

```
from google.colab import files
files.upload()

Choose files | kaggle.json
• kaggle.json(application/json) - 63 bytes, last modified: 19/03/2024 - 100% done
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"motasim","key":"edb0026b64d86ab44468bd4617859e5d"}'}

!mkdir -p ~/.kaggle

!cp kaggle.json ~/.kaggle/

! kaggle datasets download -d rifakhan22/suspicious

Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Downloading suspicious.zip to /content
 97% 82.0M/84.4M [00:01<00:00, 79.0MB/s]
100% 84.4M/84.4M [00:01<00:00, 61.9MB/s]

import zipfile
zip_ref = zipfile.ZipFile('/content/suspicious.zip', 'r')
zip_ref.extractall('/content')
zip_ref.close()

import tensorflow as tf
from tensorflow import keras
from keras import Sequential
from keras.layers import Dense,Conv2D,MaxPooling2D,Flatten,BatchNormalization,Dropout

import cv2
import os
import random
import shutil
import numpy as np

from sklearn.model_selection import train_test_split

def extract_frames_from_video(video_path, train_output_path, valid_output_path, train_ratio=0.7, frame_size=(256, 256)):
    # Create output directories if they don't exist
    if not os.path.exists(train_output_path):
        os.makedirs(train_output_path)
    if not os.path.exists(valid_output_path):
        os.makedirs(valid_output_path)

    # Open the video file
    cap = cv2.VideoCapture(video_path)
    frame_count = 0

    # Read until video is completed
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        # Resize frame
        frame = cv2.resize(frame, frame_size)
        # Decide whether to save the frame in train or valid directory based on the ratio
        if random.random() < train_ratio:
            output_dir = train_output_path
        else:
            output_dir = valid_output_path
        # Save frame
        cv2.imwrite(os.path.join(output_dir, f"frame_{frame_count}.jpg"), frame)
        frame_count += 1

    # Release the video capture object
    cap.release()

print(f"Frames extracted from {video_path}: {frame_count}")
```

```
def extract_frames_from_videos_in_folder(folder_path, train_output_folder, valid_output_folder, train_ratio=0.7, frame_size=(256, 256)):  
    # List all files in the folder  
    files = os.listdir(folder_path)  
  
    # Iterate over files  
    for file in files:  
        if file.endswith(".mp4"):  
            video_path = os.path.join(folder_path, file)  
            train_output_path = os.path.join(train_output_folder, os.path.splitext(file)[0])  
            valid_output_path = os.path.join(valid_output_folder, os.path.splitext(file)[0])  
            # Extract frames from video  
            extract_frames_from_video(video_path, train_output_path, valid_output_path, train_ratio, frame_size)  
  
# Path to the folder containing video files with class 'NO'  
videos_folder = "/content/Suspicious Activity Detection/no/"  
train_output_folder = "/content/Extracted Frames From Video Dataset/train/no/"  
valid_output_folder = "/content/Extracted Frames From Video Dataset/valid/no/"  
  
# Extract frames from videos in the folder with 70% frames saved in train directory and 30% frames saved in valid directory  
extract_frames_from_videos_in_folder(videos_folder, train_output_folder, valid_output_folder, train_ratio=0.7)  
  
Frames extracted from /content/Suspicious Activity Detection/no/nofi142.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi018.mp4: 69  
Frames extracted from /content/Suspicious Activity Detection/no/nofi083.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi087.mp4: 36  
Frames extracted from /content/Suspicious Activity Detection/no/nofi057.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi061.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi137.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi048.mp4: 50  
Frames extracted from /content/Suspicious Activity Detection/no/nofi003.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi039.mp4: 63  
Frames extracted from /content/Suspicious Activity Detection/no/nofi145.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi119.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi024.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi022.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi127.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi080.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi067.mp4: 40  
Frames extracted from /content/Suspicious Activity Detection/no/nofi126.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi144.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi013.mp4: 50  
Frames extracted from /content/Suspicious Activity Detection/no/nofi141.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi117.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi066.mp4: 40  
Frames extracted from /content/Suspicious Activity Detection/no/nofi096.mp4: 50  
Frames extracted from /content/Suspicious Activity Detection/no/nofi020.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi036.mp4: 50  
Frames extracted from /content/Suspicious Activity Detection/no/nofi124.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi131.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi050.mp4: 50  
Frames extracted from /content/Suspicious Activity Detection/no/nofi031.mp4: 48  
Frames extracted from /content/Suspicious Activity Detection/no/nofi002.mp4: 85  
Frames extracted from /content/Suspicious Activity Detection/no/nofi101.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi065.mp4: 40  
Frames extracted from /content/Suspicious Activity Detection/no/nofi069.mp4: 40  
Frames extracted from /content/Suspicious Activity Detection/no/nofi121.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi097.mp4: 50  
Frames extracted from /content/Suspicious Activity Detection/no/nofi136.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi128.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi095.mp4: 69  
Frames extracted from /content/Suspicious Activity Detection/no/nofi133.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi113.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi146.mp4: 60  
Frames extracted from /content/Suspicious Activity Detection/no/nofi012.mp4: 50
```

```
frames extracted from /content/Suspicious Activity Detection/no/not1112.mp4: 60

# Path to the folder containing video files with class 'YES'
videos_folder = "/content/Suspicious Activity Detection/yes/"
train_output_folder = "/content/Frames Extracted From Video Dataset/train/yes/"
valid_output_folder = "/content/Frames Extracted From Video Dataset/valid/yes/"

# Extract frames from videos in the folder with 70% frames saved in train directory and 30% frames saved in valid directory
extract_frames_from_videos_in_folder(videos_folder, train_output_folder, valid_output_folder, train_ratio=0.7)

# import the libraries as shown below
import tensorflow as tf
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt

train = ImageDataGenerator(rescale= 1/255)
validation = ImageDataGenerator(rescale= 1/255)

train_ds = train.flow_from_directory('/content/Frames Extracted From Video Dataset/train',
                                    target_size= (224, 224),
                                    batch_size= 32,
                                    class_mode= 'categorical')

validation_ds = train.flow_from_directory('/content/Frames Extracted From Video Dataset/valid',
                                         target_size= (224, 224),
                                         batch_size= 32,
                                         class_mode= 'categorical')

#Train using VGG16 Model
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras import layers, models, optimizers

# Load the pre-trained VGG16 model
vgg_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Freeze the layers in the VGG16 model
for layer in vgg_model.layers:
    layer.trainable = False

# Add custom layers on top of VGG16
model = models.Sequential()
model.add(vgg_model)
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
# For ignoring 50% of neuron during training.
model.add(layers.Dropout(0.5))
model.add(layers.Dense(2, activation='softmax'))

model.summary()

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

```

# Train the model
history = model.fit(train_ds,
                     steps_per_epoch=train_ds.samples // train_ds.batch_size,
                     epochs = 15,
                     validation_data=validation_ds,
                     validation_steps=validation_ds.samples // validation_ds.batch_size)

Epoch 1/15
373/373 [=====] - 81s 190ms/step - loss: 0.4447 - accuracy: 0.8876 - val_loss: 0.0445 - val_accuracy: 0.9876
Epoch 2/15
373/373 [=====] - 68s 182ms/step - loss: 0.0569 - accuracy: 0.9801 - val_loss: 0.0124 - val_accuracy: 0.9994
Epoch 3/15
373/373 [=====] - 69s 185ms/step - loss: 0.0303 - accuracy: 0.9904 - val_loss: 0.0039 - val_accuracy: 0.9996
Epoch 4/15
373/373 [=====] - 68s 183ms/step - loss: 0.0528 - accuracy: 0.9804 - val_loss: 0.0175 - val_accuracy: 0.9965
Epoch 5/15
373/373 [=====] - 67s 181ms/step - loss: 0.0409 - accuracy: 0.9841 - val_loss: 0.0033 - val_accuracy: 0.9994
Epoch 6/15
373/373 [=====] - 68s 183ms/step - loss: 0.0603 - accuracy: 0.9763 - val_loss: 0.0156 - val_accuracy: 0.9967
Epoch 7/15
373/373 [=====] - 68s 181ms/step - loss: 0.0280 - accuracy: 0.9898 - val_loss: 0.0066 - val_accuracy: 0.9986
Epoch 8/15
373/373 [=====] - 68s 183ms/step - loss: 0.0343 - accuracy: 0.9849 - val_loss: 9.8359e-04 - val_accuracy: 1.00
Epoch 9/15
373/373 [=====] - 68s 182ms/step - loss: 0.0442 - accuracy: 0.9820 - val_loss: 0.0047 - val_accuracy: 0.9992
Epoch 10/15
373/373 [=====] - 68s 183ms/step - loss: 0.0456 - accuracy: 0.9773 - val_loss: 0.0033 - val_accuracy: 0.9990
Epoch 11/15
373/373 [=====] - 68s 183ms/step - loss: 0.0533 - accuracy: 0.9746 - val_loss: 0.0053 - val_accuracy: 0.9990
Epoch 12/15
373/373 [=====] - 68s 183ms/step - loss: 0.0614 - accuracy: 0.9672 - val_loss: 0.0047 - val_accuracy: 0.9994
Epoch 13/15
373/373 [=====] - 68s 184ms/step - loss: 0.0414 - accuracy: 0.9768 - val_loss: 0.0023 - val_accuracy: 0.9996
Epoch 14/15
373/373 [=====] - 69s 185ms/step - loss: 0.0350 - accuracy: 0.9839 - val_loss: 0.0041 - val_accuracy: 0.9994
Epoch 15/15
373/373 [=====] - 68s 183ms/step - loss: 0.0444 - accuracy: 0.9802 - val_loss: 0.0020 - val_accuracy: 0.9998

```

Evaluate the model

```
loss, accuracy = model.evaluate(validation_ds, steps=validation_ds.samples // validation_ds.batch_size)
```

```
161/161 [=====] - 21s 127ms/step - loss: 0.0020 - accuracy: 0.9998
```

#Print Validation Accuracy and Loss percentage

```
print("Validation Loss: {:.2f}%".format(loss * 100))
print("Validation Loss:", loss)
print("Validation Accuracy: {:.2f}%".format(accuracy * 100))
print("Validation Accuracy:", accuracy)
```

```
Validation Loss: 0.20%
Validation Loss: 0.0019549685530364513
Validation Accuracy: 99.98%
Validation Accuracy: 0.9998059272766113
```

Evaluate the model

```
evaluation = model.evaluate(validation_ds)
print("Validation Loss:", evaluation[0])
print("Validation Accuracy:", evaluation[1])
```

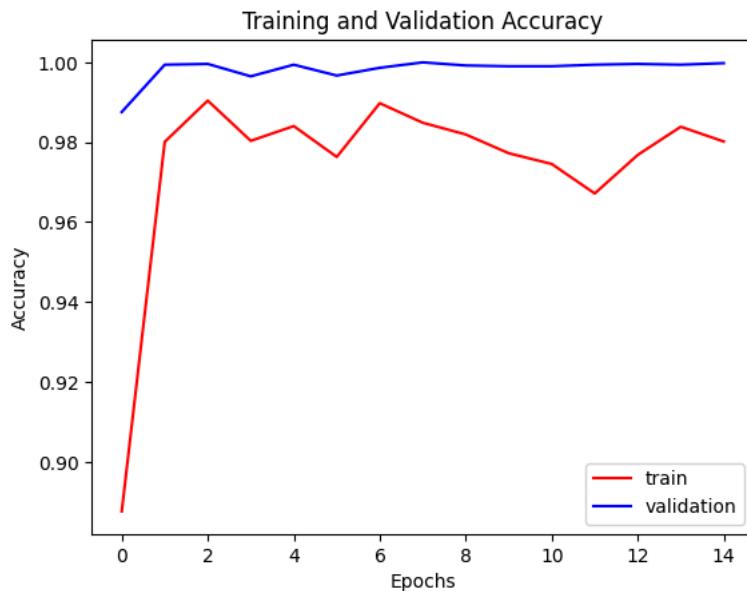
```
162/162 [=====] - 25s 156ms/step - loss: 0.0019 - accuracy: 0.9998
```

```
Validation Loss: 0.0019463562639430165
```

```
Validation Accuracy: 0.9998067617416382
```

#Plot Graph for Training and Validation Accuracy

```
plt.plot(history.history['accuracy'], color='red', label='train')
plt.plot(history.history['val_accuracy'], color='blue', label='validation')
plt.legend()
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.show()
```



```
#Validation Evaluation Graph in forms of Bar and Metric
valid = ImageDataGenerator(rescale= 1/255)

valid_ds = train.flow_from_directory('/content/Frames Extracted From Video Dataset/valid',
                                    target_size= (224, 224),
                                    batch_size= 32,
                                    class_mode= 'categorical')

loss, accuracy = model.evaluate(valid_ds)

print("Validation Loss:", loss)
print("Validation Accuracy:", accuracy)
plt.bar(['Validation Accuracy', 'Validation Loss'], [accuracy, loss], color=['blue', 'red'])
plt.title('Validation Evaluation')
plt.ylabel('Value')
plt.show()

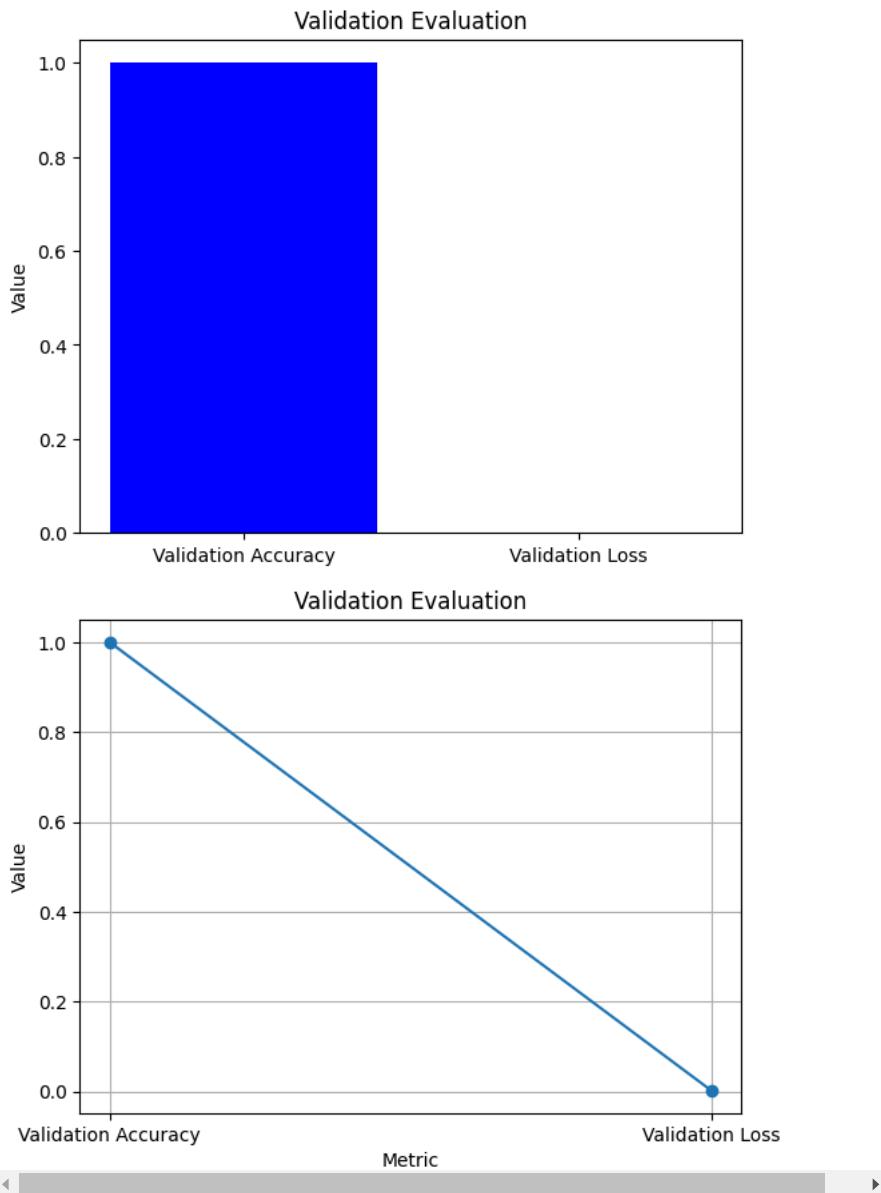
metrics = ['Validation Accuracy', 'Validation Loss']
values = [accuracy, loss]

plt.plot(metrics, values, marker='o')
plt.title('Validation Evaluation')
plt.xlabel('Metric')
plt.ylabel('Value')
plt.grid(True)
plt.show()
```

```

Found 5175 images belonging to 2 classes.
162/162 [=====] - 20s 125ms/step - loss: 0.0019 - accuracy: 0.
Validation Loss: 0.0019463554490357637
Validation Accuracy: 0.9998067617416382

```



```

# Function to preprocess each frame
def preprocess_frame(frame):
    # Resize the frame to match the model input size
    resized_frame = cv2.resize(frame, (224, 224))
    # Normalize the frame
    normalized_frame = resized_frame / 255.0
    return normalized_frame

# Load the video from user input
video_path = input("Enter the path to the video: ")
cap = cv2.VideoCapture(video_path)

Enter the path to the video: /content/V_sample_02.mp4

# Check if the video is loaded successfully
if not cap.isOpened():
    print("Error: Unable to load the video.")
else:
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

    # Preprocess the frame

```

```
...  
preprocessed_frame = preprocess_frame(frame)  
  
# Reshape the preprocessed frame for model input  
test_input = np.expand_dims(preprocessed_frame, axis=0)  
  
# Get prediction probabilities  
predictions = model.predict(test_input)  
print("Predictions Probabilities:\n", predictions)  
  
# Display the frame with predictions (optional)  
plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))  
plt.axis('off')  
plt.title("Predictions Probabilities: {}".format(predictions))  
plt.show()  
  
# Press 'q' to exit  
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break  
  
# Release everything when done  
cap.release()  
cv2.destroyAllWindows()
```

1/1 [=====] - 1s 1s/step

Predictions Probabilities:

[[0.99039525 0.00960473]]

Predictions Probabilities: [[0.99039525 0.00960473]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.991746 0.008254]]

Predictions Probabilities: [[0.991746 0.008254]]



1/1 [=====] - 0s 24ms/step

Predictions Probabilities:

[[0.98595756 0.01404238]]

Predictions Probabilities: [[0.98595756 0.01404238]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.9751024 0.02489755]]

Predictions Probabilities: [[0.9751024 0.02489755]]



1/1 [=====] - 0s 23ms/step

Predictions Probabilities:

[[0.9669622 0.03303777]]

Predictions Probabilities: [[0.9669622 0.03303777]]



1/1 [=====] - 0s 31ms/step

Predictions Probabilities:

[[0.9783945 0.02160549]]

Predictions Probabilities: [[0.9783945 0.02160549]]



1/1 [=====] - 0s 31ms/step

Predictions Probabilities:

[[0.9763822 0.02361784]]

Predictions Probabilities: [[0.9763822 0.02361784]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.9839451 0.01605493]]

Predictions Probabilities: [[0.9839451 0.01605493]]





1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.967374 0.03262602]]

Predictions Probabilities: [[0.967374 0.03262602]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[0.90920705 0.09079298]]

Predictions Probabilities: [[0.90920705 0.09079298]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[0.91043353 0.08956651]]

Predictions Probabilities: [[0.91043353 0.08956651]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.9159364 0.08406359]]

Predictions Probabilities: [[0.9159364 0.08406359]]





1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[0.92415476 0.07584526]]

Predictions Probabilities: [[0.92415476 0.07584526]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.95769453 0.04230555]]

Predictions Probabilities: [[0.95769453 0.04230555]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.88571745 0.11428255]]

Predictions Probabilities: [[0.88571745 0.11428255]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.5816822 0.41831774]]

Predictions Probabilities: [[0.5816822 0.41831774]]





1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.792186 0.20781396]]

Predictions Probabilities: [[0.792186 0.20781396]]



1/1 [=====] - 0s 35ms/step

Predictions Probabilities:

[[0.72875655 0.27124348]]

Predictions Probabilities: [[0.72875655 0.27124348]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[0.85325634 0.14674367]]

Predictions Probabilities: [[0.85325634 0.14674367]]



1/1 [=====] - 0s 40ms/step

Predictions Probabilities:

[[0.74707663 0.25292334]]

Predictions Probabilities: [[0.74707663 0.25292334]]





1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[0.90103513 0.09896486]]

Predictions Probabilities: [[0.90103513 0.09896486]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.84651935 0.15348063]]

Predictions Probabilities: [[0.84651935 0.15348063]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[0.98253644 0.01746353]]

Predictions Probabilities: [[0.98253644 0.01746353]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.99428093 0.00571907]]

Predictions Probabilities: [[0.99428093 0.00571907]]





1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.9877067 0.01229327]]

Predictions Probabilities: [[0.9877067 0.01229327]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.9849246 0.01507541]]

Predictions Probabilities: [[0.9849246 0.01507541]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.9762315 0.02376851]]

Predictions Probabilities: [[0.9762315 0.02376851]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[0.99563855 0.0043615]]

Predictions Probabilities: [[0.99563855 0.0043615]]





1/1 [=====] - 0s 26ms/step

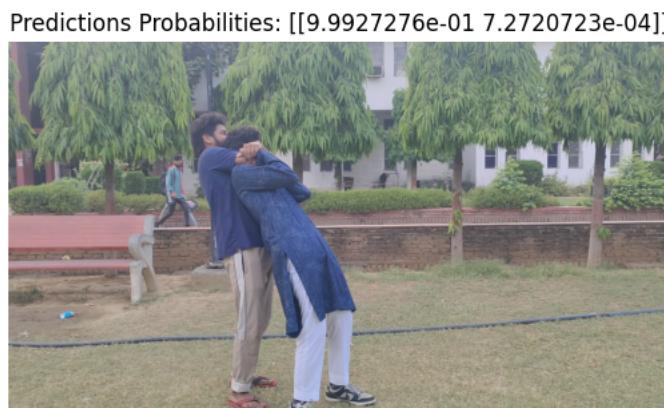
Predictions Probabilities:
[[0.99383706 0.00616299]]

Predictions Probabilities: [[0.99383706 0.00616299]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:
[[9.9927276e-01 7.2720723e-04]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:
[[0.9981785 0.00182156]]

Predictions Probabilities: [[0.9981785 0.00182156]]



1/1 [=====] - 0s 34ms/step

Predictions Probabilities:
[[0.99887365 0.0011263]]

Predictions Probabilities: [[0.99887365 0.0011263]]





1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.996759 0.00324096]]

Predictions Probabilities: [[0.996759 0.00324096]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[0.9983754 0.00162461]]

Predictions Probabilities: [[0.9983754 0.00162461]]



1/1 [=====] - 0s 39ms/step

Predictions Probabilities:

[[9.994599e-01 5.400916e-04]]

Predictions Probabilities: [[9.994599e-01 5.400916e-04]]



1/1 [=====] - 0s 35ms/step

Predictions Probabilities:

[[9.9975723e-01 2.4278222e-04]]

Predictions Probabilities: [[9.9975723e-01 2.4278222e-04]]





1/1 [=====] - 0s 45ms/step

Predictions Probabilities:

[[0.99837375 0.0016262]]

Predictions Probabilities: [[0.99837375 0.0016262]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[0.9987551 0.00124489]]

Predictions Probabilities: [[0.9987551 0.00124489]]



1/1 [=====] - 0s 37ms/step

Predictions Probabilities:

[[0.9988488 0.00115113]]

Predictions Probabilities: [[0.9988488 0.00115113]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.9958469 0.00415306]]

Predictions Probabilities: [[0.9958469 0.00415306]]





1/1 [=====] - 0s 29ms/step

Predictions Probabilities:

[[0.99394256 0.00605741]]

Predictions Probabilities: [[0.99394256 0.00605741]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.99694747 0.00305254]]

Predictions Probabilities: [[0.99694747 0.00305254]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.9987804 0.00121954]]

Predictions Probabilities: [[0.9987804 0.00121954]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9921989e-01 7.8006083e-04]]

Predictions Probabilities: [[9.9921989e-01 7.8006083e-04]]





1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.996455e-01 3.545122e-04]]

Predictions Probabilities: [[9.996455e-01 3.545122e-04]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9978095e-01 2.1907871e-04]]

Predictions Probabilities: [[9.9978095e-01 2.1907871e-04]]



1/1 [=====] - 0s 38ms/step

Predictions Probabilities:

[[9.9977475e-01 2.2529840e-04]]

Predictions Probabilities: [[9.9977475e-01 2.2529840e-04]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9966049e-01 3.3949106e-04]]

Predictions Probabilities: [[9.9966049e-01 3.3949106e-04]]





1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9987769e-01 1.2227995e-04]]

Predictions Probabilities: [[9.9987769e-01 1.2227995e-04]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9994993e-01 5.0032952e-05]]

Predictions Probabilities: [[9.9994993e-01 5.0032952e-05]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9998367e-01 1.6277179e-05]]

Predictions Probabilities: [[9.9998367e-01 1.6277179e-05]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9996281e-01 3.7224032e-05]]

Predictions Probabilities: [[9.9996281e-01 3.7224032e-05]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

$[9.9986756e-01 \ 1.3241319e-04]$

Predictions Probabilities: $[9.9986756e-01 \ 1.3241319e-04]$



1/1 [=====] - 0s 39ms/step

Predictions Probabilities:

$[9.999082e-01 \ 9.183586e-05]$

Predictions Probabilities: $[9.999082e-01 \ 9.183586e-05]$



1/1 [=====] - 0s 42ms/step

Predictions Probabilities:

$[9.9988544e-01 \ 1.1459151e-04]$

Predictions Probabilities: $[9.9988544e-01 \ 1.1459151e-04]$



1/1 [=====] - 0s 36ms/step

Predictions Probabilities:

$[9.9984705e-01 \ 1.5292445e-04]$

Predictions Probabilities: $[9.9984705e-01 \ 1.5292445e-04]$

Predictions Probabilities: [[9.9967670e-01 3.2335386e-04]]



1/1 [=====] - 0s 55ms/step

Predictions Probabilities:

[[9.9967670e-01 3.2335386e-04]]

Predictions Probabilities: [[9.9967670e-01 3.2335386e-04]]



1/1 [=====] - 0s 39ms/step

Predictions Probabilities:

[[0.99727684 0.00272318]]

Predictions Probabilities: [[0.99727684 0.00272318]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.99310887 0.00689116]]

Predictions Probabilities: [[0.99310887 0.00689116]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.9879726 0.01202735]]

Predictions Probabilities: [[0.9879726 0.01202735]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:
[[0.875126 0.12487397]]

Predictions Probabilities: [[0.875126 0.12487397]]



1/1 [=====] - 0s 38ms/step

Predictions Probabilities:
[[0.96452564 0.03547434]]

Predictions Probabilities: [[0.96452564 0.03547434]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:
[[0.9500675 0.04993248]]

Predictions Probabilities: [[0.9500675 0.04993248]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.99738866 0.00261141]]

Predictions Probabilities: [[0.99738866 0.00261141]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[0.99804926 0.00195078]]

Predictions Probabilities: [[0.99804926 0.00195078]]



1/1 [=====] - 0s 32ms/step

Predictions Probabilities:

[[0.969397 0.03060304]]

Predictions Probabilities: [[0.969397 0.03060304]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.9525563 0.04744369]]

Predictions Probabilities: [[0.9525563 0.04744369]]



1/1 [=====] - 0s 26ms/step

1/1 [=====] - 0s 20ms/step

Predictions Probabilities:

[[0.9931537 0.00684637]]

Predictions Probabilities: [[0.9931537 0.00684637]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[0.99509746 0.00490254]]

Predictions Probabilities: [[0.99509746 0.00490254]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.99890363 0.00109641]]

Predictions Probabilities: [[0.99890363 0.00109641]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.996809e-01 3.191178e-04]]

Predictions Probabilities: [[9.996809e-01 3.191178e-04]]



1/1 [=====] - 0s 33ms/step

Predictions Probabilities:

[[9.9993622e-01 6.3756444e-05]]

Predictions Probabilities: [[9.9993622e-01 6.3756444e-05]]



1/1 [=====] - 0s 41ms/step

Predictions Probabilities:

[[9.999225e-01 7.7108025e-06]]

Predictions Probabilities: [[9.9999225e-01 7.7108025e-06]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[9.999690e-01 3.104872e-05]]

Predictions Probabilities: [[9.999690e-01 3.104872e-05]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9999094e-01 9.0499743e-06]]

Predictions Probabilities: [[9.9999094e-01 9.0499743e-06]]

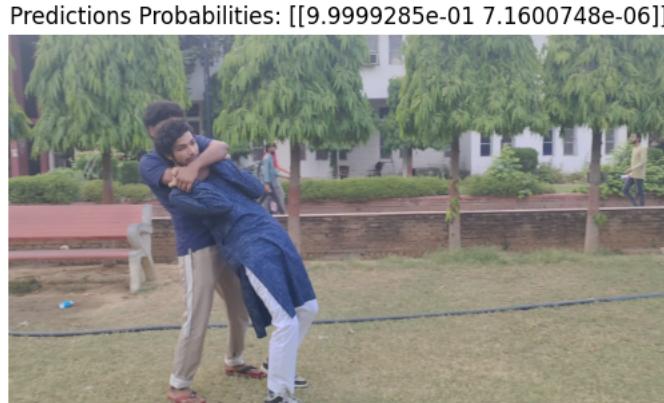




1/1 [=====] - 0s 31ms/step

Predictions Probabilities:

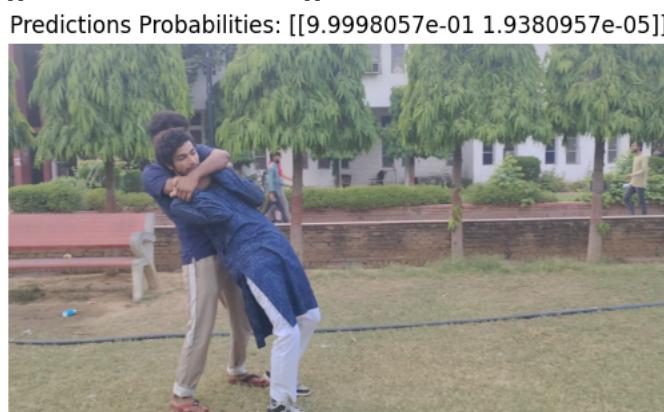
[[9.999285e-01 7.1600748e-06]]



1/1 [=====] - 0s 41ms/step

Predictions Probabilities:

[[9.9998057e-01 1.9380957e-05]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.999454e-01 5.457585e-05]]

Predictions Probabilities: [[9.999454e-01 5.457585e-05]]



1/1 [=====] - 0s 42ms/step

Predictions Probabilities:

[[9.999155e-01 8.445682e-05]]

Predictions Probabilities: [[9.999155e-01 8.445682e-05]]





1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9948311e-01 5.1688065e-04]]

Predictions Probabilities: [[9.9948311e-01 5.1688065e-04]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9981922e-01 1.8080331e-04]]

Predictions Probabilities: [[9.9981922e-01 1.8080331e-04]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[9.9951732e-01 4.8272804e-04]]

Predictions Probabilities: [[9.9951732e-01 4.8272804e-04]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9971825e-01 2.8170727e-04]]

Predictions Probabilities: [[9.9971825e-01 2.8170727e-04]]





1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.99040526 0.00959473]]

Predictions Probabilities: [[0.99040526 0.00959473]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[0.9988451 0.00115488]]

Predictions Probabilities: [[0.9988451 0.00115488]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[9.9930799e-01 6.9200894e-04]]

Predictions Probabilities: [[9.9930799e-01 6.9200894e-04]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.95375454 0.04624547]]

Predictions Probabilities: [[0.95375454 0.04624547]]





1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.9351026 0.06489744]]

Predictions Probabilities: [[0.9351026 0.06489744]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[0.96356666 0.03643337]]

Predictions Probabilities: [[0.96356666 0.03643337]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[0.9758386 0.02416145]]

Predictions Probabilities: [[0.9758386 0.02416145]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[0.9925741 0.0074259]]

Predictions Probabilities: [[0.9925741 0.0074259]]





1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9985754e-01 1.4243621e-04]]

Predictions Probabilities: [[9.9985754e-01 1.4243621e-04]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[9.9968684e-01 3.1319636e-04]]

Predictions Probabilities: [[9.9968684e-01 3.1319636e-04]]



1/1 [=====] - 0s 54ms/step

Predictions Probabilities:

[[9.9982989e-01 1.7005163e-04]]

Predictions Probabilities: [[9.9982989e-01 1.7005163e-04]]



1/1 [=====] - 0s 43ms/step

Predictions Probabilities:

[[9.9958807e-01 4.1193247e-04]]

Predictions Probabilities: [[9.9958807e-01 4.1193247e-04]]





1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

$[[9.9989223e-01 \ 1.0775261e-04]]$

Predictions Probabilities: $[[9.9989223e-01 \ 1.0775261e-04]]$



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

$[[0.9969994 \ 0.00300061]]$

Predictions Probabilities: $[[0.9969994 \ 0.00300061]]$



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

$[[9.9954337e-01 \ 4.5665991e-04]]$

Predictions Probabilities: $[[9.9954337e-01 \ 4.5665991e-04]]$



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

$[[0.9979068 \ 0.00209316]]$

Predictions Probabilities: $[[0.9979068 \ 0.00209316]]$





1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9998963e-01 1.0350384e-05]]

Predictions Probabilities: [[9.9998963e-01 1.0350384e-05]]



1/1 [=====] - 0s 30ms/step

Predictions Probabilities:

[[9.9997926e-01 2.0750846e-05]]

Predictions Probabilities: [[9.9997926e-01 2.0750846e-05]]



1/1 [=====] - 0s 44ms/step

Predictions Probabilities:

[[9.9999964e-01 3.7443016e-07]]

Predictions Probabilities: [[9.9999964e-01 3.7443016e-07]]



1/1 [=====] - 0s 29ms/step

Predictions Probabilities:

[[9.999999e-01 8.346713e-08]]

Predictions Probabilities: [[9.999999e-01 8.346713e-08]]





1/1 [=====] - 0s 41ms/step

Predictions Probabilities:

[[9.999913e-01 8.674781e-06]]

Predictions Probabilities: [[9.999913e-01 8.674781e-06]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9999976e-01 2.3414914e-07]]

Predictions Probabilities: [[9.9999976e-01 2.3414914e-07]]



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[9.9999404e-01 5.9509835e-06]]

Predictions Probabilities: [[9.9999404e-01 5.9509835e-06]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[1.000000e+00 3.4243133e-09]]

Predictions Probabilities: [[1.000000e+00 3.4243133e-09]]





1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 2.2555852e-08]]$

Predictions Probabilities: $[[1.000000e+00 \ 2.2555852e-08]]$



1/1 [=====] - 0s 40ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 3.0504437e-09]]$

Predictions Probabilities: $[[1.000000e+00 \ 3.0504437e-09]]$



1/1 [=====] - 0s 38ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 3.2984857e-09]]$

Predictions Probabilities: $[[1.000000e+00 \ 3.2984857e-09]]$



1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

$[[1.00000e+00 \ 5.05709e-10]]$

Predictions Probabilities: $[[1.00000e+00 \ 5.05709e-10]]$





1/1 [=-----] - 0s 38ms/step

Predictions Probabilities:

$[[1.00000e+00 \ 9.55146e-09]]$

Predictions Probabilities: $[[1.00000e+00 \ 9.55146e-09]]$



1/1 [=-----] - 0s 30ms/step

Predictions Probabilities:

$[[1.00000e+00 \ 1.958049e-09]]$

Predictions Probabilities: $[[1.00000e+00 \ 1.958049e-09]]$



1/1 [=-----] - 0s 27ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 4.797756e-10]]$

Predictions Probabilities: $[[1.000000e+00 \ 4.797756e-10]]$

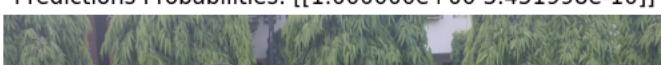


1/1 [=-----] - 0s 29ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 3.451998e-10]]$

Predictions Probabilities: $[[1.000000e+00 \ 3.451998e-10]]$

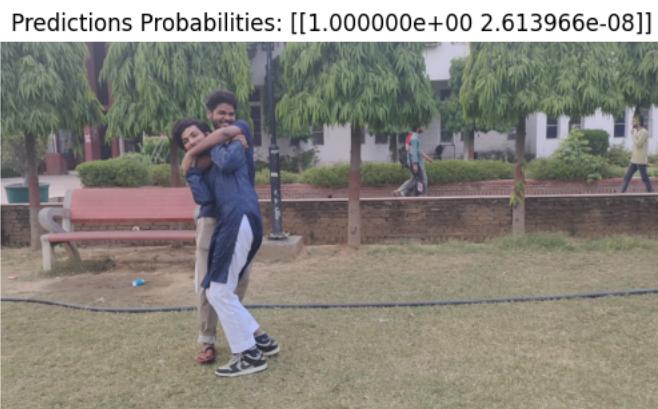




1/1 [=====] - 0s 32ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 2.613966e-08]]$



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

$[[1.000000e+00 \ 2.7412106e-08]]$

Predictions Probabilities: $[[1.000000e+00 \ 2.7412106e-08]]$



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

$[[9.999993e-01 \ 7.049668e-07]]$

Predictions Probabilities: $[[9.999993e-01 \ 7.049668e-07]]$



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

$[[9.999994e-01 \ 5.738833e-07]]$

Predictions Probabilities: $[[9.999994e-01 \ 5.738833e-07]]$





1/1 [======] - 0s 25ms/step

Predictions Probabilities:

[[9.9999654e-01 3.4135926e-06]]

Predictions Probabilities: [[9.9999654e-01 3.4135926e-06]]



1/1 [======] - 0s 25ms/step

Predictions Probabilities:

[[9.9999797e-01 2.0247842e-06]]

Predictions Probabilities: [[9.9999797e-01 2.0247842e-06]]



1/1 [======] - 0s 28ms/step

Predictions Probabilities:

[[9.999989e-01 1.097621e-06]]

Predictions Probabilities: [[9.999989e-01 1.097621e-06]]



1/1 [======] - 0s 26ms/step

Predictions Probabilities:

[[9.999927e-01 7.219074e-06]]

Predictions Probabilities: [[9.999927e-01 7.219074e-06]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:
[[9.999962e-01 3.865565e-06]]

Predictions Probabilities: [[9.999962e-01 3.865565e-06]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:
[[9.9999833e-01 1.6563113e-06]]

Predictions Probabilities: [[9.9999833e-01 1.6563113e-06]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:
[[9.9999809e-01 1.8841864e-06]]

Predictions Probabilities: [[9.9999809e-01 1.8841864e-06]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:
[[9.9999416e-01 5.8827804e-06]]

Predictions Probabilities: [[9.9999416e-01 5.8827804e-06]]



1/1 [======] - 0s 41ms/step

Predictions Probabilities:

[[9.999640e-01 3.603741e-05]]

Predictions Probabilities: [[9.999640e-01 3.603741e-05]]



1/1 [======] - 0s 56ms/step

Predictions Probabilities:

[[9.9982905e-01 1.7097253e-04]]

Predictions Probabilities: [[9.9982905e-01 1.7097253e-04]]

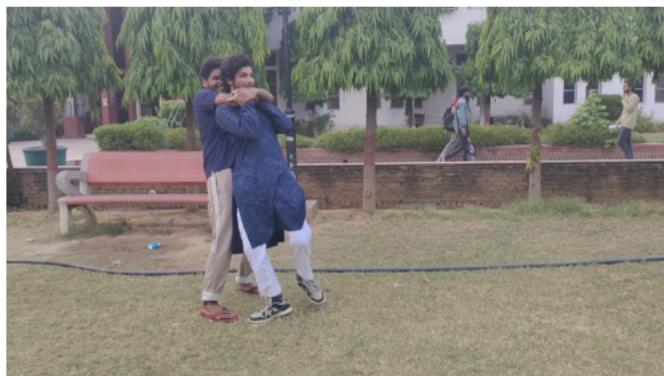


1/1 [======] - 0s 28ms/step

Predictions Probabilities:

[[9.9949360e-01 5.0642516e-04]]

Predictions Probabilities: [[9.9949360e-01 5.0642516e-04]]



1/1 [======] - 0s 39ms/step

Predictions Probabilities:

[[9.9949360e-01 5.0642516e-04]]

[[9.999510e-01 4.902062e-05]]

Predictions Probabilities: [[9.999510e-01 4.902062e-05]]

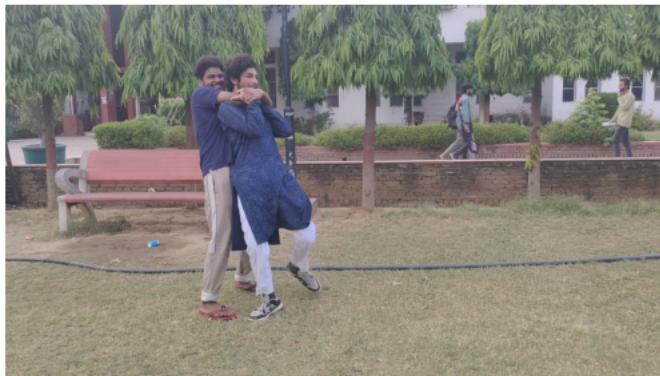


1/1 [======] - 0s 42ms/step

Predictions Probabilities:

[[9.99902e-01 9.829027e-06]]

Predictions Probabilities: [[9.99902e-01 9.829027e-06]]



1/1 [======] - 0s 25ms/step

Predictions Probabilities:

[[9.99957e-01 4.318681e-06]]

Predictions Probabilities: [[9.99957e-01 4.318681e-06]]



1/1 [======] - 0s 26ms/step

Predictions Probabilities:

[[9.99918e-01 8.210812e-06]]

Predictions Probabilities: [[9.99918e-01 8.210812e-06]]



1/1 [======] - 0s 26ms/step

Predictions Probabilities:

[[9.9999952e-01 4.6684565e-07]]

Predictions Probabilities: [[9.9999952e-01 4.6684565e-07]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.999976e-01 2.3103929e-07]]

Predictions Probabilities: [[9.999976e-01 2.3103929e-07]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.999988e-01 1.4577138e-07]]

Predictions Probabilities: [[9.999988e-01 1.4577138e-07]]

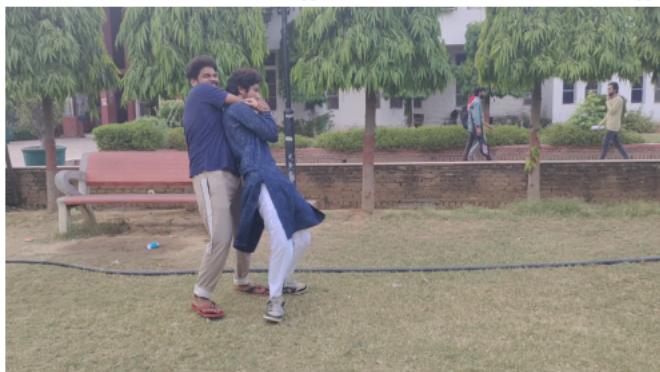


1/1 [=====] - 0s 28ms/step

Predictions Probabilities:

[[9.99994e-01 6.416055e-07]]

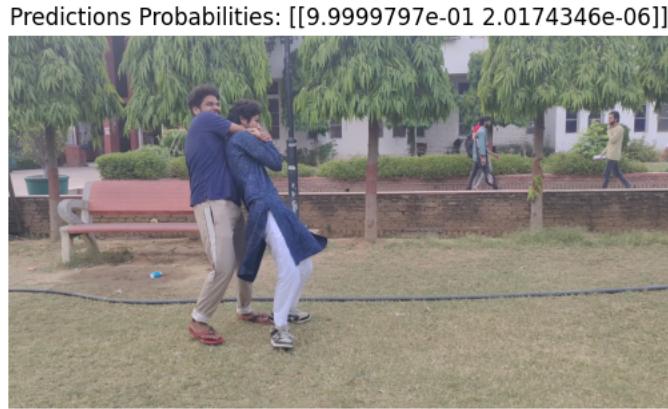
Predictions Probabilities: [[9.99994e-01 6.416055e-07]]



1/1 [=====] - 0s 27ms/step

Predictions Probabilities:

[[9.9999797e-01 2.0174346e-06]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.999981e-01 1.943096e-06]]

Predictions Probabilities: [[9.999981e-01 1.943096e-06]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9999595e-01 4.1031112e-06]]



1/1 [=====] - 0s 25ms/step

Predictions Probabilities:

[[9.9999750e-01 2.5074419e-06]]

Predictions Probabilities: [[9.9999750e-01 2.5074419e-06]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9998569e-01 1.4294811e-05]]

Predictions Probabilities: [[9.9998569e-01 1.4294811e-05]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.999752e-01 2.484266e-05]]

Predictions Probabilities: [[9.999752e-01 2.484266e-05]]



1/1 [=====] - 0s 26ms/step

Predictions Probabilities:

[[9.9979931e-01 2.0074025e-04]]

Predictions Probabilities: [[9.9979931e-01 2.0074025e-04]]



1/1 [=====] - 0s 31ms/step

Predictions Probabilities:

[[9.9991703e-01 8.2945684e-05]]

Predictions Probabilities: [[9.9991703e-01 8.2945684e-05]]





1/1 [=====] - 0s 45ms/step

Predictions Probabilities:

[[9.9983239e-01 1.6752075e-04]]

Predictions Probabilities: [[9.9983239e-01 1.6752075e-04]]

