cloud_detection_panch

June 17, 2021

If you plan on using this implementation, please cite our work:

@INPROCEEDINGS{Grabowski2021IGARSS, author={Grabowski, Bartosz and Ziaja, Maciej and Kawulok, Michal and Nalepa, Jakub}, booktitle={IGARSS 2021 - 2021 IEEE International Geoscience and Remote Sensing Symposium}, title={Towards Robust Cloud Detection in Satellite Images Using U-Nets}, year={2021}, note={in press}}

1 Demo of the cloud detection using U-Net architecture on panchromatic data

This document presents the cloud detection on example Landsat 8 panchromatic images using trained U-Net model. The full script can be found in cloud_detection/exp_panchromatic.py.

First, we import neccessary libraries.

[ ]:
```
import numpy as np
from pathlib import Path
from tensorflow import keras

from cloud_detection.models import unet
from cloud_detection.evaluate_L8CCA import evaluate_model
from cloud_detection.losses import (
    JaccardIndexLoss,
    JaccardIndexMetric,
    DiceCoefMetric,
    recall,
    precision,
    specificity,
)
```

Next, we set the parameters for the experiment. These parameters are the following:

- dpath - path to the dataset.
- rpath - path to directory where results should be stored.
- mpath - path to trained model weights.
- vids - tuple of IDs of images which should be used to create visualizations. If contains '*' visualizations will be created for all images in the datasets.
- eval_imgs - IDs of images to evaluate.

- batch_size - size of generated batches, only one batch is loaded to memory at a time.
- thr - threshold for determining whether pixels contain the clouds.
- learning_rate - learning rate for training (needed to load the trained model).
- bn_momentum - momentum of the batch normalization layer.
- bands - band(s) to load (in this case, panchromatic band is loaded).
- bands_names - name(s) of the band(s) to load.
- resize - whether to resize image to match GT.
- normalize - whether to normalize the data.
- standardize - whether to standardize the data.

[2]:
```
dpath = Path("datasets/clouds/
 ↪Landsat-Cloud-Cover-Assessment-Validation-Data-Partial")
rpath = Path("artifacts/panch_cloud_detection_demo/")
mpath = "examples/panch_cloud_model_weights/best_weights"
vids = ["*"]
eval_imgs = ["LC81390292014135LGN00", "LC81940222013245LGN00"]
batch_size = 8
thr = 0.5
learning_rate = 0.01
bn_momentum = 0.9
bands = (8,)
bands_names = ("panchromatic",)
resize = True
normalize = True
standardize = False
```

We create the instance of the untrained U-Net model. Next, we load the trained weights into the model.

[ ]:
```
model = unet(input_size=1, bn_momentum=bn_momentum)
model.compile(
    optimizer=keras.optimizers.Adam(lr=learning_rate),
    loss=JaccardIndexLoss(),
    metrics=[
        keras.metrics.binary_crossentropy,
        keras.metrics.binary_accuracy,
        JaccardIndexLoss(),
        JaccardIndexMetric(),
        DiceCoefMetric(),
        recall,
        precision,
        specificity,
    ],
)
model.load_weights(mpath)
```

We create the directory to store the results of the model evaluation. Next, we evaluate the model using example Landsat 8 images. The following files are created for each image:

- gt.png - image of the ground-truth cloud mask.
- pred.png - image of the model prediction.
- masks.png - visualisation of the model prediction. Yellow color denotes True Positives, red color denotes False Positives and purple color stands for False Negatives.
- unc.png - uncertainty map, where pixels with uncertain prediction scores are marked in yellow (Note: In the case of the tested model, in most cases almost all of the pixels' prediction scores are very low or very high, which means that the map will almost always not include any yellow pixels.).

If the model's prediction Jaccard Index Metric does not exceed 0.6, the following files are also created:

- roc.html - ROC curve.
- prec_recall.html - precision-recall curve.
- activation_hist.html - histogram of the model's activations scores (please note the logarithmic scale).

[4]:
```python
rpath.mkdir(parents=True, exist_ok=True)
metrics_L8CCA, _ = evaluate_model(
    model=model,
    thr=thr,
    dpath=dpath,
    rpath=rpath / "eval_vis",
    vids=vids,
    batch_size=batch_size,
    bands=bands,
    bands_names=bands_names,
    img_ids=eval_imgs,
    resize=resize,
    normalize=normalize,
    standardize=standardize
)
```

```
Processing Urban-LC81940222013245LGN00
Scene prediction took 132.01824760437012 seconds
Average inference time: 132.01824760437012 seconds
Creating visualisation for LC81940222013245LGN00
/home/bgrabowski/Documents/machine-learning/cloud_detection/utils.py:261:
UserWarning:

artifacts/panch_cloud_detection_demo/eval_vis/LC81940222013245LGN00/gt.png is a
low contrast image

Lossy conversion from int64 to uint8. Range [0, 1]. Convert image to uint8 prior
to saving to suppress this warning.
Processing Barren-LC81390292014135LGN00
Scene prediction took 120.77470016479492 seconds
Average inference time: 126.39647388458252 seconds
Creating visualisation for LC81390292014135LGN00
```

```
/home/bgrabowski/Documents/machine-learning/cloud_detection/utils.py:261:
UserWarning:

artifacts/panch_cloud_detection_demo/eval_vis/LC81390292014135LGN00/gt.png is a
low contrast image

Lossy conversion from int64 to uint8. Range [0, 1]. Convert image to uint8 prior
to saving to suppress this warning.
Will make insights for LC81390292014135LGN00
thr dist variance: 0.00031485004379845925
thr dist mean: 0.21809102287495089
Optimal thr: 0.99871
```

Finally, we process the output metrics to obtain the mean metrics for the model evaluation.

[5]:
```python
mean_metrics_L8CCA = {}
for key, value in metrics_L8CCA.items():
    mean_metrics_L8CCA[key] = np.mean(list(value.values()))
print(mean_metrics_L8CCA)
```

```
{'L8CCA_binary_crossentropy': 1.3964084, 'L8CCA_binary_accuracy': 0.9127327,
'L8CCA_jaccard_index_loss': 0.20696175, 'L8CCA_jaccard_index_metric':
0.79303956, 'L8CCA_dice_coeff_metric': 0.84953636, 'L8CCA_recall': 0.98438203,
'L8CCA_precision': 0.7995982, 'L8CCA_specificity': 0.8915087,
'L8CCA_normalized_mutual_info_score': 0.6823843342159708,
'L8CCA_adjusted_rand_score': 0.7050384008832693}
```