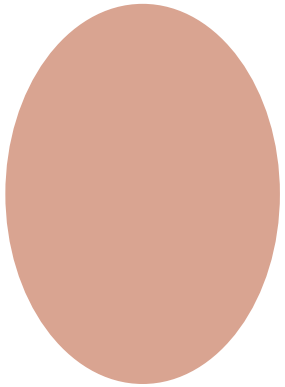
AI-based augmentation of camera data

KI-basierte Augmentierung von Kameradaten

Team Members



KEVAL

AMAN SURESH RUPARELIYA

VIKAS ANIL TIWARI

JOSHI VARDHAN LELLA

JONAS RAPP

SABRINA RIEGER

LENNART MASSARCZYK

FELIX NEUSINGER

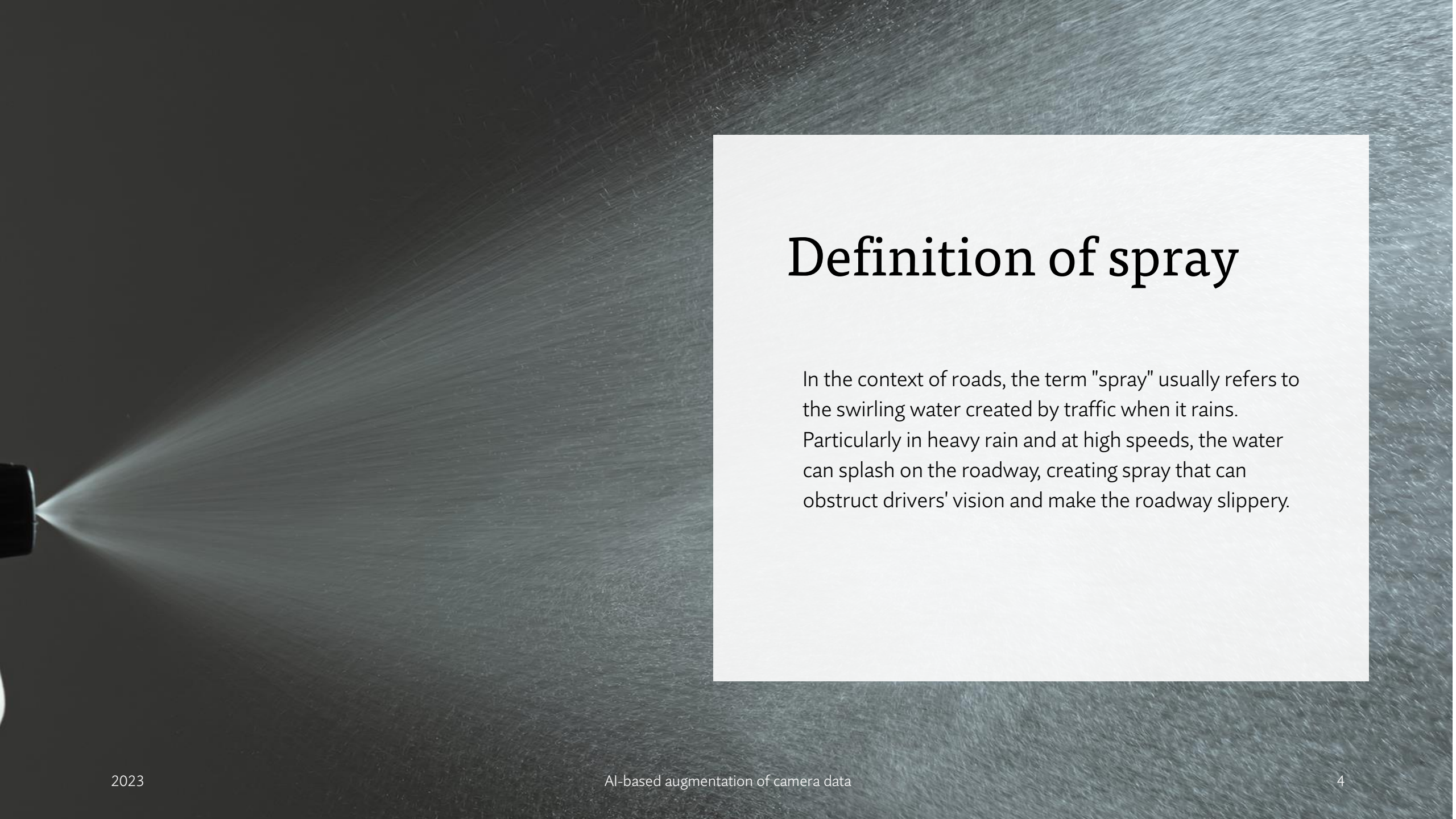
ASHOKA GOLITHADKA UDAYASHANKARA

Clear

Target

The goal of this project is to implement spray on good weather images with the help of artificial intelligence

Wet



Definition of spray

In the context of roads, the term "spray" usually refers to the swirling water created by traffic when it rains. Particularly in heavy rain and at high speeds, the water can splash on the roadway, creating spray that can obstruct drivers' vision and make the roadway slippery.

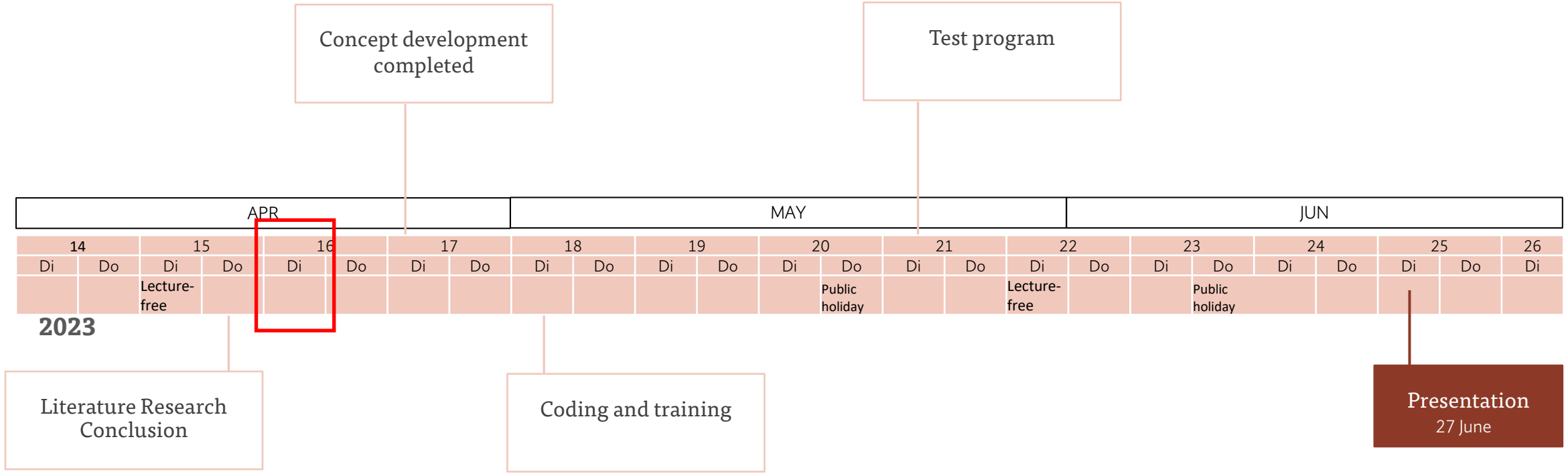
General information

Meeting with project leader and presentation of results	Tuesday 11:35 - 13:05 Room C203
Group discussion and division of work packages	Thursday 18:30 - 20:00 Discord / in person (Room tbd)

Tools

Organization	Miro	https://miro.com/app/board/uXjVMYoGMsI=/
Data exchange	GitHub	https://github.com/thi-spray-augmentation/ss23
Programming	Python	
Online Meeting	Discord	https://discord.com/channels/1087746349476360262/1087746349476360265
Input data	Moodle	https://moodle.thi.de/course/view.php?id=8930

Milestone plan



Agenda 18.04.

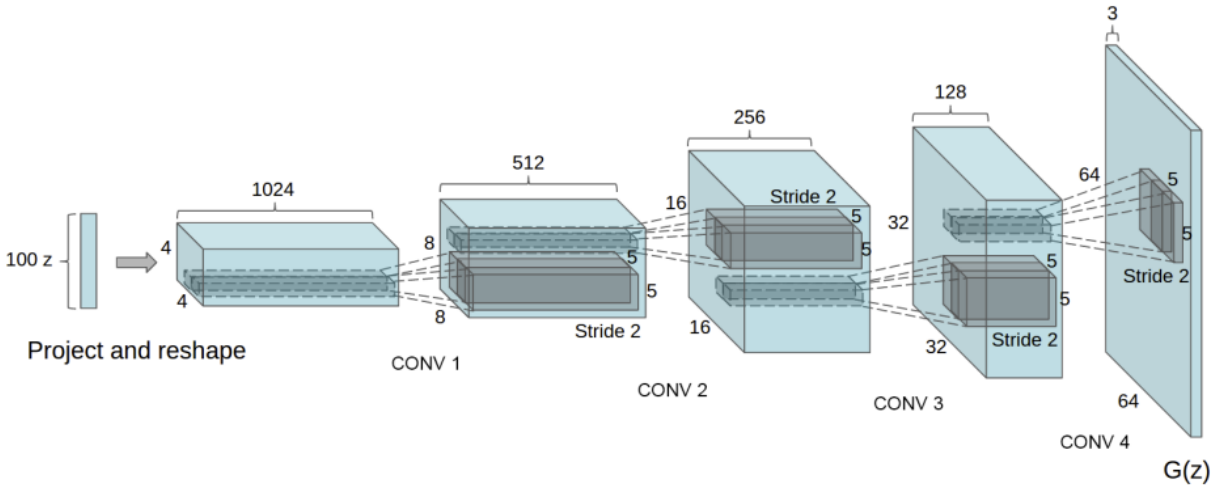
1. Present approaches
2. Questions
3. Define / decide concept
4. Define Next Meeting

Approaches

Deep Convoluti-onal GAN	Neural Style Transfer (modified VGG-19) & Day2Dusk (GAN -> UNIT)	Image processing of Segmentation masks	Python & Opencv to introduce Spray
Lennart & Felix	Jonas & Sabrina	Aman & Keval	Ashoka G U

1. Deep Convolutional GAN

A DCGAN is a direct extension of the GAN described above, except that it explicitly uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively



Problems

- Figuring out how to feed good weather pics to learned network
- A lot of different approaches with different depth of documentation on the use
- Difficult to fully understand

Advantages

- Is capable of learning nearly every weather condition
- Not really complicated to set up a simple network
- Realistic looking pictures

How much training input is needed?

- Min. 300 bad weather pictures

Own assessment

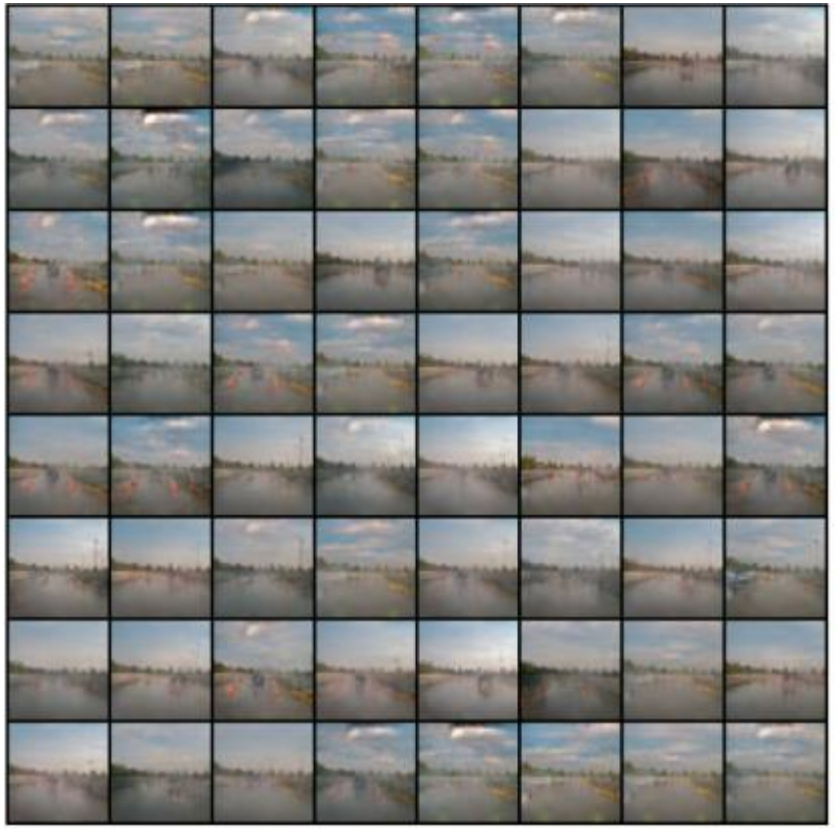
- We just scrached the surface with this approach, but dcGAN has a lot of potential

-> good to use further

1. Deep Convolutional GAN- Results

Before

After



2. Neural Style Transfer (modified VGG-19) & Day2Dusk (GAN -> UNIT)

Testing of existing DNN for image augmentation and manipulation given by MathWorks. Try to adapt these towards our needs and get a deeper insight into structure and basic settings regarding training of DNN.

Problems

- Preprocessing of data (expected input size of DNN) was partly not correct
- Extremely long training using whole dataset for training

Advantages

- Is capable of learning nearly every weather condition
- Not really complicated to set up a simple network
- Realistic looking pictures

How much training input is needed?

Could not be identified yet, but using our dataset + basic augmentation should deliver a hopefully okay result already

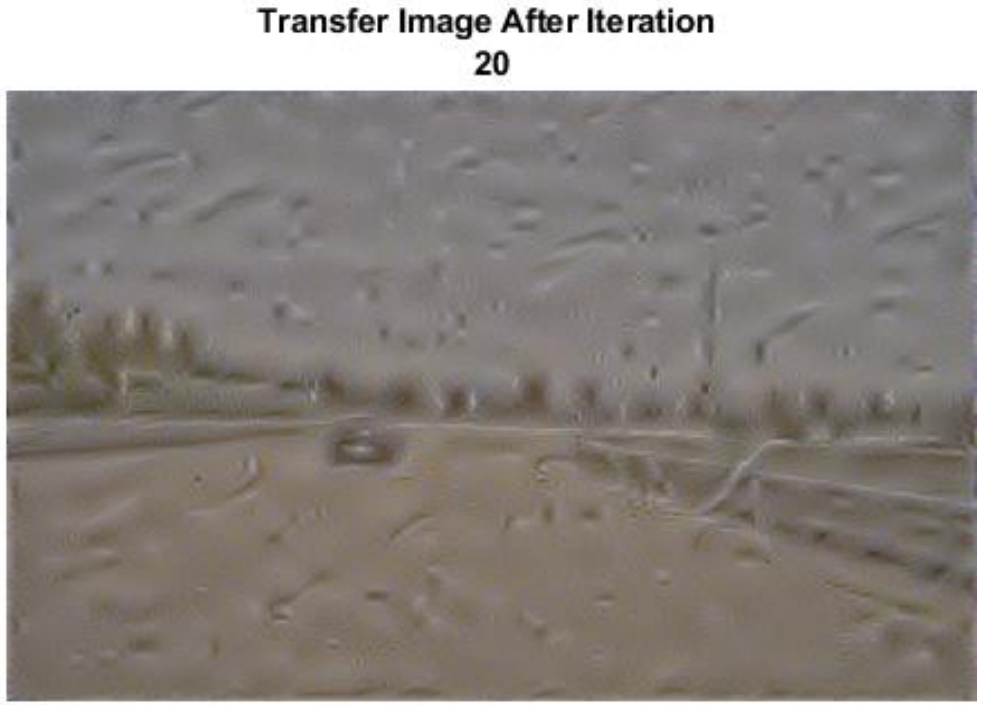
Own assessment

- If applying correctly should be able to generate the desired results

2. Neural Style Transfer (modified VGG-19) & Day2Dusk (GAN -> UNIT) - Results

Before

After



2. Neural Style Transfer (modified VGG-19) & Day2Dusk (GAN -> UNIT) - Results

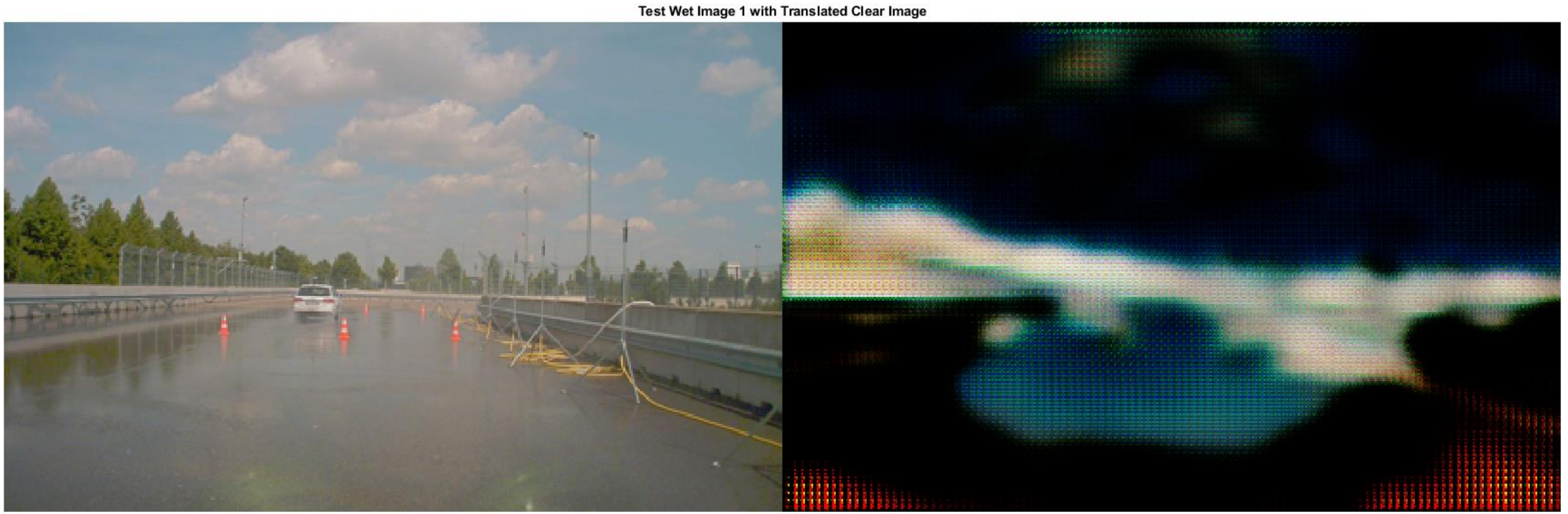
Before

After

2. Neural Style Transfer (modified VGG-19) & Day2Dusk (GAN -> UNIT) - Results

Before:

After:



3. Image processing of Segmentation masks

This approach involves creating segmentation model which creates masks of noise(spray) from the wet image dataset. Then masks of cars from the dry image dataset.
Then we apply various image blending techniques to get the desired results.

Problems

- We need to generate masks for all images of both the datasets.
- The 'Segment Anything Model (SAM)' which was released by Meta could only create masks for cars.
- The quality of the output is unpredictable at the moment.

Advantages

- This approach is comparatively less complicated to apply.
- The masks can also be used for supervised learning in GAN.

How much training input is needed?

- The images in our dataset are enough.
- But binary masks are required for all images.

Own assessment

- I am not sure about the quality of the result this approach can produce working alone. It can be paired with other approaches to boost the output.

3. Segmentation masks - Results

Dry

Wet

4. Python & Opencv to introduce Spray

OpenCV (Open Source Computer Vision) is a free, open-source computer vision and machine learning library that allows developers to perform a wide range of image and video processing tasks.
OpenCV is used in a variety of applications, such as object detection, image processing, facial recognition, and robotics. It provides a large set of pre-built functions and algorithms for image and video analysis, including basic image processing functions like filtering and edge detection, as well as advanced algorithms like feature detection and machine learning

Problems

- Detecting the exact location
- Spray pattern and Intensities
- Adding noise and blur depending on the weather condition
- Need to do trial and error

Advantages

- Not complex
- Specific
- No need of CNN

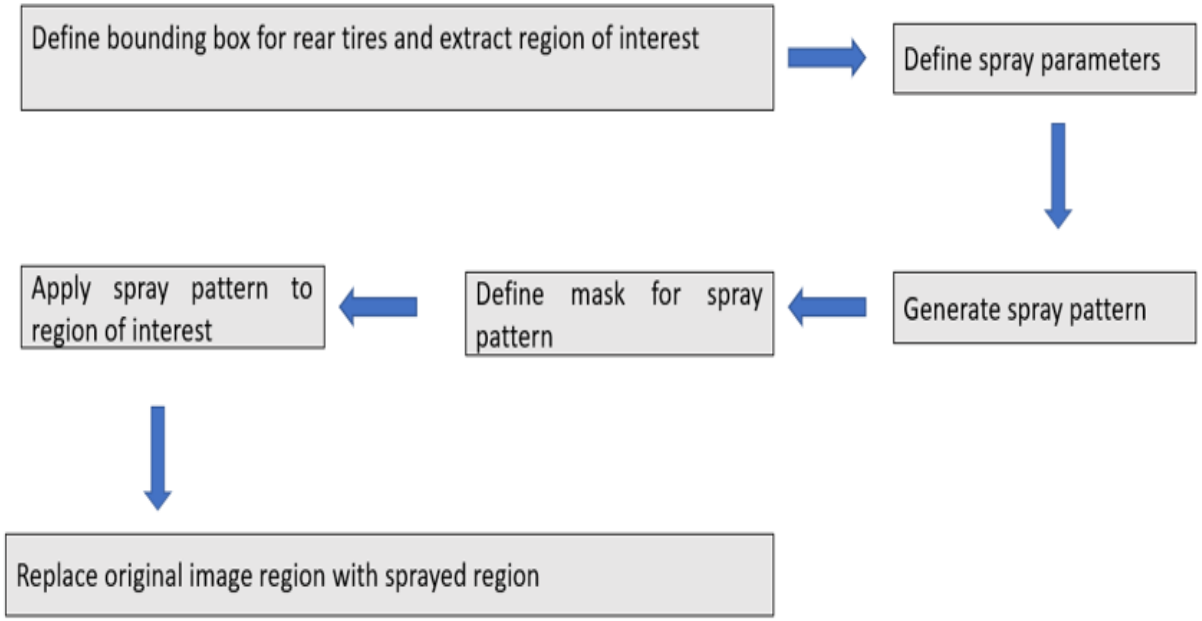
How much training input is needed?

- Not able to judge as of now, need to try many patterns and intensities

Own assessment

- To try with different spray patterns and intensities
- To change the brightness and check (Conditions)
- Can we use YOLO to detect water spray and water content present in the road and then introduce it to a dry image?

4. Python & Opencv to introduce Spray - Steps



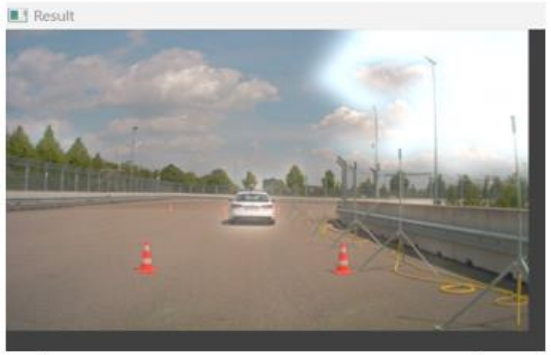
1. **Read** the image
2. **Convert** the image to grayscale
3. Apply **Gaussian blur** to the grayscale image
4. Apply **Canny edge detection** to the blurred image
5. Find **contours** in the edge detected image
6. Create a **mask** of zeros with same dimensions as original image
7. Draw **contours on the mask**
8. Apply bitwise operation on original image and mask to get only the part of original image where water spray is present.

4. Python & Opencv to introduce Spray - Steps

Trail 1

Dry road without spray

Wet road with spray



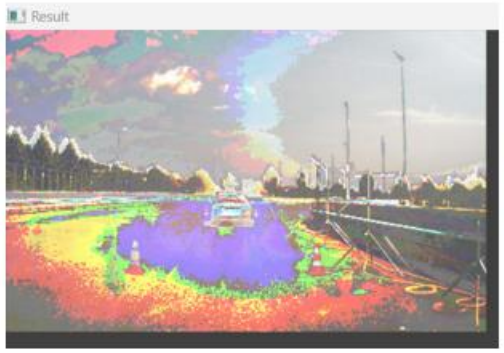
Output

4. Python & Opencv to introduce Spray - Steps

Trail 2

Dry road without spray

Wet road with spray

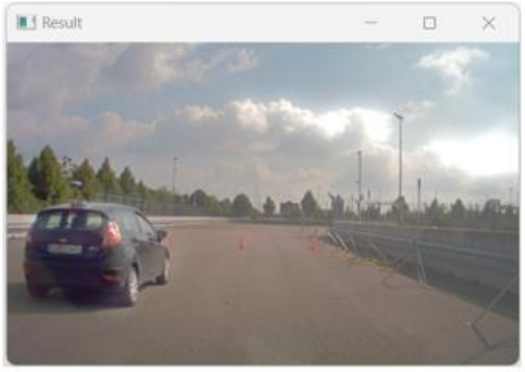


Output

4. Python & Opencv to introduce Spray - Steps

Trail 3

Dry road without spray



Output

Wet road with spray

Questions

1. Yolo – algorithm (need the tires)
2. Definition of the target?
3. Model spray? – intensive?
4. What is our data used for?
5. Day / Night also needed?