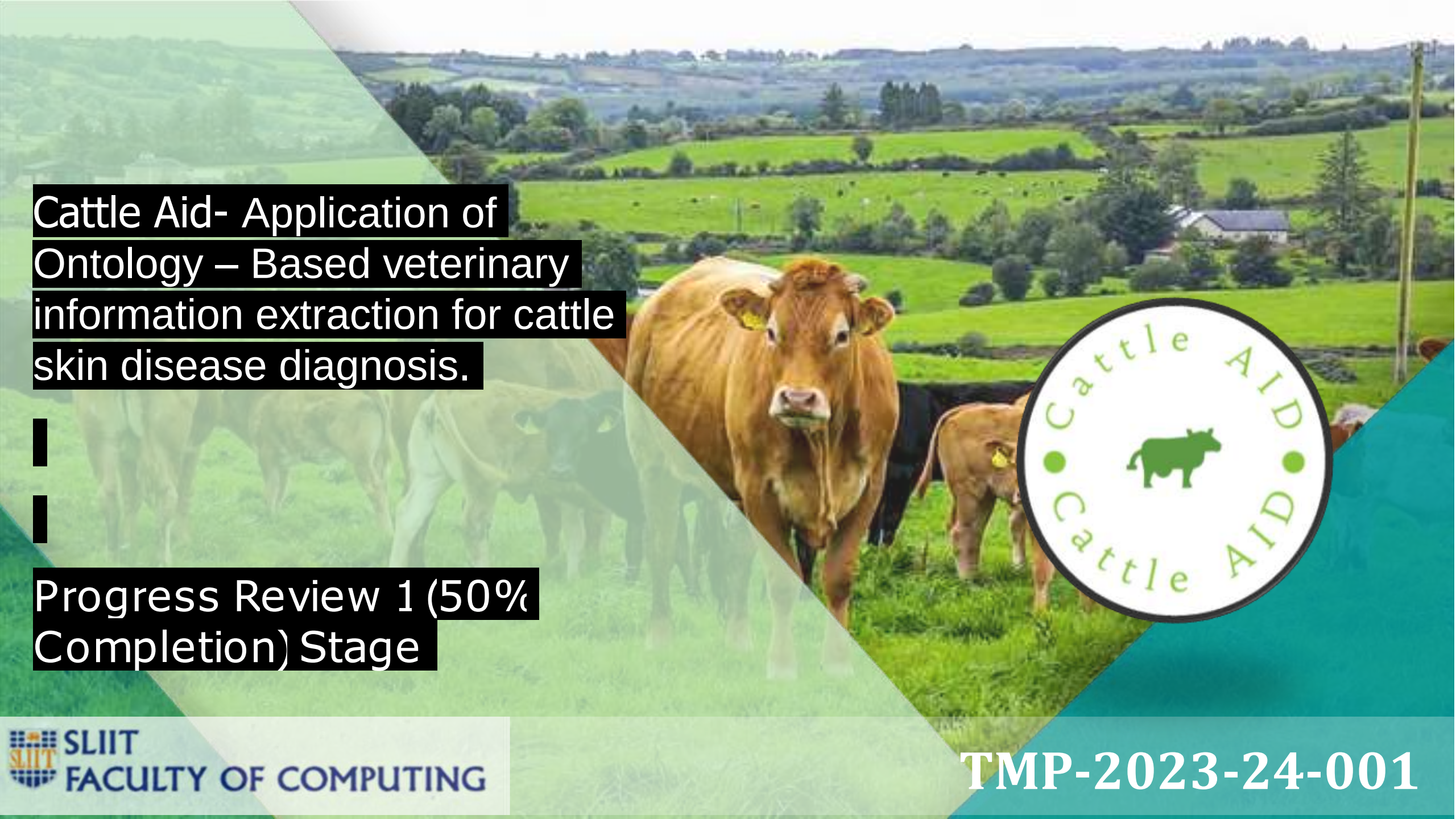
Cattle Aid- Application of Ontology – Based veterinary information extraction for cattle skin disease diagnosis.



Progress Review 1 (50% Completion) Stage

SLIIT
FACULTY OF COMPUTING

TMP-2023-24-001

M e e t O u r T e a m !

Co-Supervisor :
Dr. Anuradha Jayakody

Supervisor :
Mrs. Suranjini Silva

Nawarathna N.N
IT20654658

Jayawardhana K.A.D.D.S
IT20651824

M.G.K.Chethani
IT20659226

Rubasinghe L.M
IT20299552

SLIIT
FACULTY OF COMPUTING

Data Gathering

Name: Thamali Wickramsigne

Email: tamalikwickramsinghe@gmail.com

Contact No: 0774289479

Profession: Veterinary Surgeon at Pet care solution hospital

It is with great pleasure and anticipation that I am writing to extend my warmest welcome to your esteemed research group, which is planning to visit our veterinary clinic and research institution for the purpose of gathering data on cattle skin diseases.

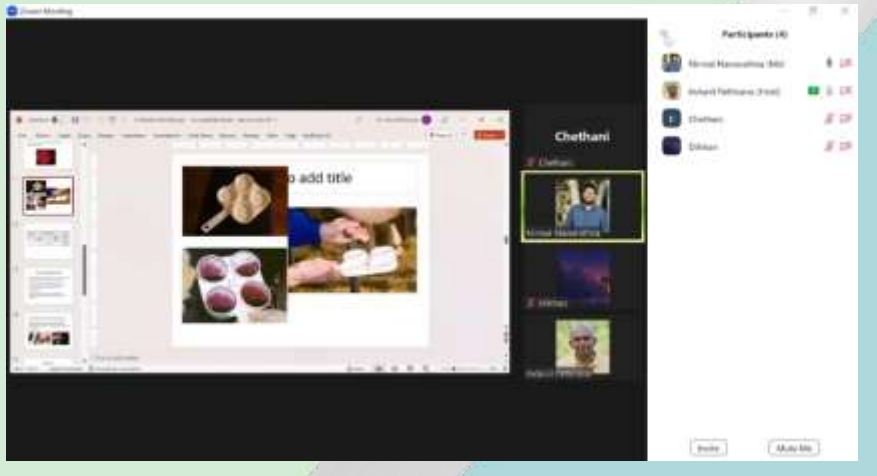
We are genuinely thrilled to learn about your upcoming research project and your keen interest in cattle skin diseases. As veterinarians, we understand the vital role that research plays in advancing the well-being of livestock and the agricultural industry. Your visit to our facility will provide a unique opportunity for your students to observe, gather data, and verify information related to cattle skin diseases.

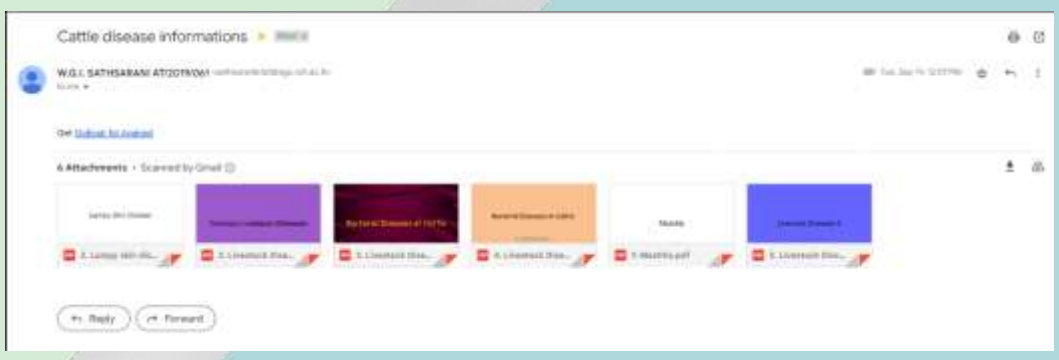
We are committed to supporting your research efforts in any way possible. Our team of experienced veterinarians, research staff, and state-of-the-art facilities will be at your disposal during your visit. You will have the chance to interact with our experts, access our database of clinical cases, and have hands-on experience with the diagnosis and treatment of cattle skin diseases.

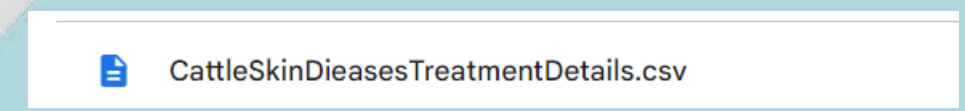
Furthermore, we will provide you with all necessary resources, including access to our extensive library, laboratory equipment, and our network of local farmers and ranchers who can offer valuable insights into cattle skin health. Please let us know your preferred dates for the visit so that we can make the necessary arrangements. If there are specific areas of interest or research methodologies you would like to focus on during your stay, kindly inform us in advance so that we can tailor the visit to your needs.

We are confident that your research group's visit will be mutually beneficial and contribute significantly to the understanding of cattle skin diseases. We look forward to the knowledge exchange and collaboration that this visit promises to foster. Once again, I extend my warmest welcome to your research group, and we eagerly await your visit. Safe travels and we look forward to a productive and enlightening partnership.

Sincerely,



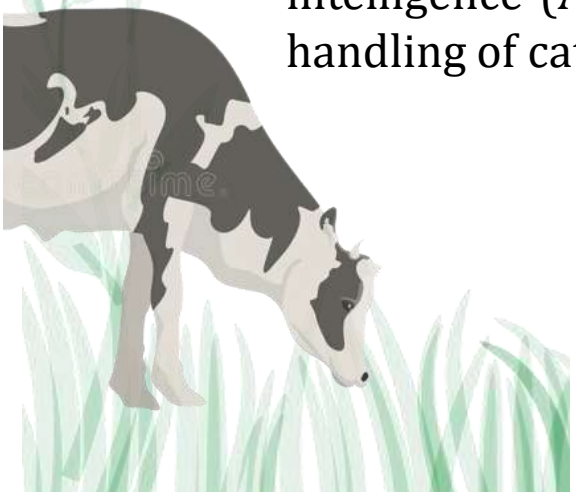




SLIIT FACULTY OF COMPUTING

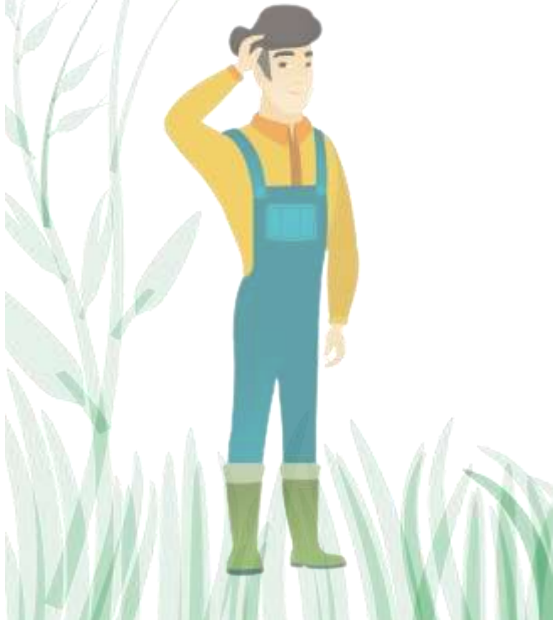
INTRODUCTION

In the dynamic landscape of veterinary medicine, technological innovations have emerged as crucial tools for addressing the challenges faced by livestock industries. In this context, our research endeavours to revolutionize the diagnosis and management of cattle skin diseases through the integration of advanced technologies and domain-specific knowledge. Our research topic, "Application of Ontology-Based Veterinary Information Extraction for Cattle Skin Disease Diagnosis," encapsulates a multidimensional approach that harnesses the power of artificial intelligence (AI) and ontology-based information extraction to enhance the understanding and handling of cattle skin diseases.



RESEARCH QUESTION

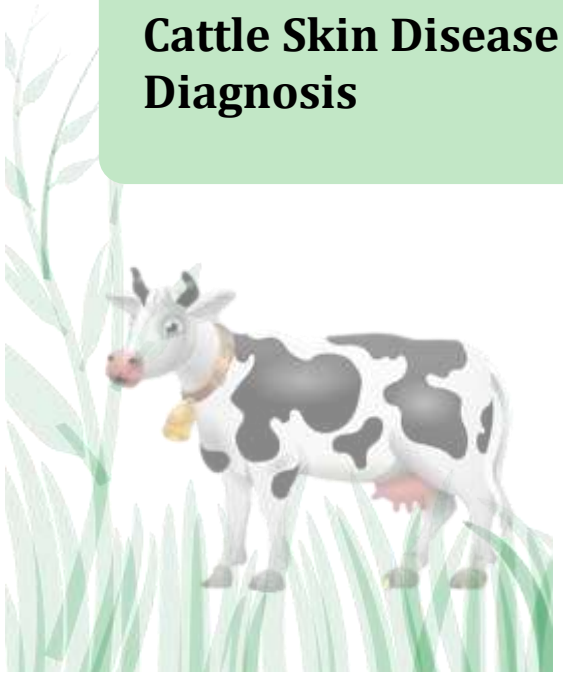
The cattle industry in Sri Lanka, a vital component of the agricultural sector, is pivotal to the nation's economy. However, the occurrence of skin diseases among cattle remains a pressing concern, impacting animal welfare, productivity, and economic stability. The conventional methods of diagnosing these diseases often rely on subjective visual assessments, leading to inconsistencies and delays in treatment. Moreover, the accurate assessment of disease severity and the efficient extraction of relevant information from diverse data sources pose formidable challenges.



OBJECTIVES

Main Objective

Ontology-Based Veterinary Information Extraction for Cattle Skin Disease Diagnosis



Sub Objectives

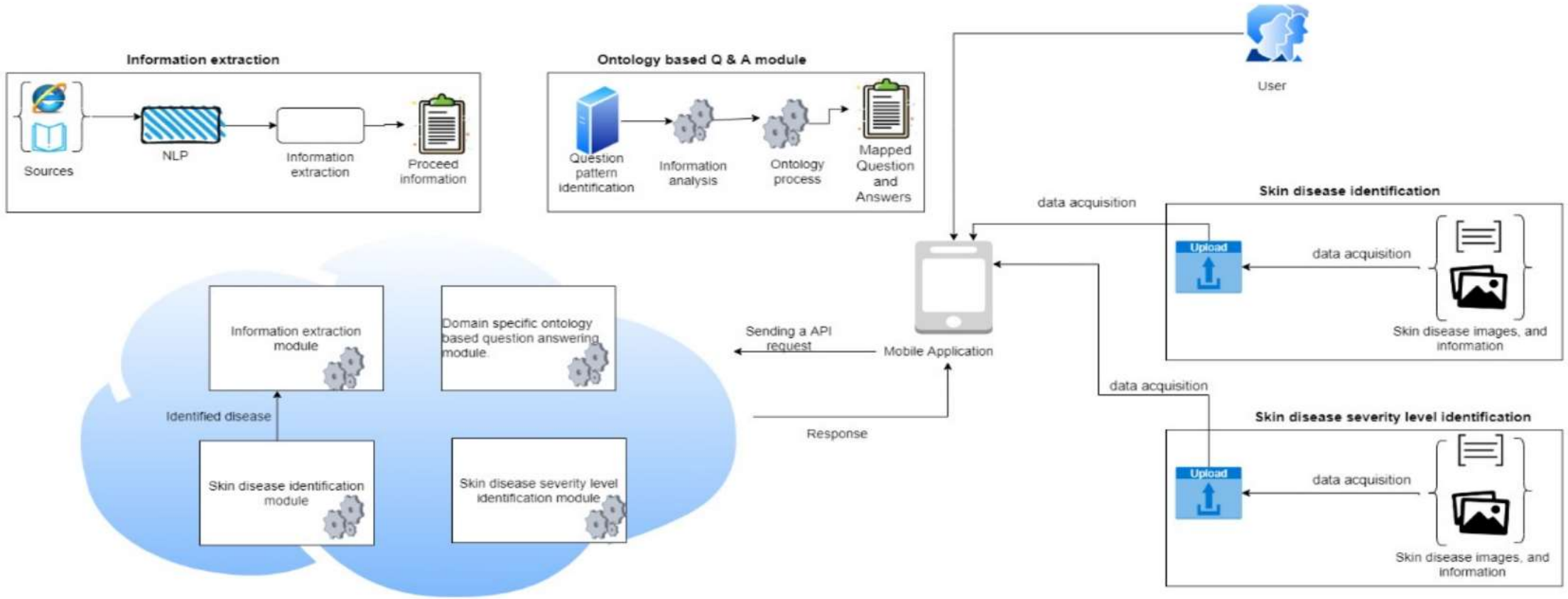
Enhanced Cattle Skin Disease Detection and Severity Assessment Using Image Processing and Advanced Models

Ontology-Based Information Extraction from Data Resources

User specific knowledge based up to date using reinforcement learning.

AI-Driven Smart Assistant for Cattle Skin Diseases

SYSTEM DIAGRAM



IT20654658|Nawarathna N.N

Bachelor of Science (Hons) in Information Technology

Specializing in Software Engineering

Enhanced Cattle Skin Disease Detection and Severity Assessment Using Image Processing and Advanced Models

BACKGROUND

Objective: To improve the identification and severity assessment of cattle skin diseases using image processing and advanced models.

Key Tasks:

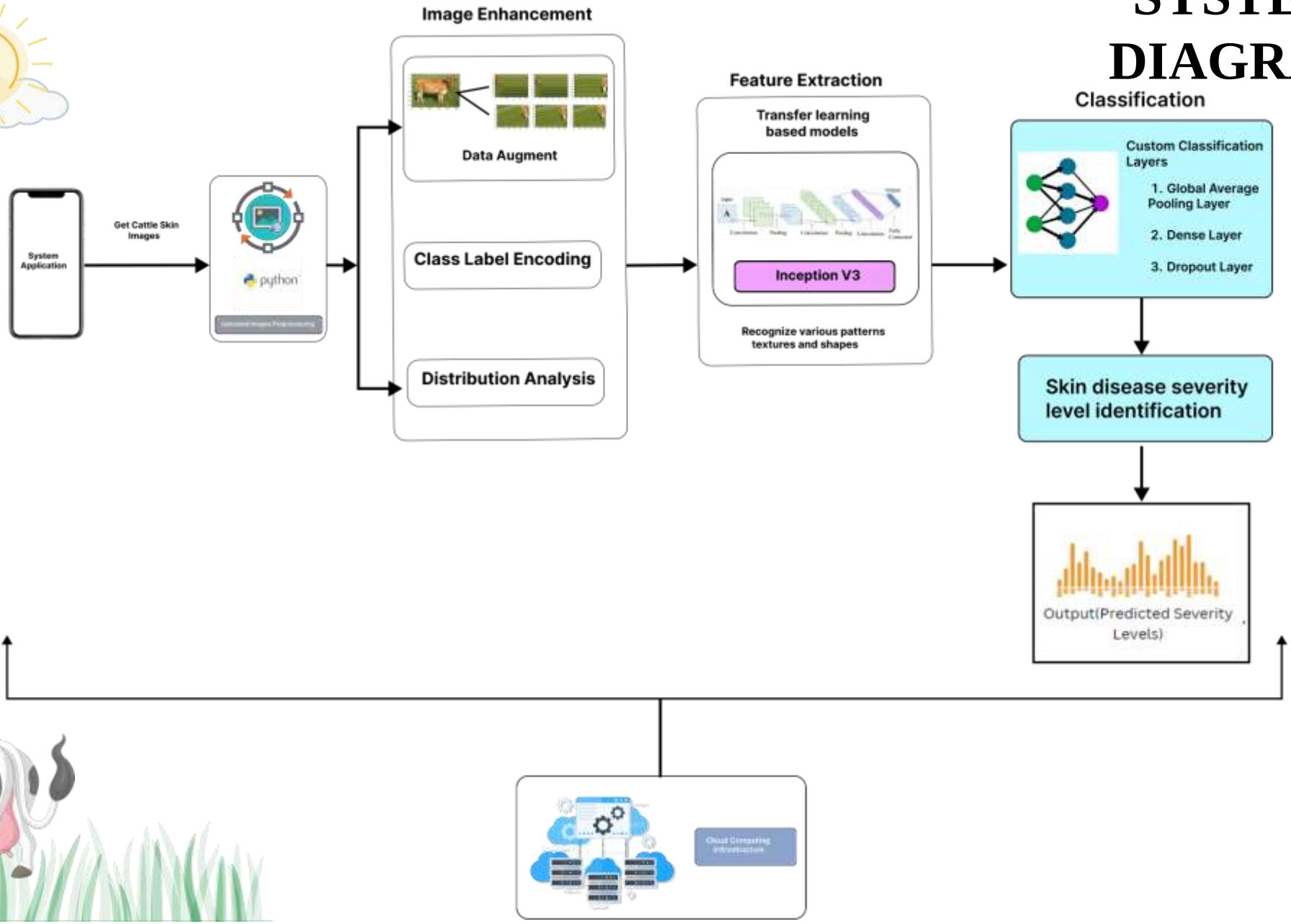
1. ***Identify Cattle Skin Diseases:*** Implement advanced image processing techniques to accurately identify common cattle skin diseases.
2. ***Severity Assessment:*** Develop a severity assessment model for early detection and monitoring of disease progression.

Targeted Diseases:

Focus on "**Lumpy Skin**" and "**Papillomatosis**," two of the most prevalent cattle skin diseases.

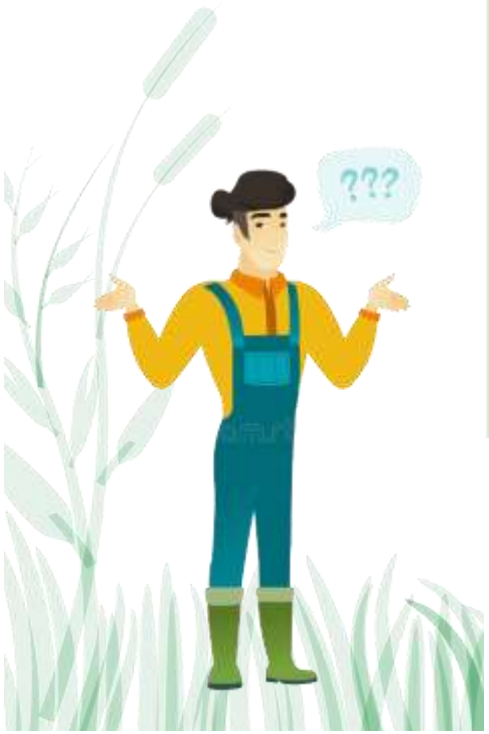
papillomatosis

SYSTEM DIAGRAM



RESEARCH QUESTION

- What image processing techniques work best for early and accurate detection of "Lumpy Skin" and "Papillomatosis" in cattle?
- How can advanced machine learning and deep learning models be used to evaluate the severity of cattle skin diseases, and what are the appropriate performance metrics to gauge their accuracy?



SPECIFIC OBJECTIVE

Cattle Skin Disease Detection
and
Severity Assessment

SUB OBJECTIVE

- Binary Classification Model
- Advanced Models Implementation
- Severity Assessment

Completed Sub Objectives

- Binary Classification Model
- Advanced Models Implementation

Data Collection & Data Preprocessing

Data Collection

Image Data was collected from Google, Kaggle and Mendeley

After collecting the images containing cattle I had to label them. I labeled them as lumpycows and helthycows.

Data Preprocessing

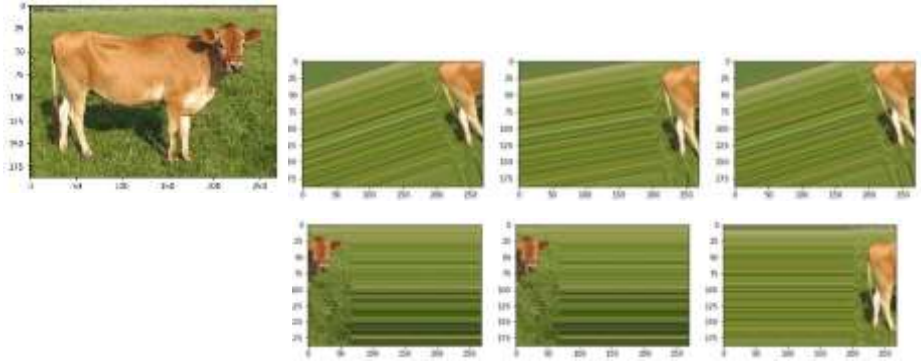
Used to increase a model's accuracy, as well as reduce its complexity

Following image preprocessing techniques were used:

1. **Data Augmentation :- Rotation, Shear, Zoom, Horizontal Flip, Fill Mode**
2. **Image Loading and preprocessing :- Resizing , Conversion to RGB , Normalization**
3. **Label Encoding**
4. **Distributed Analysis**

Data Preprocessing

Used Keras's ImageDataGenerator class to Augment our data.



Step 2: Data Augmentation

```
[2]  # Data Augmentation for cattle classification
     datagen = ImageDataGenerator(
         rotation_range=20,
         width_shift_range=0.2,
         height_shift_range=0.2,
         shear_range=0.2,
         zoom_range=0.2,
         horizontal_flip=True,
         fill_mode='nearest',
         rescale=1.0 / 255.0
     )
```

1012 Images

```
Number of lumpycows images: 497
Number of healthycows images: 515
```

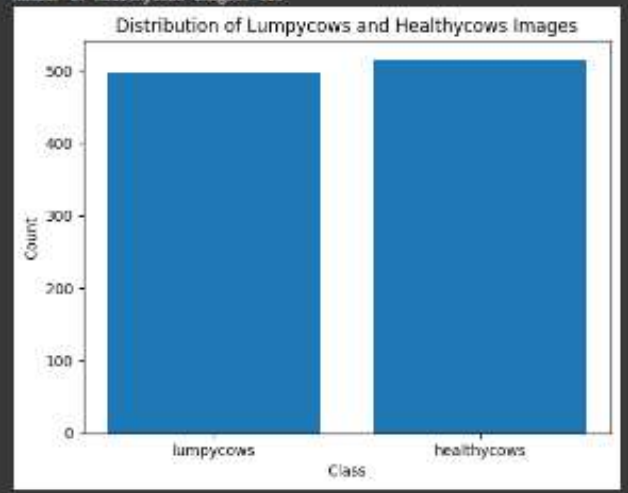
Step 4: Load and Preprocess Data

```
[4]  # Define the load_and_preprocess_data function to load and preprocess images
     def load_and_preprocess_data(data_dir, label):
         images = []
         labels = []

         for filename in os.listdir(data_dir):
             if filename.lower().endswith((".jpg", ".jpeg", ".png")):
                 image = Image.open(os.path.join(data_dir, filename))
                 image = image.resize((256, 256))
                 image = image.convert("RGB")
                 image = np.array(image)
                 if image.shape == (256, 256, 3):
                     image = image / 255.0
                     images.append(image)
                     labels.append(label)

         return np.array(images), labels

     # Load and preprocess the dataset for cattle classification
     lump_dir = '/content/drive/MyDrive/Research_Models_Datasets/cattles/cows/lumpycows'
     helth_dir = '/content/drive/MyDrive/Research_Models_Datasets/cattles/cows/healthycows'

     lump_images, lump_labels = load_and_preprocess_data(lump_dir, label='lumpycows')
     helth_images, helth_labels = load_and_preprocess_data(helth_dir, label='healthycows')
```

Step 5: Data Distribution Analysis

```
# Print the counts for the dataset
print('Number of lumpycows images:', len(lump_images))
print('Number of healthycows images:', len(helth_images))

# Combine the data and labels for the dataset
X = np.concatenate([lump_images, helth_images], axis=0)
y = lump_labels + helth_labels

# Data for the dataset
labels = ['lumpycows', 'healthycows']
counts = [len(lump_images), len(helth_images)]

# Create a bar plot for the dataset
plt.bar(labels, counts)
plt.xlabel('Class')
plt.ylabel('Count')
plt.title('Distribution of Lumpycows and Healthycows Images')

# Display the plot for the dataset
plt.show()
```

```
Number of lumpycows images: 497
Number of healthycows images: 515
```



Select the best algorithm for the model with highest accuracy

First Model

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(150,150, 3)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Dropout(0.3))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(4, activation='softmax'))
```

```
[ ] accuracy = np.sum(pred==y_test)/np.size(pred)
    print("Accuracy on testing dataset: {:.2f}%".format(accuracy*100))

    Accuracy on testing dataset: 69.15%
```

69.15%

Second Model

```
# Step 6: Build Deep Learning Model with Transfer Learning
from tensorflow.keras.applications import MobileNet
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense
from tensorflow.keras.models import Model

# Load the MobileNet model pre-trained on ImageNet data
base_model = MobileNet(input_shape=(256, 256, 3), include_top=False, weights='imagenet')

# Freeze the layers of the base model
for layer in base_model.layers:
    layer.trainable = False

# Add custom classification layers on top of the base model
x = GlobalAveragePooling2D()(base_model.output)
x = Dense(256, activation='relu')(x)
output = Dense(1, activation='sigmoid')(x)

# Create the new model
model = Model(inputs=base_model.input, outputs=output)

# Compile the model
model.compile(optimizer='adam', loss=tf.losses.BinaryCrossentropy(), metrics=['accuracy'])
```

```
WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is not in [128, 160,
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilene
17225924/17225924 [==============================] - 2s 0us/step
```

```
[ ] # Step 12: Evaluate on Testing Dataset
    y_true = [0, 1, 0, 1, 1, 0, ...]  # Ground truth labels
    y_pred = [0, 1, 1, 1, 0, 0, ...]  # Model predictions

    # Calculate accuracy
    accuracy = sum(1 for yt, yp in zip(y_true, y_pred) if yt == yp) / len(y_true)
    print("Accuracy on testing dataset: {:.2f}%".format(accuracy * 100))

    Accuracy on testing dataset: 71.43%
```

71.43%

Last Model

Step 6: Model Building and Compilation

```
[ ] # Create a more complex model (InceptionV3) for cattle classification
    base_model = InceptionV3(input_shape=(256, 256, 3), include_top=False, weights='imagenet')
    for layer in base_model.layers:
        layer.trainable = False

    x = GlobalAveragePooling2D()(base_model.output)
    x = Dense(256, activation='relu')(x)
    x = Dropout(0.5)(x)
    output = Dense(1, activation='sigmoid')(x)

    model = Model(inputs=base_model.input, outputs=output)

    # Compile the model for cattle classification
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception
87910968/87910968 [==============================] - 0s 0us/step
```

Step 10: Test Dataset Accuracy

```
[ ] # Evaluate on the test dataset
    test_accuracy = model.evaluate(test_dataset)
    test_accuracy_percentage = test_accuracy[1] * 100  # Multiply by 100 to get the percentage

    print(f'Test Accuracy: {test_accuracy_percentage:.2f}%')

    59/59 [==============================] - 8s 129ms/step - loss: 0.2973 - accuracy: 0.8694
    Test Accuracy: 86.94%
```

86.94%

Select the best algorithm for the model with highest accuracy

Step 6: Model Building and Compilation

```
[ ]  # Create a more complex model (InceptionV3) for cattle classification
     base_model = InceptionV3(input_shape=(256, 256, 3), include_top=False, weights='imagenet')
     for layer in base_model.layers:
         layer.trainable = False

     x = GlobalAveragePooling2D()(base_model.output)
     x = Dense(256, activation='relu')(x)
     x = Dropout(0.5)(x)
     output = Dense(1, activation='sigmoid')(x)

     model = Model(inputs=base_model.input, outputs=output)

     # Compile the model for cattle classification
     model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/incept
87910968/87910968 [==============================] - 0s 0us/step
```

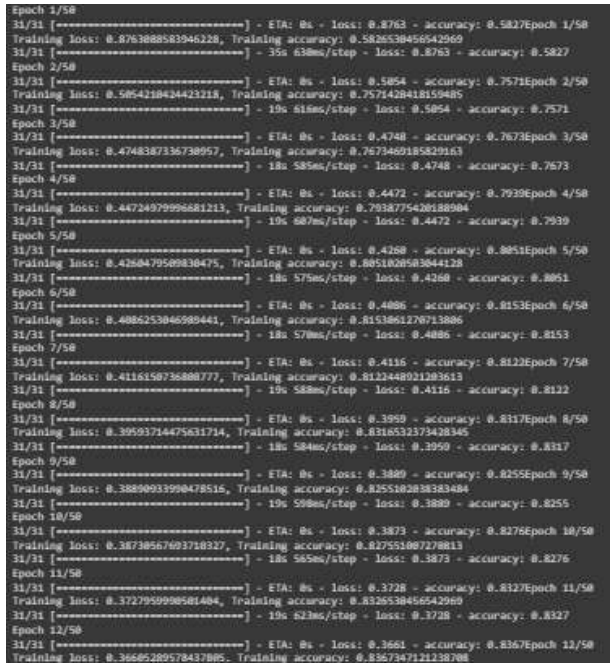
Model	Testing Accuracy	Key Features
First Model	69.15%	Basic CNN architecture , Minimal data
Second Model	71.43%	Data augmentation, Improved accuracy
Last Model	86.94%	Transfer learning (Inception V3) , Data Augmentation , significantly improved accuracy , and fast training

use of transfer learning with Inception V3, data augmentation, and dropout layers, which collectively led to a remarkable accuracy of 86.94%.

Training the Model

Step 7: Custom Callback and Training

```
[ ]  # Define custom callbacks
     class CustomCallback(tf.keras.callbacks.Callback):
         def on_epoch_end(self, epoch, logs=None):
             print(f'Epoch {epoch+1}/{epochs}')
             print(f'Training loss: {logs["loss"]}, Training accuracy: {logs["accuracy"]}')
             if 'val_loss' in logs and 'val_accuracy' in logs:
                 print(f'Validation loss: {logs["val_loss"]}, Validation accuracy: {logs["val_accuracy"]}')

     custom_callback = CustomCallback()

     # Train the model for cattle classification
     epochs = 50

     history = model.fit(
         train_data,
         steps_per_epoch=train_data.samples // train_data.batch_size,
         epochs=epochs,
         validation_data=val_data,
         validation_steps=val_data.samples // val_data.batch_size,
         callbacks=[custom_callback]
     )
```

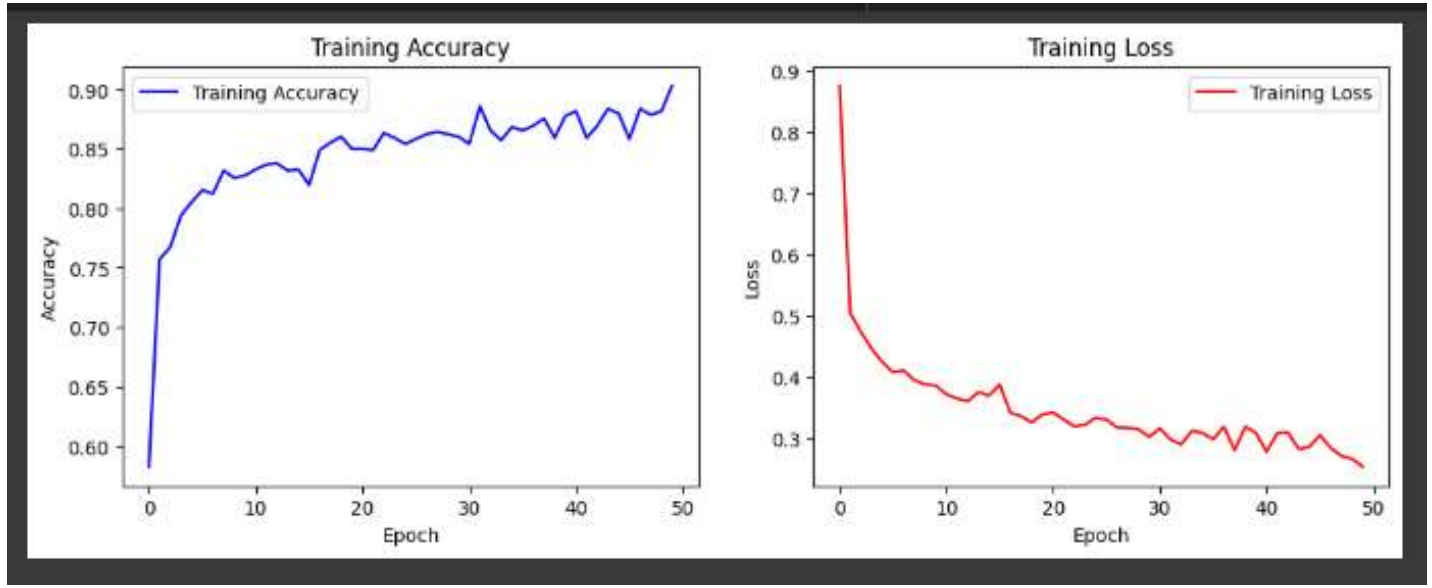


This code defines a custom callback to monitor the training of a cattle classification model. It prints the training loss and accuracy at the end of each epoch during 50 epochs of training. This helps in tracking model performance.

Training the Model

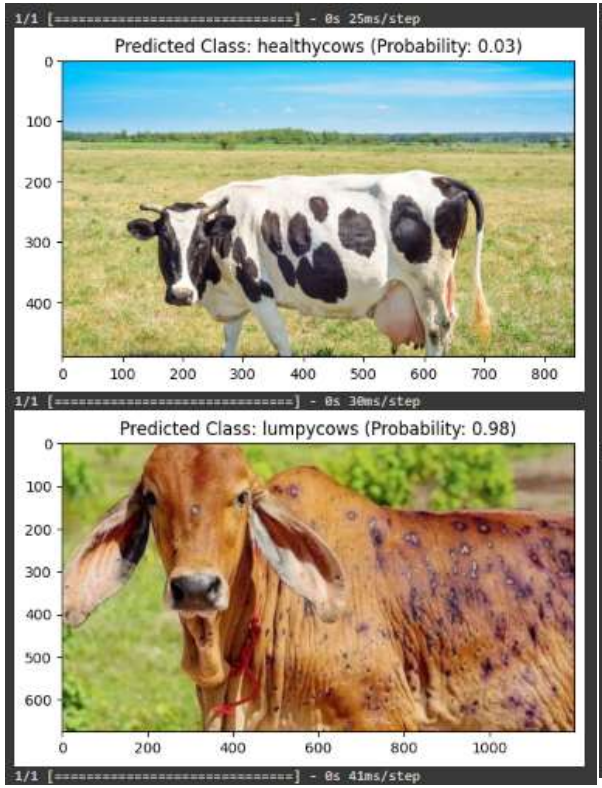
Step 8: Plot Training Results

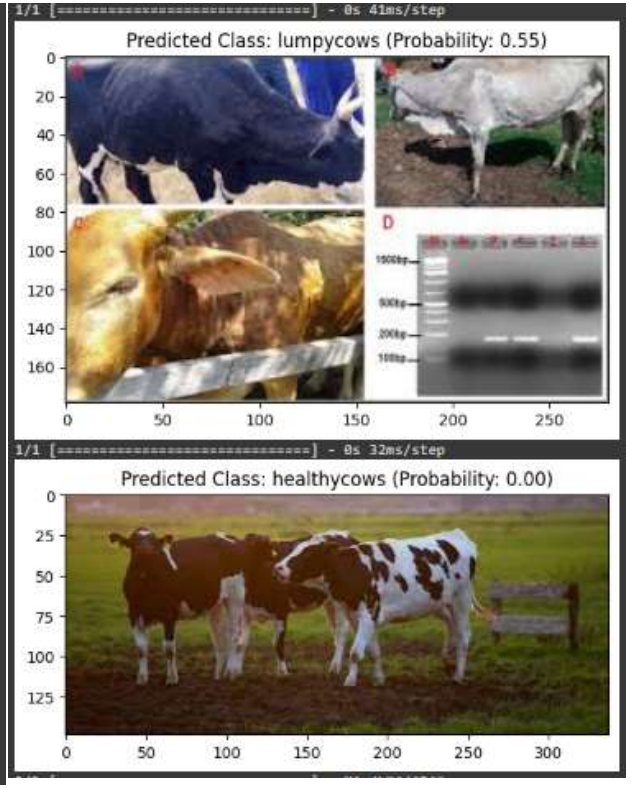
```
[ ]  # Plot training accuracy and loss for cattle classification
     plt.figure(figsize=(12, 4))
     plt.subplot(1, 2, 1)
     plt.plot(history.history['accuracy'], label='Training Accuracy', color='b')
     plt.xlabel('Epoch')
     plt.ylabel('Accuracy')
     plt.title('Training Accuracy')
     plt.legend()

     plt.subplot(1, 2, 2)
     plt.plot(history.history['loss'], label='Training Loss', color='r')
     plt.xlabel('Epoch')
     plt.ylabel('Loss')
     plt.title('Training Loss')
     plt.legend()

     plt.show()
```



Identify skin diseases

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt

# List of file paths to test images
test_image_paths = [
    '/content/drive/MyDrive/Research_Models_Datasets/cattles/hcow.jpg',
    '/content/drive/MyDrive/Research_Models_Datasets/cattles/lcow.jpg',
    '/content/drive/MyDrive/Research_Models_Datasets/cattles/cows/lumpycows/img1113.jpg',
    '/content/drive/MyDrive/Research_Models_Datasets/cattles/cows/healthycows/imgs006.jpg',
    '/content/drive/MyDrive/Research_Models_Datasets/cattles/LSD.jpg',
    # Add more image paths here
]

for image_path in test_image_paths:
    # Load the test image
    img = cv2.imread(image_path)

    # Preprocess the image (resize and normalize)
    resized_img = cv2.resize(img, (256, 256))  # Resize the image
    normalized_img = resized_img / 255.0  # Normalize the pixel values

    # Make a prediction
    prediction = model.predict(np.expand_dims(normalized_img, 0))

    if prediction > 0.5:
        predicted_class = 'lumpycows'
    else:
        predicted_class = 'healthycows'

    # Display the image and prediction
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))  # Convert from BGR to RGB
    plt.title(f"Predicted Class: {predicted_class} (Probability: {prediction[0][0]:.2f})")
    plt.show()
```





Completion and Future works

Completed Component

- Collecting cattle image datasets
- Binary Classification Model
- Advanced Models Implementation

Future Implementation

- severity assessment of cattle skin diseases
- Implement the Mobile application

Functional, Non-Functional Requirements

Functional Requirements

Data Gathering & Preprocessing:

Image Enhancement

CNN Model Selection and Transfer Learning:

Severity Assessment Module:

Non-Functional Requirements

User-friendly interfaces

Availability

Scalability

Security

WIREFRAMES


Main Page
Cattle AID
Cattle Type
Cattle Age
Cattle Breed
Register


Cattle Identification
Upload Image:
Choose Image
Upload


Cattle Identification
Upload Image:
Choose Image
Upload
Result Image:
Healthy Cattle

References

[1] N. S. ALKolifi ALEnezi, “A method of skin disease detection using image processing and machine learning,” Procedia Computer Science, vol. 163, pp. 85–92, 2019. doi:10.1016/j.procs.2019.12.090

[2] S. Gambhir, S. Khanna, and P. Malhotra, "Machine Learning Based Diagnosis of Lumpy Skin Disease," 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, 2023, pp. 1-5, doi: 10.1109/ICAIA57370.2023.10169125.

[3] R. M. N. A. Rathnayaka, K. G. S. N. Anuththara, R. J. P. Wickramasinghe, P. S. Gimhana, L. Weerasinghe and G. Wimalaratne, "Intelligent System for Skin Disease Detection of Dogs with Ontology-Based Clinical Information Extraction," 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, NY, USA, 2022, pp. 0059-0066, doi 10.1109/UEMCON54665.2022.9965696.

[4] A. Kumar, B. Kumar, and H. S. Negi, "Predicting Lumpy Skin Disease using Various Machine Learning Models," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 412-416, doi: 10.1109/CISES58720.2023.10183604.

IT20651824|Jayawardhana K.A.D.D.S

Bachelor of Science (Hons) in Information Technology

Specializing in Software Engineering

Creating an Ontology-based Information Extraction system for the Sri Lankan Cattle Domain.

INTRODUCTION - BACKGROUND

important of Ontology-based Information Extraction system for the Sri Lankan Cattle

- **Centralized Knowledge Repository**
- **Efficient Data Retrieval**
- **Enhanced Decision-Making**
- **Support for Chatbots and Automation**

RESEARCH QUESTION

How can an Ontology-based Information Extraction System be developed and implemented to effectively manage and provide access to comprehensive information in the Sri Lankan cattle domain?

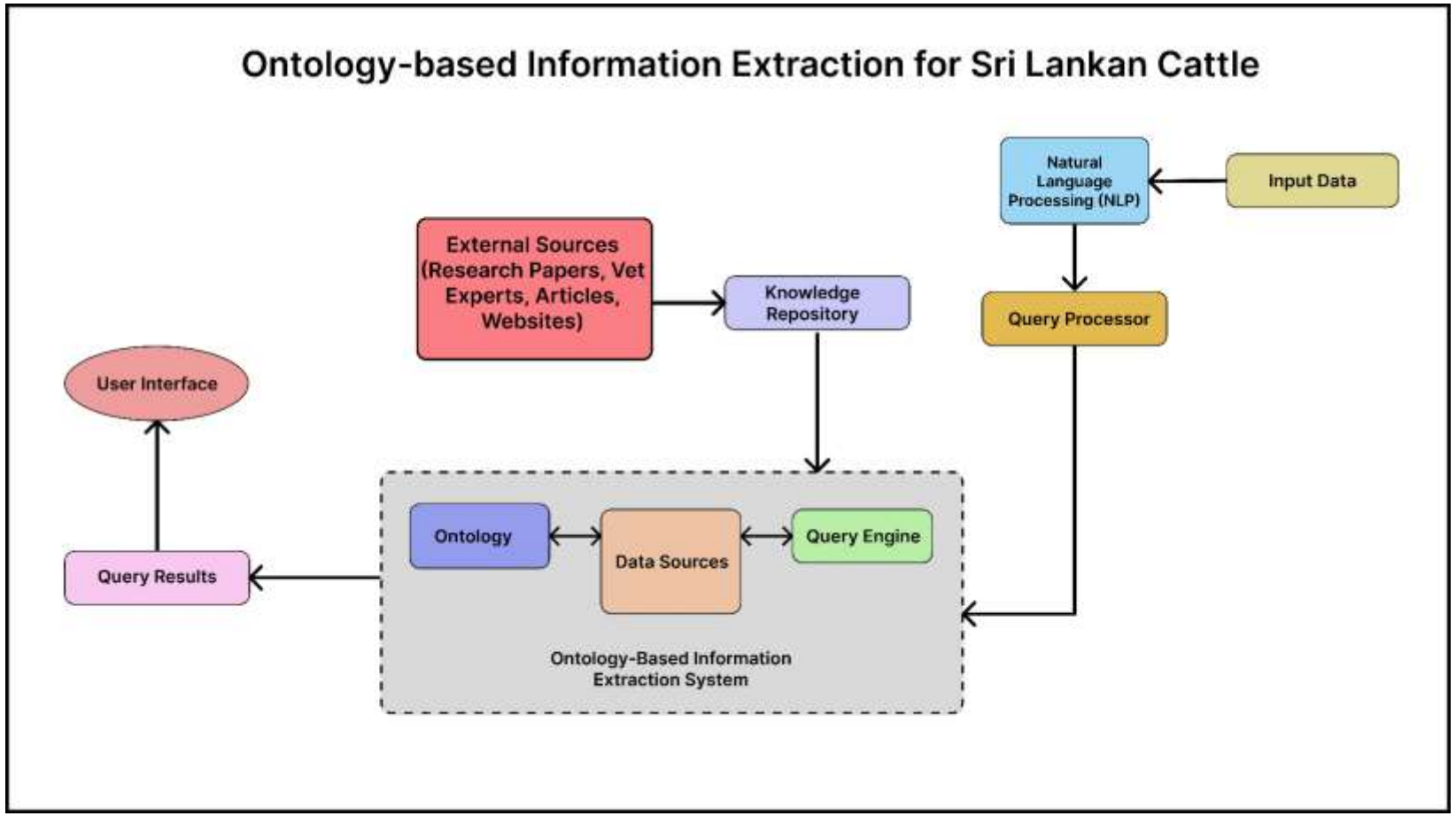
Specific Objective

To create an Ontology-based Information Extraction System tailored to the Sri Lankan cattle domain, enabling efficient access to comprehensive and structured cattle-related knowledge.

Sub Objective

- Ontology Development
- Data Integration
- Query and Reasoning Capabilities

SYSTEM DIAGRAM



- Protégé - ontology development
- OWL (Web Ontology Language) - standard ontology language
- RDF (Resource Description Framework) - represent data in a structured format
- SPARQL Query and DL Query - query language for querying RDF data
- Pellet Reasoner - enabling ontology reasoning
- HermiT 1.4.3 - enabling ontology reasoning
- NLTK (Natural Language Toolkit)
- Android Studio - building user interfaces

1. Ontology Design and Development

2. Data Collection and Integration

3. Ontology Reasoning and Inference

4. Natural Language Processing (NLP)

5. Query Processing and Optimization

6. User Interface Development

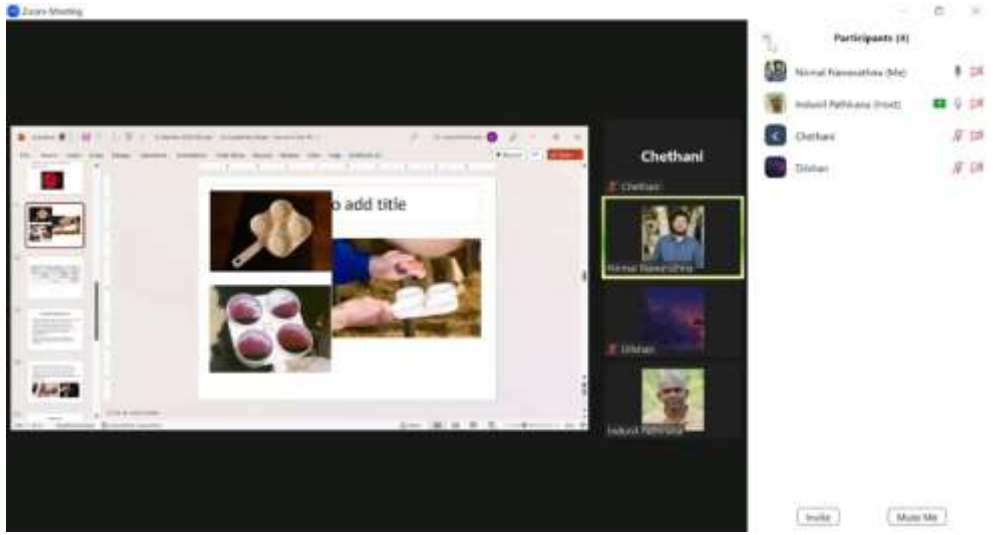
7. Testing, Evaluation, and Maintenance

Data Collection and Integration

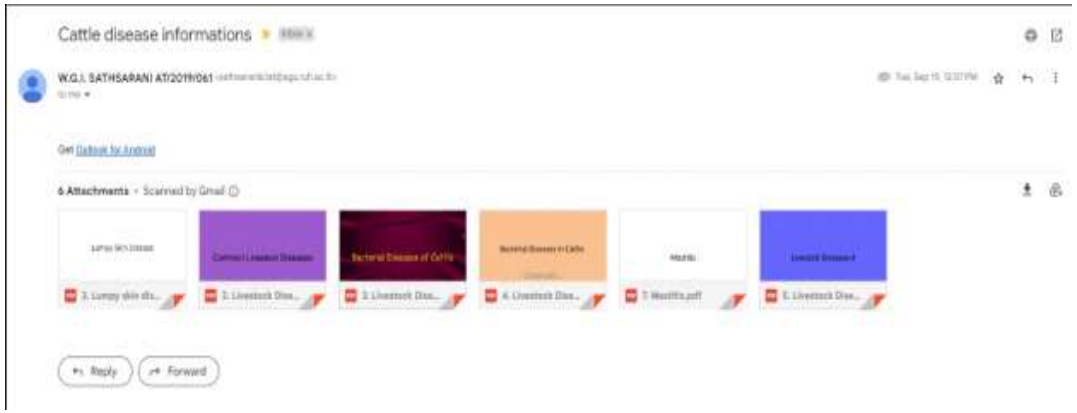
To develop a comprehensive ontology, we leveraged expert knowledge and relevant documents. Notable contributors to our data collection effort include:

- **Expert Consultations:** We met with veterinary surgeon Dr. Thamali and consulted with Dr. Indunil N. Pathirana, a senior lecturer in the animal department at the University of Ruhuna. Their insights enriched our ontology.

- **Document References:** We also referenced research papers, articles, and publications to ensure our ontology aligns with established research and best practices, guaranteeing the reliability and up-to-date nature of our knowledge repository.



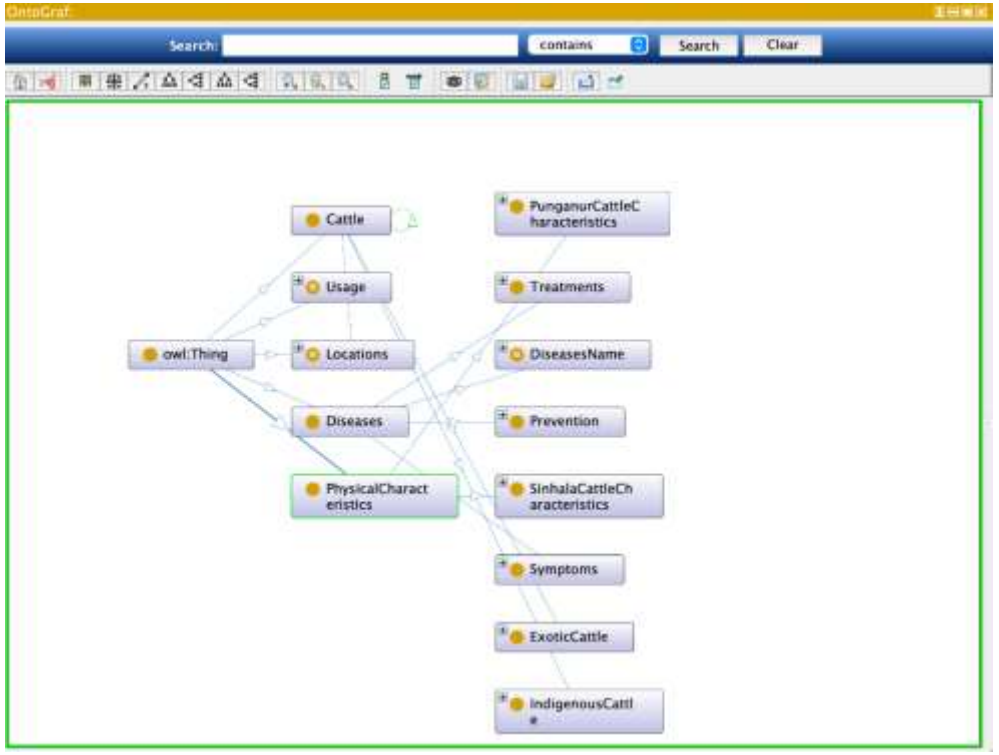
Get Expert Consultation from Dr. Indunil N. Pathirana



Refer related documentation

Ontology Design and Development

In the process of designing and developing our ontology for the Sri Lankan cattle domain, the first crucial step was to structure the main class and its subclasses. This hierarchical organization forms the backbone of our ontology, providing a clear framework for categorizing and organizing knowledge.



SriLankaCattleOntology.rdf

Users > dilshansudharaka > Desktop > Ontology > Sri Lanka Cattle Ontology > SriLankaCattleOntology.rdf

```
<!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Cattle -->

<owl:Class rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Cattle"/>


<!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Diseases -->

<owl:Class rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Diseases"/>


<!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#DiseasesName -->

<owl:Class rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#DiseasesName">
    <owl:equivalentClass>
        <owl:Class>
            <owl:oneOf rdf:parseType="Collection">
                <rdf:Description rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#LumpySkin"/>
                <rdf:Description rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Papillomatosis"/>
            </owl:oneOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Diseases"/>
</owl:Class>


<!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#ExoticCattle -->

<owl:Class rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#ExoticCattle">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Cattle"/>
</owl:Class>
```

Initializing Individual Instances

In the second critical step of ontology development, we initiated the individual instances within our ontology. These individual instances represent specific entities within the Sri Lankan cattle domain, further enriching the knowledge repository and making it more context-specific. Below, we highlight some of the key topics and instances that have been introduced.

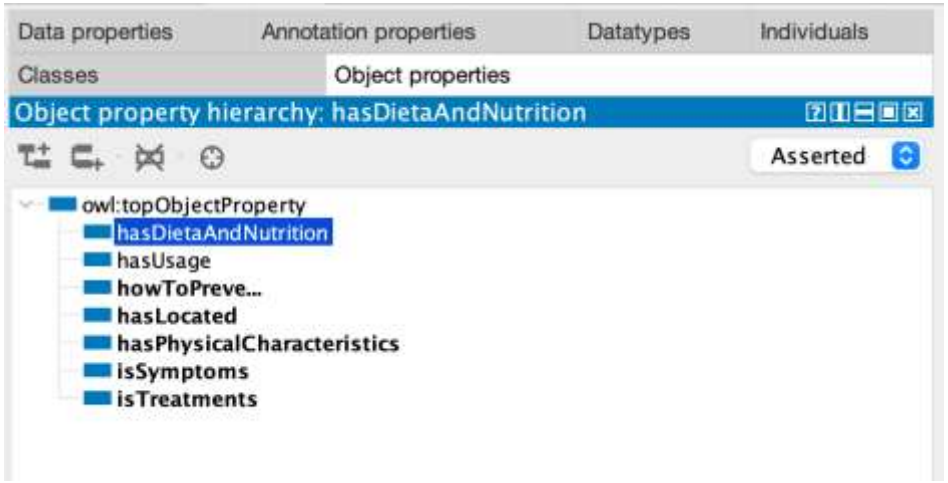
SriLankaCattleOntology.rdf

Users > dilshansudharaka > Desktop > Ontology > Sri Lanka Cattle Ontology > SriLankaCattleOntology.rdf

```
451
452
453     <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Avoiding_Contact_with_Wild_Animals -->
454
455     <owl:NamedIndividual rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Avoiding_Contact_with_Wild_Animals">
456         <rdf:type rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#PapillomatosisPrevention"/>
457     </owl:NamedIndividual>
458
459
460
461     <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#CoastalAreas -->
462
463     <owl:NamedIndividual rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#CoastalAreas"/>
464
465
466
467     <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#CoatType_-_short_sleek_coat -->
468
469     <owl:NamedIndividual rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#CoatType_-_short_sleek_coat">
470         <rdf:type rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#SinhalaCattleCharacteristics"/>
471     </owl:NamedIndividual>
472
473
474
475     <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Coat_Type_-_Smooth_sleek_coat -->
476
477     <owl:NamedIndividual rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Coat_Type_-_Smooth_sleek_coat">
478         <rdf:type rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#PunganurCattleCharacteristics"/>
479     </owl:NamedIndividual>
480
481
482
483     <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Color_-_Brown -->
484
485     <owl:NamedIndividual rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Color_-_Brown"/>
486
487
```

Establishing Relationship

In this step, we created relationships that connect different classes and link classes to individual instances within the ontology, aligning them with the complex and interrelated structure of the Sri Lankan cattle domain. These relationships facilitate the organization of information, allowing us to represent how cattle types relate to specific diseases, how treatments are associated with particular symptoms, how medicines are linked to diseases, and how geographical distribution affects cattle types, all within a structured and interconnected framework.



SriLankaCattleOntology.rdf

Users > dilshansudharaka > Desktop > Ontology > Sri Lanka Cattle Ontology > SriLankaCattleOntology.rdf

```
    <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasDietaAndNutrition -->

    <owl:ObjectProperty rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasDietaAndNutrition"/>



    <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasLocated -->

    <owl:ObjectProperty rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasLocated">
        <rdfs:domain rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Cattle"/>
        <rdfs:range rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Locations"/>
    </owl:ObjectProperty>



    <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasPhysicalCharacteristics -->

    <owl:ObjectProperty rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasPhysicalCharacteristics">
        <rdfs:domain rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Cattle"/>
        <rdfs:range rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Cattle"/>
    </owl:ObjectProperty>



    <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasUsage -->

    <owl:ObjectProperty rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#hasUsage"/>



    <!-- http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#howToPrevent -->

    <owl:ObjectProperty rdf:about="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#howToPrevent">
        <rdfs:domain rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#Prevention"/>
        <rdfs:range rdf:resource="http://www.semanticweb.org/dilshansudharaka/ontologies/2023/10/cattle-ontology#DiseasesName"/>
    </owl:ObjectProperty>
```

Completed Component

1. Ontology Design and Development
2. Data Collection and Integration
3. Ontology Reasoning and Inference
5. Query Processing and Optimization

Future Implementation

4. Natural Language Processing (NLP)
6. User Interface Development
7. Testing, Evaluation, and Maintenance

Functional Requirements

- Query Processing and Retrieval
- Ontology Maintenance and Update
- Knowledge Integration

Non-Functional Requirements

- Scalability
- Performance and Speed
- Security and Privacy

References

[1] [Online]. Available: https://www.agrimin.gov.lk/web/.

[2] [Online]. Available: http://repo.lib.sab.ac.lk:8080/xmlui/bitstream/handle/susl/2294/2022-Aginsight-Proceeding-282-284.pdf?sequence=1&isAllowed=y.

[3] [Online]. Available: https://www.agriculture.gov.au/biosecurity-trade/pests-diseasesweeds/animal/lumpy-skindisease#:~:text=Lumpy%20skin%20disease%20(LSD)%20is,humans%20by%20eating%20affected%20meat.

[4] K. G. S. N. A. R. J. P. W. P. S. G. L. W. a. G. W. R. M. N. A. Rathnayaka, "Intelligent System for Skin Disease Detection of Dogs with Ontology Based Clinical Information Extraction," 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, NY, USA, 2022, pp. 0059-0066, doi: 10.1109/UEMCON54665.2022.9965696.

[5] D. D. A. M. S. F. a. H. Z. F. Gutierrez, "Hybrid Ontology-Based Information Extraction for Automated Text Grading," 2013 12th International Conference on Machine Learning and Applications, Miami, FL, USA, 2013, pp. 359-364, doi: 10.1109/ICMLA.2013.73.

[6] G. Z. F. K. Q. Z. a. P. Q. L. Qian, "Tree Kernel-Based Semantic Relation Extraction Using Unified Dynamic Relation Tree," 2008 International Conference on Advanced Language Processing and Web Information Technology, Dalian, China, 2008, pp. 64-69, doi: 10.1109/ALPIT.2008.26.

References

[7] A. S. A. K. J. a. R. G. T. N. Ujjwal, "Exploiting Machine Learning for Lumpy Skin Disease Occurrence Detection," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964656.

[8] B. K. a. H. S. N. A. Kumar, "Predicting Lumpy Skin Disease using Various Machine Learning Models," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 412-416, doi: 10.1109/CISES58720.2023.10183604.

IT20659226|M.G.K.Chethani

Bachelor of Science (Hons) in Information Technology

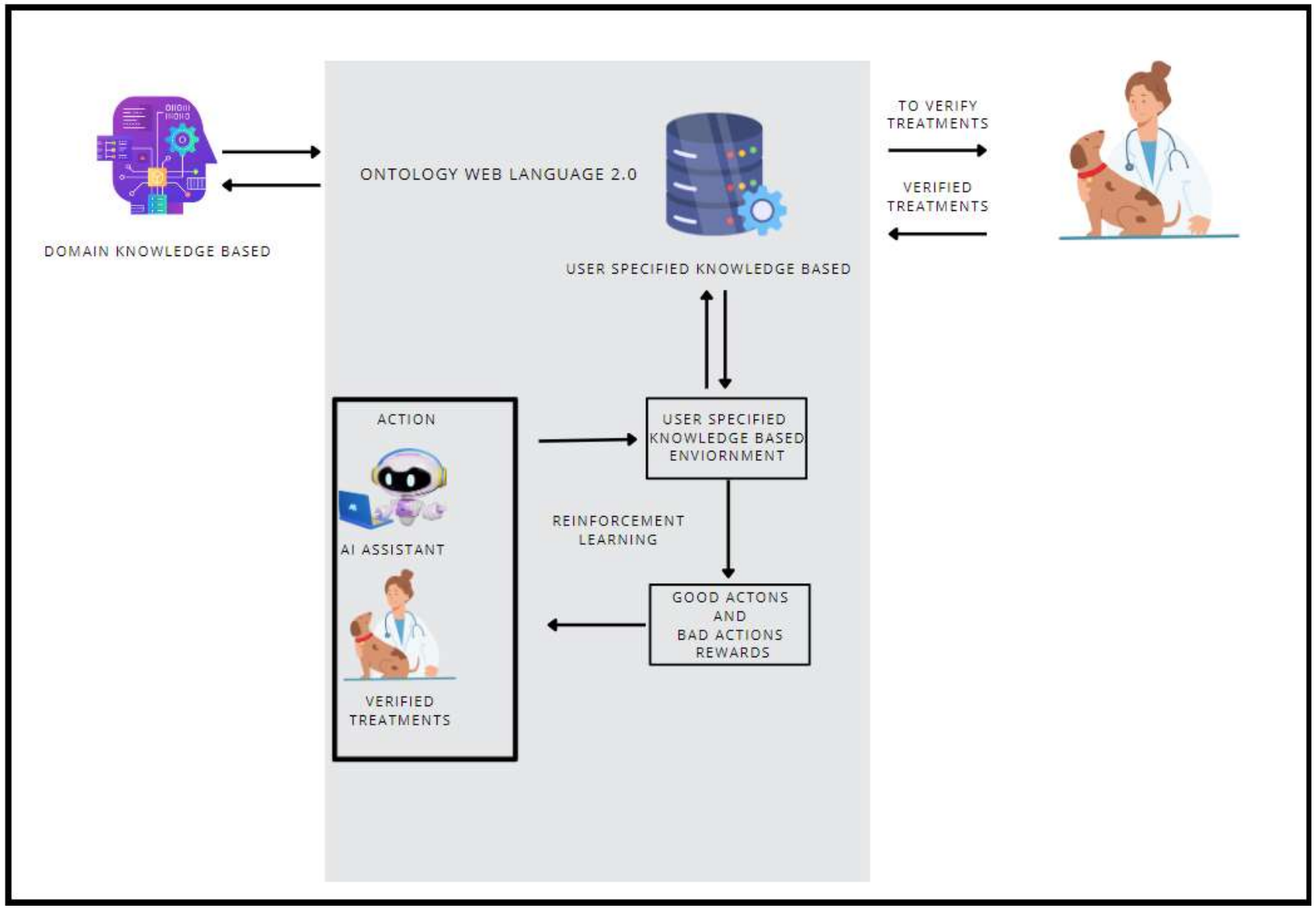
Specializing in Information Technology

User specific knowledge based up to date using reinforcement learning.

INTRODUCTION - BACKGROUND

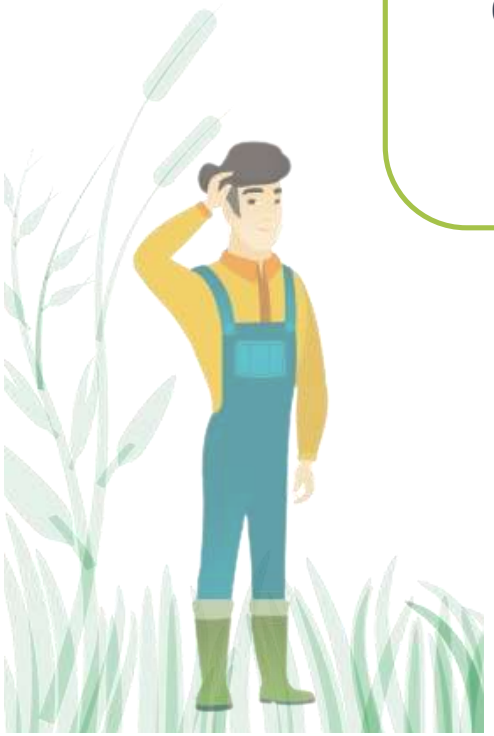
This component I am to creating a user specific knowledge based by filtering the treatment details from the domain knowledge based, we contact a vet through a mobile application to clarify that suggest treatment details are can be used for those treatments, verified treatment details and suggestions needs up to date in the user specific knowledge based. When user asking questions from AI assistant according to that user specific knowledge based should be up to date.

SYSTEM DIAGRAM



RESEARCH QUESTION

How can advanced technologies, be effectively applied to improve the diagnosis and management of cattle skin diseases in the livestock industry?



Data Collecting and analyzing

Developing a mobile application for verify veterinary information from vet

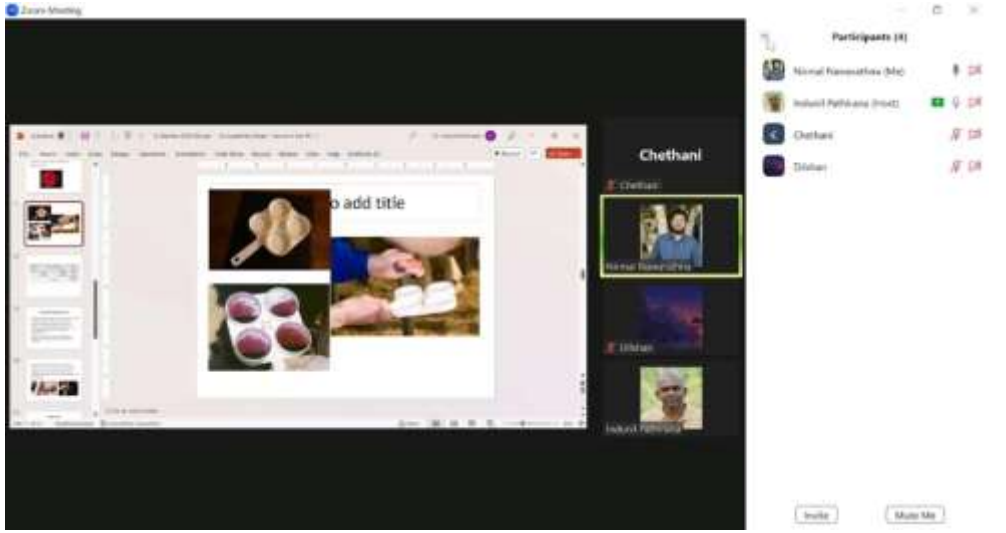
Knowledge Representation and Storage

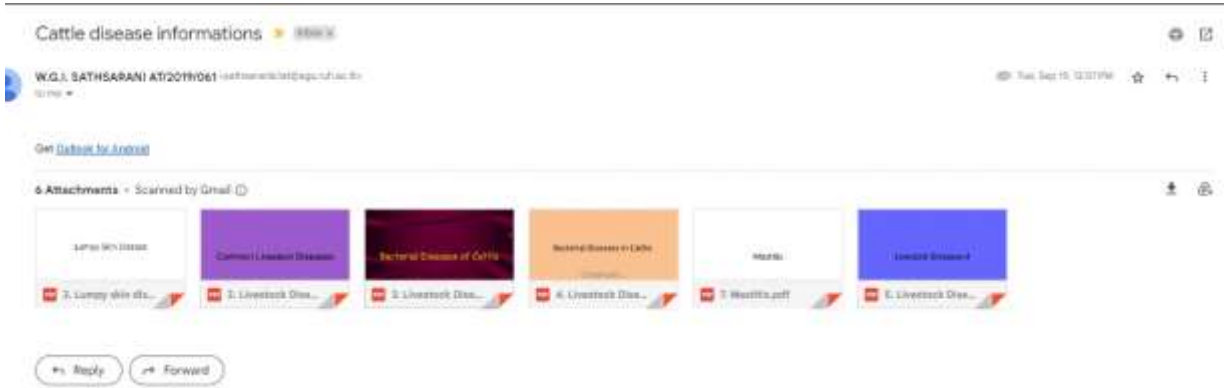
Knowledge Update and Adaptation

Data Collecting and analyzing

SUB OBJECTIVE 01

- Meet a veterinary surgeon to collect data, Dr.Thamali
- Dr. Indunil N. Pathirana consulting-senior lecture in animal department of University of Ruhuna.

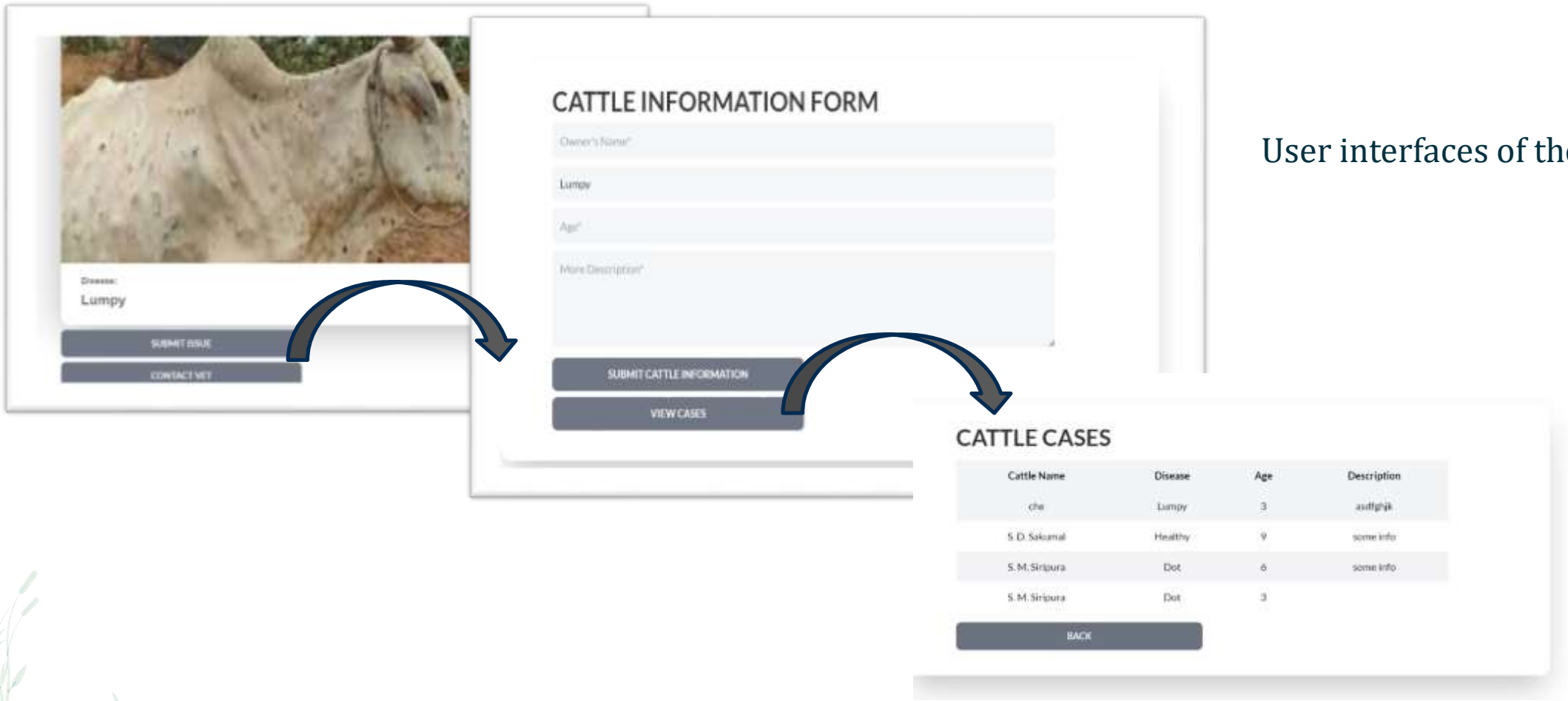




Cattle skin Disease	Serverity	Treatments
Ringworm (Dermatophytosis)	Low, Medium High Low,Medium,High	Topical antifungal cream, ointments containing compounds like miconazole or clotrimazole. Oral antifungal medications Clean and disinfect housing and equipment to minimize the spread of the disease.
Mange (Sarcoptic or Demodectic)		Ivermectin or doramectin injections for sarcoptic mange Miticides and topical treatments may be prescribed for demodectic mange
Dermatophilosis (Rain Rot)		Antibiotics like penicillin or oxytetracycline for secondary bacterial infections
Lice Infestation	Low, Medium High	Insecticides specifically designed for cattle, such as permethrin or pyrethrin-based products Injectable products for severe infestations
Photosensitization		Identify and remove the causative plants from the diet. Provide shade and protection from sunlight

Developing the application for verify veterinary information from vet

SUB OBJECTIVE 02



User interfaces of the vet contacting

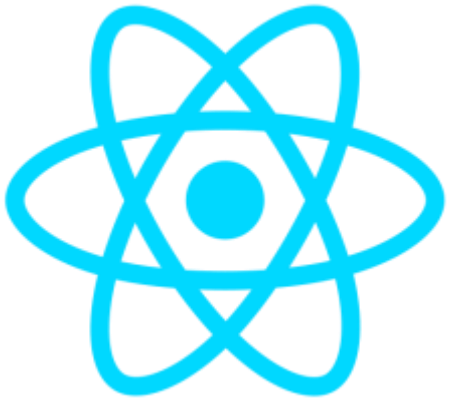
Developing the application for verify veterinary information from vet

Backend implementations

```
const router = require("express").Router();

router.post("/add", async (req, res) => {

    console.log(req.body);

    const newrec = new Checking({
      owner: req.body.owner,
      disease: req.body.disease,
      age: req.body.age,
      description: req.body.description,
      img: req.body.img,
      doctor: req.body.doctor,
      result: req.body.result,
    });

    try {
      const savedRec = await newrec.save();
      res.status(201).json(savedRec);
    } catch (err) {
      console.log(err);
      res.status(500).json(err);
    }
});
```

```
//GET ALL USER
router.get("/", async (req, res) => {
//    const query = req.query.new;
  try {
    // const checkings = query
    //    ? await User.find().sort({ _id: -1 }).limit(5)
    //    : await User.find();
    const checkings = await User.find().sort({ _id: -1 })
    res.status(200).json(checkings);
  } catch (err) {
    res.status(500).json(err);
  }
});
```

DB of the specific user cattle information with the diseases



- React
- Express Js
- MongoDB
- Python
- OWL (Web Ontology Language) - standard ontology language
- Reinforcement learning algorithms

Completed

Data Collecting and analyzing

Developing the application to contact from the vet.

Future works

Knowledge Update and Adaptation

Knowledge-based Decision Making

Functional Requirements

Data collecting and analyzing

Knowledge Update and Adaptation

Knowledge Based decision making

Non-Functional Requirements

User-friendly interfaces

Availability

Scalability

Security

References

[1] Ghayvat, H., Al-Sharabi, M., & Mukhopadhyay, S. C. (2017). Ontology-based knowledge representation for smart agriculture. IEEE Transactions on Industrial Informatics, 13(6), 3172-3180

[2] Nauta, M. J., Borne, J. J. G. C., Berg, P., & Klei, L. (2015). Development of an ontology for cattle disease and associated phenotypes. The 6th International Conference on Biomedical Ontology (ICBO-2015), CEUR Workshop Proceedings

[3] Thomas, J., & Stiles, J. (2013). Ontology-based decision support system for diagnosis and treatment of cattle diseases. Computers in Industry, 64(6), 756-767

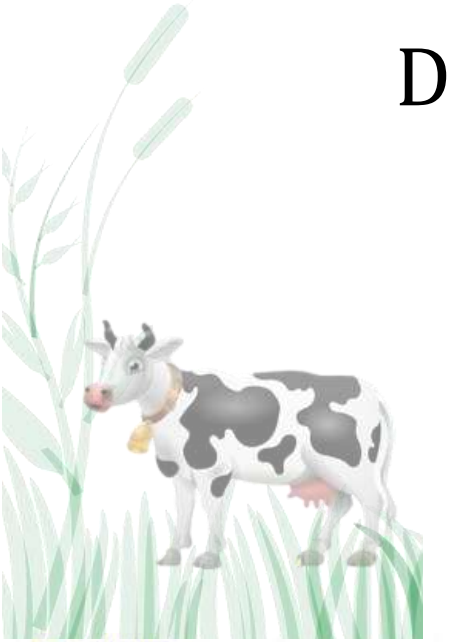
[4] Krithivasan, R., Venkatesan, R., & Samy, S. G. (2019). Ontology-based diagnosis and management of cattle skin diseases using machine learning. Computers in Biology and Medicine, 109, 173-182.

IT20299552|Rubasinghe L.M

Bachelor of Science (Hons) in Information Technology

Specializing in Information Technology

- AI-Driven Smart Assistant for Cattle Skin Diseases

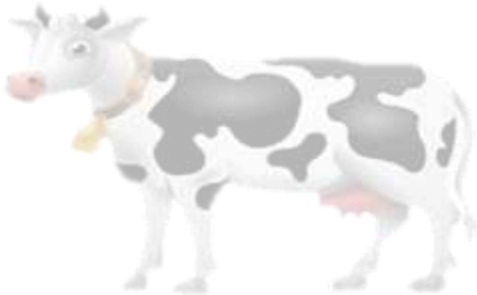


INTRODUCTION - BACKGROUND

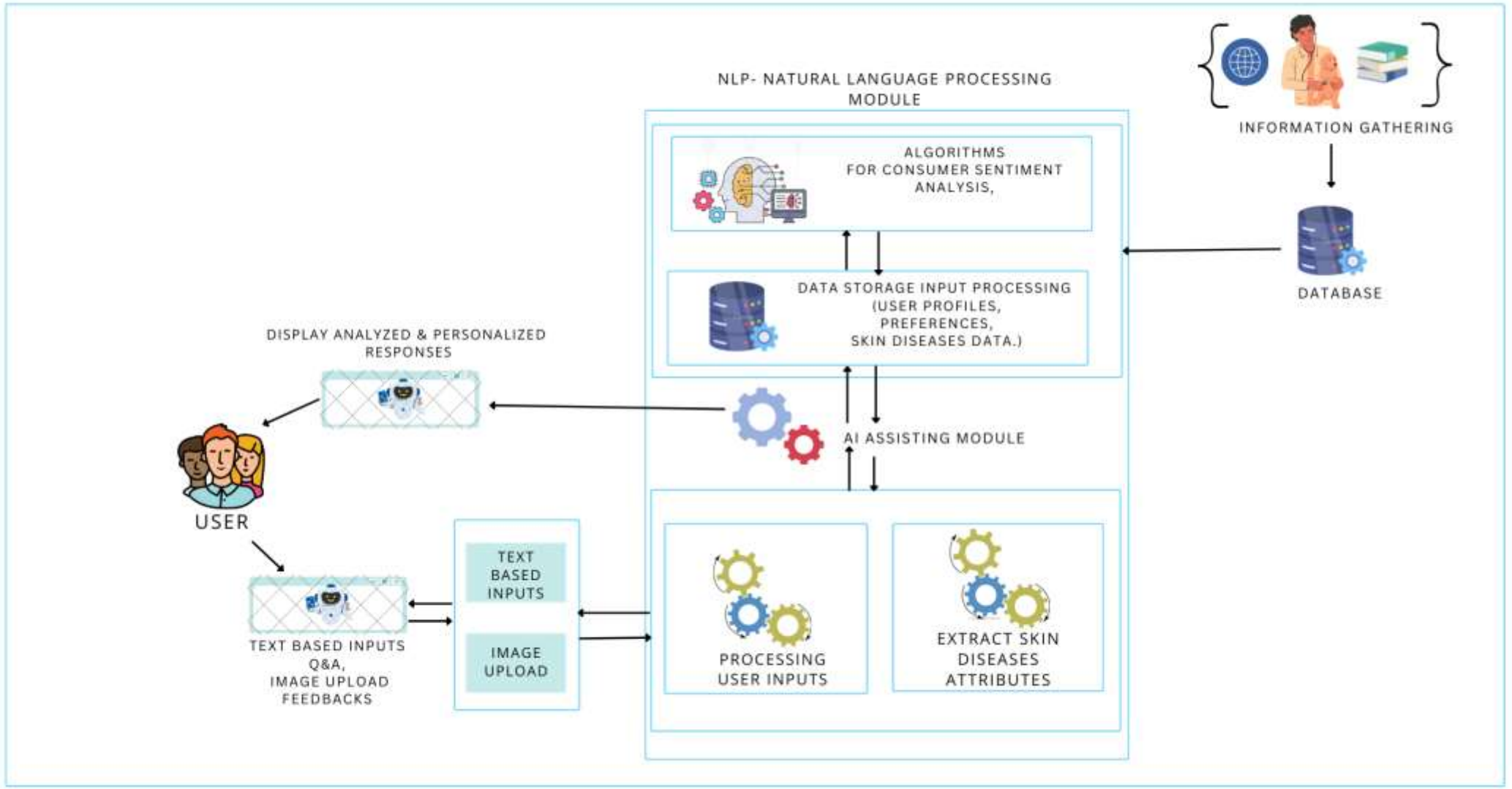
- Focus on building AI-driven smart assistant using sentiment analysis, tailored to the domain of cattle skin diseases.
- The smart assistant can interact with users (veterinarians, cattle farmers, etc.) to provide information about disease detection, severity assessment and potential treatments.
- Sentiment analysis can be integrated to understand user emotions and responses, enhancing the interaction experience.
- By processing textual interaction the assistant offers personalized advice, treatment suggestions, and educational information, fostering better collaboration and informed decision-making.

Importance of the AI Smart Assistant for Cattle Skin Diseases

- **Early Disease Detection -** AI assistants can help identify skin diseases in cattle at an early stage. Timely detection allows for prompt treatment and containment, reducing the risk of disease spread within the herd.
- **Emotional Support -** An emotionally intelligent AI assistant can provide emotional support to farmers and caretakers who may experience stress and anxiety when dealing with disease outbreaks, creating a more empathetic and supportive environment.
- **Educational Resource -** AI assistants can serve as educational tools, helping farmers and veterinarians learn about different cattle skin diseases, their symptoms, and effective treatment methods.
- **Continuous Monitoring -** AI assistants can continuously monitor the health and well-being of cattle, allowing for proactive disease prevention and management.
- **Reduced Economic Losses -** Cattle farmers often suffer economic losses due to disease outbreaks. AI assistants can minimize these losses by assisting with early intervention and disease management.



SYSTEM DIAGRAM



RESEARCH QUESTION

In the field of handling the health of cattle, an important gap exists in the capability of artificial intelligence (AI) to understand and react to the complicated emotional expressions of veterinarians and farmers. The difficulty is in creating an AI system that can dynamically change its responses depending on developing emotional context within domain-specific conversations on cattle skin diseases.

Specific Objectives

- Data Analyzing of cattle skin diseases
- Develop an Emotionally Intelligent AI Smart Assistant
- Implement an Advanced Dialog Management
- Generate personalized care plans
- Address Ethical Considerations and Data Privacy

Sub Objectives

- Analyze Data
- Real-Time Updates
- Data Integration and Compatibility
- Anonymization and Data Security
- Dynamic response and adaption

METHODOLOGY

TECHNOLOGIES, TECHNIQUES, ALGORITHMS

Tools & Libraries

Visual Code

Google Colab

Python Libraries for NLP

spaCy for text processing

Algorithms

Web Scrapping

Clustering Algroithms

BERT, VADER for sentiment analysis

TextBlob for processing textual data.

Technology Stacks

Rasa, DialogFlow for conversational experience

React/ ReactNative

Node.Js/ Java- Android

Technologies & Services

Git/ GitHub/ GitLab

Cloud Services

NoSQL Databases

TensorFlow

Completion and Future works

Completed Tasks

- Collecting different cattle skin disease images/ treatment datasets.
- Trained a model to identify cattle skin diseases
- Partly trained smart assistant to interact with the user
- Developed a web application to be converted to a mobile application.

Future Tasks

- Enhanced the smart assistant with sentiment analyzing.
- Trained the models to identify more skin diseases/ treatment details.
- Enhanced the UI designs of the web application.
- Develop a mobile application with all integrations.

Data Preprocessing

```
# Load your CSV dataset
data = pd.read_csv('/content/drive/MyDrive/Cattles Project/Projet/training/_classes.csv')

# Define the image directory (the folder where your images are stored)
image_directory = '/content/drive/MyDrive/Cattles Project/Projet/training/'

# Preprocess the data
image_paths = data['filename'].apply(lambda x: image_directory + x)
labels = data.drop(columns=['filename'])

# Load and preprocess images
X = []
for img_path in image_paths:
    img = image.load_img(img_path, target_size=(48, 48), grayscale=True)
    img = image.img_to_array(img)
    img /= 255.0
    X.append(img)

X = np.array(X)
y = labels.values
```

Train the model

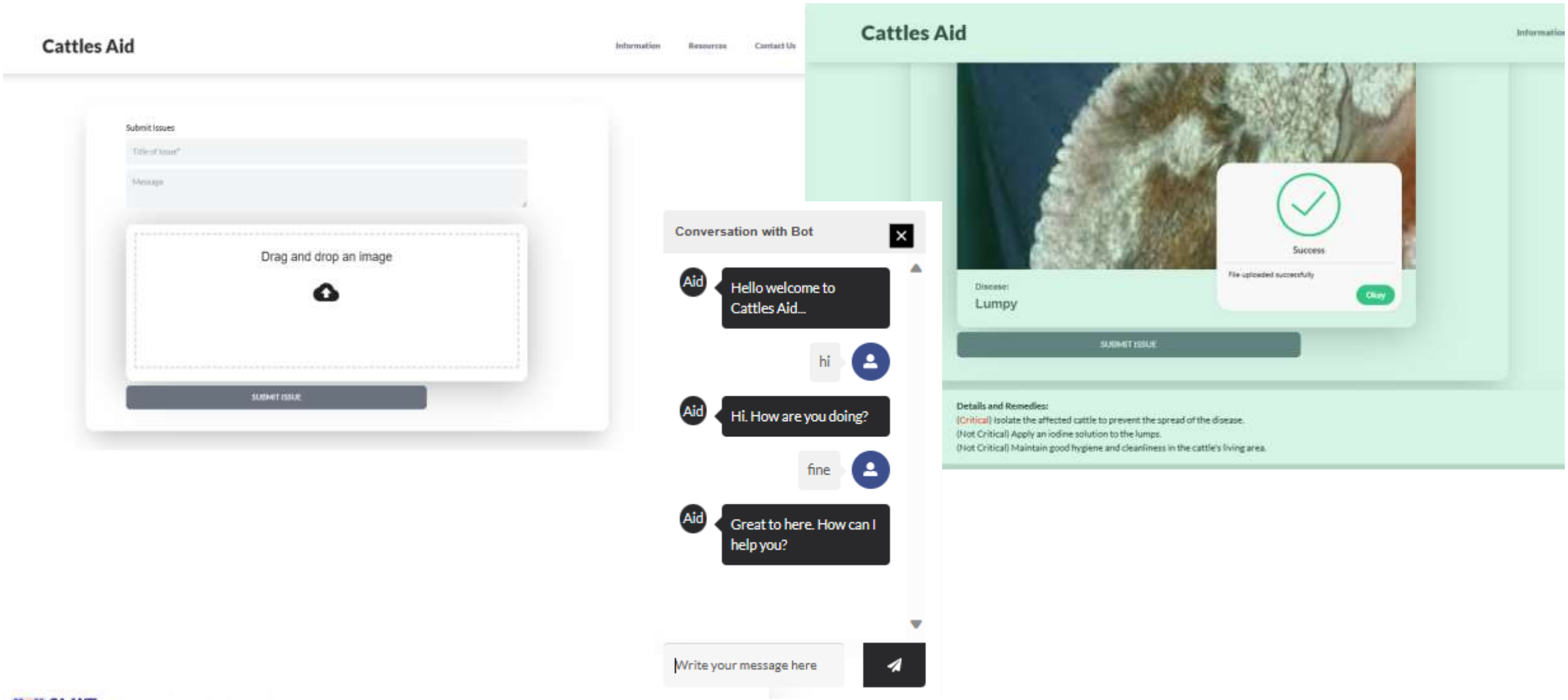
```
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
batch_size = 32
epochs = 150

history = model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs,
                    verbose=1, validation_data=(X_val, y_val))

# Save the model
model.save('model_file3.h5')
```

Validating the Dataset

```
# Split data into train and validation sets
X, y = shuffle(X, y, random_state=42)  # Shuffle the data
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)
```

Defining the model

```
# Define your model
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48, 1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(y.shape[1], activation='sigmoid'))  # Output layer with the number of categories
```


Cattles Aid
Information
Resources
Contact Us
Submit Issues
Drag and drop an image
SUBMIT ISSUE
Conversation with Bot
Aid
Hello welcome to Cattles Aid...
hi
Aid
Hi. How are you doing?
fine
Aid
Great to here. How can I help you?
Write your message here
Cattles Aid
Success
File uploaded successfully
Okay
Disease:
Lumpy
SUBMIT ISSUE
Details and Remedies:
(Critical) Isolate the affected cattle to prevent the spread of the disease.
(Not Critical) Apply an iodine solution to the lumps.
(Not Critical) Maintain good hygiene and cleanliness in the cattle's living area.

Functional and Non-Functional Requirements

Functional Requirements

USER INPUTS ANALYSIS

REAL-TIME EMOTIONAL MONITORING

DYNAMIC RESPONSE ADAPTION

DIALOG MANAGEMENT

PRIVACY AND DATA PROTECTION

ENHANCE THE USER EXPERIENCE

USER FEEDBACK AND RATING SYSTEM

APIS AND INTEGRATIONS

Non - Functional Requirements

- Performance
- Accuracy and Personalization
- Usability and User Experience
- Reliability and Availability
- Security
- Compatibility
- User Experience and Engagement

References

- [1] Kumar T, Rajesh & .K, Abinaya. (2020). An AI Based Chat-Bot for Providing Health Services. Test Engineering and Management. 83. 3721-3726.
- [2] A. Parikh, K. Patel and B. Shah, "A service oriented collaborative model for cattle health care system," *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, 2015, pp. 1352-1356, doi: 10.1109/ICACCI.2015.7275800
- [3] P. Y. Niranjan, V. S. Rajpurohit and R. Malgi, "A Survey on Chat-Bot system for Agriculture Domain," *2019 1st International Conference on Advances in Information Technology (ICAIT)*, Chikmagalur, India, 2019, pp. 99-103, doi: 10.1109/ICAIT47043.2019.8987429.
- [4] S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas and B. Santhosh, "Artificial Intelligence-based Voice Assistant," *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, London, UK, 2020, pp. 593-596, doi: 10.1109/WorldS450073.2020.9210344.
- [5] J. Bang, H. Noh, Y. Kim and G. G. Lee, "Example-based chat-oriented dialogue system with personalized long-term memory," *2015 International Conference on Big Data and Smart Computing (BIGCOMP)*, Jeju, Korea (South), 2015, pp. 238-243, doi: 10.1109/35021BIGCOMP.2015.7072837.