReLu) to introduce non-linearity to make the model classify better and to improve computational
- implemented three fully connected layers from which the first two were of size 4096 and after that, a layer with 1000 channels for 1000.

VGG_19 Model

```
7] # import the vgg16 model
   from keras.applications.vgg19 import VGG19

8] vgg_19 = VGG19(weights='imagenet',include_top=False,input_shape=(224,224,3))

   Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
   80134624/80134624 [==============================] - 1s 0us/step

9] vgg_19.trainable=False

0] # Set the vgg16 model
   model_2=Sequential()
   model_2.add(vgg_19)
   model_2.add(Flatten())
   model_2.add(Dense(128, activation='relu'))
   model_2.add(Dropout(0.2))
   model_2.add(Dense(10, activation='softmax'))
```

Fig 6.12: VGG-19 model

6.5 Source Code

```
#list of useful imports that  I will use
%matplotlib inline
import os
import tqdm
import matplotlib.pyplot as plt
import pandas as pd
import cv2
import numpy as np
from glob import glob
import seaborn as sns
import random
from keras.preprocessing import image
import tensorflow as tf
from tensorflow.keras.models import Model
from keras.utils.np_utils import to_categorical # convert to one-hot-encoding
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D,
GlobalMaxPooling2D,Input
#from keras.optimizers import RMSprop
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
Using TensorFlow backend.
# Run this cell to mount your Google Drive.
```

```
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
file = '/content/drive/MyDrive/Animal/Animal_data (1).zip'
```

Unziping data set:

```
import zipfile as zf
data_zip = zf.ZipFile(file)
data_zip.extractall()
!ls
Animal_data  drive  sample_data
data = 'Animal_data'
Images = []
import os
for dirname, _, filenames in os.walk(data):
for filename in filenames:
img = os.path.join(dirname, filename)
Images.append(img)
len(Images)
999
Class_label = []
for i in Images:
j = i.split("/")
#print(j[1])
Class_label.append(j[1])
Class_label[:10]
len(Class_label)
# Shuffle two lists with same order
# Using zip() + * operator + shuffle()
temp = list(zip(Images, Class_label))
random.shuffle(temp)
Images, Class_label = zip(*temp)
data = pd.DataFrame(list(zip(Images, Class_label)), columns =['Image_path', 'Class_label'])
data.head(5)
data.Class_label.value_counts()
```

```
sns.set(style="whitegrid")
plt.figure(figsize=(10, 5))
ax = sns.countplot(x="Class_label", data=data,    palette=sns.color_palette("cubehelix", 4))
plt.xticks(rotation=90)
plt.title("Class Label Counts", {"fontname":"fantasy", "fontweight":"bold",
"fontsize":"medium"})
plt.ylabel("count", {"fontname": "serif", "fontweight":"bold"})
plt.xlabel("Class_label", {"fontname": "serif", "fontweight":"bold"})
from sklearn.utils import resample
# Separate majority and minority classes
df_c0 = data[data['Class_label']== 'Giraffe']
df_c1 = data[data['Class_label']== 'Deer']
df_c2 = data[data['Class_label']== 'Camel']
df_c3 = data[data['Class_label']== 'Cow']
df_c4 = data[data['Class_label']== 'Rabbit']
df_c5 = data[data['Class_label']== 'Pig']
df_c6 = data[data['Class_label']== 'Horse']
df_c7 = data[data['Class_label']== 'Monkey']
df_c8 = data[data['Class_label']== 'Tiger']
df_c9 = data[data['Class_label']== 'Elephant']
# Downsample majority class
df_c0_upsampled = resample(df_c0, replace=True,  n_samples = 200,random_state=123)
df_c1_upsampled = resample(df_c1, replace=True,  n_samples = 200,random_state=123)
df_c2_upsampled = resample(df_c2, replace=True,  n_samples = 200,random_state=123)
df_c3_upsampled = resample(df_c3, replace=True,  n_samples = 200,random_state=123)
df_c4_upsampled = resample(df_c4, replace=True,  n_samples = 200,random_state=123)
df_c5_upsampled = resample(df_c5, replace=True,  n_samples = 200,random_state=123)
df_c6_upsampled = resample(df_c6, replace=True,  n_samples = 200,random_state=123)
df_c7_upsampled = resample(df_c7, replace=True,  n_samples = 200,random_state=123)
df_c8_upsampled = resample(df_c8, replace=True,  n_samples = 200,random_state=123)
df_c9_upsampled = resample(df_c9, replace=True,  n_samples = 200,random_state=123)
# Combine minority class with downsampled majority class
df_upsampled                                                                    =
pd.concat([df_c0_upsampled,df_c1_upsampled,df_c2_upsampled,df_c3_upsampled,df_c4_ups
```

```
 ampled,df_c5_upsampled,df_c6_upsampled,df_c
# Display new class counts
df_upsampled['Class_label'].value_counts()
#counts of top 10 drugs
sns.set(style="whitegrid")
 plt.figure(figsize=(10, 5))
 ax = sns.countplot(x="Class_label", data=df_upsampled,
palette=sns.color_palette("cubehelix", 4))
plt.xticks(rotation=90)
 plt.title("Class Label Counts", {"fontname":"fantasy", "fontweight":"bold",
"fontsize":"medium"})
 plt.ylabel("count", {"fontname": "serif", "fontweight":"bold"})
 plt.xlabel("Class_label", {"fontname": "serif", "fontweight":"bold"})
 data = df_upsampled.sample(frac=1)
 data.head()
```

Resize the images

```
def resize_images(img):
file = Image.open(img)
img = file.convert('RGB')
img_bgr= img.resize((224, 224))
img_bgr = np.array(img_bgr)
return img_bgr
from PIL import Image
#save resized images into images.
images = [resize_images(img) for img in data['Image_path']]
# print number of classes in our dataset
num_classes = len(np.unique(data['Class_label']))
num_classes
10
# save the class into class_names
class_names = list(data['Class_label'])
# Print the shape of the image
images[10].shape
```

```
(224, 224, 3)
#See the image with class label
plt.imshow(images[5])
plt.title(class_names[5])
 #See the image with class label
plt.imshow(images[10])
plt.title(class_names[10])
#See the image with class label
plt.imshow(images[400])
plt.title(class_names[400])
#See the image with class label
plt.imshow(images[70])
plt.title(class_names[70])
#See the image with class label
plt.imshow(images[100])
plt.title(class_names[100])
#See the image with class label
plt.imshow(images[150])
plt.title(class_names[150])
```

Label encoder:

```
from sklearn.preprocessing import LabelBinarizer
enc = LabelBinarizer()
y = enc.fit_transform(data['Class_label'])
data['Class_label'][:10]
images = np.array(images)
images.shape
(2000, 224, 224, 3)
```

CNN code:

```
#Set the CNN model
```

```
batch_size = None
model - Sequential()
model.add(Conv2D (filters = 32, kernel_size = (5,5), padding = Same, activation 'relu',
batch_input_shape = (batch_size, 224, 224, 3)))
model.add(Conv2D(filters = 32, kernel_size = (5,5), padding= "Same, activation 'relu'))
model.add(MaxPool2D (pool_size=(2,2))) model.add(Dropout(0.2))
model.add(Conv2D (filters = 64, kernel_size = (3,3), padding
"Same,
activation 'relu")) model.add(Conv2D (filters = 64, kernel_size = (3,3), padding "same",
activation = 'relu'))
model.add(MaxPool2D (pool_size=(2,2), strides=(2,2))) model.add(Dropout(0.3))
model.add(Conv2D (filters 128, kernel_size = (3,3), padding activation 'relu')) = "Same"
model.add(Conv2D(filters = 128, kernel_size= (3,3), padding = "Same', activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2))> model.add (Dropout(0.4))
model.add(GlobalMaxPooling2D())
model.add(Dense (256, activation = "relu"))
model.add(Dropout(e.s))
model.add(Dense(10, activation = "softmax"))
model.summary()
```

VGG-16 code

```
import the vgg16 model from keras.applications.vgg16 import VGG16
vgg-VGG16(weights='imagenet", include_top=False, input_shape=(224,224,3))
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/ver16/veg16 weights tf dim ordering tf kernels notop.hs 58889256/58889256
15 eus/step
vgg.trainable=False
#Set the vgg16 model model_1=Sequential()
model_1.add(vgg)
model_1.add(Flatten())
model_1.add(Dense(128, activation="relu"))
model_1.add(Dropout(0.2))
model_1.add(Dense(10, activation='softmax'))
```

VGG-19 code

```
import the vgg16 model
from keras.applications.vgg19 import VGG19
Vgg 19 VGG19(weights='imagenet'', include_top=False, input_shape=(224,224,3))
Downloading data from https://storage.googleapis.com/tensorflow/keras-
agglications/v8819/ver19 weights tf dim ordering tf kernels notop.hs 80134624/80134624
[] 15 Bus/step
vgg 19.trainable-False
# set the vgg16 nodel
model_2 Sequential()
model_2.add(vgg_19)
model_2.add(Flatten())
model_2.add(Dense (128, activation'relu'))
model_2.add(Dropout(0.2))
model_2.add(Dense(10, activation='softmax'))
```

CHAPTER 7

RESULTS

```
['Animal_data/Horse/images (1).jpg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.24.52 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.24.50 AM (1).jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.24.33 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.25.19 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.24.49 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.25.01 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.25.10 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.25.17 AM.jpeg',
 'Animal_data/Horse/WhatsApp Image 2022-07-06 at 12.24.58 AM (1).jpeg']
```

Fig 7.1 Datasets

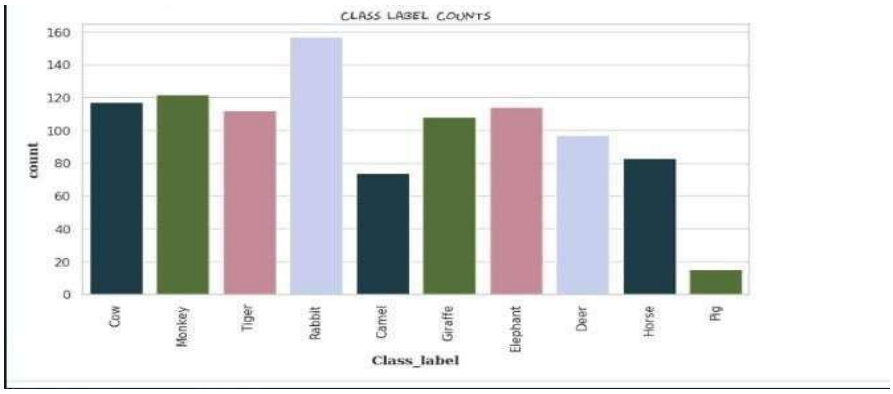


Fig 7.2: Class Label Counts

In the above image it shows the count of images of different animals in the given dataset.

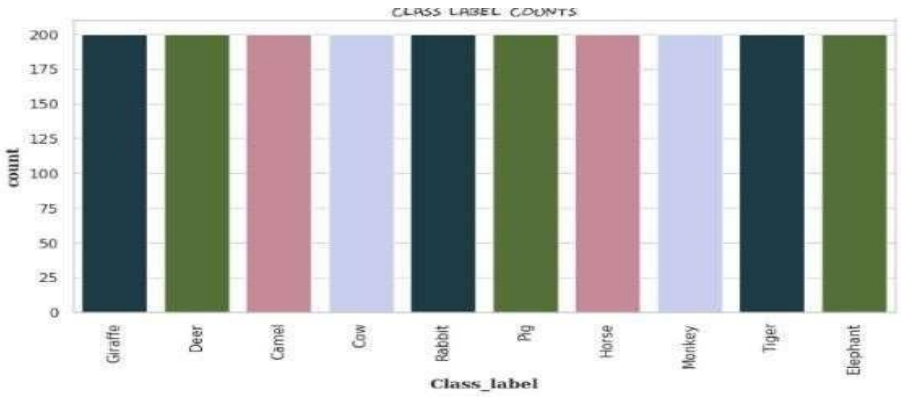


Fig 7.3: Resample data

In above image it will resample the data and make all the datasets into equal number

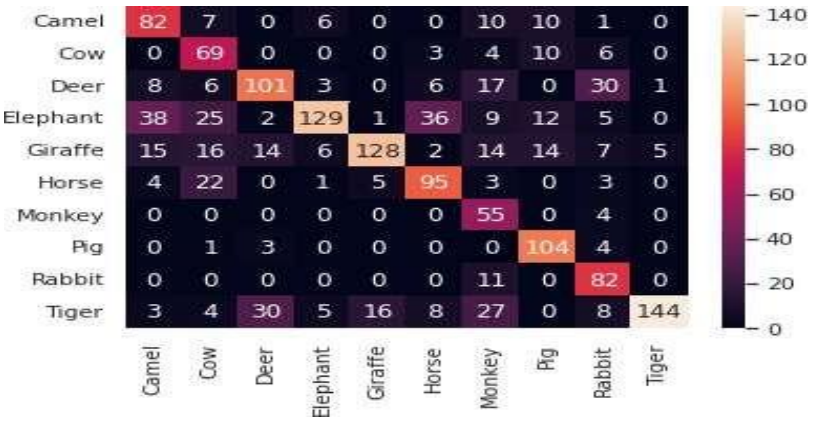


Fig 7.4: Confusion Matrix for CNN

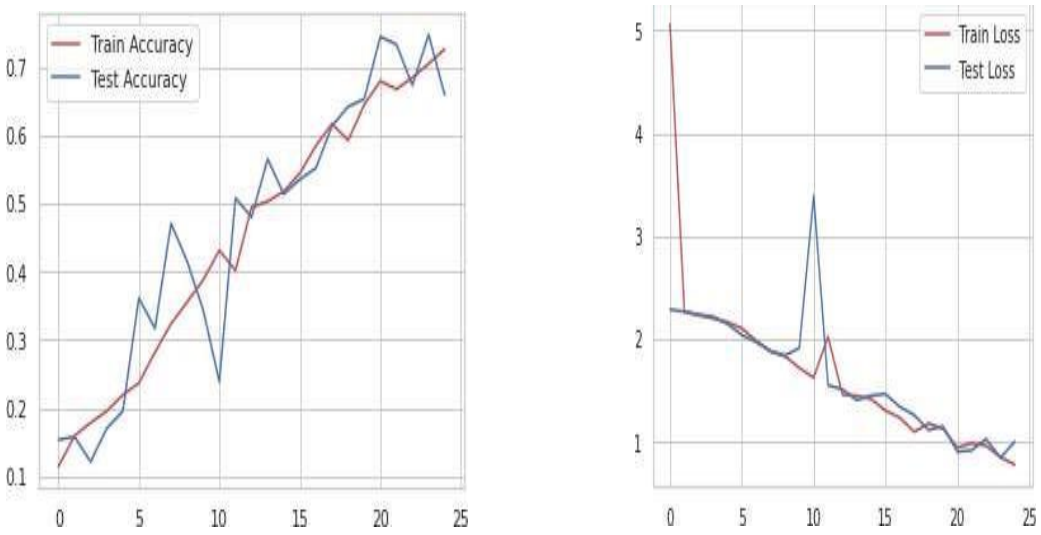
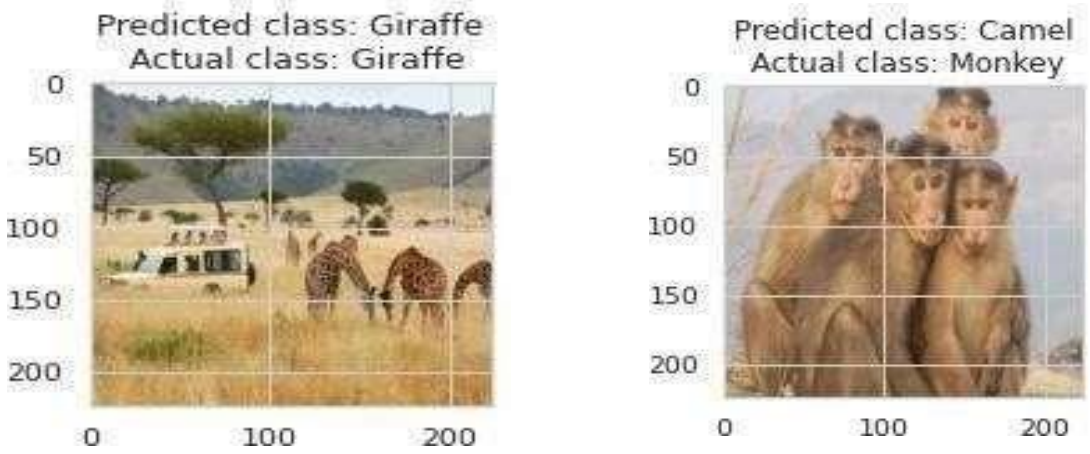


Fig 7.5: CNN Accuracy plots

The above diagram represents that accuracy plot illustrates the performance of the CNN on the training and validation datasets over different training epochs



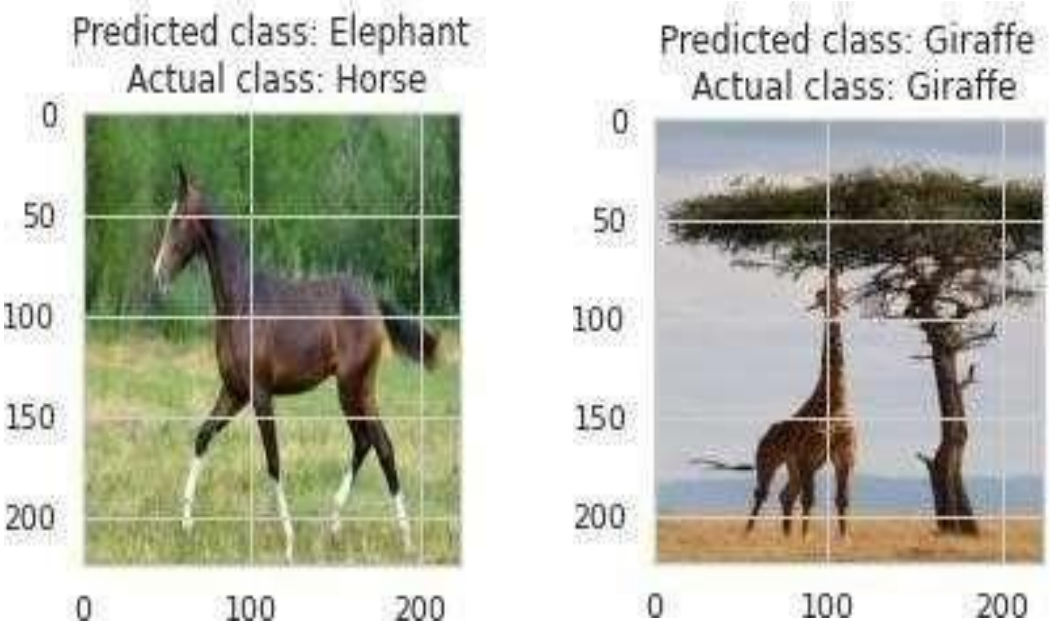


Fig 7.6: CNN Model Outputs

It will show the actual class and predicted class. If both the actual class and predicted class have same image then it will be the correct output.

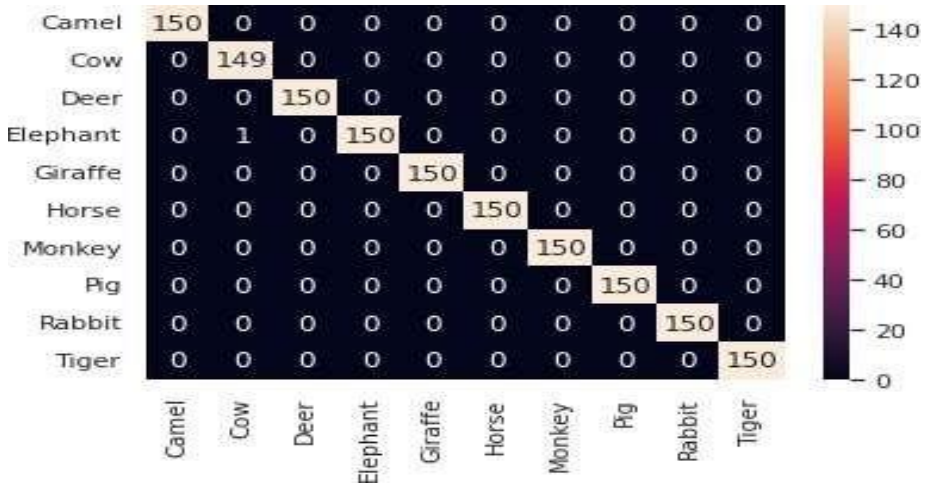


Fig 7.7: VGG-16 model confusion matrix

The above output represents the VGG-16 model confusion matrix. Compared to CNN we can see that VGG-16 have trained the classes in their respective rows and columns.

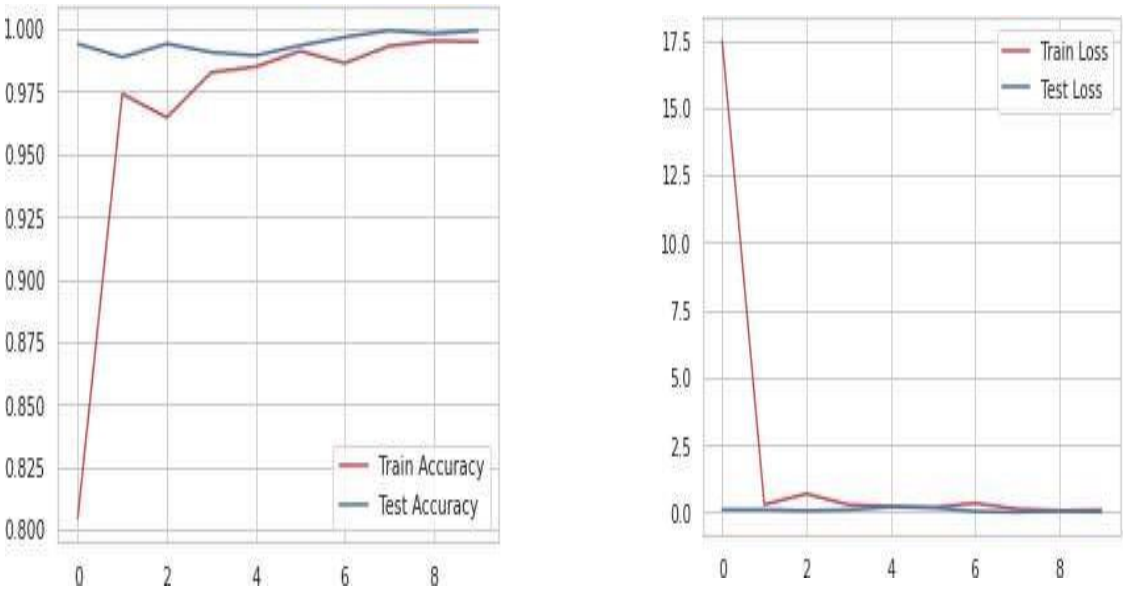


Fig 7.8: VGG-16 Model Accuracy and Loss Plots

The above diagram consists of VGG-16 model accuracy and loss plots. It will show the train and test accuracy of the model.

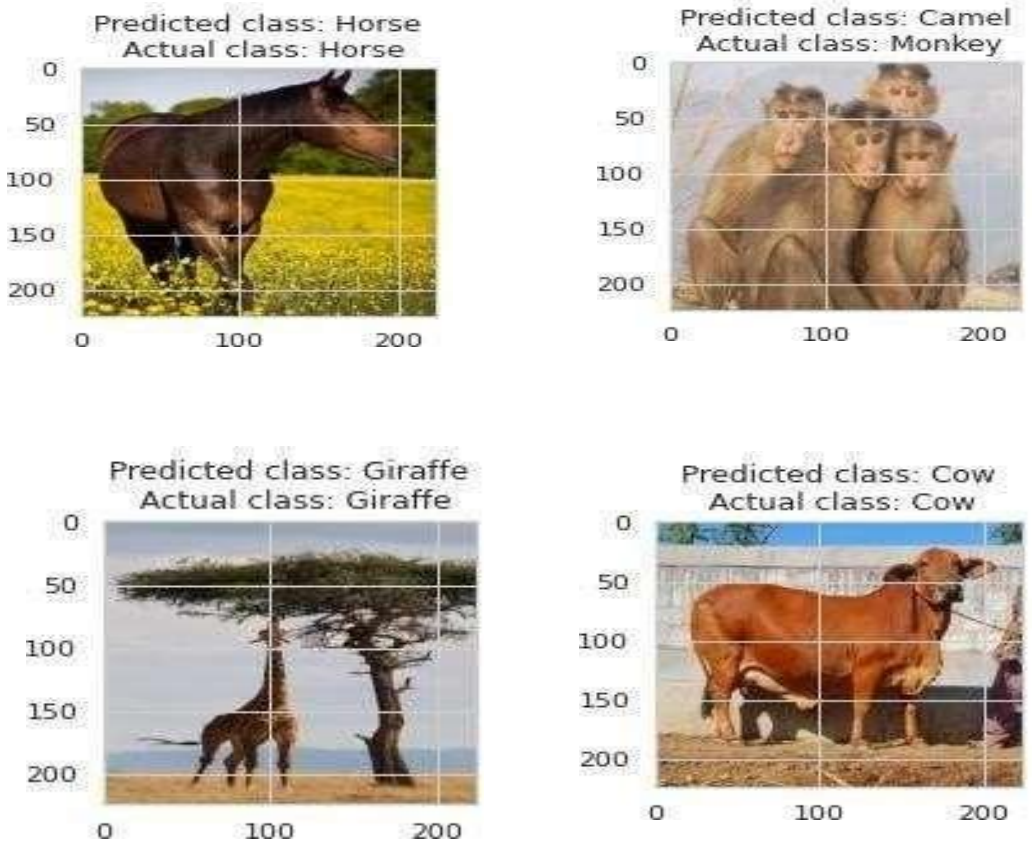


Fig 7.9: VGG-16 model output Images

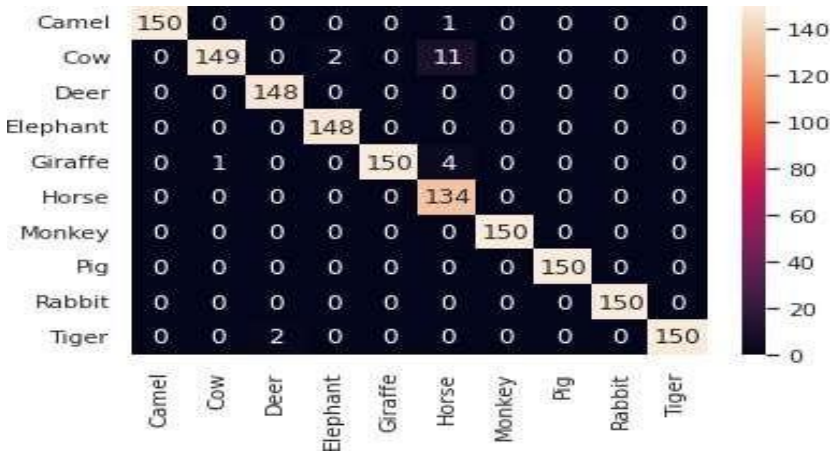


Fig 7.10: VGG-19 model confusion matrix

The above matrix is VGG-19 model confusion matrix which provides a detailed breakdown of the model's predictions and the actual outcomes across different classes. It will calculate the performance matrix

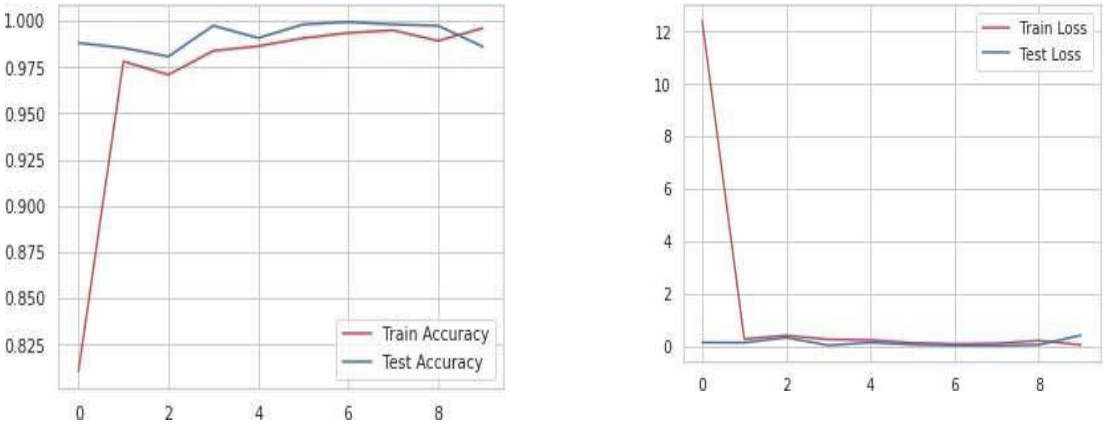
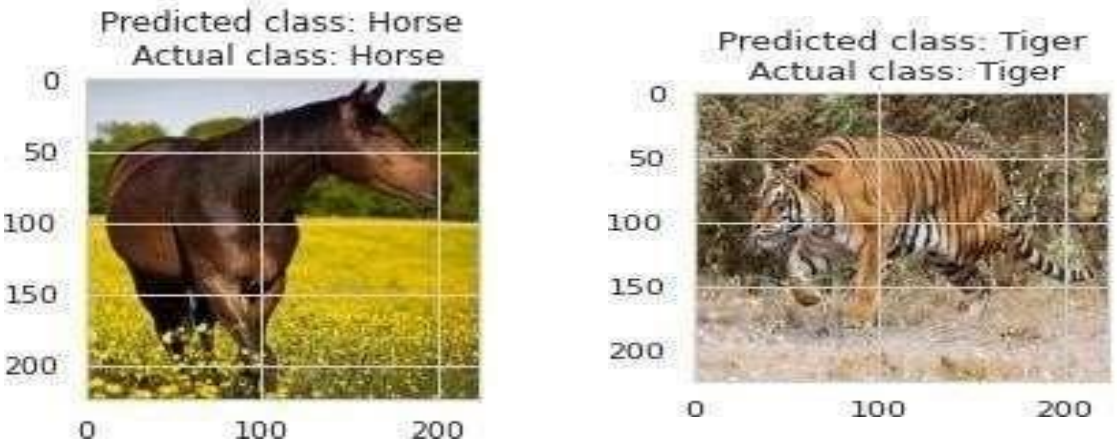


Fig 7.11: VGG-19Model Accuracy and Loss Plots



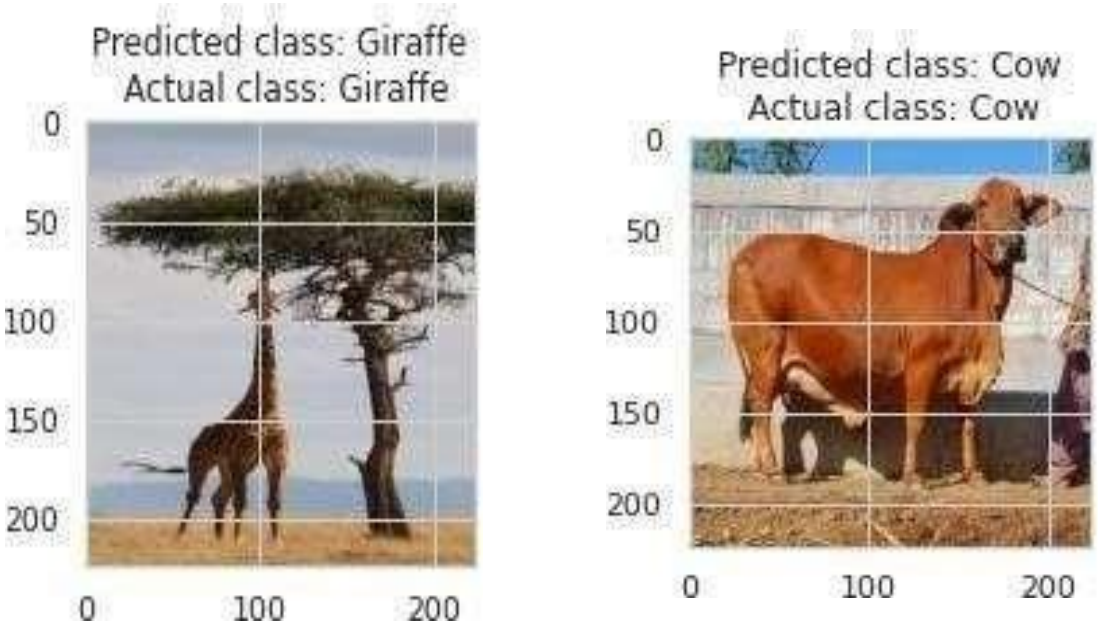


Fig 7.12: VGG-19 model output Images

Fig 7.13: Accuracy table

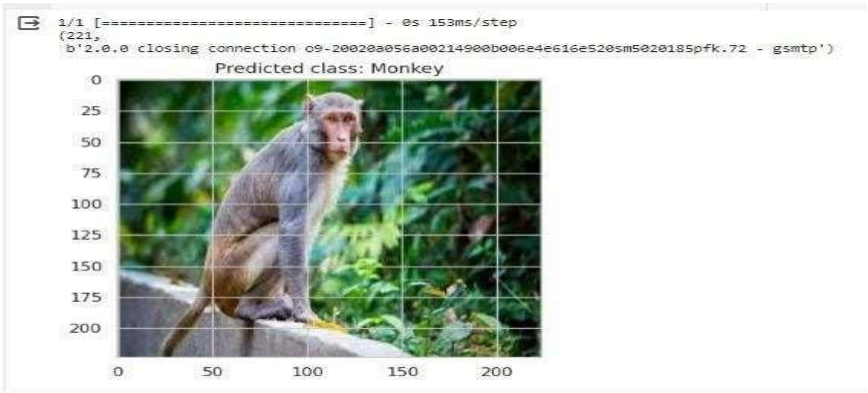


Fig 7.14: Predicted class

In above image the animal was detected and the predicted class was detected as a monkey.

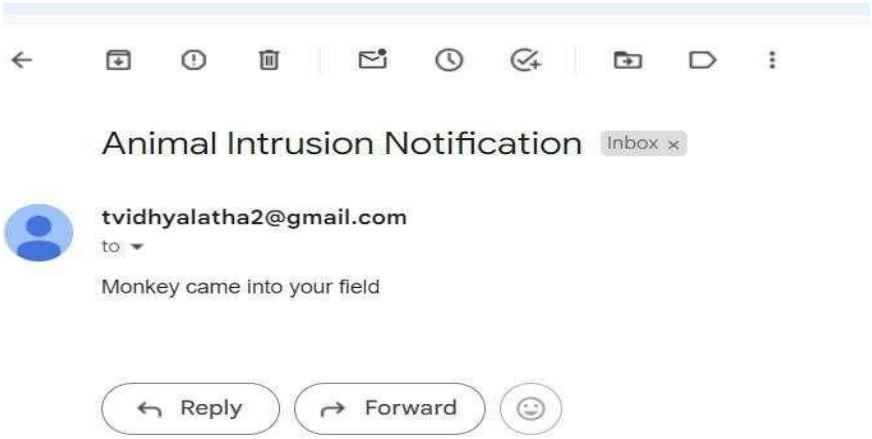


Fig 7.15: Mail Notification

CONCLUSION

The problem of crop by wild animals has become a major social problem in current time. In other words, while utilizing his/her crop production, every farmer should be aware and take into consideration the fact that animals are living beings and need to be protected from any potential suffering. It requires urgent attention and an effective solution. Thus, this project carries a great social relevance as it will help farmers in protecting their fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection of their fields. Safeguarding Agriculture Using deep learning to prevent animal invasion can have a wide range of future applications such as detecting wild animals in residential areas, monitoring wildlife migration patterns, protecting farm lands, and alerting drivers about animals on the road. The technology can also be adapted for other types of intrusion detection, such as identifying human or monitoring activity in restricted areas

REFERENCES

[1]. Hardiki Patil and Namrata Ansari, "Automated Wild-Animal Intrusion Detection and Repellent System using Artificial Intelligence of Things", of 4th international conference on Advances in Science & Technology (ICAST-2021).

[2]. Dr. Sabeenian R.S, N. Deivanai and B.Mythili "Wild Animals Intrusion Detection using Deep Learning Techniques", (IJPR) IEEE-2020.

[3]. K.Balakrishna, Fazil Mohammed, C.R.Ullas, S.K.Sonakshi and C.M.Hema, "Application of IOT and machine learning in crop protection against animal intrusion",Global transitions proceedings, November 2021.

[4]. Vidhyalatha T, Y Sreeram, E Purushotham "Animal Intrusion Detection using Deep Learning and Transfer Learning Approaches", International Journal of Human Computations & Intelligence, 2022.

[5]. Faseema M M and Ambily Jacob, "Wild Animal Detection in Agriculture farms using Deep Convolutional Neural Networks", International Journal of Creative Research Thought, March 2020.

[6]. Xue, Wenling, Ting Jiang, and Jiong Shi. "Animal intrusion detection based on convolutional neural network". 2017 17th international symposium on communications and information technologies (ISCIT). IEEE, 2017.

[7]. Ravoor, Prashanth C., T. S. B. Sudarshan, and Krishnan Rangarajan. "Digital Borders: Design of an Animal Intrusion Detection System Based on Deep Learning". International Conference on Computer Vision and Image Processing. Singapore: Springer Singapore, 2020.

[8]. Santhanam, Sanjay, et al. "Animal detection for road safety using deep learning". 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA). IEEE, 2021.

[9]. Bindu D et al. "Prevention of Wild Animals Entering Into the Agriculture Fields. International Journal of Engineering, Basic sciences, Management & Social studies", Volume 1, Issue 1, May 2017.

[10]. Krishnamurthy b et al. "Solar Fencing Unit and Alarm for Animal Entry Prevention". International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137, Volume – 02, Issue – 05, May – 2017, PP – 128-135.

[11]. Kshama S.Bhise. "Wildlife Animal Tracking Using RFID and GSM Technology. International Journal of Scientific & Engineering Research", Volume 7, Issue 2, February-2016 ISSN 2229-5518.

[12]. V. Deshpande. "Design and Implementation of an Intelligent Security System for Farm Protection from Wild Animals". International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611.

[13]. S. R. Chourey, P. A. Amale et al, IETE Zonal Seminar "IOT Based Wireless Sensor Network for Prevention of Crops from Wild Animals" -2017 Special Issue of International Journal of Electronics, Communication & Soft Computing Science and Engineering, ISSN: 2277-9477.

[14]. S. J. Sugumar and R. Jayaparvathy, "An early warning system for elephant intrusion along the forest border areas", Current Science, vol. 104, pp. 1515–1526, 2013. View at Google Scholar.

[15]. R. Radha, K. Kathiravan, V. Vineeth, J. Sanjay and S. Venkatesh, "Prevention of monkey trespassing in agricultural field using application agricultural specific flooding approach in wireless sensor network"., 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Chennai, 2015, pp. 106- 111.

Safeguarding Agriculture: Using Deep Learning to prevent Animal Invasions

Lakshminatha Reddy Chinapureddy,[1, a)] Moulika Kalimisetti, Bhavana Chigurela, Anusha Utti, Chandana Tupakula[2,3,4,5 b)]

[1]*Assistant Professor, Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, Anantapur, India.*
[2,3,4,5] *Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology*

[a)]*lakshminathareddy.cse@srit.ac.in*,
[b)]*204g1a0559@srit.ac.in*
[c)]*204g1a0523@srit.ac.in*
[d)]*204g1a0514@srit.ac.in*
[e)]*214g5a0526@srit.ac.in*

Abstract. The menace of crop damage due to animal, attacks poses a significant threat to agricultural yields. With the expansion of cultivated land encroaching upon wildlife habitats, instances of crop raiding have escalated, exacerbating the human-wildlife conflict. Conventional mitigation measures employed by farmers have proven inadequate, and the impracticality of hiring guards to surveil crops deters a viable solution. Prioritizing Safeguarding crops from animal-induced destruction is imperative to guarantee the security of both people and animals, giving the welfare of the animals top priority by safely distracting them.
To address this challenge, a pioneering project is underway to develop an algorithm for wildlife detection. By leveraging image recognition technology, this algorithm aims to classify animals based on their visual profiles, enabling more efficient monitoring. The project entails installing cameras across the farm to surveil the surroundings continually throughout the day. By implementing this innovative approach, we endeavor to mitigate crop damage caused by animals while simultaneously promoting harmonious coexistence between humans and wildlife. This Project gives software to prevent Animal Invasions.

Keywords: Deep Learning, Transfer Learning Models, CNN (Convolutional Neural Networks), VGG16, VGG19.

INTRODUCTION

One of the major societal challenges affecting farming is the destruction of agricultural produce by wild animals. The disruption of wild animals by wild animals has always been a problem for ranchers. The harvest is threatened by a number of species, including deer, wild hogs, moles, elephants, monkeys, and others. The presence of these animals poses a significant threat as they can consume crops undetected by farmers, ultimately resulting in ruined harvests.

Their ability to roam freely across fields without detection exacerbates the risk of crop damage, highlighting the urgent need for effective monitoring and mitigation measures. As a result, the yield may suffer a great loss, and additional financial security will be needed to deal with the damage.

When utilizing his invention, Every farmer should understand that there are animals in the region as well, and that they must be shielded from any potential harm. An immediate solution should be found to this problem, along with a strong arrangement. As a result, our paper hopes find a solution to the central objective of the paper is to devise strategies aimed at preventing wild animals from encroaching upon agricultural fields in order to address this problem.

Additionally, it emphasizes the importance of protecting these animals by employing non-lethal methods, such as scare tactics, to deter them from the fields, rather than resorting to lethal means. Additionally, the charity works to protect people from attacks by animals. In order to build a monitoring and

deterrent system to safeguard crops from animal assaults, we are employing an integrated strategy that makes use of a number of deep learning techniques and algorithms to strengthen safety precautions against animal incursion, involving IOT, sensing units, communicating devices for preliminary actions, diverting animals, and alerting farmers. Review techniques for digital image-based animal detection are presented in this research.

Applications based on sensors are useful in a variety of real-world situations. Using such devices might minimize expenses, guarantee stability and dependability, facilitate remote monitoring with less energy use, and make it easier for farmers to obtain information.

LITERATURE SURVEY

Researchers dedicated their efforts to detecting the animals.

Wenling Xue, Ting Jiang [1]. This research proposes an there are many different manifestations of animal, human conflict throughout the nation, such as the threat posed by monkeys in urban areas and wild pigs that raid crops. One of the most important issues the world is currently experiencing is finding workable solutions to the conflict between humans and animals. This paper describes the use of a wireless sensor network based on UWB technology for intrusion detection deployment. Ultrawide band (UWB) signal features are automatically learned by feature analysis using a convolutional neural network. In the end, the SVM or Softmax classifier is used to classify people and animals.

Prashanth C Ravoor , Sudarshan T S B1 , and Krishnan Rangarajan2 [2]. This research presents a novel end-to-end design of a distributed cross-camera tracking system (a "Digital Border") based on computer vision for the purpose of detecting animal trespass using deep learning networks. In addition to sending out notifications when an animal incursion is detected, the system provides useful data like the intruders' species and number, their approximate location, and their last known direction of movement.

Dr.r.s. Sabeeniani, N.Deivanai, B. Mythili [3]. Protecting crops from animal harm and carefully rerouting animals away from agriculture are essential since both human and animal safety are equally important. In order to get over the aforementioned problems and accomplish our objective, we use machine learning (ML) to identify animals that are visiting our farm by utilizing the deep neural network concept, a subset of computer vision. In this project, the entire farm will be periodically monitored by means of a camera that records the surroundings continuously. Using a machine learning model, we are able to detect the animal when it enters and play the appropriate sounds to frighten it away. The various CNN libraries and ideas that were used to construct the model are listed in this study.

Fazil Mohammed, C.R.Ullas, C.M.Hema, S.K.Sonakshi [4]. Animal entering is a major crop yield issue that reduces farmer earnings and jeopardizes food security. This proposed paradigm presents machine learning and the Internet of Things as solutions to this problem. The machine algorithm is interfaced with the ESP8266 Wireless Fidelity module, Pi Camera, Buzzer, and LED, and is run on a Raspberry Pi. Machine learning methods such as Single Shot Detection and Region-based Convolutional Neural Network are essential for classifying animals and identifying objects in images. Based on the experiment results, the Single Shot Detection method outperforms the region-based Convolutional Neural Network approach.

Sanjay Santhanam, Sudhir Sidhaarthan B, Sai Sudha Panigrahi, Suryakant Kumar Kashyap [5]. Animals crossing the road unintentionally has remained to be a prominent cause of traffic fatalities over time. Cars find it challenging to notice the forest's inhabitants because of its winding, gloomy roadways. For truckers, blind spot zones provide challenges. This research proposes a model that can recognize animals and notify the driver. The system alerts the driver to an approaching animal by sounding a three-second alert when the machine recognizes an object as an animal. Since the information is openly accessible, a large variety of species are still recognized by this model. The accuracy of the neural network algorithm model is 91%.

Bindu D et al [6]. Explains the preservation of agricultural fields has been the primary focus and a challenging subject in this essay. Over the years, the crop field has been repeatedly attacked by animals from the protected areas (PAs), making its protection a top priority. The methods currently in use are ineffective, so in this article we'll present a workable method to fend them off: we'll build a system that analyses animal behaviour, recognises the animal, and produces a sound that agitates the animal in addition to notifying the designated person via message. Additionally, we offer a multi-class categorization by accurately identifying species and displaying a zero false alarm rate.

Krishnamurthy B, Divya M [7]. In this project explains how agriculture provides a range of raw materials for industry in addition to meeting people's food demands. However, because to animal intervention in

agricultural areas, a major loss of crops is expected. A crop's vulnerability to wild animals exists. As a result, it's critical to keep an eye out for any animal presence nearby. After then, a number of devices should be activated to drive away the dangerous animals. We suggest a strategy to keep wild animals away from farms.

Detecting animal attack from the outside of farms is the main use of operational amplifier circuits. The recommended monitoring program's goal is to provide early warning of possible wild animal ingress and danger. To protect your property and crops, the Solar Electric Fencing system is a cutting-edge alternative to conventional fencing methods.

Kshama S. Bhise1 [8]. In this work, explains how the initiative is used to find animals in national parks or wildlife reserves. This project makes use of an RFID (Radio Frequency Identification Device) module and a GSM (Global System Mobile) modem. Radio waves are used in a technology called "radio frequency identification," or "RFID," to wirelessly broadcast an object's or person's identity (represented by a unique serial number). It is included in the broad category of automatic identification technologies. This paper is used to trace an animal's whereabouts in a national park or wildlife reserve. For this, this paper makes use of a zigbee and an RFID module. These SMS will be sent to government officials or forest officers, containing the area where the animals are observed.

Prof Abhinav, V. Deshpandey [9]. This study explains the suggested strategy for using widely used wired network devices to protect farms from wild animals. This strategy is used in conjunction with other established techniques to increase the effectiveness of farm animal protection. A functioning amplifier circuit's main use is to detect animal infiltration from the outside of farms. The goal of the suggested monitoring programme is to alert people in advance to potential wild animal intrusion and damage.

S. R. Chourey , P. A. Amale [10]. The approach in order to protect farmers' crops from wild animals, this paper reviews a comprehensive technical solution that combines wireless sensor networks (WSN) and the Internet of Things (IOT). It has every kind of sensor, controller, and actuator needed for WSN, with the Raspberry Pi serving as the system's central component.

PROPOSED SYSTEM

In this project, we suggest a unique methodology that blends Convolutional Neural Networks (CNN) with the VGG16 model as a key component Setting up a network of surveillance cameras strategically placed around the farm. These cameras will continuously capture images of the surrounding throughout the day. The system will be designed to send email notifications to farmers, alerting them to the presence of intruders in real- time, allowing for prompt response.

MATERIAL AND METHODOLOGY

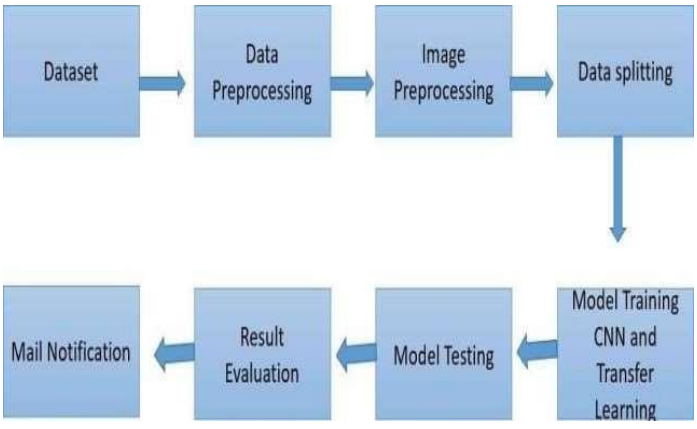


FIGURE 1: Block diagram for suggested approach

Methodology Overview

Fetching Data

Fetch, in the realm of computing, entails the retrieval of data through software, scripts, or hardware. Subsequently, once retrieved, this data can either be transferred to another location or presented on a screen for user interaction.

Dataset

Loading the Images Dataset involves retrieving a collection of images representing various aspects. Researchers typically employ data loading techniques using programming languages like Python and libraries such as TensorFlow or PyTorch. This dataset is essential for training and evaluating machine learning models, especially those designed for tasks such as quality assessment or classification in the context of image analysis.

Data Cleaning

The laborious process of preparing data for analysis is known as data cleaning. This is accomplished by eliminating or altering data that is inconsistent, absent, superfluous, repetitious, or handled in an improperly structured manner. However, it goes beyond merely arranging rows or deleting information to accommodate new data. Data cleaning demands substantial effort. Its significance cannot be overstated, particularly in fostering a data-driven culture and ensuring precise forecasts. It entails correcting grammatical and syntactic mistakes and standardizing data sets. Making corrections for errors like blank fields finding redundant data points.

Data Preprocessing

Many machine learning algorithms necessitate data to adhere to specific formatting guidelines, thus datasets typically require preparation before yielding valuable insights. Some datasets contain missing, invalid, or challenging values for algorithms to process effectively. Missing or invalid data impedes algorithmic use, resulting in less accurate or misleading outcomes. While some datasets are relatively clean but require shaping, others lack essential business context, emphasizing the necessity for feature enrichment. Effective data preparation entails the creation of clean and meticulously curated data, ultimately leading to more practical and accurate model outcomes.

Model Training

A dataset that an ML algorithm utilizes to become trained is called a training model. It consists of matched sets of input data that influence the output along with sample output data. This model is used to input data into the algorithm by comparing the processed output with the sample output. The model is modified through this iterative process called "model fitting," taking into account the correlation findings. The link between the training and validation datasets is a crucial component in ensuring the accuracy of the model. In order to help an ML algorithm find patterns and provide predictions, model training entails feeding it data.

System Implementation

A. Data Prepocessing

Preparing raw data for analysis is known as data pre-processing, and it's a crucial the initial stage of developing a machine learning model. Raw data is frequently unclean and poorly formatted, requiring cleaning and formatting before any additional processes can be carried out. This process is crucial for ensuring that the data is in a usable state for training and testing machine learning algorithms.
Includes the following steps:

- *Obtaining the dataset***:** In image reconstruction using CAEs, the encoder is responsible for transforming input images into a compressed latent representation. This encoded representation contains essential

features extracted from the input, facilitating efficient storage and subsequent reconstruction during the decoding phase. The encoder's role is crucial in capturing meaningful information for faithful image representation.

- *Importing Libraries:* Typically, the initial step involves importing the necessary libraries participating in the program. A library is a collection of modules that may be used and called from within an application
- *Importing Datasets:* First, finding the CSV file's directory is necessary because many datasets are in CSV formats and then use the read.csv method in R- Studio to read it.
- *Finding Missing Data :* Following data pre-processing, the subsequent step involves addressing missing data within the datasets. The effectiveness of machine learning models can be significantly impacted by missing data. Implementing strategies for handling missing values present in the dataset is imperative.
- *Encoding Categorical Data :* Whenever we have a text data need do apply text processing and clean it. In this text preprocessing first step punctuation symbols removal. First step want to remove some punctuation removal there is no using this symbol and get create some high dimensionality. In the second step, we can remove stop words from the dataset using NLTK and then proceed with tokenization. Stop words are common words that often carry little meaning and can be removed to improve text analysis accuracy. In this step split the sentence into words and apply stemming. Stemming is nothing but convert the word into base form for example beautiful, beauty, be stain the base form is beauty.
- *Dataset Splitting into Training and Test Sets: The next important step is to divide our dataset into two separate sets: a training set and a test set.* The purpose of this division is to train our machine learning models on the Training set, then measure their predicted accuracy by assessing how well they perform on the test set. It's essential to ensure that our model performs well not only on the Training set but also on unseen data, represented by the test set. This practice helps prevent over fitting, when a model fails to generalize to new data despite doing remarkably well on the Training set. By striving for robust performance on both the Training, Test sets, we aim to develop machine learning models capable of accurately predicting outcomes across different datasets.

B. Normalizing Numeric data

In data preprocessing for machine learning purposes, normalization is a common technique utilized. Its goal is to maintain the intrinsic variations in the ranges of values while standardising the values of numerical columns within a dataset to a scale. This process becomes necessary when features exhibit disparate ranges, ensuring that each feature contributes equally during model training.

Normalization in the context of machine learning refers to transforming data to a unit sphere or, alternately, translating it into a specific range, usually [0,1]. For several machine learning techniques, normalization and standardization are helpful, particularly when Euclidean distance is being used.

C. *Deep Learning Algorithms*

A. CNN

The ability of artificial intelligence to link human and computer skills has advanced significantly. To pull off incredible accomplishments, both beginners and specialists concentrate on different aspects of the subject. The domain of computer vision is among these many. Enabling machines to perceive and comprehend the world similarly to humans is the aim of this field.

A plethora of activities, including picture and video identification, recommendation systems, natural language processing, image analysis, and classification, will be possible for these robots to accomplish with the use of this enormous amount of data. Deep Learning has made significant strides in computer vision throughout time, mostly thanks to one particular algorithm called CNN.

Send the data straight to the model after image processing and data splitting are finished; CNN architecture must first be defined. Initial import model sequentially, followed by an input layer that fixes the input shape images and provides the activation function. The input layer is then added, where 32 filters, a (5,5) kerner size, max pooling, and dropouts are added.

This dropout aids in preventing over fitting in our model. Added two input layers with 64 and 128 filters, the same max pooling, a 5.5% kernel size, and dropouts using the Relu activation function once more.

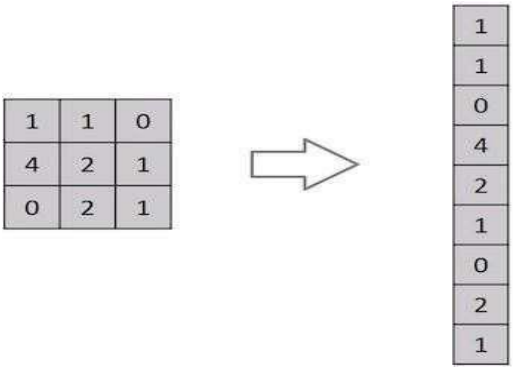


FIGURE 1: Flattening of a 3x3 image matrix into a 9x1 vector

For incredibly simple binary images, the technique could be able to predict the class with an average precision score, but it performs badly or not at all for complex images with relationships between individual pixels. The image measures five inches in height, five inches in width, and one number of channels (such as RGB).

- *Pooling layer*: The Pooling Layer's function is to reduce the spatial extent of the Convolved Feature, just like the Convolutional Layer does. By doing this, the dimensionality of the data will be minimized and less processing power will be required to handle it. Additionally useful for creating positionally and rotationally invariant dominating features, which sustains the model's training process.

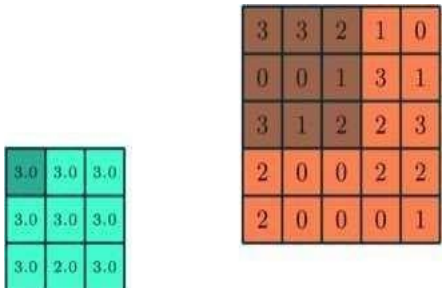


FIGURE 3: Pooling layer

Ordinary pooling and severe pooling are the two categories of pooling. Max Pooling delivers the largest value from the region covered by the kernel, whereas Average Pooling yields the mean of all the values from the image covered by the kernel.

B. VGG16

In the field of image recognition, K. Simonyan and A. Zisserman's convolutional neural network, known as VGG16, is a seminal model. It sprang to prominence after achieving an astounding using the ImageNet dataset, which contains over 14 million images distributed over 1000 categories, the test accuracy was 92.7%

top-5, during the ILSVRC-2014 competition. What sets VGG16 apart is its architectural innovation: the replacement of larger kernel-sized filters in the initial layers with multiple consecutive 3x3 kernel-sized filters.

This design choice allows for deeper network architectures without overly increasing the number of parameters, striking an optimal balance between model complexity and computational efficiency. By leveraging this configuration, VGG16 excels at capturing intricate features and patterns within images, enabling robust and accurate classification. The cascading arrangement of smaller filters enhances the model's ability to extract meaningful features while maintaining a deep architecture.

VGG16's success underscores the significance of architectural advancements in deep learning, paving the way for subsequent models and further advancements in computer vision research.

C. VGG19

Neural Convolution The massive dataset that contained more than a million photos was used to train VGG-19. ImageNet compilation. This neural network building architecture boasts an impressive depth of 19 layers, making it proficient at classifying images into one of 1000 object categories. These categories include a broad range of products, from commonplace goods like mouse, pencils, and keyboards to different animals. Through its rigorous training process, VGG-19 has acquired a nuanced understanding of features present in images across this broad spectrum. When processing input images, VGG-19 expects a fixed size of 224x224 pixels with RGB channels, resulting in a matrix shape of (224,224,3).

In essence, VGG-19 serves as a powerful tool for image classification tasks, leveraging its deep architecture to extract intricate features and make accurate predictions. The layers within the VGG-19 model are structured to progressively abstract and refine features from the input image, enabling it to discern intricate patterns and classify images with high accuracy. After being trained on a million photos from the ImageNet database, a convolutional neural network with 19 layers named VGG-19 was able to identify 1000 different objects, including a mouse, keyboard, pencil, and different animals.

Consequently, a vast array of image-rich feature representations have been trained into the network. An RGB fixed-size picture with dimensions of (224 * 224) was the network's input, indicating that the matrix's structure was (224,224,3).

1. As a single preprocessing step, the mean RGB value of every pixel across the whole training set was eliminated.
2. They used kernels with a size of (3 * 3) and a stride size of one pixel to cover the entire image. Spatial padding was used to preserve the image's spatial resolution.

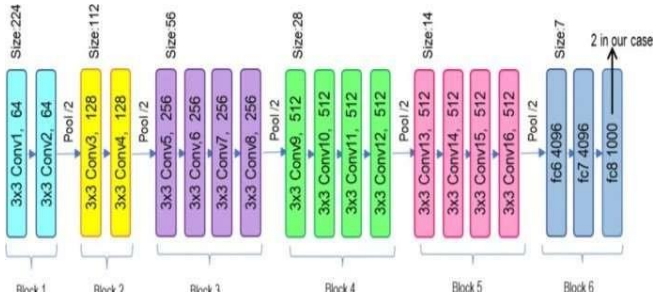


FIGURE 4: VGG19 Architecture

EXPERIMENTS AND RESULTS

The suggested model will train the elephant, boar, and monkey picture collection by creating Convolutional Neural Network and transfer learning models. The saved model will be run on the driver code in order to compare the learned images with the new test images from the live capture. By means of speakers,

an unpleasant noise is generated. If one of the trained animals is found during the live capture in order to scare it away.

Various test images are given, and the model's accuracy is confirmed by identifying their classes.

CNN ,VGG Model Graphs

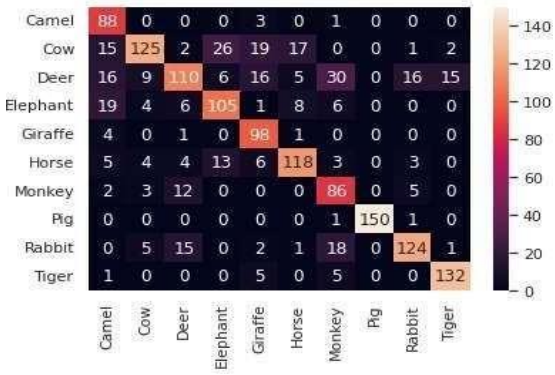


FIGURE 5: Confusion Matrix of CNN

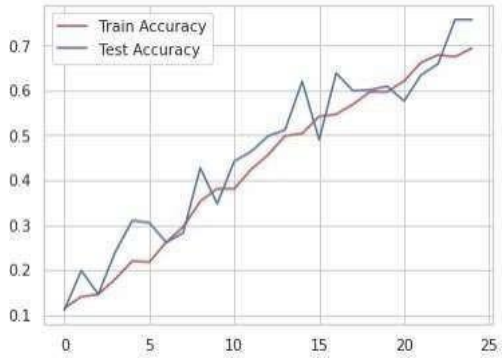


FIGURE 6: Train and Test Accuracy of CNN

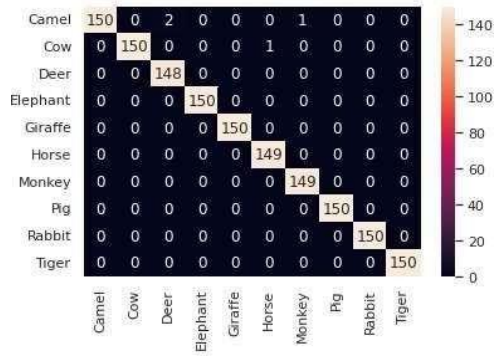


FIGURE 7: Confusion Matrix for VGG16

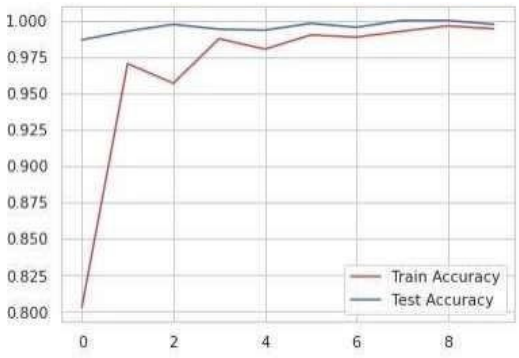


FIGURE 8: Train and Test Accuracy for VGG16

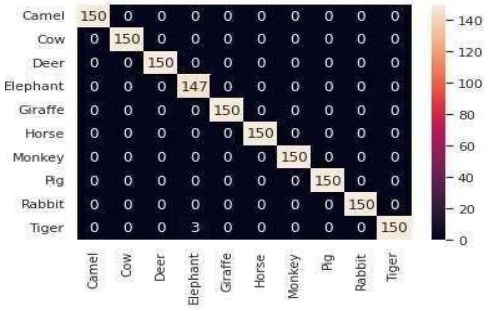


FIGURE 9: Confusion Matrix for VGG19

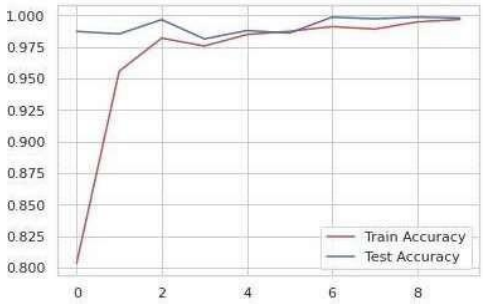


FIGURE 10: Train and Test Accuarcy for VGG19

By using CNN got 75% accuracy we can check it from confusion metric also. Seeing this confusion matrix which class images are correctly classified and which class images Misclassified we can identify easily. Some camel images classified into cow, deer and elephant and some deer images classify into monkey and rabbit.

These all are the miss classified data points. We need to decrease these are miss classified data points. So, we can see train and test data accuracy each class.

By using VGG models we can predict animals with more accuracy then CNN. By using VGG16 predict animals with accuracy of 96% and VGG19 with accuracy of 98%.

CONCLUSION

In the current day, The problem of wild animals eating crops has become quite significant. Put differently, every farmer should utilize the crop productivity that he or she should be conscious of and mindful of the reality that animals are sentient beings who require protection from potential harm. It must be dealt with right away and successfully. As a result, this initiative has a great deal of social significance because it will free farmers from the needless labor associated with field protection, assist them in safeguarding their farms and preventing them form suffering large financial damages.

ACKNOWLEDGMENTS

We extend our sincere gratitude to the contributors and collaborators involved in the development of this project aimed at safeguarding agriculture to prevent animal invasions. Special thanks are extended to the teams and researchers who pioneered the utilization of neural networks, particularly Convolutional Neural Networks (CNN) and VGG Model, in the realm of agriculture prevention. Their ground-breaking research serves as the basis for our strategy, which enables us to use machine learning to classify animal images more accurately and effectively. The knowledge and combined efforts of those committed to developing agricultural and animal prediction technology made this initiative possible.

REFERENCES

1. Xue, Wenling, Ting Jiang, and Jiong Shi. *Animal intrusion detection based on convolutional neural network*. 2017 17th international symposium on communications and information technologies (ISCIT). IEEE, 2017.
2. Ravoor, Prashanth C., T. S. B. Sudarshan, and Krishnan Rangarajan. *Digital Borders: Design of an Animal Intrusion Detection System Based on Deep Learning*. *International Conference on Computer Vision and Image Processing*. Singapore: Springer Singapore, 2020.

3. Sabeenian, R., N. Deivanai, and B. Mythili. *Wild animals intrusion detection using deep learning techniques*. *Int. J. Pharm. Res* 12.4 (2020): 1053-1058.
4. Balakrishna, K., et al. *Application of IOT and machine learning in crop protection against animal intrusion*. Global Transitions Proceedings 2.2 (2021): 169-174.
5. Santhanam, Sanjay, et al. *Animal detection for road safety using deep learning*. 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA). IEEE, 2021.
6. Bindu D et al. *Prevention of Wild Animals Entering Into the Agriculture Fields*. International Journal of Engineering, Basic sciences, Management & Social studies, Volume 1, Issue 1, May 2017.
7. Krishnamurthy b et al. *Solar Fencing Unit and Alarm for Animal Entry Prevention*. International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137, Volume – 02, Issue – 05, May – 2017, PP – 128-135.
8. Kshama S.Bhise. *Wildlife Animal Tracking Using RFID and GSM Technology*. International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016 ISSN 2229-5518.
9. V. Deshpande. *Design and Implementation of an Intelligent Security System for Farm Protection from Wild Animals.* International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611.
10. S. R. Chourey, P. A. Amale et al, IETE Zonal Seminar IOT Based Wireless Sensor Network for Prevention of Crops from Wild Animals -2017 Special Issue of International Journal of Electronics, Communication & Soft Computing Science and Engineering, ISSN: 2277-9477.
11. Santhiya, S., Dhamodharan, Y., Priya, N. E.K., Santhosh, C. S., & Surekha, M. (2018). A smart farmland using Raspberry Pi crop prevention and and animal intrusion detection system. Int. Res. J. Eng. Technol, *5*(3), 3829- 3832.
12. S. J. Sugumar and R. Jayaparvathy, An early warning system for elephant intrusion along the forest border areas, Current Science, vol. 104, pp. 1515–1526, 2013. View at Google Scholar.
13. R. Radha, K. Kathiravan, V. Vineeth, J. Sanjay and S. Venkatesh, Prevention of monkey trespassing in agricultural field using application agricultural specific flooding approach in wireless sensor network., 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Chennai, 2015, pp. 106-111.
14. IshaDua, Pushkar Shukla, Ankush Mittal. A vision based human - elephant collision detection system. IEEE International Conference on Image Processing. (25 Febraury, 2016) pp: 225-229.
15. Sheela.S, Shivaram. K., Chaitra. U, Kshama. P, Sneha. K, Supriya. K. Low Cost Alert System for Monitoring the Wildlife from Entering the Human Populated Areas Using IOT Devices. International Journal of Innovative Research in Science, Engineering and Technology. (10 May, 2016) Vol. 5, Special Issue 10, May 2016.
16. Abhinav and Deshpande, Design and implementation of an intelligent security system for farm protection from wild animals, International journal of science and research,10(2),2016, 300-350.
17. Srinivasa Reddy Gudibandi M.Tech Student, JNTUA, Anantapuramu, A.P, India, Mr. M. Amarnath , Design of Smart Surveillance System using PIR and Ultrasonic Sensor , International Journal & Magazine of Engineering, Technology, Management and Research, November 2015.

AIP Conference Proceedings

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)
5th & 6th April 2024
Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Lakshminath Reddy chinapureddy, of SRIT, Anantapur has participated and presented a paper entitled "Safeguarding Agriculture: Using Deep Learning to prevent Animal Invasions" with Paper ID: ICIAET-P185 in ICIAET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

PARTICIPATION CERTIFICATE

AIP Conference Proceedings

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)
5th & 6th April 2024
Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Moulika Kalimisetti, of SRIT, Anantapur has participated and presented a paper entitled Safeguarding Agricultural : Using Deep learning to prevent Animal Invasions with Paper ID: ICIAET-P185 in ICIAET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)
5^{th} & 6^{th} *April* 2024
Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Bhavana chigurela, of SRIT, Anantapur has participated and presented a paper entitled Safeguarding Agriculture: Using Deep Learning to prevent Animal Invasions with Paper ID: ICIAET-P185 in ICIAET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)
5th & 6th April 2024
Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Anusha Utti, of SRIT, Anantapur has participated and presented a paper entitled Safeguarding Agriculture: Using Deep learning to prevent Animal Invasions with Paper ID: ICIAET-P185 in ICIAET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)
5^{th} & 6^{th} *April* 2024
Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Chandana Tupakula, of SRIT, Anantapur has participated and presented a paper entitled Safeguarding Agriculture : Using Deep Learning to prevent Animal Invasions with Paper ID: ICIAET-P185 in ICIAET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor