

# cloud\_detection

March 26, 2021

## 1 Demo of the cloud detection using U-Net architecture

This document presents the cloud detection on exemplary Landsat 8 multispectral images using trained U-Net model.

First, we import necessary libraries.

```
[1]: 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,
)
```

```
/home/bgrabowski/miniconda3/envs/cloud_detection/lib/python3.6/site-
packages/tensorflow/python/framework/dtypes.py:523: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
```

```
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
```

```
/home/bgrabowski/miniconda3/envs/cloud_detection/lib/python3.6/site-
packages/tensorflow/python/framework/dtypes.py:524: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
```

```
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

```
/home/bgrabowski/miniconda3/envs/cloud_detection/lib/python3.6/site-
packages/tensorflow/python/framework/dtypes.py:525: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
```

```
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
```

```
/home/bgrabowski/miniconda3/envs/cloud_detection/lib/python3.6/site-
```

```

packages/tensorflow/python/framework/dtypes.py:526: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_uint16 = np.dtype(["uint16", np.uint16, 1])
/home/bgrabowski/miniconda3/envs/cloud_detection/lib/python3.6/site-
packages/tensorflow/python/framework/dtypes.py:527: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint32 = np.dtype(["qint32", np.int32, 1])
/home/bgrabowski/miniconda3/envs/cloud_detection/lib/python3.6/site-
packages/tensorflow/python/framework/dtypes.py:532: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_resource = np.dtype(["resource", np.ubyte, 1])

```

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 visualisations. If contains '\*' visualisations 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.

```

[2]: dpath = Path("datasets/clouds/
↳Landsat-Cloud-Cover-Assessment-Validation-Data-Partial")
rpath = Path("artifacts/cloud_detection_demo/")
mpath = "examples/cloud_model_weights/best_weights"
vids = ["*"]
eval_imgs = ["LC81390292014135LGN00", "LC80460282014171LGN00"]
batch_size = 8
thr = 0.5
learning_rate = 0.01
bn_momentum = 0.9

```

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

```

[3]: model = unet(input_size=4, 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)

```

/home/bgrabowski/miniconda3/envs/cloud\_detection/lib/python3.6/site-packages/tensorflow/python/framework/tensor\_util.py:542: DeprecationWarning:

np.asscalar(a) is deprecated since NumPy v1.16, use a.item() instead

[3]: <tensorflow.python.training.checkpointable.util.CheckpointLoadStatus at 0x7fa37a3dc400>

We create the directory to store the results of the model evaluation. Next, we evaluate the model using exemplary 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]: 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,
    img_ids=eval_imgs
)

```

Processing Urban-LC80460282014171LGN00

Scene prediction took 125.74753499031067 seconds

Average inference time: 125.74753499031067 seconds

Creating visualisation for LC80460282014171LGN00

/home/bgrabowski/Documents/machine-learning/cloud\_detection/utils.py:190:

UserWarning:

artifacts/cloud\_detection\_demo/eval\_vis/LC80460282014171LGN00/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 127.28587102890015 seconds

Average inference time: 126.51670300960541 seconds

Creating visualisation for LC81390292014135LGN00

/home/bgrabowski/Documents/machine-learning/cloud\_detection/utils.py:190:

UserWarning:

artifacts/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: 3.4040578907701817e-06

thr dist mean: 0.3529804072739952

Optimal thr: 1.0

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

```
[5]: 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': 2.5090632, 'L8CCA_binary_accuracy': 0.8431798,  
'L8CCA_jaccard_index_loss': 0.29946098, 'L8CCA_jaccard_index_metric': 0.700539,  
'L8CCA_dice_coeff_metric': 0.7829864, 'L8CCA_recall': 0.95701694,  
'L8CCA_precision': 0.6854487, 'L8CCA_specificity': 0.8053814}
```