In [35]:
```
from tensorflow.keras.models import load_model
```

In [36]:
```
model = load_model('../../Downloads/mycrop.h5')
```

In [37]:
```
import cv2
```

In [38]:
```
crop_class_labels = ['jute', 'maize', 'wheat', 'sugarcane', 'rice']
```

In [39]:
```
import os
```

In [42]:
```
import cv2
import os
import numpy as np
from tensorflow.keras.models import load_model

def get_crop_prediction(filename, model, class_labels):
    image_path = os.path.join('../../dataset/Crop Prediction/'+filename)
    image = cv2.imread(image_path)

    if image is None:
        print(f"Error reading the image: {filename}")
        return None

    image = cv2.resize(image, (224, 224))
    image = image / 255.0
    image = image.reshape(1, 224, 224, 3)
    predictions = model.predict_on_batch(image)
    predicted_class_index = np.argmax(predictions)
    predicted_crop = class_labels[predicted_class_index]

    return predicted_crop
```

In [43]:
```
model = load_model('../../Downloads/mycrop.h5')
crop_class_labels = ['jute', 'maize', 'wheat', 'sugarcane', 'rice']
filenames = os.listdir('../../dataset/Crop Prediction/')
for file in filenames:
    prediction = get_crop_prediction(file, model, crop_class_labels)
    if prediction is not None:
        print(f"{file}: Predicted Crop - {prediction}")
```

```
Error reading the image: .ipynb_checkpoints
jute1.jpg: Predicted Crop - jute
jute2.jpg: Predicted Crop - jute
maize1.jpg: Predicted Crop - wheat
maize2.jpg: Predicted Crop - wheat
rice1.jpg: Predicted Crop - wheat
rice2.jpg: Predicted Crop - wheat
sugarcane1.jpg: Predicted Crop - wheat
sugarcane2.jpg: Predicted Crop - sugarcane
wheat1.jpg: Predicted Crop - wheat
wheat2.jpg: Predicted Crop - sugarcane
```

In [ ]:
```

```