# Detecting Convective Clouds in Geostationary Satellite data with Convolutional Neural Networks

Dr. William Clemens
dida Datenschmiede GmbH

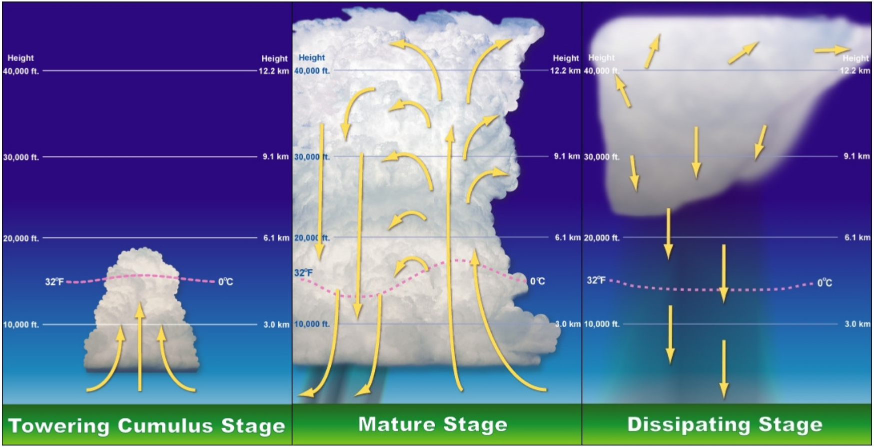
27, January 2020

# Overview

- Problem
- Methodology
- Results
- Conclusions and potential future work

# Problem

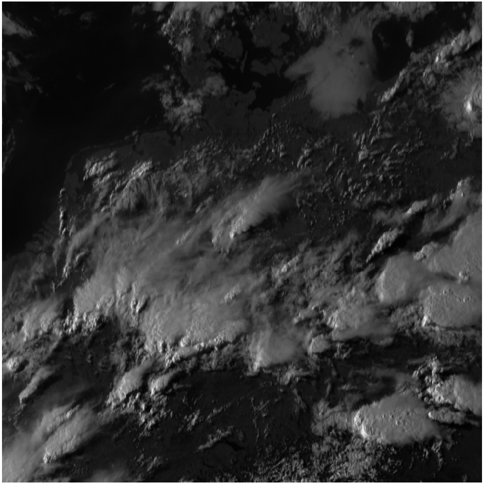
# What are Convective Clouds?




Source: Wikipedia / NOAA

# Meteosat Second Generation

- Geostationary satellites that take very high temporal resolution (5min or 15min) but low spatial resolution images (∼1km or ∼3km).



# Methodology

# Methodology

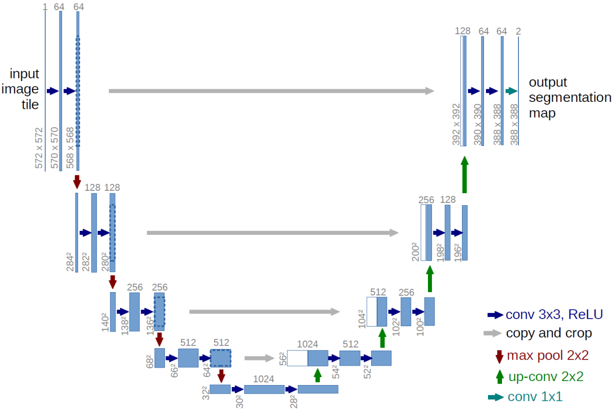
- We want to predict a class for each pixel in our image.

# Methodology

- We want to predict a class for each pixel in our image.
- In order to do this we use a convolutional neural network architecture called a U-Net:

# Methodology

- We want to predict a class for each pixel in our image.
- In order to do this we use a convolutional neural network architecture called a U-Net:




arXiv:1505.04597

## Labelling

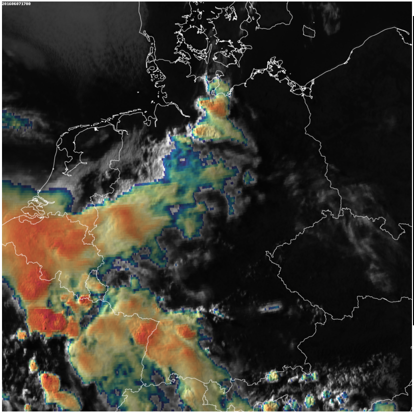
- There is no real ground truth unfortunately.

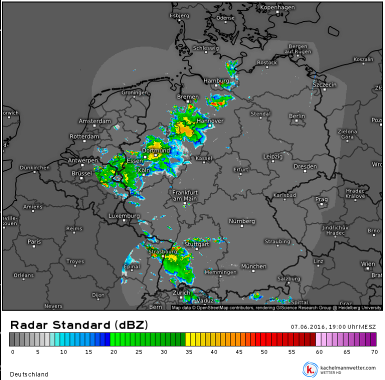
## Labelling

- There is no real ground truth unfortunately.
- We needed to get labelled images. This unfortunately needed to be done by hand.

# Labelling

- There is no real ground truth unfortunately.
- We needed to get labelled images. This unfortunately needed to be done by hand.
- We present the labellers with a false colour image that combines the HRV and one IR channel as well as a radar image and ground observations.





# Data Augmentation

- Labelling data is slow.

# Data Augmentation

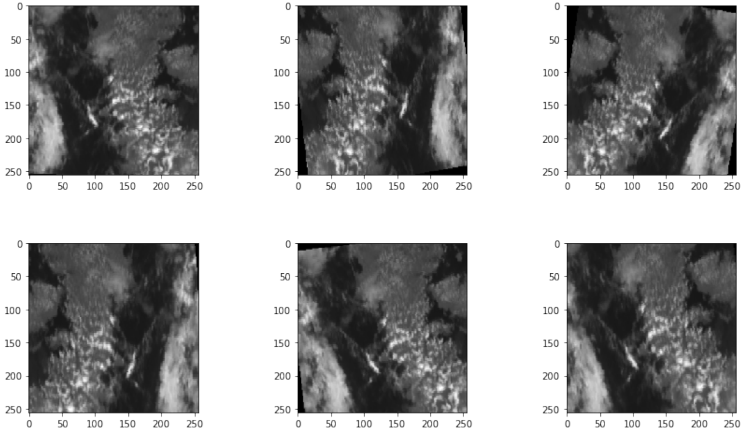
- Labelling data is slow. We obtained ~1750 images.

# Data Augmentation

- Labelling data is slow. We obtained ∼1750 images.
- We need to make the most of what we have.

# Data Augmentation

- Labelling data is slow. We obtained ~1750 images.
- We need to make the most of what we have.
- Make random transformations to the data and labels and feed them to the network.



# Loss function

- We us the smooth dice loss for our loss function.

# Loss function

- We us the smooth dice loss for our loss function.
- $$1 - \frac{2\sum y_t y_p + 1}{\sum y_t + \sum y_p + 1} \tag{1}$$
- Where $y_t$ are the true labels and $y_p$ are the predicted labels.

## Loss function

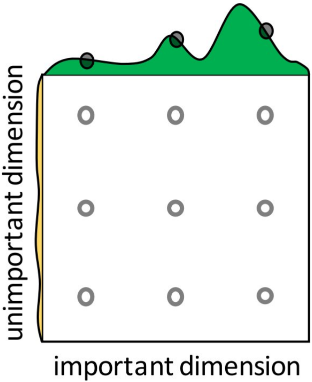
- We us the smooth dice loss for our loss function.
- $$1-\frac{2\sum y_t y_p+1}{\sum y_t+\sum y_p+1} \tag{1}$$
- Where $y_t$ are the true labels and $y_p$ are the predicted labels.
- This leads to better convergence in this case than the more common binary cross entropy.

# Hyperparameters

We have a lot of hyperparameters when defining our model:

We have a lot of hyperparameters when defining our model:

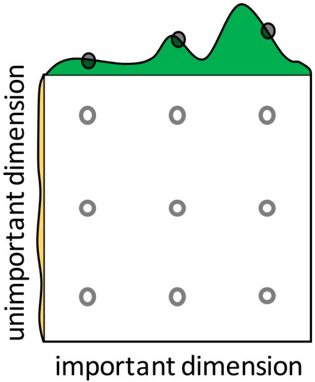
| Hyperparameters | Possible values |
|---|---|
| Data Augmentation | flips, maximum angles, maximum zoom... |
| Number of "recursions" for the UNET | 1, 2, 3, 4, 5... |
| Nonlinearity | ReLU, ELU, CELU... |
| Method of upsampling | Transposed convolution, pixel shuffle |
| Optimisation algorithm | RMSProp, Adam... |
| Dropout percentage | 0, 0.2, 0.5... |
| Batch size | 4, 8, .... |
| Learning rate | 0.001, 0.005 ... |
| Learning rate annealing | none, one cycle, cosine... |




Source: Yao, Quanming et al. (2018).

- The solution is to do a hyperparameter search.

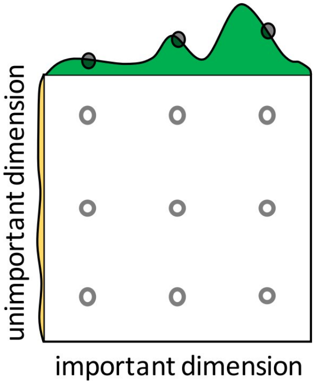
# Hyperparameter optimisation



Source: Yao, Quanming et al. (2018).

- The solution is to do a hyperparameter search.
- It's better to do a random search than a grid search.

## Hyperparameter optimisation

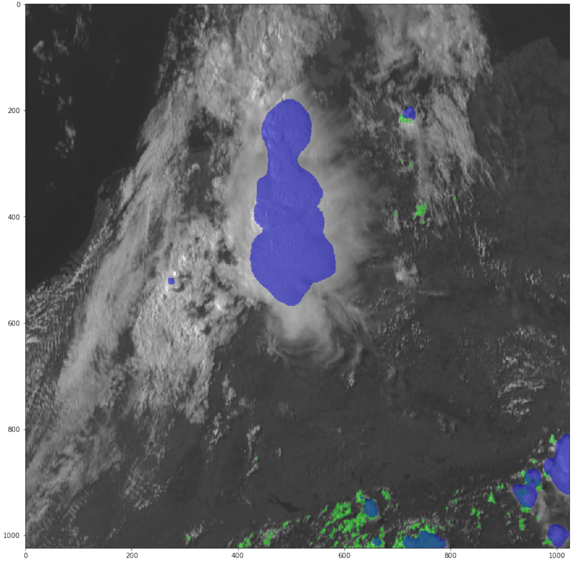



Source: Yao, Quanming et al. (2018).

- The solution is to do a hyperparameter search.
- It's better to do a random search than a grid search.
- We NEED to have a separate validation dataset here.

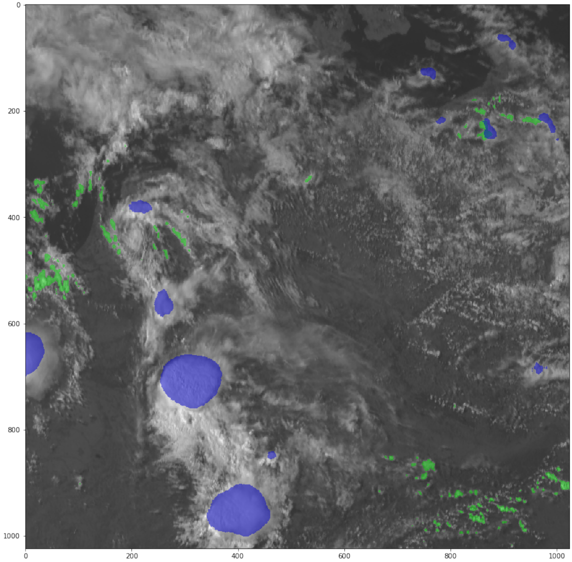
# Results

# Results




201206291500

# Results




201206281500

## Results

- Measuring performance is difficult.

# Results

- Measuring performance is difficult.
- We perform very well against a test set drawn from the labels ( 98% accuracy)

# Results

- Measuring performance is difficult.
- We perform very well against a test set drawn from the labels ( 98% accuracy)
- However assessing the performance of the labels themselves is not trivial. (No real ground truth.)

# Extensions and Conclusion

# Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.

# Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.
- Incorporate physics / meteorology knowledge.

# Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.
- Incorporate physics / meteorology knowledge. (Either through pre or post processing for the network)

# Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.
- Incorporate physics / meteorology knowledge. (Either through pre or post processing for the network)
- Take advantage of the time series nature of the data.

## Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.
- Incorporate physics / meteorology knowledge. (Either through pre or post processing for the network)
- Take advantage of the time series nature of the data.
- Investigate different model architectures e.g. DeepLabV3.

# Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.
- Incorporate physics / meteorology knowledge. (Either through pre or post processing for the network)
- Take advantage of the time series nature of the data.
- Investigate different model architectures e.g. DeepLabV3.
- Investigate using non solar channels for night time predictions.

# Extensions

- Improve training data. Ideally use some automatically acquired data that removes human error.
- Incorporate physics / meteorology knowledge. (Either through pre or post processing for the network)
- Take advantage of the time series nature of the data.
- Investigate different model architectures e.g. DeepLabV3.
- Investigate using non solar channels for night time predictions.

# Conclusions

- Convolutional neural networks are well suited to processing this kind of data.

## Conclusions

- Convolutional neural networks are well suited to processing this kind of data.
- In particular they can take advantage of shape information on scales larger than one pixel to differentiate features that look similar in raw pixel values.

## Conclusions

- Convolutional neural networks are well suited to processing this kind of data.
- In particular they can take advantage of shape information on scales larger than one pixel to differentiate features that look similar in raw pixel values.