

# Wolkenkamera: Big Data Projekt

Marcel Steger, Jan Zickermann

Universität Hamburg

19. März 2018

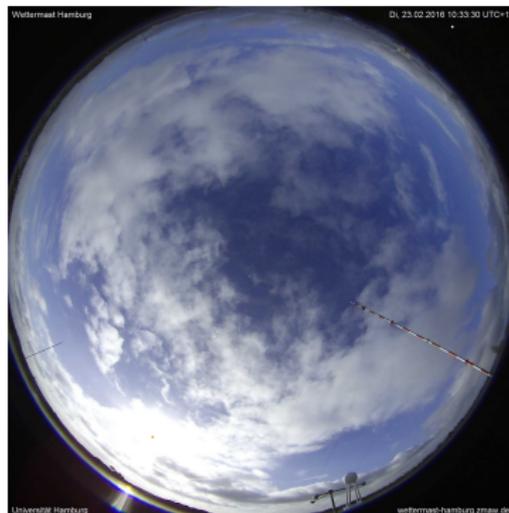
Betreuer: Tobias Finn, Dr. Julian Kunkel

# Übersicht

- 1 Projektthema
- 2 Methode
  - Messdaten
- 3 Model
  - VGG16
- 4 Ergebnisse
  - Accuracy

# Projektthema: Wolkenkamera

Ziel: anhand von Photographien einer stationären Kamera die Höhe einer Wolke zu ermitteln. Hierfür sollen neuronale Netze verwendet werden.



*python*: tensorflow (machine learning), scipy/numpy (preprocessing)

# Verfügbare Messdaten

Zuordnungen des *Wolkenbildes* mit Zeitstempel:

- Wolkenhöhenmessung (Ceilometer)
- Regenmenge (Regensensor)
- Sonneneinstrahlung

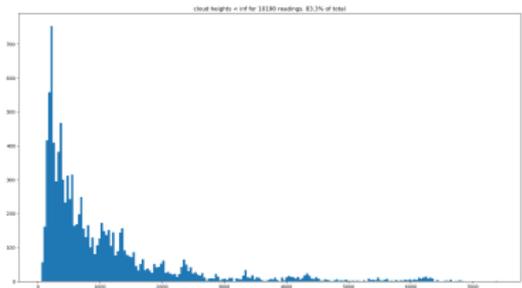
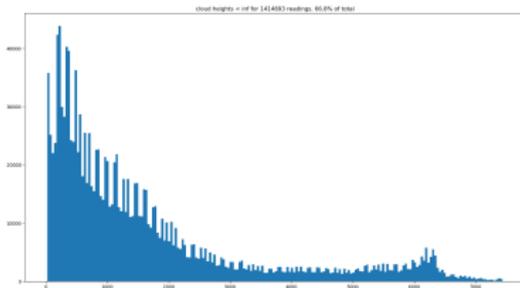
value	function
height	nearest
rain_detection	most_frequent
rain_quantitiy	max
sun_detection	most_frequent
sun_insolation	mean

# Verwendbarkeit der Messdaten

- Wolkenbilder
  - Mit Sonnenwinkel  $> 7^\circ$
  - Präzise Bildprojektion mit Polynom
- Wolkenhöhenmessung (Ceilometer)
  - Klarer Himmel  $\rightarrow$  Höhenmessung:  $\infty$
  - Keine eindeutige Regression für  $\infty$
  - Einteilung in ausgewogene Höhenklassen
- Regenmenge (Regensensor)
  - Tropfen auf Linse, obwohl  $RD = 0$
  - $RR > 0$  ist selten (nicht für Training geeignet)
- Sonneneinstrahlung
  - Einfach verteilte Messwerte
  - Schwache Verbindung zur Wolkenhöhe

# Verteilung der Messdaten

## Höhenverteilung:



## Regenmengen- und Sonneneinstrahlungsverteilung:

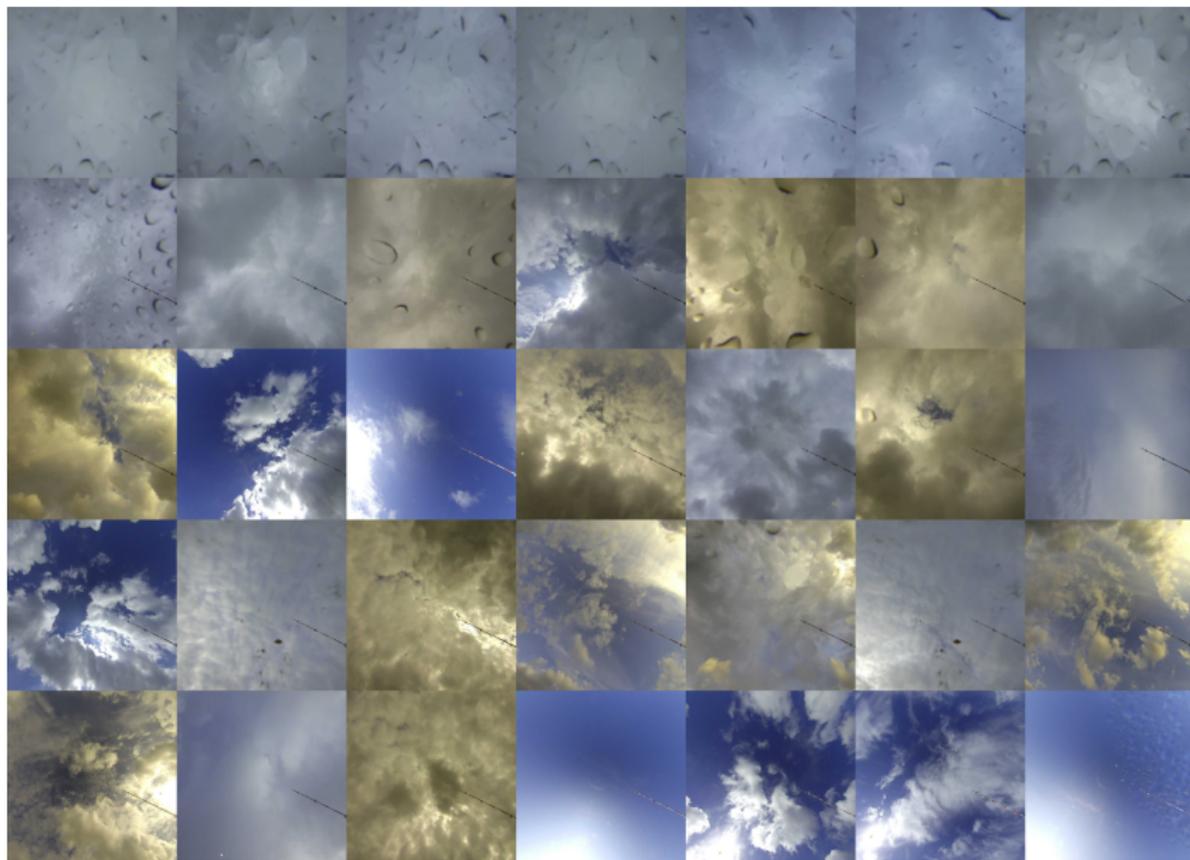


# Höhenklasseneinteilung (*bins*)

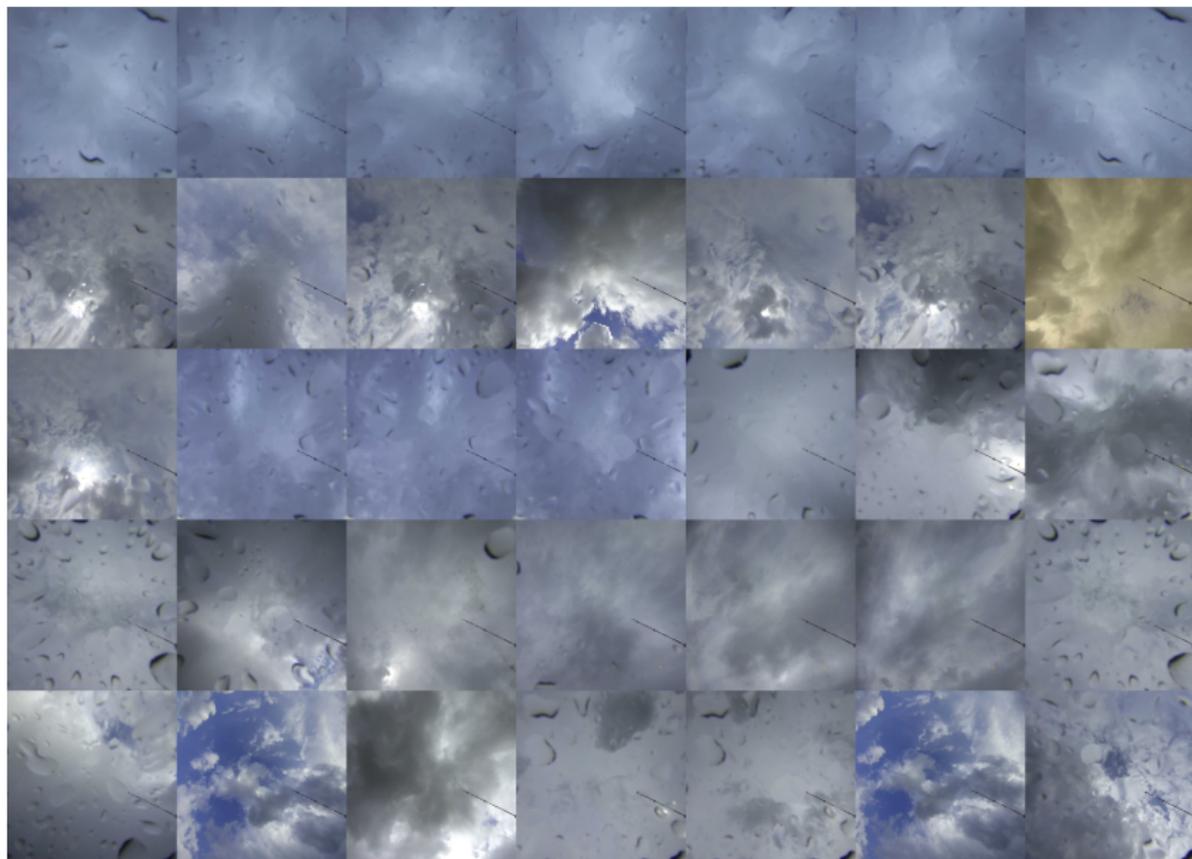
- Ausbalancierte Höhenklassen
  - Klasse  $\infty$ : kann nicht unterteilt werden
  - Höhenklassen haben gleiche Anteile der Höhenverteilung
  - Verteilung basierend auf **train/validation** Höhen

height_class	range
0	$\leq 360$
1	$\leq 840$
2	$\leq 1530$
3	$> 1530$
4	$\infty$

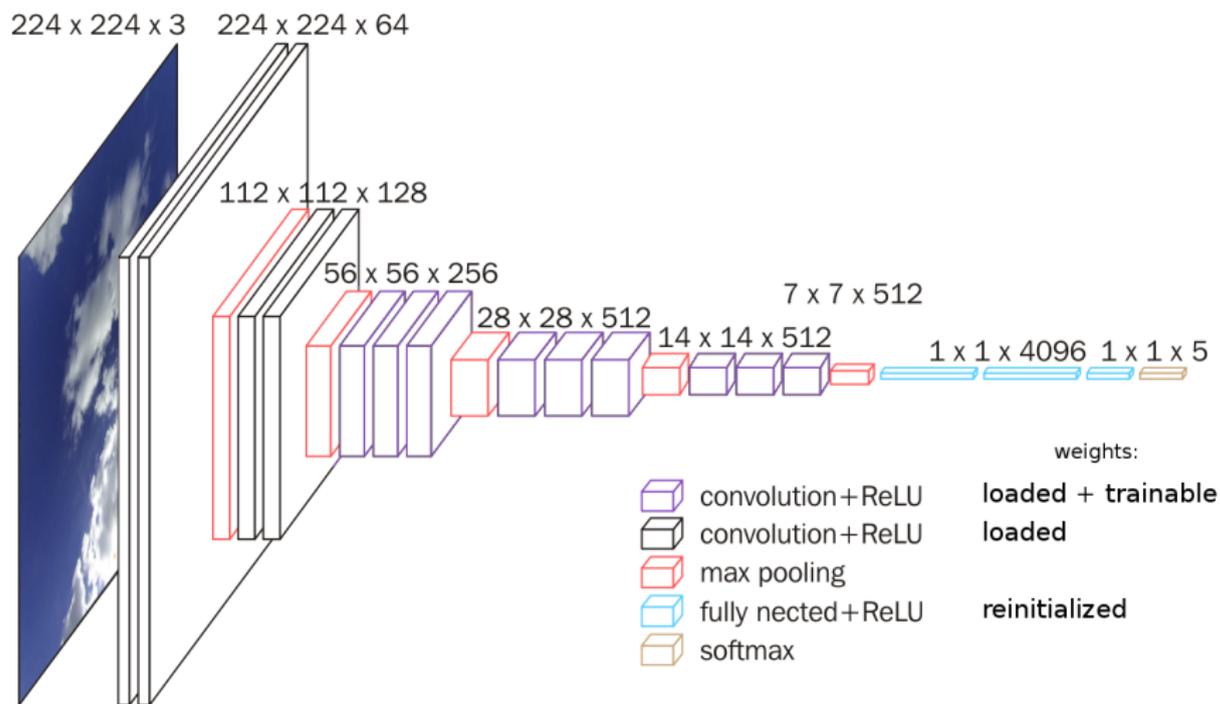
# Höhenklassen Bilderbeispiele ( $RD = 0$ )



# Höhenklassen Bilderbeispiele ( $RD = 1$ )



# VGG16 *transfer learning*



[2]

## CNN training

- Dataset ist aufgeteilt in  $\frac{1}{2}$  *train* und  $\frac{1}{2}$  *validation (shuffled)*
- 224x224x3 Wolkenbild  $\rightarrow$  5 Höhenklassen
- Klassifikation: *AdamOptimizer* minimiert  
*loss: sparse\_softmax\_cross\_entropy\_with\_logits*
- *Overfitting* vorbeugen
  - *Dropout*: für beide *fully connected layer*
  - *Augmentation*: mit zufälligen Bildspiegelungen
  - Initial *underfitting* verhindern

# Vorherige Herangehensweisen

## 1 Einfache Höhen Regression

- $\infty$  Höhen werden auf  $2 \cdot \max_{1 \leq i \leq n} h_i$  gesetzt
- $loss = |y - t|$ , oder  
 $loss = |\log(y + 1) - \log(t + 1)|$

## 2 Höhen Regression + Höhen Klassifikation

- 10 Höhenklassen (nicht ausbalanciert)

$$loss_{reg} = \begin{cases} |\log(y + 1) - \log(t + 1)| & \text{falls } t \leq \infty \\ 0 & \text{sonst} \end{cases}$$

## 3 Höhen Reg. + Regenmenge Reg. + Sonneneinstrahlung Reg. + Höhen Klassifikation

- Angepasste Gewichtungen der Summanten für  $loss$  Summe

## Durchlauf mit höchster Accuracy

- 1235 steps = 1 Epoche, batch size = 10
- 4 Epochen mit keinem *dropout* + 5 Epochen mit 0.5 *dropout*
- Batch accuracy train/ validation:

accuracy/batch\_accuracy



## Accuracy der Höhenklassen

height_class	accuracy
$\leq 360$	0.87
$\leq 840$	0.61
$\leq 1530$	0.64
$> 1530$	0.56
$\infty$	0.66
all	0.67

- Berechnung basiert auf *validation* Daten

# Abschluss

- Erreichte Ergebnisse
  - Datenverarbeitung  $\rightarrow$  CNN  $\rightarrow$  Höhenklassen
  - Model Lernerfolg:  $Accuracy = 0.67 > 0.20$
- Ausstehendes
  - Dokumentation, Bericht
  - kein GAN aufgrund von Zeitknappheit

# Quellen

-  VGG16 weights, 18.03.2018: [https://www.cs.toronto.edu/~frossard/vgg16/vgg16\\_weights.npz](https://www.cs.toronto.edu/~frossard/vgg16/vgg16_weights.npz)
-  Karen Simonyan, Andrew Zisserman. (2014). Very Deep Convolutional Networks For Large-Scale Image Recognition. CoRR  
<http://arxiv.org/abs/1409.1556>

## Bildquellen:

-  Wolkenbild mit Regen, 05.11.2017 : [https://github.com/LEX2016WoKaGru/pyclamster/blob/master/examples/images/wettermast/Image\\_Wkm\\_Aktuell\\_1.jpg?raw=true](https://github.com/LEX2016WoKaGru/pyclamster/blob/master/examples/images/wettermast/Image_Wkm_Aktuell_1.jpg?raw=true)
-  VGG16, 15.03.2018:  
<https://www.packtpub.com/graphics/9781786462961/graphics/747e2d04-e8c2-49ee-8767-a76f3934f13e.png>