yolov5を利用したend-to-end物体検知

2020年10月07日

Yolo v5とは

YOLO vの名称をめぐる論争が起こるなか、Ultralyticsは最近、YOLOv5のリリースを発表しました。背景をご説明すると、YOLO（You Only Look Once）の最初の三つのバージョンは、ジョゼフ・レッドモンによって作成されました。これに続いてアレクセイ・ボチコブスキーがダークネットで作成したYOLOv4は、従来のバージョンと比べて、高い平均適合率（AP）と速い処理速度を誇っています。

現在、Ultralyticsは、YOLOv4に匹敵する平均適合率（AP）を持ち、推論処理時間がより速いYOLOv5をリリースしています。この発表を受け、YOLOv5は本当にYOLOv4と同じ精度を維持しつつ、改善されたバージョンなのだろうかと多くの人が疑問を呈しています。その答えがどのようなものになるとしても、この一連の流れは間違いなく、物体検出コミュニティがどれほど急速に進化しているかを示しています。

UltralyticsがYOLOv3を移植した以来、pytorchを使用したモデルの作成やデプロイが非常に簡単になったので、私はぜひともYOLOv5を試してみたいと思いました。YOLOv5を使用したところ、Ultralyticsはこのバージョンでさらにプロセスを簡素化しており、上記の疑問に対しては、結果が全てを物語っています。

この記事では、YOLOv5を使用して検出モデルを作成し、データセットの作成およびアノテーションから、優れたライブラリを用いた学習および推論に至るまでを実行します。次のようなプロセスを含むYOLOv5の実装に焦点を置いています。

- トイデータセットの作成
- 画像データにアノテーションを付与
- プロジェクト構造の作成
- YOLOv5のトレーニング
- 予測

データセット作成

作業するための画像データセットが手元にないので、Open Image Dataset（OID）からデータをダウンロードします。これは、分類と検出どちらにも利用できるアノテーション付き画像データを含むすばらしいリソースです。ただし、今回はアノテーション方法を学ぶため、OIDが提供するアノテーションを使用せずに、独自にアノテーションを作成することにします。

①　OIDv4から画像をダウンロード

Open Image Datasetから画像をダウンロードするために、OIDv4_ToolKitのクローンを作成して、全ての要件をインストールすることから始めます。

```
git clone https://github.com/EscVM/OIDv4_ToolKit
cd OIDv4_ToolKit

pip install -r requirements.txt
```

これで、このフォルダ内の`main.py`スクリプトを使用して、多クラス用の画像とアノテーションをダウンロードすることができます。 クリケットボールとサッカーボールの画像データをダウンロードして、独自のデータセットを作成します。ここでの学習タスクは、サッカーボールとクリケットボールを検出することです。

```
python3 main.py downloader --classes Cricket_ball Football --type_csv all -y --limit 500
```

上記のコマンドは、以下の構造を持つ「OID」という名前のディレクトリを作成します。

次に進む前に、データのアノテーションを開始するため、同じフォルダ内の全ての画像をコピーする必要があります。これは、手動で行うこともできますし、再起的なglob関数を用いてプログラムすることもできます。

```
from glob import glob

os.system("mkdir Images")
images = glob(r'OID/**/*.jpg', recursive=True)
for img in images:
    os.system(f"cp {img} Images/")
```

②　画像データにアノテーションを付与

アノテーションにはオープンソースのツールを利用します。以下のように、プロジェクトを作成し、ラベルを設定すれば、すぐにアノテーションを行うことができます。

アノテーションが終了したら、YOLO形式でデータをエクスポートします。必要に応じて、JSON形式（COCO）またはXML形式（Pascal VOC）でアノテーションを取得することもできます。

YOLO形式でエクスポートすると、各画像ごとに[the class_id, x_center, y_center, width,height]の.txtファイルが作成されます。また、obj.namesという名称のファイルも作成され、これはクラス名にclass_idをマップするために役立ちます。以下に例を記載しておきます。

アノテーションファイルの座標は0から1の範囲でスケーリングされます。また、class_idは、0からスタートするobj.namesファイルに従って、クリケットボールの場合は0、サッカーボールの場合は1になります。このエクスポートを利用して作成される他のファイルもいくつかありますが、この例では使用しません。

エクスポートが完了すれば、モデルの学習の際、次に行うトレーニングと検証のためにファイルのいくつかを再配置するだけですみます。データセットは、以下のとおり、画像とアノテーションの両方を含む単一のフォルダになります。

```
dataset
    - 0027773a6d54b960.jpg
    - 0027773a6d54b960.txt
    - 2bded1f9cb587843.jpg
    - 2bded1f9cb587843.txt
    --
    --
```

プロジェクト設定

独自の物体検出器をトレーニングするため、UltralyticsのYOLOv5を使用します。まず、リポジトリのクローンを作成し、依存関係をインストールします。

```
git clone https://github.com/ultralytics/yolov5 # clone repo
cd yolov5
pip install -U -r requirements.txt
```

次に独自のデータセットを保存する「training」という名称のフォルダを作成します。

```
!mkdir training
```

独自のデータセットフォルダをこのフォルダにコピーし、シンプルな`train_val_folder_split.ipynb`ノートブックを利用して`train/validation`フォルダを作成します。以下のコードは、トレーニングおよび検証フォルダを作成し、それらに画像を入れるために利用できます。

```
import glob, os
import random

# put your own path here
dataset_path = 'dataset'

# Percentage of images to be used for the validation set
percentage_test = 20

!mkdir data
!mkdir data/images
!mkdir data/labels
!mkdir data/images/train
!mkdir data/images/valid
!mkdir data/labels/train
!mkdir data/labels/valid

# Populate the folders
p = percentage_test/100
for pathAndFilename in glob.iglob(os.path.join(dataset_path, "*.jpg")):
    title, ext = os.path.splitext(os.path.basename(pathAndFilename))
    if random.random() <=p :
        os.system(f"cp {dataset_path}/{title}.jpg data/images/valid")
        os.system(f"cp {dataset_path}/{title}.txt data/labels/valid")
    else:
        os.system(f"cp {dataset_path}/{title}.jpg data/images/train")
        os.system(f"cp {dataset_path}/{title}.txt data/labels/train")
```

これを実行すると、データフォルダ構造は下の画像のようになり、画像用とアノテーション用の二つのディレクトリができます。

ここで、`training`フォルダに構成ファイルを二つ追加する必要があります。

① **Dataset.yaml**:「dataset.yaml」ファイルには、トレーニング画像および検証画像のパスとクラスが含まれます。

```
# train and val datasets (image directory or *.txt file with image paths)
train: training/data/images/train/
val: training/data/images/valid/

# number of classes
nc: 2

# class names
names: ['Cricketball', 'Football']
```

② **Model.yaml**: ネットワーク作成の際、大小様々な複数のモデルを利用できます。例えば、`yolov5/models`ディレクトリの`yolov5s.yaml`ファイルは、パラメータ数700万個の小さなYOLOモデルですが、`yolov5x.yaml`は、9600万個のパラメータを持つ最大のYOLOモデルです。このプロジェクトでは、パラメータ数5000万個の`yolov5l.yaml`ファイルを使用します。これを行うためには、`yolov5/models/yolov5l.yaml`のファイルをトレーニングフォルダにコピーし、プロジェクトの要件に従って`nc`（クラス数）を2に変更します。

```
# parameters
nc: 2 # change number of classes
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple
```

Yolo v5のトレーニング

現在のところ、トレーニングフォルダは次のようになっています。

上記の手順を終了すれば、モデルのトレーニングに移れます。以下のコマンドを実行するだけで簡単に、構成ファイルの場所や様々なパラメータを提供できます。train.pyファイルには他のオプションもありますが、ここでは、私が注目に値すると思ったものを記載しておきます。

```
# Train yolov5l on custom dataset for 300 epochs
$ python train.py --img 640 --batch 16 --epochs 300--data training/dataset.yaml --cfg training/yolov5l.yaml --weights ''
```

複数のGPUを使用していると、pytorchの最新バージョン（1.5）で上記のコマンドを実行する際にエラーが発生することがあります。その場合は、以下のコードを利用して単一のGPUで実行することを選択できます。

```
# Train yolov5l on custom dataset for 300 epochs
$ python train.py --img 640 --batch 16 --epochs 300--data training/dataset.yaml --cfg training/yolov5l.yaml --weights ''
--device 0
```

トレーニングの開始は、自動的に作成される`train_batch0.jpg`ファイルを見て確認できます。このファイルには、最初のバッチの教師データ用アノテーションが含まれます。また、テスト画像のグラウンドトゥルースが含まれる`test_batch0_gt.jpg`も確認することができます。以下のような感じになります。

結果

テンソルボードを使用し、ブラウザで`localhost:6006`にアクセスして学習結果を確認するためには、別のターミナルタブで次のコマンドを実行してください。

```
tensorboard --logdir=runs
```

以下は様々な検証指標です。トレーニング終了時、これらの指標も`results.png`ファイルに保存されます。

予測

Ultralytics YOLOv5は、新しいデータに対する結果を確認するための様々な方法を提供しています。画像を検出するためには、`inference/images`という名称のフォルダに入れ、検証の平均適合率（AP）に従って最良の重みを使用した推論を実行するだけです。

`detect.py`ファイルを用いて、動画でもこれを行うことができます。

```
python detect.py --weights weights/best.pt --source inference/videos/messi.mp4 --view-img --output inference/output
```

ここでは`view-img`フラグを使用して出力の表示を指定し、inference/outputの場所に出力を保存します。すると、この場所に`.mp4`ファイルが作成されます。ネットワークがボールを検出し、すばらしい速度と精度で推論するのを最初に見たときは非常に驚きました。

`--source`を0に指定して、ウェブカメラをソースとして使用することもできます。`detect.py`ファイルの他の様々なオプションもご確認ください。

この記事では、独自のデータセットを利用して、YOLOv5物体検出モデルを作成する方法を説明いたしました。Ultralyticsが物体検出モデルを非常に簡単に作成する方法を提供してくれたことは高く評価したいと思います。さらに、Ultralyticsはモデルの結果を視覚化する様々な方法も提供しています。

今回作成したデータセットを使って実験をしたい場合は、Kaggleのアノテーション付きデータとGithubのコードをダウンロードしてください。

当サイトでは、機械学習の技術ガイドを定期的に更新しております。多クラス画像分類や文書分類のガイドも公開しているので、もぜひご覧ください。

LionbridgeのAI学習データサービスについて

当社はAI向け教師データの作成やアノテーションのサービスを提供し、研究開発を支援しています。世界の各タイムゾーンを渡る、100万人のコントリビューターが登録されており、大規模なAIプロジェクトも素早く仕上げることができます。無料トライアルやご相談は、こちらからお気軽にお問い合わせください。

※ 本記事は2020年6月22日、弊社英語ブログに掲載された寄稿記事に基づいたものです。