Klassifikation von Satellitenbildern

Bilderkennung mit künstlichen neuronalen Netzen

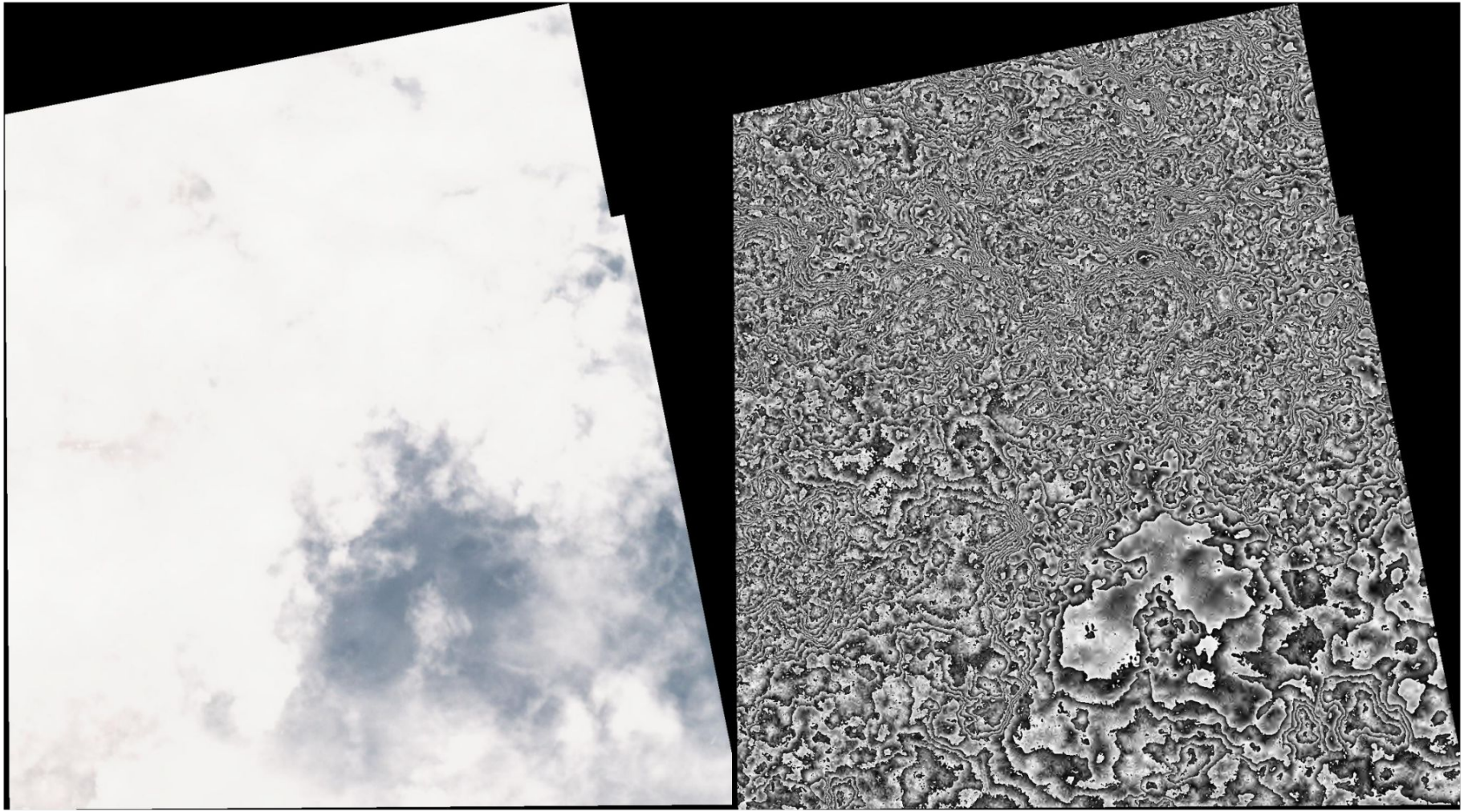
Problemstellung

Binäre Klassifikation von Satellitenaufnahmen:

Ist ein Schiff zu sehen oder nicht?

Datensatz

- 1751 Satellitenaufnahmen von Seegebieten
- 4-Kanal BGR + Near-Infrared

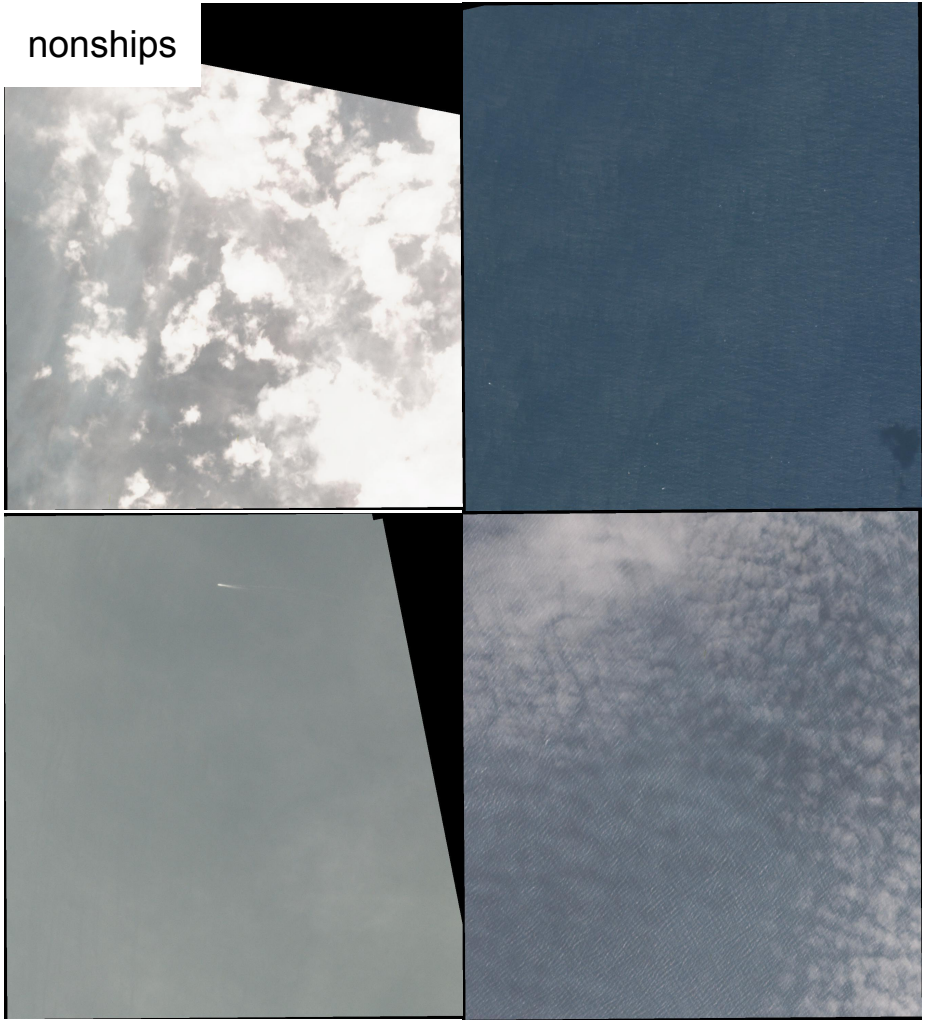


Datensatz

- 1751 Satellitenaufnahmen von Seegebieten
- 4-Kanal BGR + Near-Infrared
- 871 ship, 880 noship

ships

nonships



Datensatz

- 1751 Satellitenaufnahmen von Seegebieten
- 4-Kanal BGR + Near-Infrared
- 871 ship, 880 noship
- Format: TIFF mit 16bit pro Kanal

Datensatz

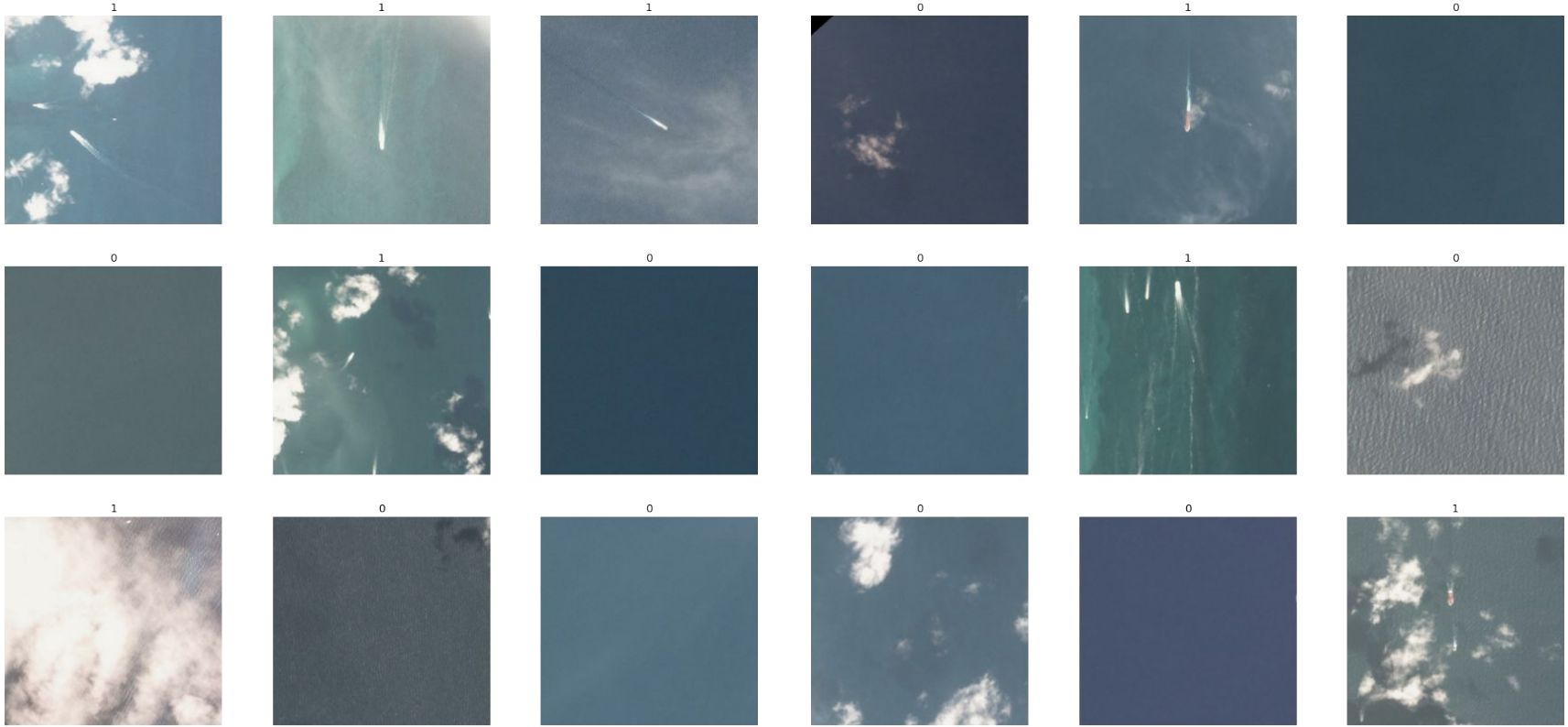
- Erste Erkenntnis: TIFF = pain
  - kaum Unterstützung für TIFF in TF/Keras, sodass der gesamte Datensatz ins RAM geladen werden müsste
  - wenn TIFF-Support, dann nur für 8bit, nicht für 16bit
  - vortrainierte Netze sind in der Regel nur auf RGB(A)-Daten trainiert, nicht auf zusätzlichem NIR-Kanal

  => Umwandeln in jpg (RGB), NIR-Daten entfernen

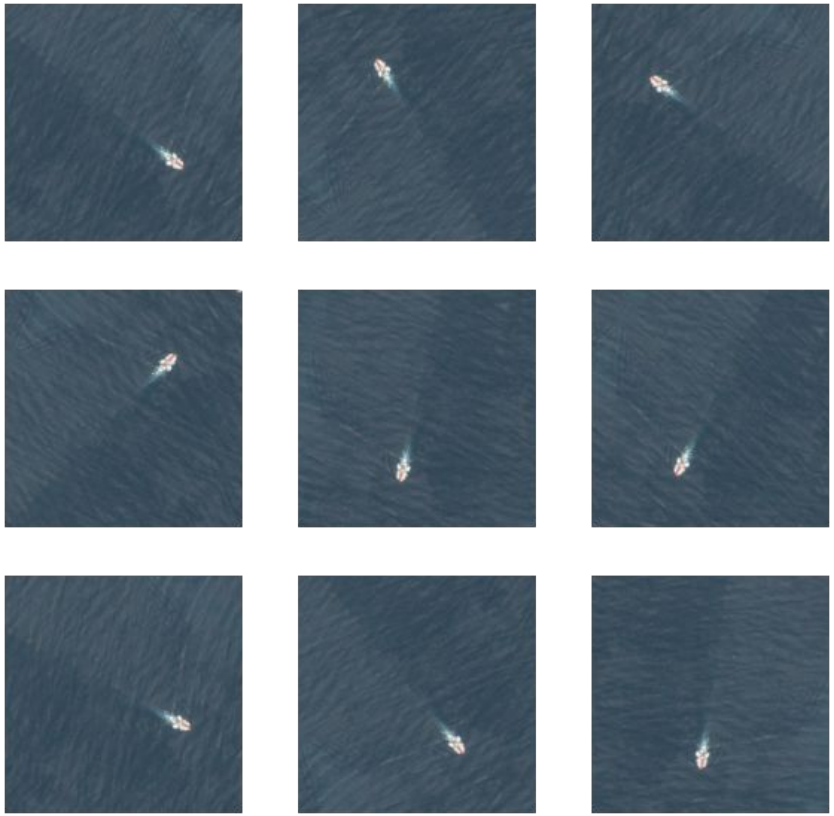
- Zweite Erkenntnis: oft finden sich auf den nonship-Bildern im Außenbereich doch Schiffe, und schwarze Ränder

  => auf Mitte zuschneiden

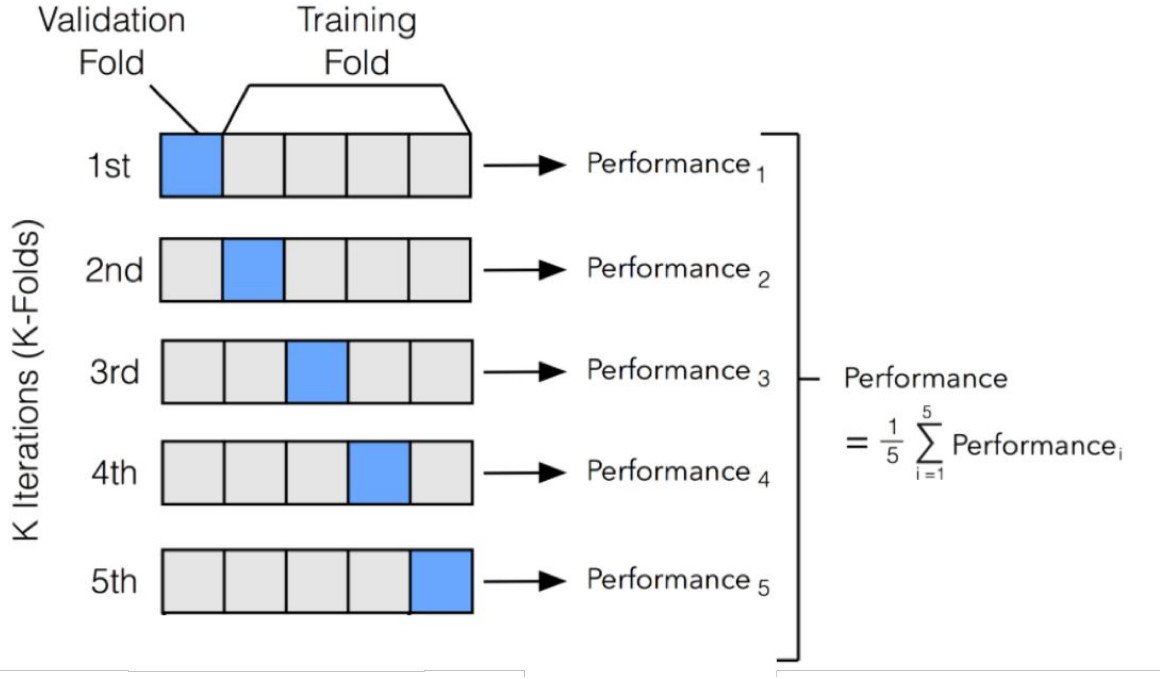
Vorverarbeitete Bilder (jpg, 600x600)



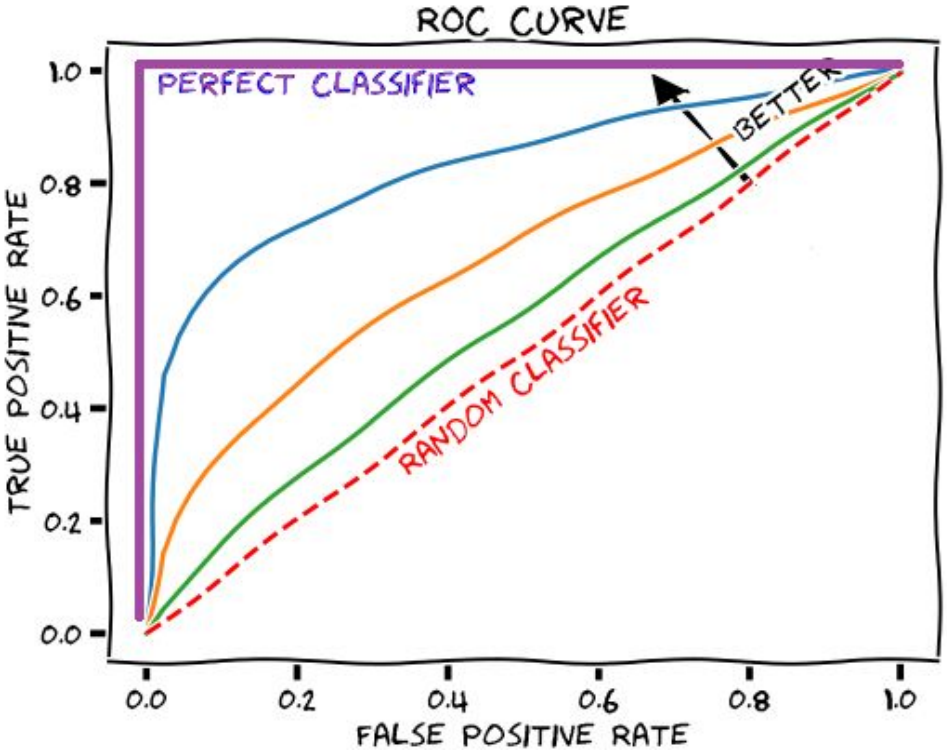
Data Augmentation



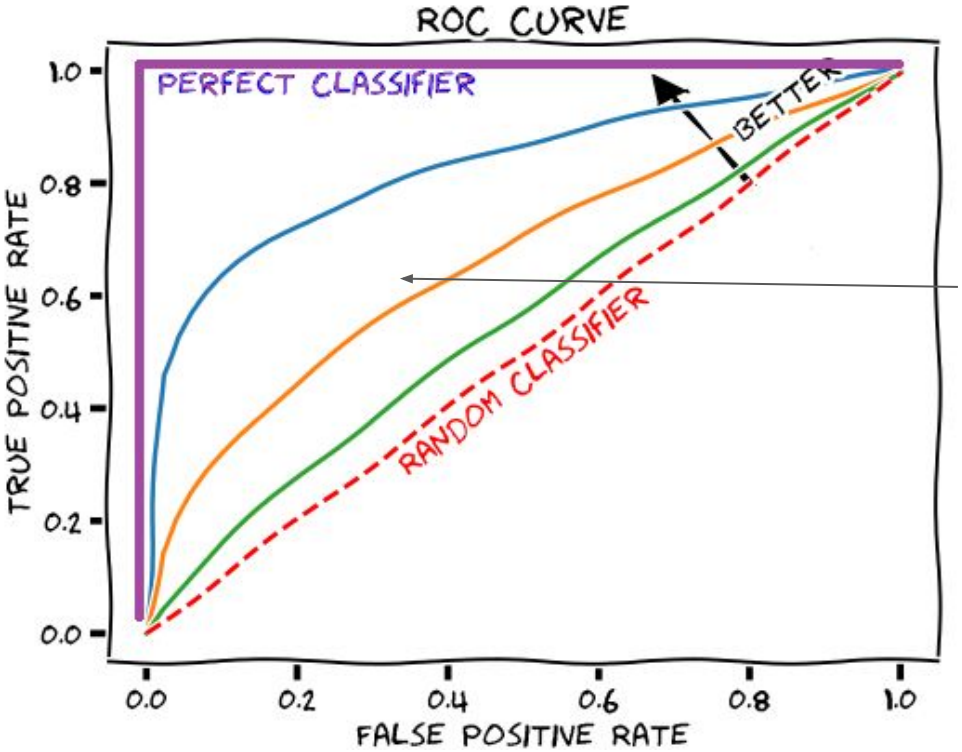
5-fold Crossvalidation



Metriken: ROC

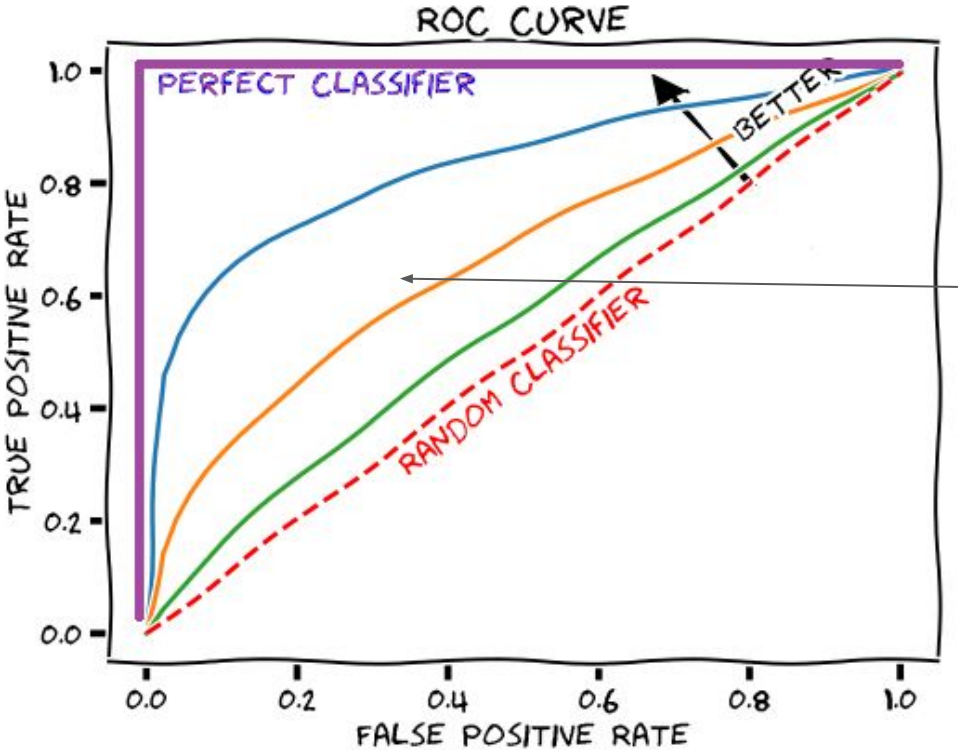


Metriken: ROC, AUC



AUC = Area Under Curve
(random: 0.5, perfect: 1.0)

Metriken: ROC, AUC, F1



AUC = Area Under Curve
(random: 0.5, perfect: 1.0)

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

(random: 0.5, perfect: 1.0)

Getestete Netzwerke

VGG16

Resnet50V2

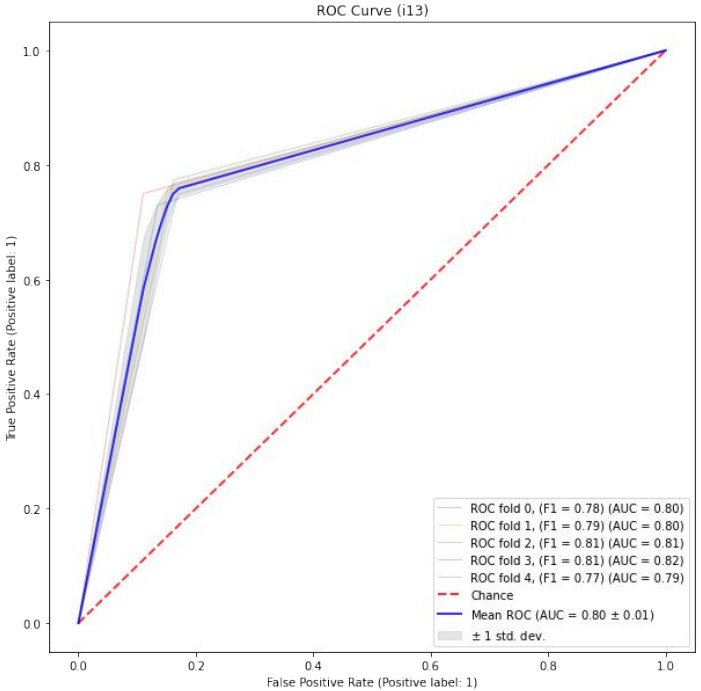
InceptionResnetV2

NASNetLarge

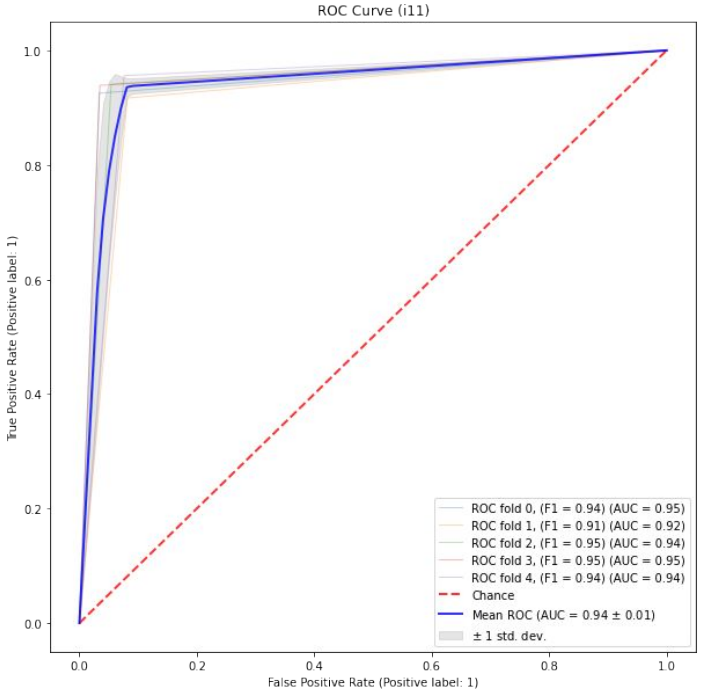
EfficientNetB4

Custom

Ergebnisse

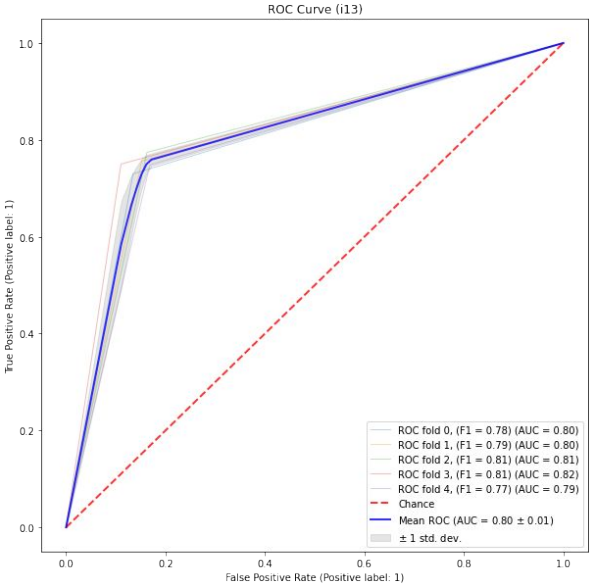


VGG16

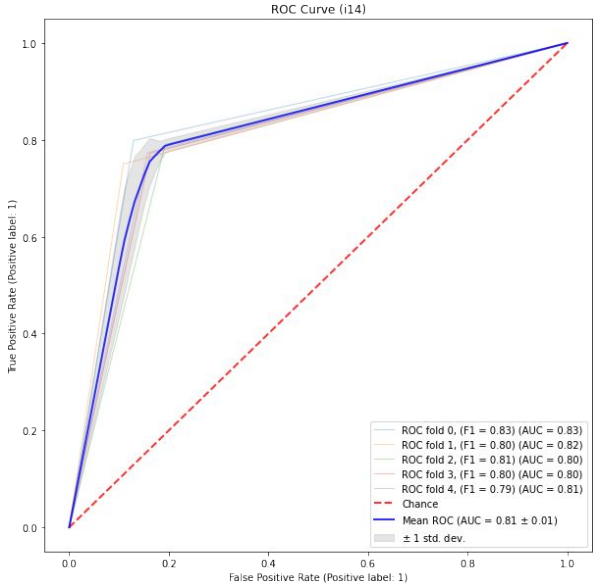


EfficientNetB4
(& InceptionResnetV2)

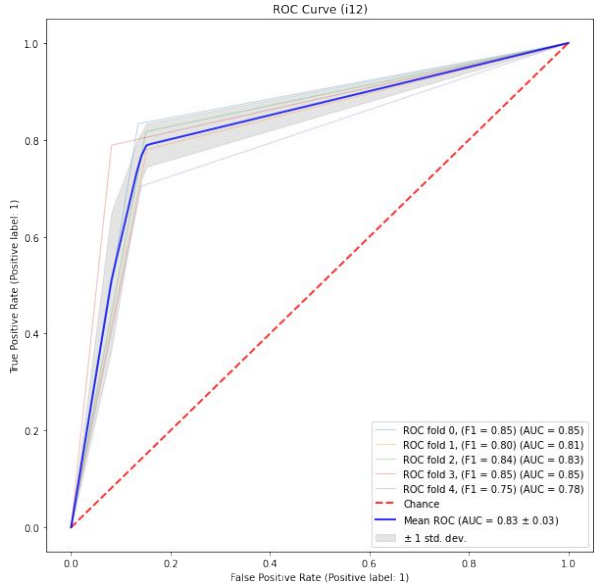
Ergebnis: the bad



VGG16

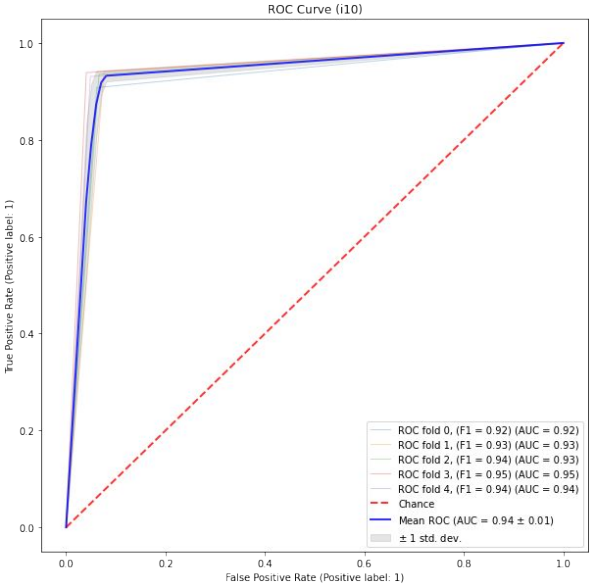


Resnet50V2

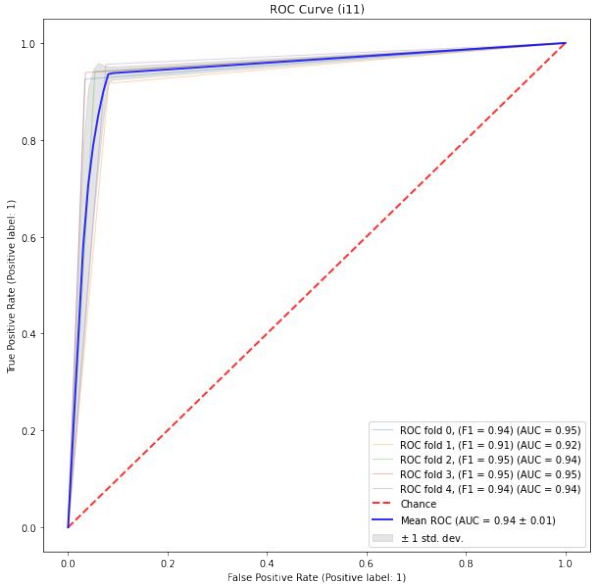


NASNetLarge

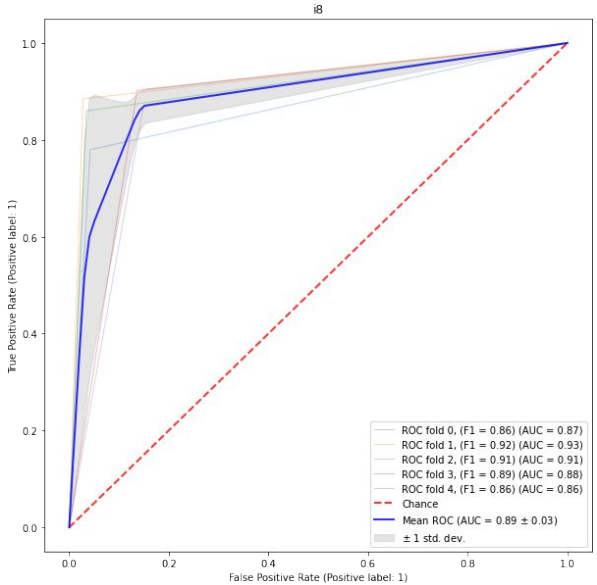
Ergebnis: the good



InceptionResnetV2

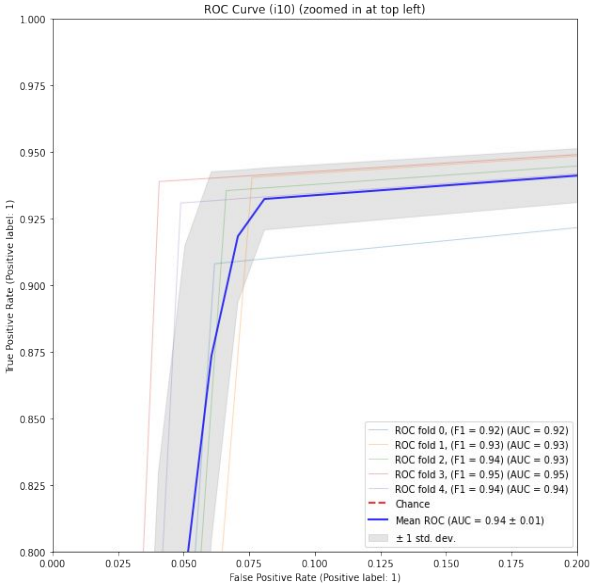


EfficientNetB4

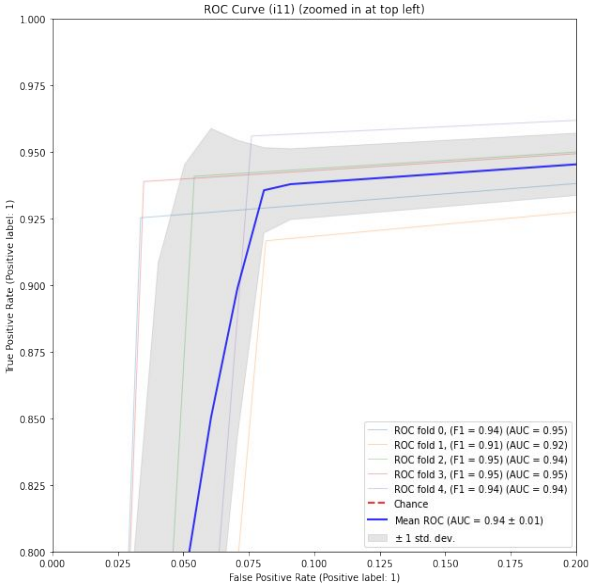


Custom

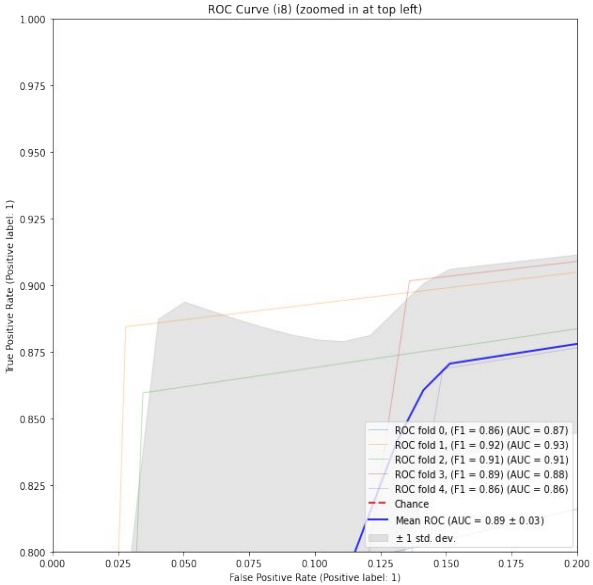
Ergebnis: the good



InceptionResnetV2



EfficientNetB4



Custom

Erkenntnisse

- Eigene Netze zu definieren erfordert Erfahrung, Zeit und iteratives Feintuning
- Einige vortrainierte Netze performen *zu* schlecht. Fehler in der Verwendung?
- Datensatz ist vermutlich zu klein
- Schwierige Problemstellung, gelegentlich ist Fragestellung ship/nonship nicht eindeutig beantwortbar (Wolken)
- Ideen:
  - Dritte Klasse definieren?
  - Auf größerem Datensatz trainieren, z.B. Airbus Ship Detection Challenge, und Übertragbarkeit testen
- Ergebnis: Mit vortrainierten Netzen sind mit relativ geringem Aufwand und minimalem spezifischen Fachwissen brauchbare Leistungen erreichbar

Fragen?

Bildquellen:

https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

http://ethen8181.github.io/machine-learning/model_selection/model_selection.html