

Landbouwschade: Satellietbeelden rekenmodel/QGIS3.4Model

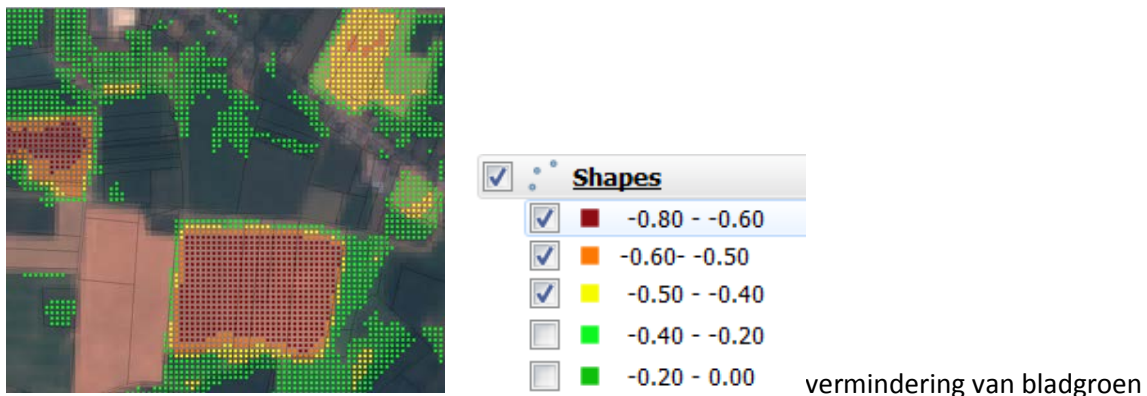
Gewenst resultaat

Als hulpmiddel bij het verwerken van landbouwschade willen we via satellietbeelden berekenen hoeveel schade (verlies aan chlorofyl) er aan het gewas is.



Boven: beeld op noordelijk Tongeren na verwerking Satellietbeelden.

Onder: detail van het beeld, met luchtfoto op de achtergrond en perceelaanduiding.



Tips van Vito (Jurgen Everaerts)

- De grijswaarde die je in een beeld (in dit geval infrarood, dus Sentinel-2 Band 8) vindt, is afhankelijk van heel wat factoren
 - Hoe de zon staat (evolutie van de seizoenen, kijkrichting van de S2 camera)
 - De samenstelling van de atmosfeer (vooral luchtvochtigheid en inversielagen, maar ook pollutie) zorgt ervoor dat het ingestraalde zonlicht verstrooid en geabsorbeerd wordt (tweemaal: één keer van boven naar de grond, dan nog eens vanop de grond naar de satelliet)

- Hiervoor wordt een correctie gemaakt, maar daar zit nogal wat onzekerheid op
- Dat maakt dat het heel moeilijk is om de grijswaarde te relateren aan een fysische grootheid die aan vegetatie gekoppeld is.
- Maar: er zijn oplossingen die minder afhankelijk zijn van de individuele kleurenbanden
 - NDVI: je neemt het verschil in grijswaarde tussen de rode en de infrarode band, en deelt het door de som:
 - $$NDVI = \frac{B08 - B04}{B08 + B04}$$
 - Hierbij gaat men ervan uit dat beide kleurenbanden ongeveer dezelfde residuele fout hebben, die door deze combinatie wordt “weggewerkt”.
 - De NDVI is dus een stabielere maat voor “sterke en goed-gevoede vegetatie”

Nodige data

De satellietbeelden van 2 verschillende dagen.

Per dag de infrarodeband en de rode band (van de grijswaarde) dus de B08 en B04 van de data.


Eventueel


- Gemeentegrens
- Percelenlaag
- Orthofoto als achtergrondlaag

Werkwijze:


QGIS3.2 openen

- Rasters laden
- Eventueel werkzone bepalen (gemeentegrens) >> Extractie op raster
- Raster berekenen via onderstaand model of pythonscript

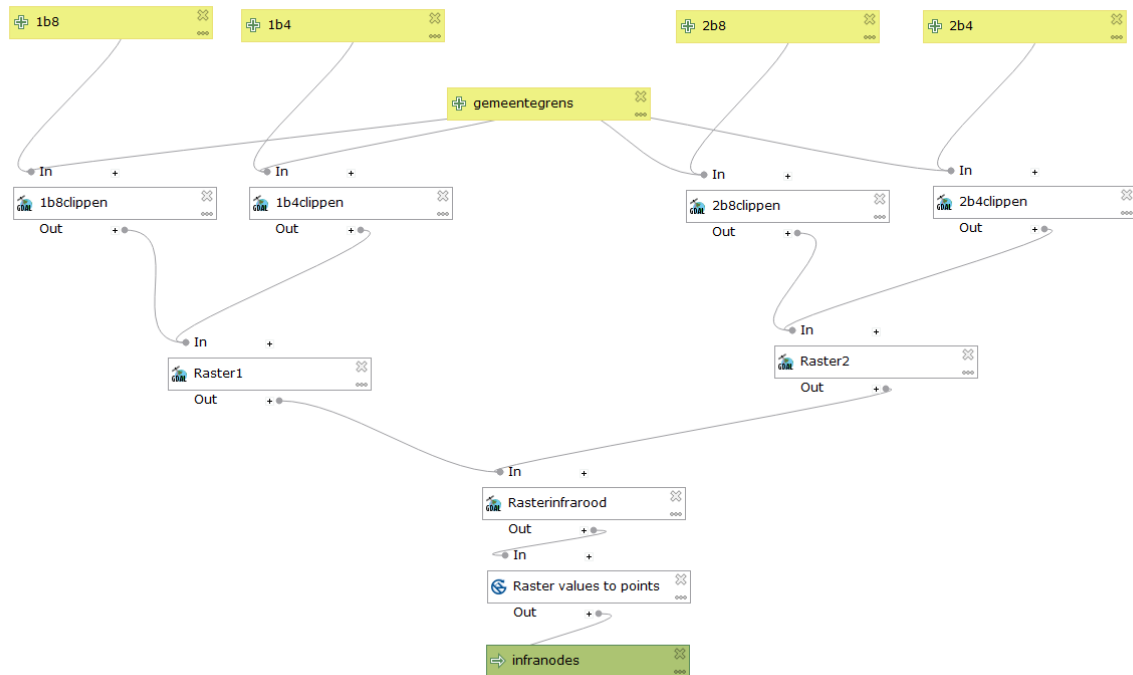
 2xSatellietNaarBladgroenInfo.model3

 2xSatellietNaarBladgroenInfo.py

- Resultaat raster omzetten naar punten >> raster value tot points
- Style laden

 infraroodModel.qml

Het model: 2xSatellietNaarBladgroenInfo.model3



Laat in QGIS3.2 het model 2xSatellietNaarBladgroenInfo.model3 lopen op de satellietbeelden.

Of pythonscript

Ook meegeleverd.

```

1  #clprasterGemeentegrens-name
2  #1b4=raster
3  #1b8=raster
4  #2b4=raster
5  #2b8=raster
6  #gemeentegrens=vector
7  #infranodes=output outputVector
8  results={}
9  outputs['gdal:clprasterbymasklayer_1']=processing.run('gdal:clprasterbymasklayer',
10 { 'ALPHA_BAND': false, 'CROP_TO_CUTLINE': true, 'DATA_TYPE': 5, 'INPUT': parameters['1b4'], 'KEEP_RESOLUTION': false, 'MASK': paramete
rs['gemeentegrens'], 'NODATA': QgsExpression('').evaluate(),
[ 'ALPHA_BAND': false, 'CROP_TO_CUTLINE': true, 'DATA_TYPE': 5, 'INPUT': parameters['1b4'], 'KEEP_RESOLUTION': false, 'MASK': paramete
rs['gemeentegrens'], 'NODATA': QgsExpression('').evaluate() }], context=context, feedback=feedback)
11 outputs['gdal:clprasterbymasklayer_2']=processing.run('gdal:clprasterbymasklayer',
{ 'ALPHA_BAND': false, 'CROP_TO_CUTLINE': true, 'DATA_TYPE': 5, 'INPUT': parameters['1b8'], 'KEEP_RESOLUTION': false, 'MASK': paramete
rs['gemeentegrens'], 'NODATA': QgsExpression('').evaluate(),
[ 'ALPHA_BAND': false, 'CROP_TO_CUTLINE': true, 'DATA_TYPE': 5, 'INPUT': parameters['1b8'], 'KEEP_RESOLUTION': false, 'MASK': paramete
rs['gemeentegrens'], 'NODATA': QgsExpression('').evaluate() }], context=context, feedback=feedback)
12 outputs['gdal:clprasterbymasklayer_3']=processing.run('gdal:clprasterbymasklayer',
{ 'ALPHA_BAND': false, 'CROP_TO_CUTLINE': true, 'DATA_TYPE': 5, 'INPUT': parameters['2b4'], 'KEEP_RESOLUTION': false, 'MASK': paramete
rs['gemeentegrens'], 'NODATA': QgsExpression('').evaluate(),
[ 'ALPHA_BAND': false, 'CROP_TO_CUTLINE': true, 'DATA_TYPE': 5, 'INPUT': parameters['2b4'], 'KEEP_RESOLUTION': false, 'MASK': paramete
rs['gemeentegrens'], 'NODATA': QgsExpression('').evaluate() }], context=context, feedback=feedback)
13 outputs['gdal:rastercalculator_2']=processing.run('gdal:rastercalculator', { 'BAND_A': 1, ['BAND_A': 1], ['BAND_A': 1,
...

[ 'BAND_A': 1, ['BAND_A': 1], ['BAND_A': 1, ['BAND_A': 1], ['BAND_A': 1, ['BAND_A': 1]]]]], 'FORMULA': a-
b, 'INPUT_A': outputs['gdal:rastercalculator_2']['OUTPUT'], 'INPUT_B': outputs['gdal:rastercalculator_1']
['OUTPUT']]]], 'NO_DATA': QgsExpression('').evaluate(), 'RTYPE': 5, context=context, feedback=feedback)
16 outputs['saga:rastervaluestopoints_1']=processing.run('saga:rastervaluestopoints',
{ 'GRIDS': outputs['gdal:rastercalculator_3']['OUTPUT'], 'NODATA': true, 'GRIDS': outputs['gdal:rastercalculator_3']
['OUTPUT'], 'NODATA': true, 'TYPE': 0, context=context, feedback=feedback)
17 results['infranodes']=outputs['saga:rastervaluestopoints_1']['SHAPES']
18 return results

```

resultaat

