

Satellite Images in Cloudy Weather

Project report

Øget konkurrencekraft i landbruget gennem brug af kunstig intelligens - AP1B: Satellitbilleder også i gråvej

Data Science, Digital, SEGES

December 21, 2018



Summary

The objective of the project was to generate biomass maps over danish fields from remote sensing, even in cloudy weather. A state-of-art machine learning method, called a conditional generative adversarial network (cGAN), is utilized to generate Sentinel 2 images only from existing Sentinel 1 data. Our results are promising and shows that our generated Sentinel 2 images often are as close to the ground truth Sentinel 2 images as the previous cloud-free Sentinel 2 images. However, our generator does not capture all local changes in biomass in the field, thus the system is not yet ready for production. Future work for improving the generated images is presented lastly in this notebook.

Table of Contents

1. [Introduction](#)
2. [Methods](#)
 - A. [Data collection](#)
 - B. [Feature set](#)
 - C. [Model](#)
 - D. [Performance evaluation](#)
3. [Results](#)
 - A. [Quantitative results](#)
 - B. [Qualitative results](#)
4. [Discussion](#)
 - A. [Future work](#)

Introduction

In 2016 it became possible to analyze crop growth based on satellite images covering all of Denmark. The Sentinel 2 (S2) satellite acquires an image in the visible light spectrum of the entire country every 5th day, see Figure 1. Such images may be used to compute the green biomass of each field covering part of Denmark. The computed biomasses are displayed in the IT-systems CropSAT and CropManager which are extensively used by farmers and agricultural advisors. The only disadvantage is that Sentinel 2 cannot acquire useful images over areas with cloudy weather. A consequence of this is that only a small percentage of the images produced by Sentinel 2 in the spring of 2017 were useful for computing biomasses due to bad and cloudy weather.

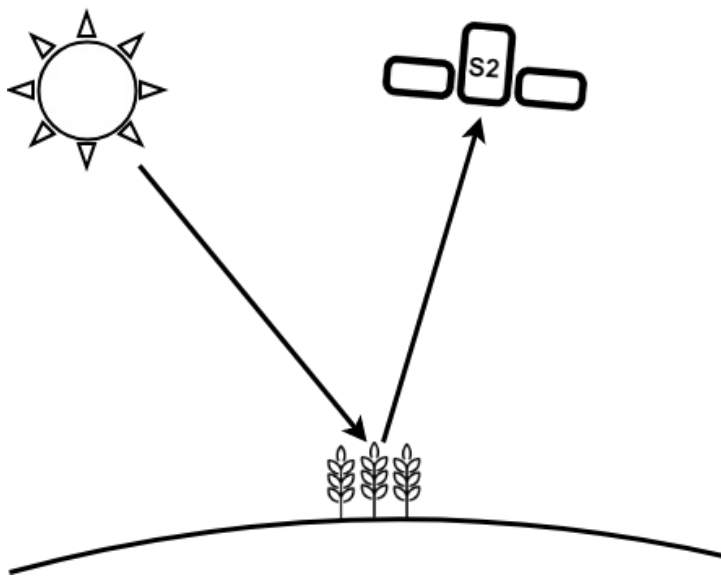


Figure 1: The hyperspectral camera on the Sentinel 2 (S2) satellite photographs the crop.

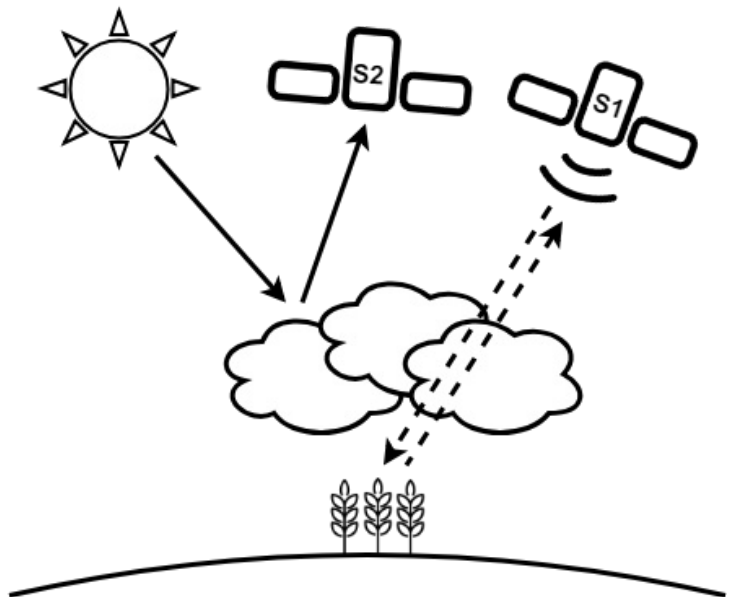
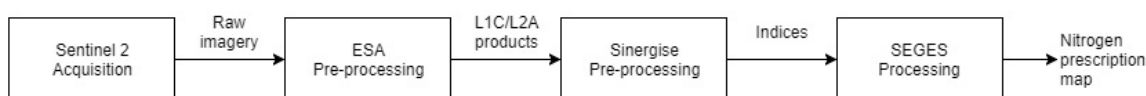


Figure 2: Clouds covers the crop and only the Sentinel 1 satellite can observe the crop.

In addition to the Sentinel 2 satellite, we also have access to radar images from the Sentinel 1 (S1) satellite. The quality of radar images is independent of cloudy weather, as illustrated by Figure 2. The goal of this project is to use machine learning methods to establish a mathematical connection between Sentinel 1 and Sentinel 2 images. It is expected that it is possible to find such a connection based on satellite images from at least two growing seasons. When a reliable connection has been found, the usefulness of CropSAT and CropManager no longer depends on the weather conditions.

Essentially, this task is an investigation of the feasibility of replacing the current pipeline (see in Figure 3) for computing nitrogen prescription maps with a new proposed pipeline (see below) based on Sentinel 1 imagery.

Current pipeline



Proposed new pipeline

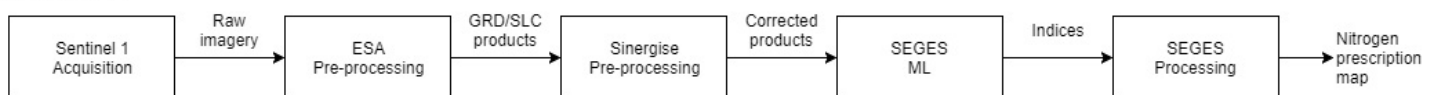


Figure 3: Current and proposed pipelines for satellite imaging processing in SEGES.

Methods

In this chapter we describe the data collection resulting in the utilized feature set. We also describe the machine learning method used for training a model on the feature set, and finally we describe how we evaluate the performance of this model.

Data collection

The collection of satellite images was performed based on the 17.385 danish fields, shown in Figure 4. The criteria for selecting these fields were (as shown in `./MVP3_all_fields_data_set/SIICW-72_field_polygon_dataset.ipynb`):

- Each field has the same polygon in 2016, 2017, and 2018, thus the field have not changed significantly with respect to its areal or the area it covers (see the field polygon normalization in MVP5 of the BDICG project).
- The main crop on each field was either winter wheat or silage maize also in 2016, 2017, and 2018.

This resulted in 103.982 pairs of Sentinel 1 level-1 Ground Range Detected (GRD) (<https://sentinel.esa.int/web/sentinel/missions/sentinel-1/data-products>) and Sentinel 2 level-2 A (<https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products>) satellite images, as shown in `./MVP3_all_fields_data_set/SIICW-139_field_raster_model.ipynb`. The average size of the image pairs is 32 pixels wide and 37 pixels high, where each pixel represents 10x10 meters of the field. The largest field is 141 pixels wide and 183 pixels high. Each Sentinel 1 images contains a channel for each of the two polarisations: VV and VH. Each Sentinel 2 image contains a channel for each of the 12 hyperspectral bands: B01, B02, B03, B04, B05, B06, B07, B08, B8A, B09, B11, B12, and a channel for each of the following three vegetation indices: Normalized Difference Vegetation Index (NDVI) (<https://www.indexdatabase.de/db/i-single.php?id=58>), Normalized Difference RedEdge (NDRE) (<https://www.indexdatabase.de/db/i-single.php?id=223>), Modified Soil Adjusted Vegetation Index 2 (MSAVI2) (http://wiki.landscapetoolbox.org/doku.php/remote_sensing_methods:modified_soil-adjusted_vegetation_index), we computed based on the hyperspectral bands.

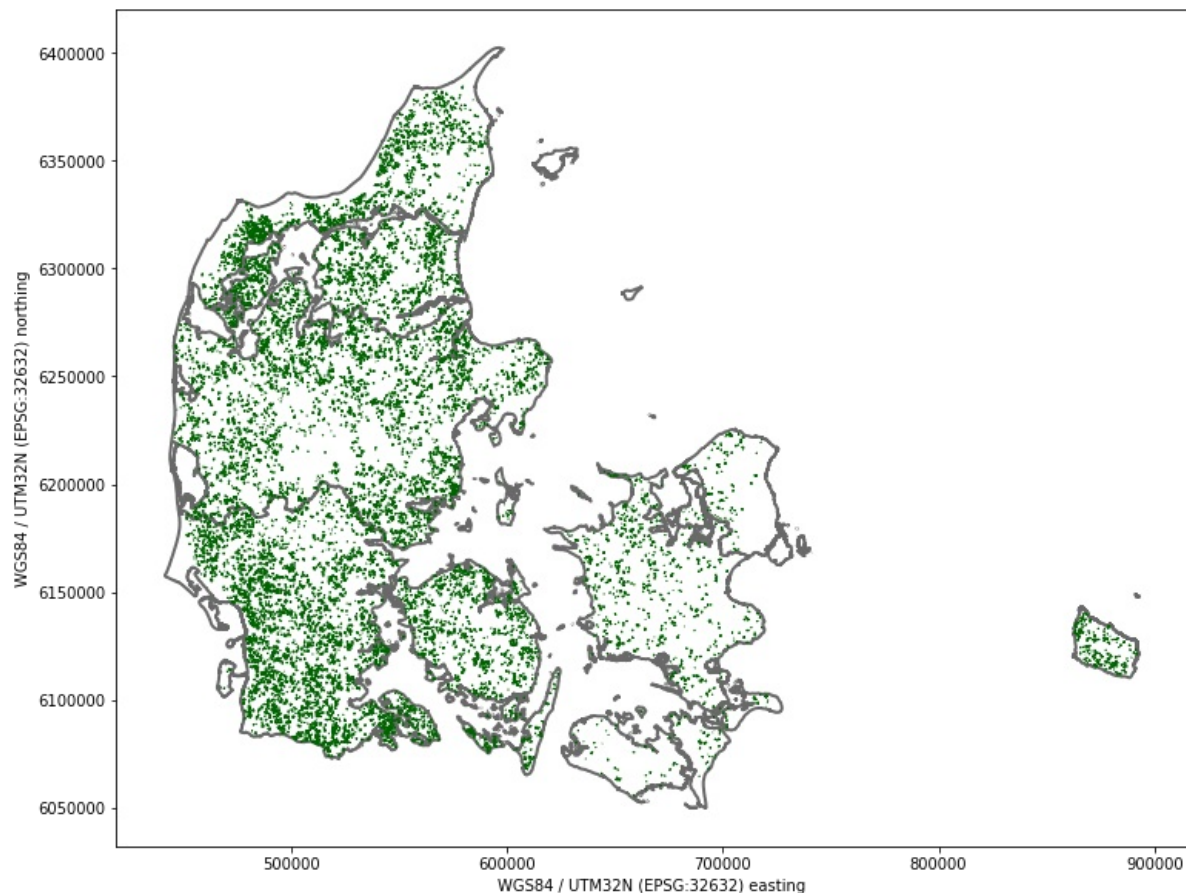


Figure 4: Location of all 17.385 fields (from `./MVP3_all_fields_data_set/SIICW-139_field_raster_model.ipynb`)

Feature set

The collected 103.982 pairs of Sentinel 1 and Sentinel 2 satellite images, was split into a train, a validation, and a test dataset, of the sizes:

- train: 83.233 image pairs.
- validation: 10.149 image pairs.
- test: 10.600 image pairs.

This split is performed such that images of the same field only exist in one of the datasets, as shown in `./MVP4_first_pix2pix_model/SIICW-178_dataset_split_and_attributes.ipynb`.

Model

As our machine learning method for generating realistic Sentinel 2 images based on Sentinel 1 data, we use the pix2pix model first proposed in "[Image-to-Image Translation with Conditional Adversarial Nets \(https://phillipi.github.io/pix2pix/\)](https://phillipi.github.io/pix2pix/)" which is a conditional generative adversarial network (cGAN). A cGAN consists of two parts: a generator and a discriminator, each conditioned on some input data, i.e. in our case the Sentinel 1 data. The generator has the task of generating an image based on an input image, and discriminator has the task of determining if a given image is generated or real. Thereby, the generator can be trained by using the adversarial loss determined by how good the discriminator is to classify images as generated or real.

As concluded in `./MVP4_first_pix2pix_model/pix2pix_setup.md` we chose to use the PyTorch implementation of pix2pix [found here \(https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix\)](https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix), as it is the more suitable implementation to our usage. However, modifications of loading our satellite image data were needed before we could perform our experiments on our satellite image dataset, see `MVP4_first_pix2pix_model/SIICW-163_execute_S1_S2_training.ipynb` for implementation details.

We have experimented with different tasks for the network (i.e. generate the Sentinel 2 image with B04, B05, and B08 or generate the Sentinel 2 NDVI image) and different hyperparameter settings (i.e. L1 regularization constant to 100 or 200, and either 5 or 6 epochs). All experimental networks were trained with only 5 to 6 epochs due to the large size of 83.233 training image pairs. Our best results are based on generating the Sentinel 2 NDVI image using a L1 regularization constant of 200, and the results of this setting will be presented in this report.

Performance evaluation

We evaluate the performance of our generator network based on the pixel-wise error-difference between the generated and the real (i.e. the ground truth) NDVI image, named "fake_error". We also present the pixel-wise error-difference between an NDVI image on a previous and current (i.e. the ground truth) date, named "historic_error". Based on each error-difference, we also report the mean-absolute-error. The evaluation is based on the 10.149 image pairs in the validation dataset, which results in 7.008.884 pixel error samples.

Results

This section presents the results, as computed in the notebook `./MVP4_first_pix2pix_model/SIICW-175_final_visual_examples_of_inferred_S2_images_veg_index.ipynb`.

Quantitative results

The following two graphs based on quantitative data show the pixel-wise error-difference (graph to the left side) and the pixel-wise mean-absolute-error (graph to the right side), both containing the "fake_error" as the blue plot, and "historic_error" as the red plot.

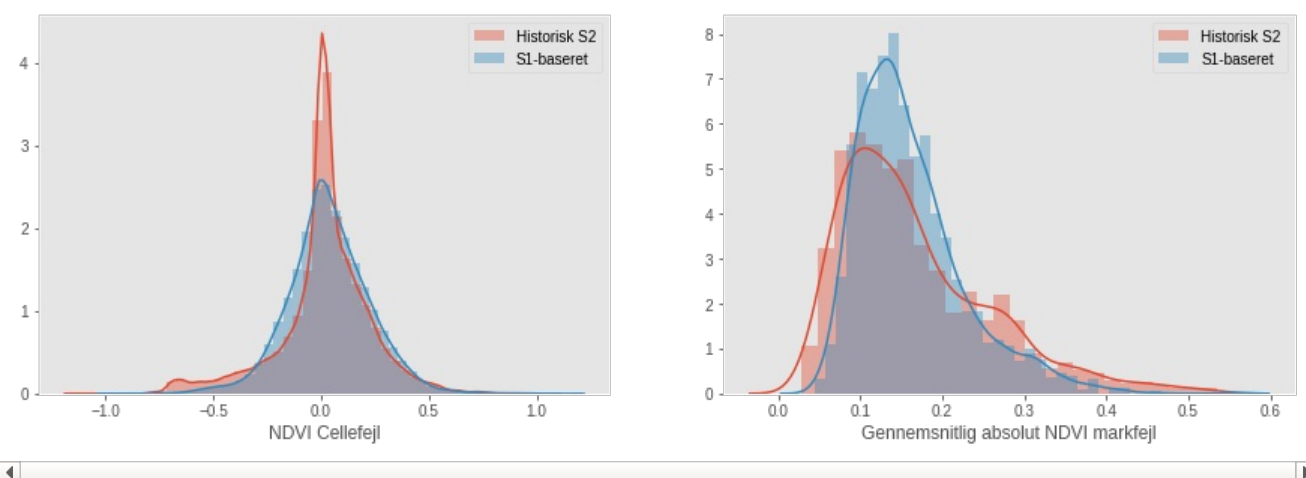
In [17]:

```
fig, axes = plt.subplots(1, 2, figsize=(15, 15/(1.62*2)))
sns.distplot(df_error_table['NDVI_historic_error'].dropna(), ax=axes.flat[0], axlabel=False, label='Historisk S2')
sns.distplot(df_error_table['NDVI_fake_error'].dropna(), ax=axes.flat[0], axlabel=False, label='S1-baseret')
#axes.flat[0].set_title('Cellefejl')
axes.flat[0].set_xlabel('NDVI Cellefejl')
axes.flat[0].legend()
sns.distplot(df_field_mae_error['NDVI_historic_error'].dropna(), ax=axes.flat[1], axlabel=False, label='Historisk S2')
sns.distplot(df_field_mae_error['NDVI_fake_error'].dropna(), ax=axes.flat[1], axlabel=False, label='S1-baseret')
#axes.flat[1].set_title('Gennemsnitlig markfejl')
axes.flat[1].set_xlabel('Gennemsnitlig absolut NDVI markfejl')
axes.flat[1].legend()

for fig_ext in fig_exts:
    fig.savefig('poster/histograms{}'.format(fig_ext))
```

/usr/local/continuum/miniconda3/envs/py36_v2/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

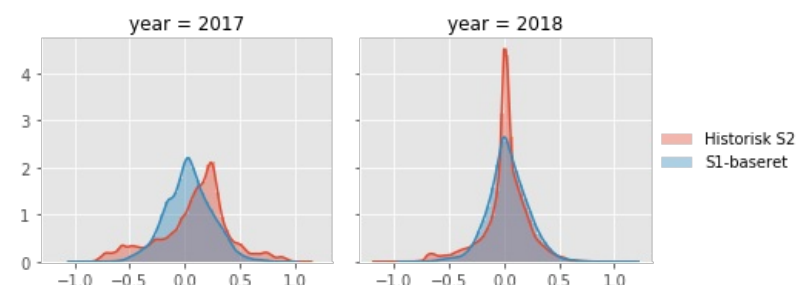
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval



Since the year 2017 had many cloudy days compared to 2018, the individual pixel-wise error-difference of the two years is presented below.

In [8]:

```
#for index in ['NDVI', 'NDRE', 'MSAVI2']:
for index in ['NDVI']:
    g = sns.FacetGrid(df_error_table, col='year', col_wrap=2, margin_titles=True)
    g.map(sns.distplot, f'{index}_historic_error', label=f'Historisk S2')
    g.map(sns.distplot, f'{index}_fake_error', label=f'S1-baseret', color=ggplot_colors[1]).add_legend().set_xlabel('')
    plt.show()
    print('\n\n')
```



Qualitative results

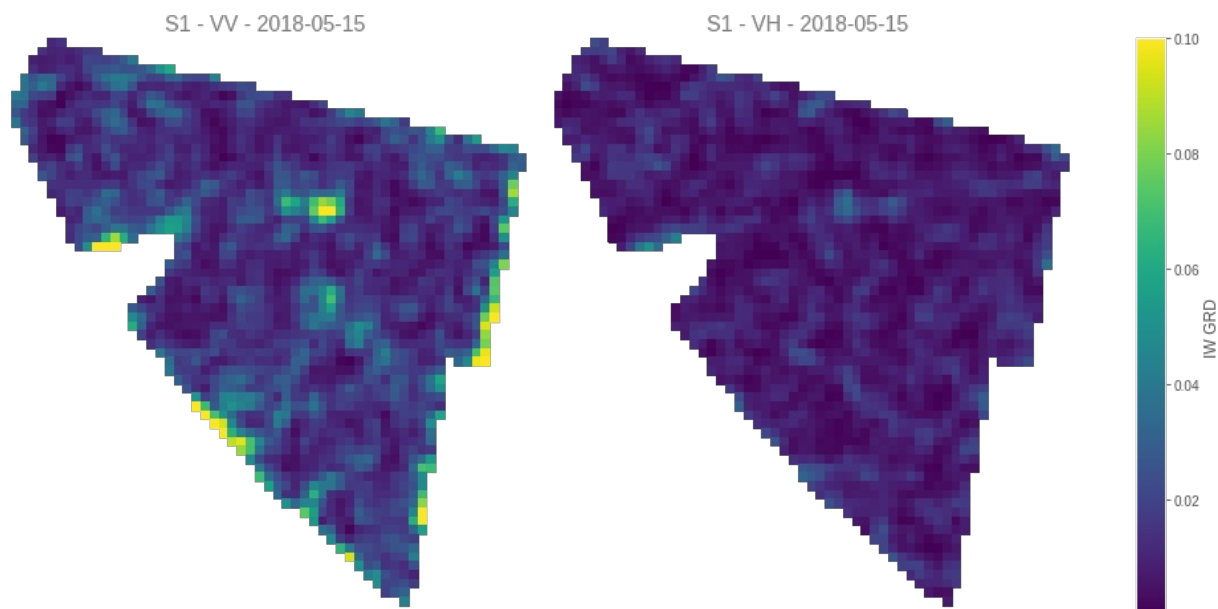
The following visualisation, based on qualitative data, shows the given input data to the pix2pix generator. The input data consist on the Sentinel 1 VV (seen on the left side) and VH (seen on the right side) polarisations.

In [12]:

```
S1_VV = ds.S1_GRD_IW_VV.dropna('time', how='all').sel(time='2018-05-15').where(ds.field_polygon)
S1_VV = xr.where(S1_VV > 0.1, 0.1, S1_VV)
S1_VH = ds.S1_GRD_IW_VH.dropna('time', how='all').sel(time='2018-05-15').where(ds.field_polygon)
pol_index = pd.Index(['VV', 'VH'], name='polarization')
S1_pol = xr.concat([S1_VV, S1_VH], pol_index)
S1_pol.name = 'IW GRD'

fg = S1_pol.plot(col='polarization', size=7)
for ax in fg.axes.flat:
    ax.set_axis_off()
    ax.set_title('S1 - ' + ax.get_title().split()[-1] + ' - 2018-05-15', fontdict=fontdict)

for fig_ext in fig_exts:
    fg.fig.savefig('poster/S1_polarizations{}'.format(fig_ext))
```

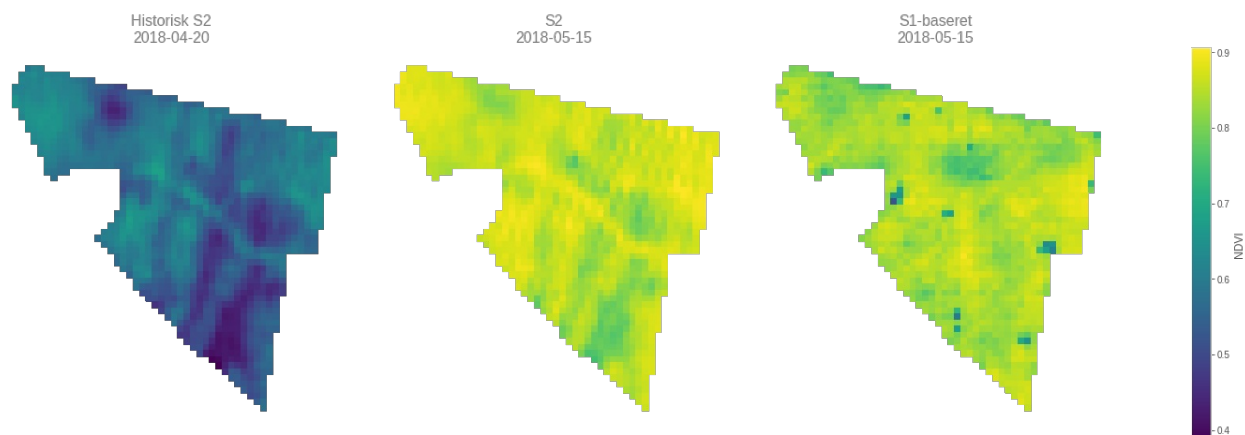


The visualisation below show the comparison of a historical NDVI image (seen on the left side), the ground truth NDVI images (seen in the center), and the pix2pix generated NDVI image (seen on the right side), which is generated using the visualized input data from above.

In [18]:

```
imgs_to_show = {
    11390: '2018-05-15',
    #9096: '2018-05-15'
}

for ID_DDS_field, img_time in imgs_to_show.items():
    ds = dask.delayed(xr.open_dataset)(DATASET_PATH / f'{ID_DDS_field}.nc',
                                     chunks={}).compute()
    visualize_biomass_image_alternatives(
        ds, img_time, visualization_path=Path('poster/'), aspect=1, size=7, erosion_radius=5)
```



Discussion

Our graphs of the quantitative error-difference show that our pix2pix generator can not generate NDVI images with as small error-difference as all of the historic NDVI images. However, the mean-absolute-errors of the generated NDVI images are more centered in the graph, thus the error deviation between different generated images is smaller than the historic NDVI images. We suspect this deviation of the historic NDVI images is due to the deviation in the time span between the acquisition time of the ground truth image and the previous NDVI image. This suspicion is also supported by the tendency shown in the graphs of the error-difference per year. These graphs show that our generated images have on average a smaller error-difference than the historic NDVI image in 2017 and thus is better to use. We believe this is because of the many cloudy days in 2017, which result in a big time span between the acquisition time of cloud-free Sentinel 2 images. Thereby, we conclude further investigation is needed but we assume that the error-difference shows that it can be beneficial to use our pix2pix generated NDVI images instead of the previous NDVI image, when the acquisition time of the previous NDVI image is too large.

The qualitative results show that the pix2pix generator can not capture all of the local NDVI variation in the field, which clearly is captured in the historic NDVI image, here we mainly refer to the spots of lower NDVI values found at the same locations of the field both in the ground truth image and historic NDVI image. However, we find it impressive that the pix2pix generator can capture some of the NDVI variation and generate a NDVI image with NDVI values similar to the ground truth image, i.e. from 0.75 to 0.90 NDVI. These positive results are promising and present a chance that further work can improve the pix2pix generator to capture all of the local NDVI variation, thus making the generated NDVI a better data foundation for predictive models than the historical NDVI image.

Future work

Future work of this project is as follows:

- Change the size of the images given to the network from 256x256 pixels to 128x128 pixels, to avoid excessive padding of null values for most images.
- Download more pictures with higher cloud coverage percentage and investigate that all cloudy images are excluded by the training dataset by computing the s2cloudless cloud mask.
- Restrict the data domain by:
 - Train and generate only satellite images with acquisition time within the growth season of the crop.
- Include other data or exclude data as input data, for example:
 - Exclude the Sentinel 1 VV polarisation, as it seems to contain noise.
 - Exclude the border pixels (i.e. by using spatial erosion), as the border often contains noise from trees or vegetation than the crop on the field.
 - Include the compute Gray-Level Co-Occurrence Matrix (GLCM) based on the Sentinel 1 data.
 - Include the Danish elevation model (DTM).
 - Include the static features as location of the field, image acquisition time.
- Improve the pix2pix training process by:
 - Visualise the input data channels and the generated bands or vegetation indices during the training process.
 - Improve the speed of the training process, such that more epoch can be performed in the training data, i.e. by representing images by uint8 values instead of the current float values.
 - Perform a learning curve analysis to conclude of similar performance can be reached with fewer training samples, thus a faster training.
- Improve model generalisation via image augmentation, for example by:
 - Random crop images.
 - Random flip images.
- Improve the pix2pix generator' abilities by:
 - Perform hyperparameter tuning, for example using grid search.
 - Perform image normalisation, for example by normalizing all images values to a normal distribution.
- Investigate error-difference results, for example to answer the questions:
 - Is the error-difference different for images from different regions of Denmark, or different acquisition times over the year?