

Cloud Identification in Satellite Images using Artificial Intelligence

Viva

Kenza Tazi

18th March 2019

Supervisors: Ulrik Egede, Helen Brindley.

Summary

- 1. Introduction and motivation**
- 2. Methodology**
 - 2.1. Data and collocation
 - 2.2. Neural networks
- 3. Results**
- 4. Conclusion**
- 5. References**
- 6. Appendix**



1. Introduction and motivation

Clouds:

- 70% of the Earth's surface area [1]
- Water cycle and energy budget [2]
- Largest uncertainty in weather and climate forecasting [3]
- If misidentified, lead to incorrect measurements [4]

Problematic conditions (*see Fig. 1*):

- Snow and sea ice
- Edges (e.g. coastlines)
- Reflective surfaces (e.g. lakes, rivers)
- Twilight
- Particulate matter.

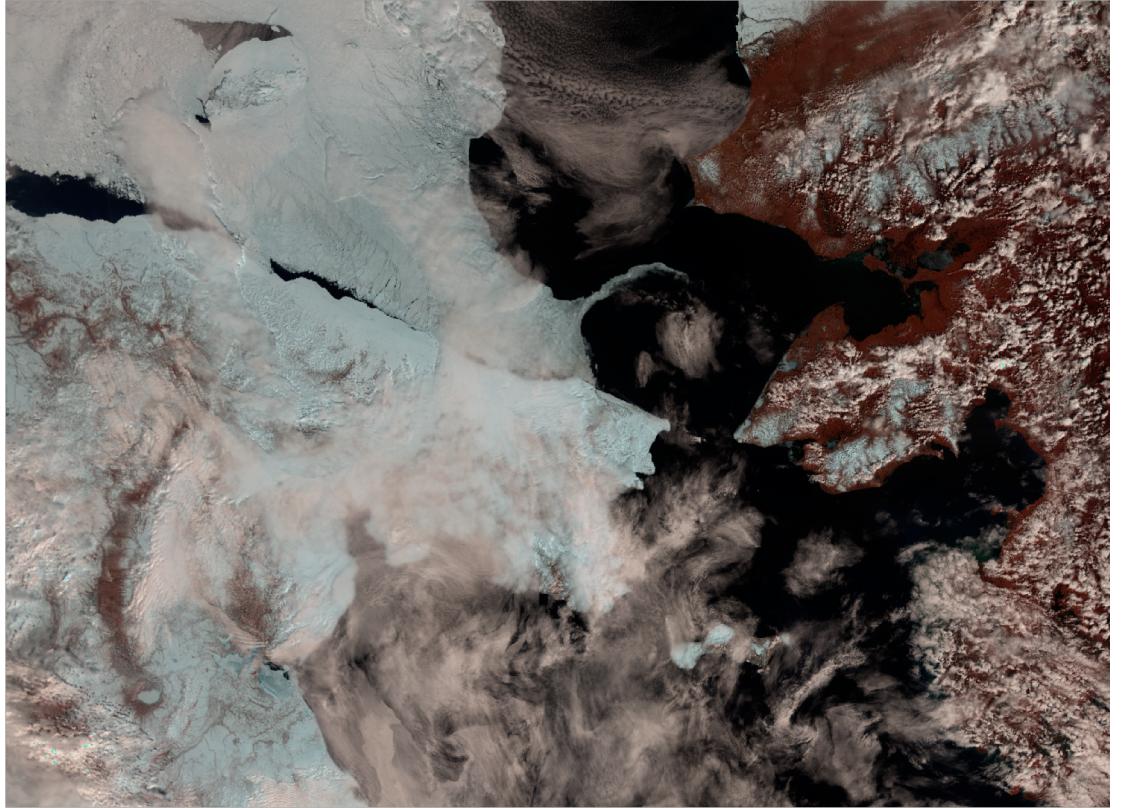


Fig. 1: Example of problematic scene: false colour image created from SLSTR data taken on the 31st May 2018 at (73.36, 170.62).

3. Methodology

Use Artificial Neural Networks (ANN) for classification.

Why ANN?

- Non-linear functions
- Fast to apply
- Limited knowledge of the system
- Flexible input types

Requirements:

- Lots of relevant input data
- Corresponding truth data.



Solution:

Satellite Data Collocation

between Sentinel 3 and CALIPSO (= finding data taken at the same place and same time by both satellites)

3. Methodology – Data and collocation

SLSTR:

- Sentinel 3A and Sentinel 3B
- *Passive*, poor at identifying clouds
- Swath width = 1200km

CALIOP:

- CALIPSO
- *Active*, good at identifying clouds
- Swath width = 70 m

Collocation:

- 3.4 million collocated pixels within 353m, and less than 20 minutes apart
- Data entirely over polar regions

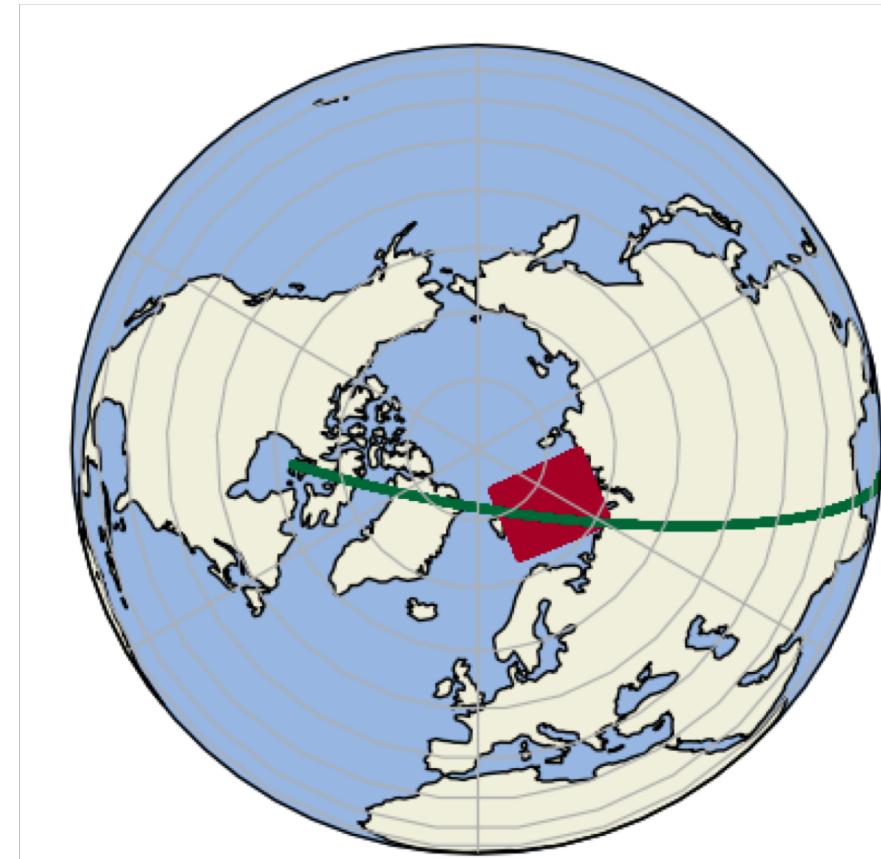


Fig. 2: Map illustrating CALIOP track (green) over an SLSTR scene (red).

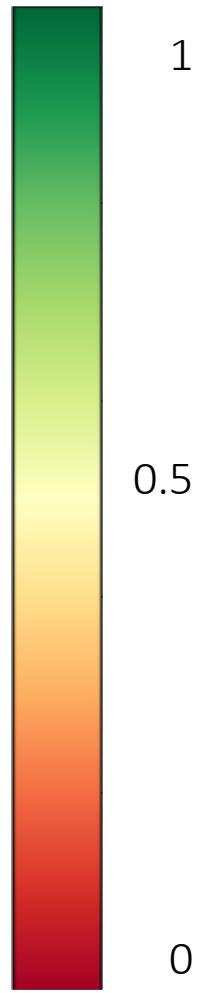
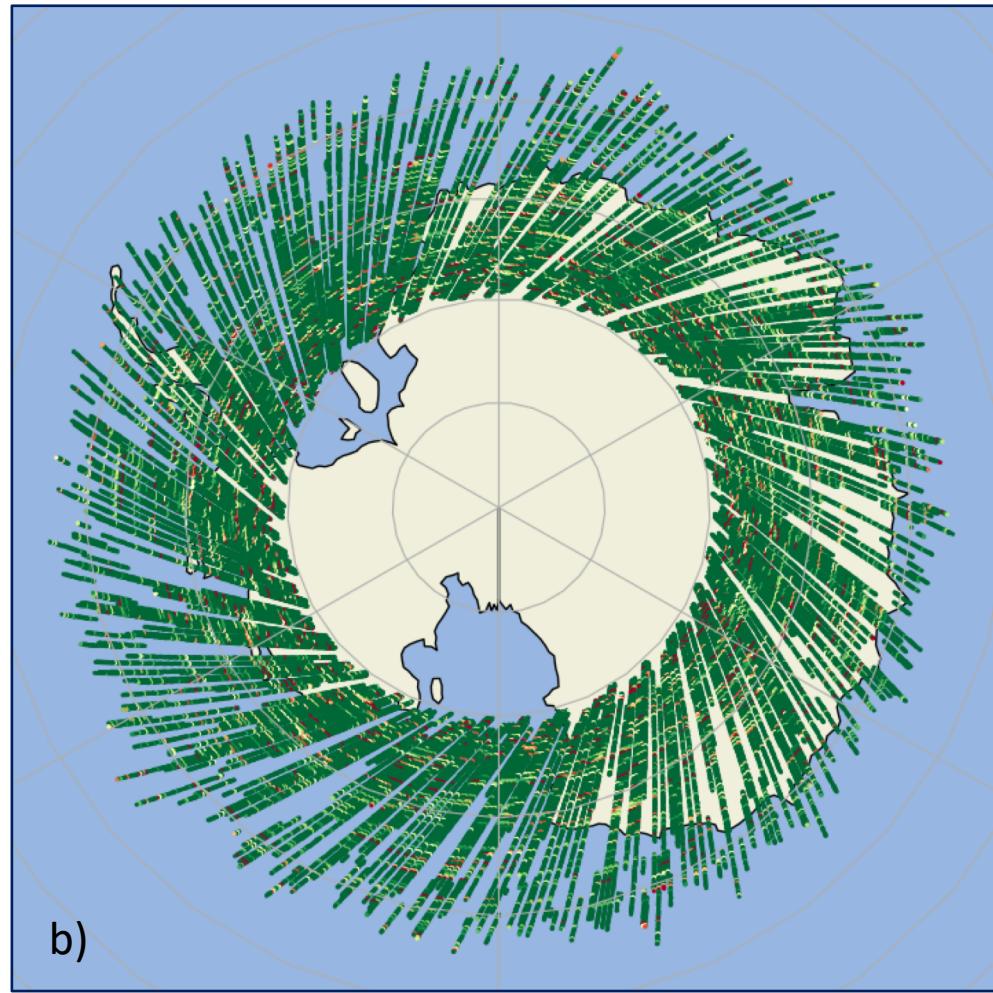
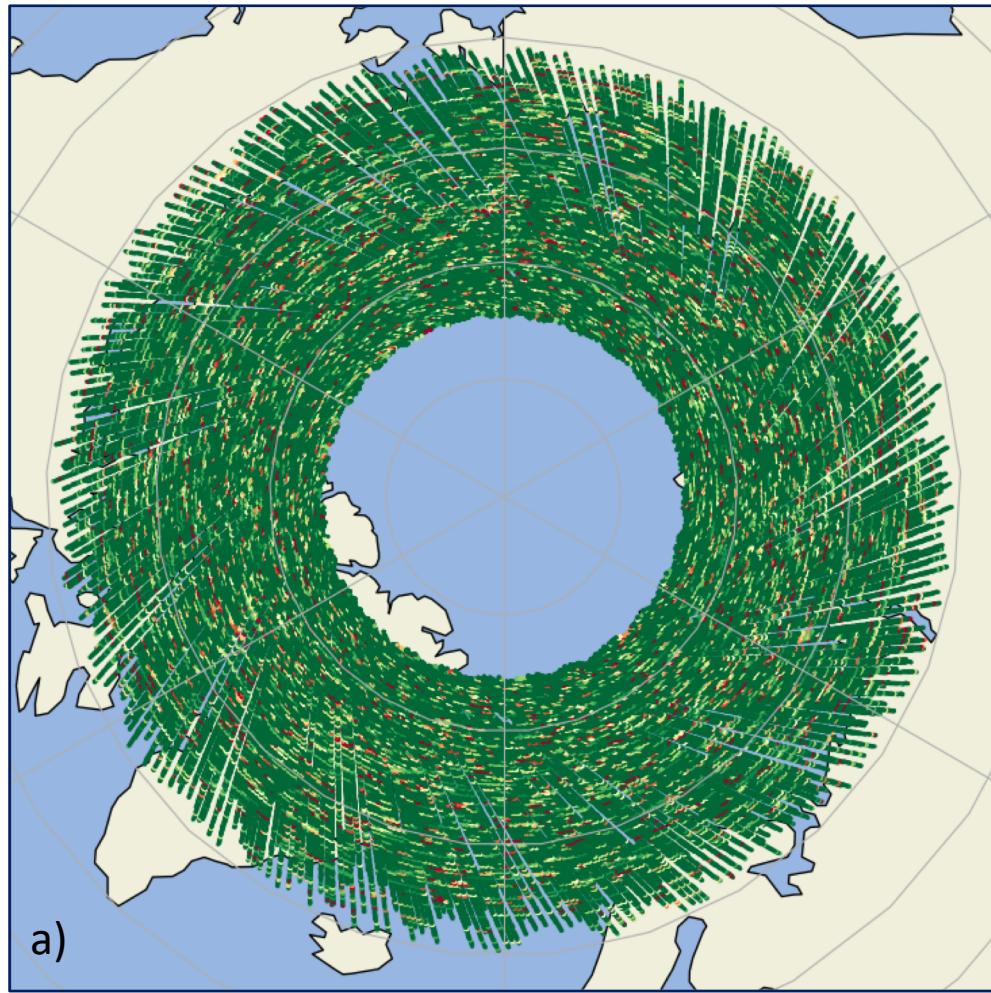


Fig. 3: Average model agreement with CALIPSO plotted over a) the Arctic and b) the Antarctic.

3. Methodology – Neural networks

- Focus on our most successful ANN:
Feed Forward Neural Network (FFN)
- Steps :
 1. Sum inputs into neurons
 2. Repeat multiple times to form layers
 3. Apply optimisation algorithm to set the weights of the inputs so the output matches the truth data
 4. Update weights with each new data point

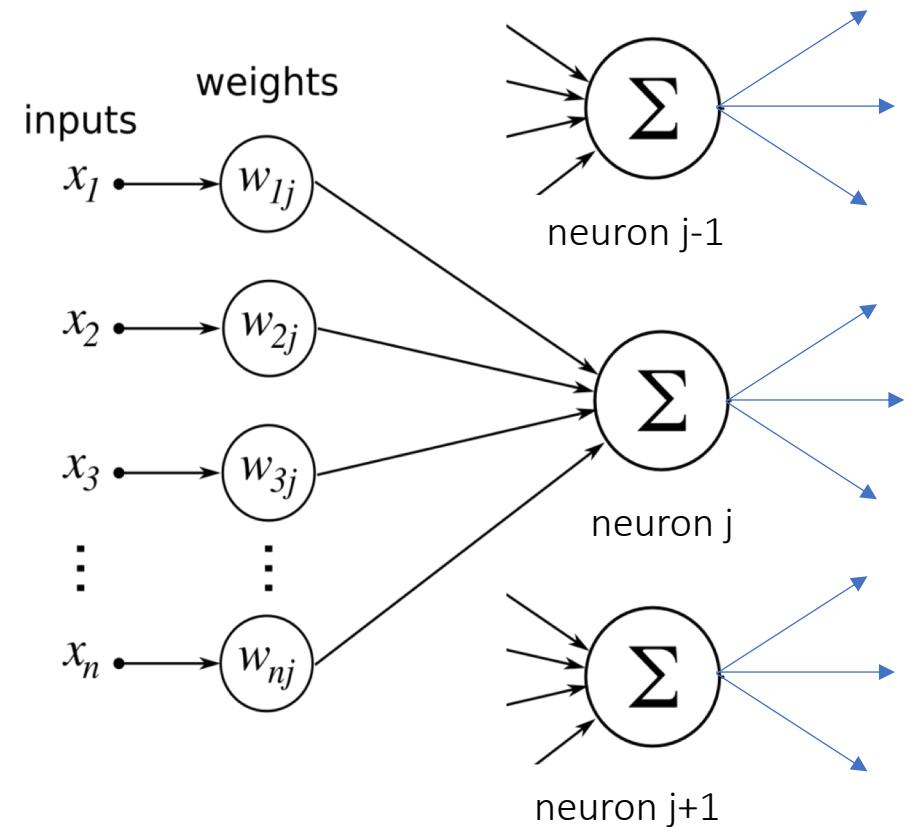


Fig 4: Simplified diagram of FFN [5].

⇒ By showing the model SLSTR data but also the correct classification (CALIOP data) it can learn to recognise clouds

- **Inputs (individual pixel):**
 - Radiance in 9 wavelength channels
 - Latitude and longitude
 - Solar and satellite zenith angle
 - Surface type information
- **Truth data:**
 - 1 bit (1 for cloud, 0 for clear)
- **Output:**
 - Confidence from 0 to 1, where:
 - $1 \Leftrightarrow 100\%$ confident pixel is cloudy
 - $0 \Leftrightarrow 100\%$ confident pixel is clear
 - $0.5 \Leftrightarrow$ model is unsure



4. Results

89% average accuracy on unseen data

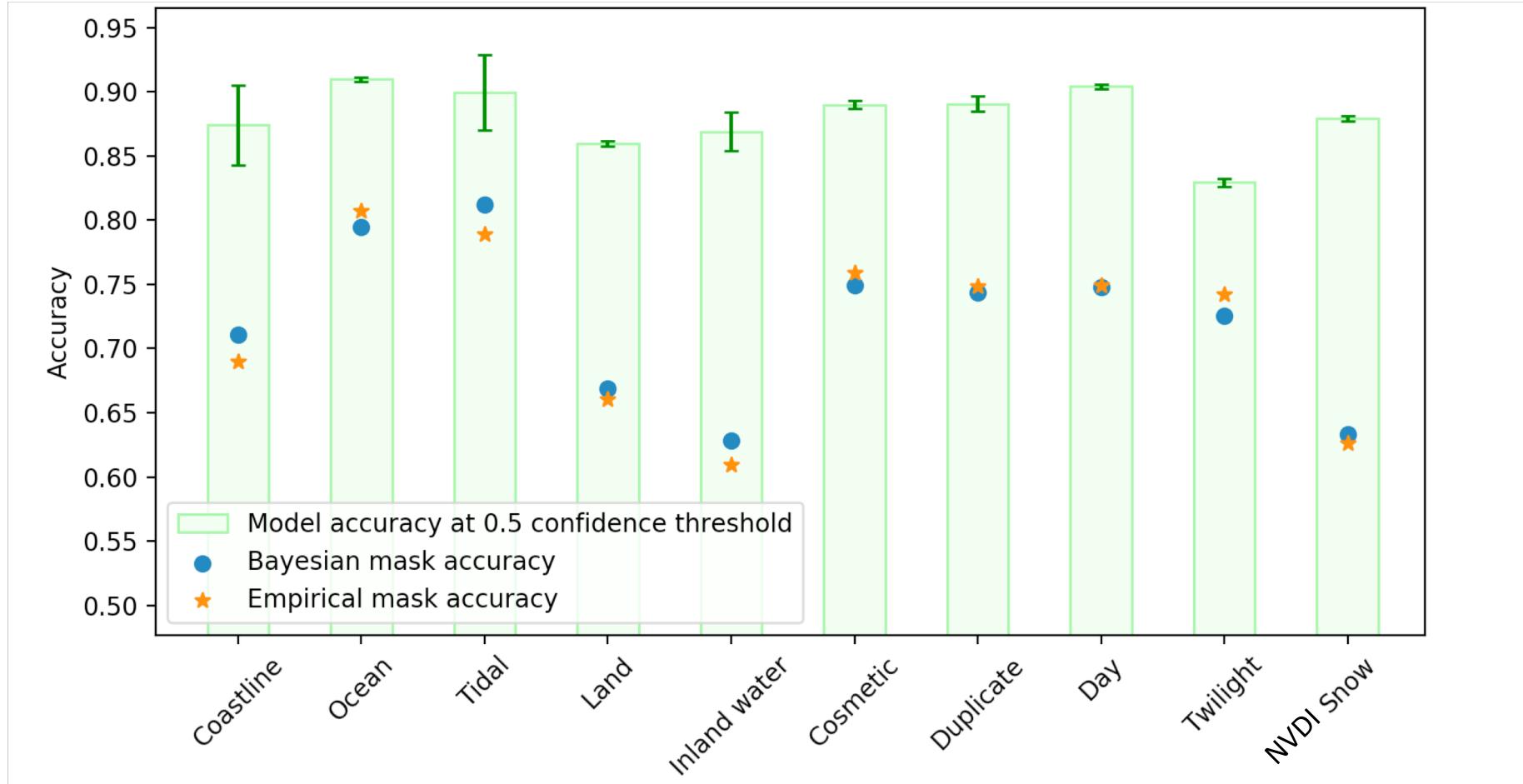


Fig. 6: Model accuracies as function of surface type.

Receiver Operator Curve (ROC)
 \Leftrightarrow performance for all thresholds

Performance metric = area
under the curve (AUC)

tp = number of true positives
 tn = number of true negatives
 fp = number of false positives
 fn = number of false negatives

$$\text{False positive rate} = \frac{fp}{tn+fp}$$

$$\text{True positive rate} = \frac{tp}{tp+fn}$$

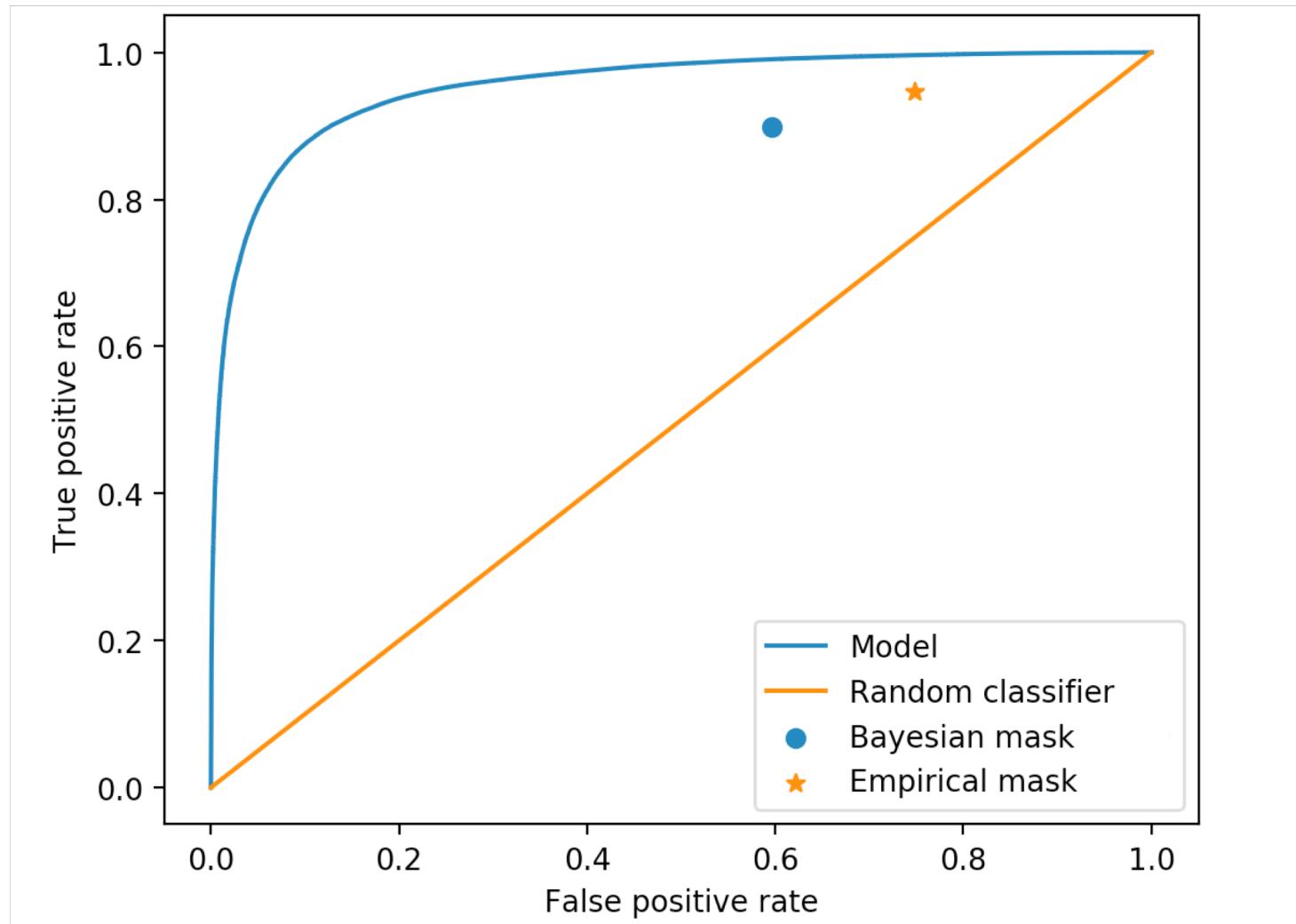


Fig. 7: FFN model ROC with comparisons to empirical and Bayesian masks.

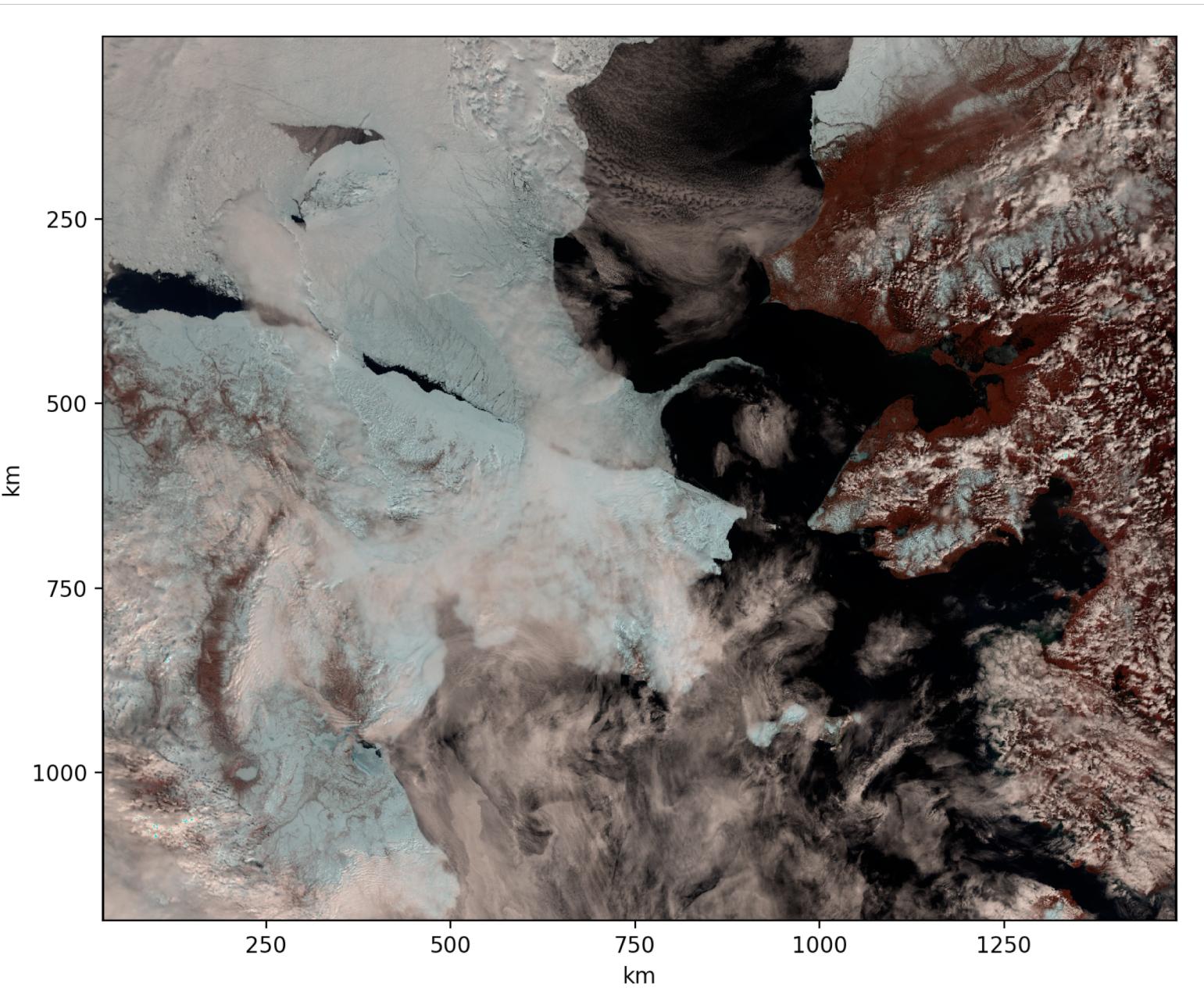
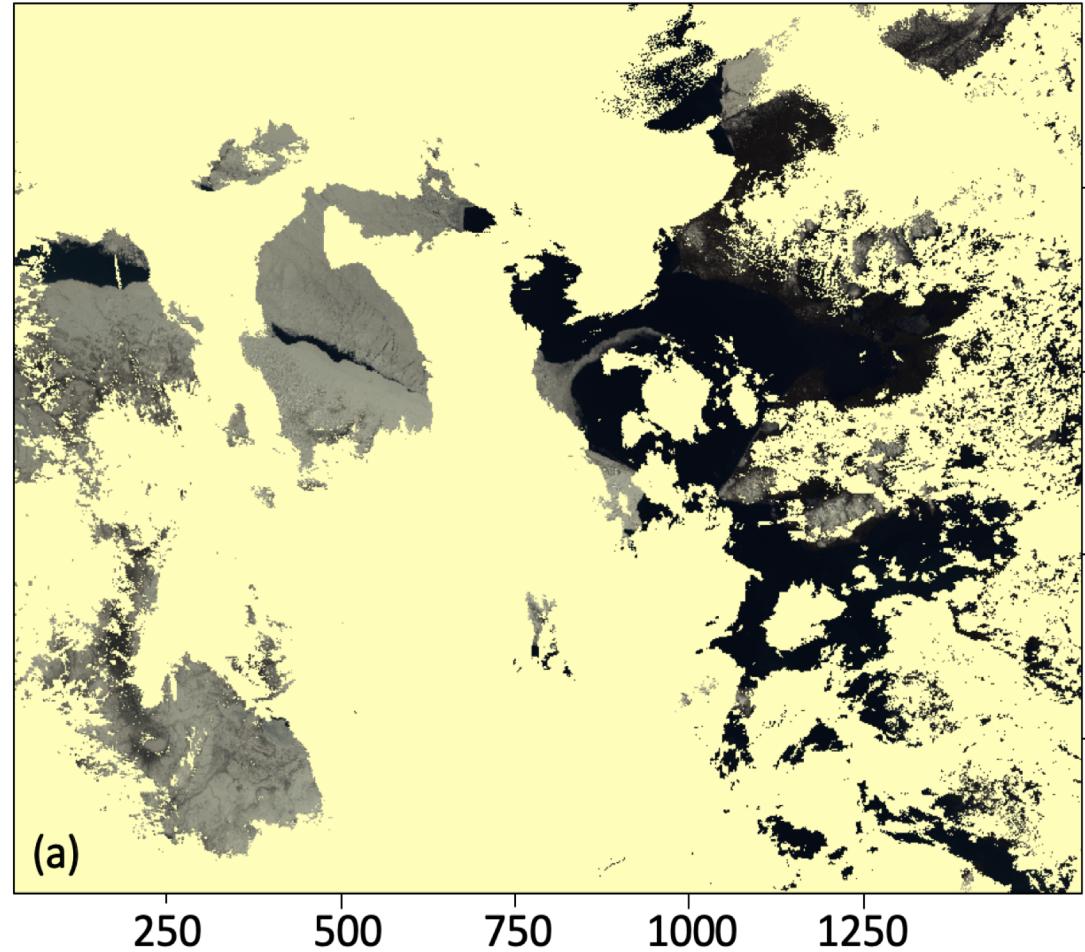


Fig. 1: Example of problematic scene: false colour image created from SLSTR data taken on the 31st May 2018 at (73.36, 170.62).

FFN model mask



Bayesian mask

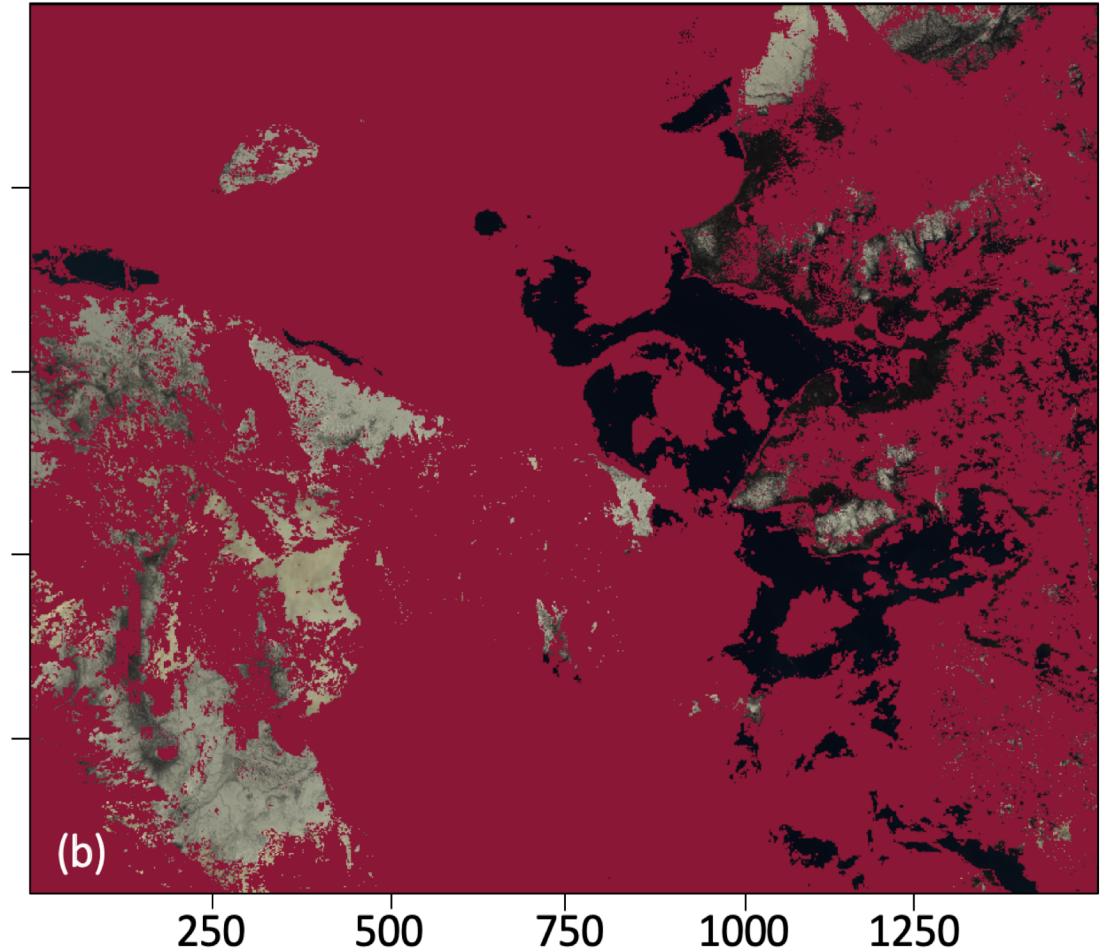
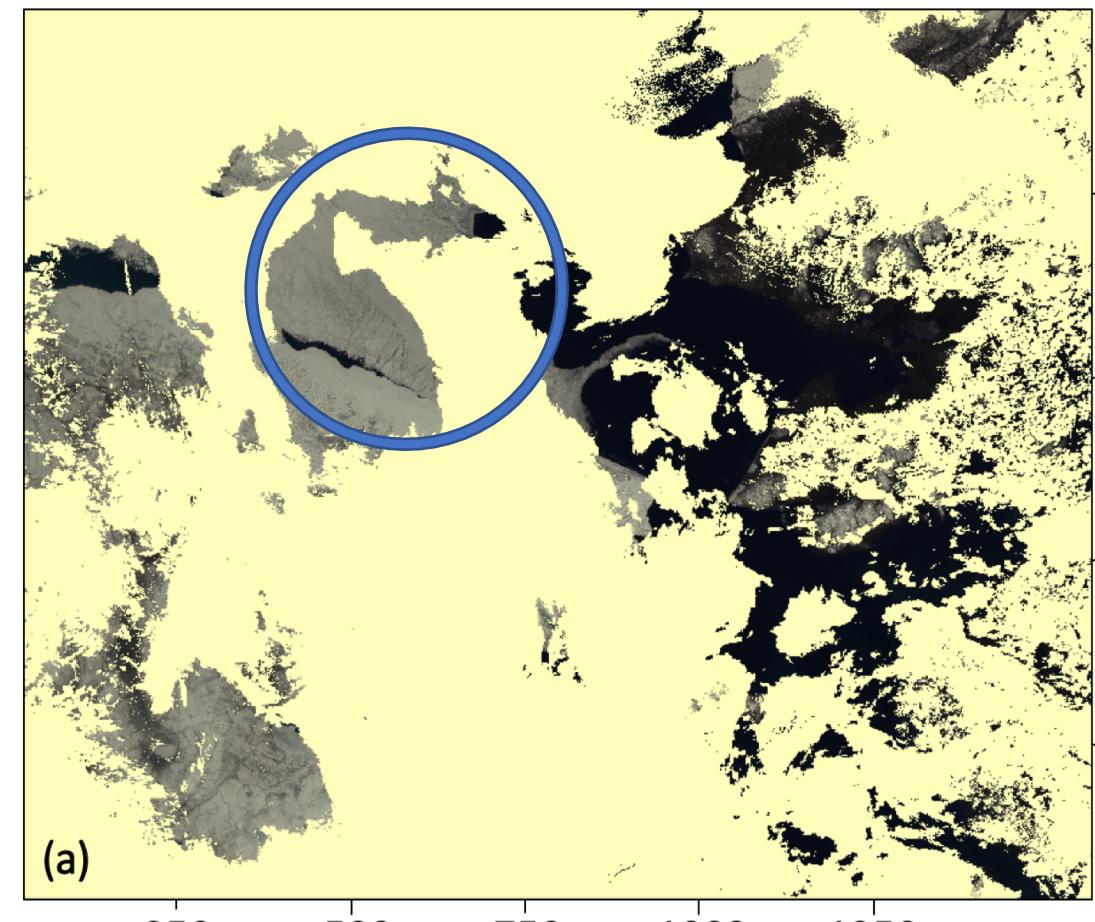


Fig. 7: Scene superimposed with FFN model mask and the Bayesian mask included with the SLSTR files.

FFN model mask



Bayesian mask

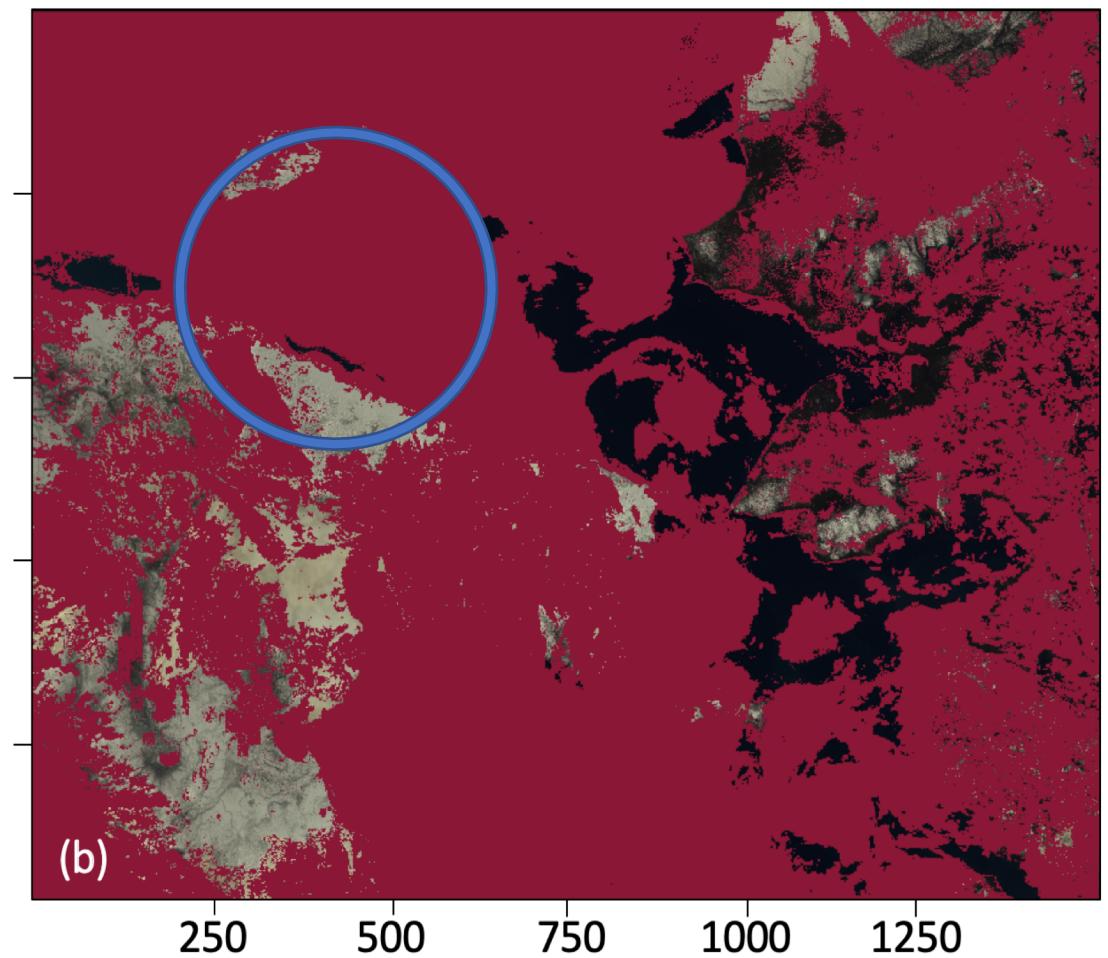
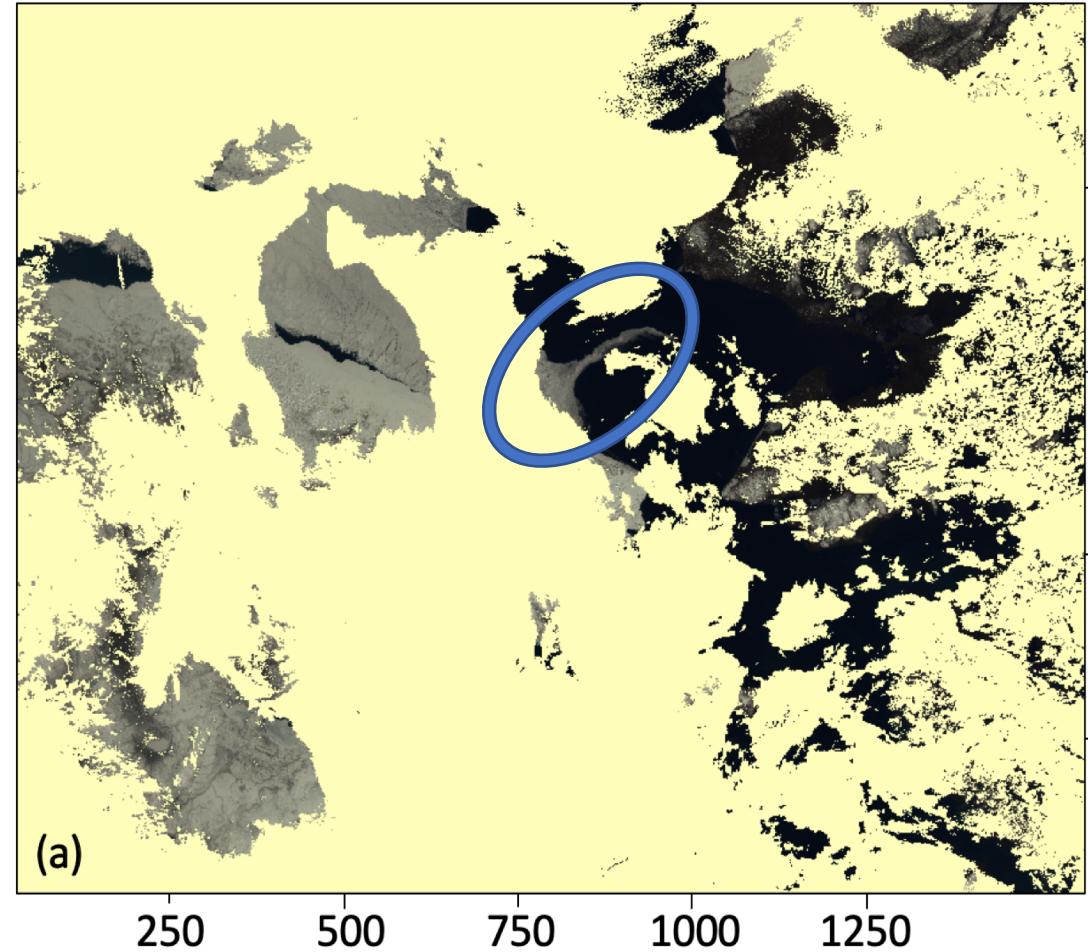


Fig. 7: Scene superimposed with FFN model mask and the Bayesian mask included with the SLSTR files.

FFN model mask



Bayesian mask

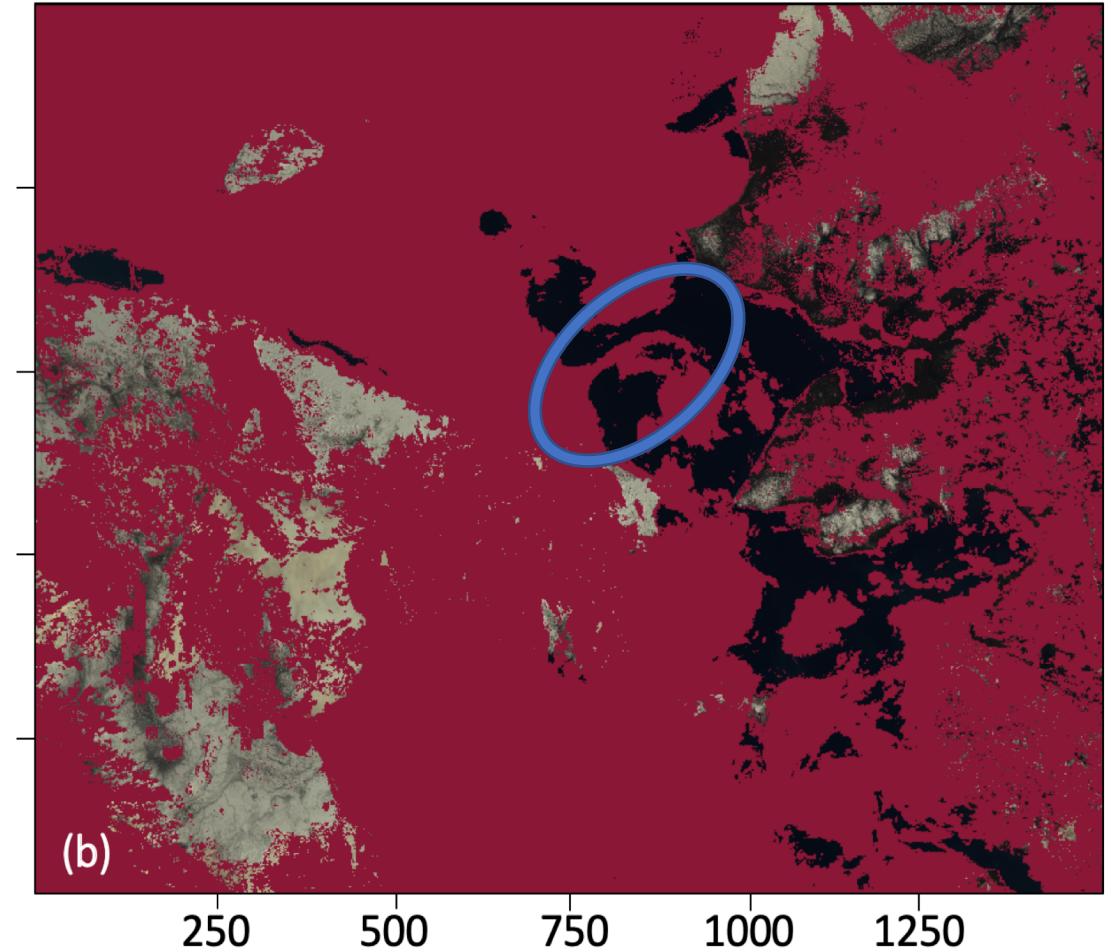
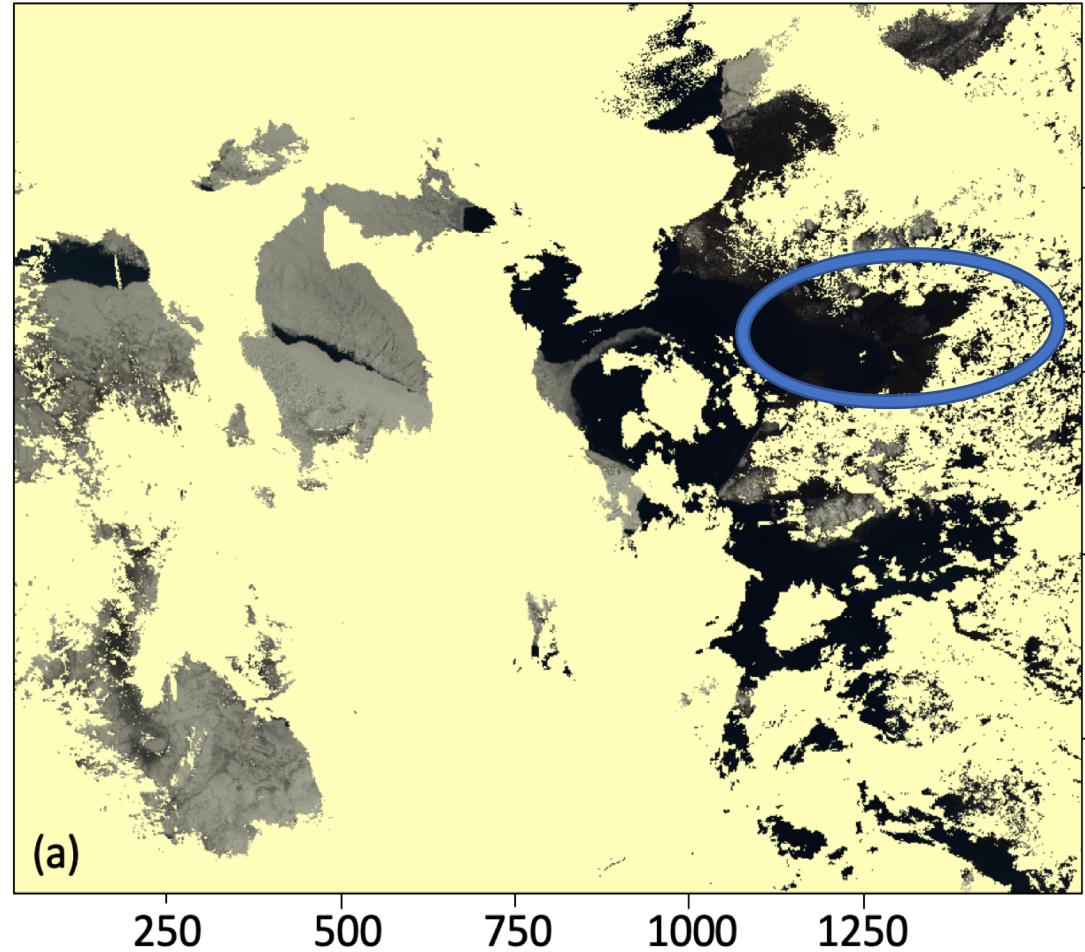


Fig. 7: Scene superimposed with FFN model mask and the Bayesian mask included with the SLSTR files.

FFN model mask



Bayesian mask

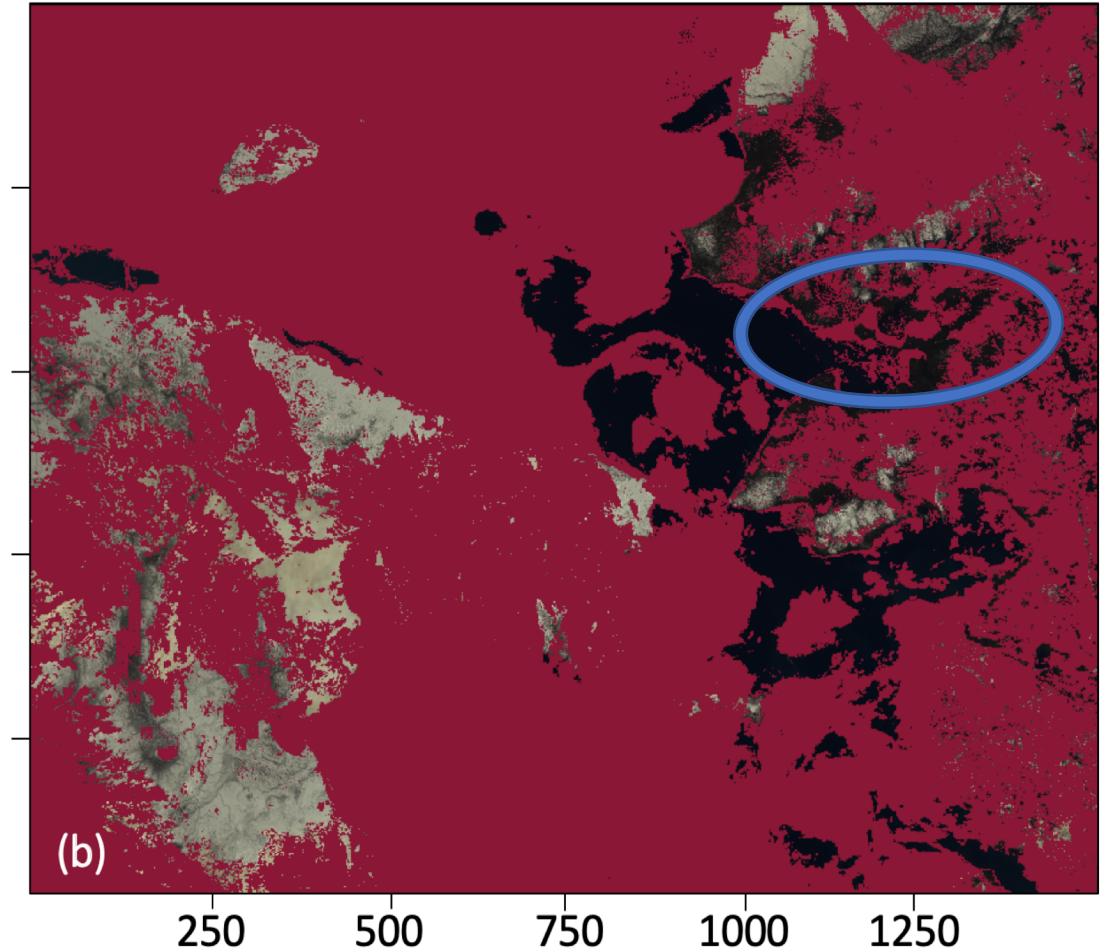


Fig. 7: Scene superimposed with FFN model mask and the Bayesian mask included with the SLSTR files.

5. Conclusion

Summary of results :

- Better performance in all conditions.

Research implications:

- Better weather and climate predictions,
- More usable Earth Observation data.

Future developments:

- Train model on lower latitude data,
- Multi-categorical classification (determine cloud types),
- Regression (calculate cloud optical depths).

6. References

Inline references

- [1] Stubenrauch CJ et al. Assessment of global cloud datasets from satellites: Project and database initiated by the GEWEX radiation panel. *Bulletin of the American Meteorological Society*. 2013 Jul;94(7):1031-49.
- [2] Kiehl JT, Trenberth KE. Earth's annual global mean energy budget. *Bulletin of the American Meteorological Society*. 1997 Feb;78(2):197-208.
- [3] Sus O et al. The Community Cloud retrieval for CLimate (CC4CL)-Part 1: A framework applied to multiple satellite imaging sensors. *Atmospheric Measurement Techniques*. 2018 Jun 1;11(6).
- [4] Anze Zupanc. Improving Cloud Detection with Machine Learning. Dec 2017. Available at: <https://medium.com/sentinel-hub/improving-cloud-detection-with-machine-learning-c09dc5d7cf13>
- [5] Christoph Burgmer. Diagram of an artificial neuron. Jul 2005. Available at: https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png

False colour images

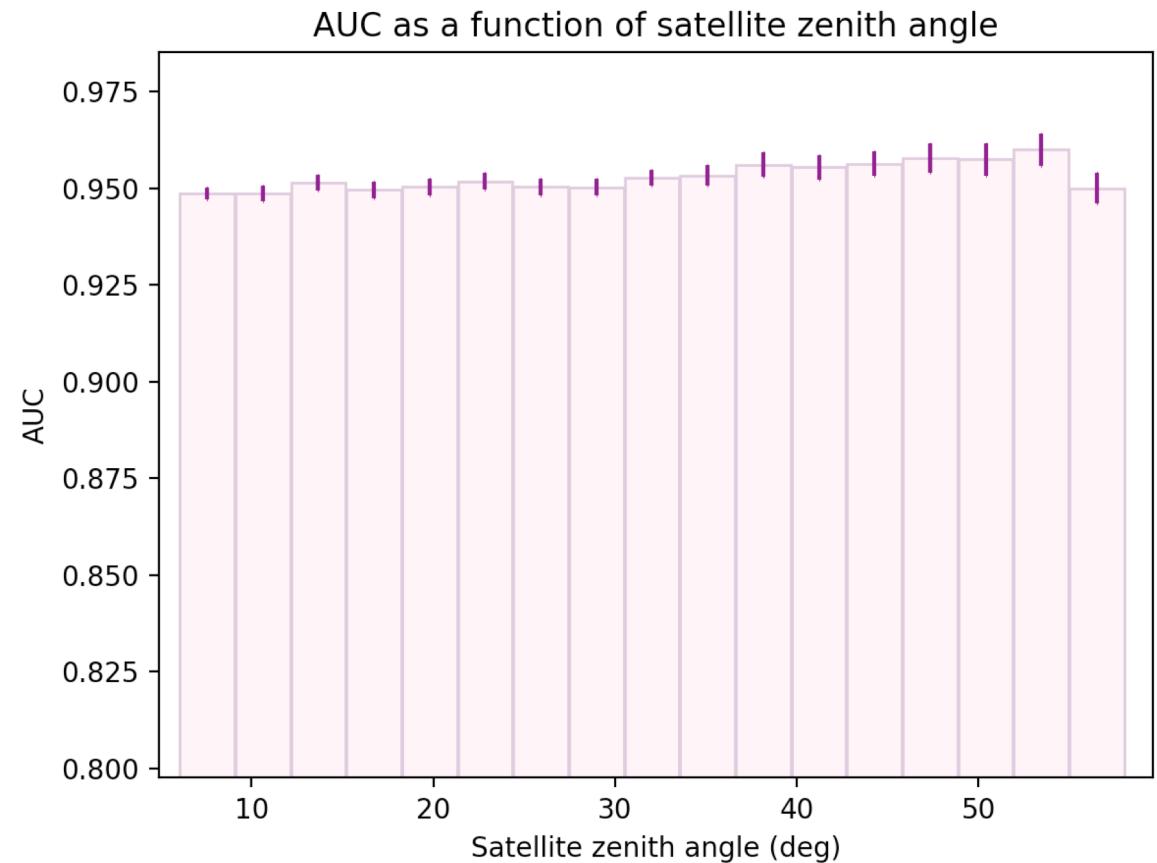
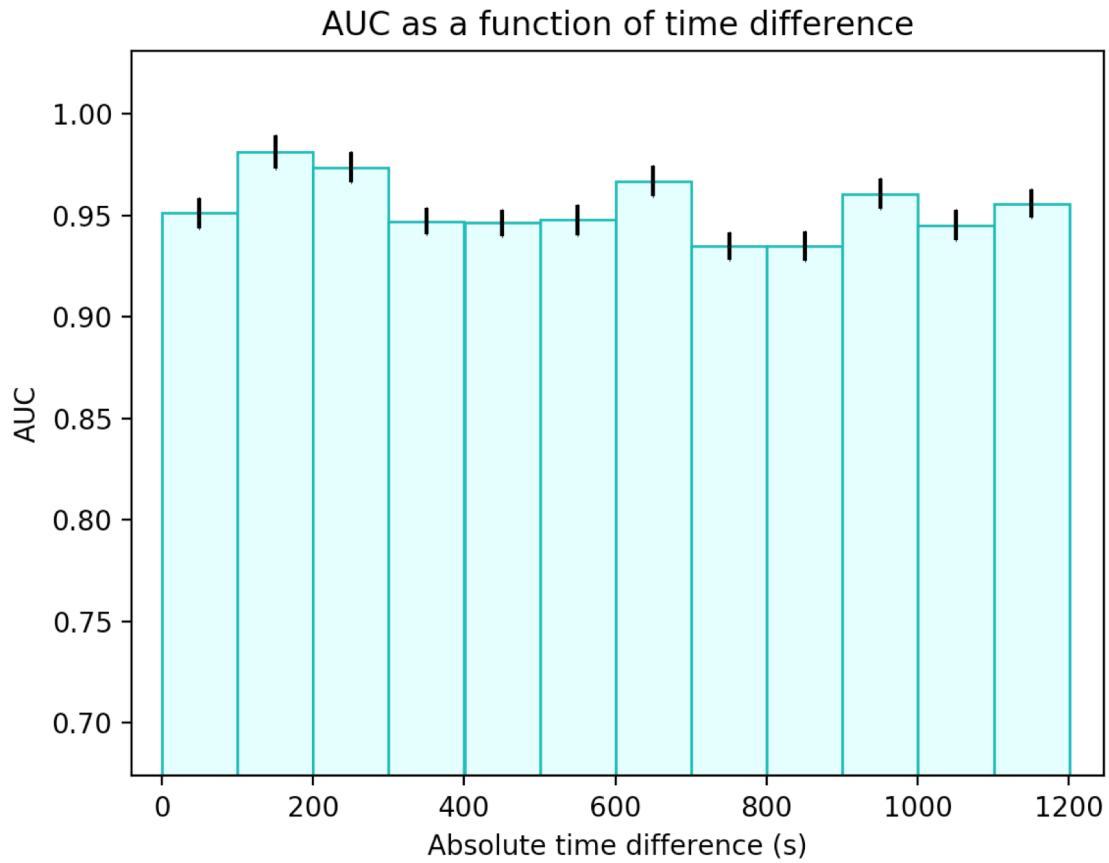
Page 1: 29th May 2018 at 11:30:03 UTC

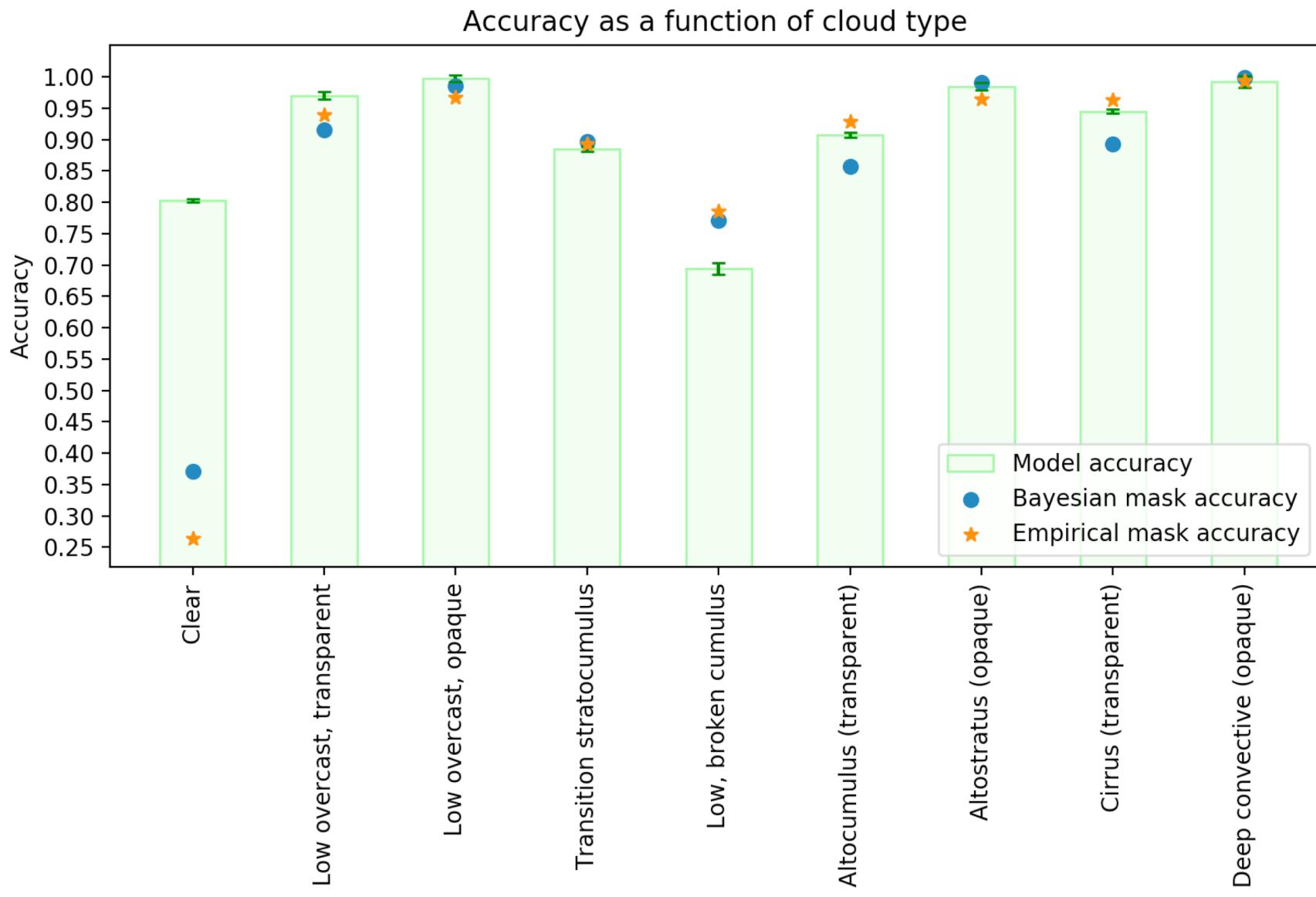
Page 2: 22nd August 2018 at 00:06:19 UTC

Page 3, 9, 10: 31st May 2018 at 22:30:36 UTC

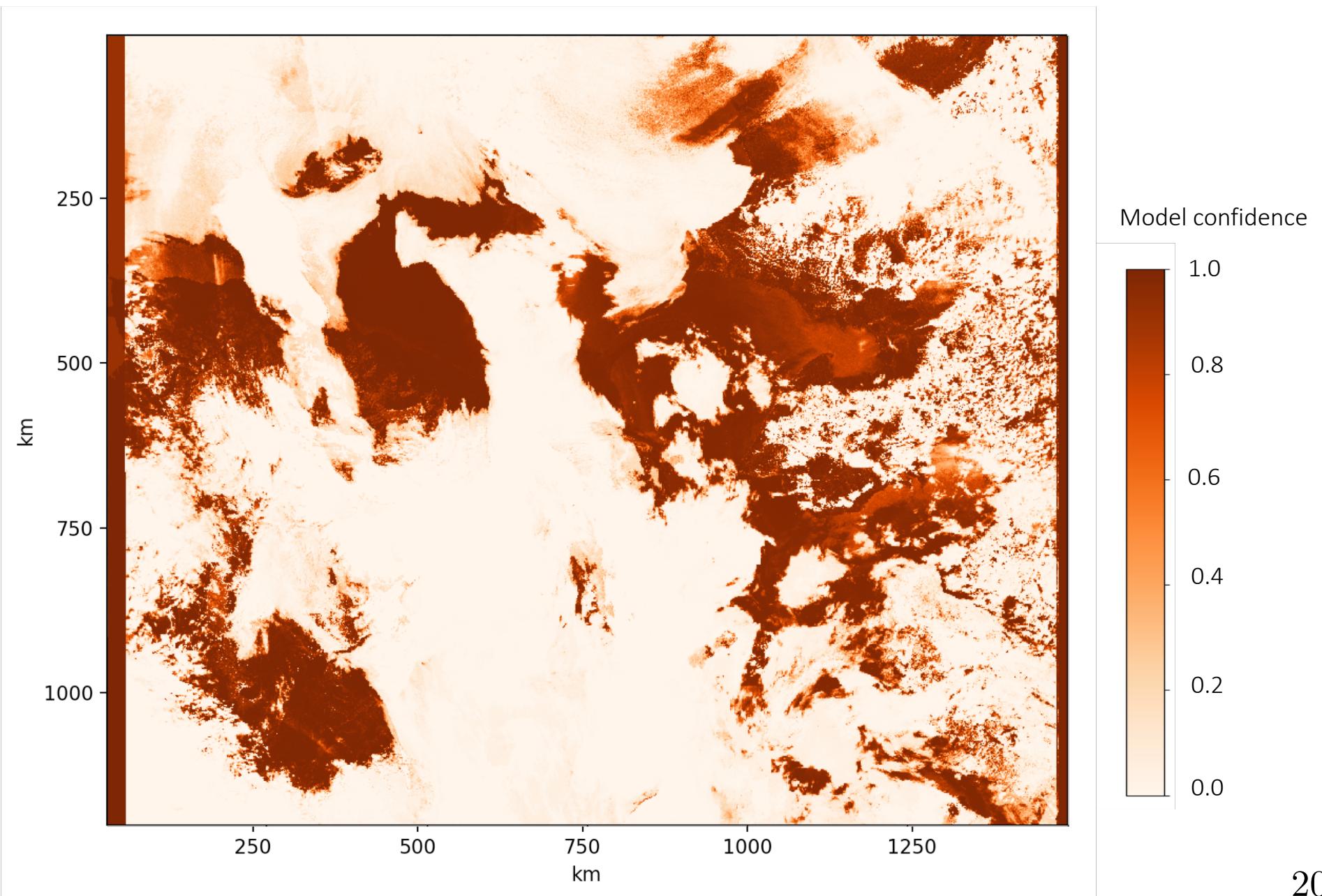
Page 11: 29th August 2018 at 20:09:50 UTC

7. Appendix

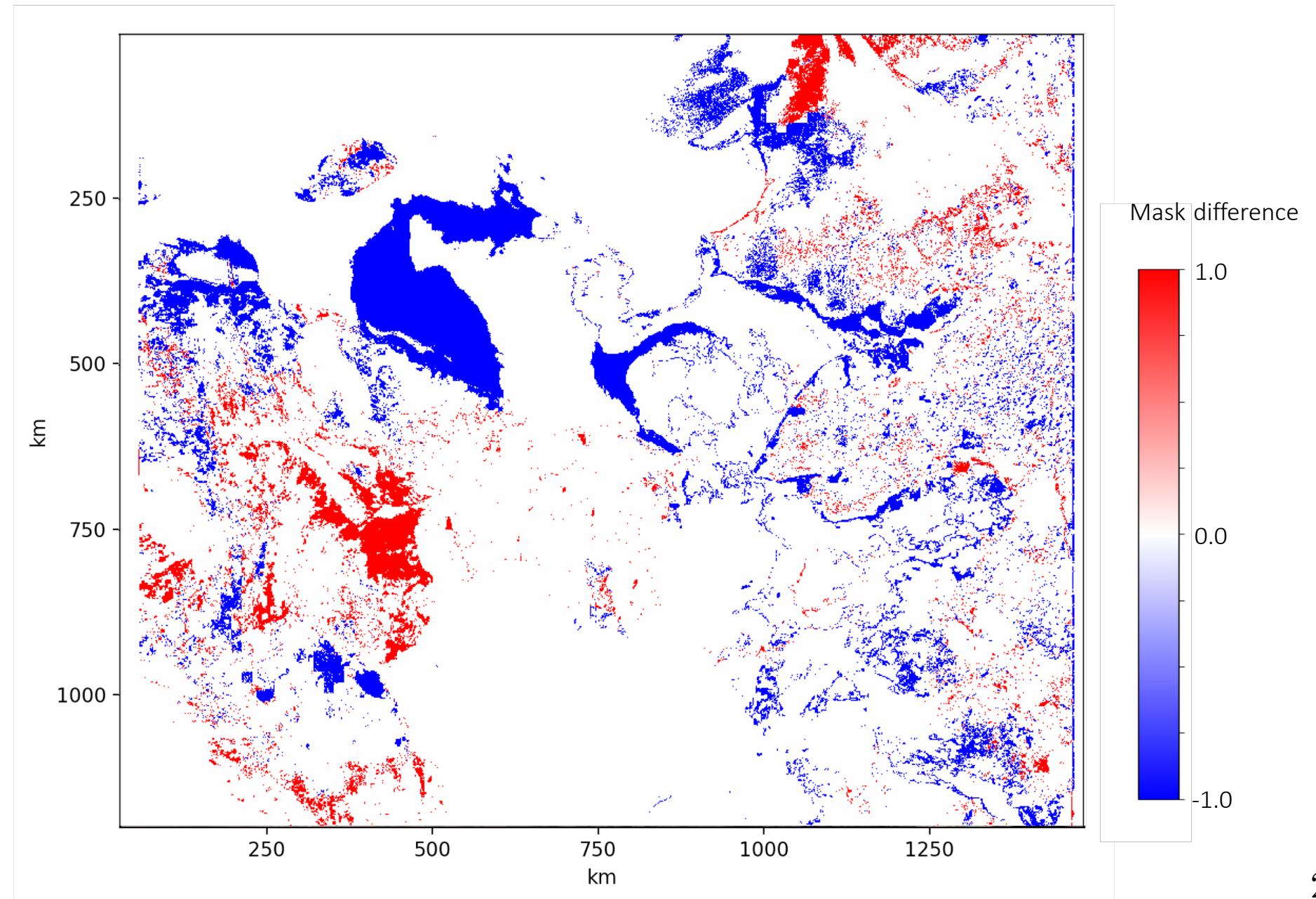




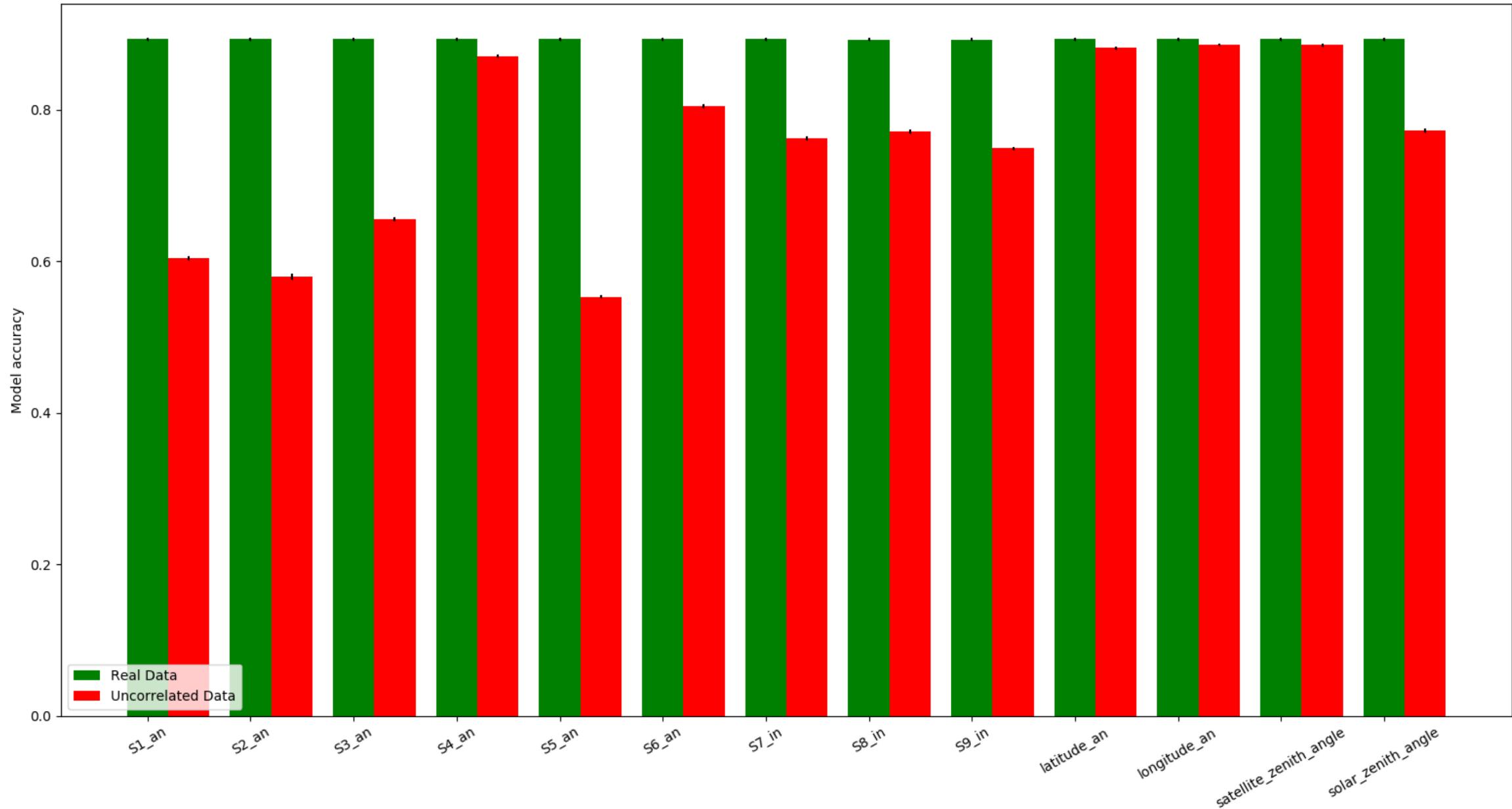
FFN model output



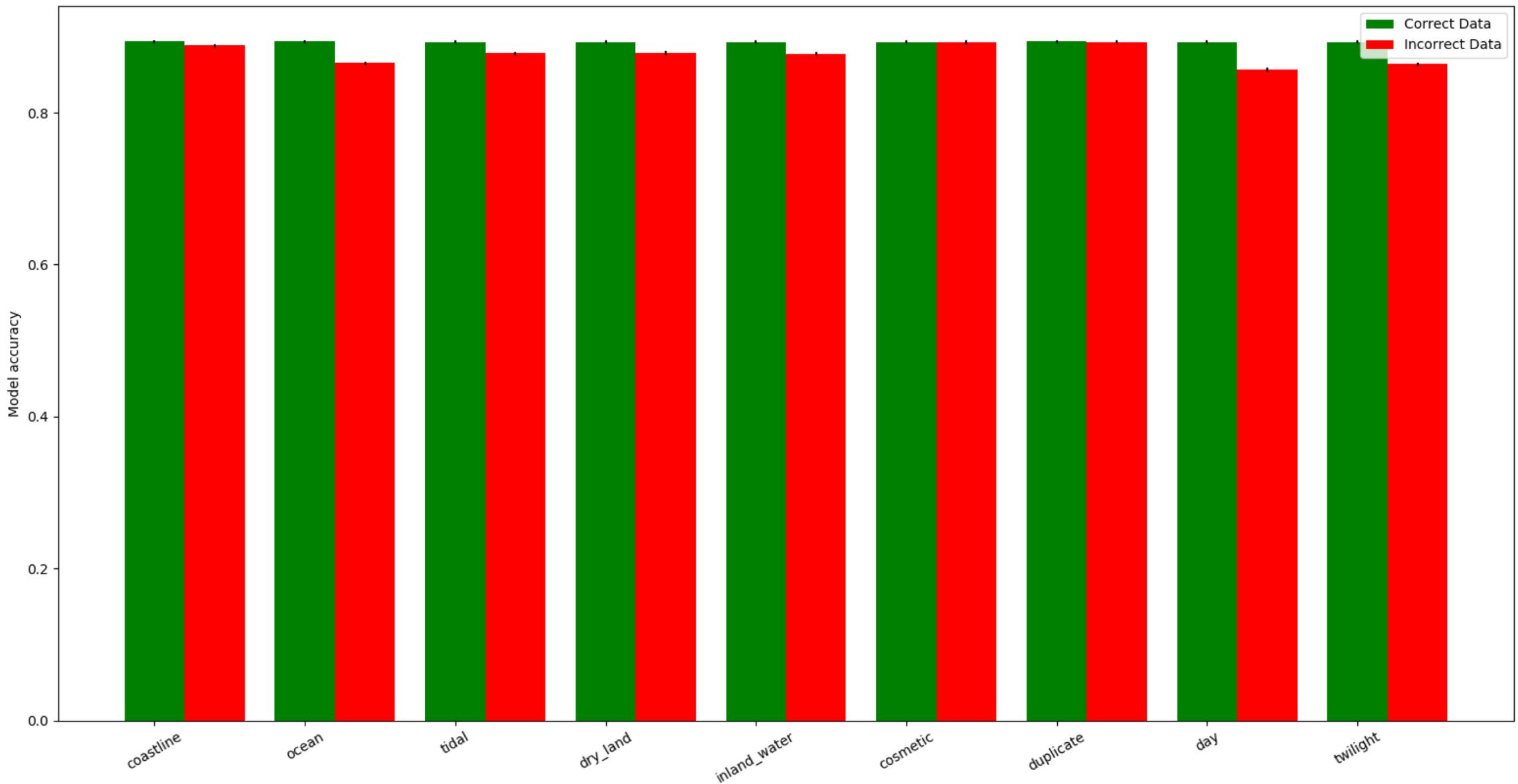
Difference of the
FFN binary
mask and the
Bayesian mask.



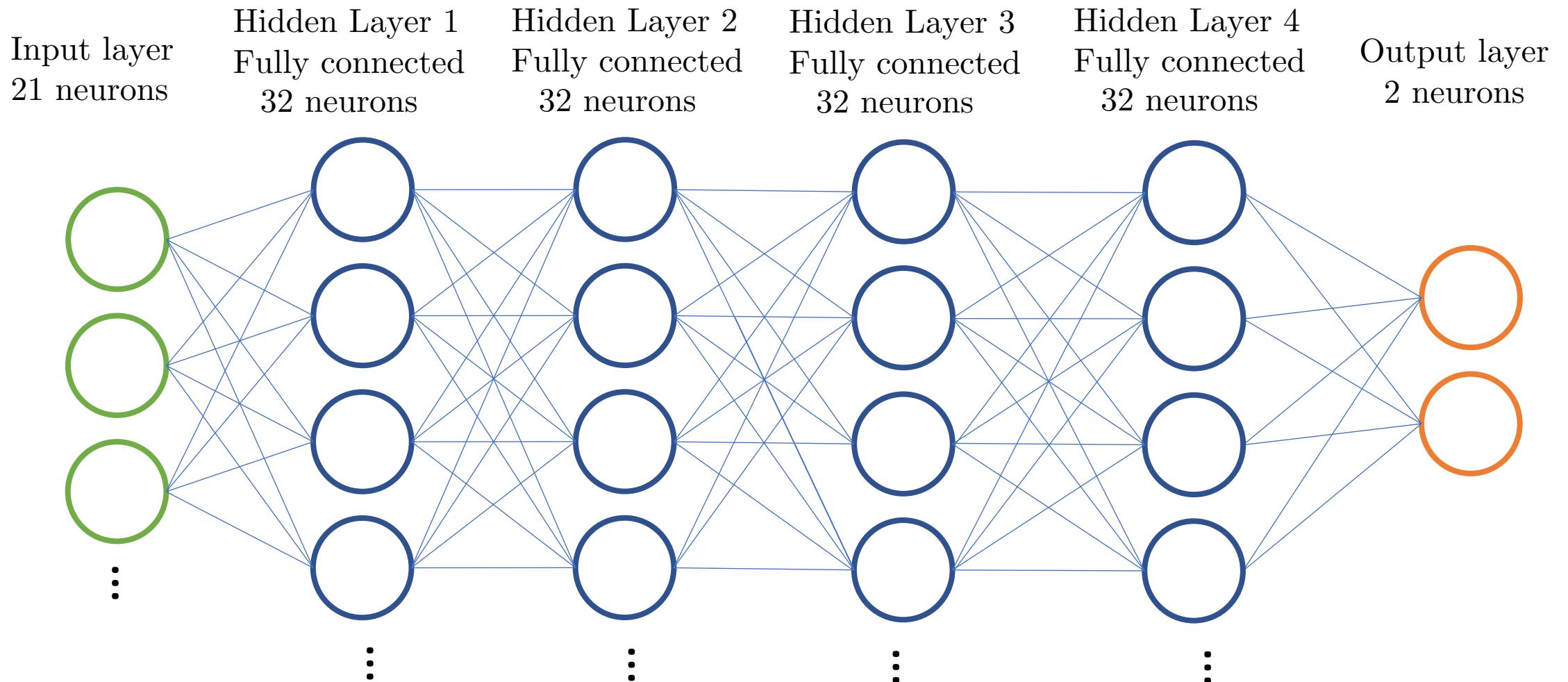
Model sensitivity — Shuffling — Radiance channels, position, angles



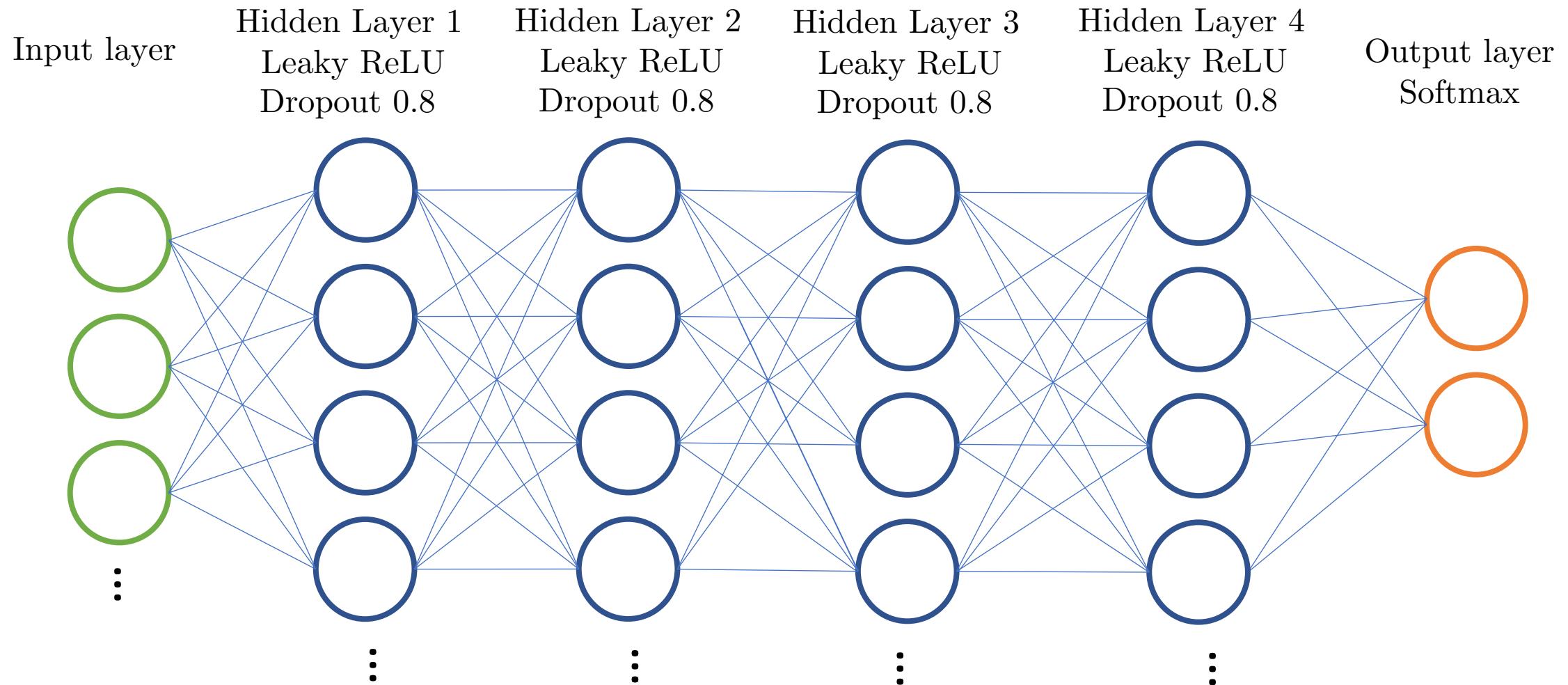
Model sensitivity — Shuffling — Surface types



Model structure — Neurons and layers



Model structure — Activation functions and dropouts



Model structure — Other hyperparameters

- Optimisation algorithm = Adaptive moment estimation optimisation algorithm (ADAM)
- Learning rate = 10^{-4}
- Epoch numbers = 160
- Batch size = 64