```
# Import necessary libraries
import os
import cv2
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
data_dir = '/content/drive/MyDrive/Dataset_ML'
img_height, img_width = 64, 64
```

```
# Prepare image data and labels
images = []
labels = []
```

```
# Iterate through each folder in the dataset directory
for label in os.listdir(data_dir):
    shot_dir = os.path.join(data_dir, label)
    if os.path.isdir(shot_dir):
        for img_name in os.listdir(shot_dir):
            img_path = os.path.join(shot_dir, img_name)
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)  # Read in grayscale
            img = cv2.resize(img, (img_height, img_width))   # Resize image
            img = img / 255.0  # Normalize the pixel values
            images.append(img)
            labels.append(label)

# Convert images to numpy array and labels to numpy array
images = np.array(images)
labels = np.array(labels)
```

```
# Reshape images for Random Forest (flatten each image to 1D array)
X = images.reshape(images.shape[0], -1)
y = labels

# Encode labels into integers
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Train the Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

```
▾     RandomForestClassifier      ⓘ ?
RandomForestClassifier(random_state=42)
```

```
# Predict on the test set
y_pred = rf_model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Random Forest Accuracy: {accuracy}")
```

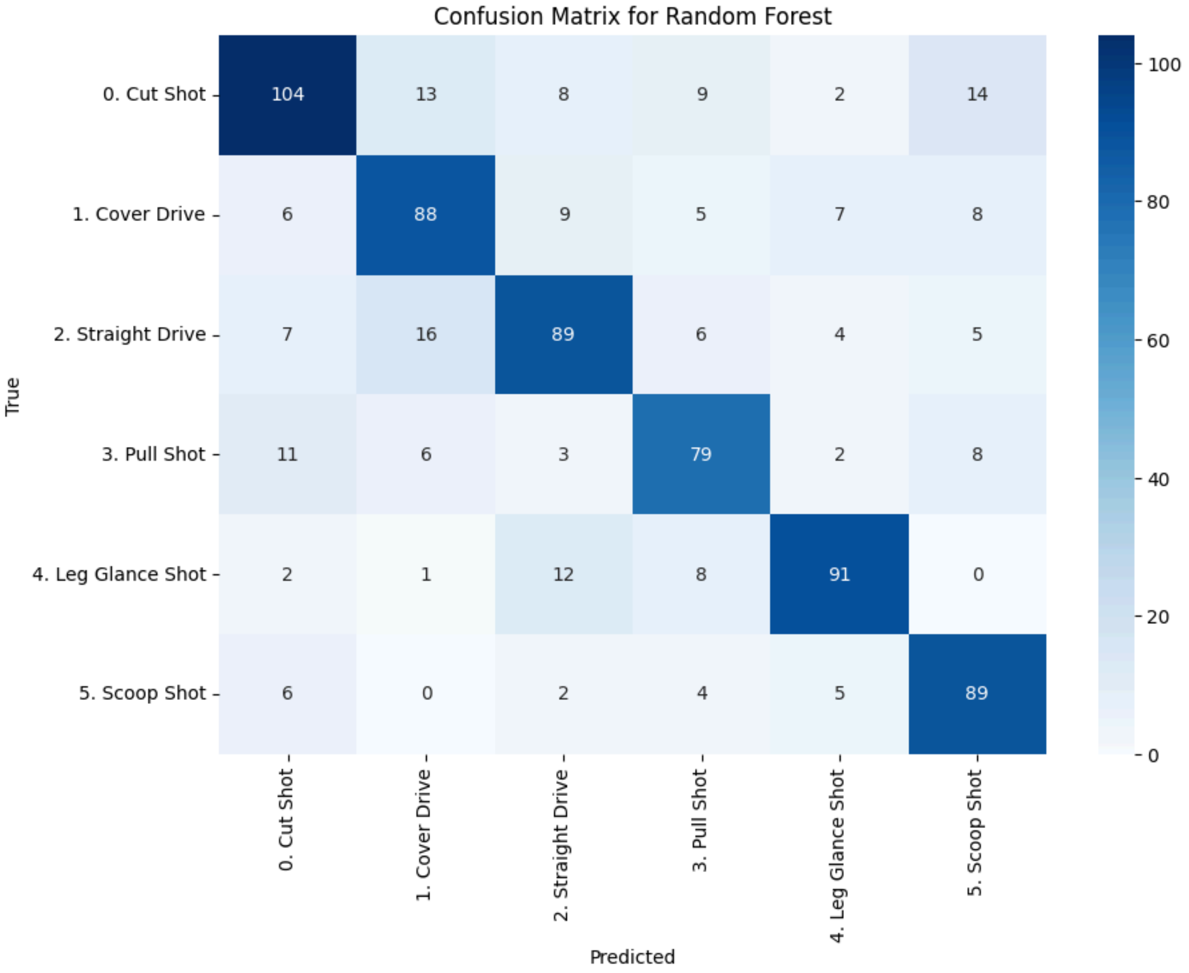
```
Random Forest Accuracy: 0.7407407407407407
```

```
# Confusion matrix
cm = confusion_matrix(y_test, y_pred)



# Visualize confusion matrix
plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=label_encoder.classes_, yticklabe
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for Random Forest')
plt.show()

# Save the model
import joblib
joblib.dump(rf_model, 'cricket_shot_rf_model.pkl')
```



	0. Cut Shot	1. Cover Drive	2. Straight Drive	3. Pull Shot	4. Leg Glance Shot	5. Scoop Shot
0. Cut Shot	104	13	8	9	2	14
1. Cover Drive	6	88	9	5	7	8
2. Straight Drive	7	16	89	6	4	5
3. Pull Shot	11	6	3	79	2	8
4. Leg Glance Shot	2	1	12	8	91	0
5. Scoop Shot	6	0	2	4	5	89

```
['cricket_shot_rf_model.pkl']
```

Start coding or generate with AI.