Simple Cloud Cover Classification CNN

Seminar Pattern Recognition

Tishana Suthenthiran

03.12.2025, University of Bern

Introduction

GOALS OF THE PROJECT

- The aim of this project is to build a simple and effective image classifier using convolutional neural networks (CNNs).
  - Goal 1: Work with an existing dataset of sky images.
  - Goal 2: Train a CNN for multi-class classification of sky images.
  - Goal 3: Evaluate the model’s accuracy and present results .

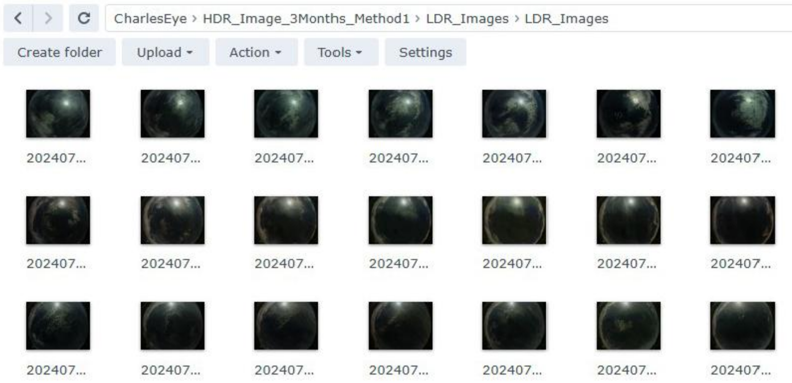
The CNN will be used for further work in irradiance prediction.

b
UNIVERSITÄT
BERN

Behind the Scenes

Fereshteh’s PhD Project

Camera setup → Image Acquisition → HDR reconstruction → Image compression

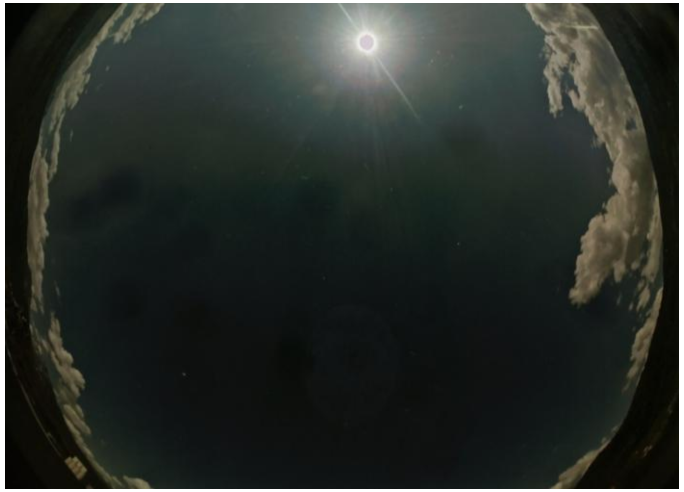


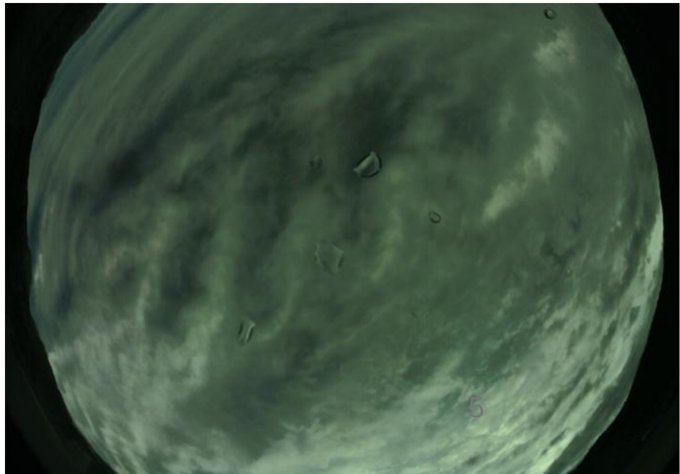
UNIVERSITÄT
BERN

My Task: Cloud Cover Classification

- Images from July to end of September 2024 from 6am to 7.55pm
- 12GB!!
- We have a large dataset of sky images, but no automatic method to classify them into labels like
  - Clear
  - Cloudy
  - Overcast

= CLOUD COVER





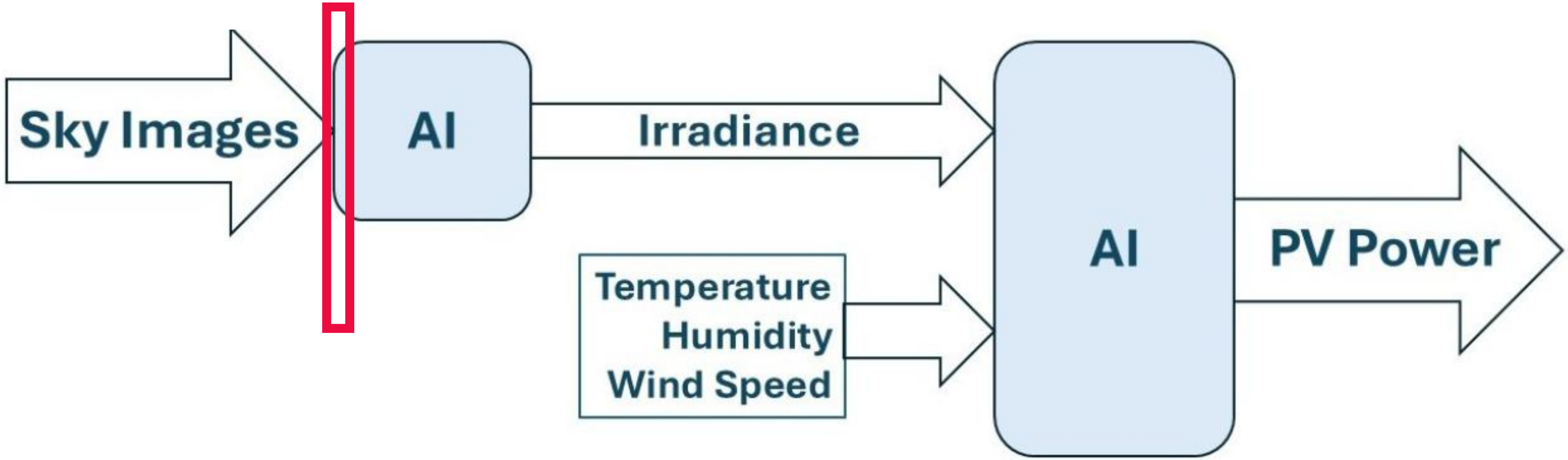
Cloud cover ←→ Irradiance

UNIVERSITÄT BERN

- The main source of uncertainty is cloud movement.
- Clouds dominate short-term PV variability and irradiance.
  - Classifying the sky gives immediate information about expected irradiance behaviour
  - Shadows reduce PV output by 50%-80% within minutes.
- Cloud classes describe distinct PV output patterns
  - Clear sky → smooth, high irradiance
  - Cloudy sky → rapid fluctuations, shading events
  - Overcast sky → stable but low irradiance

CNN as Preprocessing Step

- The CNN is not the final forecaster — it is the tool that turns raw images into useful inputs for downstream irradiance prediction.



Preprocessing

Preprocessing

UNIVERSITÄT BERN

- **Purpose**
  - Ensure only clean, informative images enter the training dataset
  - Remove frames that would confuse or degrade the CNN – sensitive to noise

Time Window 0600 - 0620 → Raindrop CNN + SVM → Color Dominance

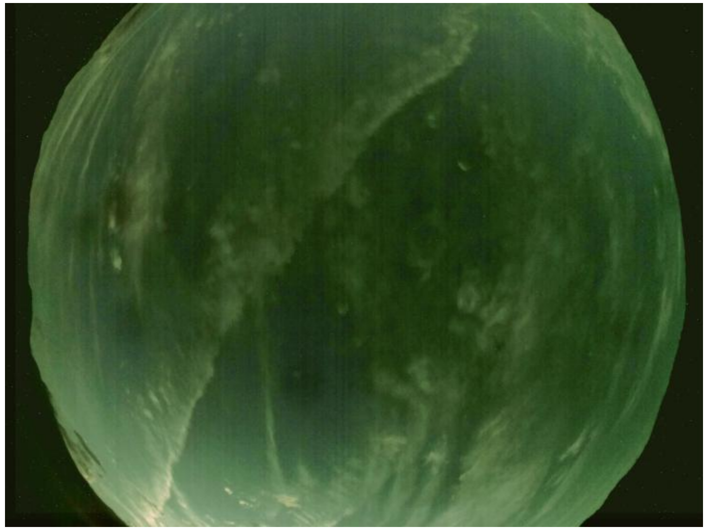
Time Window 0600 - 0620

Early morning images are often

- Heavily color-shifted (purple/blue)
- Too dark
- Clouds not visible

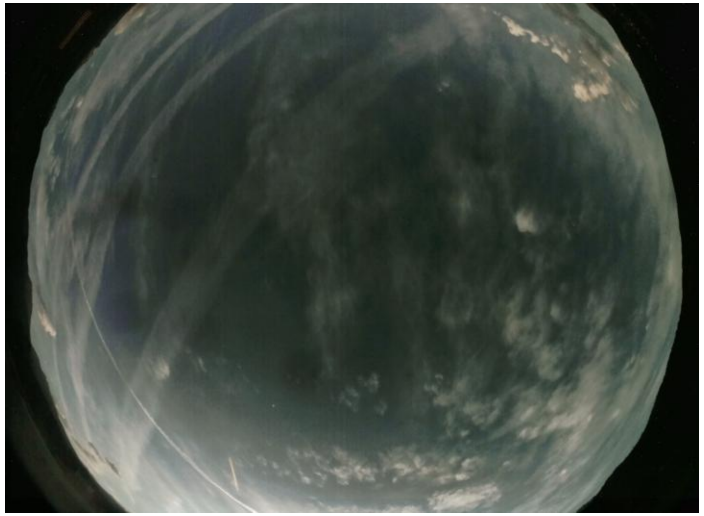
→ Can confuse CNN during training

06.07.2024 06:00

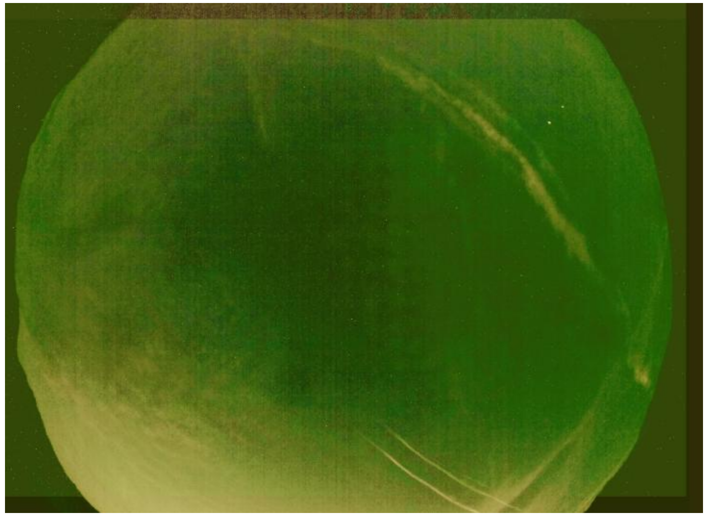


13.07.2024 06:20

06.07.2024 06:20

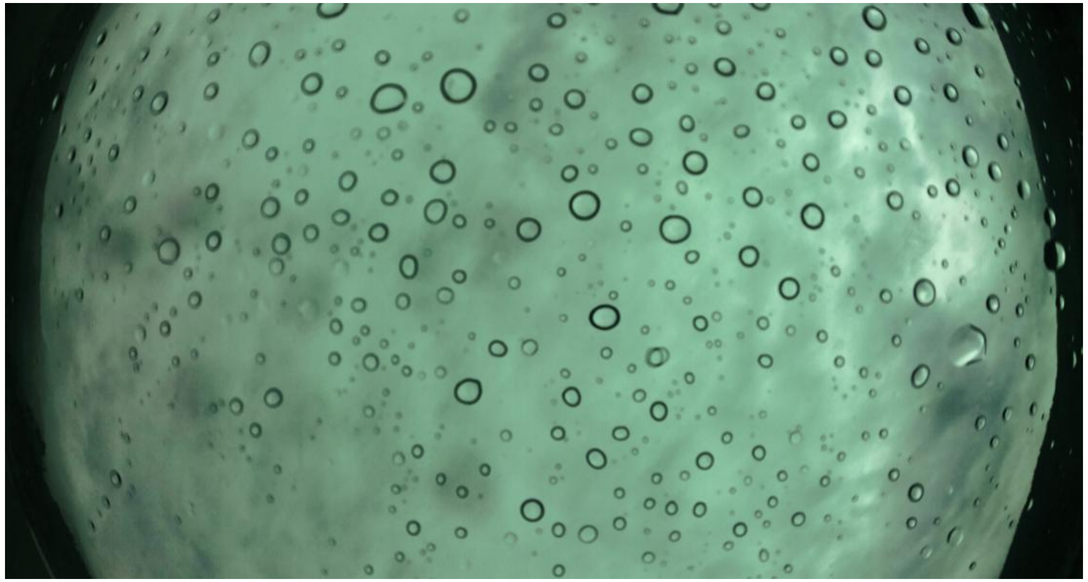


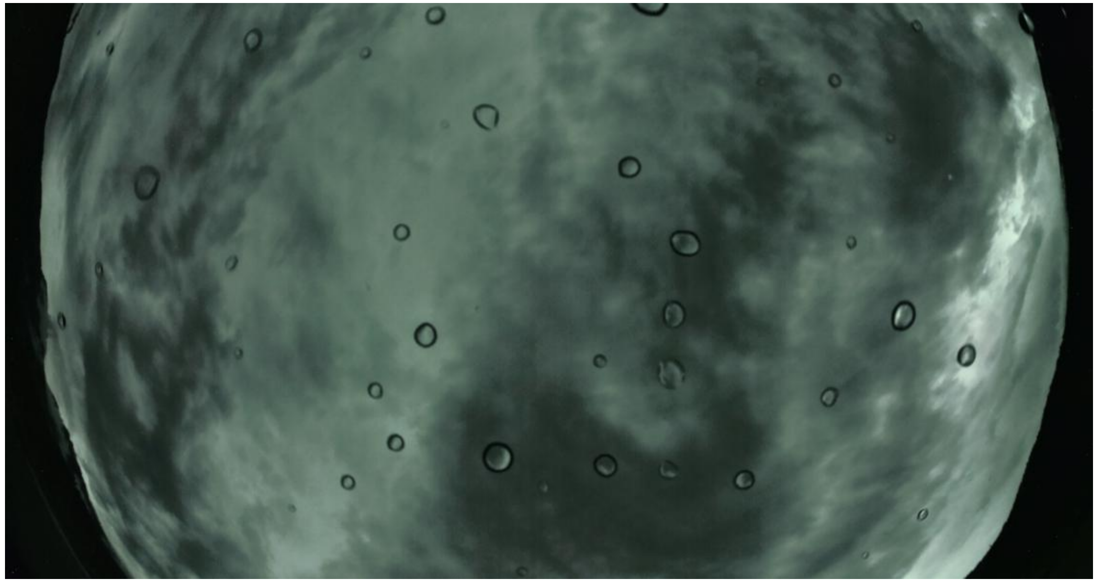
25.07.2024 06:10

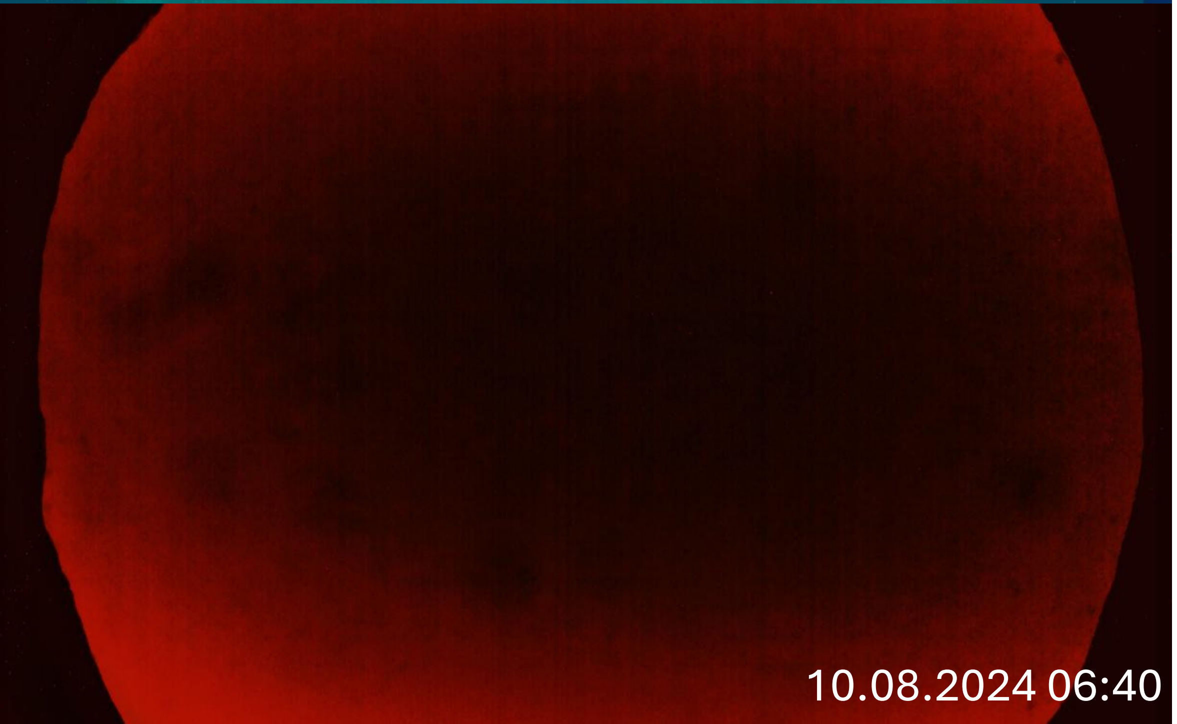


Raindrop
CNN + SVM

- Raindrops on the lens:
  - Distort the cloud structure
  - Create blurry spots unrelated to the sky
  - Can be mistaken as clouds
- Pretrained CNN + SVM classifier







Color Dominance

Algorithm:

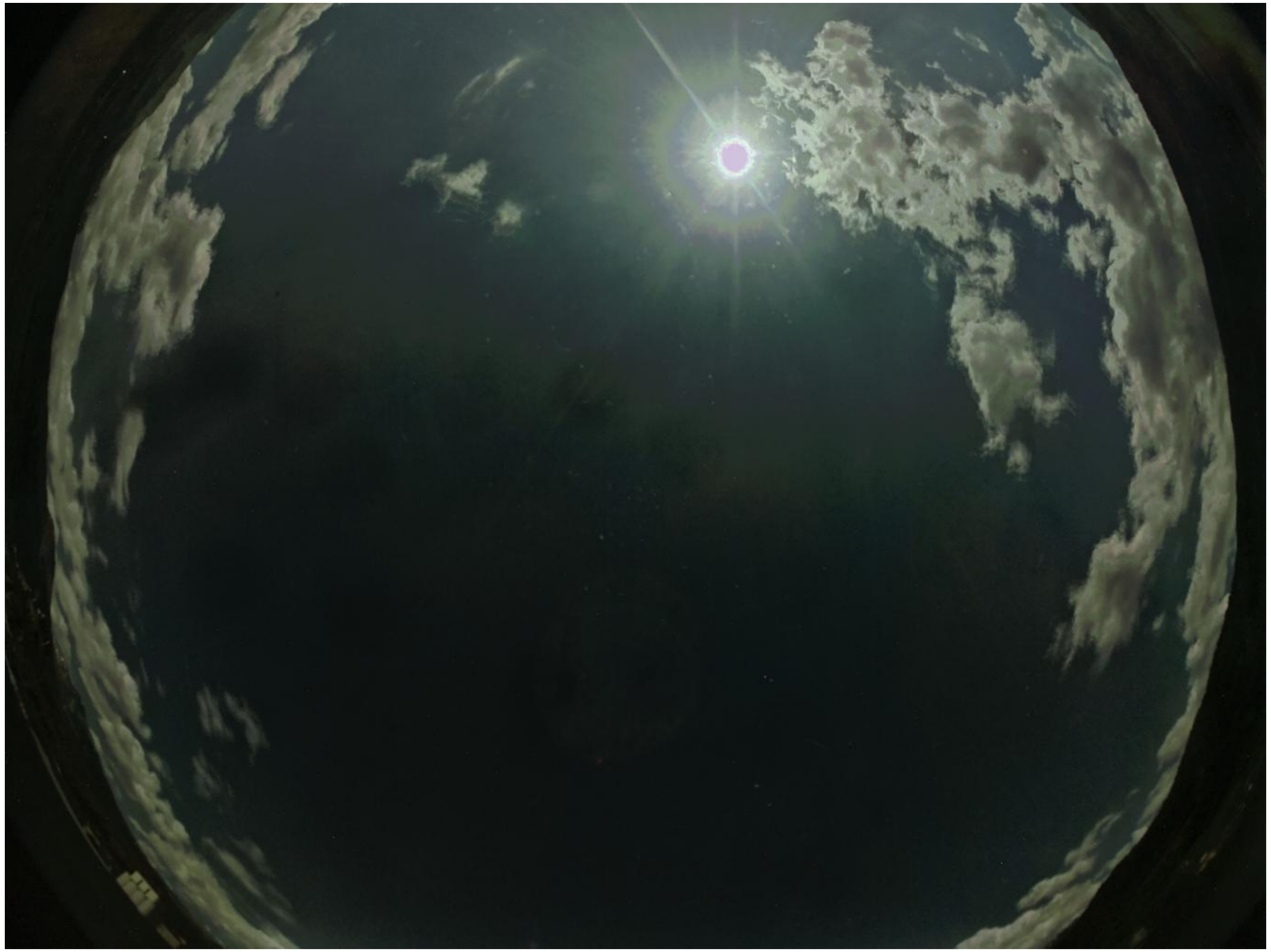
1. Compute mean color value of each channel:
   - Blue, Green, Red
2. Compute each channel’s ratio
   - ratios = [mean_B/total, mean_G/total, mean_R/total]
3. Sort the ratios
   - top_two = highest + second_highest
4. If the top two channels take **> 90%** of the total colour:
   The image is **colour-dominated**
   → move to quarantine/dark_or_tinted

Labelling

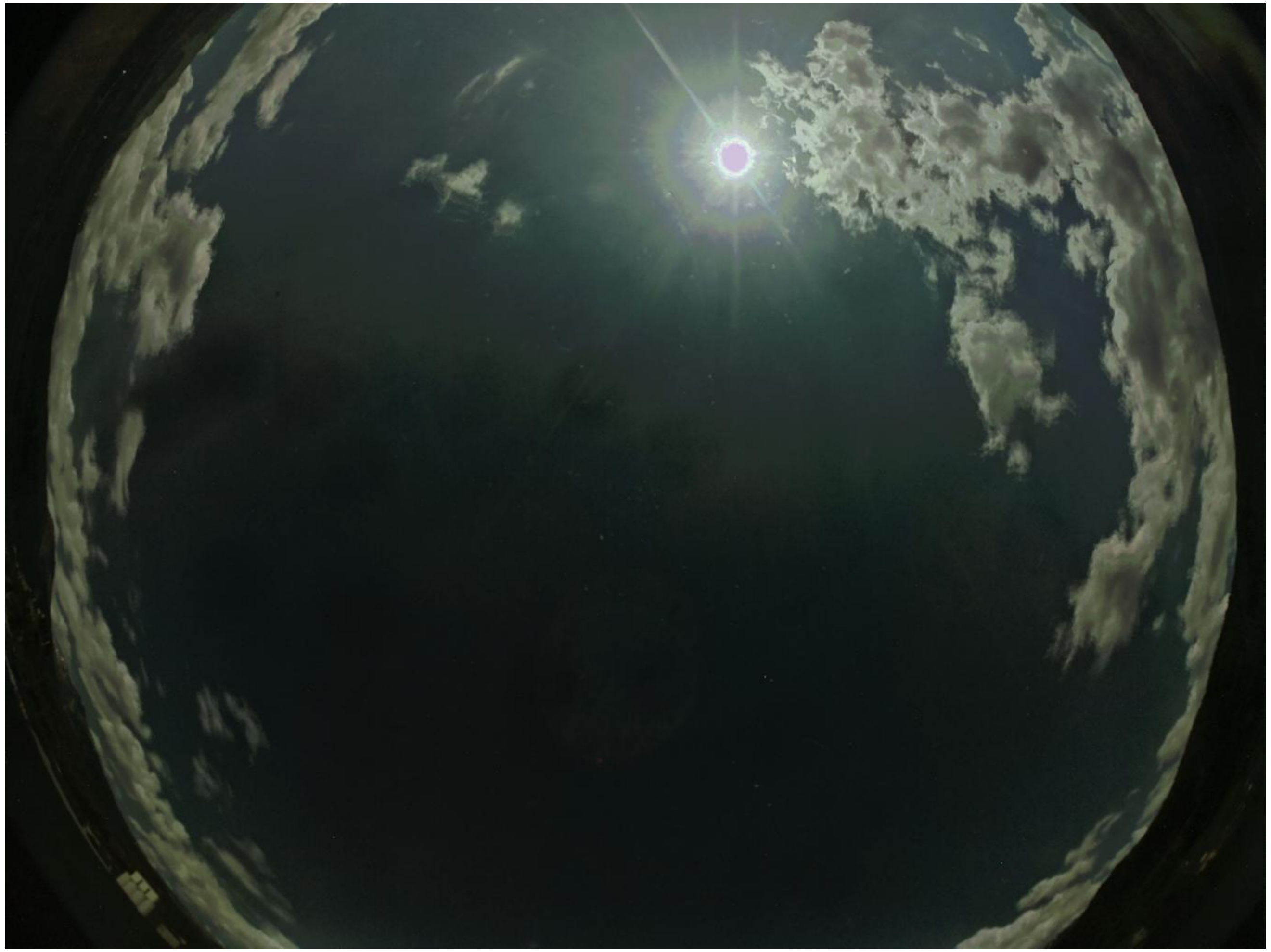
Labelling

- **Goal**
  - Assign each sky image to the correct **cloud-cover class** (Clear, Cloudy, Overcast)
- Images were labelled manually using the same labelling criteria.
- Out of 14’000 images, 3000 images were labelled → 1000 images per class.

Clear, cloudy or overcast?



Clear, cloudy or overcast?



Cloudy

Clear, cloudy or overcast?

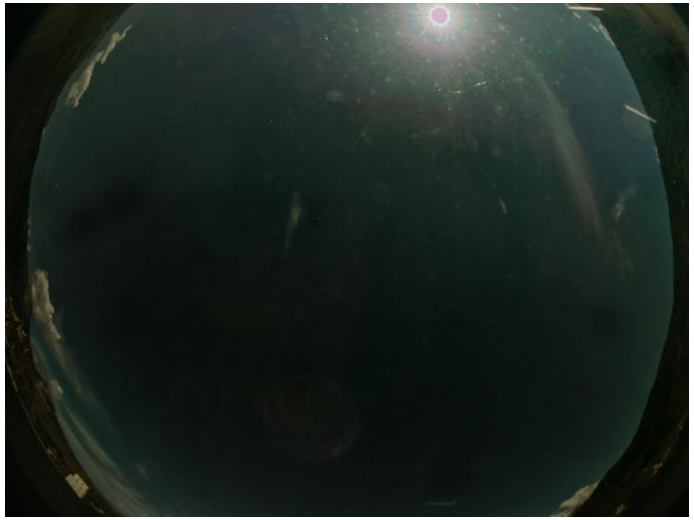
Clear, cloudy or overcast?

Cloudy

Class Clear

- Sky is almost entirely free of clouds. Direct sunlight reaches the ground with little scattering.
- Sun is visible and unobstructed – no clouds near the sun.
- 0 – 30% cloud coverage

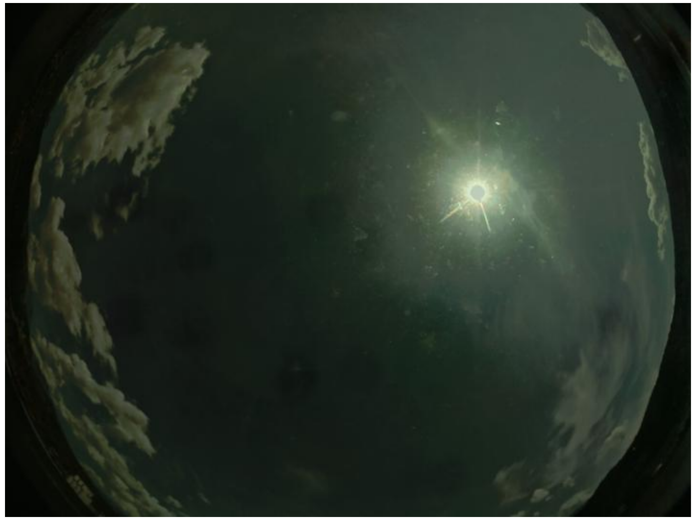


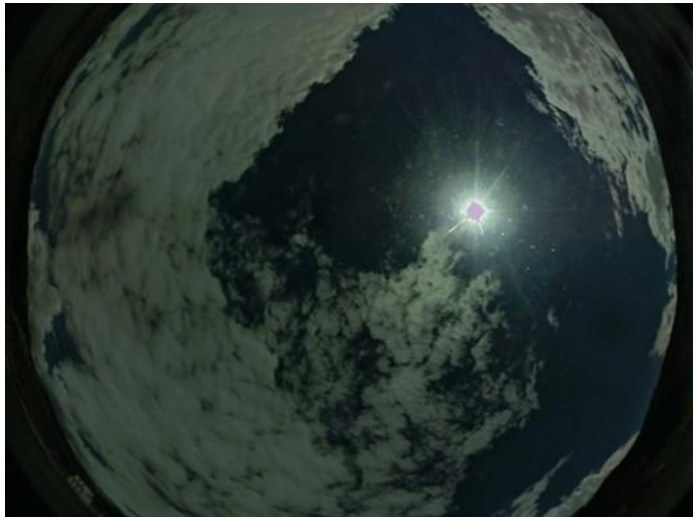


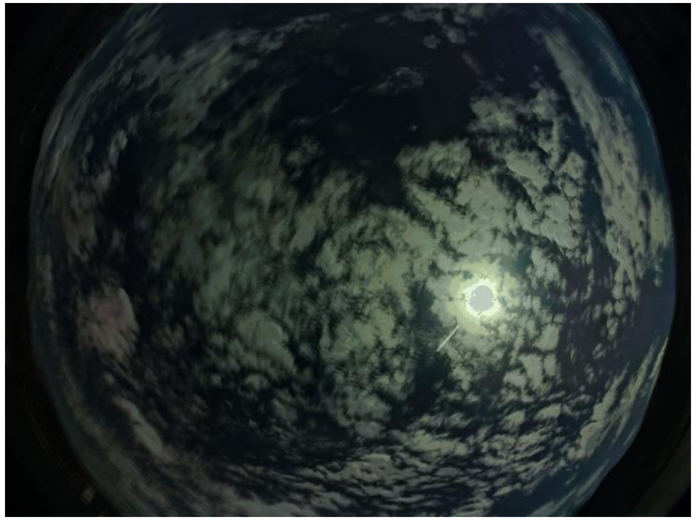
Class Cloudy

b
UNIVERSITÄT
BERN

- Sky has clouds but still shows clear patches. Sunlight sometimes gets through but is partially obstructed.
- Mixture of sky and clouds - clouds near the sun
- 30 – 70% cloud coverage

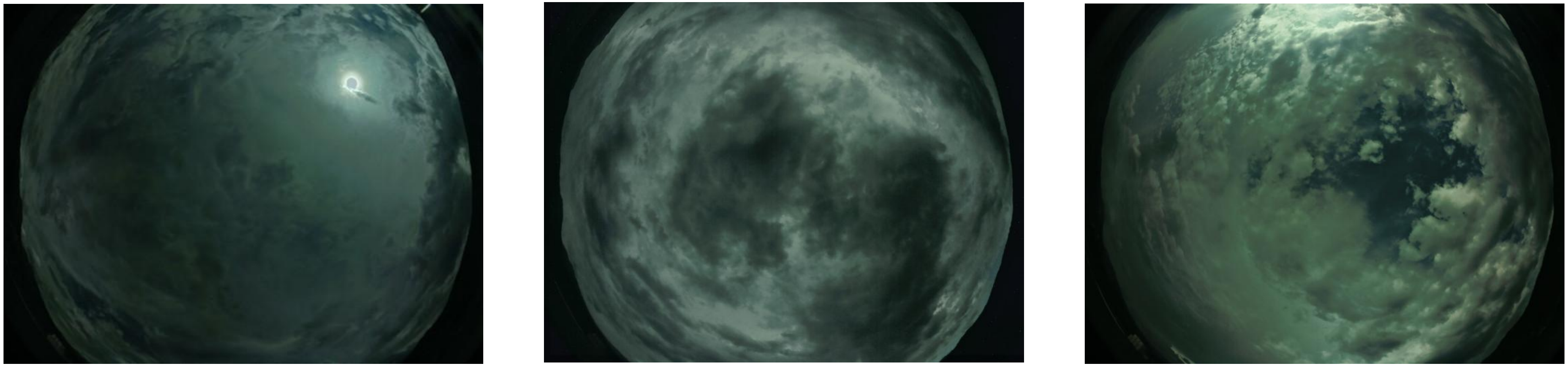






Class Overcast

- The sun is fully covered by clouds with no visible breaks around. Cloud layer blocks direct sunlight.
- No direct sunlight
- >70% cloud coverage



Definitions

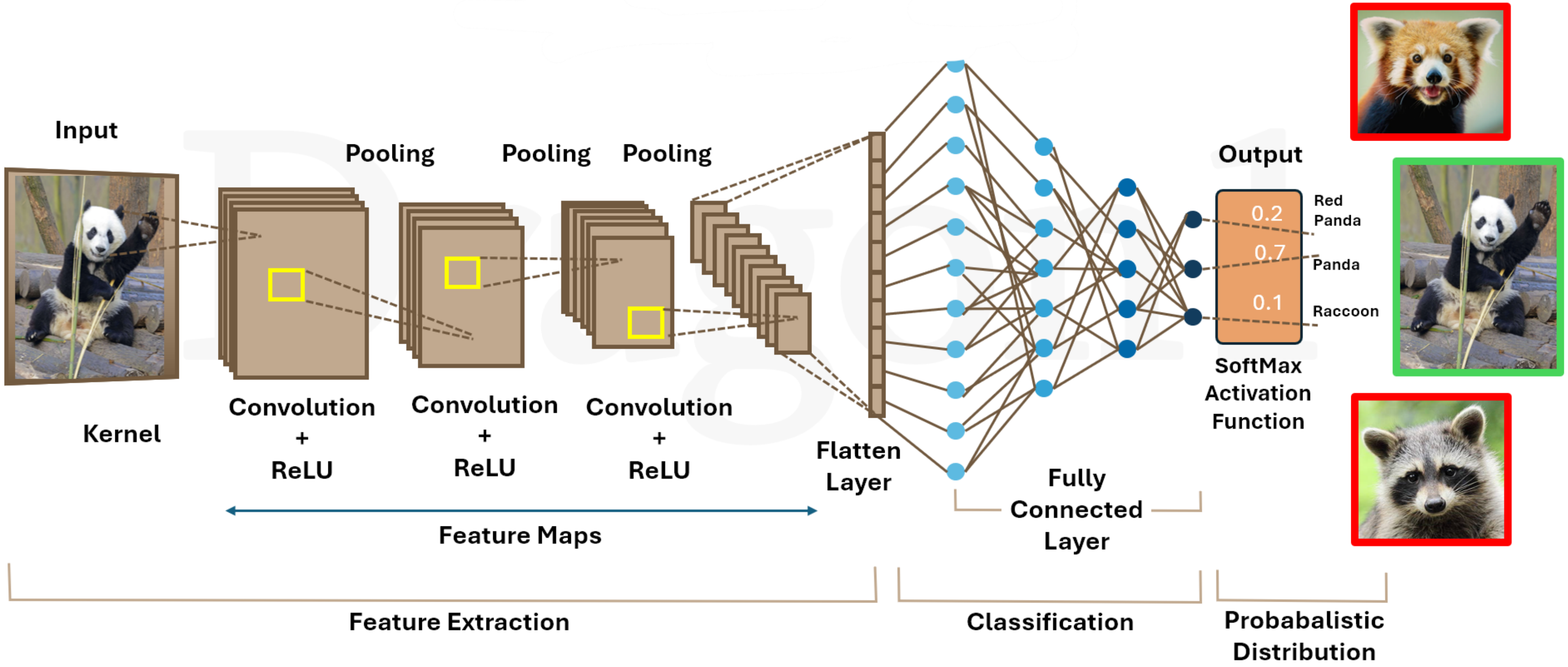
Definitions

UNIVERSITÄT BERN

- CNN
- Transfer Learning
- Fine Tuning
- Literature Research → Model

Convolutional Neural Network (CNN)

- A CNN learns by repeatedly predicting on images, measuring how wrong it is, and adjusting its filters through backpropagation until the predictions become accurate.”

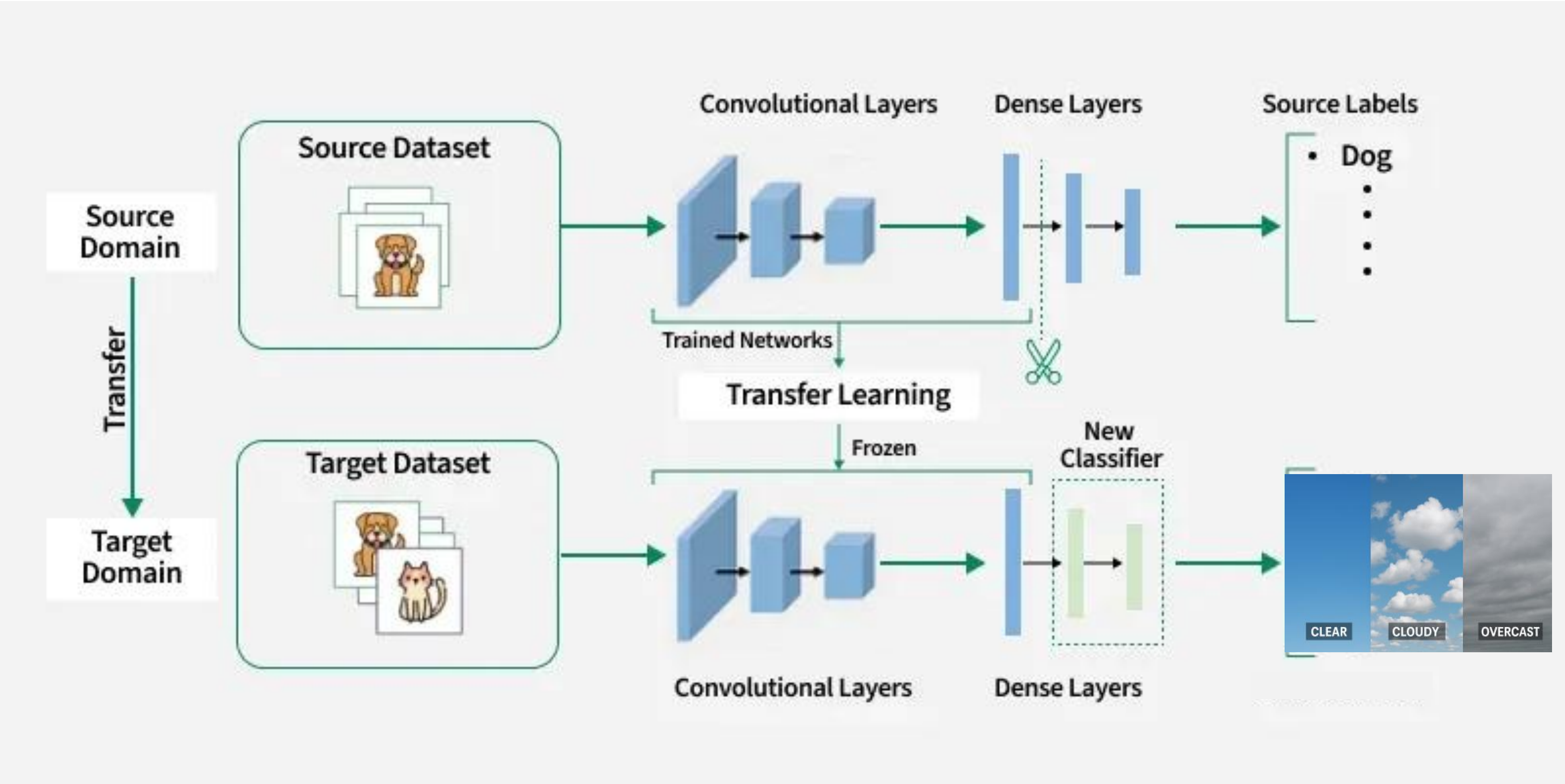


Transfer Learning

- Start with CNN trained on a large dataset

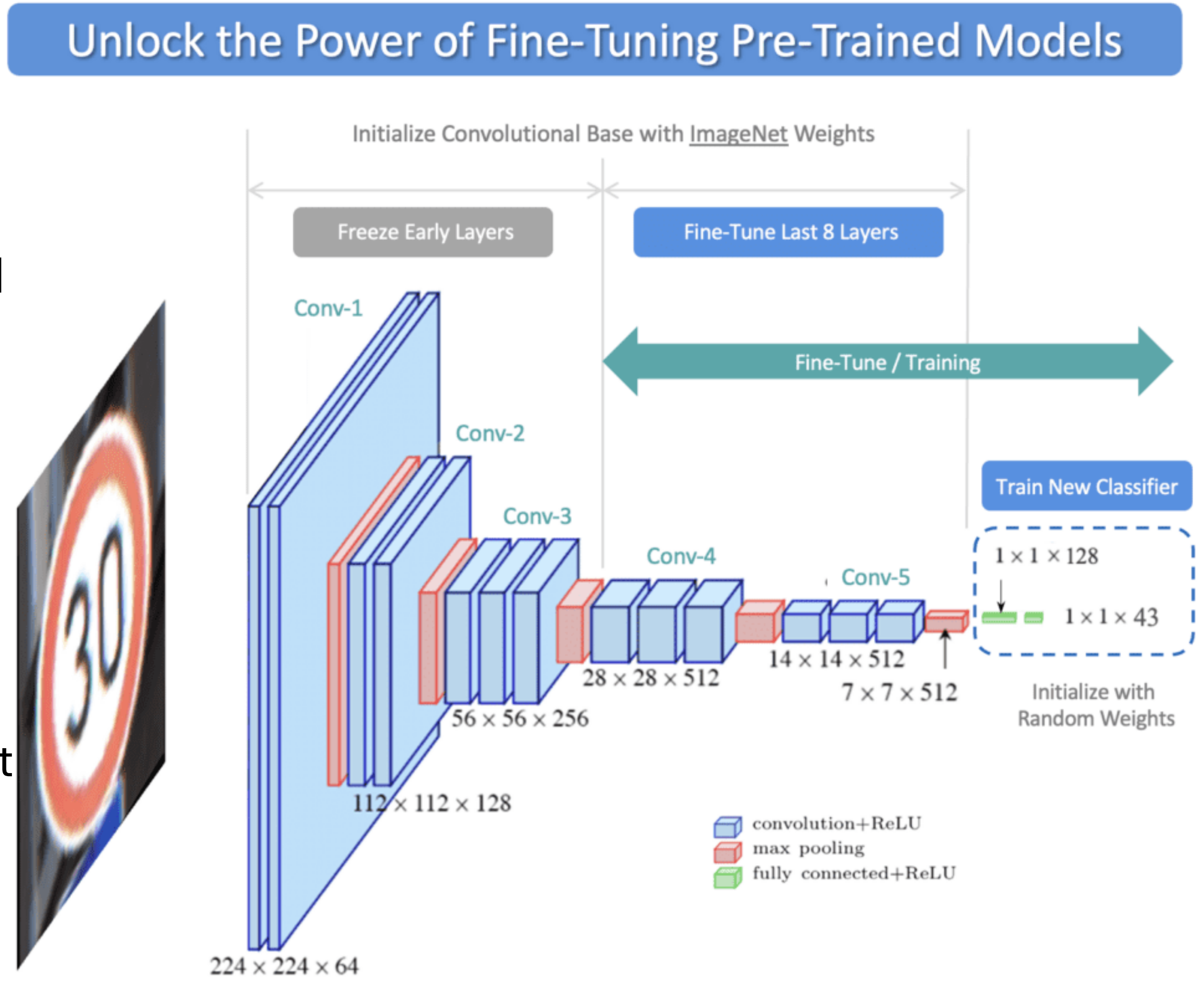
(e.g., ImageNet with 14M images and 1000 classes)

- Already knows how to detect edges, shapes, patterns like sky, tree, objects
- Cut last classification layers, keep rest **frozen**.
- add a small **new classifier** for our own categories (clear / cloudy / overcast).



Fine Tuning

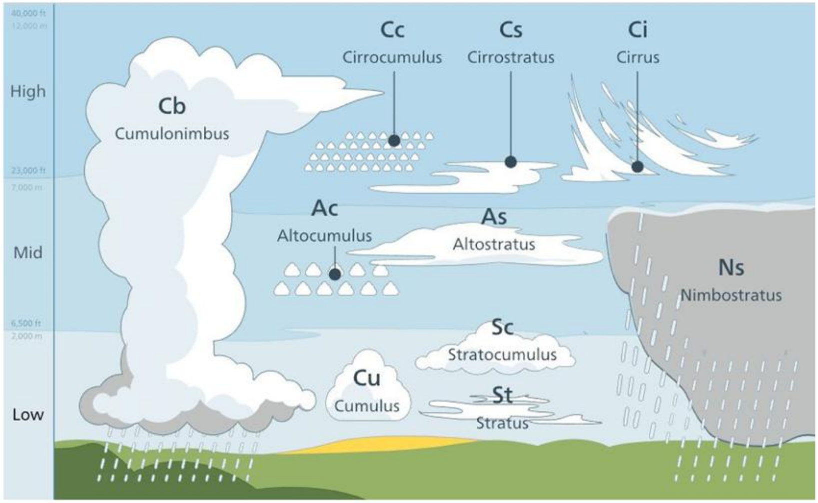
- Step 1 — Start with a Pre-Trained Model (ImageNet)
- Step 2 — Freeze the Early Layers → Keep low level features
- Step 3 — Unfreeze top M layers → to make it more task-specific
  - Layers adapt to the new dataset of clouds
- Step 4 — Replace the Classifier and train



https://learnopencv.com/fine-tuning-pre-trained-models-tensorflow-keras/

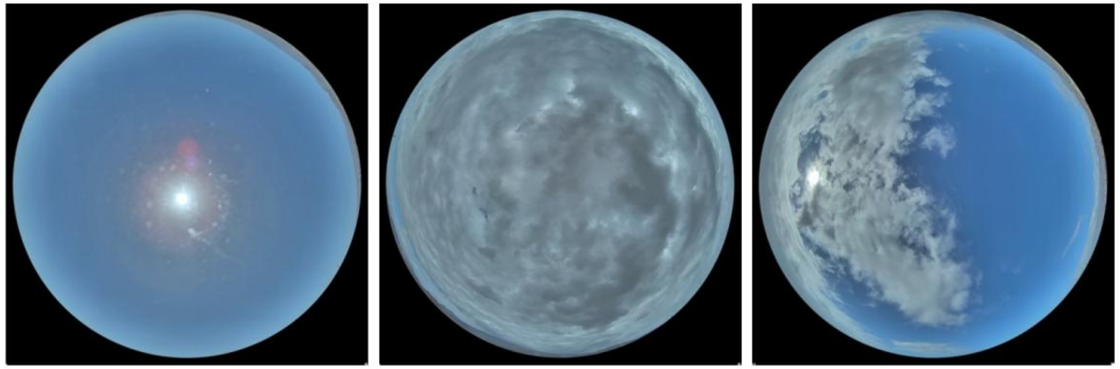
Literature Research

- Cloud Type Classification – Guzel et al. (2024)
  - 11 cloud types
  - Seven pretrained CNNs compared
  - Public Dataset: Cirrus Cumulus Stratus Nimbus
  - **Xception performed best with 97.66% accuracy**



- Weather Image Classification – Naufal et al. (2021)
  - Out of 4 CNN architectures tested, **Xception performed best with 90.21% accuracy.**

Literature Research



(a) *Clear sky* class. (b) *Cloudy sky* class. (c) *Partly cloudy sky* class.

- Sky Image Classification – Hernández-López et al. (2024)
  - Clear / Partly Cloudy / Cloudy
- Tested EfficientNet V2 (B1, B2) and ResNet models
- Best performance: **EfficientNetV2-B1/B2 with 98.09% accuracy.**

Conclusion from Literature Research

- Initially planned to compare multiple architectures, but this exceeded the scope of the project.
  - Selected **Xception**.
- Research Question

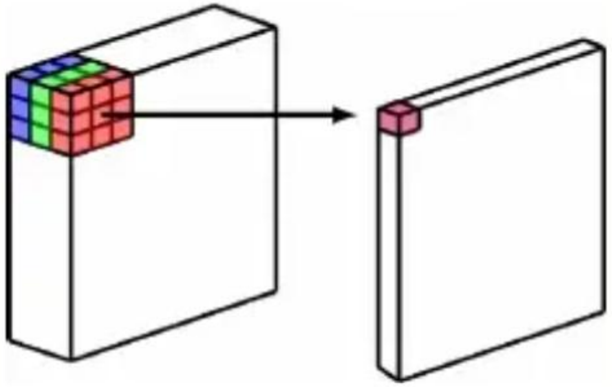
  *How effectively can a CNN based on the Xception architecture classify images into **clear, cloudy and overcast** using transfer learning?*
- Hypothesis

  Given that Xception achieved **97.66% accuracy** in cloud-type classification, it is expected to perform equally well — or better — on the simpler task of cloud-cover classification.

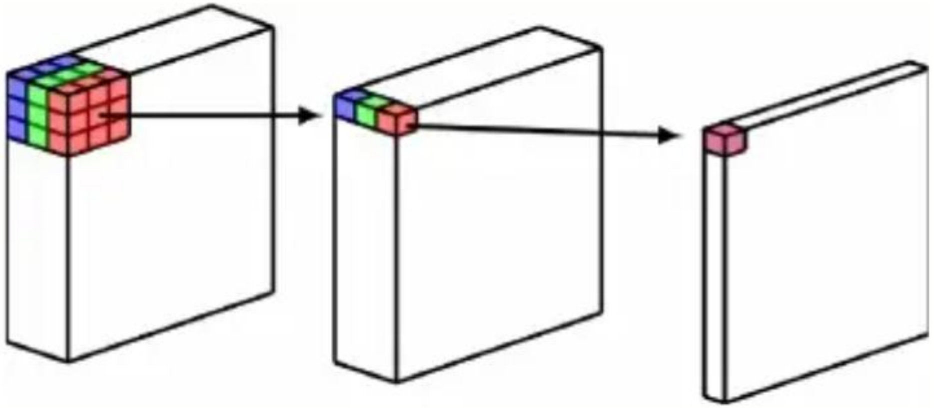
Base Model Xception

Xception

- Developed by François Chollet
  - 2017 at Google
- Uses **Depthwise Separable Convolutions**
  - First: each colour channel filtered separately (fine details)
  - Then: pointwise 1×1 convolution combines information
  - Computationally less expensive
- Pre-trained on ImageNet → strong transfer learning



(a) Conventional Convolutional Neural Network



Depthwise Convolu-
tion

Pointwise Convolution

(b) Depthwise Separable Convolutional Neural Network

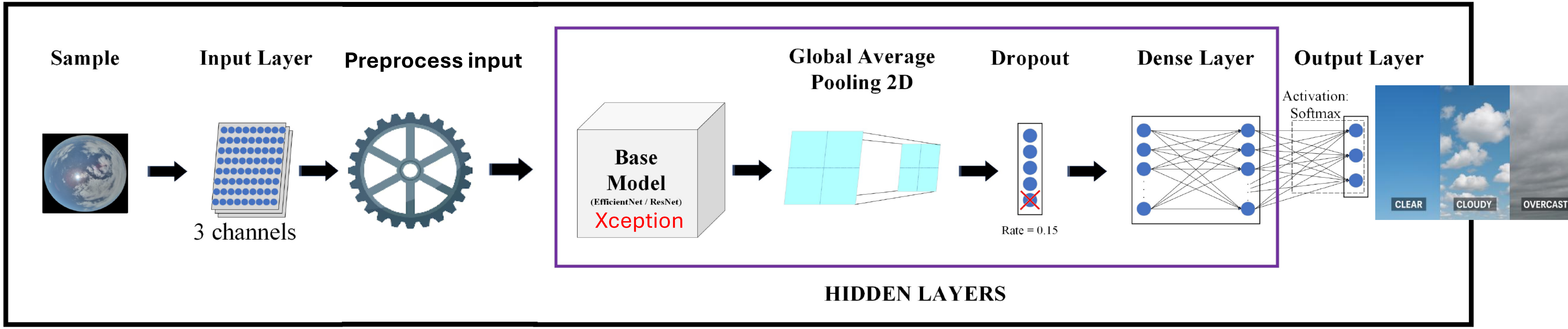
https://doi.org/10.1016/j.compag.2019.104948

Implementation Architecture


```
from tensorflow.keras.applications
import Xception


base = Xception(
   include_top = False,
   weights = "imagenet",
   input_shape=IMG_SIZE + (3,)
)
base.trainable = False
```

```
inputs = keras.Input(shape = IMG_SIZE + (3,))
x = preprocess_input(x)
x = base(x)
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dropout(0.2)(x)
outputs = layers.Dense(len(CLASS_NAMES),
                 activation="softmax")(x)
```



Hernández-López, et al. Sky Image Classification Based on Transfer Learning Approaches. *Sensors* **2024**, *24*, 3726

Training and Results

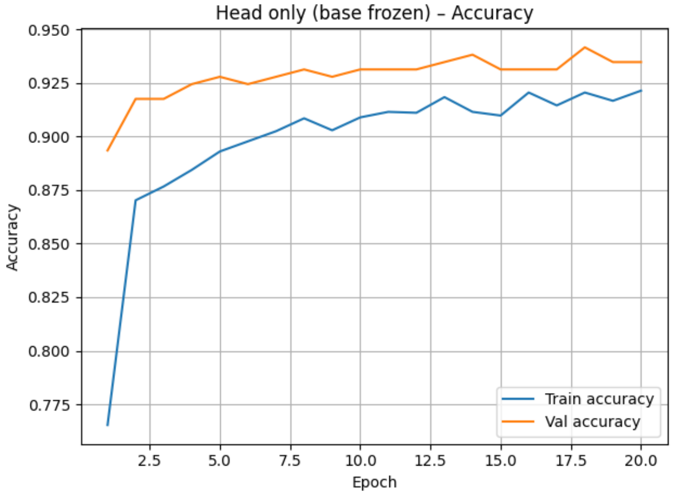
Training Setup

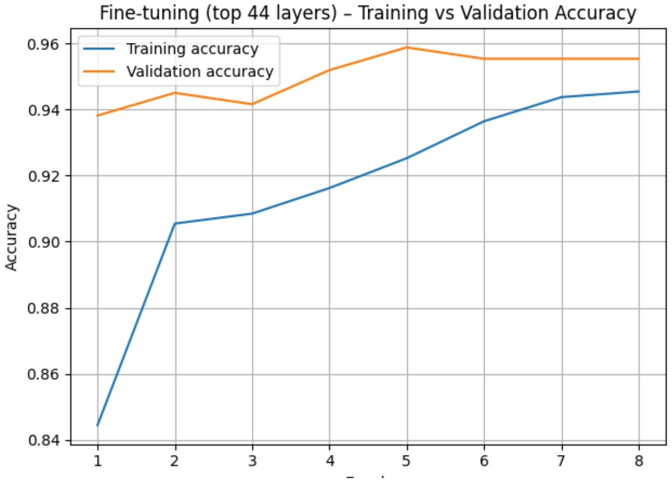
UNIVERSITÄT BERN

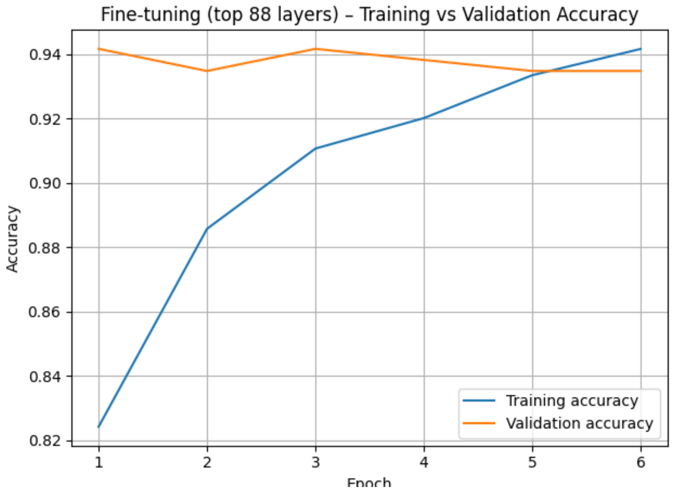
- 3 classes: **clear, cloudy, overcast**
- Split: 80% train / 10% validation / 10% test over 3000 labelled images.
- Metrics used:
  - **Accuracy** – how many images are correctly classified.
  - **Loss** – Sparse categorical cross entropy
  - **Precision / Recall / F1 (macro)** – average over the three classes, so each class counts equally.
- Two stages (20 Epochs):
  - **Stage 1:** train only the new head (Xception frozen)
  - **Stage 2: fine-tuning** with different numbers of unfrozen layers (44, 88, 132)

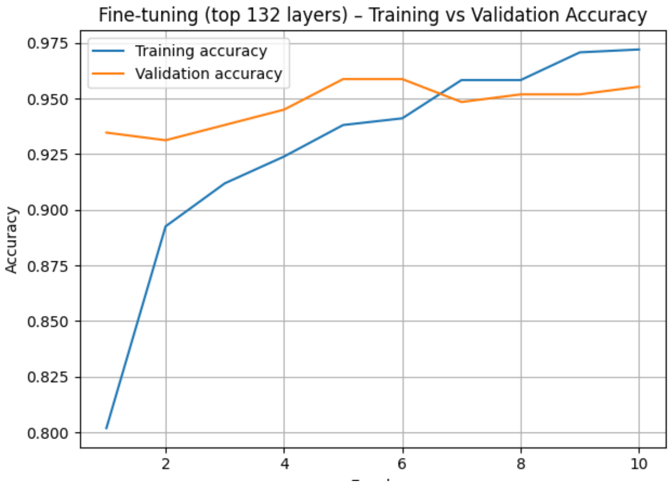
Accuracy

- Head-only training already gives strong performance
- Early stopping triggers quickly during fine-tuning
- In FT-132 and FT-88, training accuracy grows much faster than validation accuracy and even surpasses it → Overfitting
- Validation accuracy staying stable while training accuracy rises









Training set metrics

UNIVERSITÄT BERN

Train set	Accuracy	Loss	Precision	Recall	F1
Base model	0.9208	0.2117	0.9219	0.9206	0.9210
FT - 44	0.9432	0.1500	0.9436	0.9437	0.9436
FT - 88	0.9367	0.1667	0.9366	0.9370	0.9366
FT - 132	0.9712	0.0891	0.9710	0.9717	0.9712

FT – 44 = Xception Fine-tuned with 44 layers unfrozen

Validation set metrics

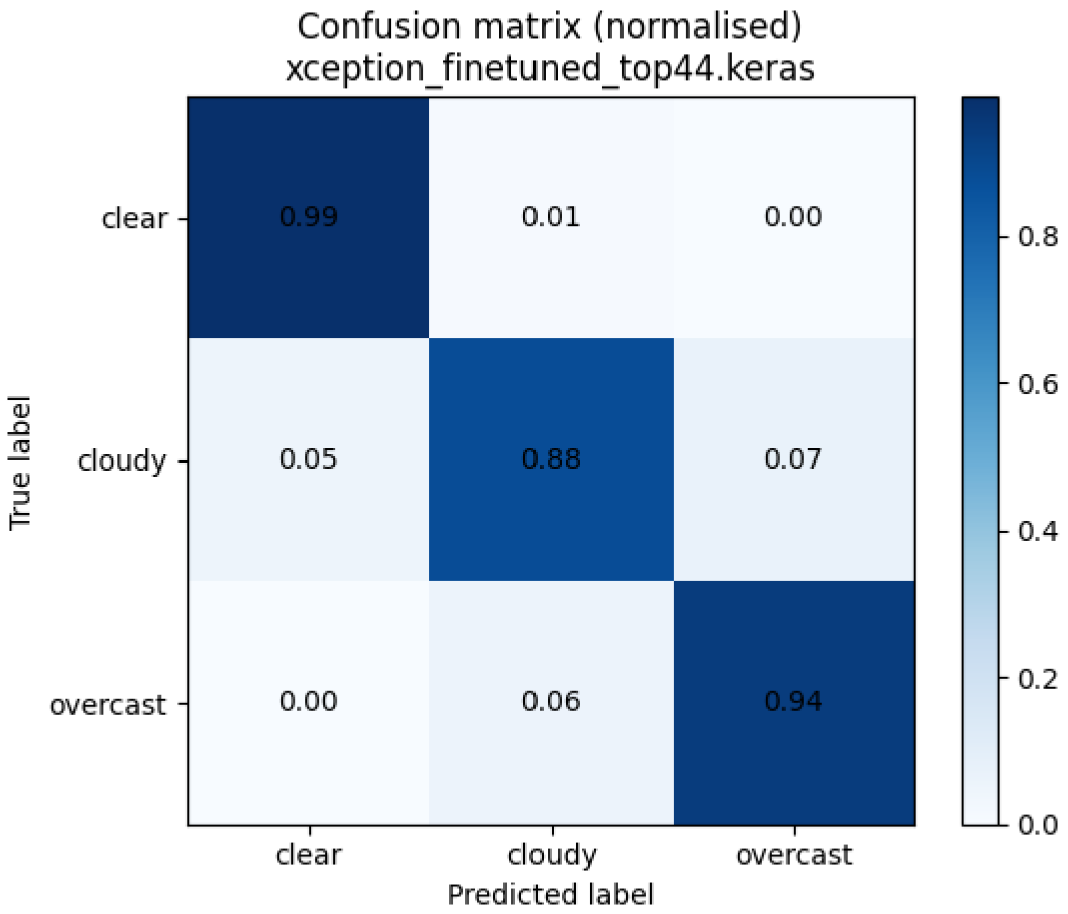
UNIVERSITÄT BERN

Val set	Accuracy	Loss	Precision	Recall	F1
Base model	0.9367	0.1650	0.9416	0.9347	0.9367
FT - 44	0.9588	0.1358	0.9626	0.9586	0.9599
FT - 88	0.9416	0.1792	0.9429	0.9437	0.9428
FT - 132	0.9450	0.1348	0.9484	0.9450	0.9462

Test set metrics

Test set	Accuracy	Loss	Precision	Recall	F1
Base model	0.9212	0.1981	0.9205	0.9203	0.9082
FT - 44	0.9384	0.1727	0.9374	0.9382	0.9374
FT - 88	0.9110	0.2386	0.9128	0.9128	0.9083
FT - 132	0.9315	0.1545	0.9303	0.9301	0.9230

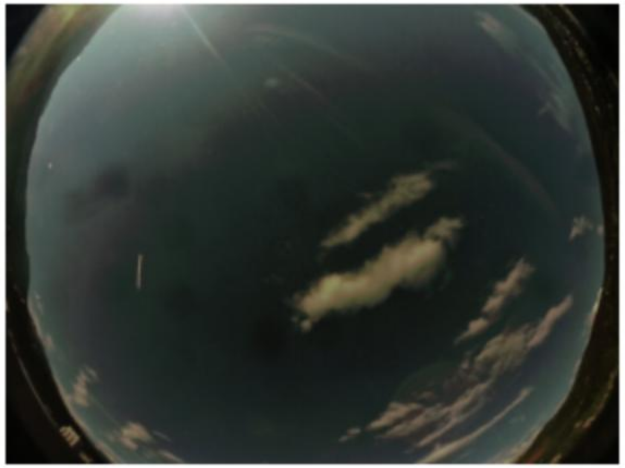
Test set metrics



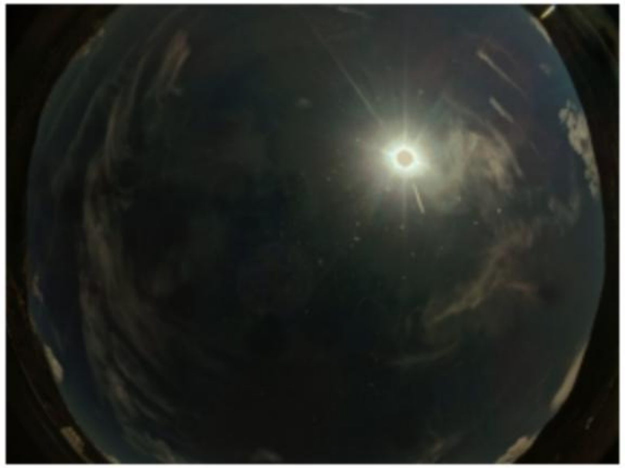
Some misclassified images

UNIVERSITÄT
BERN

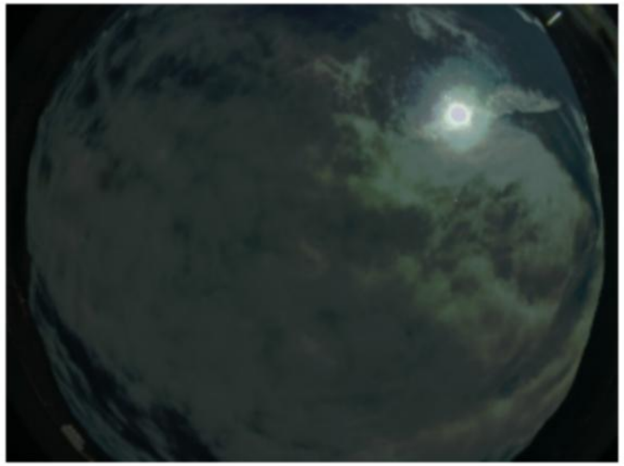
True: clear
Pred: cloudy



True: cloudy
Pred: clear



True: cloudy
Pred: overcast



Analysis of Results

- Strong classifier with all four model variants achieving high values in all metrics.
  - 91 – 97% train accuracy
  - 93 – 96% validation accuracy
  - 91 – 94 % test accuracy
  - Precision/Recall/F1 between 90% and 97%

✓The model is consistently above 90% on unseen data.

✓It correctly classifies clear / cloudy / overcast images with high reliability.

Conclusion

UNIVERSITÄT BERN

- Research Question

  *How effectively can a CNN based on the Xception architecture classify images into **clear, cloudy and overcast** using transfer learning?*

- Hypothesis

  Given that Xception achieved 97.66% accuracy in cloud-type classification, it is expected to perform equally well — or better — on the simpler task of cloud-cover classification.

  → **Hypothesis Outcome:**
  Although the model did **not reach the expected 97.66%**, it still achieved a **high accuracy of 93.86%**, confirming that Xception is highly effective for cloud-cover classification.

Conclusion

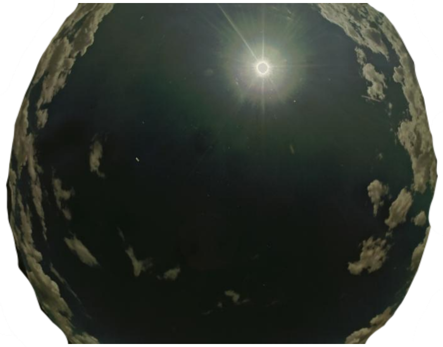
- Fine-tuning improves performance
  - **FT-44** achieves **best** result with **93.84% accuracy.**
  - **FT-88 & FT-132** show signs of **overfitting**

→ Fine-tuning increases performance by a few percentage points (92% in base model), but this comes at the cost of **higher computational load**, **longer training times**, and the risk of **overfitting**. Whether the additional accuracy is worth the extra complexity depends entirely on the application requirements.

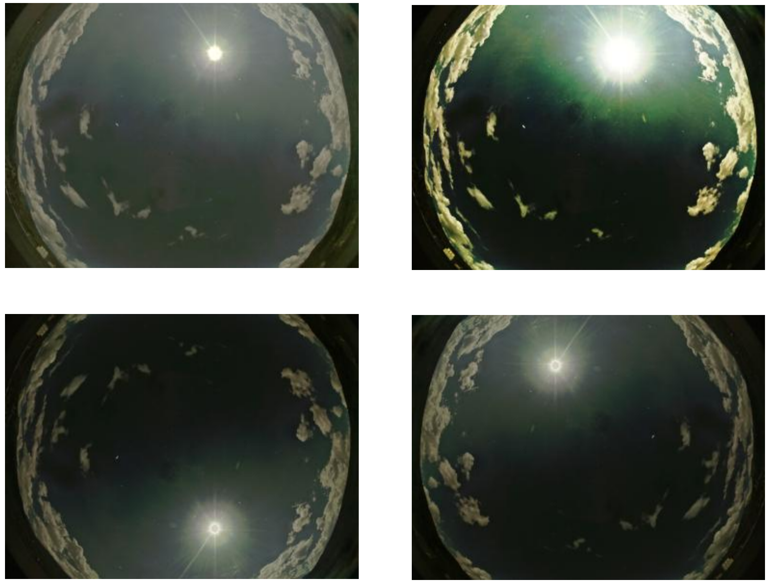
Future Work

Possible Data Preparation Steps

- **Apply circular mask and resize to 299 x 299**
  - Remove black rim
  - Cut away objects at the horizon
  - Squeeze it to 299 x 299



- **Augmentation**
  - Rotation
  - Color shifts
  - Brightness adjustments



Future Research

UNIVERSITÄT BERN

- Cloudy sky → cloud type classification
- Xception very heavyweight – try base model with less parameters. Code is built very modularly, so can be easily replaced. E.g. Efficientnet V2

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
EfficientNetV2B2	42	80.5%	95.1%	10.2M	-	-	-
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4

- CNN + other model for cloud movement and irradiance prediction

UNIVERSITÄT
BERN

That’s it!
Thank you for your attention!

UNIVERSITÄT BERN

Questions

Questions

- Do you think it was worth it to fine-tune?
- Do you have other ideas on how to predict solar irradiance? Hybrid models?
- What is the most realistic next step with the conclusions of this project?
- Do you have suggestions of other models for transfer learning?
- If you were to collect the dataset again and preprocess what would you change to make the model more robust?

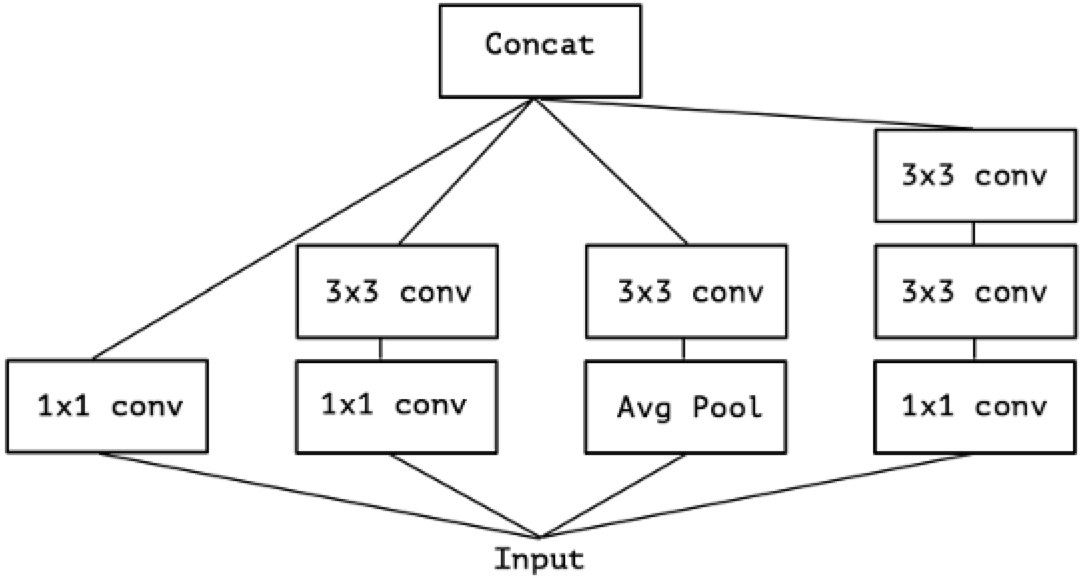
References

- F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.
- Guzel, M., Kalkan, M., Bostanci, E., Acici, K., & Asuroglu, T. (2024). Cloud type classification using deep learning with cloud images. *PeerJ. Computer science*, *10*, e1779. https://doi.org/10.7717/peerj-cs.1779
- Hernández-López, R., Travieso-González, C. M., & Ajali-Hernández, N. I. (2024). Sky Image Classification Based on Transfer Learning Approaches. *Sensors*, *24*(12), 3726. https://doi.org/10.3390/s24123726
- Kalkan M, Bostancı GE, Güzel MS, Kalkan B, Özsarı Ş, Soysal Ö, Köse G. 2022. Cloudy/clear weather classification using deep learning techniques with cloud images. Computers and Electrical Engineering 102(7804):108271 DOI 10.1016/j.compeleceng.2022.108271.
- Naufal, M. F., & Kusuma, S. F. (2022, April). Weather image classification using convolutional neural network with transfer learning. In *AIP Conference Proceedings* (Vol. 2470, No. 1, p. 050004). AIP Publishing LLC.

Evaluation Metrics

Metric	Definition
Accuracy	Percentage of all predictions that are correct.
Loss	Measures how wrong the model is during training (lower is better).
Precision	Of all images predicted as a class, how many are actually correct.
Recall	Of all real images of a class, how many the model correctly finds.
F1-Score	Harmonic mean of precision and recall; high only if both are high.

132 length vs 81 depth



Why SVM for Raindrop

b
UNIVERSITÄT
BERN

- Because raindrop detection is a **small, highly imbalanced, and very visual problem.**
  An SVM works better when you have:
- **few training images**
- **only two classes**
- **high-quality feature embeddings** (like MobileNetV2's 1280-D vectors)
- a **simple linear boundary** between classes
- Softmax requires more data and tends to overfit quickly.
  SVMs, especially with pre-trained embeddings, are **very strong for small binary problems**.

My journey

UNIVERSITÄT BERN

- Problem description
  - Why do we need it
- Preprocessing
- Transfer Learning
  - Xception architecture (warum Xception?)
- My CNN
  - First time 87%
  - Extract wrong images and see who was right me or CNN
  - # Part 1: train only the small "head" (fast, safe).#
  - Part 2: fine-tune the top layers of Xception (better accuracy).#
  - Then: evaluate (accuracy, precision, recall, F1), confusion matrix
  - Fine tuning with different layers frozen
- How to make it better
  - Preprocessing steps
- Question round

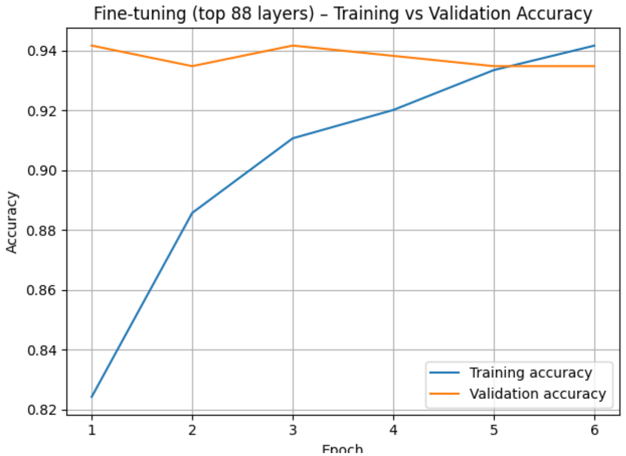
Fine Tuning – How it Works

- Freezing

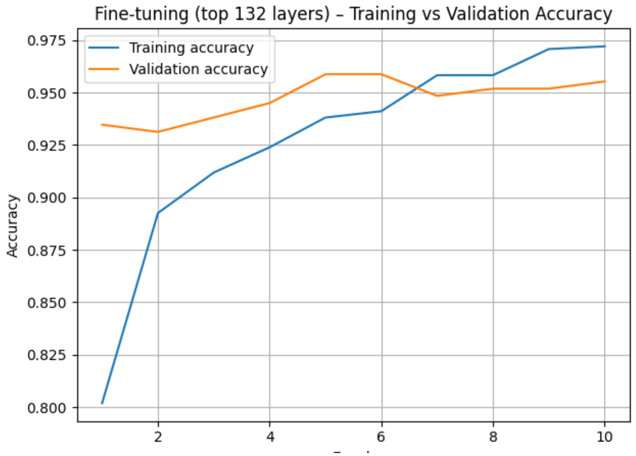
FINE TUNING

- The fine tuning process took over 1 hour
- So I split it and saved my progress in a model keras file after training the head and after fine tuning.
- Problem is fine tuning causes overfitting.
- When putting images from the new server, it might be as accuarte

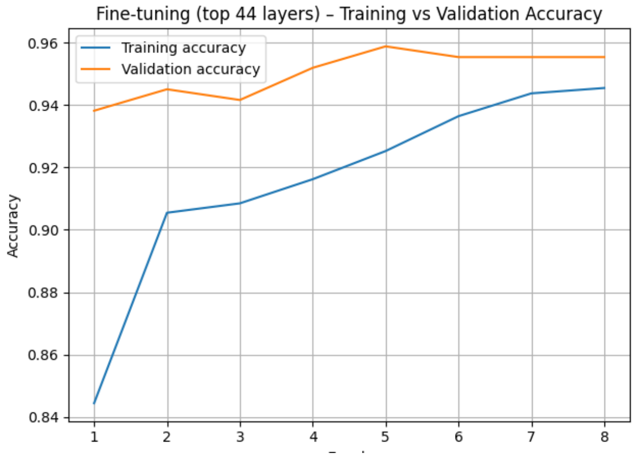
Accuracy



Fine-tuning (top 88 layers) – Training vs Validation Accuracy
Accuracy
0.94
0.92
0.90
0.88
0.86
0.84
0.82
1
2
3
4
5
6
Epoch
Training accuracy
Validation accuracy



Fine-tuning (top 132 layers) – Training vs Validation Accuracy
Training accuracy
Validation accuracy
Accuracy
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
2
4
6
8
10



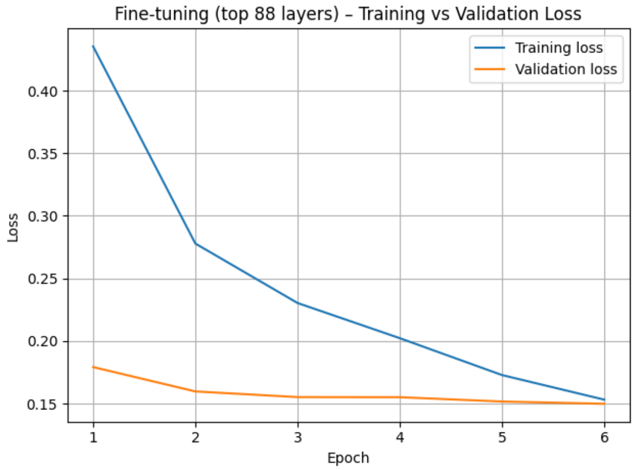
Fine-tuning (top 44 layers) – Training vs Validation Accuracy
Training accuracy
Validation accuracy
Accuracy
0.96
0.94
0.92
0.90
0.88
0.86
0.84
1
2
3
4
5
6
7
8

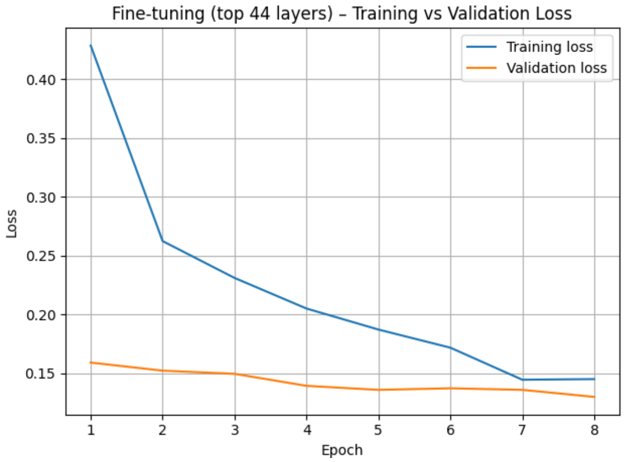
ETH Study

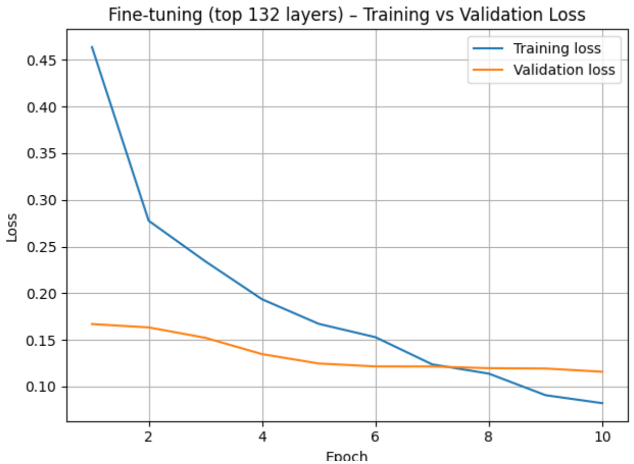
UNIVERSITÄT BERN

- https://www.research-collection.ethz.ch/server/api/core/bitstreams/47bd7a0d-5c7f-41b6-8752-7bceb3407f35/content

Loss


Fine-tuning (top 88 layers) – Training vs Validation Loss
Training loss
Validation loss
0.40
0.35
0.30
0.25
0.20
0.15
Loss
1
2
3
4
5
6
Epoch


Fine-tuning (top 44 layers) – Training vs Validation Loss
Training loss
Validation loss
0.40
0.35
0.30
0.25
0.20
0.15
Loss
1
2
3
4
5
6
7
8
Epoch


Fine-tuning (top 132 layers) – Training vs Validation Loss
Training loss
Validation loss
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
Loss
2
4
6
8
10
Epoch

In a Broader Context