Drone to Satellite V1.1

A model for upscaling of UAV imagery

User's manual

Martijn Romar, 2020

Content

Introduction

This model was designed to classify drone or unmanned aerial vehicle (UAV) imagery and using the classified image to classify external data such as satellite or aerial imagery. UAV's are able to produce high resolution imagery at a cost far lower than traditional ways of capturing data and are therefore quickly gaining popularity. The drawback of UAV's is that they are only able to cover so much area before their batteries run out. This makes acquiring imagery of large areas difficult and time consuming, and thus expensive. Therefore it can be interesting to use both UAV and external data; the UAV's provide the resolution, the satellite the coverage. However, a problem of the external data is that features visible on UAV imagery are sometimes not clearly visible on satellite or aerial imagery. Using a classified UAV image to classify the external data, this model tries to solve this problem because the computer can spot differences that the eye cannot.

This document is the manual accompanying the model and will discuss how it works and how to use it. It has already been used to classify UAV imagery in the Oostvaardersplassen in the Netherlands (Romar, unpublished work), but has been designed to be easily adopted to different areas of interest. I hope you find the model useful, and are able to use its output.

The model supplied comes with some demo data of the Oostvaardersplassen that will help you to understand the process and demonstrate what the model exactly does. It must be noted that the model requires a lot of processing power and should preferably be run at a beefy computer.

Martijn Romar, 2-3-2020

For questions, comments or feedback,

Please contact me at:

martijn.romar@student.uva.nl

How does the model work?

Basic concept

The basic concept of the model is that it classifies UAV imagery using training samples specified by the user, and using the classified UAV imagery as training samples to classify the external data. Although the model is presented here as a single entity, it consists of 4 parts: the first part to clean up the working environment ('Initializer'), the second to classify the UAV imagery and assessing the quality of the classification ('create training samples from imagery'), the third to merge the separate classifications ('Merger') and the fourth part to classify the external data (separate tools). Due to technical limitations of ArcGIS it was only possible to construct the model this way, using a models in a model.

The user provides the following inputs:

- The UAV imagery. Each filename must start with 'Input', otherwise the model will not recognize it when it iterates through the database.
- The external data.
- Training samples. These are classified areas (polygons) on the UAV imagery that the model uses to classify the imagery.
- Control samples. Same as the training samples, but other areas. The model uses these to determine the accuracy of each classification and thus checking if the classified UAV image is accurate enough to use for the classification of the external data.

The model iterates through all of the UAV images, classifies them and after accuracy control they are merged into one feature, so that all UAV images are processed separately. There are about 3 distinct phases in the model

Phase 1: UAV classification

1. The model selects the training samples that cover the UAV image and saves a copy of these features.
2. If the UAV imagery is very high resolution, the classification is not useful to apply on satellite imagery, if this is the case, the model resamples the UAV image.
3. These copied training samples and the resampled UAV image are used to train the Support Vector Machine classifier.
4. The model classifies the resampled UAV image using the .ecd file generated at step 3.

Phase 2: UAV accuracy assessment

1. The model selects the control samples that cover the UAV image and saves a copy of these features.
2. Using the control samples, the model generates 500 random accuracy assessment points.
3. The model updates the accuracy assessment points with the classified image from phase 1, step 4.
4. The model computes a confusion matrix, and thus giving an impression of accuracy by calculating k^ values.
5. If the resulting k^ is higher[1] than the value the user specified, the model deems the classification accurate enough to use in phase 3.

Phase 3: Classification of External data

1. Merge all the classified UAV images deemed good enough

2. Add classes so ArcGIS can use it as training samples
3. Using the created training samples and the external data, it trains the Support Vector Machine classifier
4. The external data is classified and the end result is produced.

The next steps are expected to be introduced in a newer version and are currently NOT included in the model.

5. The model uses the training samples generated at phase 3, step 2 to create accuracy assessment points as ground truth.
6. The model computes a confusion matrix and display a message if the k^ value is too low.

The problem with steps 5 and 6 is that the training samples that were used to classify the model are also used to create the confusion matrix. A possible solution is to select half of the samples generated at phase 3, step 2 and use one half as training samples and the other halve as control samples. This will be fixed in the next version.

Accuracy control and k^ values

The model uses k^ or khat values to determine the accuracy of the classification. This is based on information given in the ESRI ArcGIS pro course: Performing Accuracy Assessment for Image Classification (2019). K^ is a value representing a level of agreement, and a strong agreement means that resulting classification can be used for further use and/ or decision making.

The levels of agreement according to ESRI(2019):

Level of agreement	K^ or Khat value
Poor	< 0.4
Medium	0.4-0.8
Strong	> 0.8

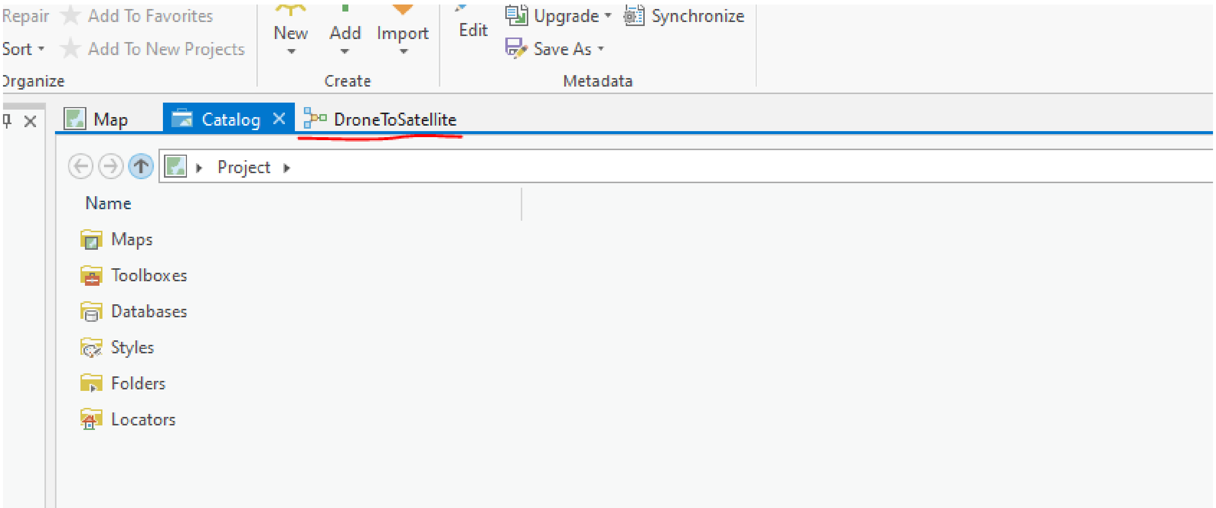
If the model disqualifies all classified UAV images, the k^ value specified by the user is too high and should be lowered.

How to use the model

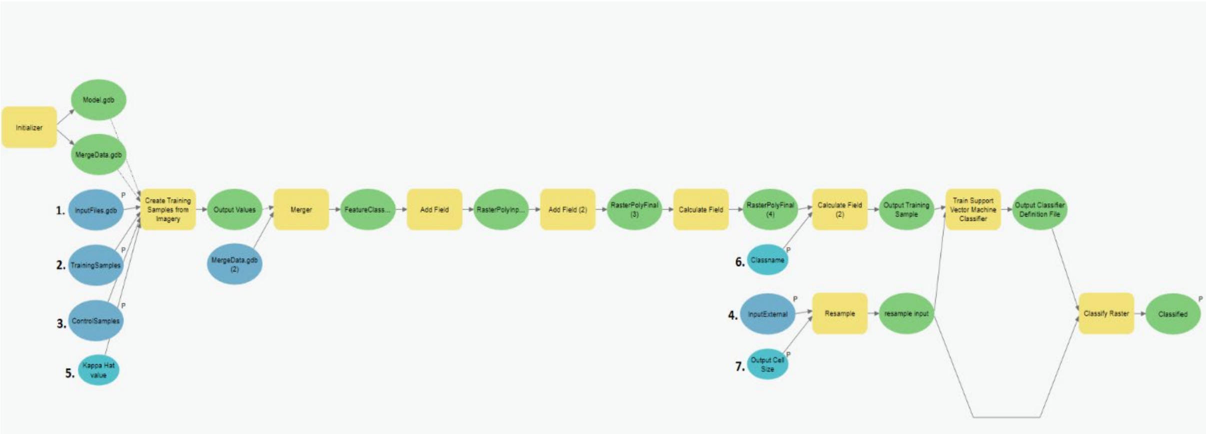
The model is designed to be user-friendly and requiring minimal meddling of the user. The model creates and handles its workspace itself, and the only requiring parameters of the user is are the following (the numbers correspond with the numbers in the figure):

1. **The UAV images**. The files of these images must put in the 'input' database and the filenames must start with 'Input' for the model to be recognized.
2. **Training samples.** These are user defined polygons that contain a ground truth for the model to base the classification on. These samples must cover ***all*** of the images, the model selects the required polygons for each image. These features must contain a column with 'classname' and 'classvalue'.
3. **Control samples.** These are technical the same as the training samples meaning that they are user defined ground truth, but must ***not*** cover the training samples as the model uses the control samples to check whether the classification is correct or not
4. **External data.** The aerial or satellite data that needs to be classified.
5. **Kappa hat value.** This is the kappa hat value that will be used to determine which classifications are fit to use and which are not. By default, it is set at 0.6. You can change it by simply changing the value in the code block.
6. **Classname.** The code block contains a code that is necessary to create the training sample from the merged classification. Due to limitations of the merge function, the resulted feature class only keeps the 'classvalue'. In this codeblock the user specifies which value corresponds with what classification. If more classes are used, simply add more elif statements like the one already present in the model.
7. **Resample cell size.** This is the cell size to which the input data set is resampled if the external has a too large resolution for acceptable processing time.

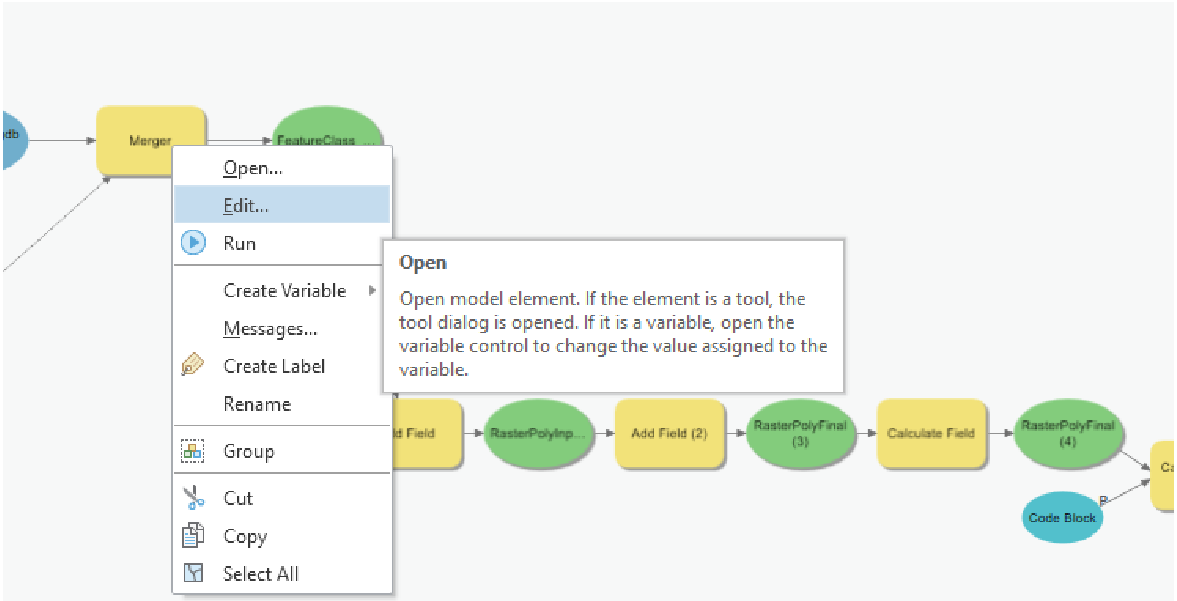
When opening the .APRX file, you can see that the 'DroneToSatellite' model is already opened. Due to limitations of ArcGIS it is only possible to run it in edit mode instead of a normal geoprocessing tool:



When clicking the DroneToSatellite tab you are greeted by the model and see this:

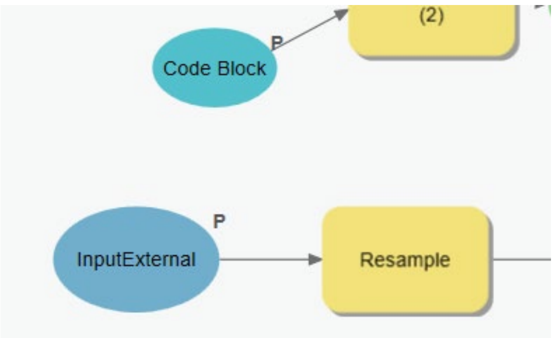


This is the model which consists of the 4 parts: three sub models and the final classification. If you are interested in how the sub models look like you can right click on it and select ‘edit’.



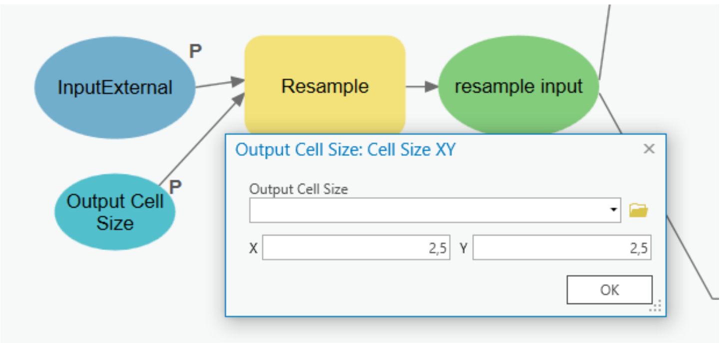
A new tab will open and you can see the sub model.

But, back to the model. You can see the processes being yellow, the outputs being green and the inputs being blue. Perhaps you have also noticed some inputs and output having a small ‘P’ on the top right side:



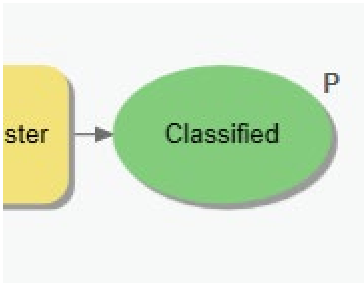
These ‘P’s are the parameters mentioned in the begin of this chapter. Please provide the required inputs as mentioned in the begin and the model should work like a charm. You can do this by right-clicking them and selecting open, which will open a small screen where you can provide the input.

Do note that some types of aerial or satellite data can have very high resolutions, and that some form of resampling is inevitable to reduce process time. You can provide the desired cell size in the light blue oval:



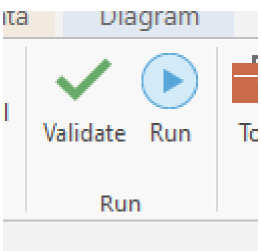
I used a factor 10. The data supplied comes in resolution of 25x25cm, and I resample it to 2.5x2.5m. You are free to use whatever factor you want, but keep in mind that there is a balance between image quality and processing time.

The only exception that was not mentioned in the beginning was the green ‘classified’ variable. This is the end result; the classified external data:



When using more than one external data file to classify e.g. satellite imagery over time, remember to rename the classified images accordingly, as the model will overwrite it if the name is not changed.

When all the inputs are put in and the parameters are set, click on the ‘validate’ button before pressing the ‘run’ button:



The model comes supplied with some demo data in the form of 2 UAV images and a small section of a satellite image, as well as training and control samples. You can run it and see what it does. The classification is not perfect though, as it uses only a fifth of the original UAV data.