

Phát hiện đối tượng

Các mô hình phát hiện đối tượng xác định thứ gì đó trong hình ảnh và các bộ dữ liệu phát hiện đối tượng đã được sử dụng cho các ứng dụng như lái xe tự động và phát hiện các mối nguy hiểm tự nhiên như cháy rừng. Hướng dẫn này sẽ chỉ cho bạn cách áp dụng các phép biến đổi cho tập dữ liệu phát hiện đối tượng sau hướng dẫn từ Albumentations.

Để chạy các ví dụ này, hãy đảm bảo bạn có phiên bản cập nhật của albumentations và cv2 đã cài đặt:

```
cài đặt pip -U albumentations opencv-python
```

Trong ví dụ này, bạn sẽ sử dụng tập dữ liệu cppe-5 để xác định phương tiện bảo vệ cá nhân y tế equipment (PPE) in the context of the COVID-19 pandemic.

Tải tập dữ liệu và xem một ví dụ:

```
>>> from datasets import load_dataset

>>> ds = load_dataset("cppe-5")
>>> example = ds['train'][0]
>>> example
{'height': 663,
 'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=943x663 at 0x7FC3DC756250>,
 'image_id': 15,
 'objects': {'area': [3796, 1596, 152768, 81002],
 'bbox': [[302.0, 109.0, 73.0, 52.0],
 [810.0, 100.0, 57.0, 28.0],
 [160.0, 31.0, 248.0, 616.0],
 [741.0, 68.0, 202.0, 401.0]],
 'category': [4, 4, 0, 0],
 'id': [114, 115, 116, 117]},
 'width': 943}
```

Tập dữ liệu có các trường sau:

- image : Đối tượng PIL.Image.Image chứa hình ảnh.

- image_id : ID hình ảnh.
- Height : Chiều cao của hình ảnh.
- width : Chiều rộng của hình ảnh.
- đối tượng : Một từ điển chứa siêu dữ liệu hộp giới hạn cho các đối tượng trong ảnh:
 - id : Id chú thích.
 - Area : Diện tích của khung giới hạn.
 - bbox : The object's bounding box (in the coco format).
 - category : The object's category, with possible values including Coverall (0) , Face_Shield (1) , Gloves (2) , Goggles (3) and Mask (4) .

Bạn có thể hình dung các bbox trên hình ảnh bằng cách sử dụng một số tiện ích đèn pin bên trong. Để làm điều này, bạn sẽ cần tham chiếu tính năng ClassLabel được liên kết với ID danh mục để bạn có thể xem lên nhãn chuỗi:

```
>>> import torch
>>> from torchvision.ops import box_convert
>>> from torchvision.utils import draw_bounding_boxes
>>> from torchvision.transforms.functional import pil_to_tensor, to_pil_image

>>> categories = ds['train'].features['objects'].feature['category']

>>> boxes_xywh = torch.tensor(example['objects']['bbox'])
>>> boxes_xyxy = box_convert(boxes_xywh, 'xywh', 'xyxy')
>>> labels = [categories.int2str(x) for x in example['objects']['category']]
>>> to_pil_image(
...draw_bounding_boxes(
...pil_to_tensor(example['image']),
...hộp_xyxy,
...colors="red",
...labels=labels,
...)
... )
```



coverall

Mask

coverall

Mask

```

>>> import albumentations
>>> import numpy as np

>>> transform = albumentations.Compose([
...albumentations.Resize(480, 480),
...albumentations.HorizontalFlip(p=1.0),
...albumentations.RandomBrightnessContrast(p=1.0),
... ], bbox_params=albumentations.BboxParams(format='coco',label_fields=['category']))

>>> image = np.array(example['image'])
>>> out = transform(
...image=image,
...bboxes=example['objects']['bbox'],
...category=example['objects']['category'],
... )

```

Bây giờ khi bạn hình dung kết quả, hình ảnh sẽ được lật lại, nhưng các hộp vẫn phải ở những nơi thích hợp.

```

>>> image = torch.tensor(out['image']).permute(2, 0, 1)
>>> boxes_xywh = torch.stack([torch.tensor(x) for x in out['bboxes']])
>>> boxes_xyxy = box_convert(boxes_xywh, 'xywh', 'xyxy')
>>> labels = [categories.int2str(x) for x in out['category']]
>>> to_pil_image(
...draw_bounding_boxes(
...hình ảnh,
...hộp_xyxy,
...colors='red',
...labels=labels
... )
... )

```

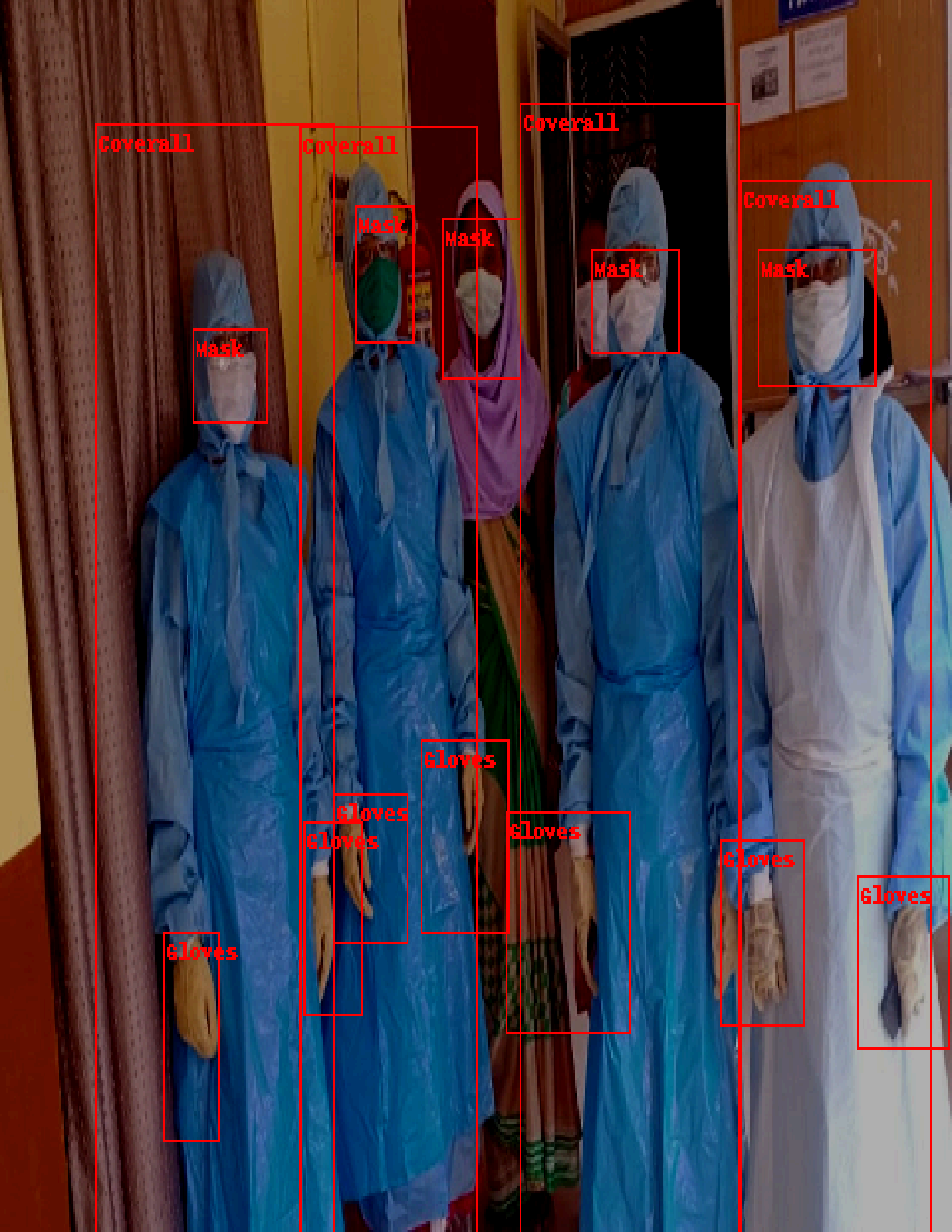
Coverall

Mask

Coverall

Mask





Coverall

Coverall

Coverall

Coverall

Mask

Mask

Mask

Mask

Mask

Gloves

Gloves
Gloves

Gloves

Gloves

Gloves

Gloves

Bây giờ bạn đã biết cách xử lý tập dữ liệu để phát hiện đối tượng, hãy tìm hiểu cách huấn luyện mô hình phát hiện đối tượng và sử dụng nó để suy luận.