Leibniz-Institut für
Astrophysik Potsdam

Detection of sunspots on digitized photographic plates

When machine-learning becomes really useful

Proof of Concept

Fournier Yori / Zooming into the universe (AG 2021) / 13 – 17 Sept. 2021

Overview

1) **Observation context**
   In which context were these photographic plates taken?

2) **Scientific context**
   What can we learn from these photographic plates?

3) **First detection attempt**
   The straight-forward method – *failed*

4) **Second detection attempt**
   Human brain-designed filter – *failed*

5) **Third detection attempt**
   Machine-designed filter – *encouraging results*

6) **Presentation of the preliminary results and discussion**

7) **About the reproducibility of this work**
   The problem and its possible solutions

Observation context

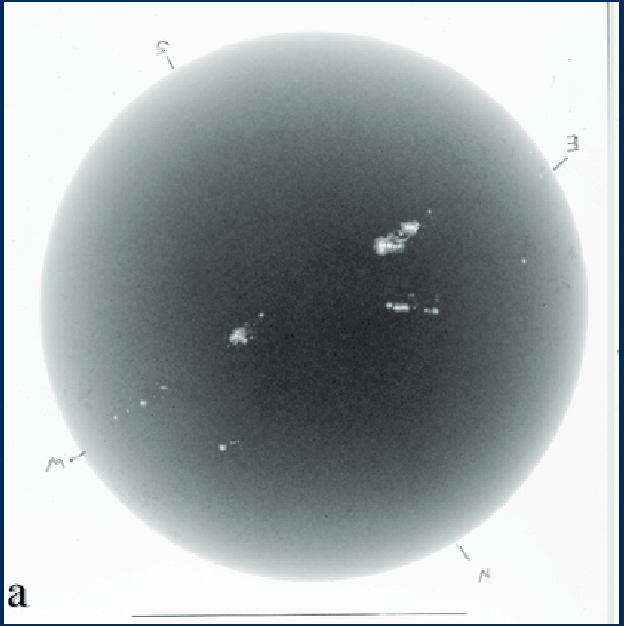
In which context were these photographic plates taken

Credit: AIP

Between 1943 and 1991 about **3700 observations of the sun** were carried out at the **Einstein Turm** in Potsdam.
(more on Einstein Turm, see: *Denker et. al. (2016) AN*)

- These observations show the entire solar disk and exhibit solar features such as: sunspots and filaments.

- The observations were developped on **photographic-plates**. The latter are made of glass, they were a **long lasting medium**, allowing **high resolution**.



Pal, Partha S. et. al. 2020 (Astron. Nachr. 2020; 341: 575– 587.)

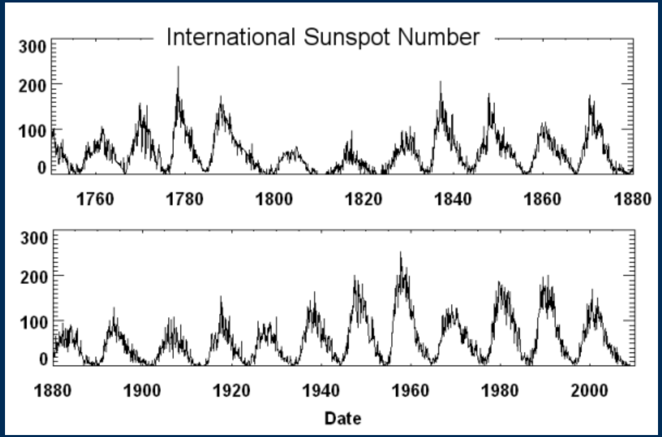
Scientific context (I)

What can we learn from these photographic plates?

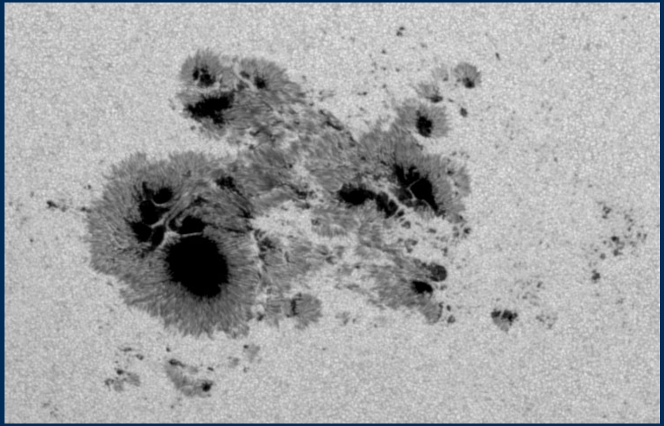
The solar magnetic field is very dynamic and exhibits short term (minutes) but also long term (tens of years) variations.

In order **to understand the dynamic of the solar** magnetic field we need to **gather** as many information as possible.

One prominent example is the **number of magnetic sunspots.**



Credit: spaceweather.com



Courtesy of NASA/SDO and the AIA, EVE, and HMI science teams.

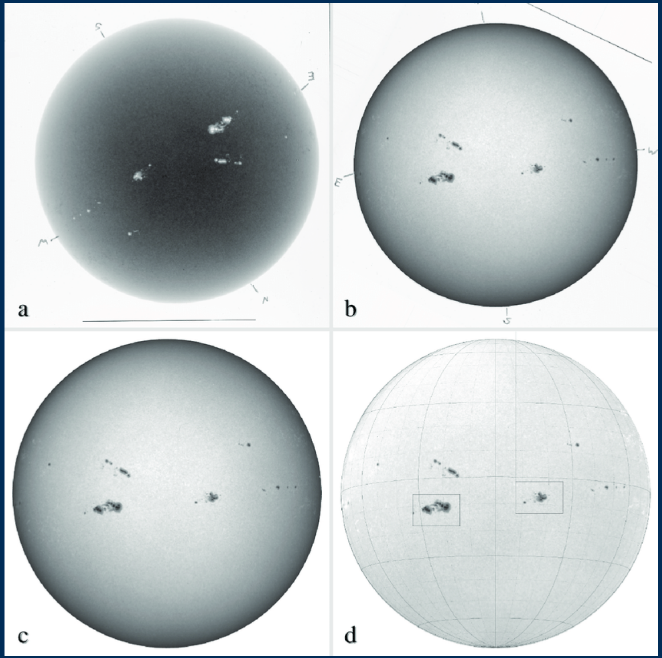
Scientific context (II)

What can we learn from these photographic plates?

All ~**3700 plates were digitized** with a high-resolution scanner.

They were preprocessed to bring homogeneity into the dataset: orientation, normalization, and limb-correction.
(see: *Pal, Partha S. et. al. 2020 AN)*

This photographic plates catalog was published the 6th Oct. 2020 as part of the **APPLAUSE project**.
(https://www.plate-archive.org/applause/documentation/data-release-dr3s/)



Pal, Partha S. et. al. 2020 (Astron. Nachr. 2020; 341: 575– 587.)

Some scientifically relevant information from these plates:

- **number of sunspots**
- **morphological properties**
- **coordinates** on the surface of the Sun (heliospheric coordinates).

These informations will be gathered to construct a derived catalog and will be published as part of DR4 of APPLAUSE.

First selection attempt

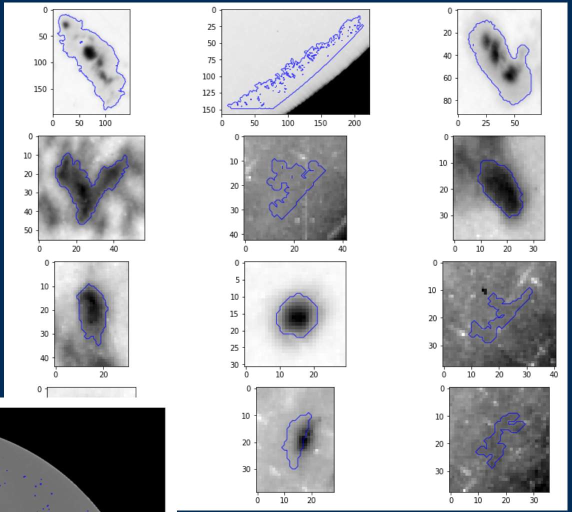
The straightforward method - *failed*

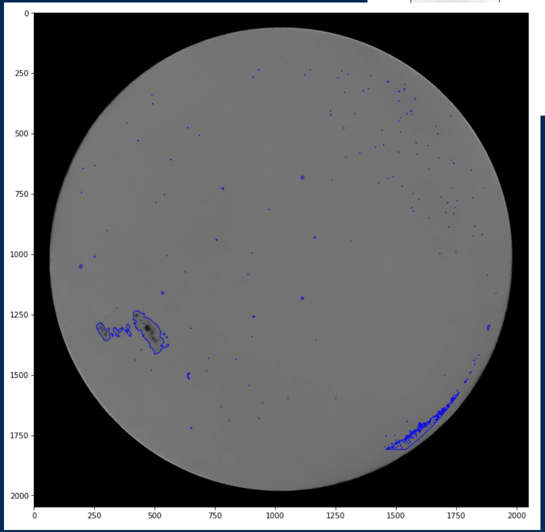
Modern image data:

- **Sunspots** are well defined: regions where the **intensity is below 94% of the mean intensity** of the solar disk.
- On modern image data sunspots can be easily **selected via threshold filtering** on intensity.

Photographic plates:

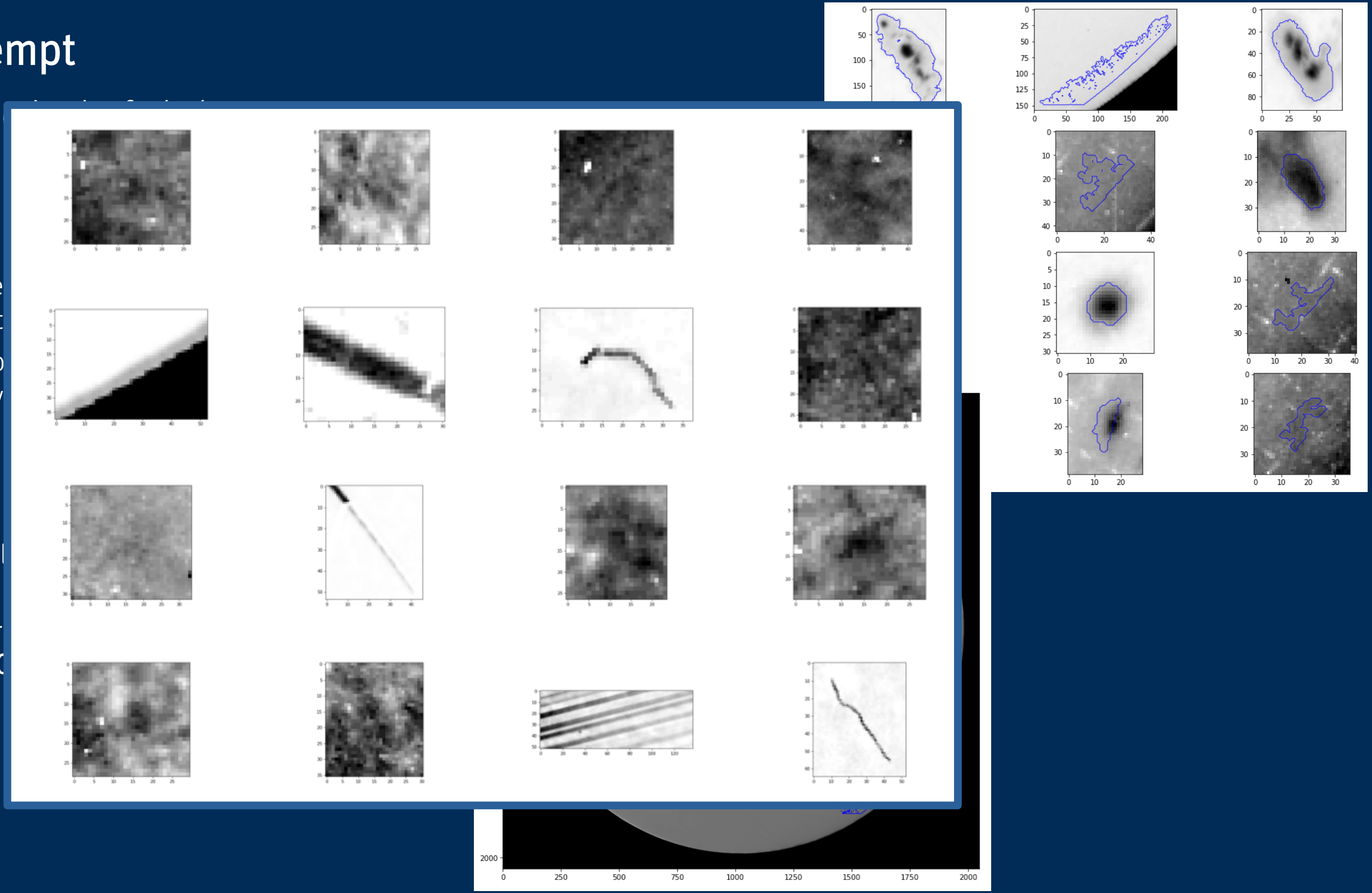
- Naturally the photographic plates are **not as clean as modern space observations.**
- On the preprocessed photographic plates, the straightforward threshold method fails: **~12,000 undeterminded features.**





First selection attempt

The straightforward m



Modern image data:

• **Sunspots** are well defined: re
94% of the mean intensity of t

• On modern image data sunsp
threshold filtering on intensity

Photographic plates:

• Naturally the photographic pl
space observations.

• On the preprocessed photogr
threshold method fails: **~12,00**

First selection attempt

The straightforward meth

Modern image data:

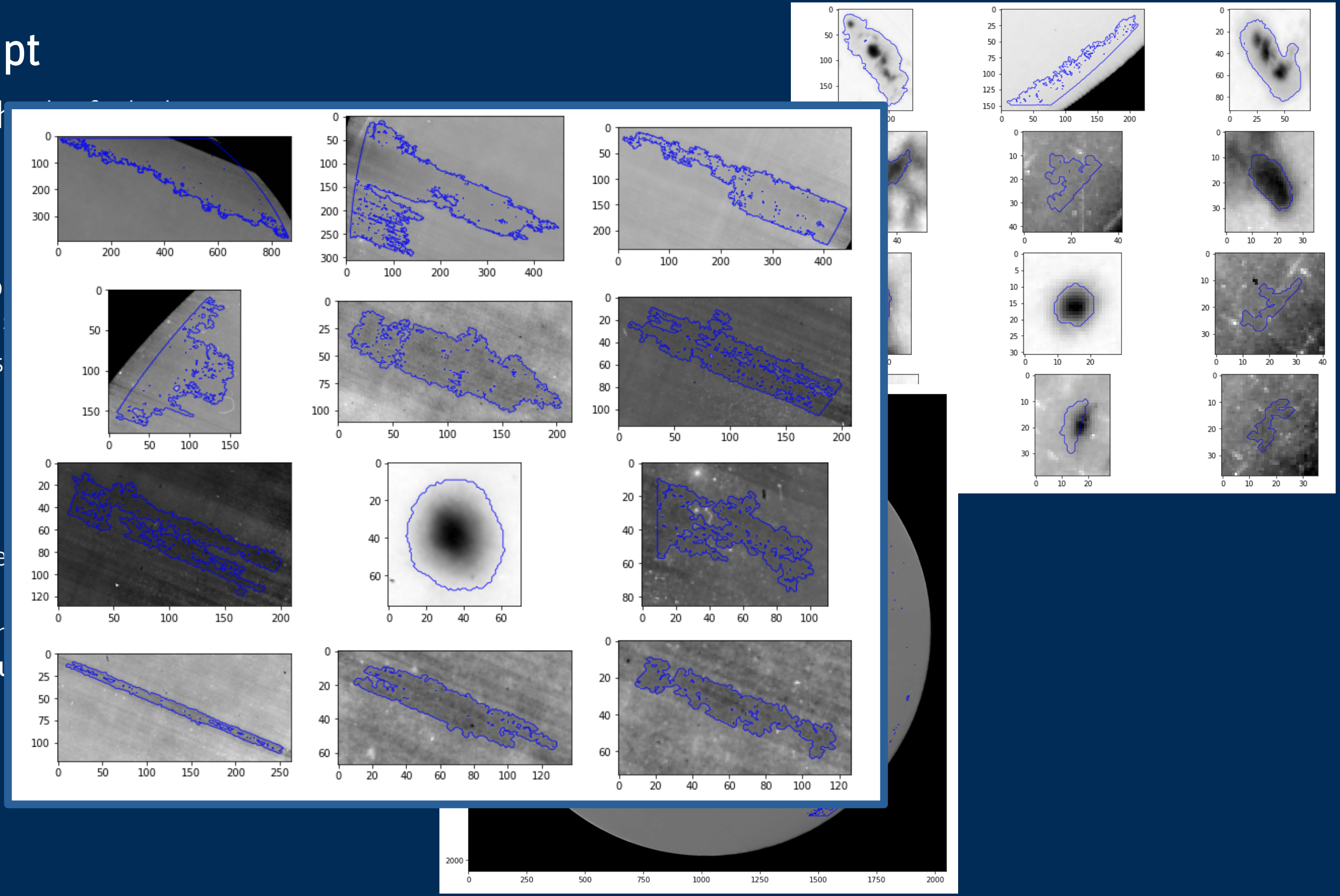
• **Sunspots** are well defined: regio
94% of the mean intensity of the

• On modern image data sunspots
threshold filtering on intensity.

Photographic plates:

• Naturally the photographic plate
space observations.

• On the preprocessed photograph
threshold method fails: **~12,000**



Second selection attempt

The human brain-designed filter - *failed*

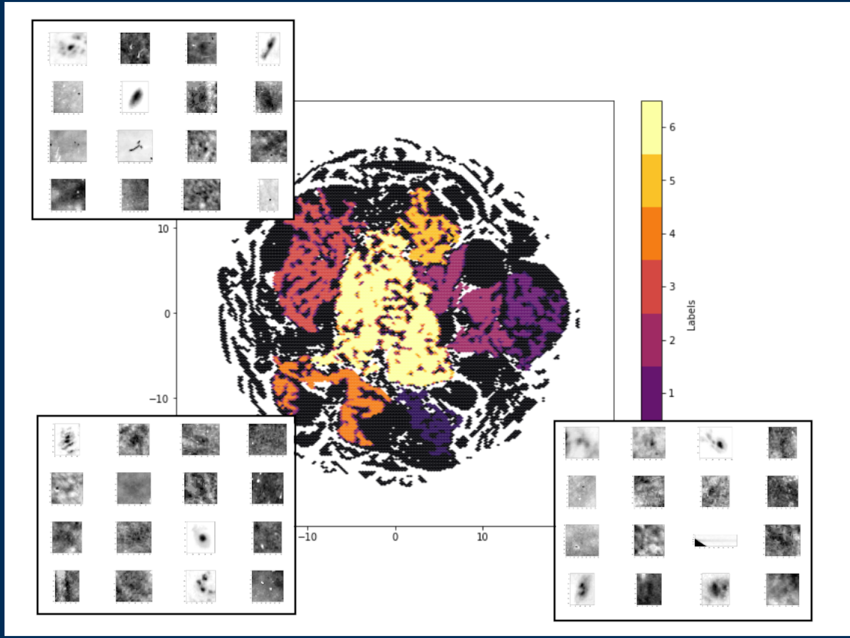
Improve filtering:

False positives show **morphological properties** that are **not sunspot-like**.

We **compute the morphological properties** of all features and try to design a filter to separate them.

Unfortunately the high dimensionality of the problem makes it difficult to design **a robust non-bias filter.**

t-SNE: t-distributed Stochastic Neighbor Embedding



Third selection attempt (I)

The machine-designed filter – *encouraging results*

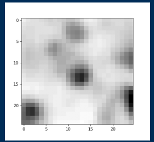
Build a filter with neuronal network:

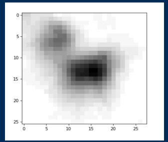
Convolutional Neural Networks (CNN) are the most frequently used network-type for image processing.

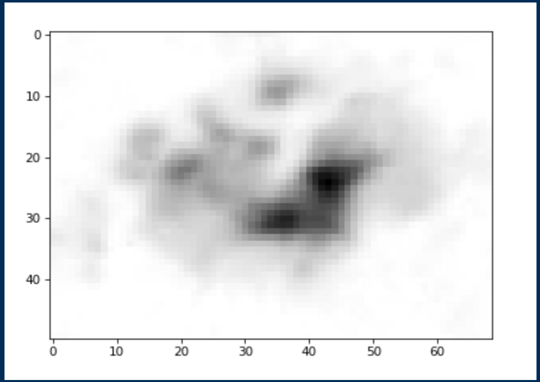
CNN is particularly adapted since it allows to **identify the relevant morphological properties** without bias.

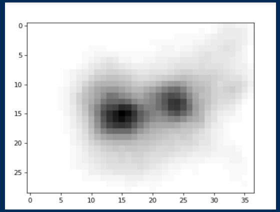
The problem is that CNN are particularly **adapted for small images**:
between 64^2 – 512^2 pixels.

In our case, due to the **wide dynamical range** of sunspots sizes, the postprocessed photographic plates are **2048^2 pixels**.









Third selection attempt (II)

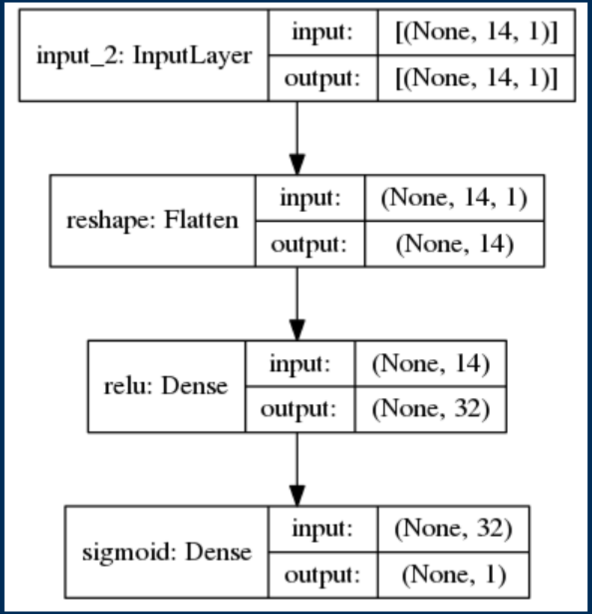
The machine-designed filter – *encouraging results*

CNN is an overkill since we already have the morphological properties of all detected features.

A simple *neural network* should be able to deliver an adapted filter.

- We have **14** relevant **morphological properties.** These **are the input of our *neural network!!*** (not the scanned plates)

- Some of these **properties must mix** to catch the high dimensionality of our problem. We arbitrarily start with **two hidden layers.**



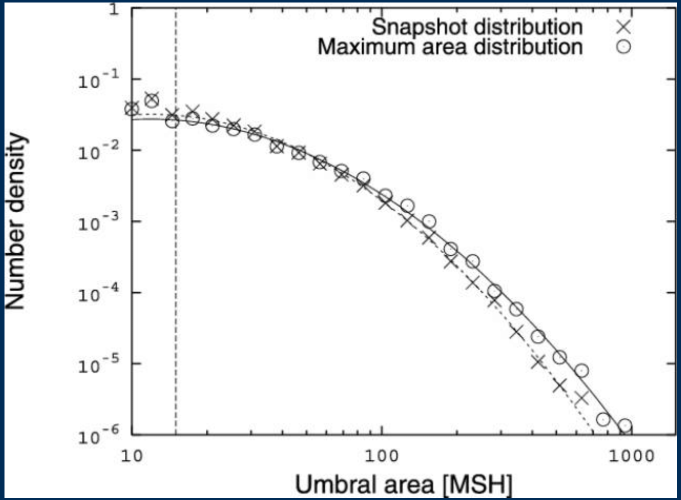
Third selection attempt (III)

The machine-designed filter – *encouraging results*

Designing the training set:

How did we **design the training** set?
Which **properties** should have the training set?

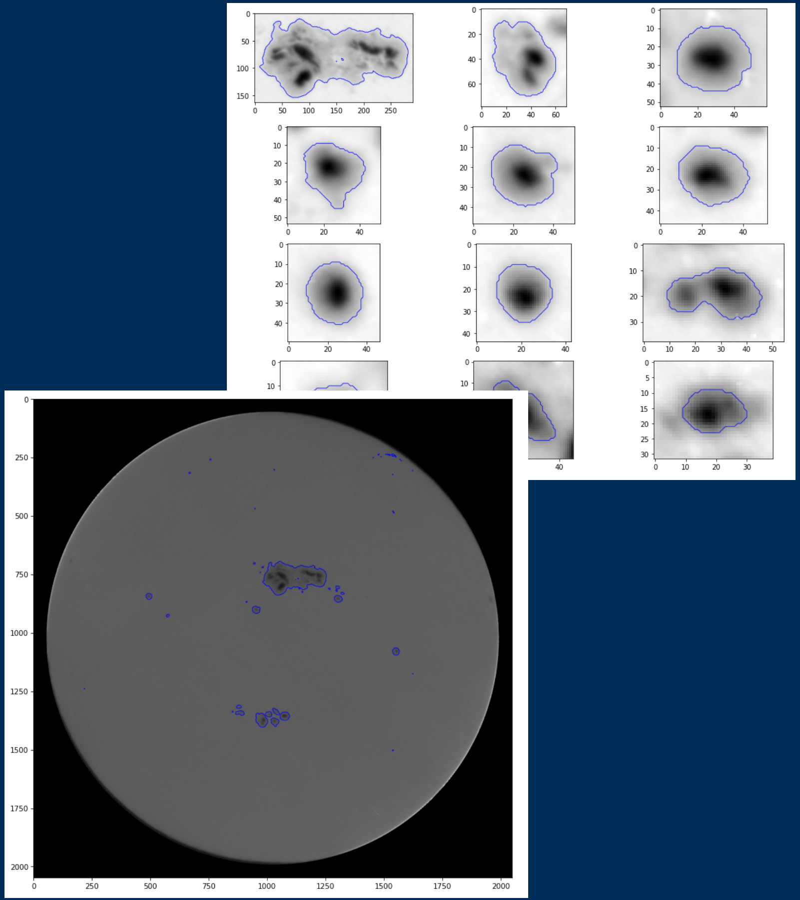
- About **50/50** sunspots and non-sunspots
- About 10% of the features should be covered
- Most of the **non-sunspot** types should be **represented**
- Most of the **sunspot** types should be **represented**
- The **population rules** should be **respected**,
i.e.: distribution of sunspots over their size



Baumann & Solanki, 2005 (A&A)

Third selection attempt (IV)

The machine-designed filter – *encouraging results*



Designing the training set: the sunspots

In the photographic-plates catalog, there is a **wide range of plate quality**. We could categorise the plates in „good“ and „bad“ quality plates.

There are about **10% of „good“ plates**, which is sufficient, and they are distributed all over the observation period between 1943 and 1991.

The „good“ plates are such that the modern selection technique (threshold) works up to a high percentage, leading to **very few „false positives“.**

This selection by „good quality“ plates naturally respects the population rules.

Third selection attempt (V)

The machine-designed filter – *encouraging results*

Designing the training set: the non-sunspots

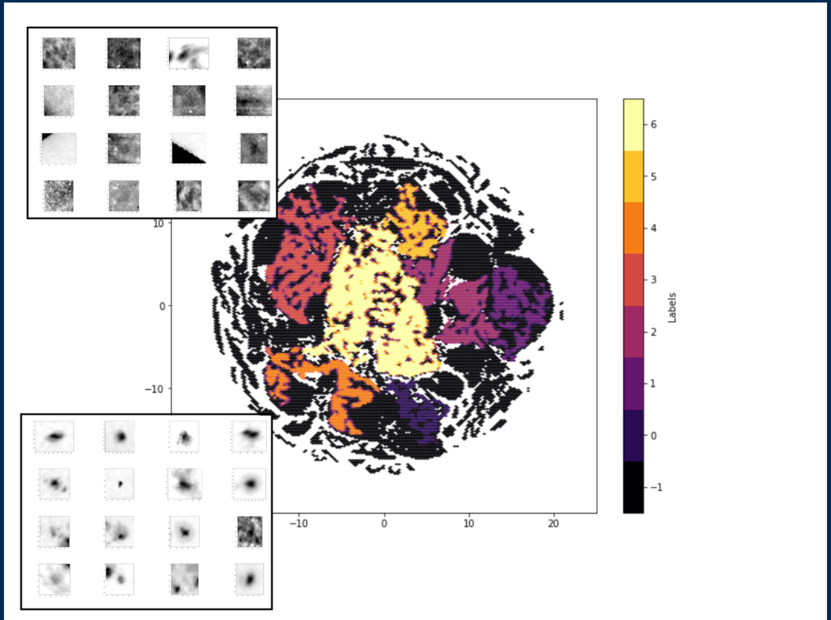
The **challenge** here is to **select a good representation** of all the non-sunspot types.

First we can **remove** all features which are **surely sunspots** (thanks to the morphological filter)

We have a soup of unknown features.

We **analyse various subsets** of the families obtained from the **tSNE maps** and try to extract homogeneous sets.

This **method is arbitrary** and difficult to automatize. No clear parameters nor possibilities to test **the statistical relevance of the selection…** (still unsatisfactory)

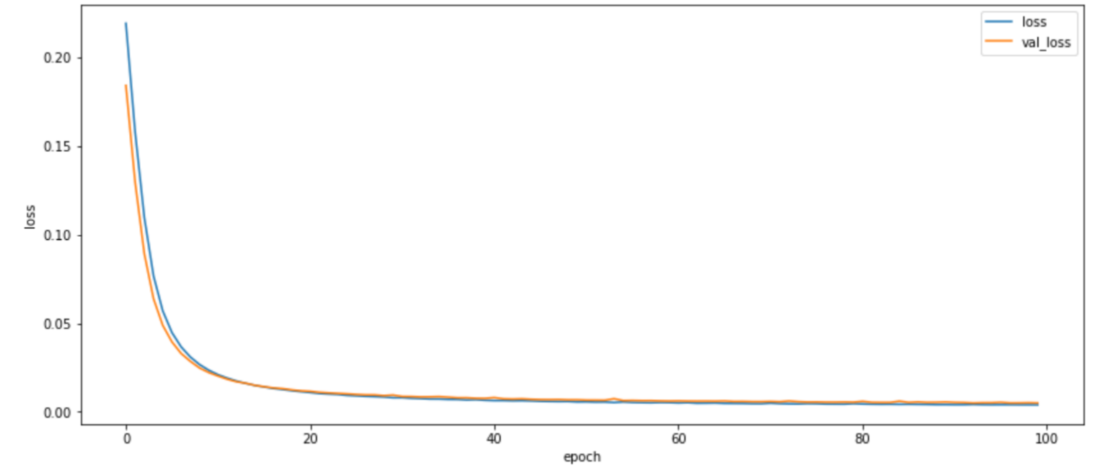


Presentation of the results (I)

The training.

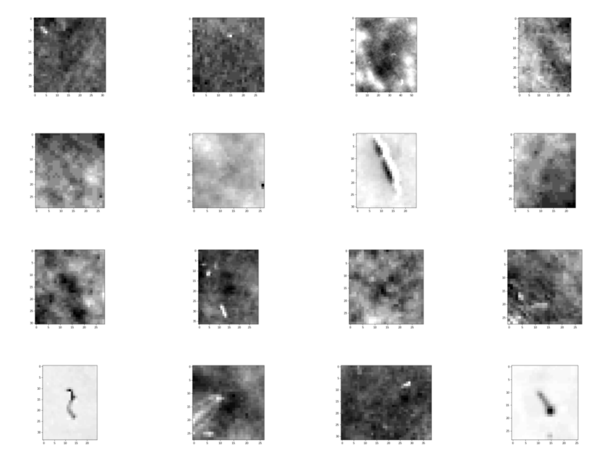
The **training** is **fast and robust** since we are exploring a **tiny parameter space**
(513 independent parameters) and using a simple neural network and no convolution layer.

To **prevent overfitting** the training data we stop the training at the begining of convergence: **epoch 40**.



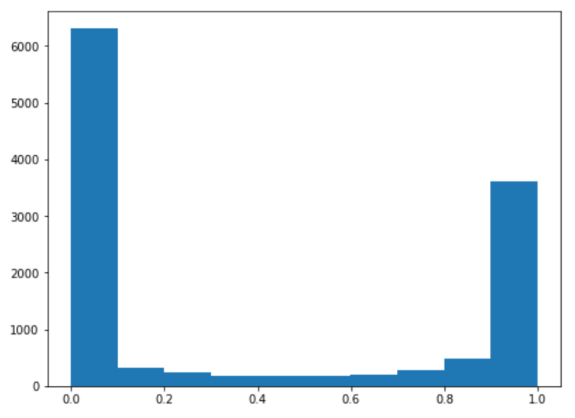
Presentation of the results (II)

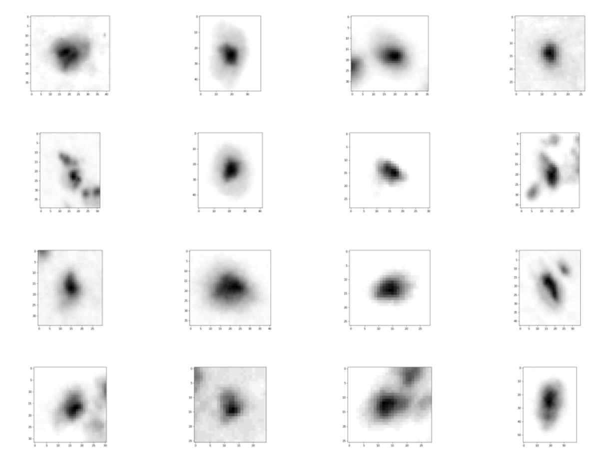
Application to the entire set.



Applying the machine-designed filter we obtain quite **encouraging results**:

- Few „False positives“ (less than 1%)
- More „False negatives“ (about 10%)
- Little amount of uncertainty





Presentation of the results (III)

Discussion

These are **preliminary results**.

The next step is **validation** of the catalog:

- **Compute the sunspots number** and compare to the known one (not trivial due to the grouping).
- Verify the **statistical distribution** of spots over their size.
- Verify the **statistical position** of spots along the years (butterfly diagram).
- Study the **intrinsic bias** of the method.

How to improve?

Improve the training-set selection.
Make the **selection of non-sunspots more systematic** and identify the parameters.

Is our trivial network adapted?

We need to carry out a **deeper study** of our neuronal network:

- Study each layer and weight of the parameter
- Study the network: number of layers, activation functions.

Intrinsic problem

We somehow reduce the parameter space by **providing the morphological properties** of the spots. Better would be that the **network extracts these properties** from the training set.

The **size of our images is too large** for a straight forward **CNN** network. However solutions exist: **slicing the images** or **randomily dropping data**.

About the reproductibility of this work

The problem and its possible solutions

The reproducibility issue:

- Make this work reproducible consumes time. Time that scientists generally don‘t have.
- Today it is reproducible, but what about tomorrow?

What we have already:

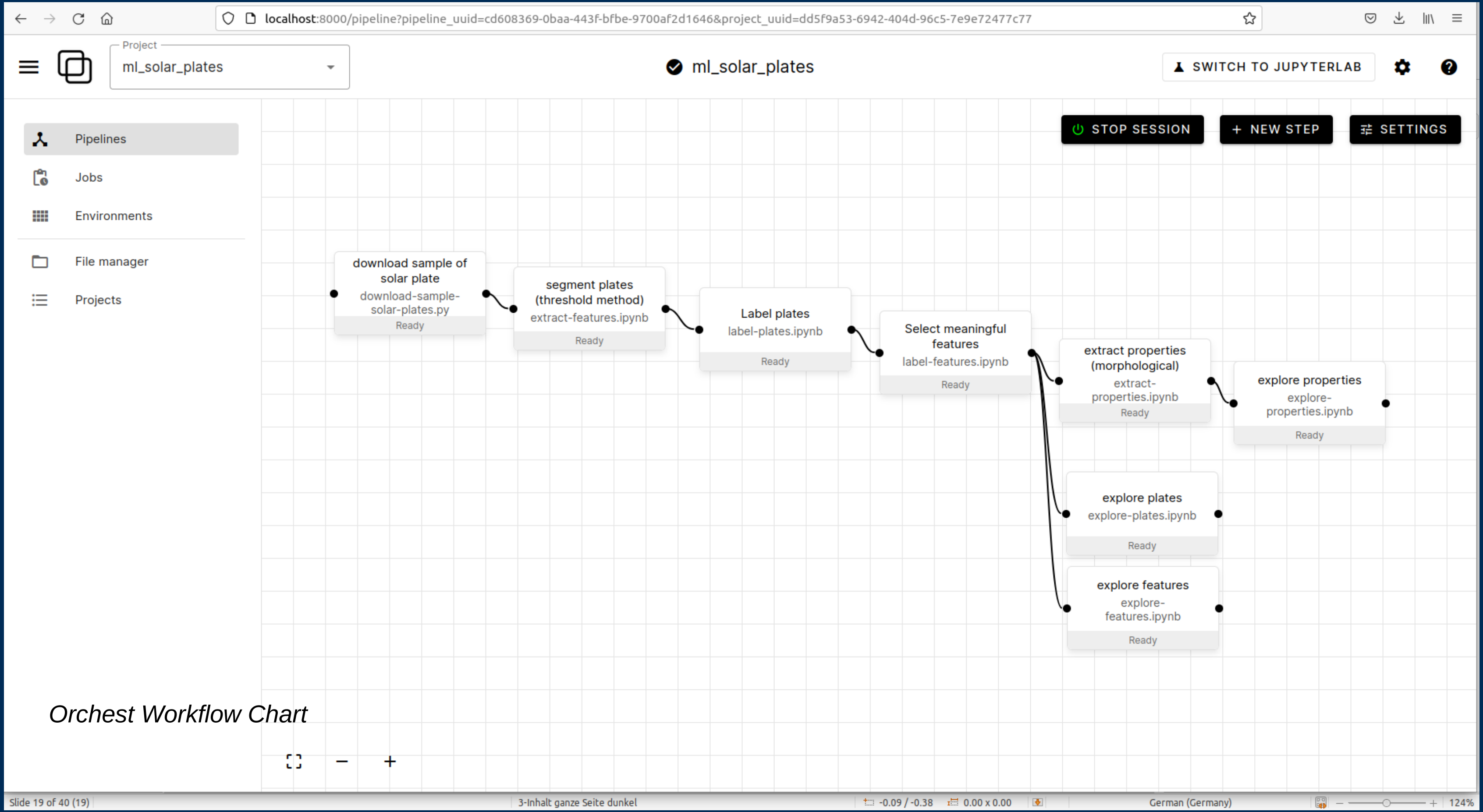
- Virtual environment: virtual machine, containers…
- Long lasting data formats: hdf5, fits, csv…
- Versioning system: for data and software
- Global identifiers: data identifier, software identifier.

What is missing:

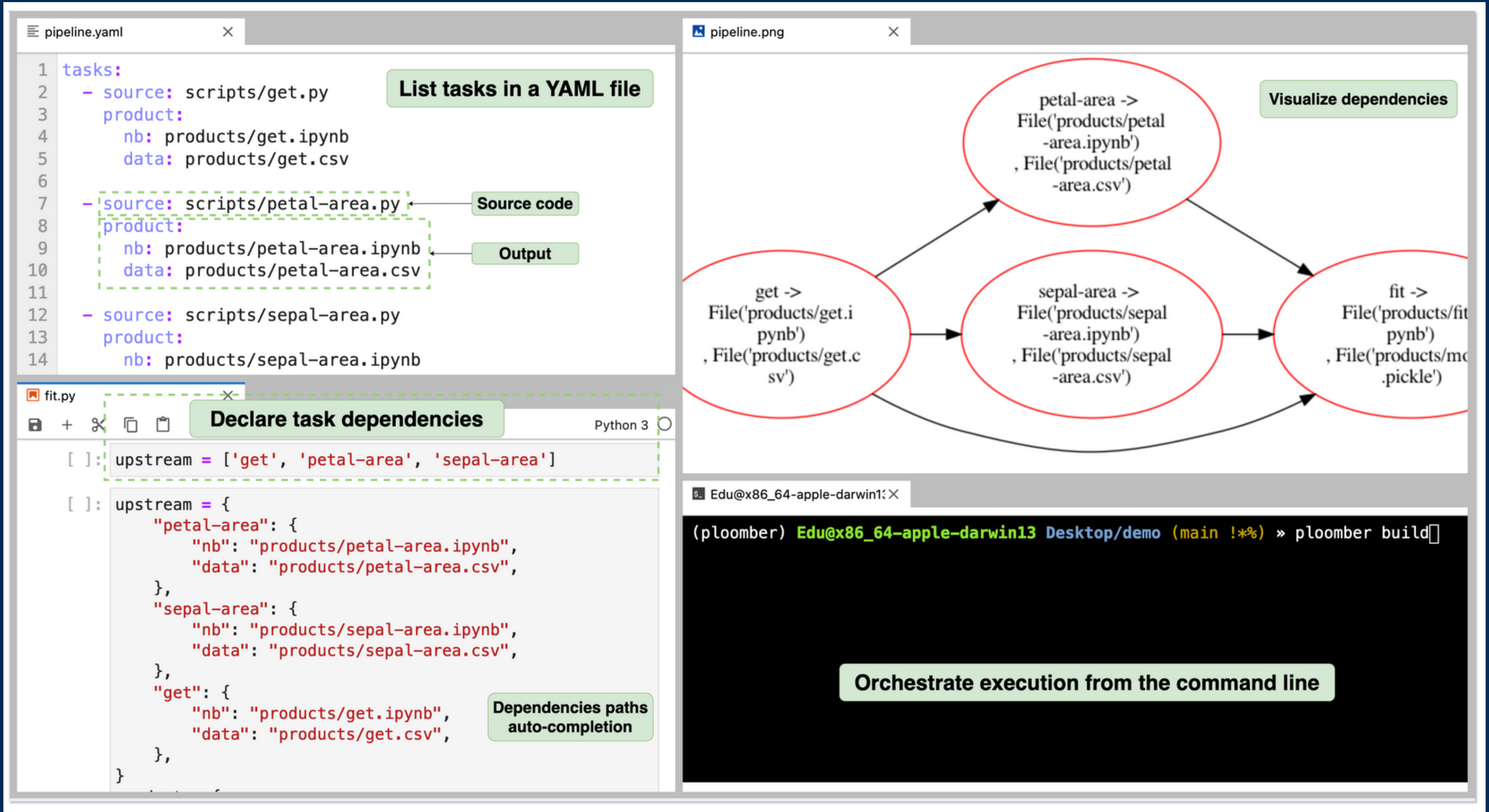
- Long lasting parallelization methods, hardware architecture and infrastructure
- Robust backend software to combine all available elements mentioned above
- Plug and play graphic interface adapted for scientific work.

Future Tools: (avant-garde)

- UltraPink
- Orchest
- Ploomber



Orchest Workflow Chart


pipeline.yaml
tasks:
- source: scripts/get.py
product:
nb: products/get.ipynb
data: products/get.csv
- source: scripts/petal-area.py
product:
nb: products/petal-area.ipynb
data: products/petal-area.csv
- source: scripts/sepal-area.py
product:
nb: products/sepal-area.ipynb
List tasks in a YAML file
Source code
Output
fit.py
Declare task dependencies
Python 3
[ ]: upstream = ['get', 'petal-area', 'sepal-area']
[ ]: upstream = {
"petal-area": {
"nb": "products/petal-area.ipynb",
"data": "products/petal-area.csv",
},
"sepal-area": {
"nb": "products/sepal-area.ipynb",
"data": "products/sepal-area.csv",
},
"get": {
"nb": "products/get.ipynb",
"data": "products/get.csv",
},
}
Dependencies paths auto-completion
pipeline.png
Visualize dependencies
petal-area -> File('products/petal-area.ipynb'), File('products/petal-area.csv')
get -> File('products/get.ipynb'), File('products/get.csv')
sepal-area -> File('products/sepal-area.ipynb'), File('products/sepal-area.csv')
fit -> File('products/fit.ipynb')
(ploomber) Edu@x86_64-apple-darwin13 Desktop/demo (main !*%) » ploomber build
Orchestrate execution from the command line