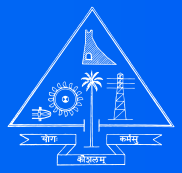
Plant Leaf Disease Detection Using ResNet-50

Ajay Jeevan Jose
S3 MCA
TCR21MCA-2004

Guided By:
Sminesh C N
Associate Professor

Department of Computer Applications
Government Engineering College, Thrissur, Kerala

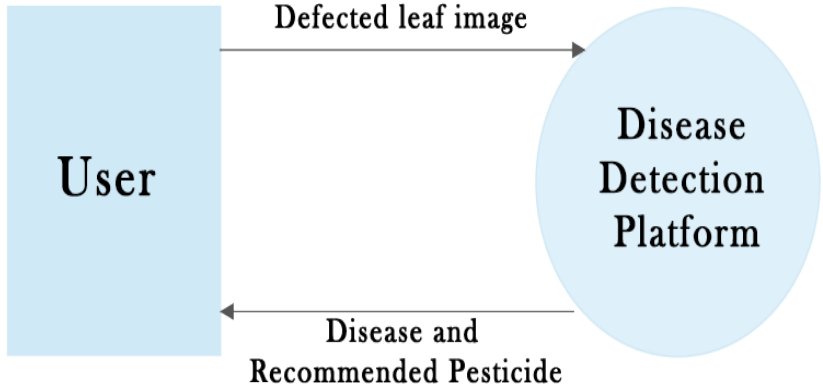
Outline



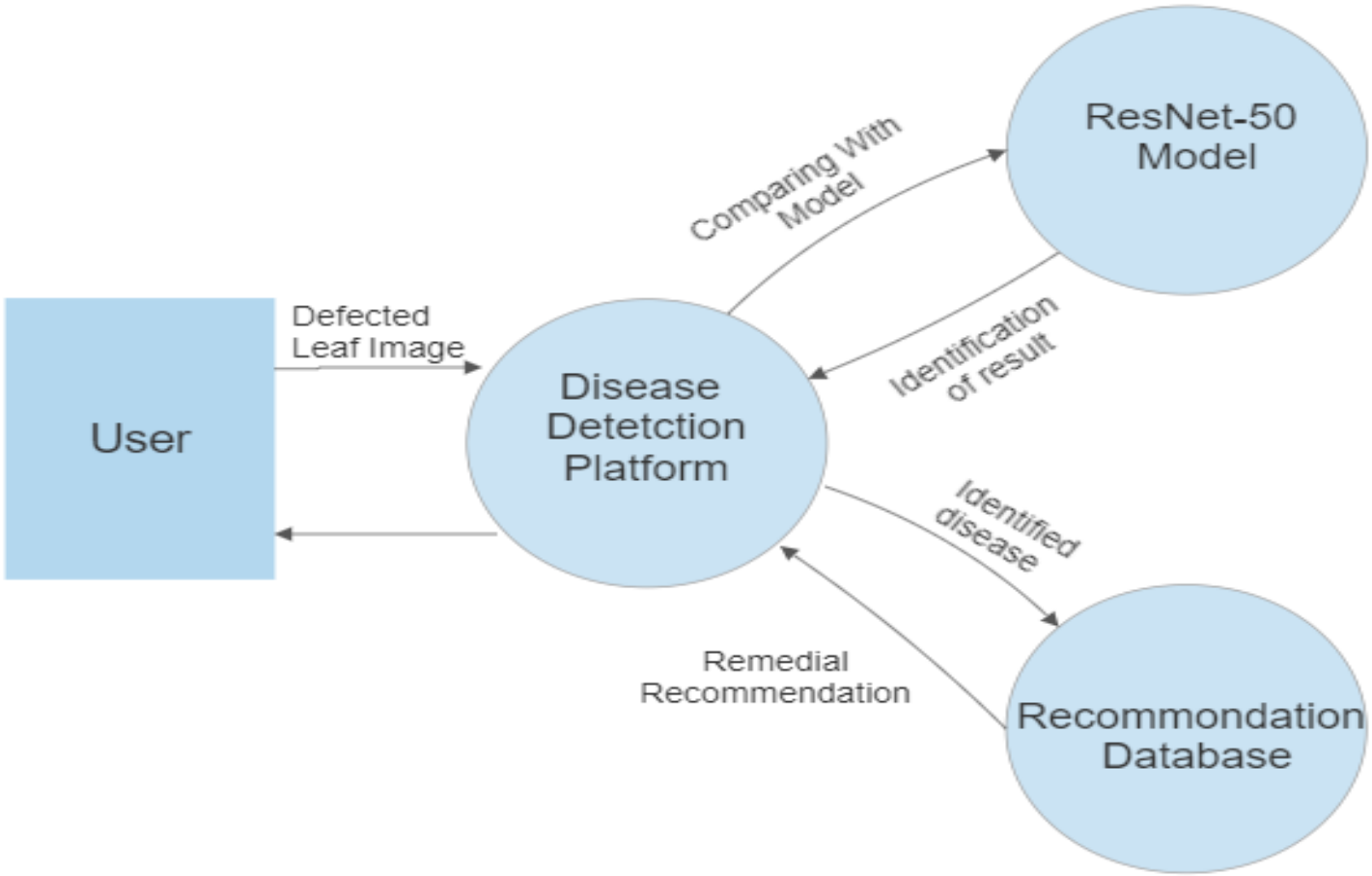
Introduction

Early disease detection is crucial for improved crop yield and quality. Due to a decline in the quality of the agricultural produce, diseased plants can cause large financial losses for individual farmers.In a nation like India, where a substantial section of the population relies on agriculture for a living, it is essential to spot the disease at its earliest stages.A precise diagnosis of the plant disease might reduce losses.The objective of this research is to develop a model that can correctly forecast whether a leaf is disease-infected or not.The main objectives of this study include identifying plant disease and suggesting pesticides that can help to reduce the crop loss.

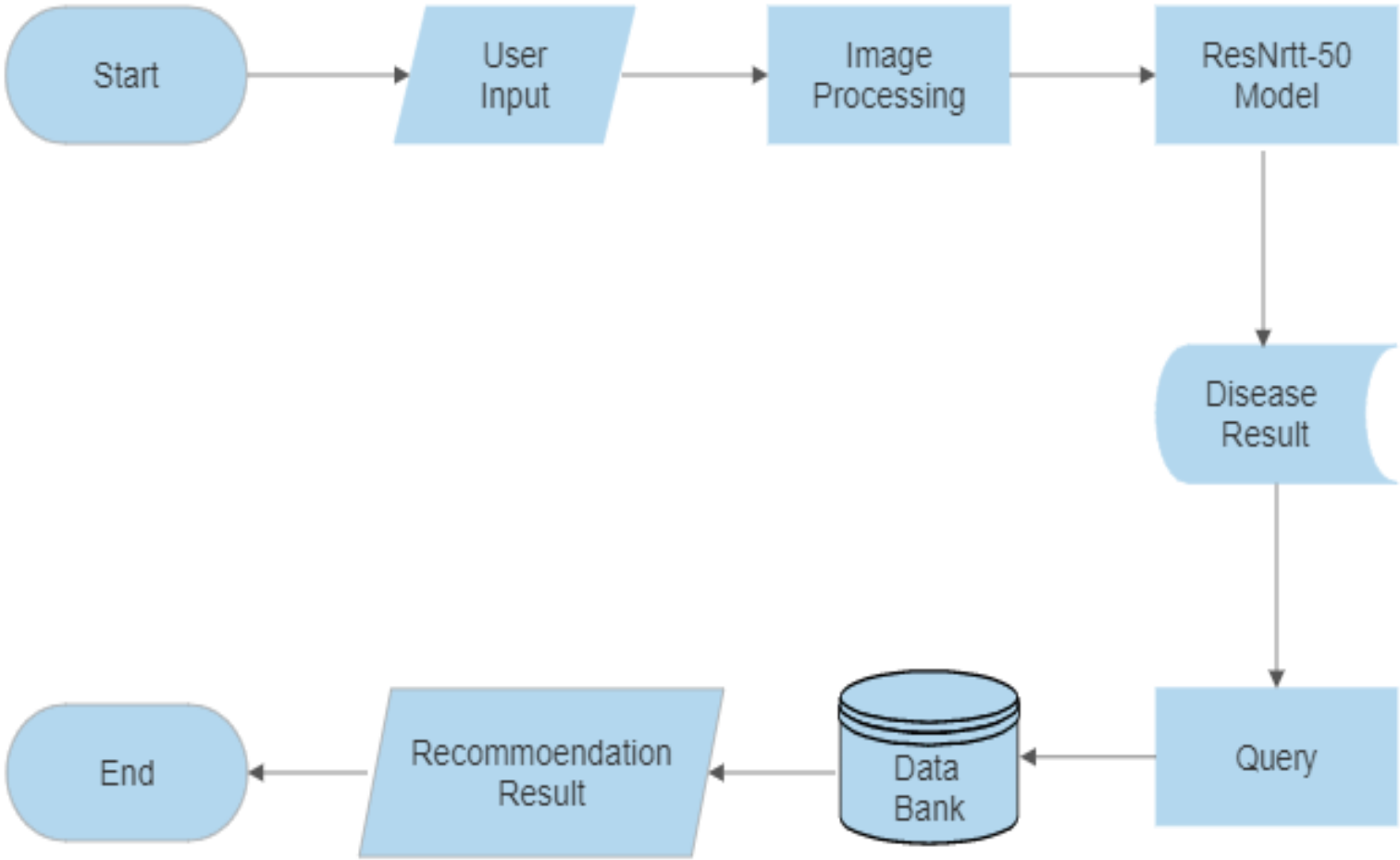
DFD Level 0


User
Defected leaf image
Disease
Detection
Platform
Disease and
Recommended Pesticide

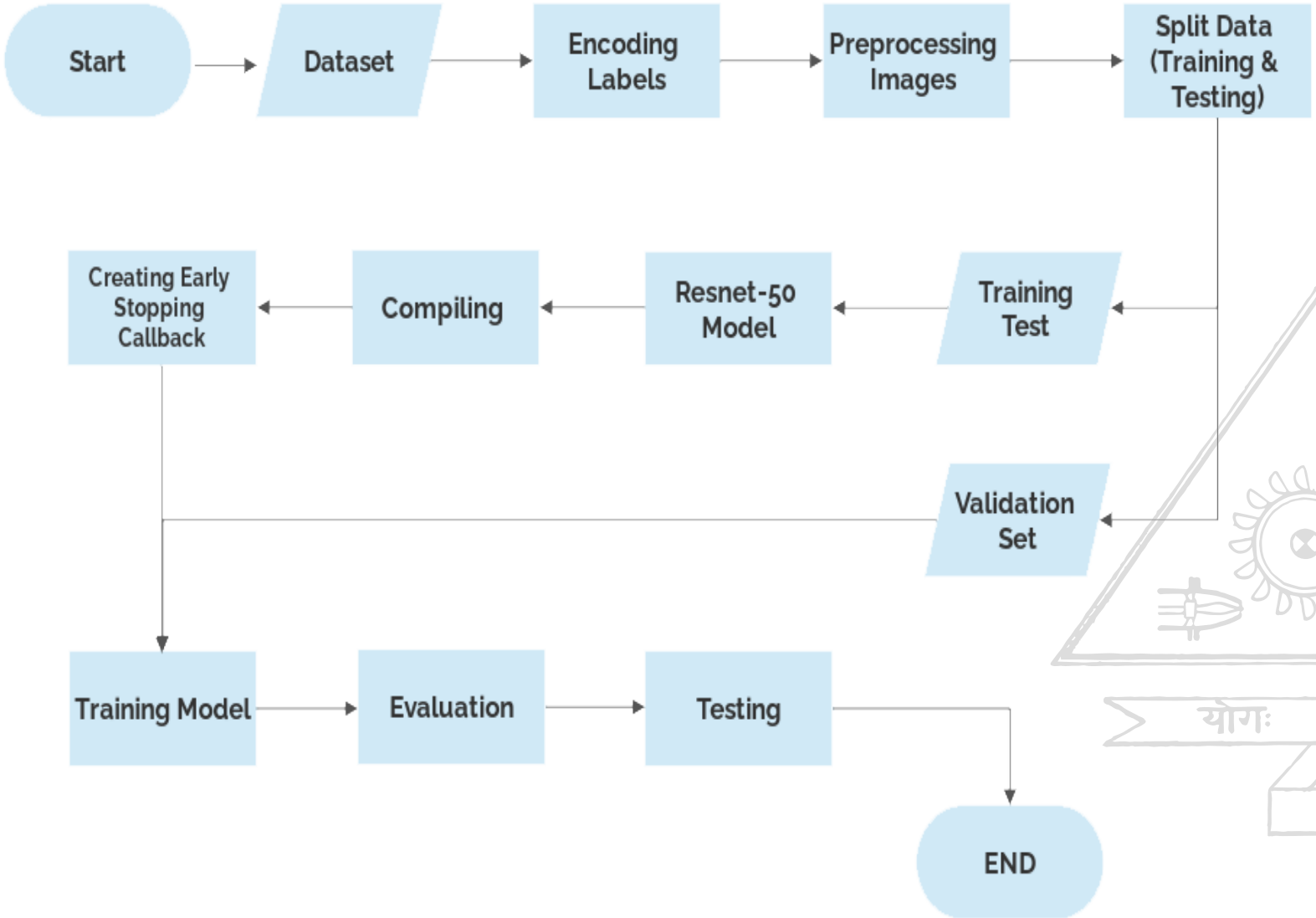
DFD Level 1


ResNet-50
Model
Comparing With
Model
Identification
of result
User
Defected
Leaf Image
Disease
Detetction
Platform
Identified
disease
Remedial
Recommendation
Recommondation
Database

System Architecture



Block Diagram of Model Creation



Input image

The leaf image of paddy and maize is to be taken from Kaggle dataset. The dataset selected for Paddy contains,

- 523 Brown Spot
- 1488 Healthy
- 565 Hispa
- 779 Leaf Blast images

and maize dataset contains

- 1146 blight
- 1306 common rust
- 574 Grey leaf spot
- 1162 Healthy images

For this project we are taking **523 images** from every datasets.

Input image

The leaf image of paddy and maize is to be taken from Kaggle dataset. The dataset selected for Paddy contains,

- 523 Brown Spot
- 1488 Healthy
- 565 Hispa
- 779 Leaf Blast images

and maize dataset contains

- 1146 blight
- 1306 common rust
- 574 Grey leaf spot
- 1162 Healthy images

For this project we are taking **523 images** from every datasets.

Dataset Collection and Pre-processing

The dataset is collected from Kaggle and imported to colab.
Steps involved in data pre-processing are,

- Encoding
- Turn into tensor set
- reshaping and normalising

```
# read, turn image into number, normalize, resize
def preprocess_image(image_path, labels=None):
    # read image
    image = tf.io.read_file(image_path)
    # turn jpeg into numbers
    image = tf.image.decode_jpeg(image, channels=3)
    # scaling / normalize (0,255) becomes (0,1)
    image = tf.image.convert_image_dtype(image, dtype=tf.float32)
    # resize to (224,224)
    image = tf.image.resize(image, size=[IMAGE_SIZE, IMAGE_SIZE])
    # return
    return image, labels
```

Model Building

Model for the project is being built using ResNet-50 model

```
model = tf.keras.Sequential([
                             # transfer learning model
                             hub.KerasLayer("https://tfhub.dev/tensorflow/resnet_50/feature_vector/1"),

                             # output layer
                             tf.keras.layers.Dense(units=num_unique_label, activation='softmax')
])
```

After creating the model it is complied and a early stopping callback is created before training the model in order to prevent overfitting.

```
# EARLYSTOPPING CALLBACK
# monitor the val loss (prevent overfitting)

early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3)
```

Model Building

Model is then trained with 50 epochs

```
history_train = model.fit(train_set, epochs=50,
                          validation_data = val_set,
                          callbacks=[early_stopping]])
```

The model is saved fro further processing.

Labelled Images

Here are some images that are labelled for training

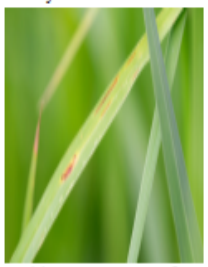
Testing

```
# upload files
uploaded=files.upload()

filename = []
test_images = []

for fn in uploaded.keys():
    filename.append(fn)
    path='/content/' + fn
    test_images.append(path)

# turn into set
test_set = tf.data.Dataset.from_tensor_slices(( tf.constant(test_images) ))

# preprocess
test_set = test_set.map(preprocess_image)

# batching
test_set = test_set.batch(batch_size=32)

# predict
test_predictions = model.predict(test_set)

label_prediction = []
for i in range(len(test_predictions)):
    label_prediction.append(unique_label[np.argmax(test_predictions[i])])

# show prediction results
for i in range(len(test_images)):
    print(label_prediction[i])
    pil_img = Image(filename=test_images[i], width=150, height=150)
    display(pil_img)
```

```
Choose Files 3 files
```

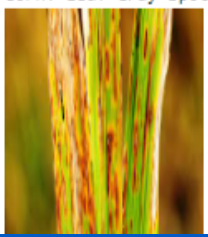
- **Brown_leaf_spot_of_rice-min.jpg**(image/jpeg) - 298948 bytes, last modified: 9/18/2022 - 100% done
- **brown-spot-1.jpg**(image/jpeg) - 86292 bytes, last modified: 9/18/2022 - 100% done
- **download (1).jpg**(image/jpeg) - 9944 bytes, last modified: 10/12/2022 - 100% done

```
Saving Brown_leaf_spot_of_rice-min.jpg to Brown_leaf_spot_of_rice-min.jpg
Saving brown-spot-1.jpg to brown-spot-1.jpg
Saving download (1).jpg to download (1).jpg
1/1 [==============================] - 2s 2s/step
Paddy: Leaf Blast
```

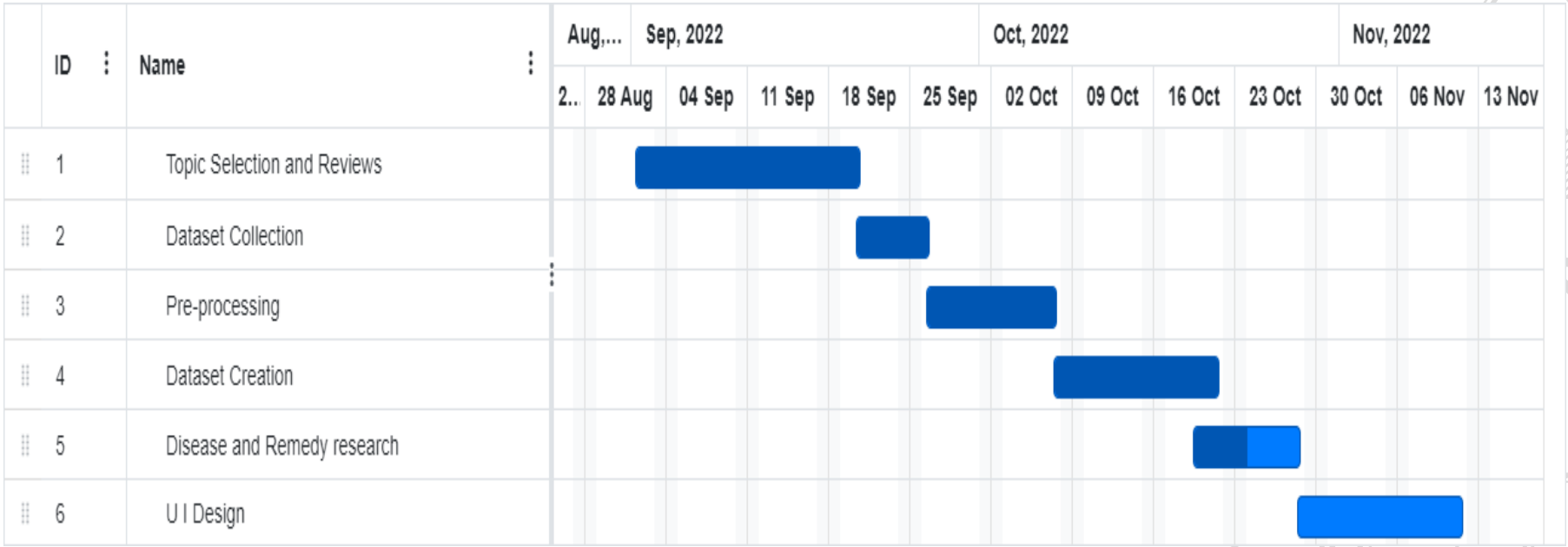


```
Paddy: Brown Leaf Spots
```

```
Corn: Leaf Gray Spots
```



Action Plan


ID
Name
Aug,...
Sep, 2022
Oct, 2022
Nov, 2022
2..
28 Aug
04 Sep
11 Sep
18 Sep
25 Sep
02 Oct
09 Oct
16 Oct
23 Oct
30 Oct
06 Nov
13 Nov
1 Topic Selection and Reviews
2 Dataset Collection
3 Pre-processing
4 Dataset Creation
5 Disease and Remedy research
6 U I Design

Thank you!