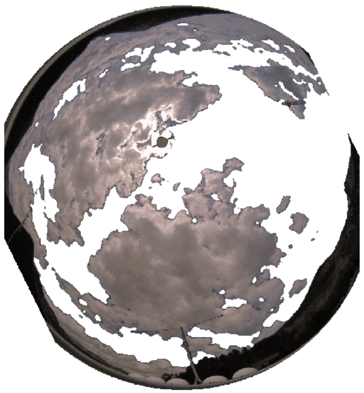
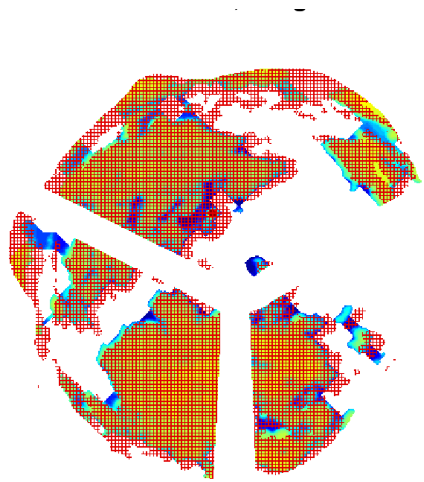
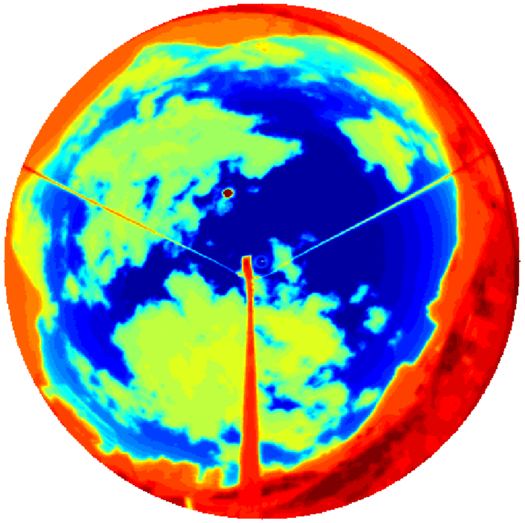
pmod wrc

Improvement of a Cloud Detection Algorithm based on All-Sky Infrared Images

Swiss Civilian Service

Author: Luc Schnell

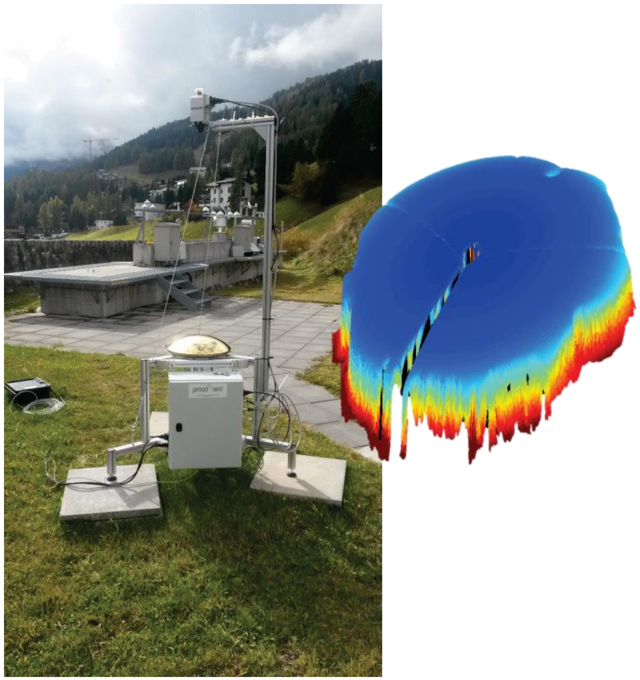
Supervisor: Dr. Julian Gröbner

Overview

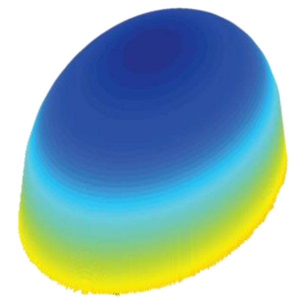
- Original Program
  - 1.1 Basic Ideas
  - 1.2 New Scripts
  - 1.3 Results
- Genetic Algorithm
  - 2.1 Basic Ideas
  - 2.2 Scripts
  - 2.3 Results
- Supervised Machine Learning
  - 3.1 Basic Ideas
  - 3.2 Scripts
  - 3.3 Results

1. Original Program

1.1 Basic Ideas



compare



$$T_B = (T_{65} - a)\left(\frac{\Theta}{65}\right)^b + a$$

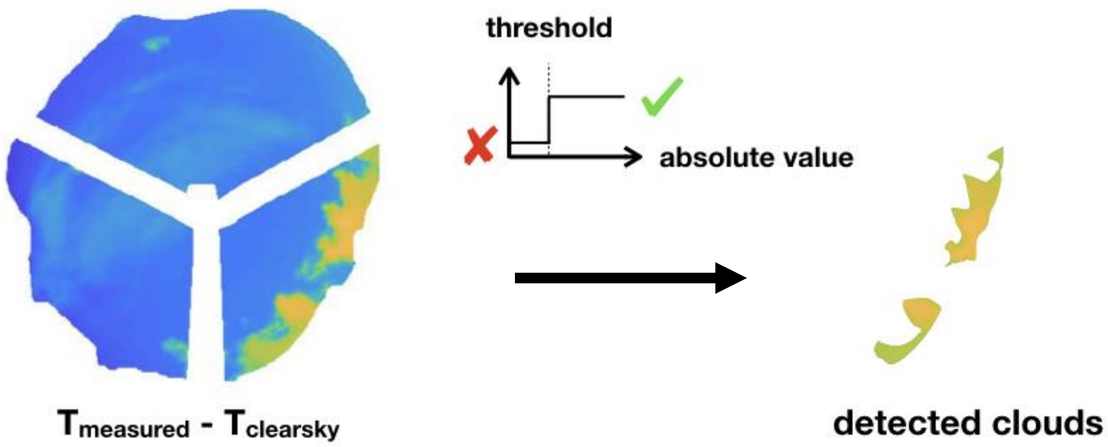
Experimental IRCCAM radiance data

Theoretical clear-sky radiance model

Sources:
[1] Gröbner et al. (2016)
[2] Aebi et al. (2018)

1.1 Basic Ideas

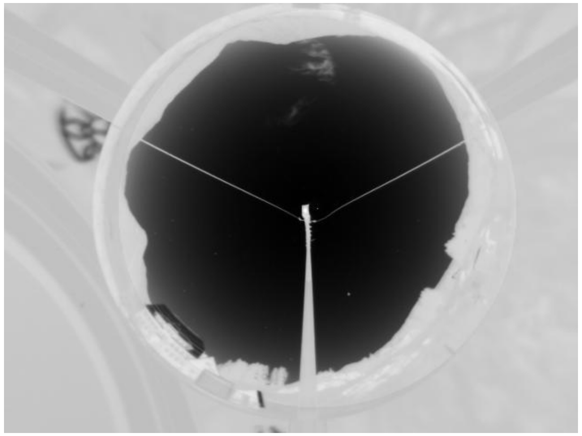
- Detects clouds based on the absolute difference between the measured radiance and the theoretically calculated clear-sky radiance.



1.2 New Script: Mask Generator

- The horizon mask can either be generated based on an edited image (default) or based on IRCCAM data taken on a clear day.

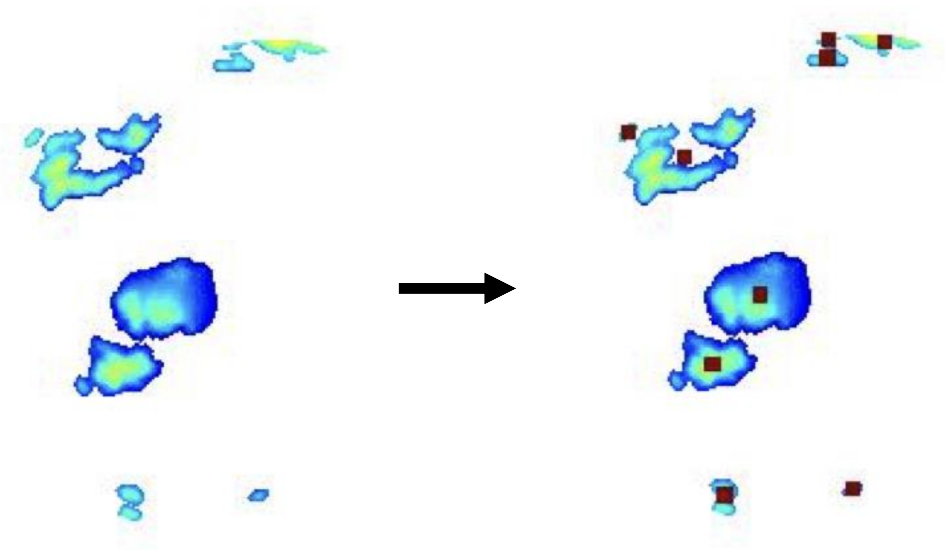
Edited image



Calibrated IRCCAM image

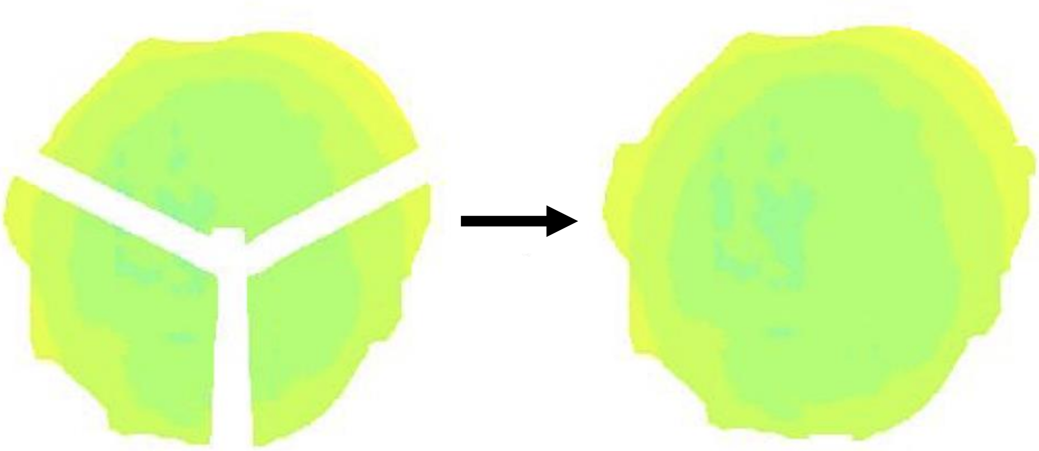
1.2 New Script: Clustering

- Finds individual cloud clusters (connected pixel regions that were classified as clouds).



1.2 New Script: Interpolate

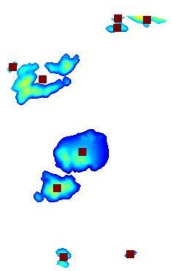
- Fills in missing regions on IRCCAM images.
- This function is based on MATLAB’s **regionfill** that numerically solves the Dirichlet boundary problem.

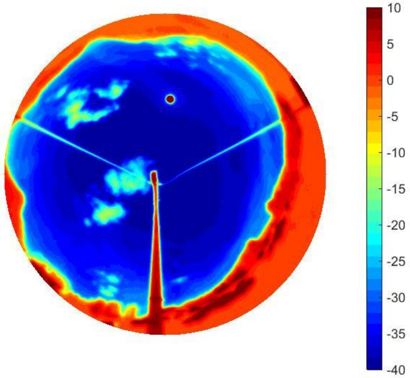


Script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Basic_Program\interpolate.m

1.3 Results

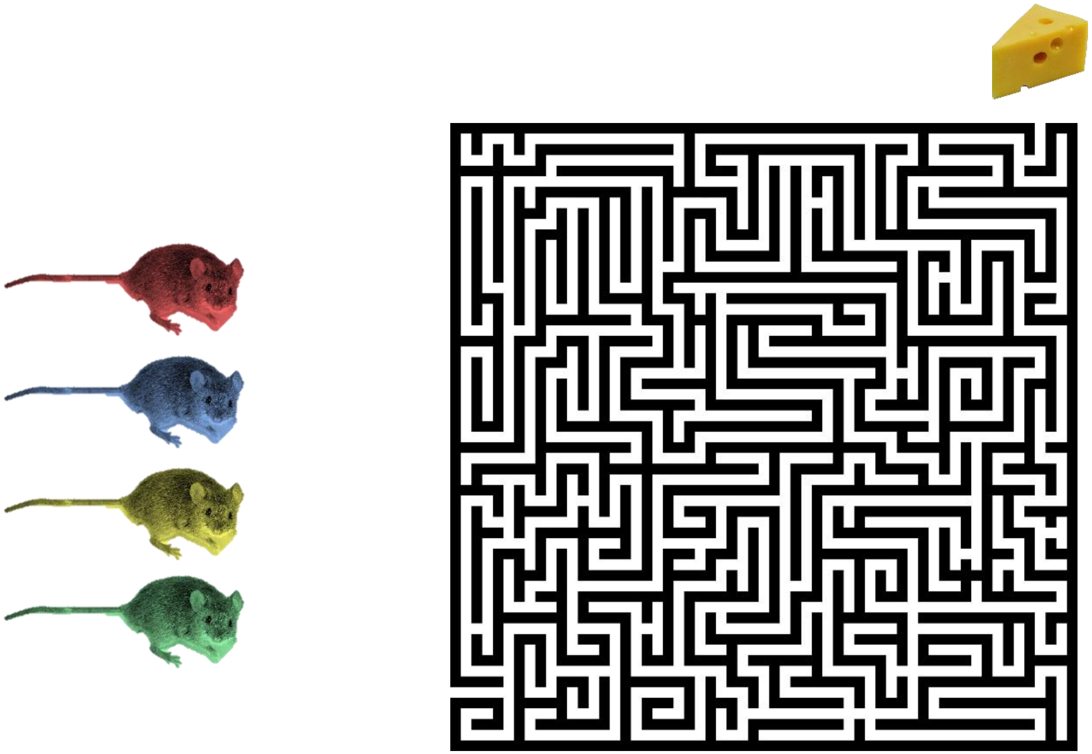
- The algorithm detects thick clouds very well.
- The detection of thin clouds, however, often poses a problem.



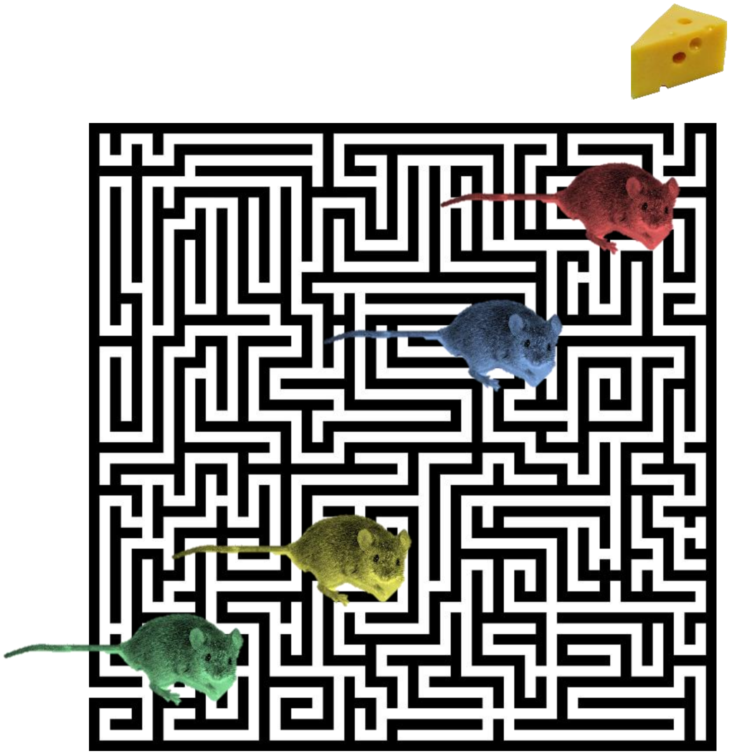


2. Genetic Algorithm

2.1 Basic Ideas

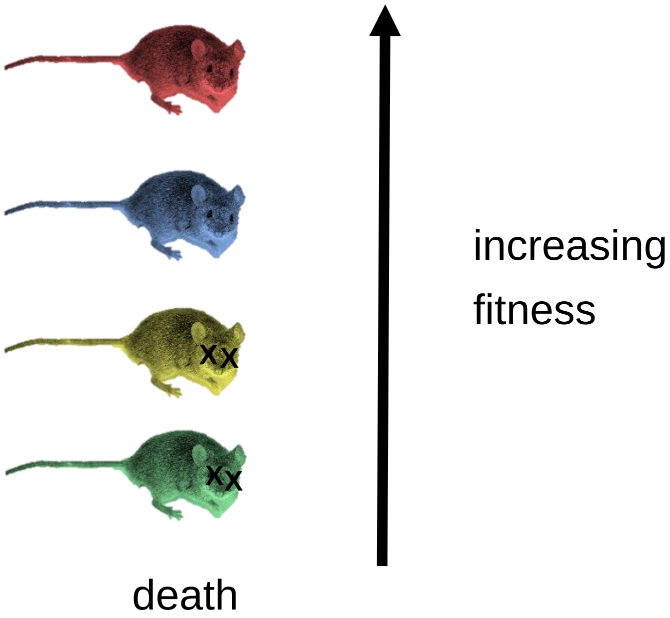


2.1 Basic Ideas

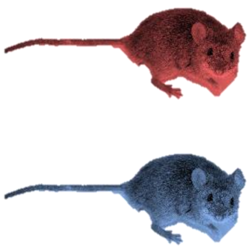


2.1 Basic Ideas

2.1 Basic Ideas



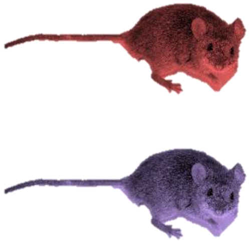
2.1 Basic Ideas



2.1 Basic Ideas

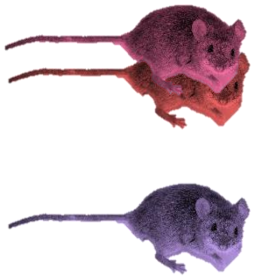
sexual
reproduction

2.1 Basic Ideas



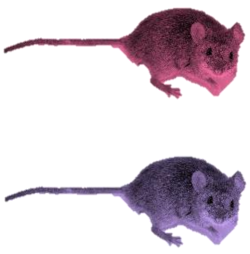
If the child is fitter than one of its parents, the parent is replaced by the child.

2.1 Basic Ideas



mutation

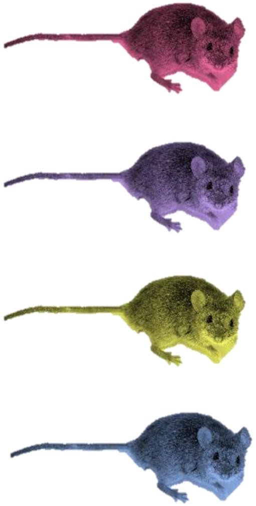
2.1 Basic Ideas



If the mutation is fitter than the original, the original is replaced by the mutation.

mutation

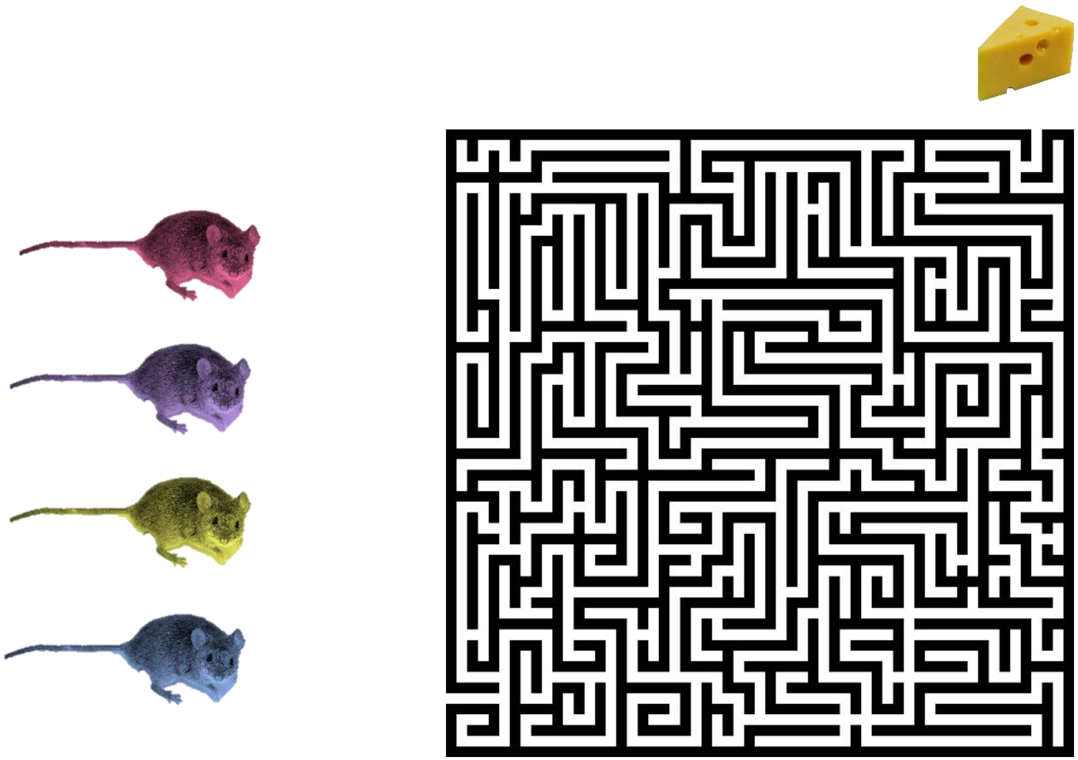
2.1 Basic Ideas



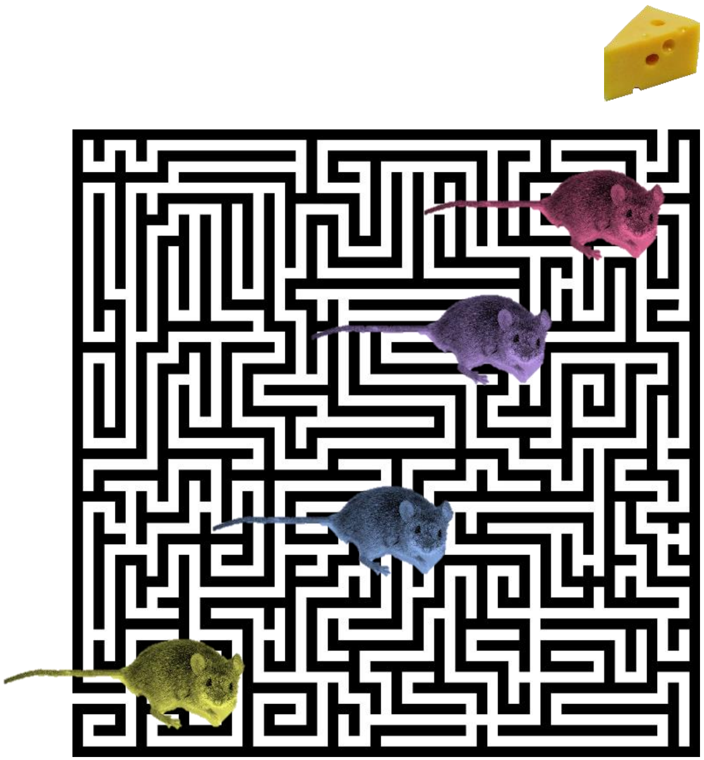
New genes are added to the gene pool (this prevents the algorithm from getting stuck in a local optimum).

new gene material
(randomly generated)

2.1 Basic Ideas



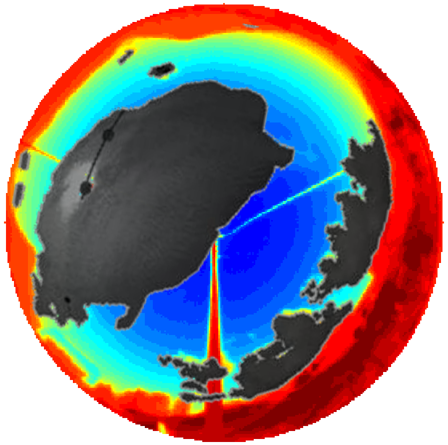
2.1 Basic Ideas

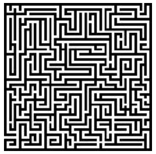


2.1 Basic Ideas

2.2 Script: Goal

- The goal is to detect all clouds on IRCCAM images.





corresponds to
the maze

2.2 Script: Model

2.2 Script: Model

model:
- parameters
- eval_training
- fitness
- info

corresponds to the mouse

2.2 Script: Model

model:
- parameters
- eval_training
- fitness
- info

corresponds to the mouse

key	**value**
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

2.2 Script: Model

model:
- parameters
- eval_training
- fitness
- info

corresponds to the mouse

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

model_parameter:
- value
- max_value
- min_value
- integer

corresponds to the color

Class script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\model_parameter.m

2.2 Script: Model

model:
- parameters
- eval_training
- fitness
- info

corresponds to the mouse

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

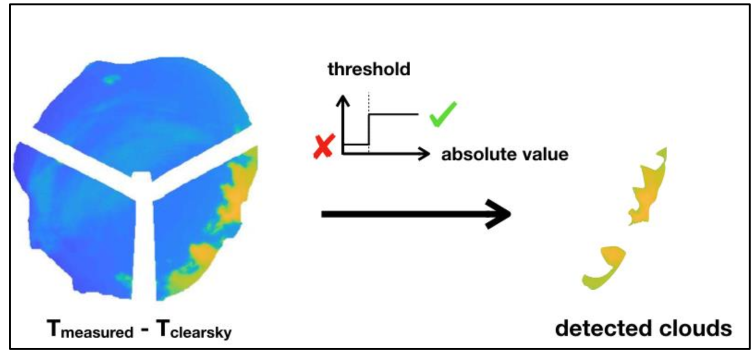
Class script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\model.m

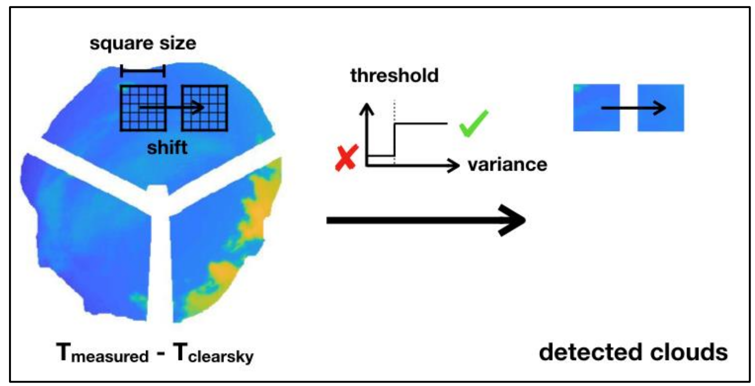
2.2 Script: Model

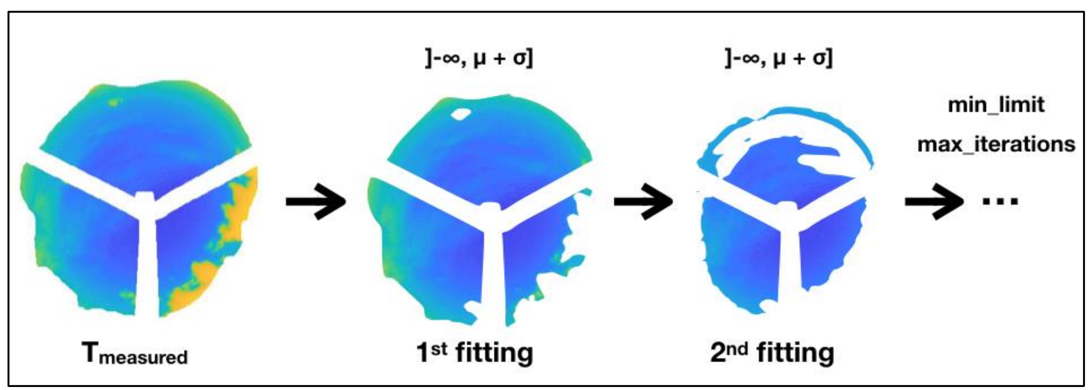
model:
- parameters
- eval_training
- fitness
- info

corresponds to the mouse

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…







2.2 Script: Model

model:
- parameters
- eval_training
- fitness
- info


corresponds to the mouse

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

Class script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\model.m

2.2 Script: Initialization

2.2 Script: Initialization

model 1:
- parameters
- eval_training
- fitness
- info

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

the values are randomly assigned in **min_value ≤ value ≤ max_value**

2.2 Script: Initialization

model 1:
- parameters
- eval_training
- fitness
- info

model 2:
- parameters
- eval_training
- fitness
- info

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

Script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\initialization.m

pmod wrc

2.2 Script: Initialization

model 1:
- parameters
- eval_training
- fitness
- info

model 2:
- parameters
- eval_training
- fitness
- info

model 3:
- parameters
- eval_training
- fitness
- info

Script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\initialization.m

2.2 Script: Initialization

model 1:
- parameters
- eval_training
- fitness
- info

model 2:
- parameters
- eval_training
- fitness
- info

model 3:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

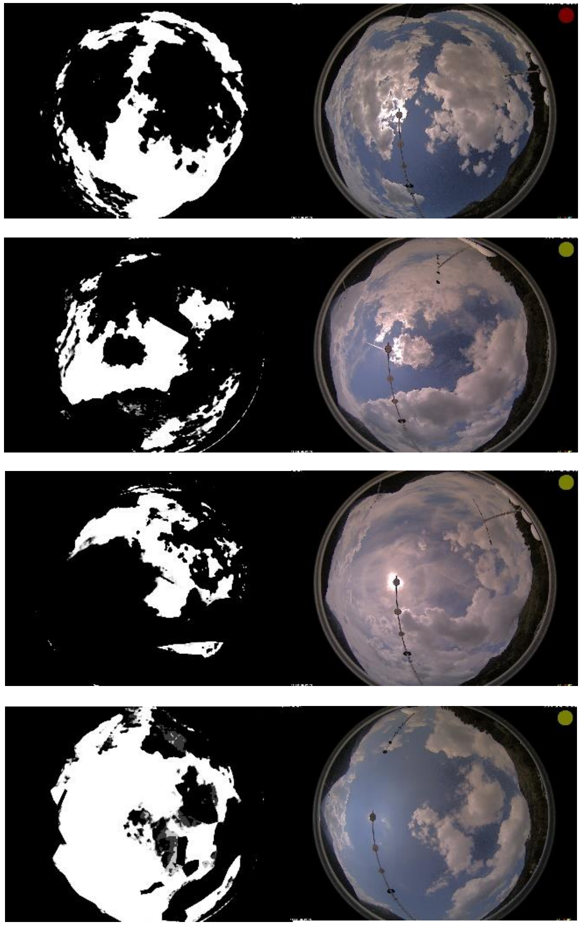
2.2 Script: Fitness

model 1:
- parameters
- eval_training
- fitness
- info

model 2:
- parameters
- eval_training
- fitness
- info

model 3:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info



Training images located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\IRCCAM_4996\data\training

2.2 Script: Prepare Images

model 1:
- parameters
- eval_training
- fitness
- info

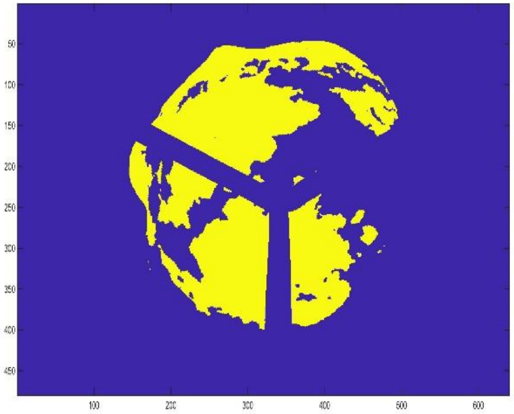
model 2:
- parameters
- eval_training
- fitness
- info

model 3:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

Calibration:

- IRCCAM angle calibration
- Position of the sun



2.2 Script: Fitness

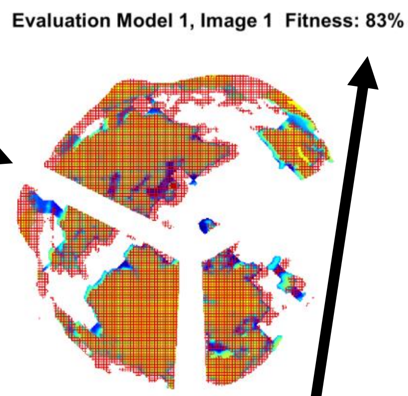
model 1:
- parameters
- eval_training
- fitness
- info

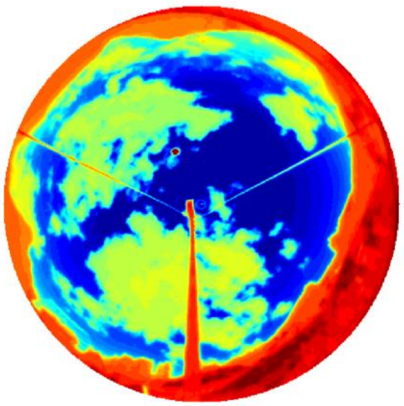
model 2:
- parameters
- eval_training
- fitness
- info

model 3:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

model evaluation





Fitness: the fraction of pixels for which the model evaluation agrees with the training data.

2.2 Script: Fitness

model 1:
- parameters
- eval_training
- fitness
- info

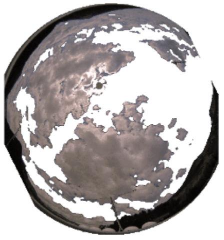
model 2:
- parameters
- eval_training
- fitness
- info

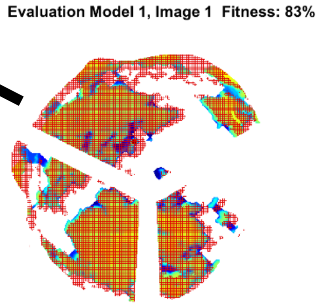
model 3:
- parameters
- eval_training
- fitness
- info

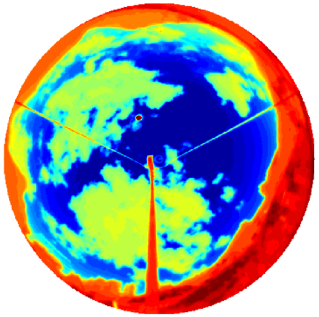
model 4:
- parameters
- eval_training
- fitness
- info

fitness saved

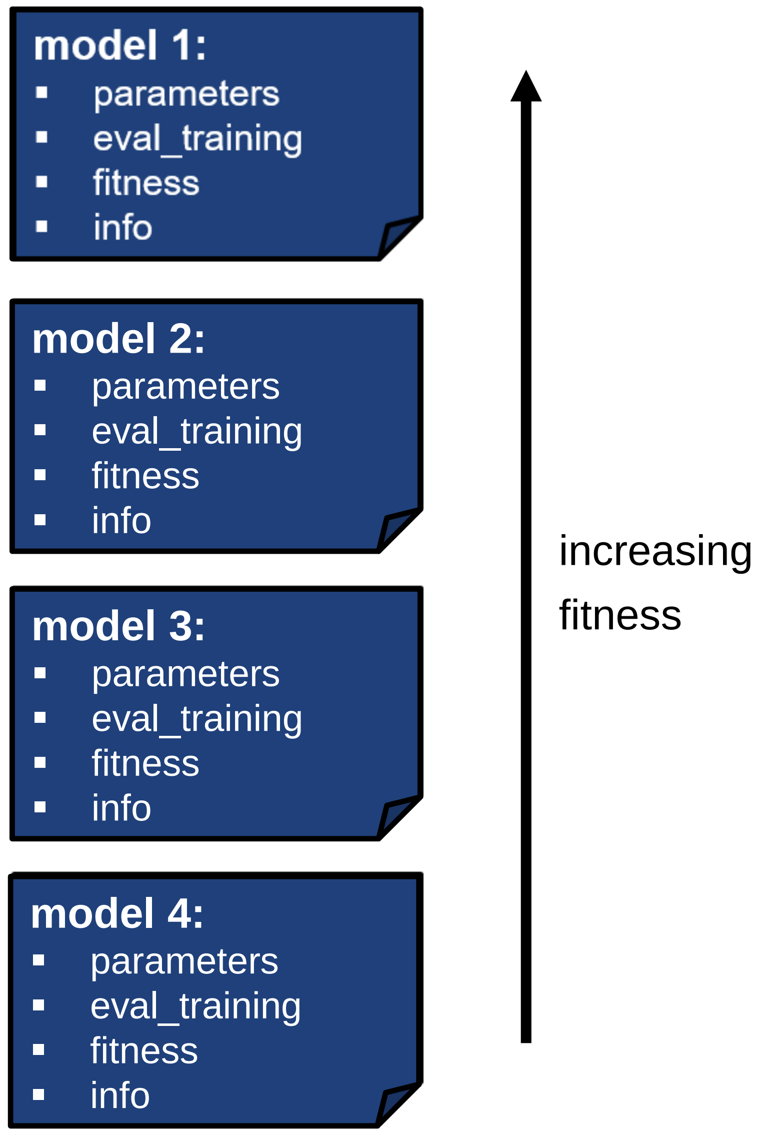
evaluation saved







2.2 Script: Fitness



2.2 Script: Death

model 1:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

model 3:
- parameters
- eval_training
- fitness
- info

model 2:
- parameters
- eval_training
- fitness
- info

2.2 Script: Death

model 1:

- parameters
- eval_training
- fitness
- info

model 4:

- parameters
- eval_training
- fitness
- info

2.2 Script: Sexual Reproduction

model 1:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

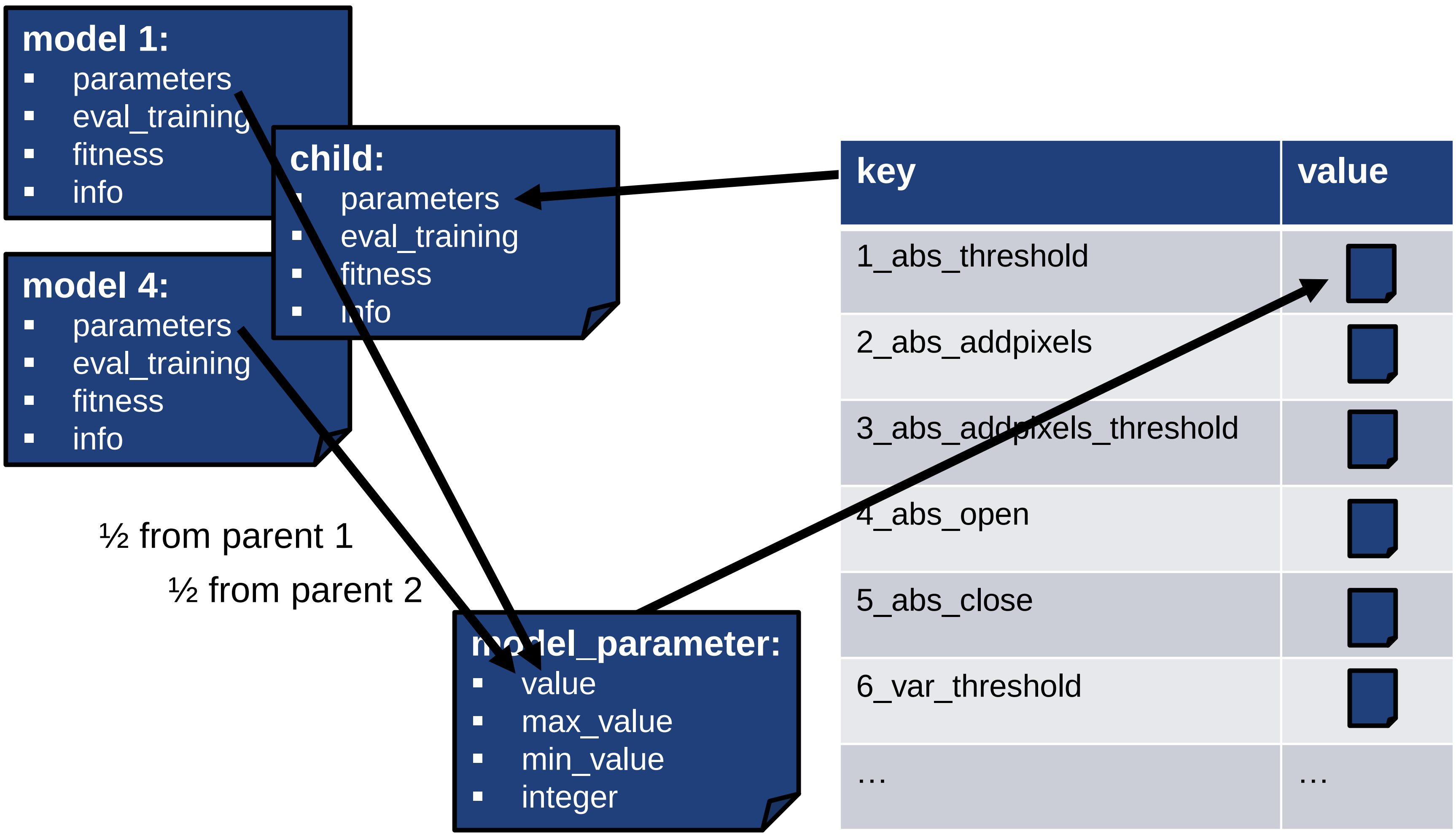
2.2 Script: Sexual Reproduction

model 1:
- parameters
- eval_training
- fitness
- info

child:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

2.2 Script: Sexual Reproduction



2.2 Script: Sexual Reproduction

model 1:
- parameters
- eval_training
- fitness
- info

child:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

Script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\reproduction.m

2.2 Script: Mutation

child:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

2.2 Script: Mutation

child mutation:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

2.2 Script: Mutation

child mutation:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

key	value
1_abs_threshold	
2_abs_addpixels	
3_abs_addpixels_threshold	
4_abs_open	
5_abs_close	
6_var_threshold	
…	…

model_parameter:
- value
- max_value
- min_value
- integer

One parameter value is changed randomly, the others are simply copied.

2.2 Script: Mutation

child mutation:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

2.2 Script: Mutation

child mutation:
- parameters
- eval_training
- fitness
- info

model 4:
- parameters
- eval_training
- fitness
- info

2.2 Script: Random Generation

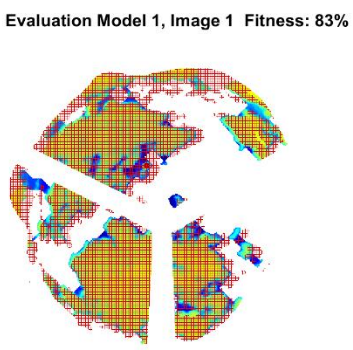
child mutation:
- parameters
- eval_training
- fitness
- info

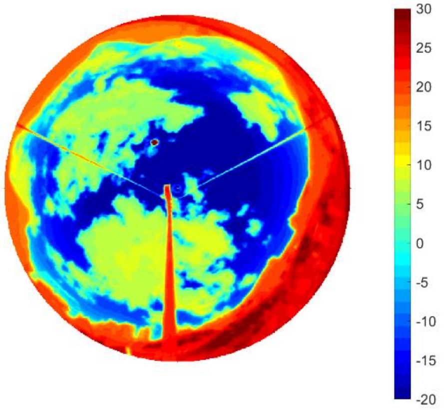
model 4:
- parameters
- eval_training
- fitness
- info

random model:
- parameters
- eval_training
- fitness
- info

random model:
- parameters
- eval_training
- fitness
- info

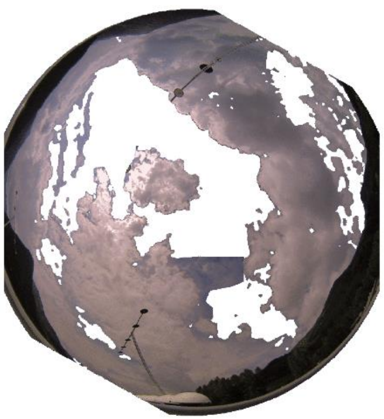
2.3 Results

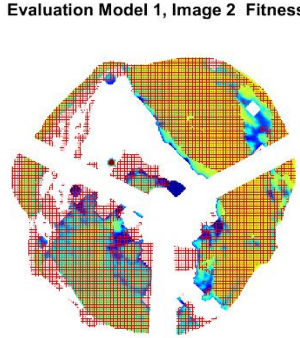


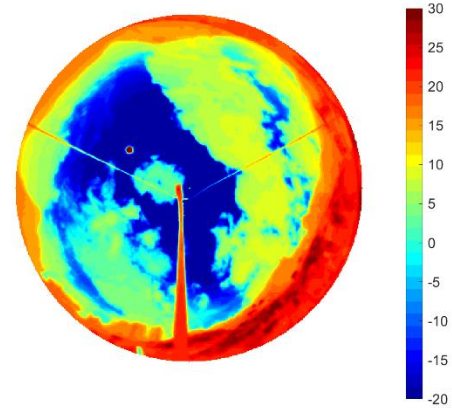


Run genetic algorithm with \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\run.m

2.3 Results



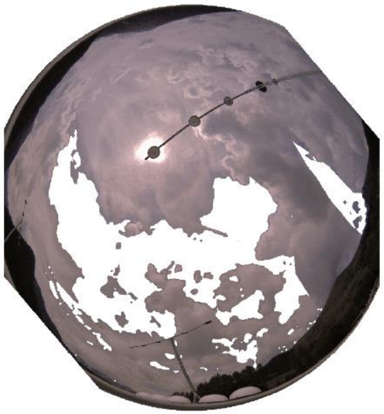


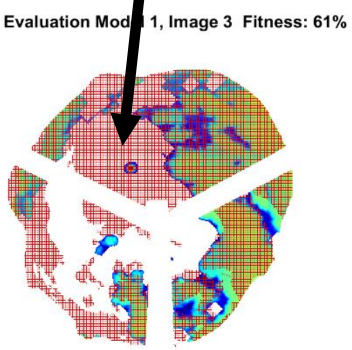


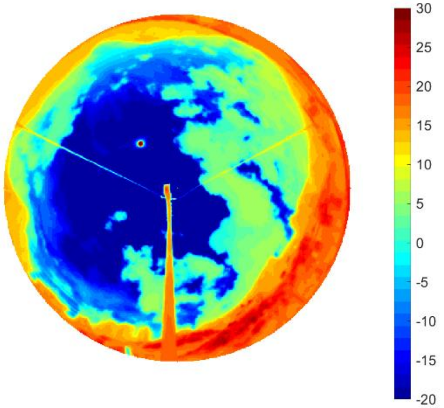
Run genetic algorithm with \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\run.m

2.3 Results

The thin clouds are not detected.







Run genetic algorithm with \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\run.m

2.3 Results

- Advantage: the genetic algorithm optimizes the model in a very general manner – one can easily add new parameters.
- Disadvantages: the training process is slow and cannot handle «delicate» parameters very well.

- The algorithm still has difficulties detecting thin clouds.

Run genetic algorithm with \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Genetic_Algorithm\run.m

3. Supervised Machine Learning

2.1 Basic Ideas

Features:

- Number of emails you have already received from that person
- Number of flagged words the email contains (e.g. «subscribe», «buy», «king of Libya», «Elvis' grandson», «Ghandi's sister», etc.)

$\Big\}\ x_i$

Predicted label: $$\hat{y}(\vec{w}, \vec{x}) = \sum_{i=1}^{d} \boldsymbol{w_i} x_i \qquad \hat{y} \in [0,1]$$

Spam

Ham

2.1 Basic Ideas

$\vec{x}^1, y^1 = 1$

$\vec{x}^2, y^2 = 0$

$\vec{x}^3, y^3 = 0$

$\vec{x}^4, y^4 = 1$

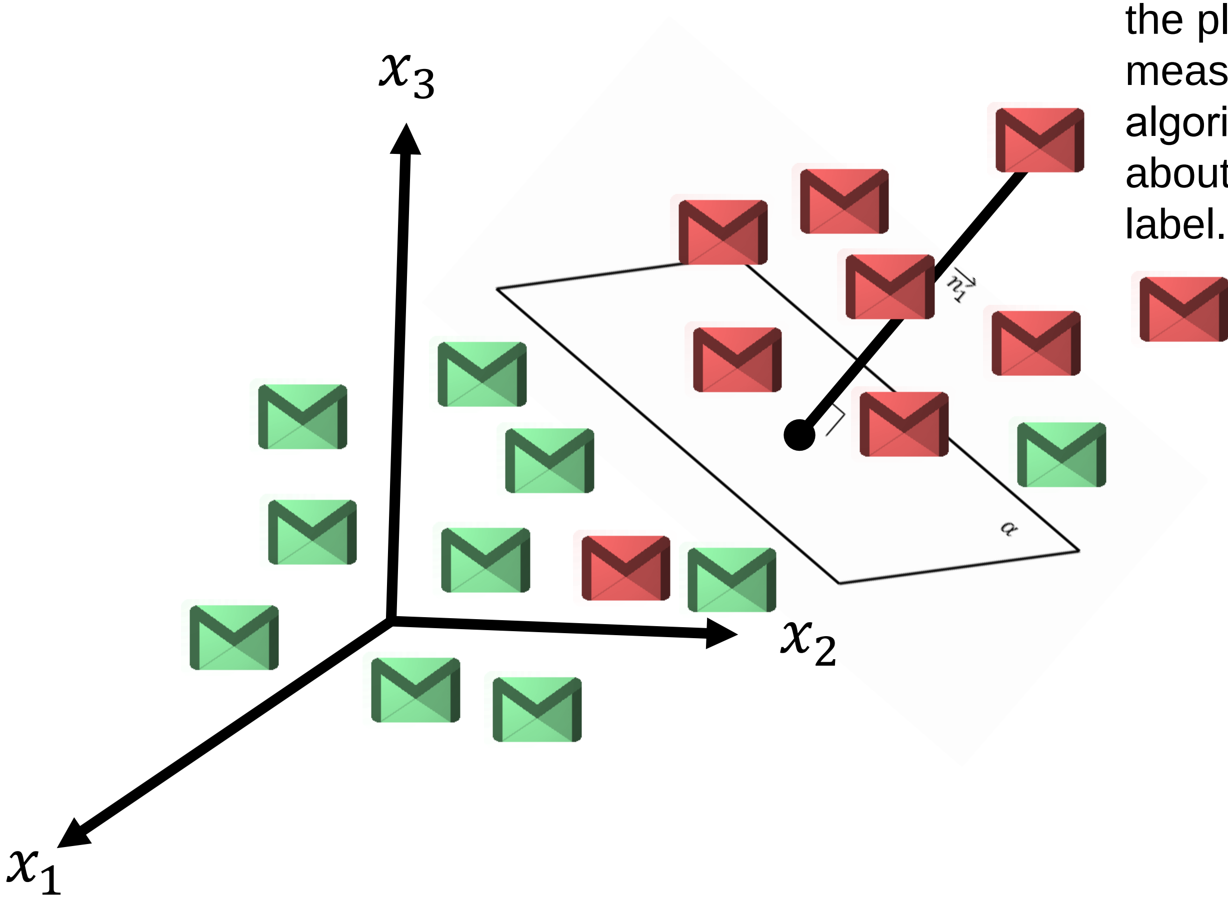
$\vec{x}^5, y^5 = 0$

$$n = 5$$

$$\hat{y}(\vec{w}, \vec{x}^j) = \sum_{i=1}^{d} \boldsymbol{w_i} \, x_i^j \qquad j \in \{0, \dots, n\}$$

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} \sum_{j=1}^{n} \mathrm{l}\left(y^j, \hat{y}(\vec{w}, \vec{x}^j)\right)$$

2.1 Basic Ideas



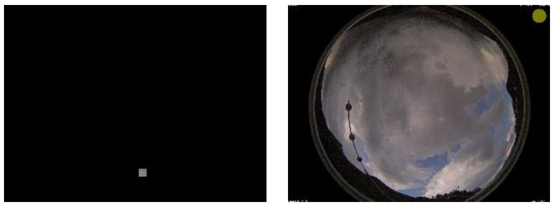
The distance from the plane gives a measure of the algorithm's certainty about a predicted label.

2.2 Script: Goal

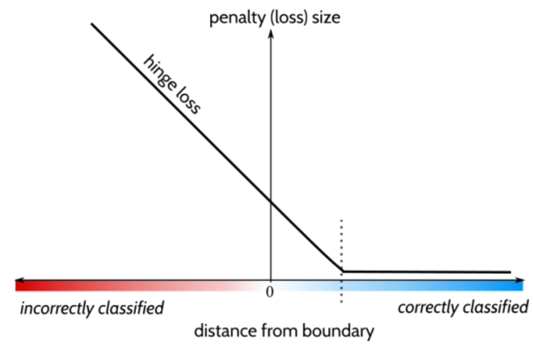
- The goal is to be able to classify an IRCCAM pixel square as cloud or clear sky.

Is it a cloud or not a cloud,
that is the question.

2.2 Script: Training



Training images with known label y_i (cloud / no cloud).



$$\begin{pmatrix} (0,0) & (0,1) & (0,2) & & (0,n) \\ (1,0) & (1,1) & (1,2) & \cdots & (1,n) \\ (2,0) & (2,1) & (2,2) & & (2,n) \\ & \vdots & & \ddots & \vdots \\ (n,0) & (n,1) & (n,2) & \cdots & (n,n) \end{pmatrix}$$

GLCM matrix

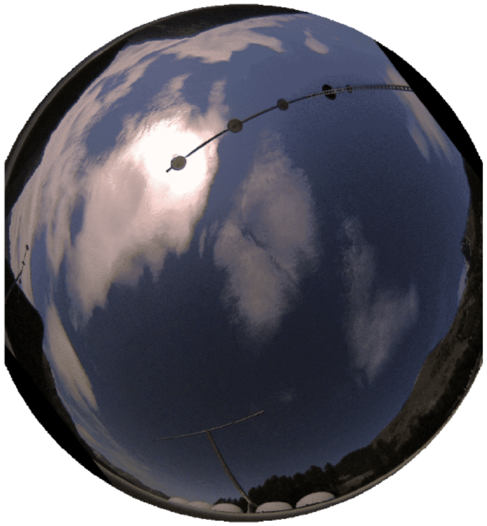
- Contrast
- Correlation
- Energy
- Homogeneity

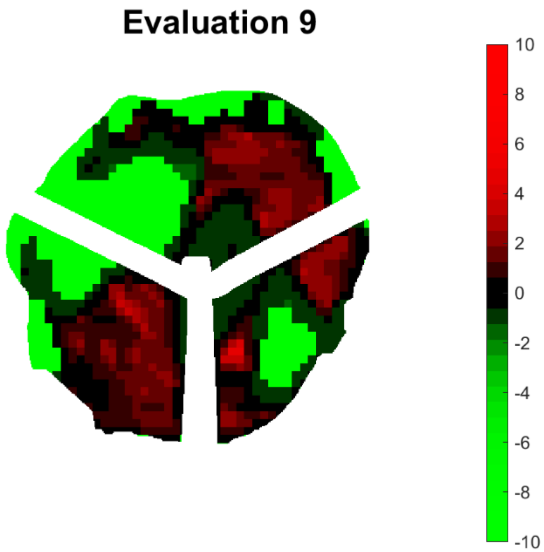
These are the x_i.

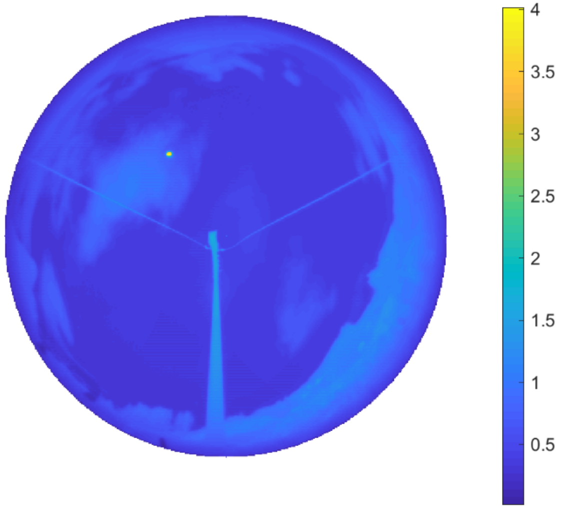
$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} \sum_{j=1}^{n} l\left(y^j, \hat{y}\left(\vec{w}, \vec{x}^j\right)\right)$$

Training script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Machine_Learning\training.m
Idea based on [3] Luo et al. (2018)

2.3 Results

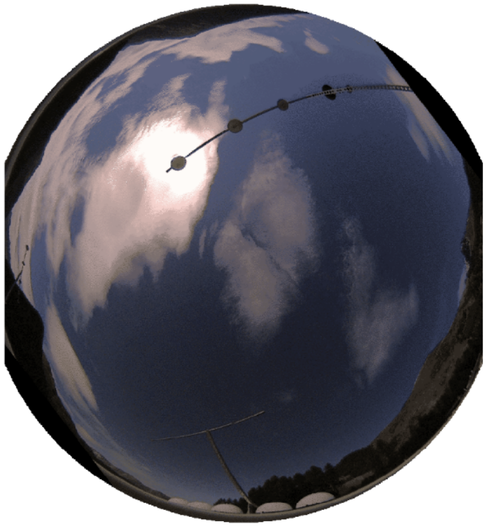


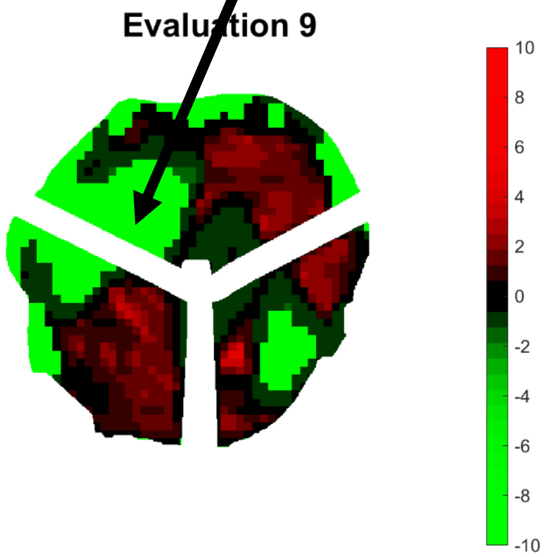


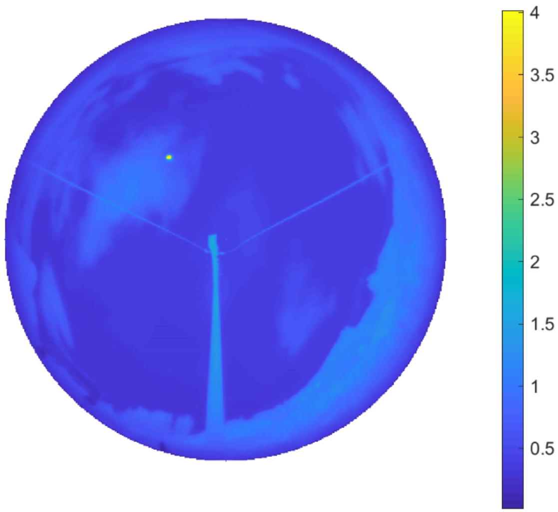


2.3 Results

The light-green regions are the clouds detected by the original program.

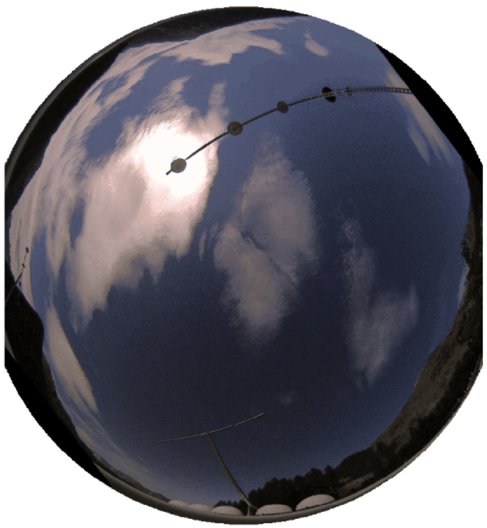


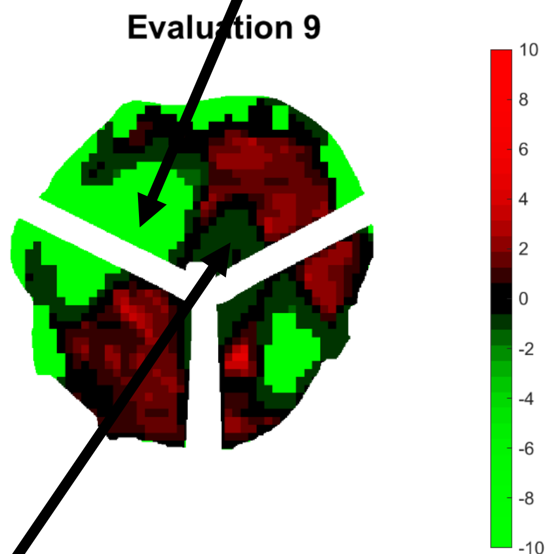


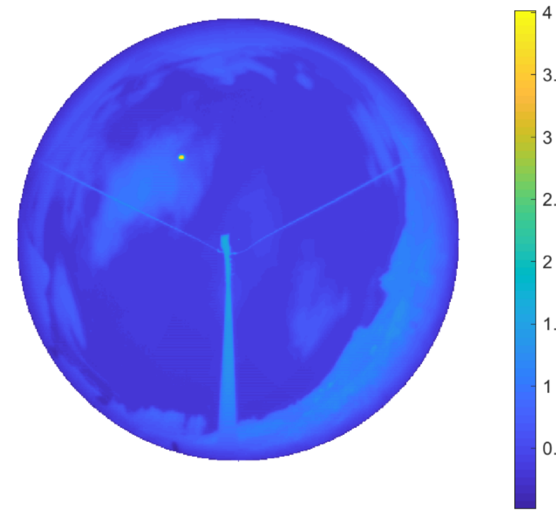


2.3 Results

The light-green regions are the clouds detected by the original program.

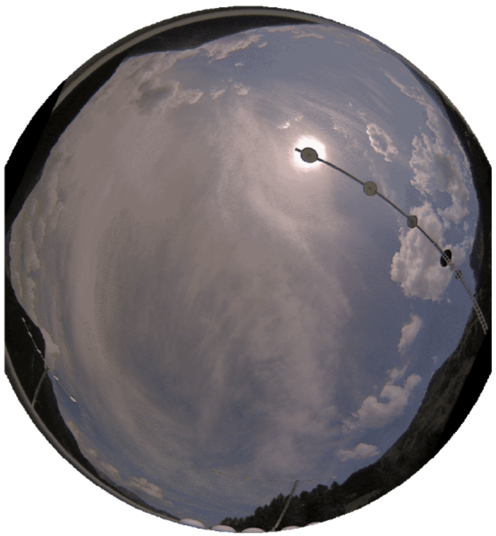


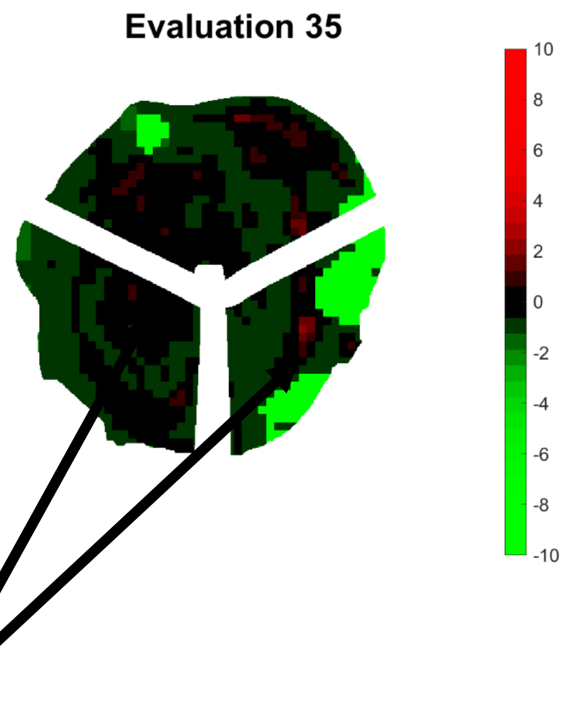


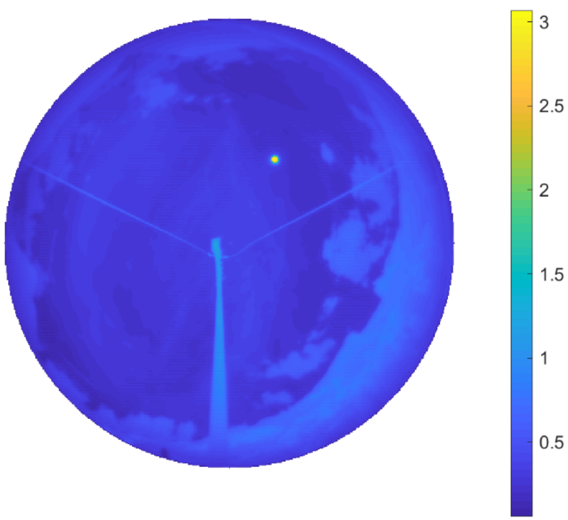


Now the algorithm is also able to detect thin clouds.

2.3 Results

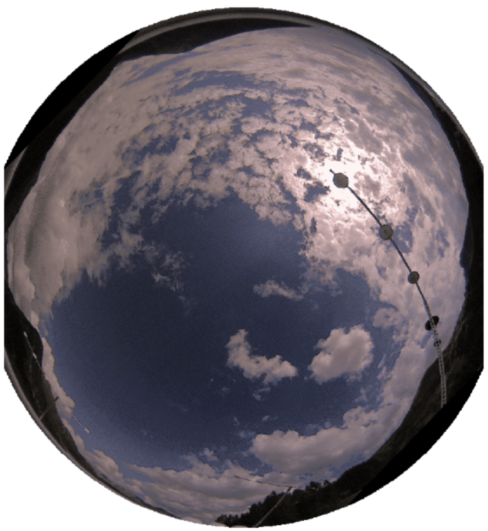


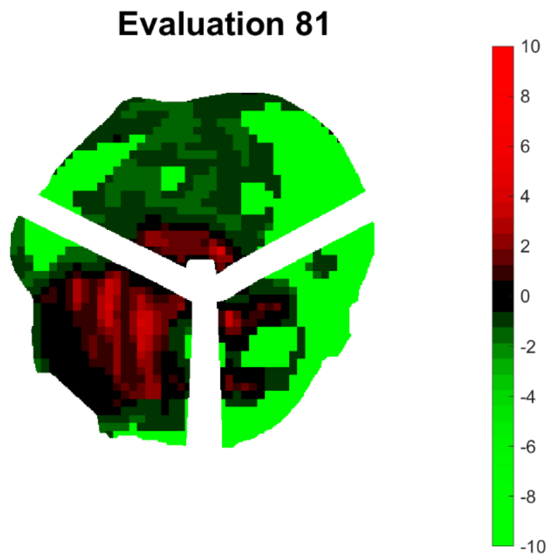


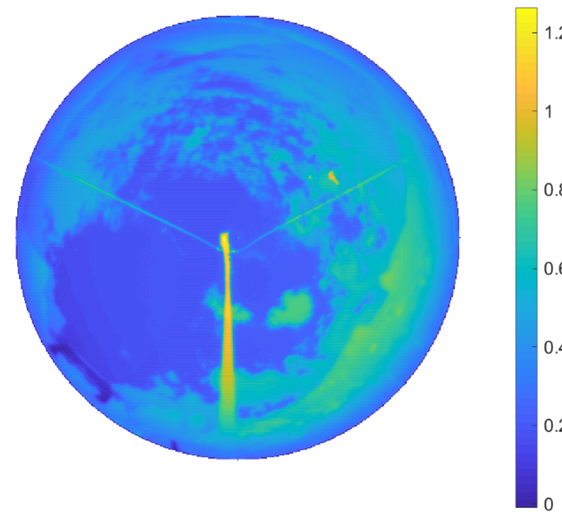


The algorithm detects thin clouds much better in general, but there still are some regions it is uncertain about.

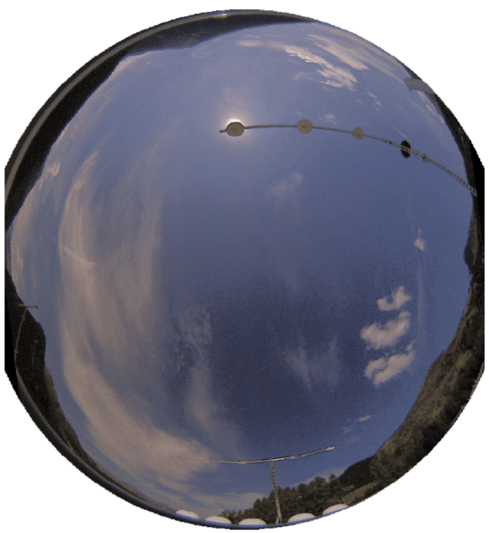
2.3 Results

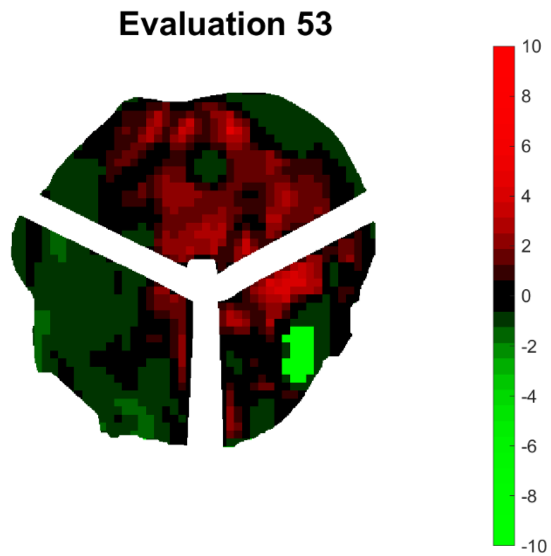


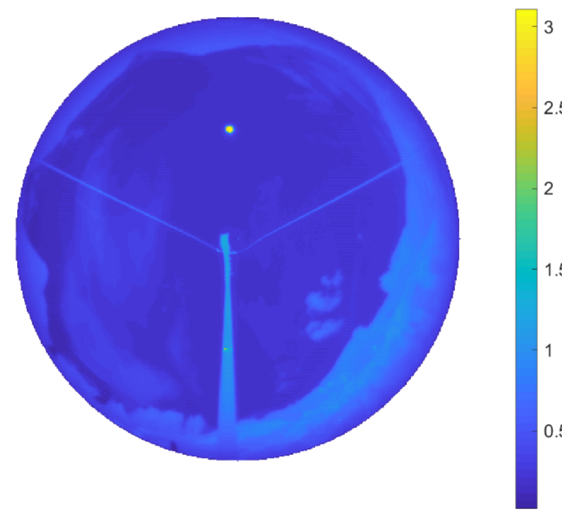




2.3 Results







pmod wrc

2.3 Results and Ideas

- The algorithm seems promising, it is mostly able to detect thin clouds. The training is much faster as compared to the training of the genetic algorithm.
- The correct classification rate on the training set is 93% (as determined via cross-validation), it does, however, seem to be a bit lower on real IRCCAM images, where the algorithm has to deal with mixed cases (partly cloud, partly clear sky) and missing data (horizon mask regions).
- To improve the algorithm further, it may be a good idea to design a more realistic training set and to increase the number of training samples.
- The machine learning algorithm could be applied to visible images first (since thin clouds are clearly discernible on visible images), which would allow the creation of ample and more realistic training samples for the training on IRCCAM images.
- It might be useful to include large-scale information in the model as well, instead of only focussing on small pixel squares.

References

[1] Gröbner, J. et al.: **"The infrared all-sky cloud camera at PMOD/WRC (IRCCAM),"** 2016.

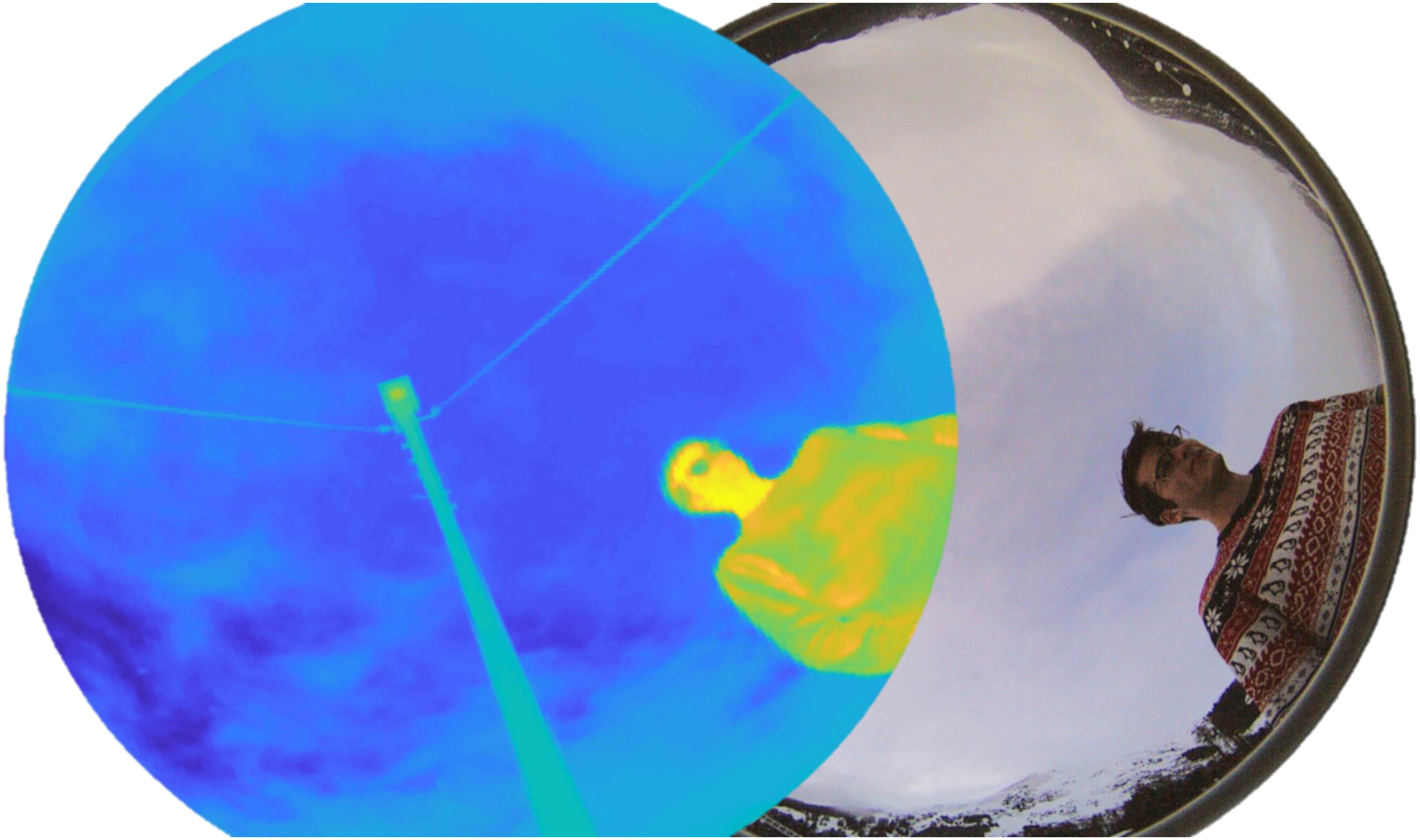
[2] Aebi, C., Gröbner, J., and Kämpfer, N.: **"Cloud fraction determined by thermal infrared and visible all-sky cameras,"** *Atmos. Meas. Tech.*, 11, 5549-5563, 2018.

[3] Luo, Q. et al.: **"Cloud classification of ground-based infrared images combining manifold and texture features,"** *Atmos. Meas. Tech.*, 11, 5351-5361, 2018.

[4] Brocard, E. et al.: **"Detection of Cirrus Clouds Using Infrared Radiometry,"** in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 595-602, Feb. 2011.

[5] McCall, J.: **"Genetic algorithms for modelling and optimisation,"** Journal of Computational and Applied Mathematics, vol. 184, no. 1, 2005.

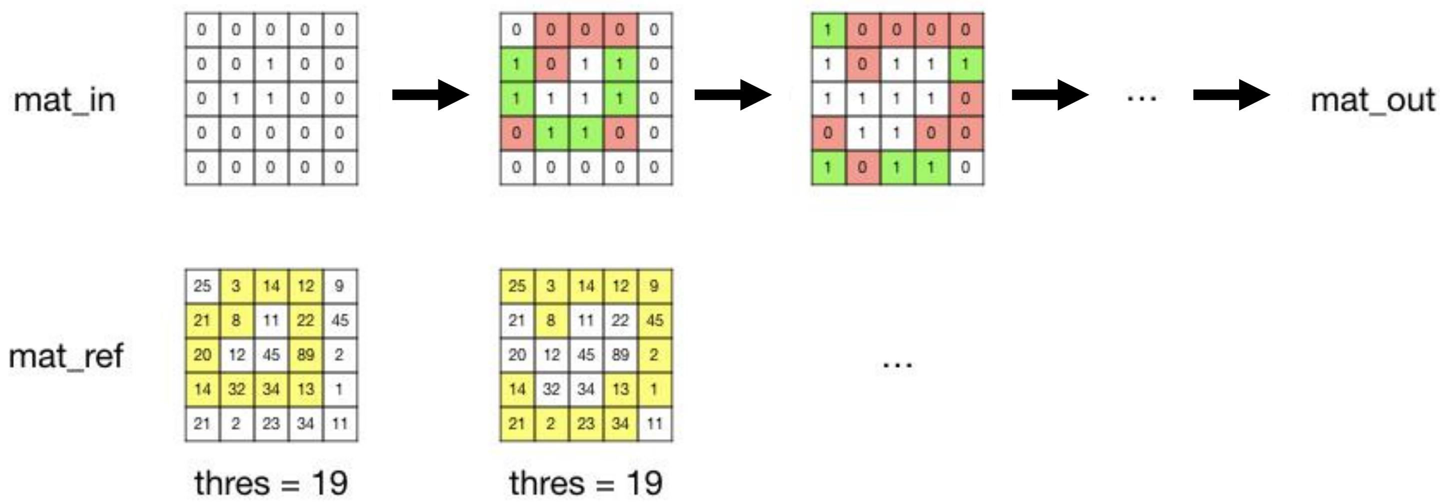
[6] Soille, P.: ***Morphological Image Analysis. Principles and Applications*,** Second Edition, Springer-Verlag Berlin Heidelberg GmbH, 2004.



Appendix

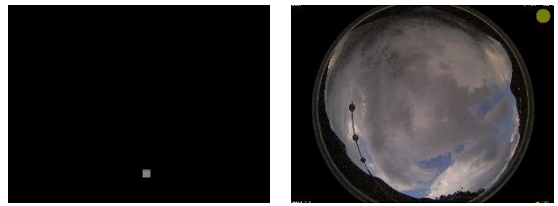
1.2 New Script: Addpixels

- Expand pixel regions with value 1 (the "seed pixels") in the logical matrix **mat_in** by a given number of layers, unless **mat_ref** is smaller than **thres** for the pixels in question.

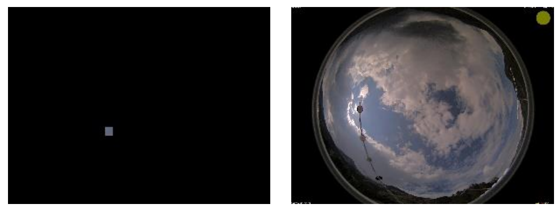


2.2 Script: Training

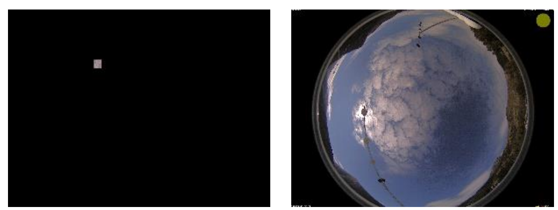
100 training pictures:



cloud



no cloud



cloud

no cloud

Training data located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Machine_Learning\training

2.2 Script: Training

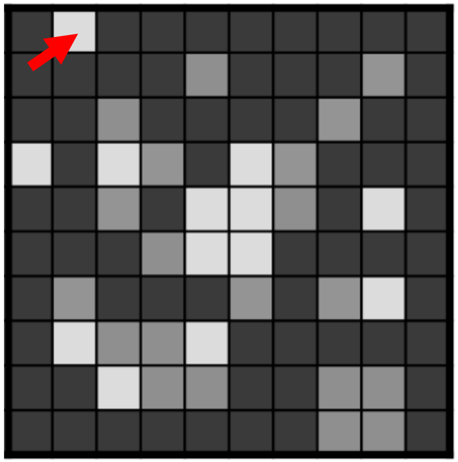
Gray-level Co-occurence Matrix (GLCM):

- The gray levels in the pixel matrix are partitioned into n bins.
- The function $\rho(x,y)$ maps each pixel in the pixel matrix to a «partner pixel»
  - Example: $\rho(x,y) = (x+1, y+1)$ maps each pixel to its diagonal neighbour.
- The $n \times n$ GLCM is calculated in the following way:

$$\begin{pmatrix} (0,0) & (0,1) & (0,2) & & (0,n) \\ (1,0) & (1,1) & (1,2) & \cdots & (1,n) \\ (2,0) & (2,1) & (2,2) & & (2,n) \\ & \vdots & & & \vdots \\ (n,0) & (n,1) & (n,2) & \cdots & (n,n) \end{pmatrix} = \begin{pmatrix} 4 & 0 & 12 & & 1 \\ 3 & 8 & 4 & \cdots & 4 \\ 0 & 12 & 2 & & 6 \\ & \vdots & & \ddots & \vdots \\ 1 & 13 & 9 & \cdots & 3 \end{pmatrix}$$

Number of pixels (x,y) that belong to bin 0 **and** have a partner $\rho(x,y)$ that belongs to bin 1

$\rho(x,y) = (x+1, y+1)$



- This is done for
  - $\rho(x,y) = (x+1, y+1)$
  - $\rho(x,y) = (x+1, y-1)$
  - $\rho(x,y) = (x-1, y+1)$
  - $\rho(x,y) = (x-1, y-1)$

Training script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Machine_Learning\training.m
Idea based on [3] Luo et al. (2018)

2.2 Script: Training

$$\begin{pmatrix} 4 & 0 & 12 & & 1 \\ 3 & 8 & 4 & \cdots & 4 \\ 0 & 12 & 2 & & 6 \\ & \vdots & & \ddots & \vdots \\ 1 & 13 & 9 & \cdots & 3 \end{pmatrix}$$

Properties calculated based on the GLCM:

- Contrast $\sum_{i,j} |i-j|^2 p(i,j)$
- Correlation $\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$
- Energy $\sum_{i,j} p(i,j)^2$
- Homogeneity $\sum_{i,j} \frac{p(i,j)}{1+|i-j|}$

These are the x_i.

Training script located under \\ad.pmodwrc.ch\Institute\projects\CRUX\IRCCAM\mat\LS\Machine_Learning\training.m