VISVESVARAYATECHNOLOGICALUNIVERSITY
JNANA SANGAMA, BELAGAVI-590018

A MINIPROJECT REPORT ON

"CLASSIFICATION OF ANIMALS USING ML ALGORITHMS"

Submitted in partial fulfillment for the award of degree of Bachelor of Engineering in Information Science & Engineering during the year 2023-24

Submitted by

AISHWARYA MN (4MH21IS025)

ANAGHA ANANTH(4MH21IS029)

DISHANTH H R (4MH21IS042)

NAIDHILE S (4MH21IS052)

Under the Supervision of

Prof. Dharmaraj B
Asst. Professor,
Dept. Of ISE, MIT Mysore.

2023-24

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING, MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE BELAWADI, SRIRANGAPATNA TALUK, MANDYA-571477

DEPARTMENTOF INFORMATION SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

MANDYA-571477

CERTIFICATE

Certified that the project work entitled **"CLASSIFICATION OF ANIMALS USING ML ALGORITHMS"** is a bonafide work carried out by **Aishwarya M N (4MH21IS005)**, **Anagha Ananth (4MH21IS008)** , **Dishanth H R (4MH21IS028)** and **Naidhile S(4MH21IS048)** in the partial fulfillment for the award of degree of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya TechnologicalUniversity, Belagavi during the academic year 2023-24. It is certified that all corrections/suggestions indicated have been incorporated in the report. The project report has been approved as it satisfies the academic requirements with respect to the Project work prescribed for Bachelor of Engineering Degree

Internship Coordinator

Dr. Dharmaraj B
Assistant Professor
Dept. of ISE, MITM

Head of the Department

Dr. Sharath Kumar Y H
Professor & Head
Dept. of IS&E, MITM

EXAMINER	SIGNATURE
1)	
2)	

ACKNOWLEDEGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be “Success is the abstract of hard work and perseverance, but steadfast all is encouragement guidance”. I acknowledge all those whose guidance and encouragement served as a beacon light and crowned my efforts with success.

I acknowledge and express my sincere thanks to our beloved Principal **Dr. B.G Naresh Kumar,** MIT Mysore who is the source of inspiration.

I would like to express my deepest sense of gratitude towards **Dr. Sharath Kumar Y H,** Professor, Head of the Department, ISE, MIT Mysore for his valuable suggestions, support and encouragement.

I would also express my gratitude to my Mini project coordinator **Dharmaraj B** , Assistant Professor, Department of Information Science and Engineering, MIT Mysore for his kind support throughout the Mini project.

I would also thank all other teaching and non-teaching staffs of the Information Science Department who has directly or indirectly helped me in completion of the seminar work. My thanks and appreciations also go to my family and friends who have willingly helped me out with their abilities.

AISHWARYA M N (4MH21IS005)

ANAGHA ANANTH(4MH21IS008)

DISHANTH H R(4MH21IS028)

NAIDHILLE S (4MH21IS048)

ABSTRACT

This project develops a sophisticated animal classification system by leveraging various machine learning algorithms for precise identification and categorization of zoo animals. The system translates a range of animal features into actionable classifications, such as species identification, using advanced data analysis techniques. By analyzing the spatial relationships and characteristics of animal attributes, the system ensures high accuracy and responsiveness in identifying different zoo animals.

The design of the system prioritizes efficiency and scalability, offering a robust alternative to traditional classification methods. Real-time performance is achieved through optimized processing of the input data, with features such as adjustable sensitivity and configurable classification thresholds to accommodate different user requirements and environmental conditions. The system's flexibility supports adaptation to various scenarios, from simple species identification to complex ecological studies, proving its practicality and effectiveness.

Furthermore, the project explores innovative applications by integrating animal classification with emerging technologies like computer vision and artificial intelligence. This integration aims to create more accurate and efficient classification models, enhancing the interaction between researchers and digital data. By enabling precise classification within various ecological settings, the system contributes to the advancement of zoological studies and biodiversity research. This approach not only improves the usability of modern classification systems but also sets a foundation for future developments in animal identification, paving the way for more seamless and interactive ecological research experiences.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 Overview

Animal classification in the context of machine learning involves the use of algorithms to identify and categorize various zoo animals based on their characteristics and attributes. Traditionally, identifying and classifying animals required manual observation and expert knowledge, which was time-consuming and prone to errors. The advancement in machine learning algorithms and data processing techniques has enabled the automation of this task, making it more efficient and accurate.

The concept revolves around training machine learning models using labeled datasets containing various features of animals, such as their physical characteristics, habitat information, and other relevant attributes. Once trained, these models can classify new, unseen data with high precision, making them valuable tools for zoological studies, wildlife conservation, and ecological research. In a typical animal classification system, data is collected from various sources, including images, videos, and sensor data. This data is then pre-processed to extract meaningful features, which are fed into machine learning algorithms like Random Forest, Support Vector Machines, or Neural Networks. These algorithms learn the underlying patterns in the data and develop a model that can predict the class of an animal based on its features.

The design of such a system emphasizes accuracy, scalability, and real-time performance. Optimized algorithms and efficient data processing pipelines ensure that the system can handle large datasets and provide quick, accurate classifications. Adjustable parameters and configurable thresholds allow the system to adapt to different environments and user requirements, enhancing its practicality and effectiveness.Furthermore, the integration of machine learning with other technologies like computer vision and big data analytics opens up new possibilities for advanced animal classification systems. This integration enables more precise and comprehensive analysis, providing deeper insights into animal behavior, habitat usage, and biodiversity.

By leveraging these advanced technologies, the animal classification system not only improves the efficiency of traditional methods but also sets a foundation for future developments in zoological research and conservation. It paves the way for more seamless and interactive research experiences, contributing to the broader understanding and preservation of wildlife.

1.2 Project Scope

Integrating animal classification into machine learning (ML) projects can significantly enhance various applications in fields like wildlife conservation, agriculture, and even pet care. By employing ML algorithms, we can create intelligent systems that automatically identify and categorize animal species based on images or other data.

For instance, consider a wildlife conservation project where you want to monitor and protect endangered species. By using ML algorithms for animal classification, you can analyze images captured by cameras placed in natural habitats. The system can automatically identify and track different animal species, providing valuable data on their populations and behaviors. This information can help conservationists make informed decisions about protective measures and habitat management.

In agricultural settings, ML-based animal classification can be used to monitor livestock health and behavior. For example, by analyzing video feeds from farm cameras, the system can identify signs of illness or distress in animals. This allows farmers to take timely actions to address health issues, improving overall animal welfare and productivity.

Similarly, in the context of pet care, ML algorithms can be used to develop apps or devices that help pet owners identify and classify different breeds or species. For instance, a pet identification app can analyze a pet's image and provide information about its breed, helping owners with care recommendations and breed-specific tips.

Incorporating ML-based animal classification into these projects adds a layer of automation and intelligence, enabling more efficient and accurate data collection and decision-making.

1.3 Existing System

For animal classification using machine learning (ML) algorithms, the existing systems often integrate various technologies and methodologies to accurately identify and categorize animal species. Here's a detailed look at how these systems typically operate:

1. **Data Collection:** The initial step involves gathering a large dataset of images or videos of animals. This data can come from various sources such as camera traps in the wild, agricultural surveillance cameras, or online databases containing images of different animal species.

2. **Data Annotation:** The collected data is annotated with labels indicating the species or type of animal. This step may involve manual labeling by experts or using semi-automated tools to classify images based on predefined categories.

3. **Feature Extraction:** ML models rely on features extracted from the images, such as color, texture, and shape. Advanced techniques involve using convolutional neural networks (CNNs) to automatically extract features from raw images, which helps in distinguishing different species.

4. **Model Training:** The annotated data is used to train ML algorithms. Common algorithms include CNNs for image classification and recurrent neural networks (RNNs) for video analysis. The training process involves feeding the model with labeled data and adjusting its parameters to minimize prediction errors.

5. **Model Validation and Testing:** After training, the model is validated and tested using separate datasets to evaluate its accuracy and generalization capabilities. This helps ensure that the model performs well on unseen data and reduces overfitting.

6. **Prediction and Classification:** Once trained and validated, the ML model can classify new, unlabeled images or videos. For instance, a camera trap image can be analyzed in real-time to identify and categorize the animal present.

7. **Integration and Deployment:** The ML model is integrated into applications or systems where it can be used for practical purposes. Examples include wildlife monitoring systems that track endangered species, agricultural systems that monitor livestock, or apps that help pet owners identify different breeds.

8. **Continuous Improvement:** The system often includes mechanisms for continuous learning. New data can be used to retrain the model and improve its accuracy over time. This process involves periodically updating the model with new annotations and features to adapt to changes in animal behavior or appearance.

By combining these technologies and methodologies, existing animal classification systems enable accurate and efficient identification and tracking of various animal species. These systems have applications in wildlife conservation, agriculture, pet care, and more, providing valuable insights and enhancing decision-making across different domains.

1.4 Proposed System

The proposed system for animal classification will use high-resolution cameras, drones, and remote sensors to capture detailed images of animals in various environments. It will integrate advanced machine learning models, including deep CNNs and transfer learning, to improve species identification. Big data analytics will process historical and real-time data to enhance predictions, while sensor fusion will combine visual, thermal, and acoustic data for better accuracy. The system will feature a user-friendly interface with real-time updates and customizable alerts, and it will be designed for scalability and continuous improvement. This approach will support wildlife monitoring, agricultural management, and pet care with precise and reliable animal classification.

CHAPTER 1

Literature Survey

2.1 Survey Papers

2.1.1 PAPER 1

TITLE : Animal Classification System: A Block Based Approach

AUTHORS : Y H Sharath Kumar, Manohar N, Chethan H K

YEAR OF PUBLICATION : 2015 (Procedia Computer Science)

EXPLANATION : This research presents a novel method for classifying animals in images using a block-based approach. The proposed system employs a graph-cut based segmentation technique to remove the background from the images. Following segmentation, the images are divided into blocks, from which color texture moments are extracted. The classification is performed using a fusion of probabilistic neural networks (PNN) and K-nearest neighbors (KNN) algorithms.

The study conducted experiments on a dataset consisting of 25 classes of animals with 4000 images. The results demonstrate that the K-nearest neighbors classifier achieves high performance in classifying the animal images. The proposed method shows significant improvement in classification accuracy due to the effective use of segmentation and feature extraction techniques.

2.1.2 PAPRE 2

TITLE : Animal Classification System Based on Image Processing & Support Vector Machine.

AUTHORS : A. W. D. Udaya Shalika, Lasantha Seneviratne

YEAR OF PUBLICATION : 2016

JOURNAL: Journal of Computer and Communications
Explanation:

EXPLANATION : The project focuses on developing a system for animal researchers and wildlife photographers, utilizing image processing and machine learning techniques to automatically detect and classify animals. The system uses a microcontroller-controlled camera, image processing algorithms, and Support Vector Machines (SVM) to identify animals in captured images. The system improves upon existing wildlife cameras by automatically recognizing specific animals and capturing images based on predefined criteria, overcoming challenges such as detecting animals from different angles and improving image quality under various conditions.

2.1.3 PAPER 3

TITLE : Animal species classification using machine learning techniques.

AUTHORS : Fahad Alharbi, Abrar Alharbi, and Eiji Kamioka

YEAR OF PUBLICATION : 2019 (MATEC Web of Conferences)

EXPLANATION : The proposed system focuses on the identification and classification of animal species, specifically distinguishing between predator and pet animals, using machine learning techniques. The method utilizes image processing to extract features, particularly from the eyes and ears, to classify animals. A dataset was created containing images of ten animals, and machine learning classifiers such as Support Vector Machine (SVM) and Multilayer Perceptron (MLP) were used to evaluate the accuracy of the classifications. The evaluation results showed that the MLP classifier achieved an accuracy of 82%, while the SVM classifier achieved 78%, indicating the effectiveness of the proposed method for animal species classification. This system aims to enhance safety for people in outdoor areas by providing real-time detection of potentially dangerous animals.

2.2 Survey Findings

- · **Alharbi et al. (2019)**, in their study published in the MATEC Web of Conferences, present a system focused on distinguishing between predator and pet animals. The method employs image processing to extract features, particularly from the eyes and ears, and utilizes machine learning classifiers such as Support Vector Machine (SVM) and Multilayer Perceptron (MLP). Their evaluation results show that the MLP classifier achieved an accuracy of 82%, while the SVM classifier achieved 78%, indicating the effectiveness of the proposed method for animal species classification.
- · **Kumar et al. (2015)**, in their paper published in Procedia Computer Science, introduce a novel block-based approach for classifying animals in images. Their system uses a graph-cut based segmentation technique to remove backgrounds, divides images into blocks, and extracts color texture moments. Classification is performed using a fusion of Probabilistic Neural Networks (PNN) and K-nearest Neighbors (KNN) algorithms. Their experiments on a dataset of 25 animal classes with 4000 images show that the KNN classifier achieves high performance, demonstrating significant improvements in classification accuracy due to effective segmentation and feature extraction techniques.
- · **Shalika and Seneviratne (2016)**, in their research published in the Journal of Computer and Communications, focus on developing a system for animal researchers and wildlife photographers. Their system utilizes image processing algorithms and SVM to automatically detect and classify animals using a microcontroller-controlled camera. This system improves upon existing wildlife cameras by recognizing specific animals and capturing images based on predefined criteria, addressing challenges such as detecting animals from different angles and improving image quality under various conditions.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Stakeholders

Stakeholders are vital to the successful development and implementation of the system, as they have diverse interests and needs that must be considered. Key stakeholders include meteorologists, who need precise data and advanced analytics for accurate weather forecasts; government agencies, which rely on forecasts for public safety, disaster management, and policy-making; farmers and agricultural businesses, who use weather data for crop planning and optimization; transportation and logistics companies, which require dependable weather information for safe and efficient operations; and the general public, who benefit from timely updates for their daily activities and safety. Additionally, technology providers and developers are crucial for designing, developing, and maintaining the IoT infrastructure and analytics platform. Addressing the needs of these stakeholders is essential for ensuring the system provides accurate, timely, and actionable weather information, thereby enhancing its effectiveness and impact.

3.2 Functional Requirements

Functional requirements for an animal classification system using ML algorithms include:

1. **Data Input:** Accept and process image or video data from various sources.
2. **Image Preprocessing**: Perform operations like resizing and normalization to prepare data for analysis.
3. **Feature Extraction:** Use machine learning models, such as CNNs, to automatically extract relevant features from images.
4. **Model Training:** Train ML algorithms on labeled datasets to recognize and classify animal species.
5. **Model Validation:** Evaluate model performance with separate validation datasets to ensure accuracy.

6. **Real-Time Classification:** Enable the system to classify images or videos in real-time or near-real-time.

7. **User Interface:** Offer a user-friendly interface to display classification results and data visualizations.

8. **Customizable Alerts:** Provide notifications for specific classification results or detected patterns.

9. **Data Management:** Manage and store classified data, training sets, and model outputs.

10. **Integration:** Integrate with other systems or applications for broader use.

3.3 Non-Funtional Requirements

These non-functional requirements ensure that the system performs well, remains reliable, and meets user needs effectively.Non-functional requirements for an animal classification system using ML algorithms include:

1. **Performance**: Ensure high-speed processing and real-time classification with minimal latency.
2. **Scalability**: Support the system's growth to handle larger datasets and more complex models without performance degradation.
3. **Reliability**: Maintain consistent and accurate operation, with minimal downtime or errors.
4. **Availability**: Ensure the system is available and operational 24/7, with high uptime and quick recovery from failures.
5. **Usability**: Provide an intuitive and user-friendly interface for ease of use by various stakeholders.
6. **Maintainability**: Design the system to be easily updated and maintained, with clear documentation and modular components.
7. **Flexibility**: Allow for adaptation to new algorithms, data sources, and requirements with minimal disruption.
8. **Security**: Protect sensitive data and ensure secure access through authentication and authorization mechanisms.
9. **Privacy**: Comply with data protection regulations to safeguard user and data privacy.
10. **Compatibility**: Ensure integration with existing systems and compatibility with various data formats and platforms.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Analysis

System analysis for the Animal Classification System Using ML involves a thorough examination of existing processes, data flow, and technological infrastructure to determine the system's requirements and design an effective solution. The analysis starts with evaluating current methods for animal image and video collection, feature extraction, and classification, identifying any limitations and areas for enhancement. Key stakeholders, including wildlife researchers, conservationists, farmers, and technology developers, are consulted to understand their needs and expectations. The analysis highlights the necessity for a robust data acquisition process, using various imaging sources like cameras and drones. It also underscores the importance of implementing advanced ML algorithms for accurate classification, and ensuring efficient data management, model training, and real-time processing capabilities. Additionally, the analysis focuses on the need for an intuitive user interface, integration with existing systems, and maintaining data security and privacy.

4.2 High Level Design

High-Level Design for Classification of animals using ML algorithms.

1. System Architecture

1.1 Data Acquisition Layer:

- Sources: CSV files containing animal data (e.g., images represented as pixel values, labels).
- Data Loading: Reading and importing CSV files into the system.

1.2 Data Preprocessing Layer:

- Data Cleaning: Handling missing values, outliers, and duplicates.
- Data Transformation: Converting raw data into a suitable format (e.g., normalizing pixel values, encoding labels).
- Data Splitting: Dividing data into training, validation, and test sets.

1.3 Feature Extraction Layer:

- Feature Engineering: Creating additional features if necessary (e.g., edge detection, color histograms).
- Automated Feature Extraction: Using deep learning models to automatically extract features from the data.

1.4 Machine Learning Layer:

- Model Training: Training ML models such as Convolutional Neural Networks (CNNs) on the preprocessed data.
- Model Validation and Testing: Evaluating the model's performance using validation and test datasets.
- Real-Time Classification: Applying the trained model to new data for real-time classification.

1.5 Data Management Layer:

- Storage: Storing the original CSV files, preprocessed data, and model outputs.
- Data Retrieval: Efficiently retrieving data for processing and analysis.

2. Data Flow

2.1. Data Collection

- Source: Images and videos of animals are collected from various sources such as cameras, drones, and databases.
- Data Storage: Collected data is stored in CSV files, containing features like pixel values and labels.

2.2. Data Loading

- Data Import: CSV files are read and imported into the system using data loading libraries.

2.3. Initial Data Processing

- Data Cleaning: Missing values, outliers, and duplicates are handled.
- Data Transformation: Data is converted into a suitable format (e.g., normalizing pixel values, encoding labels).
- Data Splitting: The data-set is divided into training, validation, and test sets.

2.4. Feature Extraction

- Feature Engineering: Additional features are created if necessary (e.g., edge detection, color histograms).
- Automated Extraction: Deep learning models automatically extract relevant features from the data.

2.5. Model Training

- Algorithm Selection: ML models such as Convolutional Neural Networks (CNNs) are selected.
- Training: Models are trained on the preprocessed data using ML frameworks.

2.6. Model Validation and Testing

- Validation: The trained model's performance is evaluated using the validation dataset.
- Testing: The final model is tested using the test dataset to ensure accuracy and robustness.

2.7. Real-Time Classification

- Inference: The trained model classifies new images or videos in real-time or near-real-time.
- Output Storage: Classification results are stored in the database for further analysis.

2.8. Data Presentation

- User Interface: A user-friendly interface displays classification results and visualizations, allowing users to interact with the system.
- Customizable Alerts: Users can set up notifications for specific classification outcomes or patterns.

2.10. Security and Privacy

- Data Protection: Implement measures to ensure data security and user privacy.
- Access Control: Manage user access and permissions to secure the system.

This structured data flow ensures the system efficiently processes and classifies animal data, providing accurate and actionable results for various stakeholders.

3. System Components and Modules

3.1. Data Acquisition Modules

- **CSV File Reader**: Reads and imports animal data from CSV files.

- **Image/Video Importer**: Imports images or videos of animals from various sources.

3.2. Data Preprocessing Modules

- **Data Cleaning**: Handles missing values, outliers, and duplicates.
- **Data Transformation**: Normalizes pixel values, encodes labels, and converts data into a suitable format.
- **Data Splitting**: Divides the dataset into training, validation, and test sets.

3.3. Feature Extraction Modules

- **Feature Engineering Tools**: Manually creates additional features like edge detection and color histograms.

- **Automated Feature Extraction**: Uses deep learning models to automatically extract features from images and videos.

3.4. Machine Learning Modules

- **Model Training**: Trains ML models (e.g., Convolutional Neural Networks) on the preprocessed data.
- **Model Validation and Testing**: Evaluates model performance using validation and test datasets.
- **Inference Engine**: Applies the trained model to classify new images or videos in real-time.

3.5. Data Management Modules

- **Storage Solutions**: Databases and file systems for storing raw data, preprocessed data, and classification results.
- **Data Retrieval Tools**: Efficiently retrieves data for processing and analysis.

3.6. Application Modules

- **User Interface**: Web and mobile applications for users to interact with the system, upload data, and view results.
- **Customizable Alerts**: Allows users to set up notifications for specific classification outcomes.

3.7. Integration Modules

- **APIs**: Interfaces for integrating the classification system with other applications and systems.

3.8. Security and Privacy Modules

- **Data Protection**: Implements measures to ensure data security and user privacy.
- **Access Control**: Manages user access and permissions

4. Use case Diagram

The system provides animal classification information to users via other interface that displays classification results and related data. The simplicity of the diagram suggests that it illustrates the basic concept of a system providing animal classification information to users, without delving into specific implementation details or additional features.

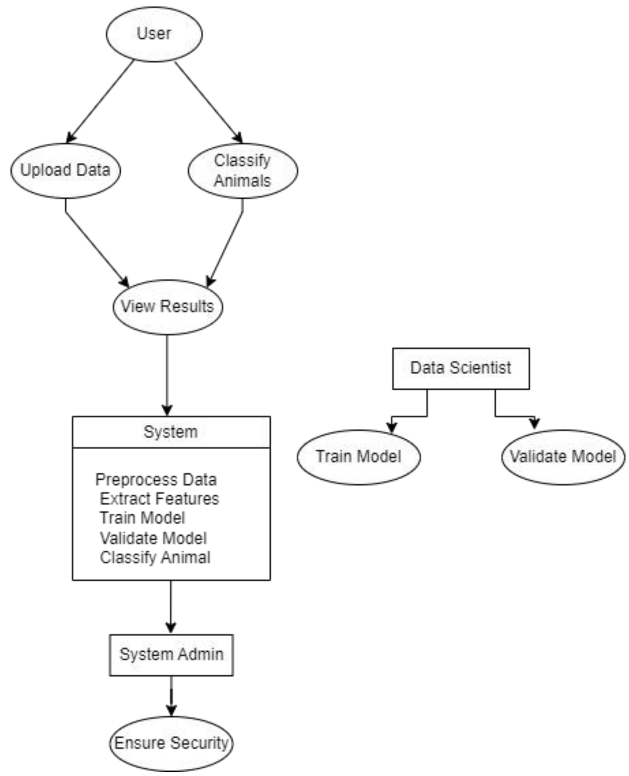


Fig:a. Use Case Diagram for Classification Of Animals

Fig: a, represents a system that provides animal classification information to users. The system displays various details such as the classification result, confidence score, and additional metadata (e.g., species, habitat). The central element in the diagram is the **System** block, which contains the different classification outputs and metadata. These outputs are presented in an organized manner, likely to be displayed or conveyed to the user.

- The **Actor/User** element represents the individual or entity interacting with the system to classify animals and view the results. The arrows pointing from the **Actor/User** towards the outputs within the **System** block indicate the flow of information. The actor or user can upload images or videos for classification, receive and view classification results, and set up notifications based on the outcomes provided by the system.

CHAPTER 5

METHADOLOGY

5.1 Explaination of Methods

1. **Data Collection and Preparation:**

- **Dataset:** The dataset likely consists of features representing various attributes of animals, such as size, weight, color, habitat, etc., along with their respective labels (e.g., species or category).
- **Data Cleaning:** Handling missing values, removing duplicates, and correcting inconsistencies to ensure high-quality data for training.
- **Feature Engineering:** Transforming raw data into meaningful features that can improve the performance of machine learning models. This might include normalization, scaling, or encoding categorical variables.

2. **Exploratory Data Analysis (EDA):**

- **Descriptive Statistics:** Analyzing basic statistics of the dataset to understand the distribution, central tendency, and spread of the data.
- **Data Visualization:** Using plots such as histograms, scatter plots, and box plots to visually inspect the relationships and distributions of features.

3. **Model Selection and Training:**

- **Algorithm Choice:** Selecting appropriate machine learning algorithms for classification, such as Decision Trees, Random Forests, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), or Neural Networks.
- **Training the Model:** Splitting the dataset into training and testing sets, and using the training set to fit the model.

- **Hyperparameter Tuning:** Adjusting the parameters of the model to improve performance, often using techniques such as grid search or random search.

4. **Model Evaluation:**

- **Performance Metrics:** Evaluating the model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess how well the model is performing.
- **Cross-Validation:** Using techniques like k-fold cross-validation to ensure the model's performance is consistent across different subsets of the data.

5. **Model Deployment:**

- **Integration:** Once the model is trained and evaluated, it can be integrated into a larger system for use in real-time classification tasks.
- **User Interface:** Developing a simple interface for users to input new data and receive classification results, possibly using web technologies or application frameworks.

6. **Continuous Improvement:**

- **Retraining:** Periodically updating the model with new data to improve its accuracy and adaptability to changes.
- **Monitoring:** Keeping track of the model's performance over time to identify any degradation in accuracy or other issues.

CHAPTER 6

IMPLEMENTATION

6.1 Used tool Explanation

The Animal Classification System using Machine Learning employs a diverse array of tools to ensure its accuracy and efficiency.

- **Data Collection:** Various tools are used to gather image and audio data of animals from sources such as wildlife cameras, field recordings, and online databases.
- **Data Preprocessing:** Tools like OpenCV and TensorFlow are employed to preprocess images and audio files. This includes tasks such as resizing images, normalizing pixel values, and extracting features from audio recordings.
- **Model Training:** Machine learning frameworks such as TensorFlow, Keras, and PyTorch are used to develop and train classification models. Convolutional Neural Networks (CNNs) are commonly utilized for image classification, while Recurrent Neural Networks (RNNs) or transformers may be used for audio data.
- **Feature Extraction:** Techniques such as Histogram of Oriented Gradients (HOG) for images and Mel-Frequency Cepstral Coefficients (MFCC) for audio are used to extract meaningful features from the raw data.
- **Model Evaluation:** The performance of classification models is evaluated using metrics like accuracy, precision, recall, and F1-score. Tools like Scikit-learn and Matplotlib help in generating performance reports and visualizations.
- **Deployment:** Once trained, models are deployed using platforms like TensorFlow Serving or Flask APIs to make real-time predictions. Docker containers or cloud services such as AWS SageMaker and Google AI Platform facilitate the deployment process.
- **User Interface:** The user interface for the system is developed using modern web frameworks such as React or Angular, and mobile frameworks like Flutter for cross-platform applications.

Interactive visualizations and results are displayed using libraries such as D3.js.

- **Notifications and Alerts:** Notifications and alerts about classification results or errors are managed using services like Twilio for SMS and Firebase Cloud Messaging for push notifications.
- **Security:** Security measures include encryption protocols such as AES and TLS for data protection, and authentication tools like OAuth to manage user access.
- **Integration and Testing:** Integration with external systems is managed via APIs. Testing is conducted using tools like JUnit for unit tests, Selenium for integration tests, and UserTesting for user acceptance testing.
- **Performance Monitoring:** Tools like New Relic and Apache JMeter are used to monitor system performance, ensuring the classification system handles increasing data volumes and user demands efficiently.

6.2 Pseudo code

```
#import all requirements

    import pandas as pd
    from sklearn.model_selection import train_test_split
    import matplotlib.pyplot as plt
    import seaborn as sns
    import tensorflow.compat.v1 as tf
    tf.disable_v2_behavior()

    import warnings
    warnings.filterwarnings("ignore")

#import dataset

    animals = pd.read_csv("zoo.csv")
    animals.head()
```

#representation and visualization of animal types using bar graph

```
animals.shape
#This dataset has 101 rows and 18 columns

#  The animal types
plt.figure(figsize=(10, 5))
sns.set_style('darkgrid')
sns.countplot(animals['class_type'])

# Correlation plot of 16 animal features
animal_name and class_type columns are not included
corr = animals.iloc[:, 1:-1].corr()

colormap = sns.diverging_palette(650, 690, as_cmap = True)
plt.figure(figsize=(14, 14))
sns.heatmap(corr, cbar = True, square = True, annot = True,
        fmt = '.2f', annot_kws = {'size': 12},
        cmap = colormap, linewidths = 0.1, linecolor = 'white')
plt.title('Correlation of Animal Features', y = 1.05, size = 15)
```

Prepare for Machine Learning

Features & Label

“””From the dataset consisting of 18 columns taken 17 columns as a feature (x_data) and 1 column as a label (y_data) used for comparison with prediction results”””

#Train and test data

```
x_data = animals.iloc[:, :-1]
y_data = animals.iloc[:, -1:]

train_x, test_x, train_y, test_y = train_test_split(x_data, y_data,
                                    test_size=0.3,
                                    random_state=42,
                                    stratify=y_data)

print('Training data has :', train_x.shape)
print('Testing data has :', test_x.shape)
```

```
train_name = train_x['animal_name']
test_name = test_x['animal_name']

train_x = train_x.iloc[:, 1:]
test_x = test_x.iloc[:, 1:]

print("Training Data has",train_x.shape)
print("Testing Data has",test_x.shape)


X = tf.placeholder(tf.float32, [None, 16])
Y = tf.placeholder(tf.int32, [None, 1])

Y_one_hot = tf.one_hot(Y, 7)
Y_one_hot = tf.reshape(Y_one_hot, [-1, 7])
```

Make weight and bias value

```
W = tf.Variable(tf.random_normal([16, 7], seed = 0), name = 'weight')
b = tf.Variable(tf.random_normal([7], seed = 0), name = 'bias')
```

Output

```
**Output = Weight * Input + Bias**
logits = tf.matmul(X, W) + b
hypotesis = tf.nn.softmax(logits)
cost_i = tf.nn.softmax_cross_entropy_with_logits_v2(logits=logits, labels=Y_one_hot)
cost = tf.reduce_mean(cost_i)
train = tf.train.GradientDescentOptimizer(learning_rate=0.05).minimize(cost)
```

Original vs prediction

```
prediction = tf.arg_max(hypotesis, 1)
correct_prediction = tf.equal(prediction, tf.arg_max(Y_one_hot, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
```

Activate Model

```
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for step in range(5001):
        sess.run(train, feed_dict = {X: train_x, Y: train_y})
        if step % 1000 == 0:
            loss, acc = sess.run([cost, accuracy], feed_dict={X: train_x, Y: train_y})
```

```
        print('Step: {:5}\tloss: {:.3f}\tAcc: {:.2%}'.format(step, loss, acc))

    train_acc = sess.run(accuracy, feed_dict={X: train_x, Y: train_y})
    test_acc, test_predict, test_correct = sess.run([accuracy, prediction, correct_prediction],
feed_dict={X: test_x, Y: test_y})
    print('Model Prediction =', train_acc)
    print('Test Prediction', test_acc)

# Result

sub = pd.DataFrame()
sub['Name'] = test_name
sub['Predict_Type'] = test_predict
sub['Origin_Type'] = test_y
sub['Correct'] = test_correct
sub

sns.countplot(sub['Correct'])

sub['Correct'].value_counts()

#True     29
#False     2
#Name: Correct, dtype: int64
```

CHAPTER 7

TESTING

To apply the concept of testing to an animal classification system using machine learning, we can map the different types of testing to the various stages and components of the system. Here's how each type of testing can be used in the context of a machine learning project for animal classification:

1. **Unit Testing**

- **Purpose:** Verify that individual components of the animal classification system work correctly in isolation.
- **Types:**
  - ➢ **Data Preprocessing Functions:** Test individual functions for data cleaning, feature extraction, and transformation to ensure they perform as expected.
  - ➢ **Model Training:** Check that the machine learning algorithms correctly fit the training data without errors.
  - ➢ **Prediction Functions:** Verify that the model can make predictions on a given input and produce outputs in the expected format.

2. Integration Testing

- **Purpose::** Ensure that different components of the system work together as expected.
- **Types:**
  - ➢ **Data Pipeline Integration:** Test the integration between data loading, preprocessing, and feature engineering to ensure smooth data flow.

- **Model and Database Integration:** Verify that trained models can interact with databases for storing and retrieving model parameters and results.
- **UI and Backend Integration:** Ensure that the user interface can accurately display classification results and interact with the backend systems.

3. System Testing

- **Purpose:** Test the complete and integrated animal classification system as a whole.
- **Types:**
  - **Functional Testing:** Validate that the system performs all required functions, such as data ingestion, model training, prediction, and reporting.
  - **Performance Testing:** Assess the system's performance under normal and peak loads, including training times and prediction speeds.
  - **Security Testing:** Ensure that the system is secure from unauthorized access and data breaches.

4. User Acceptance Testing (UAT)

- **Purpose:** Ensure the system meets endusers' needs and expectations.
- **Types:**
  - **Usability Testing:** Evaluate the user interface and overall user experience to ensure it is intuitive and user-friendly.
  - **Scenario Testing:** Test the system's ability to handle realworld scenarios, such as classifying animals from various datasets.

5. Regression Testing

- **Purpose:** Verify that recent changes or updates to the system have not adversely affected existing functionalities.
- **Types**:
    - **Model Update Validation:** Ensure that updates to the model or pipeline do not introduce new issues or degrade existing performance.

6. Load Testing

- **Purpose:** Assess the system's performance under various levels of data load and user activity.
- **Types:**
    - **Scalability Testing:** Evaluate how well the system can handle increasing amounts of data and prediction requests over time.

7. **Stress Testing**

- **Purpose:** Determine the system's behavior under extreme conditions or loads.
- **Types:**
    - **Failure Recovery Testing:** Test how the system recovers from failures or crashes when handling large datasets or complex models.

8. **Acceptance Testing**

- **Purpose:** Validate that the system meets all specified requirements and is ready for deployment.

Types:

- **Specification Compliance:** Ensure that all system specifications and requirements, such as accuracy thresholds and response times, are met.

9. Environmental Testing

- **Purpose:** Test the system's performance and reliability in different environmental conditions.
- **Types:**
  - **Dataset Variation Testing:** Conduct tests with datasets from various sources and conditions to ensure robustness and accuracy across different data environments.

10. End-to-End Testing

- **Purpose:** Test the complete workflow from data collection through classification to user notification.
- **Types:**
  - **Workflow Testing:** Validate that the entire process, from ingesting animal data to classifying and delivering results, functions seamlessly.

These testing strategies ensure that an animal classification system using machine learning is robust, reliable, and meets user expectations while performing accurately under various conditions.

7.1 Test Cases

Test Case ID	Test case Description	Test case Objective	Test case Preconditions	Expected results	Actual Output
01	Test data preprocessing function for handling missing values	Ensure the function correctly processes missing data by filling or removing them	Dataset with known missing values is available	Preprocessing function should fill/remove missing values without errors.	PASS
02	Verify model training with a small dataset	Validate the model's ability to train and achieve a baseline accuracy	Access to a small, labeled dataset	Model should train successfully and achieve a baseline accuracy.	PASS
03	Test prediction function with known inputs	Verify the model can correctly classify known inputs	Trained model and test inputs with expected outputs are available	Prediction function should return correct class for given test samples	PASS
04	Test data pipeline integrati on from loading to feature extractio n	Ensure seamless data flow through the pipeline	Data files are ready for input and the feature extraction pipeline is set up	Data should flow through the pipeline without errors, producing correct feature sets	PASS
05	Perform functional testing of the complete system	Validate all system functions, including data ingestion, training, and prediction	Full system setup with access to required datasets	System should perform all required functions seamlessly	PASS

06	Scenario testing for diverse animal classification	Validate system performance with diverse datasets	Access to datasets with varied animal classes	System should accurately classify a wide range of animals across different datasets	PASS

CHAPTER 8

SNAPSHOTS

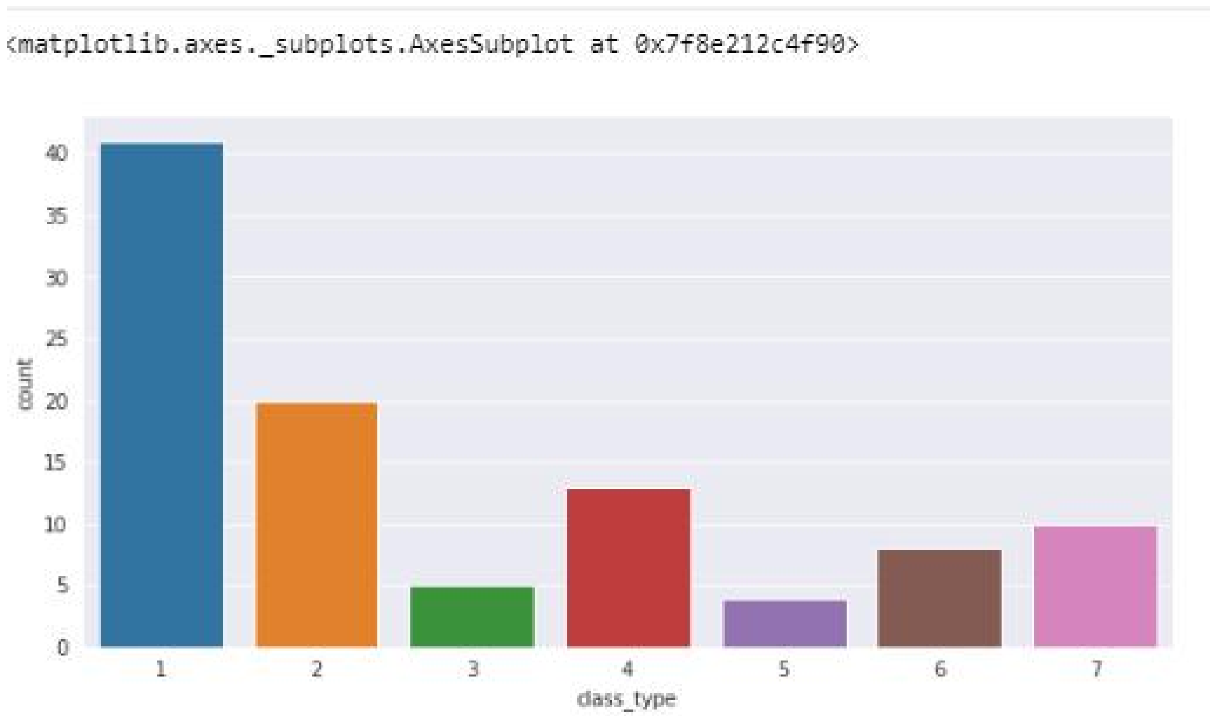


Fig:Types of animals count vs class_type graph.

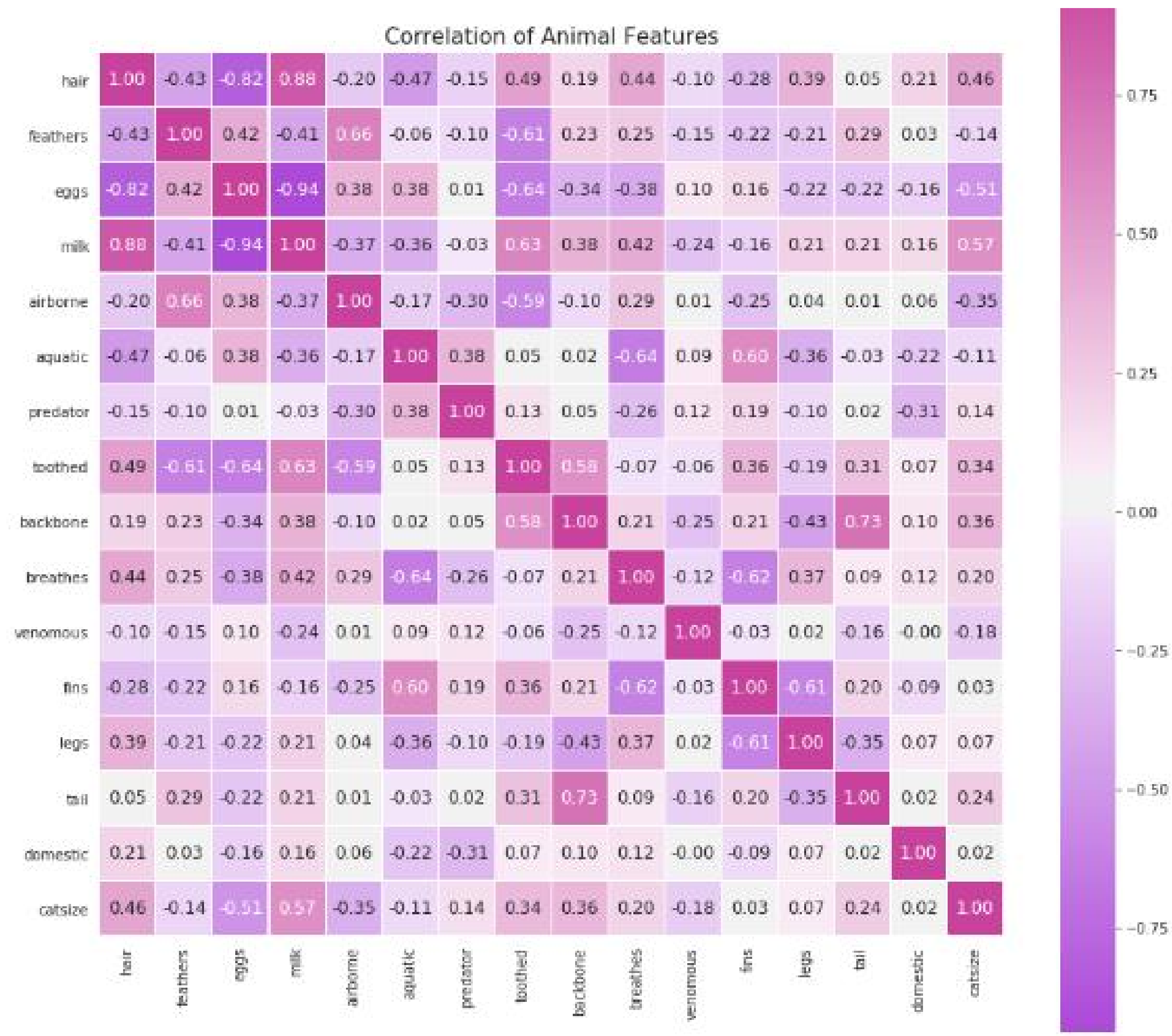
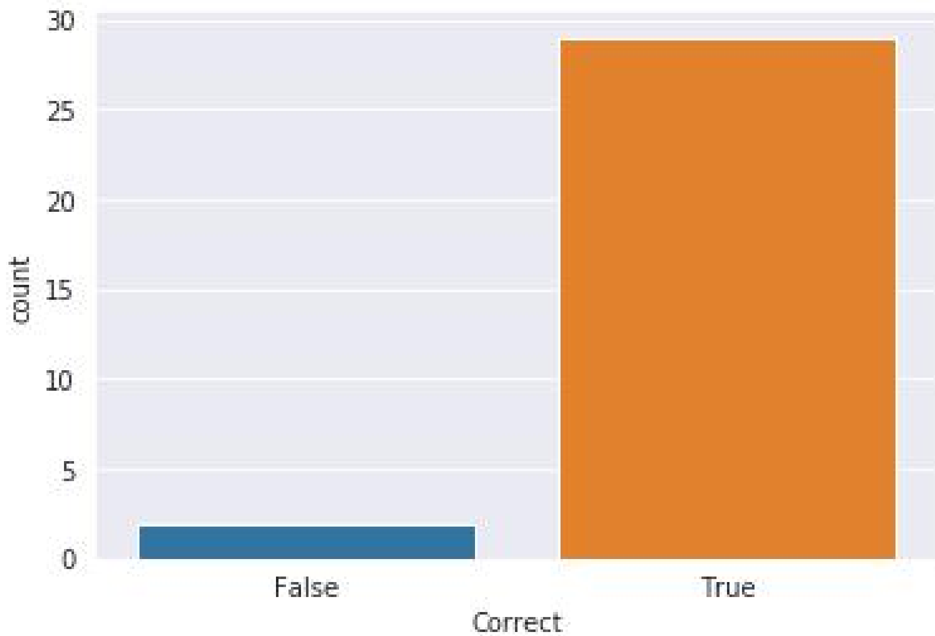


Fig: Correlation of Animal Features.

	Name	Predict_Type	Origin_Type	Correct
100	wren	2	2	True
58	penguin	2	2	True
43	lark	2	2	True
21	duck	2	2	True
10	cheetah	1	1	True
40	housefly	6	6	True
50	mongoose	1	1	True
4	boar	1	1	True
87	swan	2	2	True
80	slowworm	3	3	True
70	reindeer	1	1	True
37	hawk	2	2	True
7	carp	4	4	True
44	leopard	1	1	True
53	octopus	0	7	True
60	pike	4	4	True
84	squirrel	1	1	True
47	lynx	1	1	True
94	vole	1	1	True
89	toad	5	5	True
92	tuna	4	4	True
93	vampire	1	1	True
49	mole	1	1	True
12	chub	4	4	True
31	goat	1	1	True
76	seasnake	3	3	True
77	seawasp	4	7	False
65	pony	1	1	True
14	crab	5	7	False
51	moth	6	6	True
22	elephant	1	1	True

Fig: Test case for checking Model Prediction

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e1eecf550>
```




```
sub['Correct'].value_counts()
```

```
True     29
False     2
Name: Correct, dtype: int64
```

Fig: Prediction Accuracy bar graph

FUTURE ENHANCEMENT

Future enhancements in animal classification using machine learning aim to significantly boost accuracy, scalability, and versatility. Advances in data collection will involve deploying high-resolution imaging sensors and cameras in natural habitats and zoos, along with drones and automated systems to gather diverse data. IoT technology will enable real-time monitoring of animal behavior, providing a continuous data stream for model training.Modeling advancements will feature sophisticated neural network architectures, such as transformers, which better capture complex animal features. Techniques like transfer learning and few-shot learning will allow models to generalize from limited data, enabling classification for rare species. Data augmentation and synthetic data generation will expand training datasets, ensuring models can handle various environmental conditions and animal poses.

Cloud-based computing will facilitate large-scale model training, allowing real-time classification across networks. Edge computing will enable faster local data processing, reducing latency and improving field application responsiveness.User-centric communication will evolve with intuitive interfaces that deliver actionable insights to researchers and conservationists. Interactive dashboards and mobile apps will simplify access to classification results and insights into animal behavior.Collaboration between research institutions, conservation organizations, and tech companies will enhance data sharing and drive advancements in classification techniques. Integration with smart ecosystems will optimize wildlife monitoring, helping to mitigate human-animal conflicts and promote biodiversity.

These advancements will push the boundaries of animal classification, providing precise insights that support conservation, research, and education efforts worldwide. They will enhance ecosystem management and contribute to a broader understanding of biodiversity.

.

CONCLUSION

The evolution of animal classification using machine learning is set to transform our ability to identify and understand wildlife with unprecedented accuracy and efficiency. Through advancements in data collection, model development, and computational power, coupled with the integration of machine learning and AI, classification systems will become more precise and reliable. Enhanced communication tools will ensure that researchers and conservationists receive timely and relevant information, while collaborative efforts across academia, industry, and conservation organizations will drive continuous improvements. These developments promise not only to enhance wildlife monitoring and conservation efforts but also to optimize various sectors such as ecology, zoology, and environmental management. Ultimately, the future of animal classification will enable more informed decision-making, better biodiversity management, and a deeper understanding of ecosystems by providing actionable insights into species identification and behavior.

REFERENCES

[1] https://in.element14.com /industrial-automation-control / industrial-iot-automation.

[2] https://www.fogwing.io /industrial-iot-platform /industrial-iot-reference-book.

[3] https://www.contec.com / weather forecast and reporting system

[4] https://www.holidaygiftssearch.com / industrial_iot.

[5] https://www.rfpage.com/applications-of-industrial-internet-of-things.

[6] https://www.sciencedirect.com/topics/computer-science/industrial-internet-of-thing