

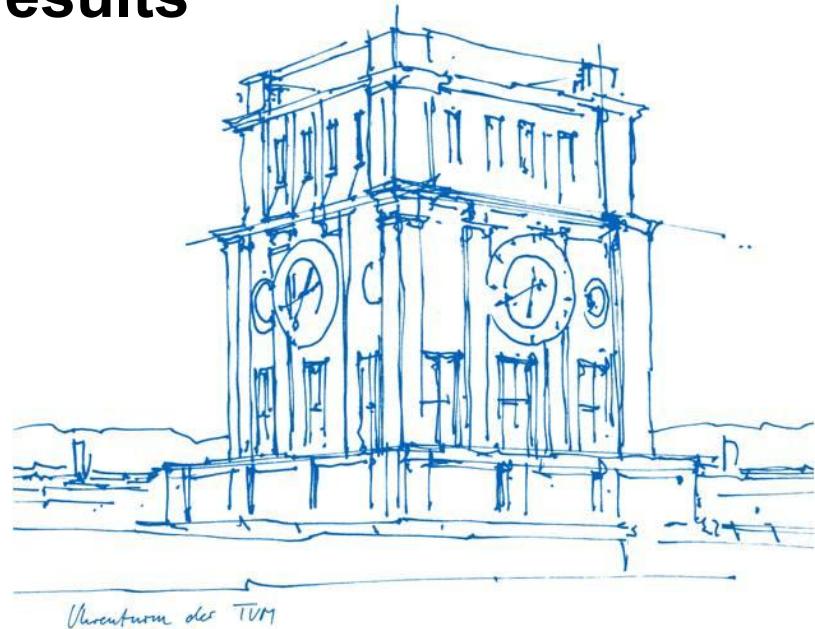
Should this have been
the first session?

Model development part 3: evaluation and visualization of results

Martin Menten (martin.menten@tum.de)

AFRICAI/ MICCAI summer school

14th September 2023



Hypothesis-driven research

- Research question:

"My method ~~is very good~~ at solving a challenging task."

- Testable hypothesis:

"My method **is better than the existing state-of-the-art**
at solving a challenging task."

Hypothesis-driven research

- Deciding on your research hypothesis shapes your entire paper

Methods sections -
evaluation metrics

Related works

"My method is better than the existing state-of-the-art
at solving a challenging task.'

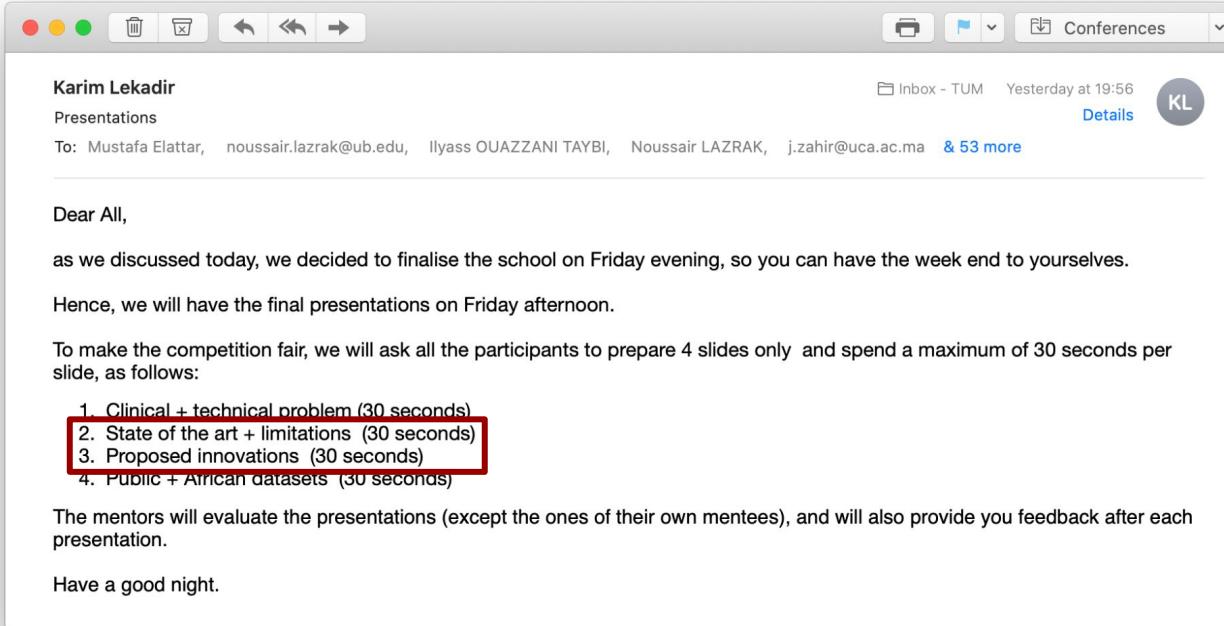
Results

Methods section - baselines

Introduction

Hypothesis-driven research

- Extra motivation by Karim



Karim Lekadir
Presentations

Inbox - TUM Yesterday at 19:56
Details KL

To: Mustafa Elattar, noussair.lazrak@ub.edu, Ilyass OUAZZANI TAYBI, Noussair LAZRAK, j.zahir@uca.ac.ma & 53 more

Dear All,

as we discussed today, we decided to finalise the school on Friday evening, so you can have the week end to yourselves.

Hence, we will have the final presentations on Friday afternoon.

To make the competition fair, we will ask all the participants to prepare 4 slides only and spend a maximum of 30 seconds per slide, as follows:

1. Clinical + technical problem (30 seconds)
2. State of the art + limitations (30 seconds)
3. Proposed innovations (30 seconds)
4. Public + African datasets (30 seconds)

The mentors will evaluate the presentations (except the ones of their own mentees), and will also provide you feedback after each presentation.

Have a good night.

Implement your evaluation pipeline early



- Provides a quantitative measure where you are with your research
- Helps you understand the strengths and limitations of the baselines
- Culls unpromising research ideas
- Focuses your writing

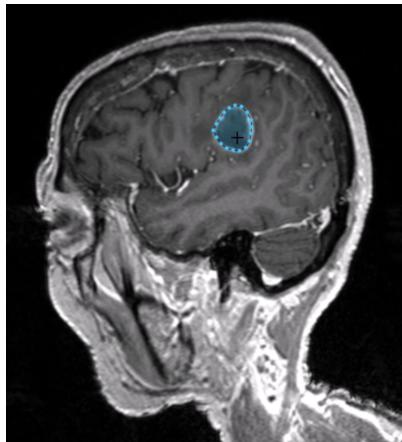
Selecting evaluation metrics - starting point

- Check how others have been evaluating their methods
- Decent chance that they know what they are doing
- Intrinsic comparability with other methods

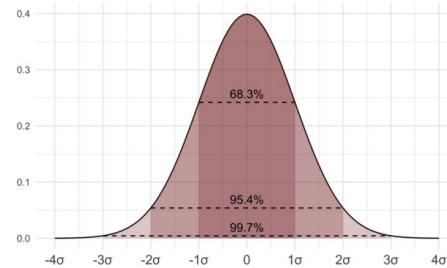


Selecting evaluation metrics - clinical relevance

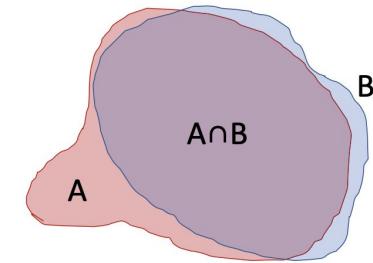
- Envisioned clinical use case may inform focus of evaluation



Brain tumor
segmentation



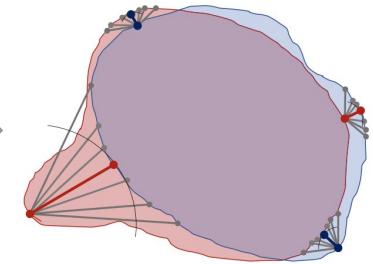
Volume quantification



Dice similarity coefficient



Surgical guidance



Hausdorff distance

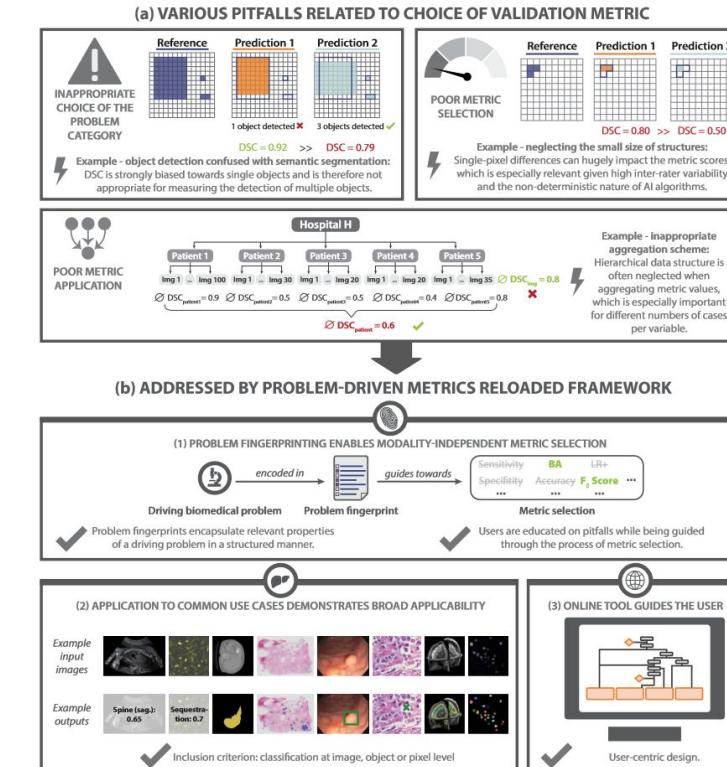
Selecting evaluation metrics - expert consensus

- Metrics reloaded

Metrics Reloaded: Recommendations for image analysis validation

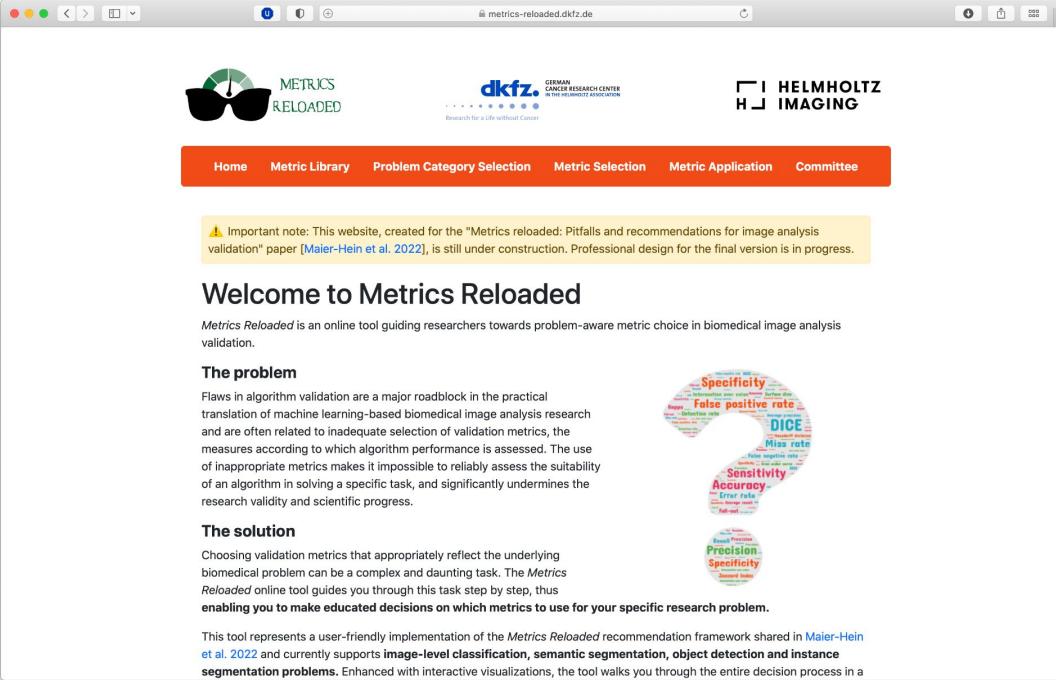
arXiv:2206.01653v6 [cs.CV] 30 Jun 2023

LENA MAIER-HEinz¹, German Cancer Research Center (DKFZ), Germany, Heidelberg University, Germany, and National Center for Tumor Diseases (NCT), Germany
 ANNAHITA HENKEL¹, German Cancer Research Center (DKFZ), Germany and Heidelberg University, Germany
 PATRICK GÖDDE, German Cancer Research Center (DKFZ), Germany, Heidelberg University, Germany, and National Center for Tumor Diseases (NCT), Germany
 MINU D. TIZABE, German Cancer Research Center (DKFZ), Germany and National Center for Tumor Diseases (NCT), Germany
 GORDON RÜTTNER, German Cancer Consortium (DKTK) and German Cancer Research Center (DKFZ), Germany, Goethe University Frankfurt, Germany, and Frankfurt Cancer Institute, Germany
 EVANGELIA CHRISTODOULOU¹, German Cancer Research Center (DKFZ), Germany
 BEN GLOCKER, Imperial College London, UK
 JONAS HÜLSE, German Cancer Research Center (DKFZ), Germany
 JENS KLEISCH, University Medical Essen, Germany
 MICHAEL KOZIUREK, Masaryk University, Czech Republic
 MAURICIO REYES, University of Bern, Switzerland
 MICHAEL A. RIEGEL, Smila Metropolitan Center for Digital Engineering, Norway and UiT The Arctic University of Norway, Norway
 CHRISTIAN WIESENBERGER, German Cancer Research Center (DKFZ), Germany
 A. EMRE KAVUR, German Cancer Research Center (DKFZ), Germany
 CAROLE H. SLIDJE, University College London, UK and King's College London, UK
 MICHAEL BAUDGARTNER, German Cancer Research Center (DKFZ), Germany
 MATTHIAS BESSENRODT, German Cancer Research Center (DKFZ), Germany
 DOREEN HECKMANN-NOTZEL, German Cancer Research Center (DKFZ), Germany and National Center for Tumor Diseases (NCT), Germany
 TIM RADTSCH, German Cancer Research Center (DKFZ), Germany
 LAURA AGUILERA-GRANADA¹, Universidad de Buenos Aires, Argentina
 CHICHA ANTONELLI, Imperial College London, UK and University College London, UK
 TAL ARBEL, McGill University, Canada
 SPYRIDON BAKAS, University of Pennsylvania, USA and Perelman School of Medicine at the University of Pennsylvania, USA
 MARIEL BENIS, Holon Institute of Technology, Israel and European Federation for Medical Informatics, Switzerland
 MATTHEW BLASCHKO, KU Leuven, Belgium
 M. JORGE CARDOSO, King's College London, UK
 VERONIKA CHERPYLINA, IT University of Copenhagen, Denmark
 GARY V. COLLINS, University of Modena, Italy
 KEYMAN FARAHANI, National Cancer Institute, USA
 LUCIANA FERRER, CONICET-UBA, Argentina
 ADRIAN GALDRAN, Universitat Pompeu Fabra, Spain and University of Adelaide, Australia
 BRAM VAN GINNEKEN, Fraunhofer MEVIS, Germany and Radboud University Medical Center, The Netherlands
 RONALD HAFNER, DFG Chair of Excellence "Physics of Life", Germany and Center for Systems Biology, Germany
 DANIEL A. HASHIMOTO, University of Pennsylvania, USA, and Perelman School of Medicine at University of Pennsylvania, USA
 MICHAEL M. HOFFMAN, University Health Network, Canada, University of Toronto, Canada, and Vector Institute, Canada
 MEREL HUIJSMAN, Radboud University Medical Center, The Netherlands
 PIERRE JANNIN, Université de Rennes 1, France and INSERM, France
 CHARLES E. KAHN, University of Pennsylvania, USA



Selecting evaluation metrics - expert consensus

- Metrics reloaded



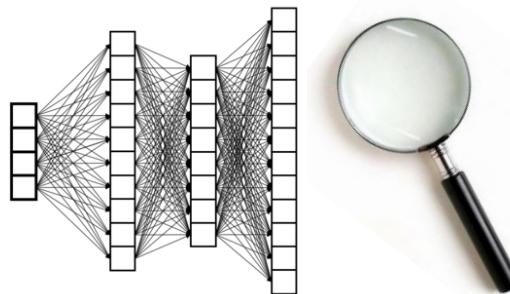
The screenshot shows the homepage of the Metrics Reloaded website, which is still under construction. The page features a header with the logo 'METRICS RELOADED' (with a stylized sun icon), the DKFZ logo ('GERMAN CANCER RESEARCH CENTER'), and the Helmholtz Imaging logo ('HELMHOLTZ IMAGING'). A navigation bar at the top includes links for Home, Metric Library, Problem Category Selection, Metric Selection, Metric Application, and Committee. A yellow warning box contains the text: '⚠ Important note: This website, created for the "Metrics reloaded: Pitfalls and recommendations for image analysis validation" paper [Maier-Hein et al. 2022], is still under construction. Professional design for the final version is in progress.' Below this, a section titled 'Welcome to Metrics Reloaded' is displayed. It includes a brief description: 'Metrics Reloaded is an online tool guiding researchers towards problem-aware metric choice in biomedical image analysis validation.' A 'The problem' section discusses the challenges of algorithm validation and the importance of selecting appropriate metrics. A 'The solution' section explains how the tool guides users through the process of choosing metrics. To the right of the text, there are two visual elements: a brain-shaped word cloud containing terms like Specificity, False positive rate, DICE, Miss rate, Sensitivity, Accuracy, Error rate, Precision, and Recall; and a circular diagram showing the relationship between Precision, Specificity, and Detection Index.

Beyond correctness

- A method may be beneficial in other aspect than correctness:



Required human effort



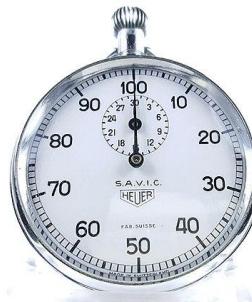
Explainability



Fairness



Cost



Run time



Carbon footprint

Tips when implementing metrics

- Beware of implementation errors
- Save results in “raw format”
in case you want to add other metrics later
- Use established libraries



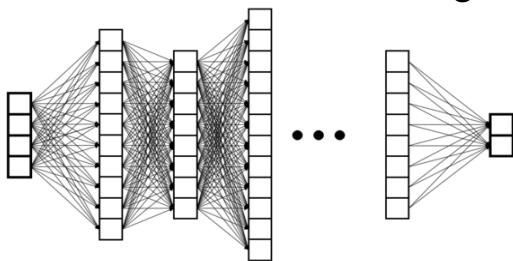
The MONAI logo, which consists of the word "MONAI" in a teal sans-serif font with a small plus sign, set against a background of a network graph with green nodes and light blue edges. Below the logo, the text "Medical Open Network for Artificial Intelligence" is written in a smaller teal font. At the bottom, there are three teal rounded rectangular buttons with white text: "Core v1.1", "Label v0.6", and "Deploy App SDK v0.5.1". A large teal banner at the bottom states "1,000,000+ downloads and counting".

The TorchMetrics library landing page. It features a purple-to-red gradient header with the title "TorchMetrics" in large white letters. Below the title, the text "Library of 90+ PyTorch metrics, optimized for distributed training" is displayed. At the bottom, there are two buttons: "pip install torchmetrics" with the number "24,621" and "Quick Start".

Picking baselines



Best performing
methods



Ablated versions of
proposed method



All baselines have to be tuned!

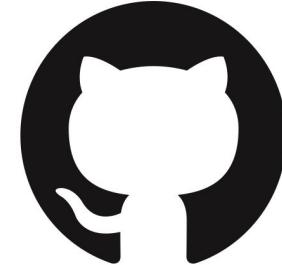
[nnU-net: Self-adapting framework for u-net-based medical image segmentation](#)

F Isensee, J Petersen, A Klein, D Zimmerer, ... - arXiv preprint arXiv ..., 2018 - arxiv.org

... For each task the **nnU-Net** automatically runs a five-fold cross-validation for three different

... in the context of the Medical Segmentation Decathlon we demonstrate that the **nnU-Net** ...

☆ Speichern 95 Zitieren Zitiert von 708 Ähnliche Artikel Alle 6 Versionen ☺



Methods with good
public code



Pre-deep learning
algorithms

Overfitting - case study 1

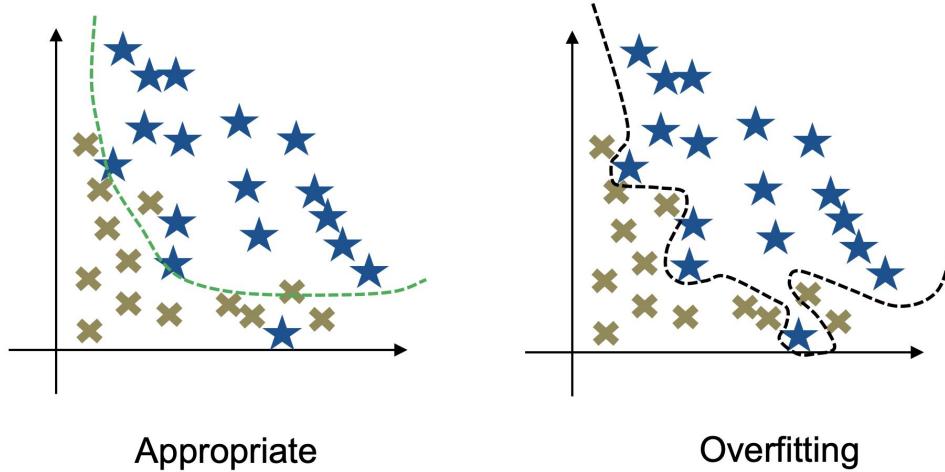
- Task: classification of skin lesion based on smart phone photographs
- Approach: train a convolutional neural network **on all data** and report performance



Problem: classical overfitting of network parameters

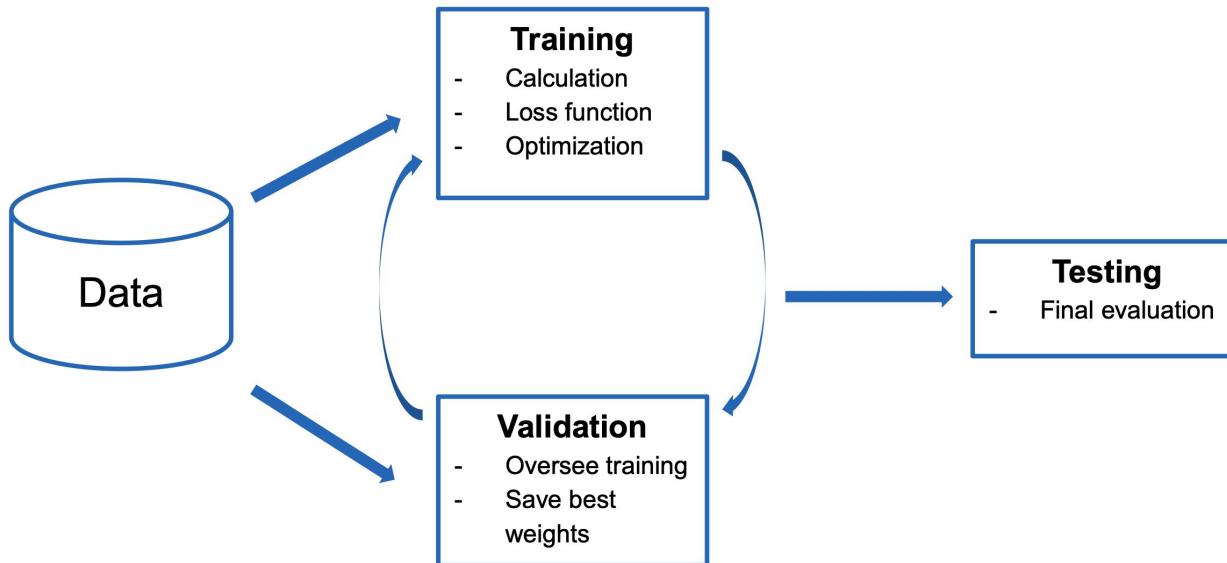
Overfitting

- Large enough models are able to memorize the entire dataset
- Model does not need to learn general patterns in the data



Overfitting

- Splitting data into training, validation and testing dataset



Overfitting - case study 2

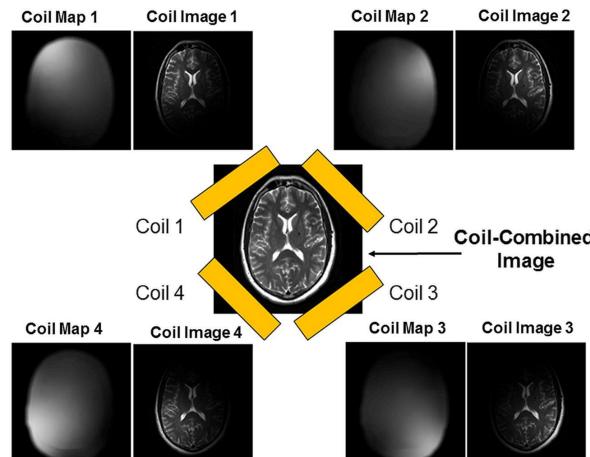
- Task: classification of skin lesion based on smart phone photographs
- Approach: train a convolutional neural network on test and validation data
- After poor test performance, add two additional convolutional blocks to your network



Problem: hyperparameter tweaking based on test performance

Overfitting - case study 3

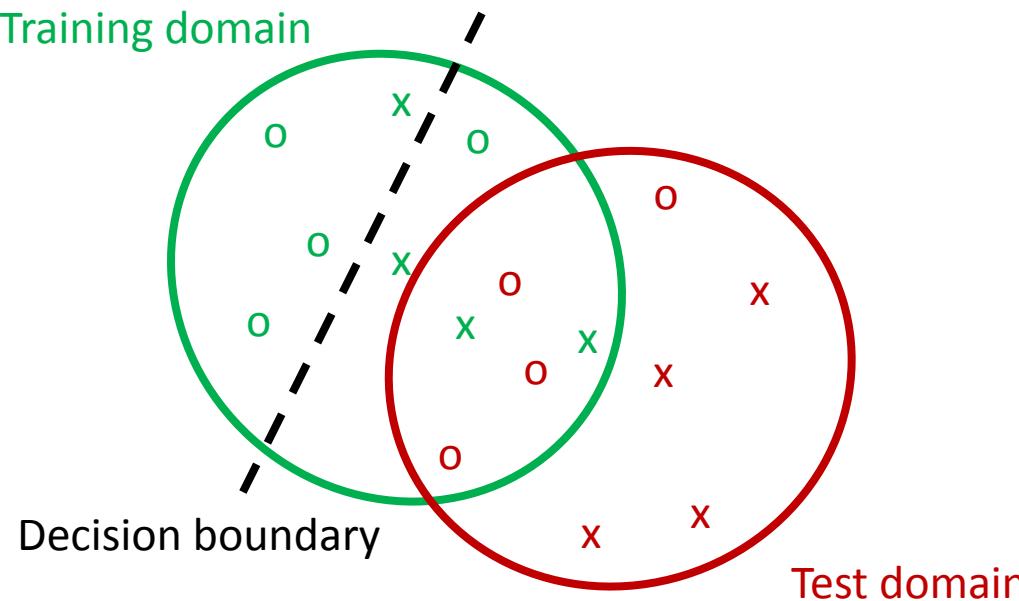
- Task: reconstruction of MR images from parallel imaging data
- Approach: train reconstruction neural network on large dataset of healthy volunteers
- Reconstruction fails when deployed on clinical data



Problem: scans in training dataset feature less artifacts and may be anatomically different

Domain shifts

- A domain shift occurs when a machine learning algorithm is trained and deployed on datasets with different characteristics



Overfitting - case study 4

- Task: forecast severity of Covid-19 infection from chest radiographs
- Approach: In order to overcome small dataset size, scans from different clinics are combined for model training



Dataset 1		Dataset 2	
Healthy	III	Healthy	III
X-ray scanner type A		X-ray scanner type B	



Problem: the algorithm learns to predict scanner type instead of disease markers

Overfitting - case study 5

- Task: predicting mortality of intensive care unit patients
- Approach: utilize rich dataset consisting of demographic data, vital sign measurements, laboratory tests, medication, caregiver notes, ...
- Algorithm fails to generalize well



Problem: the presence of a test or intervention, rather than its outcome already conveys information

Statistical tests – advantages and disadvantages

Pros

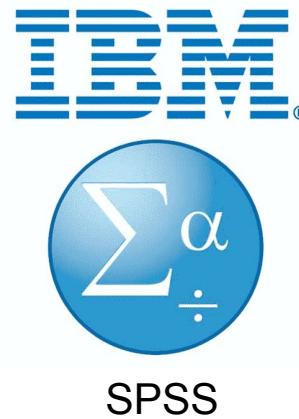
- Good scientific practice
- Makes your presented results look more impressive
- Most statistical tests used in MICCAI research can be calculated rather quickly

Cons

- Requires expertise
- Potential source for errors
- Limited sensitivity when using a small number of samples

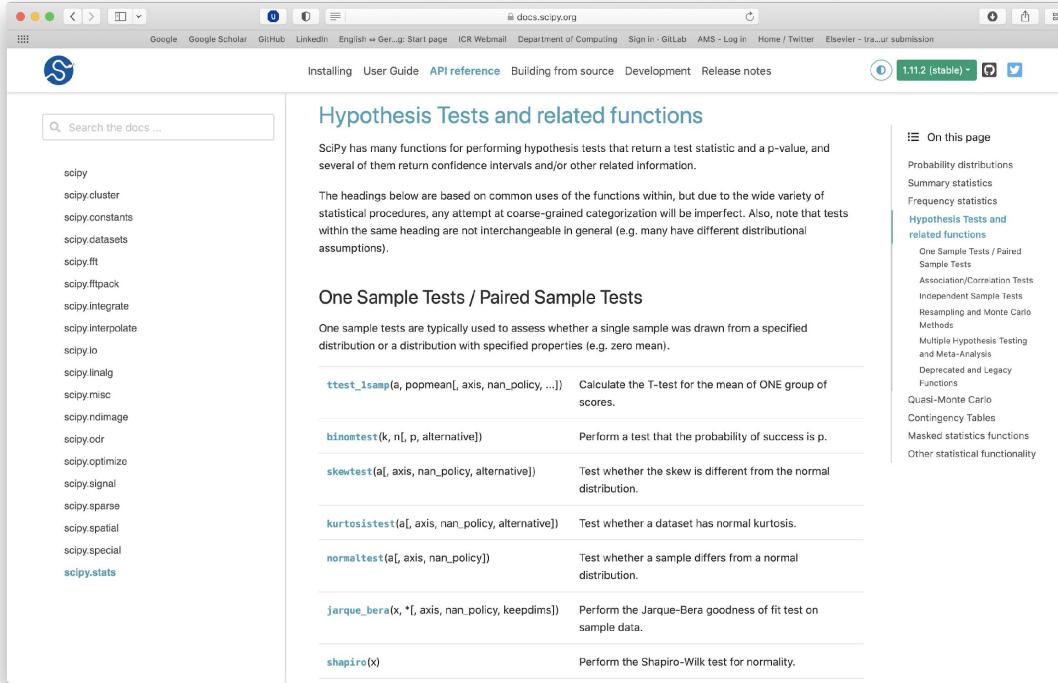
Statistical tests – software and libraries

- Dedicated statistics software is often more advanced than Python toolboxes
- Still, `scipy.stats` covers most tests needed for MICCAI research



Statistical tests – software and libraries

- Dedicated statistics software is often more advanced than Python toolboxes
- Still, `scipy.stats` covers most tests needed for MICCAI research



The screenshot shows a web browser displaying the SciPy documentation at docs.scipy.org. The page title is "Hypothesis Tests and related functions". On the left, there is a sidebar with a search bar and a list of modules: scipy, scipy.cluster, scipy.constants, scipy.datasets, scipy.fft, scipy.fftpack, scipy.integrate, scipy.interpolate, scipy.io, scipy.linalg, scipy.misc, scipy.ndimage, scipy.odr, scipy.optimize, scipy.signal, scipy.sparse, scipy.spatial, scipy.special, and scipy.stats. The main content area contains a brief introduction about hypothesis tests, followed by a section titled "One Sample Tests / Paired Sample Tests" which lists several statistical functions with their descriptions:

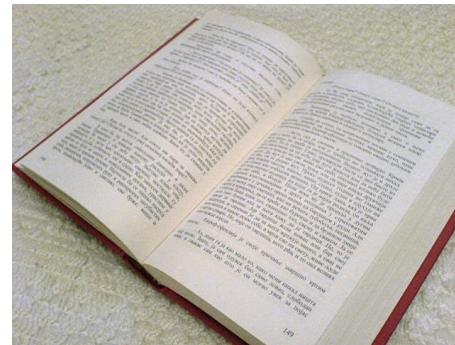
- `ttest_1samp(a, popmean[, axis, nan_policy, ...])` Calculate the T-test for the mean of ONE group of scores.
- `binomtest(k, n[, p, alternative])` Perform a test that the probability of success is p.
- `skewtest(a[, axis, nan_policy, alternative])` Test whether the skew is different from the normal distribution.
- `kurtosistest(a[, axis, nan_policy, alternative])` Test whether a dataset has normal kurtosis.
- `normaltest(a[, axis, nan_policy])` Test whether a sample differs from a normal distribution.
- `jarque_bera(x, *[, axis, nan_policy, keepdims])` Perform the Jarque-Bera goodness of fit test on sample data.
- `shapiro(x)` Perform the Shapiro-Wilk test for normality.

On the right side, there is a sidebar titled "On this page" with links to other sections: Probability distributions, Summary statistics, Frequency statistics, Hypothesis Tests and related functions (which is expanded), One Sample Tests / Paired Sample Tests, Association/Correlation Tests, Independent Sample Tests, Resampling and Monte Carlo Methods, Multiple Hypothesis Testing and Meta-Analysis, Deprecated and Legacy Functions, Quasi-Monte Carlo, Contingency Tables, Masked statistics functions, and Other statistical functionality.

Statistical tests – how to get started



Enroll in a university course



Read a statistics book



Check what others have done

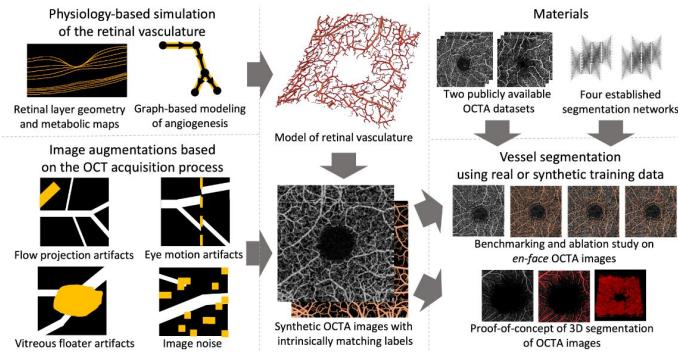


Ask a friend or colleague

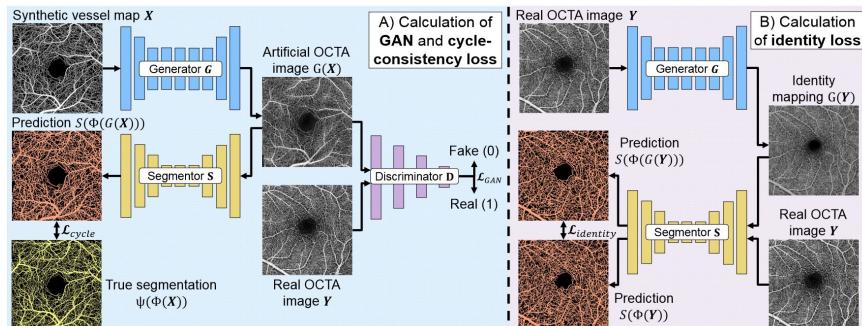


Browse the internet

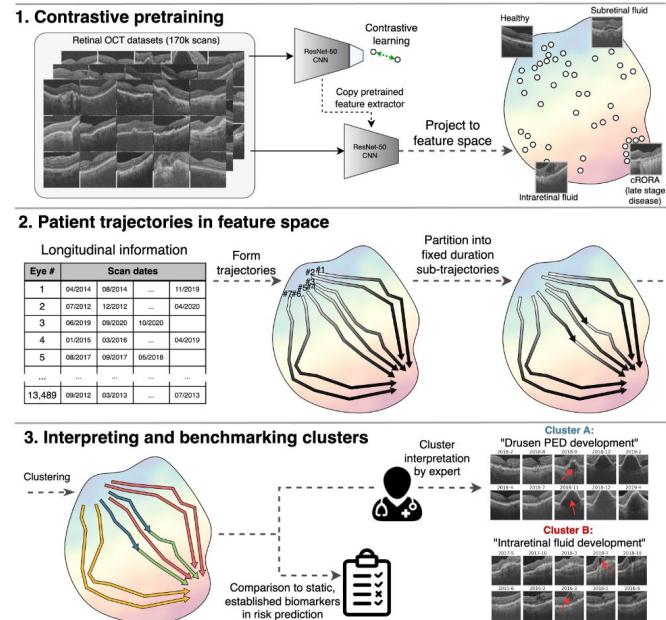
Types of scientific figures - introductory and methods figures



Graphical abstract



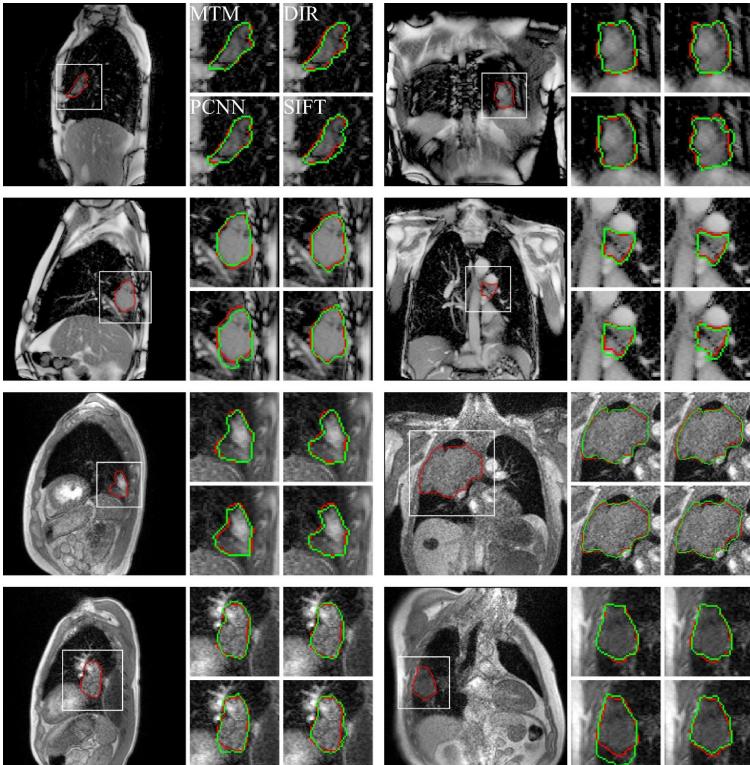
Methods figure



Study flowchart

Types of scientific figures - qualitative results

- Provides intuition about methods' strengths and weaknesses
- Consistently crop and set contrast
- Use high-enough image resolution
- Mention whether you are presenting best, worst, representative or randomly selected samples

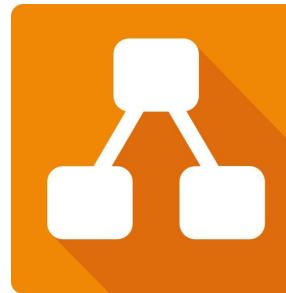


WYSIWYG tools

- What-you-see-is-what-you get makes it easier to prototype
- Rapid iteration of figures may be needed



PowerPoint



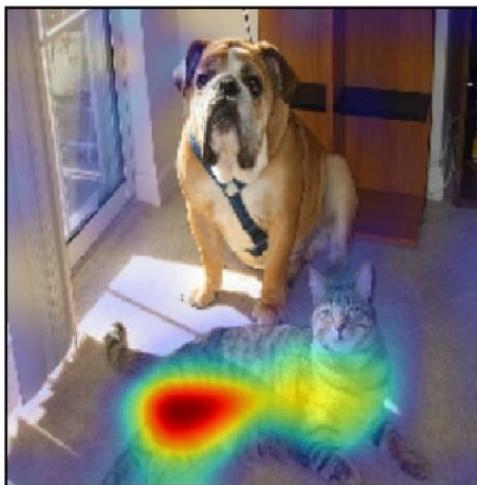
Draw.io



Gimp

Types of scientific figures - saliency maps

- Visualization of image features used by deep neural networks
- Most established one is GradCAM



arXiv:1610.02391v4 [cs.CV] 3 Dec 2019

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra

Abstract We propose a technique for producing ‘visual explanations’ for decisions from a large class of Convolutional Neural Network (CNN)-based models, making them more transparent and explainable.

Our approach – Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept (say ‘dog’) in a classification network or a sequence of words in captioning network flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. Unlike previous approaches, Grad-CAM is applicable to a wide variety of CNN model-families: (1) CNNs with fully-connected layers (e.g., VGG), (2) CNNs used for structured outputs (e.g., captioning), (3) CNNs used in tasks with multi-modal inputs (e.g., visual question answering) or reinforcement learning, all *without architectural changes or re-training*. We combine Grad-CAM with existing fine-grained visualizations to create a high-resolution class-discriminative vi-

Ramprasaath R. Selvaraju
Georgia Institute of Technology, Atlanta, GA, USA
E-mail: ramps@gatech.edu

Michael Cogswell
Georgia Institute of Technology, Atlanta, GA, USA
E-mail: cogswell@gatech.edu

Abhishek Das
Georgia Institute of Technology, Atlanta, GA, USA
E-mail: abhskd@gatech.edu

Ramakrishna Vedantam
Georgia Institute of Technology, Atlanta, GA, USA
E-mail: vrma@gatech.edu

Devi Parikh
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA
E-mail: parikh@gatech.edu

Dhruv Batra
Georgia Institute of Technology, Atlanta, GA, USA
Facebook AI Research, Menlo Park, CA, USA
E-mail: dbatra@gatech.edu

ualization, Guided Grad-CAM, and apply it to image classification, image captioning, and visual question answering (VQA) models, including ResNet-based architectures.

In the context of image classification models, our visualizations (a) lend insights into failure modes of these models (showing that seemingly unreasonable predictions have reasonable explanations), (b) outperform previous methods on the ILSVRC-15 weakly-supervised localization task, (c) are robust to adversarial perturbations, (d) are more faithful to the underlying model, and (e) help achieve model generalization by identifying dataset bias.

For image captioning and VQA, our visualizations show that even non-attention based models learn to localize discriminative regions of input image.

We devise a way to identify important neurons through Grad-CAM and combine it with neuron names [4] to provide textual explanations for model decisions. Finally, we design and conduct human studies to measure if Grad-CAM explanations help users establish appropriate trust in predictions from deep networks and show that Grad-CAM helps untrained users successfully discern a ‘stronger’ deep network from a ‘weaker’ one even when both make identical predictions. Our code is available at <https://github.com/ramps/grad-cam/>, along with a demo on CloudCV [2]¹, and a video at youtu.be/COJuB91zk6E.

1 Introduction

Deep neural models based on Convolutional Neural Networks (CNNs) have enabled unprecedented breakthroughs in a variety of computer vision tasks, from image classification [33, 24], object detection [21], semantic segmentation [37] to image captioning [55, 7, 18, 29], visual question answering [3, 20, 42, 46] and more recently, visual dialog [11, 12] and embodied question answering [10, 23]. While

¹ <http://gradcam.cloudcv.org>

Types of scientific figures - the big results table

OCTA-500	Model	Trained on	Augmentation	DSC	clDice	AUC	ACC	Recall	Precision
Supervised	U-Net	Real	Random	0.912±0.001	0.940±0.002	0.950±0.001	0.984±0.000	0.910±0.003	0.916±0.003
Traditional	Frangi OOF	-	-	0.807±0.003 0.734±0.004	0.848±0.004 0.785±0.003	0.895±0.002 0.851±0.004	0.975±0.000 0.966±0.001	0.802±0.005 0.719±0.009	0.820±0.003 0.760±0.006
Menten <i>et al.</i>	U-Net	Synthetic	Random	0.594±0.011	0.536±0.014	0.936±0.002	0.913±0.004	0.963±0.001	0.434±0.011
Ours	U-Net	Synthetic	-	0.734±0.006 0.849±0.001	0.731±0.009 0.892±0.002	0.827±0.005 0.926±0.002	0.968±0.001 0.971±0.000	0.663±0.010 0.871±0.005	0.827±0.003 0.831±0.004
			Random	0.840±0.003	0.881±0.002	0.915±0.003	0.970±0.000	0.848±0.006	0.837±0.004
			Adversarial	0.840±0.005	0.876±0.006	0.907±0.002	0.971±0.001	0.829±0.005	0.855±0.010
			GAN						
ROSE-1	Model	Trained on	Augmentation	DSC	clDice	AUC	ACC	Recall	Precision
Supervised	U-Net	Real	Random	0.717±0.004	0.700±0.006	0.799±0.003	0.843±0.003	0.678±0.006	0.774±0.005
Traditional	Frangi OOF	-	-	0.670±0.003 0.594±0.003	0.601±0.007 0.559±0.003	0.789±0.003 0.752±0.004	0.875±0.002 0.831±0.006	0.644±0.009 0.623±0.016	0.704±0.012 0.578±0.018
Menten <i>et al.</i>	U-Net	Synthetic	Random	0.606±0.004	0.534±0.005	0.746±0.003	0.861±0.001	0.556±0.008	0.673±0.004
Ours	U-Net	Synthetic	-	0.613±0.004 0.649±0.001 0.665±0.004 0.637±0.005	0.546±0.004 0.653±0.002 0.658±0.005 0.639±0.007	0.743±0.002 0.754±0.001 0.756±0.003 0.745±0.003	0.870±0.002 0.806±0.001 0.816±0.003 0.804±0.004	0.532±0.003 0.617±0.002 0.601±0.006 0.591±0.005	0.731±0.006 0.699±0.002 0.734±0.002 0.705±0.005
			Random						
			Adversarial						
			GAN						
Giarratano <i>et al.</i>	Model	Trained on	Augmentation	DSC	clDice	AUC	ACC	Recall	Precision
Supervised	U-Net	Real	Random	0.907±0.002	0.954±0.003	0.887±0.004	0.895±0.002	0.925±0.003	0.981±0.006
Traditional	Frangi OOF	-	-	0.769±0.009 0.812±0.005	0.833±0.007 0.848±0.006	0.815±0.002 0.851±0.003	0.797±0.003 0.854±0.002	0.895±0.010 0.808±0.017	0.683±0.012 0.827±0.019
Menten <i>et al.</i>	U-Net	Synthetic	Random	0.471±0.014	0.496±0.012	0.651±0.017	0.710±0.011	0.318±0.014	0.937±0.004
Ours	U-Net	Synthetic	-	0.781±0.006 0.850±0.005 0.842±0.005 0.834±0.002	0.808±0.008 0.933±0.002 0.931±0.008 0.902±0.004	0.827±0.009 0.846±0.001 0.840±0.004 0.829±0.002	0.812±0.003 0.839±0.001 0.833±0.006 0.822±0.002	0.887±0.003 0.829±0.003 0.814±0.004 0.828±0.004	0.710±0.012 0.879±0.003 0.881±0.010 0.853±0.006
			Random						
			Adversarial						
			GAN						

Types of scientific figures - the big results table

- Usually done in LaTeX
- Sparingly use separating lines
- Add hierarchy with multicolumn and/ or multirow if needed
- Consistent number of digits
- Align numbers
- Remember to update table with new results



tabulate 0.9.0

[pip install tabulate](#)

Released: Oct 6, 2022

Pretty-print tabular data

Navigation

- [Project description](#) (selected)
- [Release history](#)
- [Download files](#)

Project links

- [Homepage](#)

Statistics

GitHub statistics:

- ★ Stars: 1763
- 🍴 Forks: 162
- Open issues: 67
- Open PRs: 23

View statistics for this project via [Libraries.io](#), or by using [our public dataset on Google BigQuery](#).

Meta

License: MIT License (MIT)
Author: [Sergey Astanin](#)
Requires: Python >=3.7

Maintainers

 Sergey

Project description

python-tabulate

Pretty-print tabular data in Python, a library and a command-line utility.

The main use cases of the library are:

- printing small tables without hassle: just one function call, formatting is guided by the data itself
- authoring tabular data for lightweight plain-text markup: multiple output formats suitable for further editing or transformation
- readable presentation of mixed textual and numeric data: smart column alignment, configurable number formatting, alignment by a decimal point

Installation

To install the Python library and the command line utility, run:

```
pip install tabulate
```

The command line utility will be installed as `tabulate` to `/bin` on Linux (e.g. `/usr/bin`) or as `tabulate.exe` to `Scripts` in your Python installation on Windows (e.g. `C:\Python39\Scripts\tabulate.exe`).
You may consider installing the library only for the current user:

```
pip install tabulate --user
```

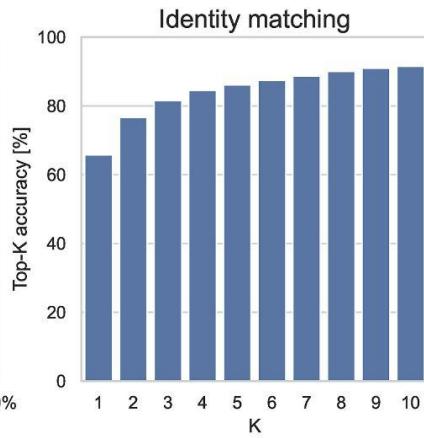
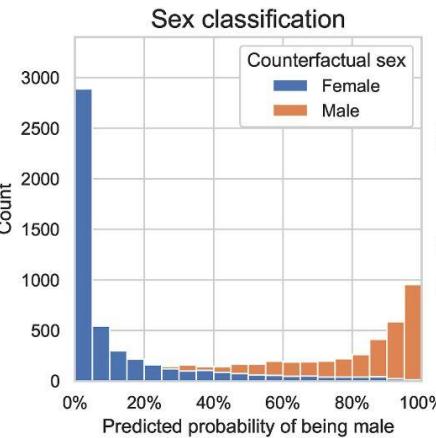
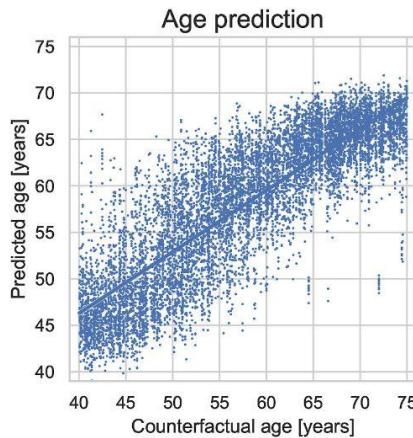
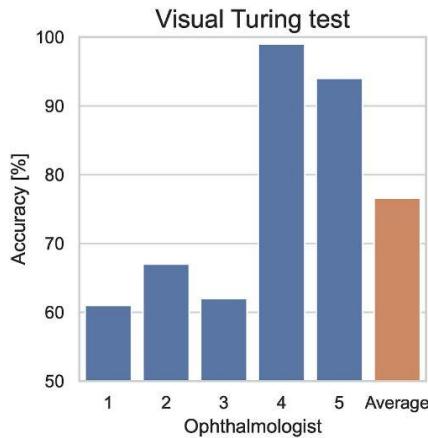
In this case the command line utility will be installed to `~/local/bin/tabulate` on Linux and to `\APPDATA\Python\Scripts\tabulate.exe` on Windows.
To install just the library on Unix-like operating systems:

```
TABULATE_INSTALL=lib-only pip install tabulate
```

On Windows:

Types of scientific figures - quantitative results figures

- Conveys high-dimensional data in an intuitive way
- Type of plot can be used to emphasize different aspects of the findings



Python plotting libraries

- Dedicated libraries for scientific figure plotting
- Can be directly integrated with the output of your algorithm



Matplotlib



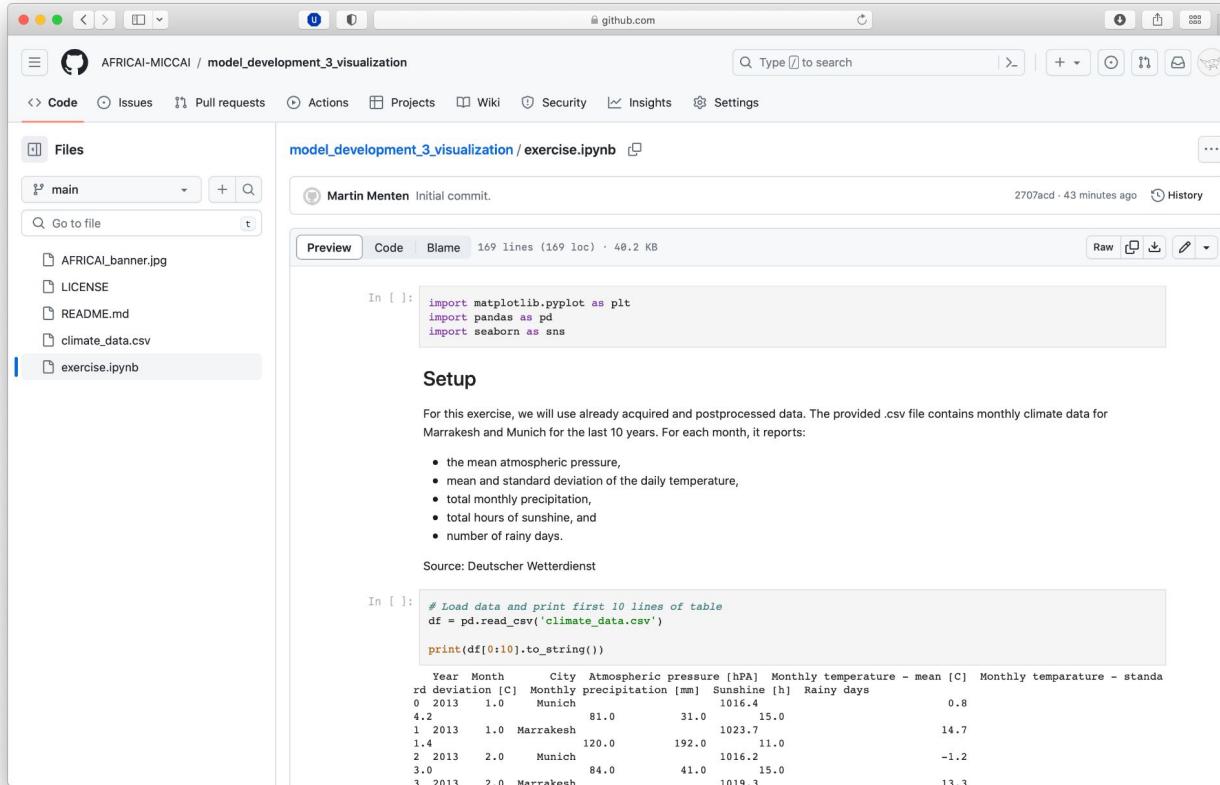
seaborn

Common mistakes when making results figures

- No clear purpose
- Too easy to understand
- Too difficult to understand
- Too small font
- Too large font
- Too many colors
- Too few colors
- Lack of consistency
- Typos



Interactive: make your own results figures



The screenshot shows a Jupyter Notebook interface within a GitHub repository. The repository is titled "AFRICAI-MICCAI / model_development_3_visualization". The notebook file "exercise.ipynb" is open. The code cell contains:

```
In [ ]: import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

The "Setup" section provides instructions for the exercise, stating:

For this exercise, we will use already acquired and postprocessed data. The provided .csv file contains monthly climate data for Marrakesh and Munich for the last 10 years. For each month, it reports:

- the mean atmospheric pressure,
- mean and standard deviation of the daily temperature,
- total monthly precipitation,
- total hours of sunshine, and
- number of rainy days.

Source: Deutscher Wetterdienst

The next code cell displays the first 10 lines of the "climate_data.csv" file:

```
In [ ]: # Load data and print first 10 lines of table
df = pd.read_csv('climate_data.csv')

print(df[0:10].to_string())
```

Year	Month	City	Atmospheric pressure [hPa]	Monthly temperature - mean [C]	Monthly temperature - standard deviation [C]	Monthly precipitation [mm]	Sunshine [h]	Rainy days	Source	
0	2013	1.0	Munich	1016.4	4.2	81.0	31.0	15.0	0.8	Deutscher Wetterdienst
1	2013	1.0	Marrakesh	1023.7	1.4	120.0	192.0	11.0	14.7	
2	2013	2.0	Munich	1016.2	3.0	84.0	41.0	15.0	-1.2	
3	2013	2.0	Marrakesh	1019.3					13.3	

Five takeaways

- Formulate and think about your research question as **testable hypothesis**
- Set up your **evaluation pipeline** including metrics and baselines early
- Beware of **overfitting**, which comes in many forms and shapes
- Include **statistical analysis** if you can
- Carefully design your **figures as they carry the results section**

List of resources

- <https://metrics-reloaded.dkfz.de/>
- <https://monai.io/>
- <https://lightning.ai/torchmetrics>
- <https://docs.scipy.org/doc/scipy/reference/stats.html>
- <https://stats.stackexchange.com/>
- <https://www.drawio.com/>
- <https://www.gimp.org/>
- <https://pypi.org/project/tabulate/>
- <https://matplotlib.org/>
- <https://seaborn.pydata.org/>