MaxiMask: Identifying contaminants in astronomical images using convolutional neural networks

ESO - 25th July 2019

Maxime Paillassa[1], Emmanuel Bertin[2], Hervé Bouy[1]

[1]Laboratoire d'astrophysique de Bordeaux, Univ. Bordeaux, CNRS, B18N, allée Geoffrey Saint-Hilaire, 33615 Pessac, France
[2]Sorbonne Université, CNRS, UMR 7095, Institut d'Astrophysique de Paris, 98 bis bd Arago, 75014 Paris, France

Table of contents

Context: source detection

Detection in astronomy

- Essential to have high completeness and low contamination for astronomical research using catalogues:
  - Main limitations in large scale surveys
  - Major sources of noise and errors in scientific analysis
- Challenging task because of:
  - Wide variety of images
  - Numerous contaminants

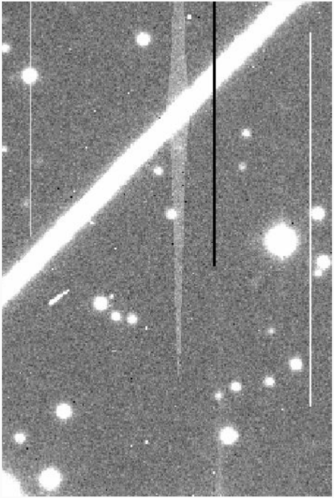


Figure 1: Example of image containing several contaminants

SExtractor source detection

- SExtractor [2]: widely used software for detection
  - Background subtraction, matched-filtering, thresholding and deblending

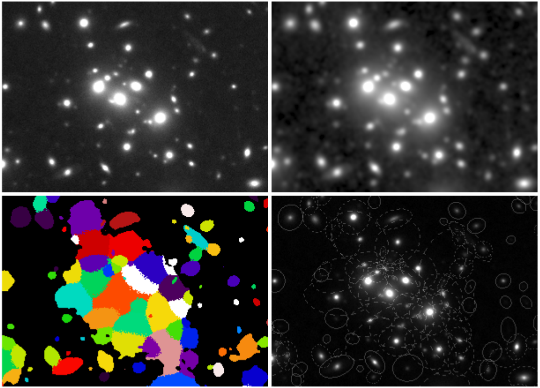


Figure 2: Example of SExtractor processing. Top left: source image, Top right: background subtraction and match-filtering, Bottom left: thresholding and deblending, Bottom right: final image with source shapes

Problems

- Problems of current detection method in SExtractor:
  - Heuristic-based filtering, thresholding and deblending
    $\rightarrow$ Lot of parameters to tune to be **universal**
  - Lack of robustness regarding contaminants
    $\rightarrow$ Need other tools to be **robust**

$\rightarrow$ Extending the heuristic-based method to convolution neural networks for automatic “intelligent” image segmentation

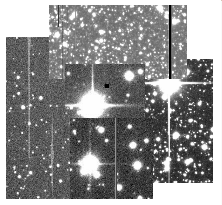
- Neural networks:
  - Robust and versatile tools
  - Exploit our amounts of data

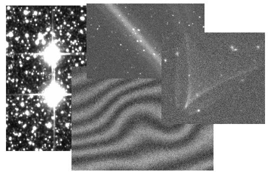
$\rightarrow$ Robust detection of astronomical sources using convolutional neural networks.

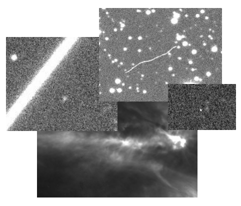
$\rightarrow$ First step: identifying contaminants in images.

Contaminants in astronomical images

- Electronic
  - Hot/Dead pixels
  - Saturated pixels
  - Persistence effects
- Optic
  - Fringes patterns
  - Diffractions spikes
  - Reflection/Refractions
- External events
  - Cosmic rays
  - Satellite trails
  - Nebulosities







MaxiMask

Building the input/output dataset

- Identify the cleanest images of our data (Cosmic Dance survey [3]) and extract and *clean* images from it

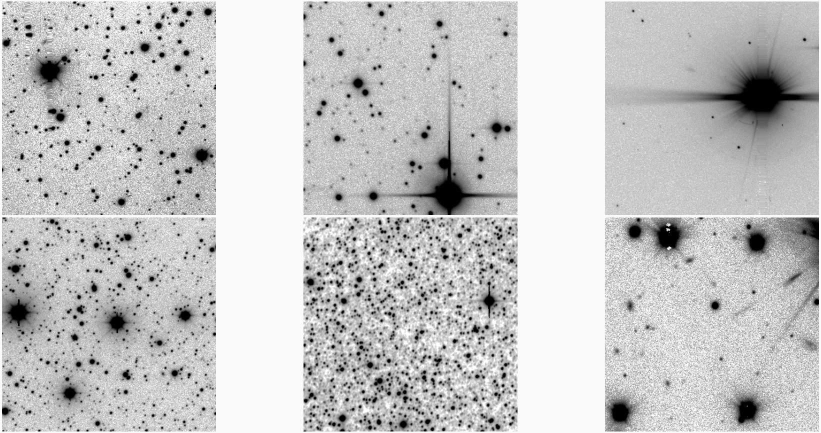
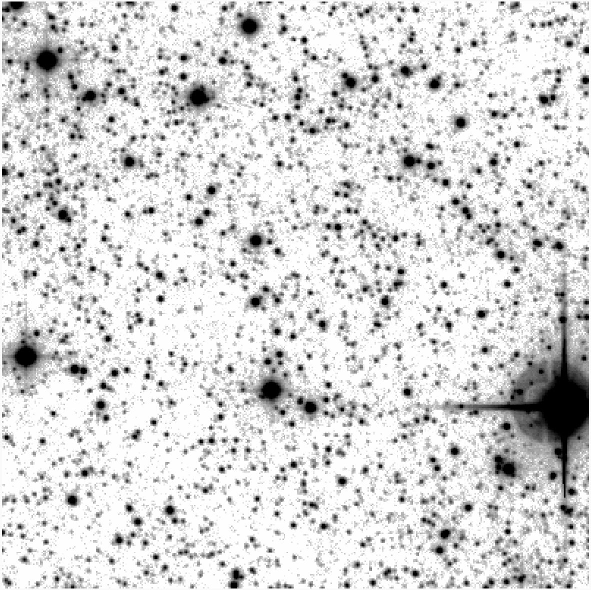
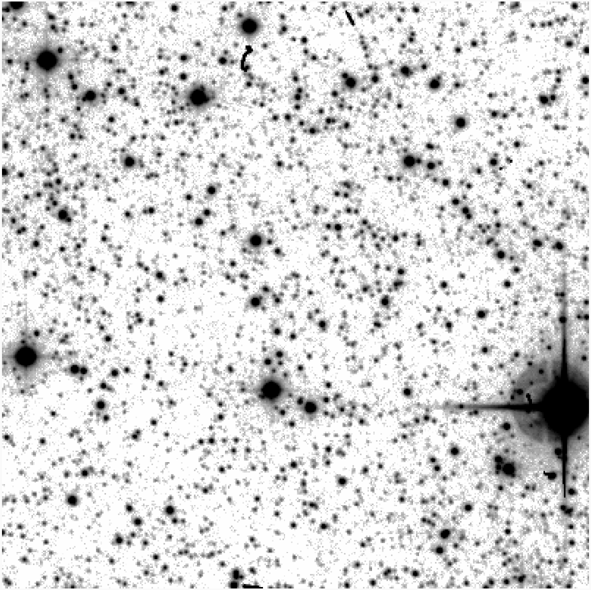


Figure 3: Left: CTIO-DECam. Center: CFHT-MegaCam. Right: HSC.

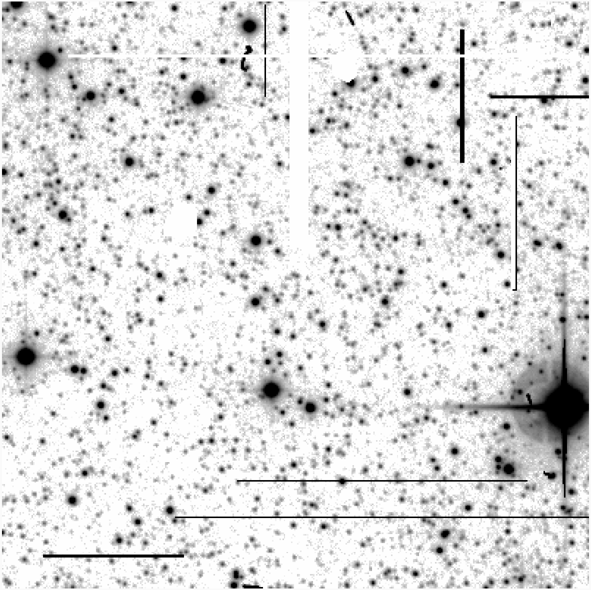
Start from a clean image



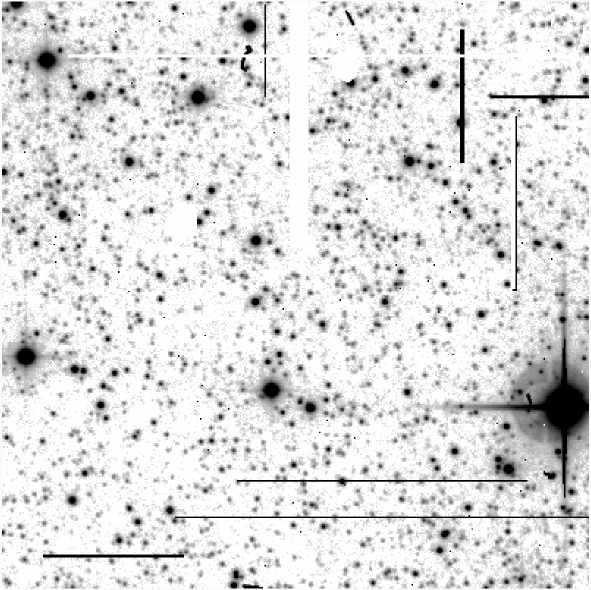
Add contaminants (1/8)



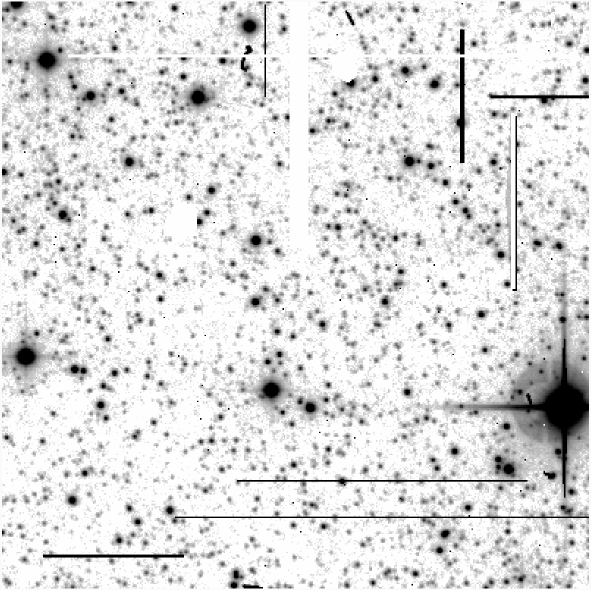
Add contaminants (2/8)



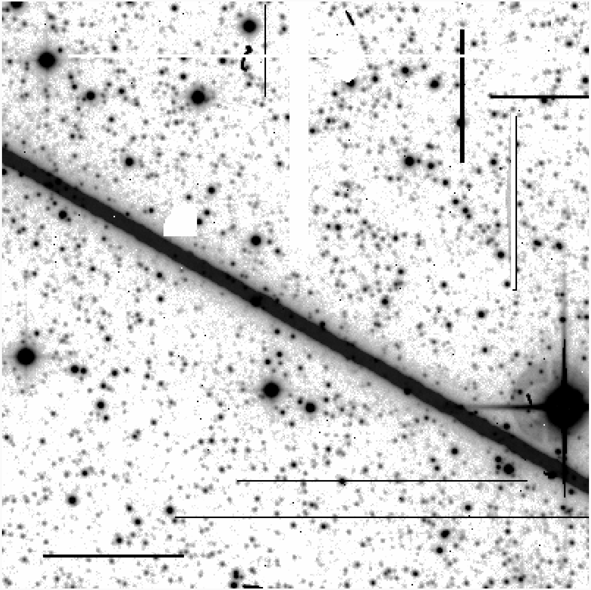
Add contaminants (3/8)



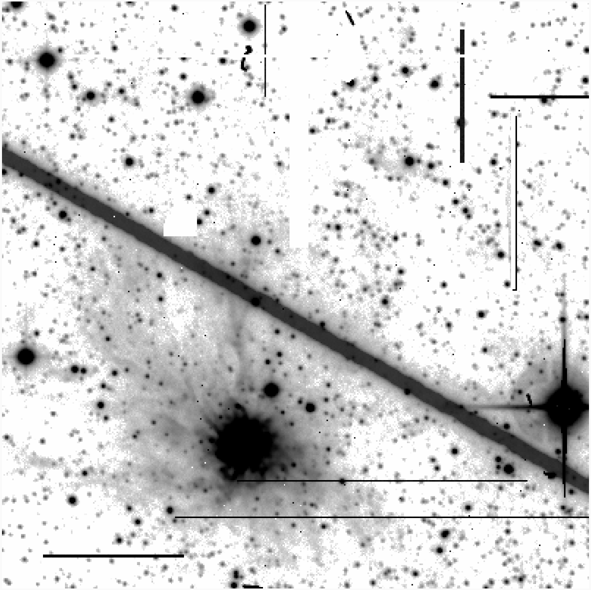
Add contaminants (4/8)



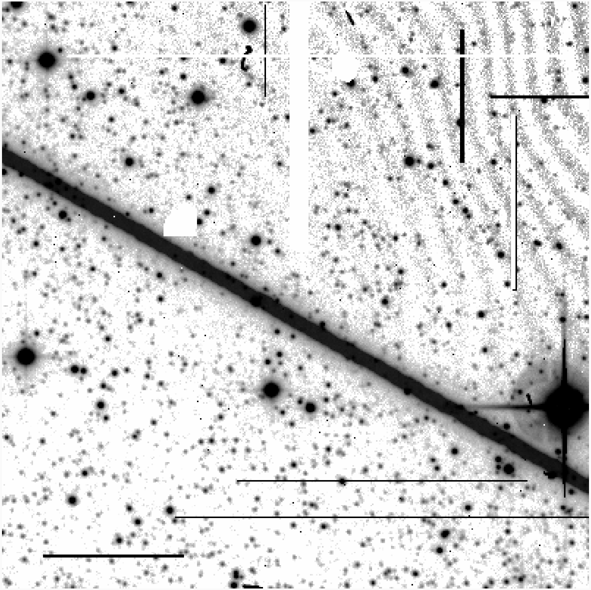
Add contaminants (5/8)



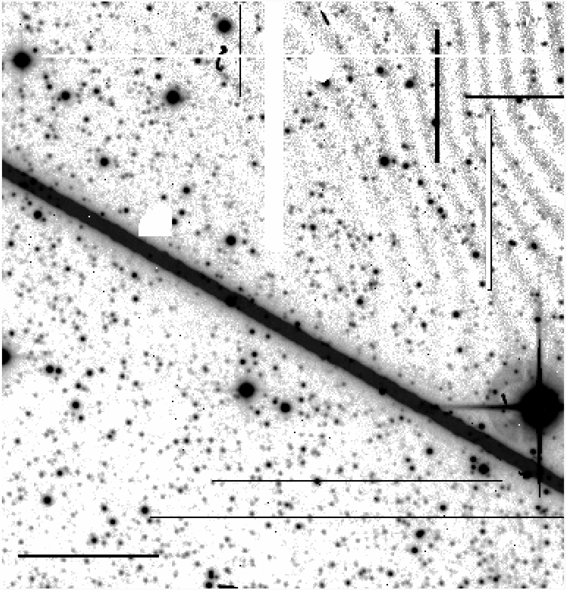
Add contaminants (6/8)



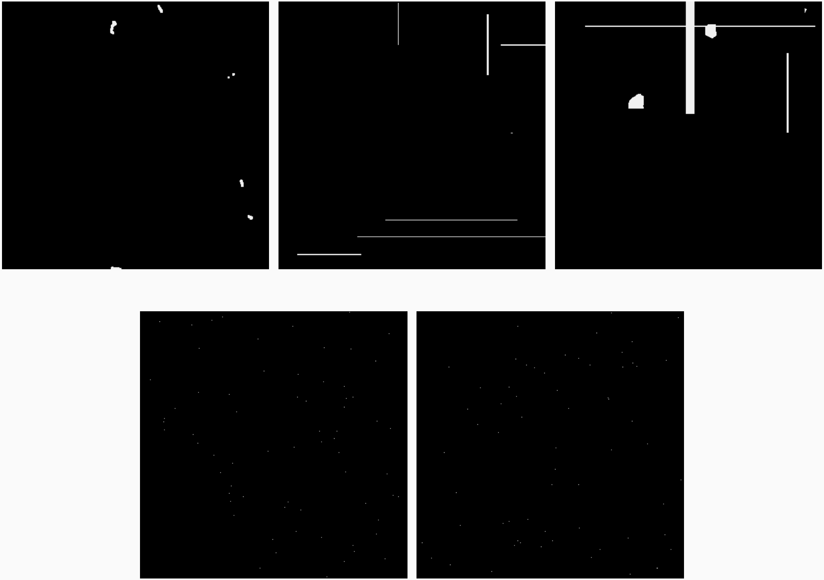
Add contaminants (7/8)



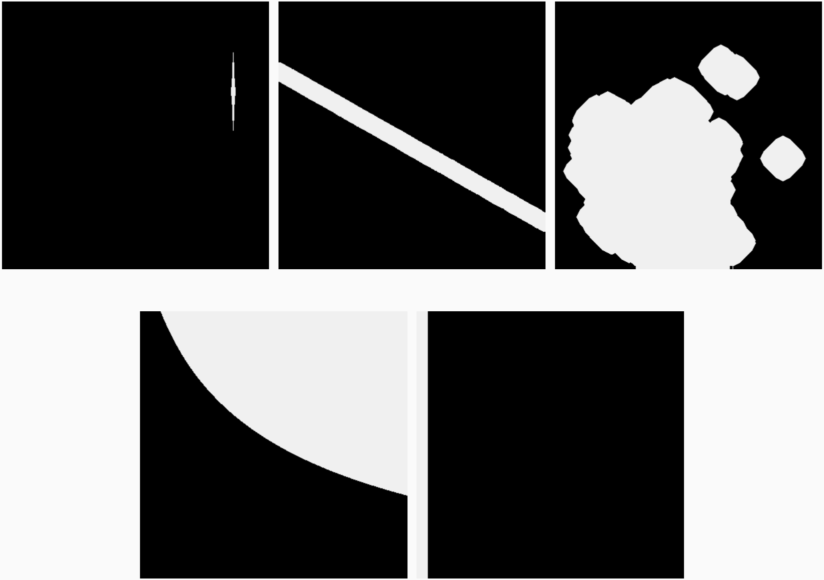
Add contaminants (8/8)



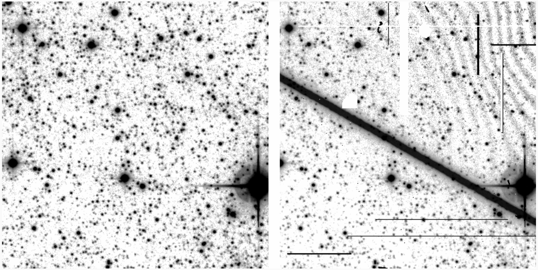
Ground truth masks (1/2)

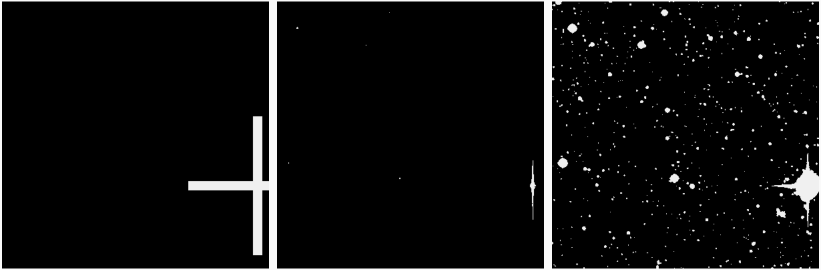


Ground truth masks (2/2)



Inherent-to-data contaminants





Learning data samples

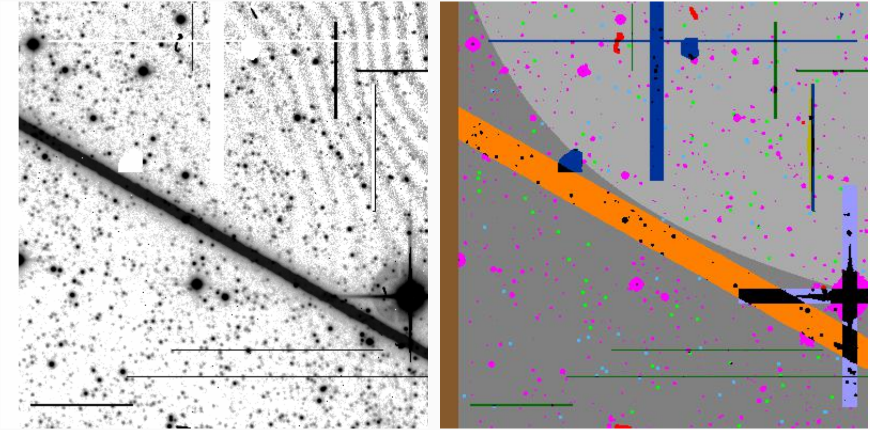


Figure 4: Left: input image. Right: ground truth.

MaxiMask CNN model

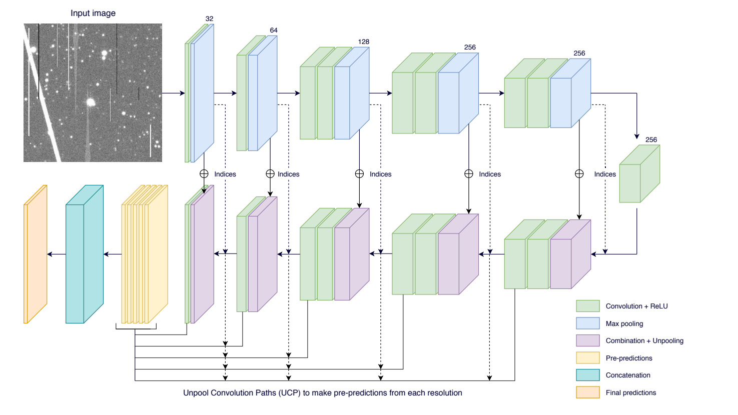


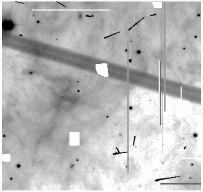
Figure 5: MaxiMask CNN model architecture [6]

- MaxiMask outputs one probability map for each given contaminant

MaxiMask training

- 50 000 images 400 × 400
- 30 epochs, Adam optimizer [4]
- Tensorflow [1]
- Nvidia TITAN X GPU

→ Approximately 24 hours



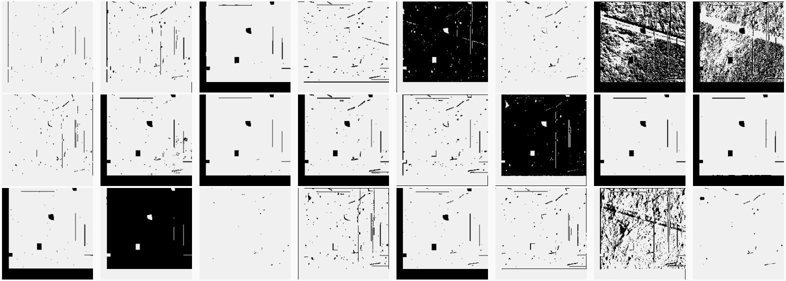


Figure 6: 24 of the 32 first layer feature maps

Qualitative test set results

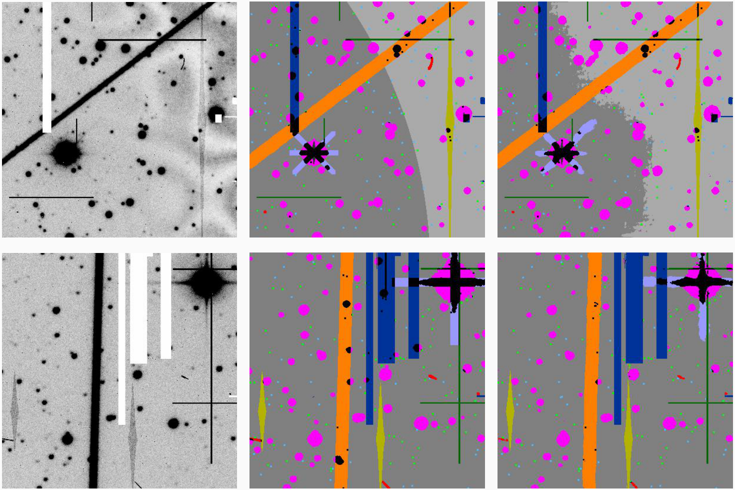


Figure 7: Left: input image. Center: ground truth. Right: prediction.

Quantitative test set results (1/2)

- True Positive Rate $= TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$
- False Positive Rate $= FPR = \frac{FP}{N} = \frac{FP}{TN+FP}$
- Purity or Precision $= PUR = \frac{TP}{TP+FP}$

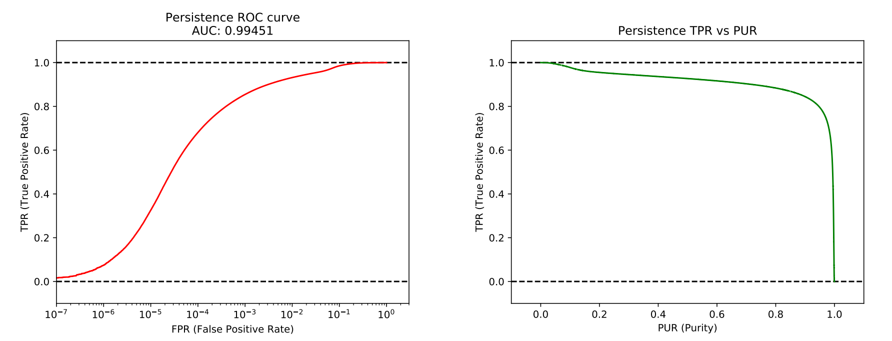
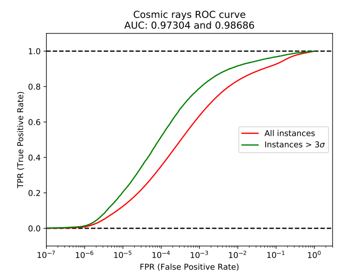
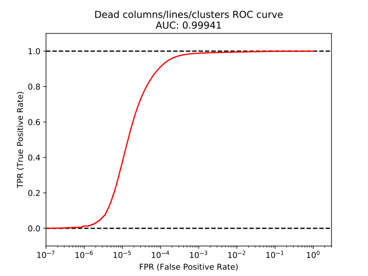
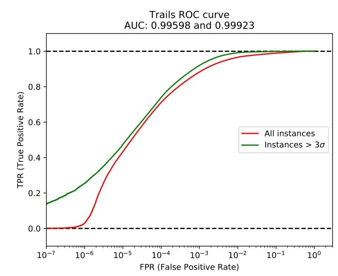


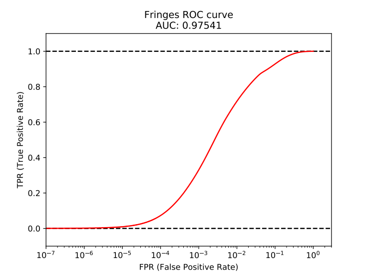
Figure 8: Left: TPR vs FPR. Right: TPR vs PUR.

Quantitative test set results (2/2)









Bayesian prior modification

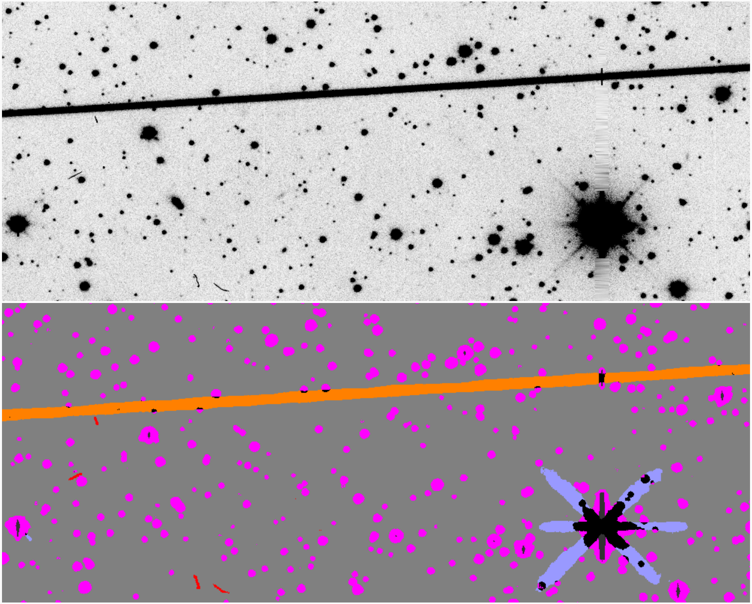
- Output probabilities can be interpreted as Bayesian posteriors [5]
- Priors (= Class proportions) can be modified to adapt the output probabilities to new (expected) class proportions:

$$P(c|\boldsymbol{x}) = \frac{P_N(c|\boldsymbol{x})}{P_N(c|\boldsymbol{x}) + \frac{P_N(c)}{1-P_N(c)}\frac{1-P(c)}{P(c)}(1 - P_N(c|\boldsymbol{x}))}$$

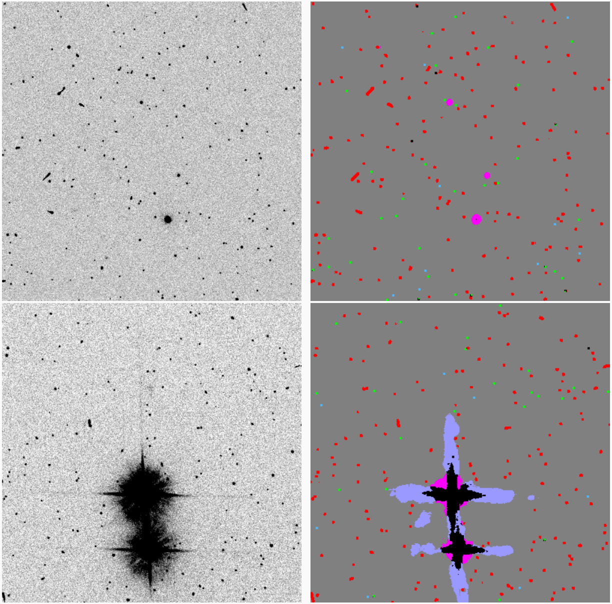
  - $P(c|\boldsymbol{x})$ = New posteriors
  - $P_N(c|\boldsymbol{x})$ = Raw neural network posteriors
  - $P(c)$ = Class c new prior (expected class c proportion in data)
  - $P_N(c)$ = Class c training prior

- Not changing the overall classifier performance
- Just rescale the probabilities along the ROC curve to better represent new expected class proportions

Real life result example (1/2)



Real life result example: unseen instrument (HST) (2/2)



Conclusion

Conclusion

- MaxiMask (+ MaxiTrack) is operational and available for inference
  `https://github.com/mpaillassa/MaxiMask`
- Still some contaminants to include/improve:
  - diffraction spikes
  - ghosts
  - reflections
  - infrared detectors contaminants
- Submitted in A&A
- Available on arXiv
  `https://arxiv.org/abs/1907.08298`

Thank you!

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al.
Tensorflow: A system for large-scale machine learning.
In *OSDI*, volume 16, pages 265–283, 2016.

Emmanuel Bertin and Stephane Arnouts.
Sextractor: Software for source extraction.
Astronomy and Astrophysics Supplement Series, 117(2):393–404, 1996.

H. Bouy, E. Bertin, E. Moraux, J.-C. Cuillandre, J. Bouvier, D. Barrado, E. Solano, and A. Bayo.
Dynamical analysis of nearby clusters. Automated astrometry from the ground: precision proper motions over a wide field.
aap, 554:A101, June 2013.

Diederik P Kingma and Jimmy Ba.
Adam: A method for stochastic optimization.
arXiv preprint arXiv:1412.6980, 2014.

Michael D Richard and Richard P Lippmann.
Neural network classifiers estimate bayesian a posteriori probabilities.
Neural computation, 3(4):461–483, 1991.

Tao Yang, Yan Wu, Junqiao Zhao, and Linting Guan.
Semantic segmentation via highly fused convolutional network with multiple soft cost functions.
Cognitive Systems Research, 2018.

MaxiMask cost function

- Final predictions loss:

$$L_f = -\frac{1}{B}\sum_{b\leq B}\sum_{p\in\mathcal{P}} w_p \sum_{c\in\mathcal{C}}\big(y_{b,p,c}\log(\hat{y}_{b,p,c})+$$

$$(1-y_{b,p,c})\log(1-\hat{y}_{b,p,c})\big) + L2_{reg}$$

  - B = Batch Size; $\mathcal{P}$ = Set of all pixels; $\mathcal{C}$ = Set of all classes.
  - $\hat{y}_{b,p,c}$ = Sigmoid class c prediction of pixel p
  - $y_{b,p,c}$ = Class c ground truth of pixel p, i.e:

$$y_{b,p,c} = \begin{cases} 1 & \text{if } p \in c \\ 0 & \text{otherwise} \end{cases}$$

  - Each pixel is weighted according to its class(es) proportion(s):

$$w_p = \sum_{c\in\mathcal{C}_p} w_c \text{ with } w_c = \frac{1}{p_c \sum_{c'} \frac{1}{p_{c'}}}$$

  - Weigth maps are then 3×3 smoothed.