
引き継ぎ資料簡易版

1 プログラムについて

disk011/usr/ogata/Mask_RCNN 以下の samples/coco/に置いてあります。

<https://arxiv.org/abs/1703.06870> の MaskRCNN を用いています。

プログラム自体は https://github.com/matterport/Mask_RCNN を使っています。環境設定等はこちらを参考にしてください。実行の際は環境が変更されていなければ gpu06 や 07 でそのまま利用できると思います。keras(tensorflow) を利用しています。

本家と違う箇所は coco.py を参考に grape.py を作成している点、grape.py を動かすために幾つかのコードを同じフォルダ内に作成した点、画像を対象にセグメンテーションを行う際の出力を調整するために visualize.py というコードを改変した点です。

2 実行について

実行の前に samples/coco/に移動し、filetext_setting.py を実行して、学習画像の一覧を作成してください。既存のブドウ画像で実行する場合はおそらく既に train_list.txt と val_list.txt が存在するはずなのでこの作業は不要です。学習する際は、Mask_RCNN に移動して以下のコマンドを打ち込んでください。

- `python3 /samples/coco/grape.py train --dataset=/disk011/usrs/ogata/Mask_RCNN --model=coco`

model の部分は前回学習したもの、学習途中のものを利用する場合は coco を last にしてください。

detection については、以下のコマンドを打ち込んでください。

- `python3 /samples/coco/grape.py evaluate --dataset=/disk011/usrs/ogata/Mask_RCNN
--model=last --image=your_file_name`

この場合 your_file_name は画像 1 枚を指します。動画を対象にしたい場合は image の代わりに video にして your_file_name の部分に動画の名前を入力してください。

デモで実際に動かしたように動かしたい場合は以下のコマンドを打ち込んでください

- `python3 samples/coco/grape.py evaluate --dataset=./ --model=last --live=True`

3 各フォルダの中身について

3.1 assets

元々の作者が用意した結果の画像群が入っています。

3.2 data

ブドウの学習用の画像が入っています。画像読み込みの簡易化のため train, val のフォルダに元画像と正解画像が両方入っています。画像サイズはオリジナルの 25 パーセントにリサイズしてあります。

3.3 images

作者が用意したもので関係ありません。

3.4 logs

学習済みのモデルが保存されます。--model=last にすることで、この中の最新の学習済みモデルが使用されます。最新が学習途中で止まった場合などは last を指定して再度実行してもらうと学習の途中から開始されます。

3.5 mrcnn

mask r-cnn のモデルや設定に関するプログラムが置いてあります。visualize.py だけ画像の detection で関係していたため一部を改変してあります。具体的には画像上に実のマスク以外を表示させていた部分（枠線、軸と実で分けた際の実である確率など）をコメントアウトしました。100 行目あたりから 170 行目あたりまでの部分になります。

畳み込みを変更するのであればこのフォルダのプログラムを参照、変更する必要があると思います。

3.6 results

いろいろ実験の結果が乱雑に入れています。

3.7 samples/coco

サンプルプログラムが入っています。

この中の coco の中に grape.py を作ってブドウのセグメンテーションを行っていました。grape.py では保存済みの動画を対象にセグメンテーションする場合 cubemap 化する処理を挟んでいるので、透視図法の動画を対象にする際はコメントアウトしてください。grape.py の 376 行目あたりに記述してあります。

同時に 364 行目あたりの vwriter の動画の幅と高さの部分もすぐ下にコメントアウトしてある行と入れ替えてください。

その他 grape.py に関しては日本語でコメントしている部分もあるので参照してください。

filetext_setting.py を実行することで、data/fastgrape/grape/以下の学習用画像の場所とサイズをテキストファイルに記述するようになっていきます。つまり、データセットの一覧を表したテキストを作ることになります。なので、異なるデータセットを利用することになれば、filetext_setting.py を実行して、一覧を更新してください。filetext_setting.py の中身を変更して train と val の両方の一覧を作成して下さい。

build_mask.py は直接実行することはありません。grape.py 内部でラベル画像の情報を取得するために利用しています。