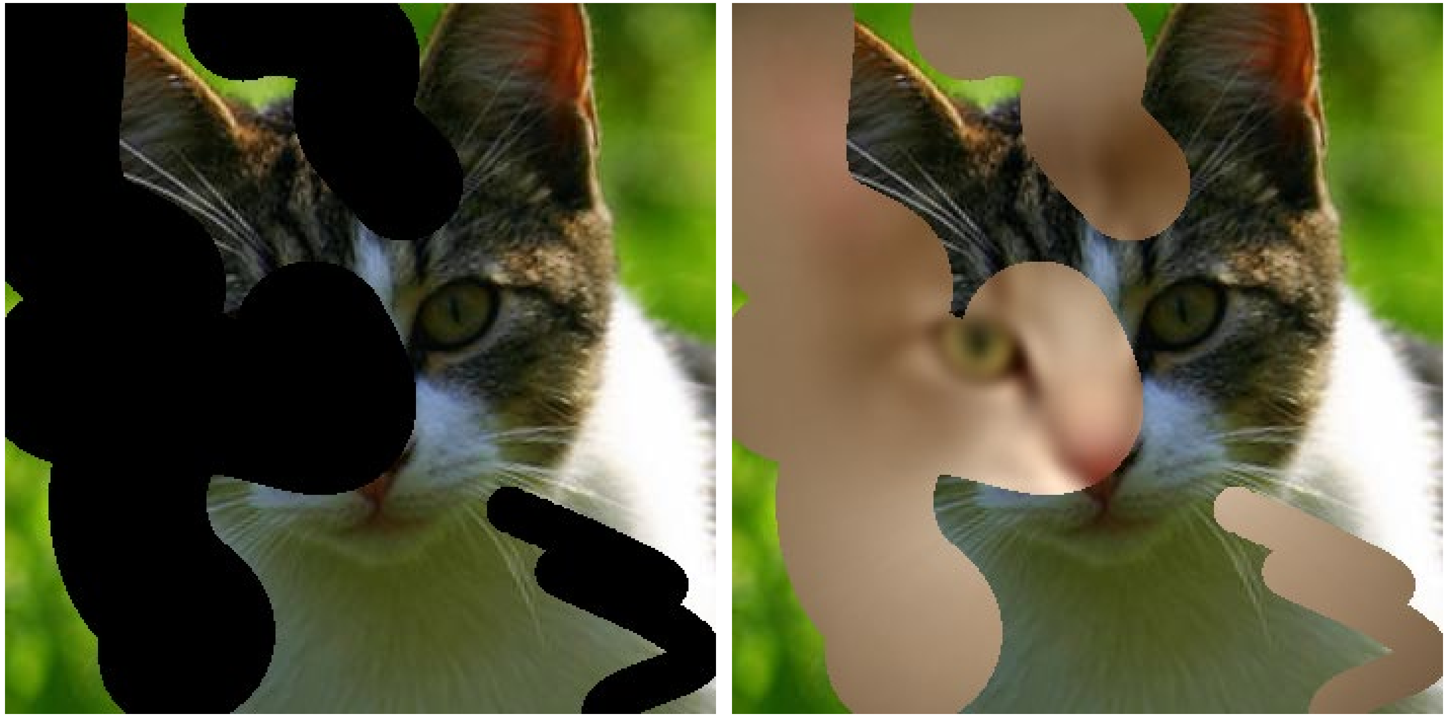
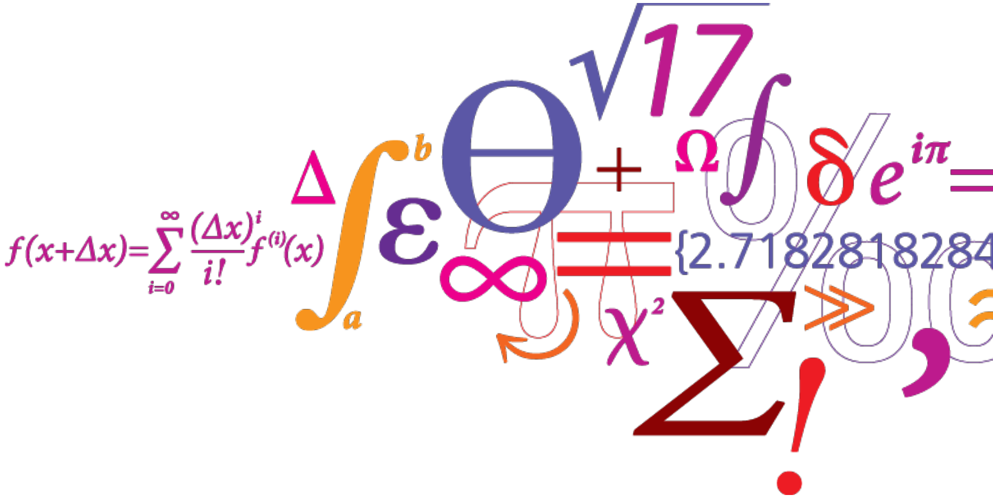
InPaint the cat challenge 2023



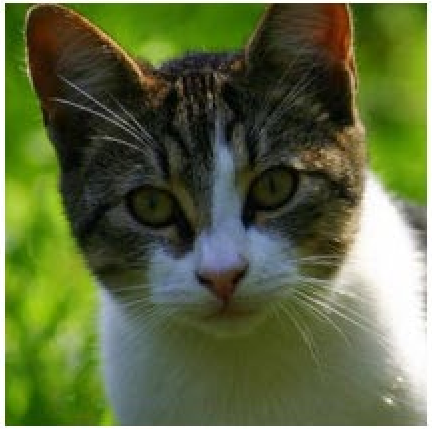
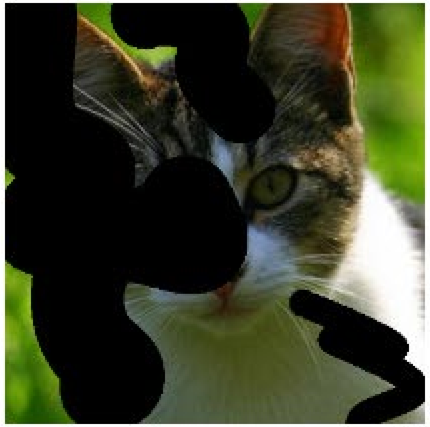
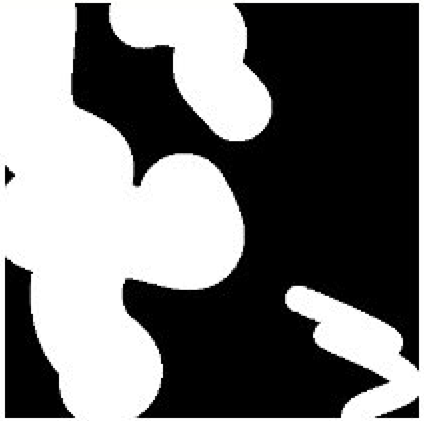
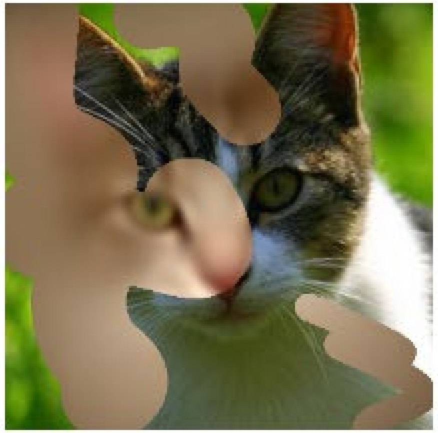


DTU Compute

Department of Applied Mathematics and Computer Science

InPaint the cat challenge

- The goal is to create an algorithm that can fill in missing parts in an image
- The missing part is known and given as a mask
- We provide you with a set of template functions that you should extend with your ideas and methods

Original	Stroke Masked	Stroke Mask	Inpainted
			

Why is this relevant?

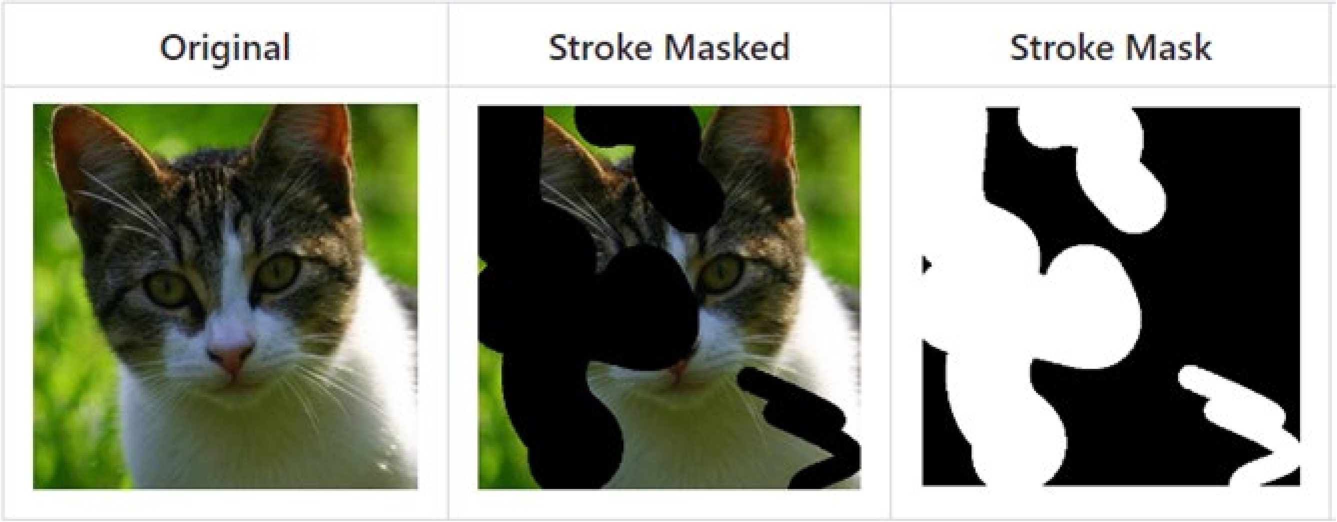
- Because we love cat photos and we can not stand seeing an incomplete photo of a cat?

Why is this relevant?

- It is a well known problem within missing-data
- Several approaches used in image painting can also be used in other missing data scenarios like tabular data
- Lots of different ways to approach the problem:
  - Geometric (inpaint using nearest geometric neighbor, Fast Marching)
  - Statistical (mean image, PCA)
  - Physical (Navier-Stokes)
  - Registration based (morph template / active appearance models)
  - Deep learning (autoencoders, diffusion models)

Data

- RGB images of 360 x 360
- Aligned using the nose and eyes
- Sets (image ids are specified in the data_splits text files):
  - **Training:** more than 4900 with original + masked images + masks
  - **Validation:** Original + masked images + masks
  - **Test:** Masked images + masks
  - **Final test:** Masked images + masks



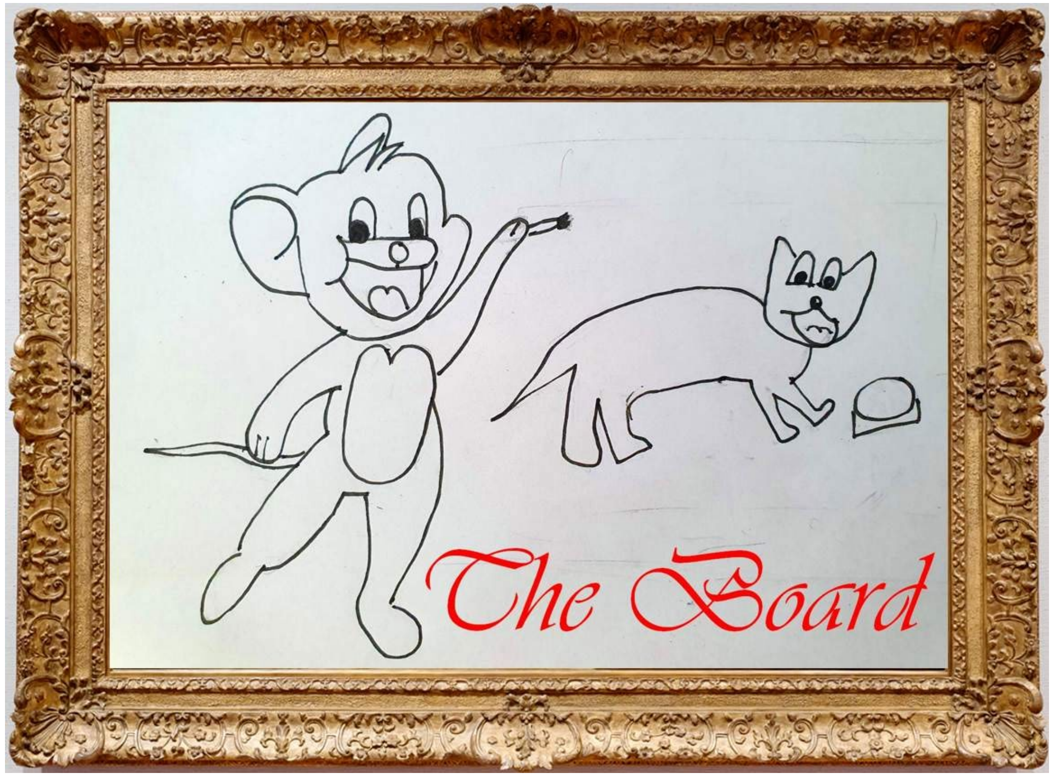
Teams

- We have pre-made 15 teams
- You should add your name and email to a team on the paper in the conference room
- Try to distribute yourself evenly
- You are responsible for gathering and organizing your team

Getting starting with the Python code

- Clone the GitHub repository
  - https://github.com/RasmusRPaulsen/MissingDataChallenge
- Download the data (if not already done)
- Change your team name in the JSON configuration file
- Follow the instruction on the GitHub repository

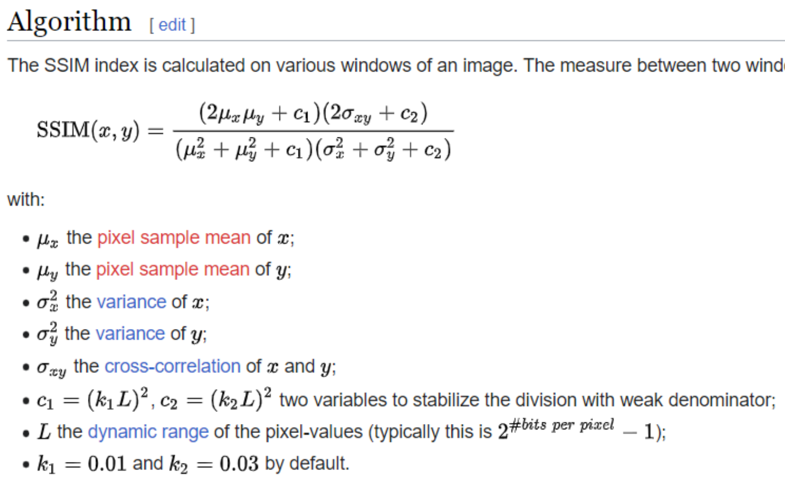
Challenge results



- The team results are computed and put on the homepage several times a day
- http://fungi.compute.dtu.dk:8080/
- Submit results to the challenge server, at least, two times daily
- You should submit the final set scores latest Thursday at 16h!

SSIM Measure

- We use the mean structural similarity measure (SSIM) to compare an InPainted image with the original image
- SSIM is a standard measure to compare the perceived difference between images
- We use the default implementation used in scikit image:
- https://scikit-image.org/docs/stable/api/skimage.metrics.html#skimage.metrics.structural_similarity
- From Wikipedia:



Algorithm [edit]

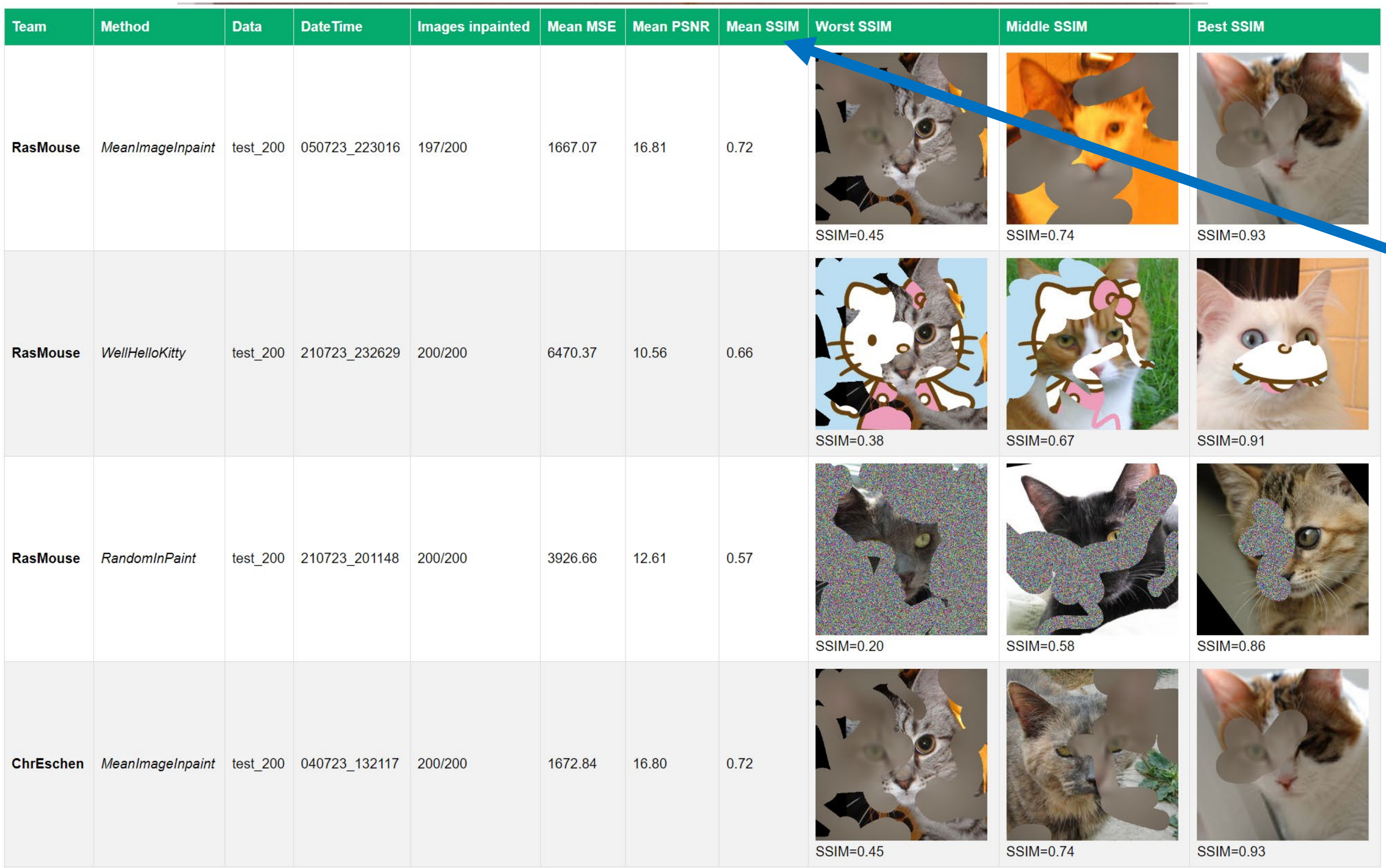
The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is:[4]

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with:

- μ_x the pixel sample mean of x;
- μ_y the pixel sample mean of y;
- σ_x^2 the variance of x;
- σ_y^2 the variance of y;
- σ_{xy} the cross-correlation of x and y;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator;
- L the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$);
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

The board – some baseline results



Team	Method	Data	DateTime	Images inpainted	Mean MSE	Mean PSNR	Mean SSIM	Worst SSIM	Middle SSIM	Best SSIM
RasMouse	*MeanImageInpaint*	test_200	050723_223016	197/200	1667.07	16.81	0.72	SSIM=0.45	SSIM=0.74	SSIM=0.93
RasMouse	*WellHelloKitty*	test_200	210723_232629	200/200	6470.37	10.56	0.66	SSIM=0.38	SSIM=0.67	SSIM=0.91
RasMouse	*RandomInPaint*	test_200	210723_201148	200/200	3926.66	12.61	0.57	SSIM=0.20	SSIM=0.58	SSIM=0.86
ChrEschen	*MeanImageInpaint*	test_200	040723_132117	200/200	1672.84	16.80	0.72	SSIM=0.45	SSIM=0.74	SSIM=0.93

Press here to sort by that column

Presentations and results

- The final project presentations and results are on Thursday from 17:10-18:45

- Each team has 4 minutes to present their project with maximum 3 slides
  - Brief review of your approach
  - Did it work as expected?

- Finally, the **final test** results are presented by the organizers

Guide and tips

- Use, at least, one non-deep learning method first

- Divide the data with ground truth into your own train / validation splits

Rules

- We are not checking for cheating and believe in fair play and that you are here to learn

- We do not recommend you to:
  - Use other teams names
  - Get images and inpaintings from external sites
  - Hack or modify the submission script

Have fun!