```
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'
Dataset URL: https://www.kaggle.com/datasets/rifakhan22/suspicious
License(s): MIT
Downloading suspicious.zip to /content
 77% 65.0M/84.4M [00:00<00:00, 86.3MB/s]
100% 84.4M/84.4M [00:00<00:00, 102MB/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 numpy as np

from sklearn.model_selection import train_test_split
```

```
def extract_frames_from_video(video_path, output_path, frame_size=(256, 256)):
    # Create output directory if it doesn't exist
    if not os.path.exists(output_path):
        os.makedirs(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)
        # Save frame
        cv2.imwrite(os.path.join(output_path, 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, output_folder, 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)
            output_path = os.path.join(output_folder, os.path.splitext(file)[0])
            # Extract frames from video
            extract_frames_from_video(video_path, output_path, frame_size)


# Path to the folder containing video files with class 'NO'
videos_folder = "/content/Suspicious Activity Detection/no/"
output_folder = "/content/Frames Extracted From Video Dataset/no/"


# Extract frames from videos in the folder
extract_frames_from_videos_in_folder(videos_folder, output_folder)
```

```
Frames extracted from /content/Suspicious Activity Detection/no/nofi046.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi045.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi056.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi051.mp4: 53
Frames extracted from /content/Suspicious Activity Detection/no/nofi124.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi024.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi138.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi068.mp4: 40
Frames extracted from /content/Suspicious Activity Detection/no/nofi077.mp4: 63
Frames extracted from /content/Suspicious Activity Detection/no/nofi009.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi095.mp4: 69
Frames extracted from /content/Suspicious Activity Detection/no/nofi106.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi072.mp4: 40
Frames extracted from /content/Suspicious Activity Detection/no/nofi075.mp4: 40
Frames extracted from /content/Suspicious Activity Detection/no/nofi097.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi081.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi033.mp4: 48
Frames extracted from /content/Suspicious Activity Detection/no/nofi142.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi076.mp4: 40
Frames extracted from /content/Suspicious Activity Detection/no/nofi127.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi054.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi042.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi129.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi090.mp4: 36
Frames extracted from /content/Suspicious Activity Detection/no/nofi012.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi098.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi017.mp4: 63
Frames extracted from /content/Suspicious Activity Detection/no/nofi055.mp4: 49
Frames extracted from /content/Suspicious Activity Detection/no/nofi137.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi038.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi132.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi004.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi140.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi023.mp4: 63
Frames extracted from /content/Suspicious Activity Detection/no/nofi013.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi115.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi136.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi125.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi070.mp4: 42
Frames extracted from /content/Suspicious Activity Detection/no/nofi005.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi088.mp4: 36
Frames extracted from /content/Suspicious Activity Detection/no/nofi059.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi047.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/no/nofi057.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi085.mp4: 38
Frames extracted from /content/Suspicious Activity Detection/no/nofi120.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi044.mp4: 63
Frames extracted from /content/Suspicious Activity Detection/no/nofi087.mp4: 36
Frames extracted from /content/Suspicious Activity Detection/no/nofi002.mp4: 85
Frames extracted from /content/Suspicious Activity Detection/no/nofi091.mp4: 38
Frames extracted from /content/Suspicious Activity Detection/no/nofi040.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi022.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi043.mp4: 57
Frames extracted from /content/Suspicious Activity Detection/no/nofi008.mp4: 34
Frames extracted from /content/Suspicious Activity Detection/no/nofi150.mp4: 40
Frames extracted from /content/Suspicious Activity Detection/no/nofi086.mp4: 36
Frames extracted from /content/Suspicious Activity Detection/no/nofi139.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/no/nofi003.mp4: 60
```

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


# Extract frames from videos in the folder
extract_frames_from_videos_in_folder(videos_folder, output_folder)
```

```
Frames extracted from /content/Suspicious Activity Detection/yes/fi112.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi001.mp4: 53
Frames extracted from /content/Suspicious Activity Detection/yes/fi085.mp4: 33
Frames extracted from /content/Suspicious Activity Detection/yes/fi094.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi016.mp4: 66
Frames extracted from /content/Suspicious Activity Detection/yes/fi061.mp4: 131
Frames extracted from /content/Suspicious Activity Detection/yes/fi147.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi017.mp4: 66
Frames extracted from /content/Suspicious Activity Detection/yes/fi048.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi093.mp4: 53
Frames extracted from /content/Suspicious Activity Detection/yes/fi037.mp4: 79
Frames extracted from /content/Suspicious Activity Detection/yes/fi047.mp4: 72
Frames extracted from /content/Suspicious Activity Detection/yes/fi109.mp4: 54
Frames extracted from /content/Suspicious Activity Detection/yes/fi132.mp4: 57
Frames extracted from /content/Suspicious Activity Detection/yes/fi113.mp4: 31
Frames extracted from /content/Suspicious Activity Detection/yes/fi115.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi136.mp4: 20
Frames extracted from /content/Suspicious Activity Detection/yes/fi086.mp4: 26
Frames extracted from /content/Suspicious Activity Detection/yes/fi043.mp4: 42
Frames extracted from /content/Suspicious Activity Detection/yes/fi145.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi071.mp4: 103
Frames extracted from /content/Suspicious Activity Detection/yes/fi066.mp4: 48
Frames extracted from /content/Suspicious Activity Detection/yes/fi063.mp4: 142
Frames extracted from /content/Suspicious Activity Detection/yes/fi021.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi119.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi020.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi142.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi073.mp4: 88
Frames extracted from /content/Suspicious Activity Detection/yes/fi134.mp4: 20
Frames extracted from /content/Suspicious Activity Detection/yes/fi095.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi009.mp4: 57
Frames extracted from /content/Suspicious Activity Detection/yes/fi133.mp4: 20
Frames extracted from /content/Suspicious Activity Detection/yes/fi054.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi038.mp4: 51
Frames extracted from /content/Suspicious Activity Detection/yes/fi067.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi069.mp4: 107
Frames extracted from /content/Suspicious Activity Detection/yes/fi138.mp4: 48
Frames extracted from /content/Suspicious Activity Detection/yes/fi051.mp4: 75
Frames extracted from /content/Suspicious Activity Detection/yes/fi050.mp4: 58
Frames extracted from /content/Suspicious Activity Detection/yes/fi083.mp4: 48
Frames extracted from /content/Suspicious Activity Detection/yes/fi014.mp4: 63
Frames extracted from /content/Suspicious Activity Detection/yes/fi087.mp4: 23
Frames extracted from /content/Suspicious Activity Detection/yes/fi123.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi011.mp4: 58
Frames extracted from /content/Suspicious Activity Detection/yes/fi045.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi077.mp4: 84
Frames extracted from /content/Suspicious Activity Detection/yes/fi129.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi052.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi036.mp4: 48
Frames extracted from /content/Suspicious Activity Detection/yes/fi108.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi058.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi111.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi118.mp4: 53
Frames extracted from /content/Suspicious Activity Detection/yes/fi144.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi003.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi120.mp4: 50
Frames extracted from /content/Suspicious Activity Detection/yes/fi140.mp4: 60
Frames extracted from /content/Suspicious Activity Detection/yes/fi055.mp4: 60
```

```
# Create training dataset with 70% of the data
train_ds = keras.utils.image_dataset_from_directory(
    directory='/content/Frames Extracted From Video Dataset',
    labels='inferred',
    label_mode='int',
    batch_size=32,
    image_size=(256, 256),
    validation_split=0.3,
    subset="training",  # This indicates that we want the training portion of the data
    seed=42
)
```

```
Found 17120 files belonging to 2 classes.
Using 11984 files for training.
```

```
# Create testing dataset with the remaining 30% of the data
validation_ds = keras.utils.image_dataset_from_directory(
    directory='/content/Frames Extracted From Video Dataset',
    labels='inferred',
    label_mode='int',
    batch_size=32,
    image_size=(256, 256),
    validation_split=0.3,
    subset="validation",  # This indicates that we want the validation portion of the data
    seed=42
)
```

```
Found 17120 files belonging to 2 classes.
Using 5136 files for validation.
```

```
# Normalize
def process(image,label):
 image = tf.cast(image/255. ,tf.float32)
 return image,label


train_ds = train_ds.map(process)
validation_ds = validation_ds.map(process)


# create CNN model
model = Sequential()
model.add(Conv2D(32,kernel_size=(3,3),padding='valid',activation='relu',input_shape=(256, 256, 3)))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2),strides=2,padding='valid'))
model.add(Conv2D(64,kernel_size=(3,3),padding='valid',activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2),strides=2,padding='valid'))
model.add(Conv2D(128,kernel_size=(3,3),padding='valid',activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2),strides=2,padding='valid'))
model.add(Flatten())
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(1,activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 32)      896

 batch_normalization (Batch  (None, 254, 254, 32)      128
 Normalization)

 max_pooling2d (MaxPooling2  (None, 127, 127, 32)      0
 D)

 conv2d_1 (Conv2D)           (None, 125, 125, 64)      18496

 batch_normalization_1 (Bat  (None, 125, 125, 64)      256
 chNormalization)

 max_pooling2d_1 (MaxPoolin  (None, 62, 62, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 128)       73856

 batch_normalization_2 (Bat  (None, 60, 60, 128)       512
 chNormalization)

 max_pooling2d_2 (MaxPoolin  (None, 30, 30, 128)       0
 g2D)

 flatten (Flatten)           (None, 115200)            0

 dense (Dense)               (None, 128)               14745728

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256
```

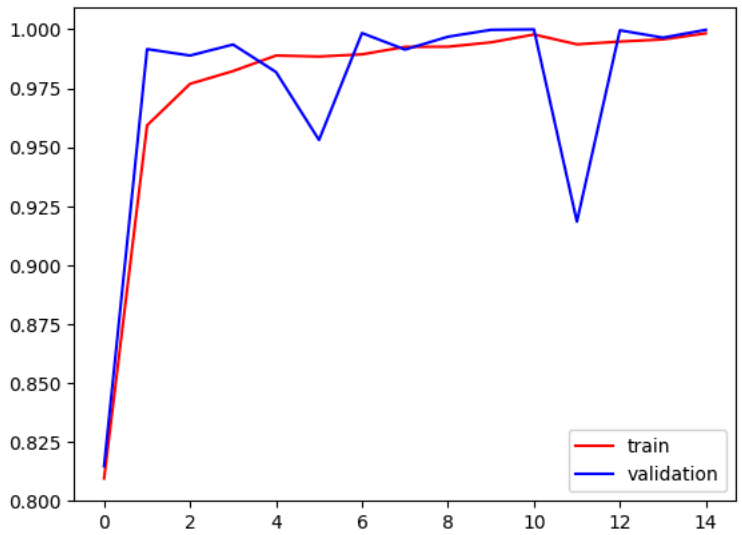
```
 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 14848193 (56.64 MB)
Trainable params: 14847745 (56.64 MB)
Non-trainable params: 448 (1.75 KB)
_________________________________________________________________
```

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

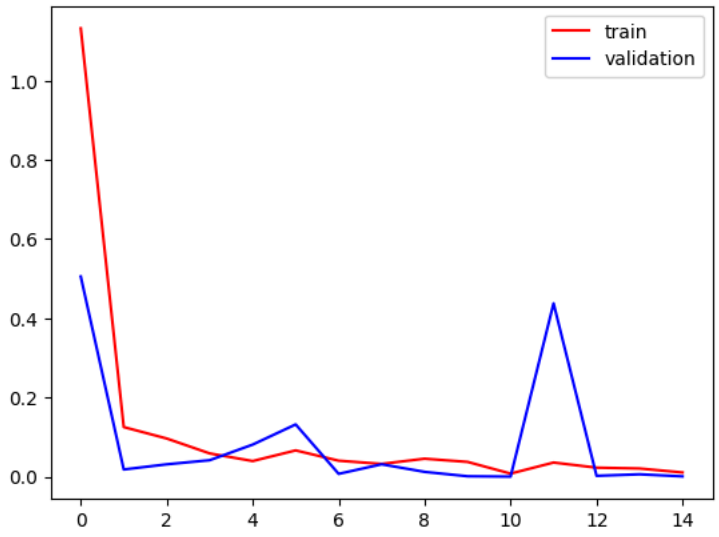
```
history = model.fit(train_ds,epochs=15,validation_data = validation_ds)
```

```
Epoch 1/15
375/375 [==============================] - 52s 111ms/step - loss: 1.1321 - accuracy: 0.8094 - val_loss: 0.5053 - val_accuracy: 0.8146
Epoch 2/15
375/375 [==============================] - 39s 103ms/step - loss: 0.1252 - accuracy: 0.9593 - val_loss: 0.0181 - val_accuracy: 0.9916
Epoch 3/15
375/375 [==============================] - 39s 103ms/step - loss: 0.0962 - accuracy: 0.9769 - val_loss: 0.0311 - val_accuracy: 0.9889
Epoch 4/15
375/375 [==============================] - 39s 102ms/step - loss: 0.0586 - accuracy: 0.9823 - val_loss: 0.0415 - val_accuracy: 0.9936
Epoch 5/15
375/375 [==============================] - 40s 105ms/step - loss: 0.0393 - accuracy: 0.9889 - val_loss: 0.0809 - val_accuracy: 0.9819
Epoch 6/15
375/375 [==============================] - 44s 118ms/step - loss: 0.0662 - accuracy: 0.9885 - val_loss: 0.1318 - val_accuracy: 0.9531
Epoch 7/15
375/375 [==============================] - 39s 104ms/step - loss: 0.0402 - accuracy: 0.9894 - val_loss: 0.0070 - val_accuracy: 0.9984
Epoch 8/15
375/375 [==============================] - 39s 105ms/step - loss: 0.0321 - accuracy: 0.9926 - val_loss: 0.0312 - val_accuracy: 0.9914
Epoch 9/15
375/375 [==============================] - 44s 117ms/step - loss: 0.0453 - accuracy: 0.9927 - val_loss: 0.0120 - val_accuracy: 0.9969
Epoch 10/15
375/375 [==============================] - 44s 118ms/step - loss: 0.0372 - accuracy: 0.9945 - val_loss: 9.3054e-04 - val_accuracy: 0.99
Epoch 11/15
375/375 [==============================] - 44s 118ms/step - loss: 0.0077 - accuracy: 0.9978 - val_loss: 9.0602e-05 - val_accuracy: 1.00
Epoch 12/15
375/375 [==============================] - 40s 105ms/step - loss: 0.0356 - accuracy: 0.9937 - val_loss: 0.4376 - val_accuracy: 0.9184
Epoch 13/15
375/375 [==============================] - 40s 105ms/step - loss: 0.0226 - accuracy: 0.9948 - val_loss: 0.0020 - val_accuracy: 0.9996
Epoch 14/15
375/375 [==============================] - 39s 104ms/step - loss: 0.0206 - accuracy: 0.9957 - val_loss: 0.0059 - val_accuracy: 0.9965
Epoch 15/15
375/375 [==============================] - 39s 103ms/step - loss: 0.0105 - accuracy: 0.9983 - val_loss: 6.1737e-04 - val_accuracy: 0.99
```

```
import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'],color='red',label='train')
plt.plot(history.history['val_accuracy'],color='blue',label='validation')
plt.legend()
plt.show()
```



```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'],color='red',label='train')
plt.plot(history.history['val_loss'],color='blue',label='validation')
plt.legend()
plt.show()
```




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

```
 161/161 [==============================] - 6s 35ms/step - loss: 6.1737e-04 - accuracy: 0.9998
 Validation Loss: 0.0006173738511279225
 Validation Accuracy: 0.9998052716255188
```

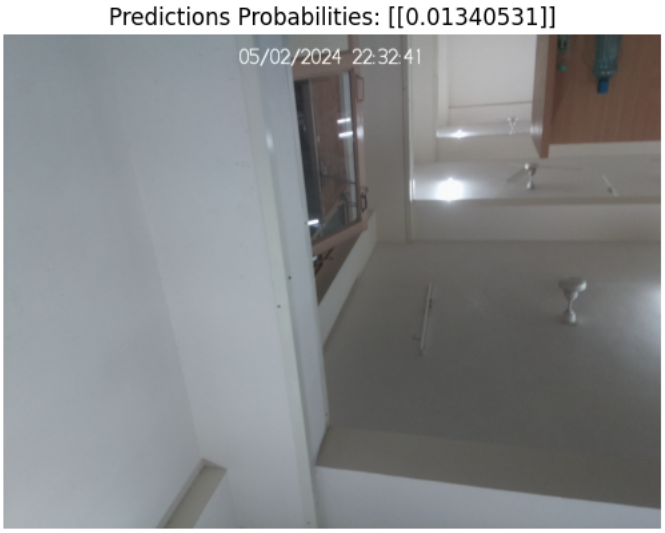
```
# Function to preprocess each frame
def preprocess_frame(frame):
    # Resize the frame to match the model input size
    resized_frame = cv2.resize(frame, (256, 256))
    # Normalize the frame
    normalized_frame = resized_frame / 255.0
    return normalized_frame


# 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/05-02-2024_22-32-41.jpg
```

```
# 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()
```

```
        plt.show()

        # Press 'q' to exit
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

# Release everything when done
cap.release()
cv2.destroyAllWindows()
```

```
 1/1 [==============================] - 0s 366ms/step
 Predictions Probabilities:
  [[0.01340531]]
```



Start coding or generate with AI.