```
!pip install mtcnn
!pip install pillow
```

```
Collecting mtcnn
  Downloading mtcnn-0.1.1-py3-none-any.whl (2.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 22.7 MB/s eta 0:00:00
Requirement already satisfied: keras>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from mtcnn) (2.14.0)
Requirement already satisfied: opencv-python>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from mtcnn) (4.8.0.76)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv-python>=4.1.0->mtcnn) (1.23.5)
Installing collected packages: mtcnn
Successfully installed mtcnn-0.1.1
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (9.4.0)
```

```
from mtcnn import MTCNN
from PIL import Image, ImageDraw
import os
import numpy as np


def draw_bounding_boxes(image_path, result_list):
    data = Image.open(image_path)
    draw = ImageDraw.Draw(data)
    for result in result_list:
        x, y, width, height = result['box']
        rect = [(x, y), (x + width, y + height)]
        draw.rectangle(rect, outline="red", width=3)

    return data



detector = MTCNN()


def process_directory(input_dir, output_dir):
    for root, dirs, files in os.walk(input_dir):
        for file in files:
            if file.lower().endswith(('.png', '.jpg', '.jpeg')):
                path = os.path.join(root, file)
                try:
                    with Image.open(path) as img:
                        pixels = np.array(img)
                        faces = detector.detect_faces(pixels)
                        if faces:
                            image_with_boxes = draw_bounding_boxes(path, faces)
                            relative_path = os.path.relpath(root, input_dir)
                            save_dir = os.path.join(output_dir, relative_path)
                            if not os.path.exists(save_dir):
                                os.makedirs(save_dir)
                            output_path = os.path.join(save_dir, file)
                            image_with_boxes.save(output_path)
                except Exception as e:
                    print(f"An error occurred with image {path}: {e}")



input_directory = "/content/drive/MyDrive/DL/data/raw"
output_directory = "/content/drive/MyDrive/DL/data/processed"



process_directory(input_directory, output_directory)

print("All images have been processed.")
```

```
1/1 [==============================] - 1s 515ms/step
1/1 [==============================] - 0s 199ms/step
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 16ms/step
```

```
1/1 [==============================] - 0s 14ms/step
34/34 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 12ms/step
1/1 [==============================] - 1s 1s/step
1/1 [==============================] - 1s 706ms/step
1/1 [==============================] - 1s 564ms/step
1/1 [==============================] - 0s 251ms/step
1/1 [==============================] - 0s 117ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
168/168 [==============================] - 1s 5ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 68ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
39/39 [==============================] - 0s 5ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 1s 1s/step
1/1 [==============================] - 1s 634ms/step
1/1 [==============================] - 0s 316ms/step
1/1 [==============================] - 0s 142ms/step
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
```