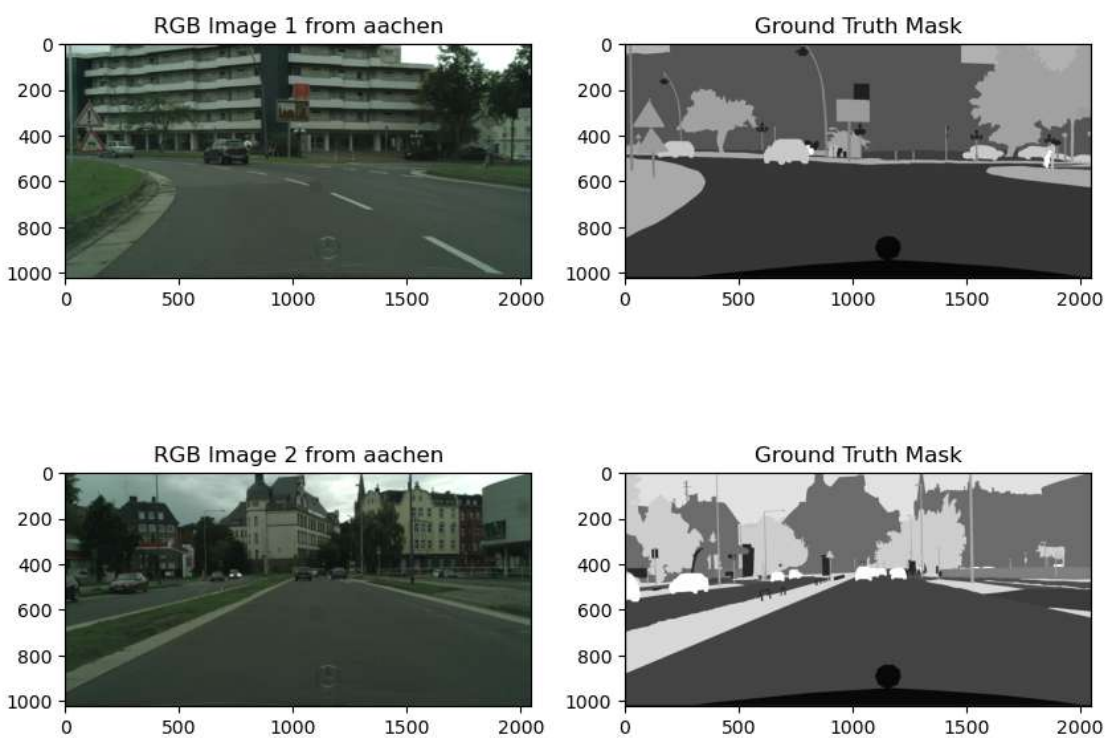
ADS_part1

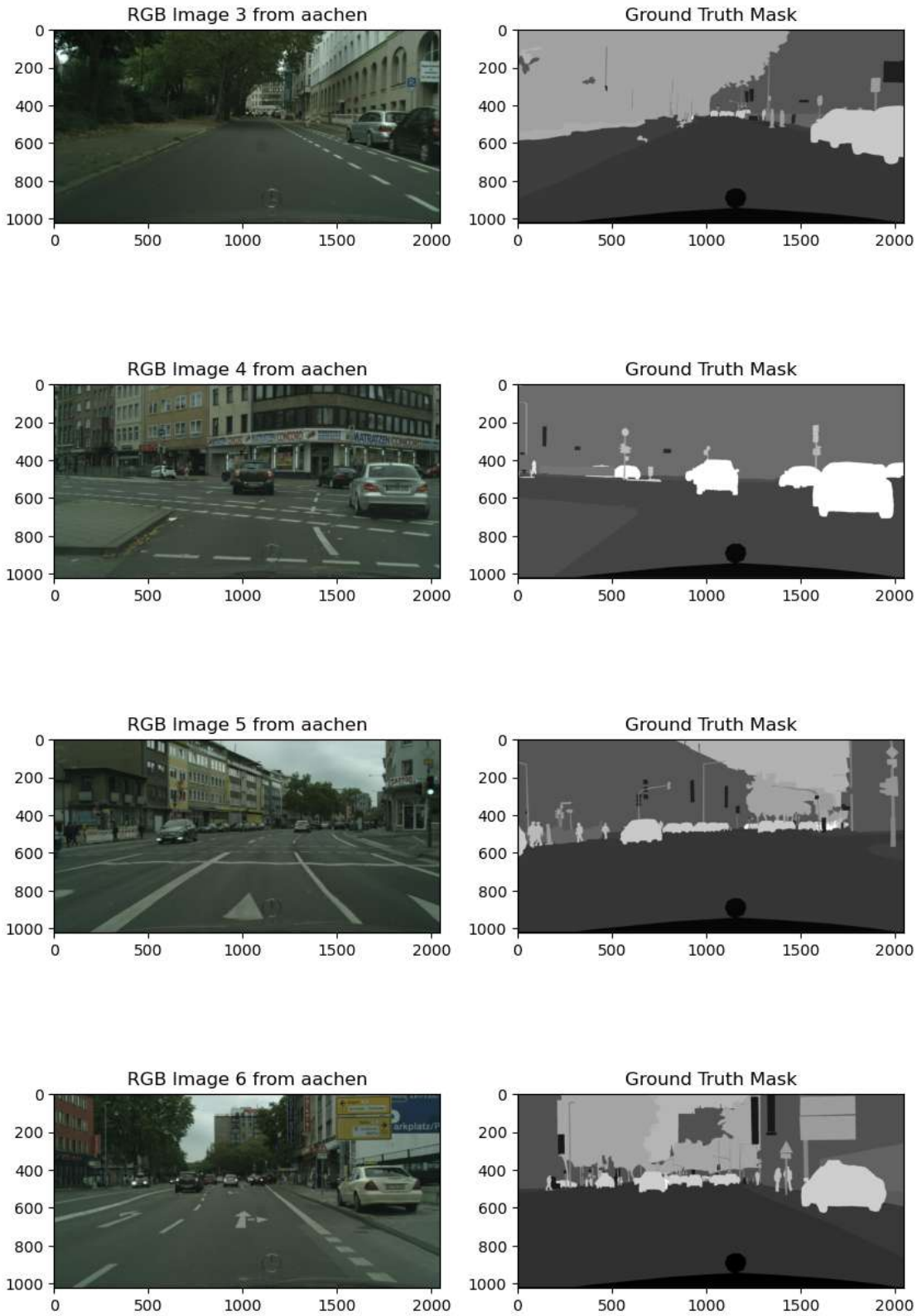
December 2, 2024

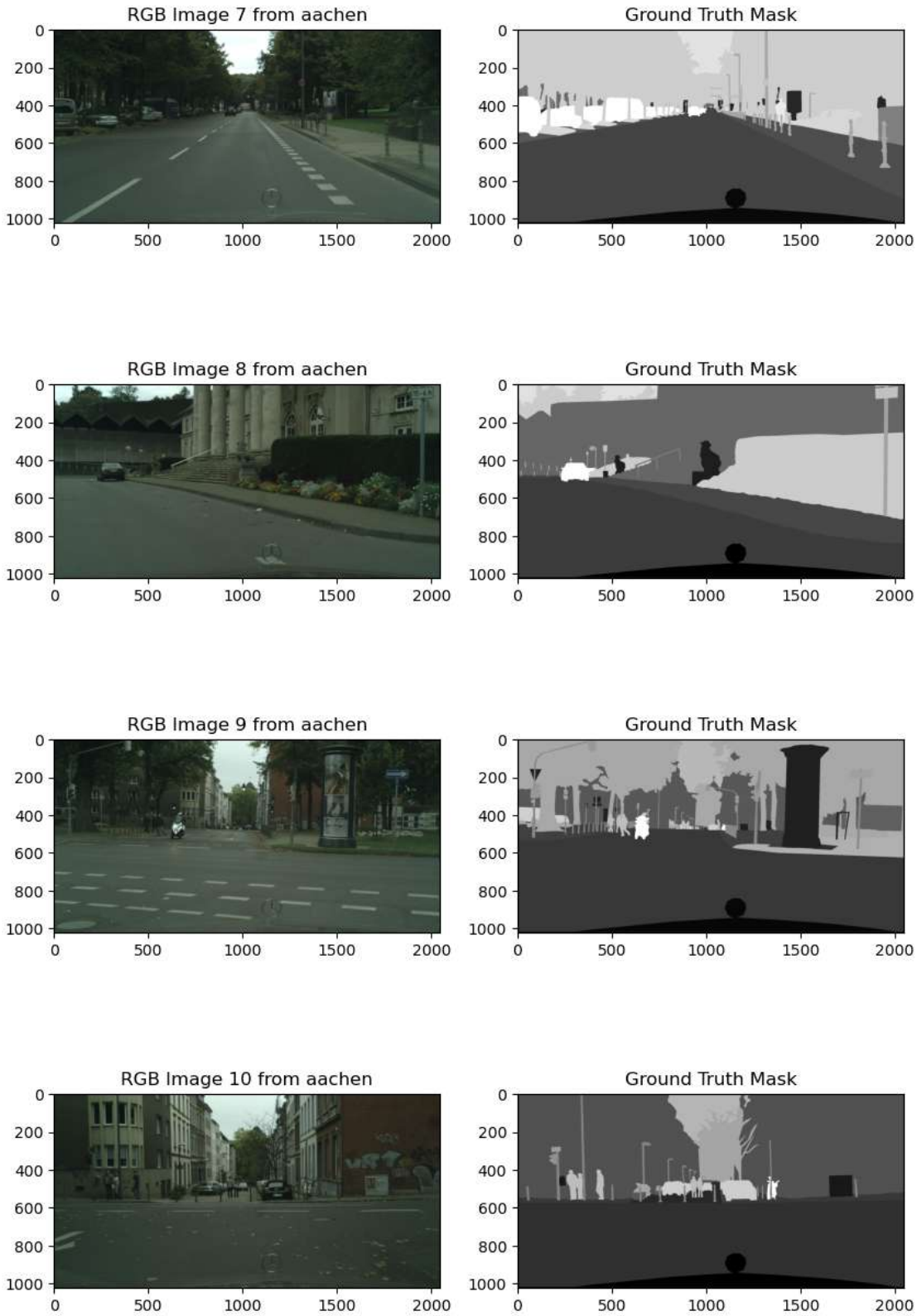
[2]:
```
import os
from PIL import Image
import matplotlib.pyplot as plt

# Define paths to the datasets
img_path = r'D:
 ↪\infosys_springboard_project\leftImg8bit_trainvaltest\leftImg8bit\train'
gt_path = r'D:\infosys_springboard_project\gtFine_trainvaltest\gtFine\train'

# Get the list of cities (assuming cityscapes-style structure)
cities = os.listdir(img_path)

# Loop through all the cities in the dataset
for city in cities:
    print(f"Processing city: {city}")

    # Get the list of image files in the city folder
    img_files = os.listdir(os.path.join(img_path, city))

    # Initialize a counter for the number of images displayed
    displayed_images = 0

    # Loop through the images and find the corresponding ground truth files
    for img_file in img_files:
        if displayed_images >= 10:  # Only display 10 images per city
            break

        # Load the RGB image
        img = Image.open(os.path.join(img_path, city, img_file))

        # Load the corresponding ground truth file (gtFine has a different␣
 ↪naming convention)
        gt_file = img_file.replace('_leftImg8bit.png', '_gtFine_labelIds.png')
        gt_path_full = os.path.join(gt_path, city, gt_file)

        # Check if the ground truth file exists before loading
        if os.path.exists(gt_path_full):
```

```
            gt = Image.open(gt_path_full)

            # Display the image and the ground truth side by side
            fig, ax = plt.subplots(1, 2, figsize=(10, 5))
            ax[0].imshow(img)
            ax[0].set_title(f'RGB Image {displayed_images + 1} from {city}')
            ax[1].imshow(gt, cmap='gray')
            ax[1].set_title('Ground Truth Mask')
            plt.show()

            displayed_images += 1  # Increment the counter after displaying an␣
↪image
        else:
            print(f"Ground truth file not found for: {img_file}")

    print(f"Finished processing city: {city}\n")
```

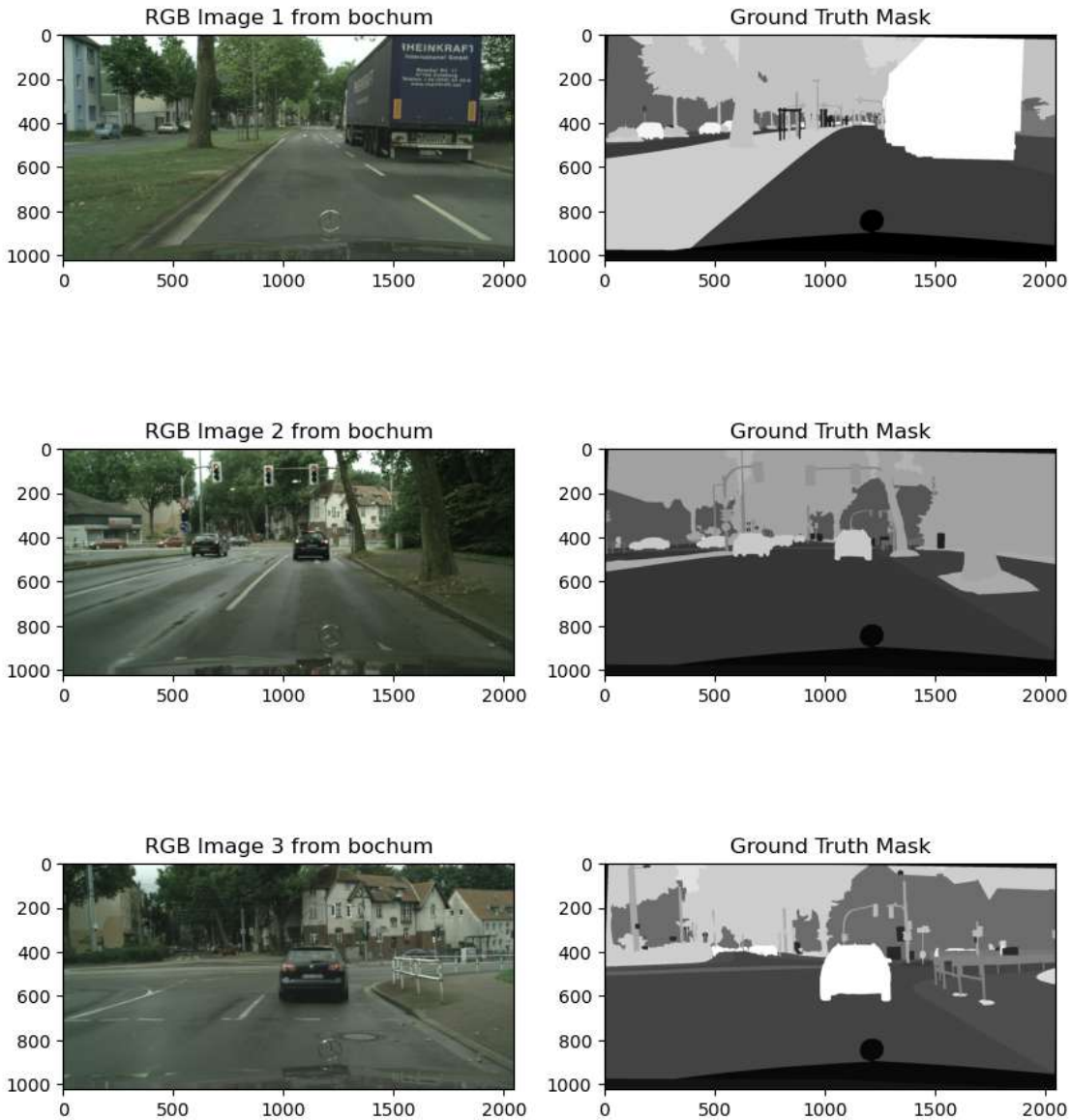
```
Processing city: aachen
```



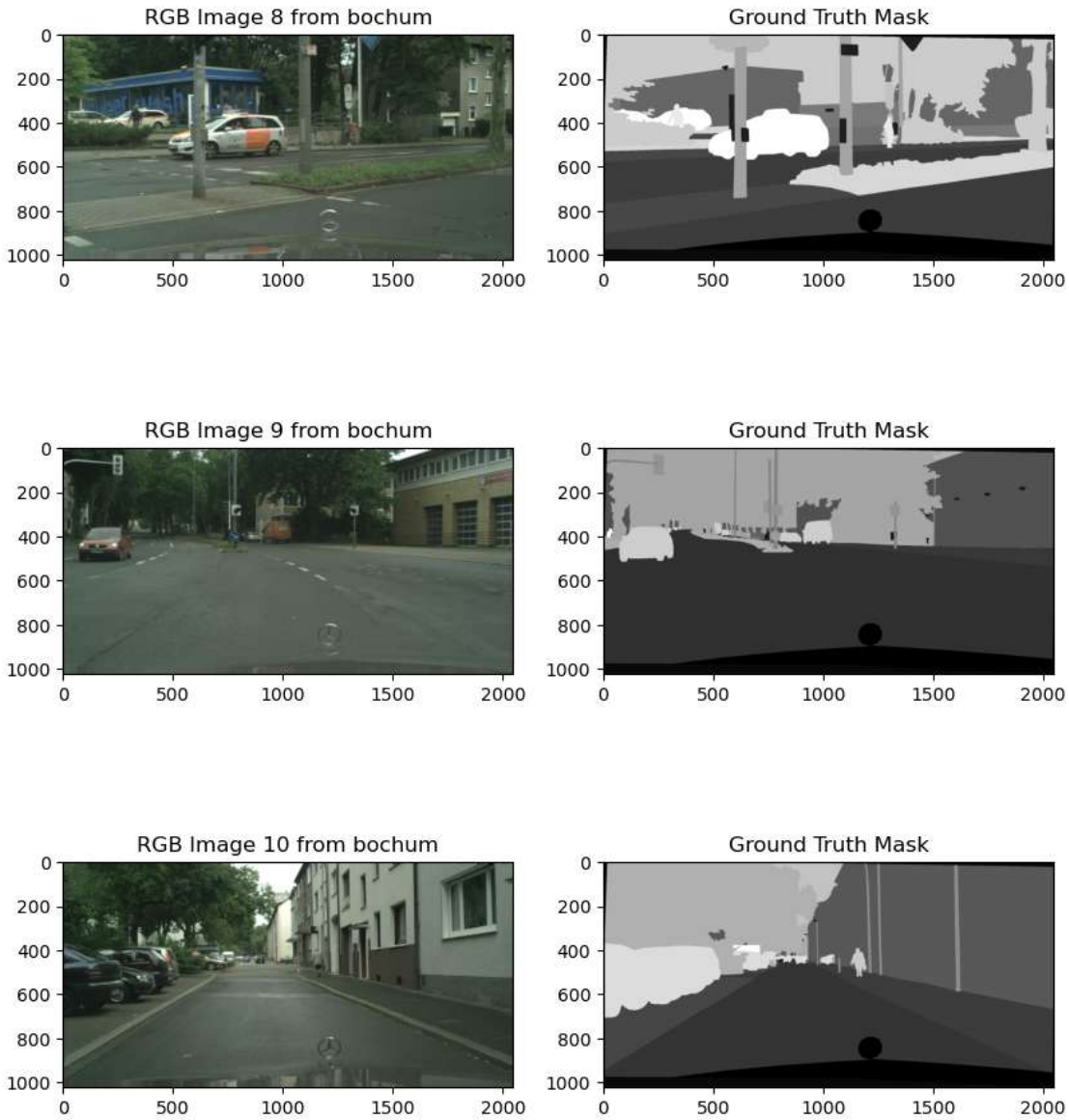

RGB Image 3 from aachen
Ground Truth Mask
RGB Image 4 from aachen
Ground Truth Mask
RGB Image 5 from aachen
Ground Truth Mask
RGB Image 6 from aachen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 7 from aachen
Ground Truth Mask
RGB Image 8 from aachen
Ground Truth Mask
RGB Image 9 from aachen
Ground Truth Mask
RGB Image 10 from aachen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

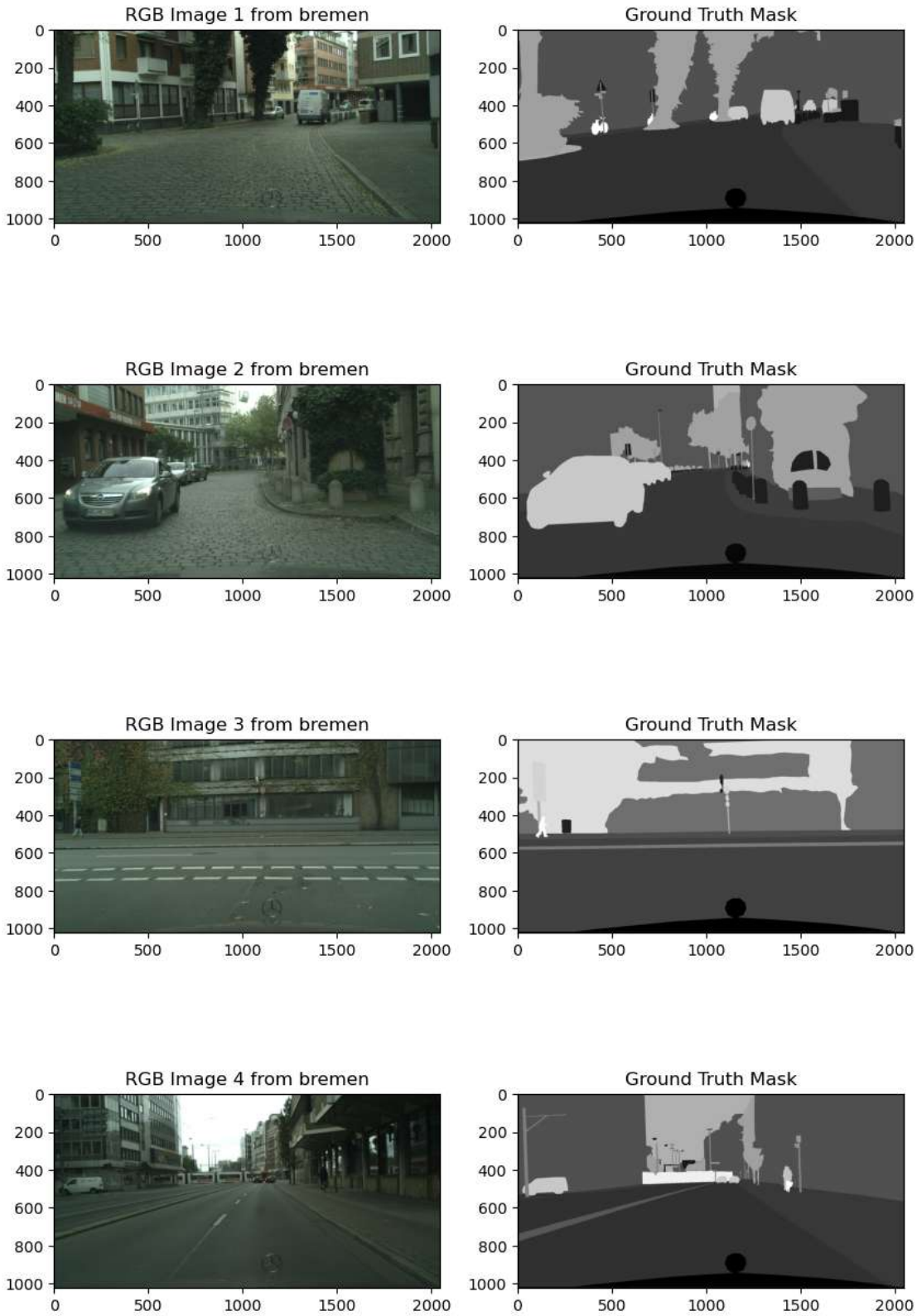
```
Finished processing city: aachen

Processing city: bochum
```

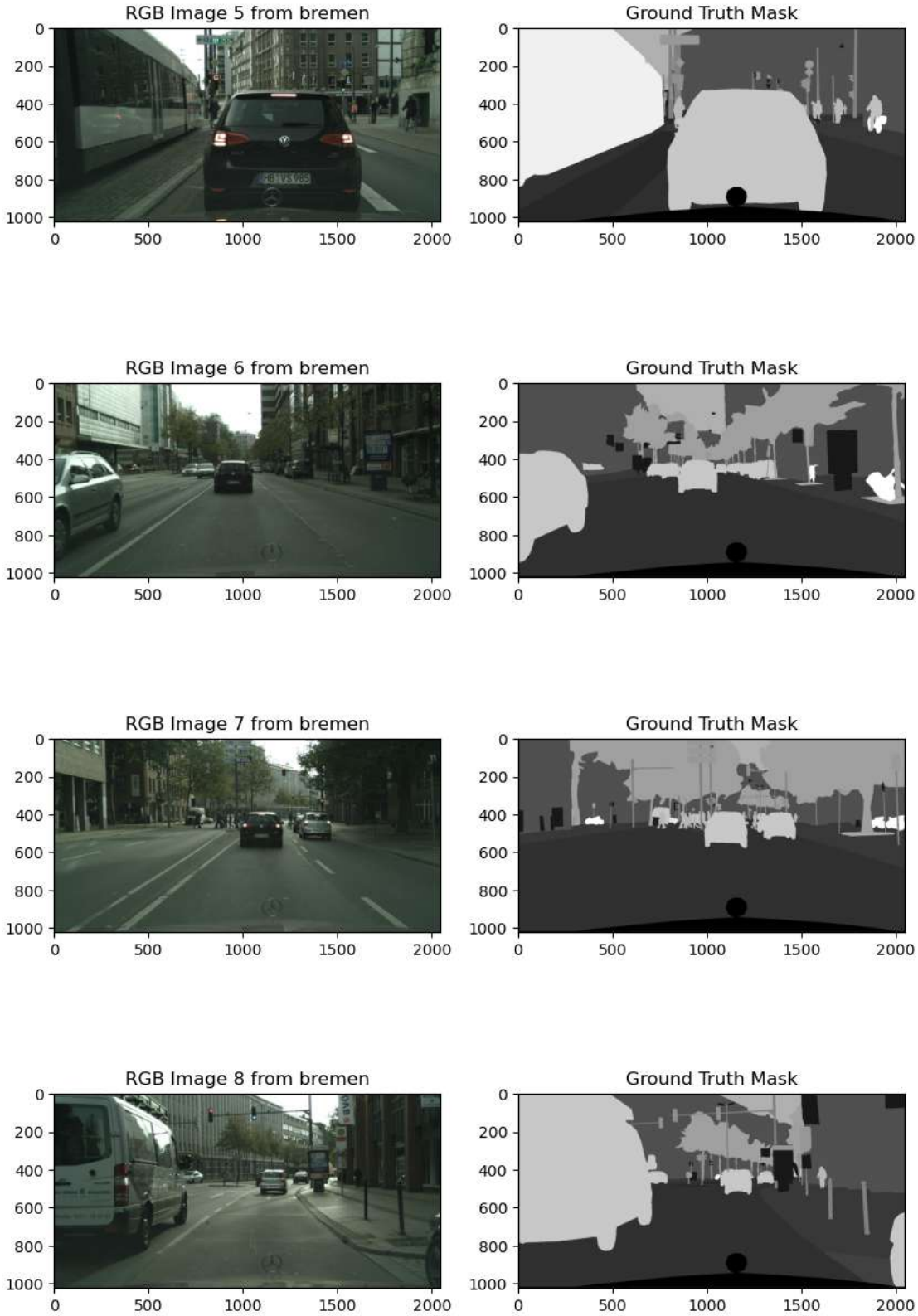


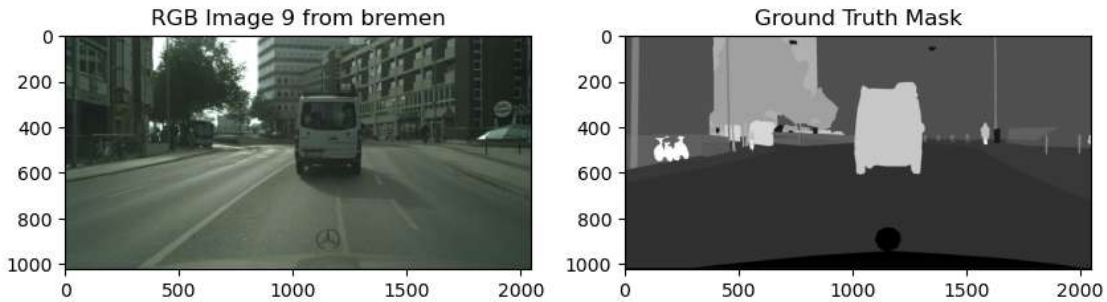
RGB Image 4 from bochum
Ground Truth Mask
RGB Image 5 from bochum
Ground Truth Mask
RGB Image 6 from bochum
Ground Truth Mask
RGB Image 7 from bochum
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

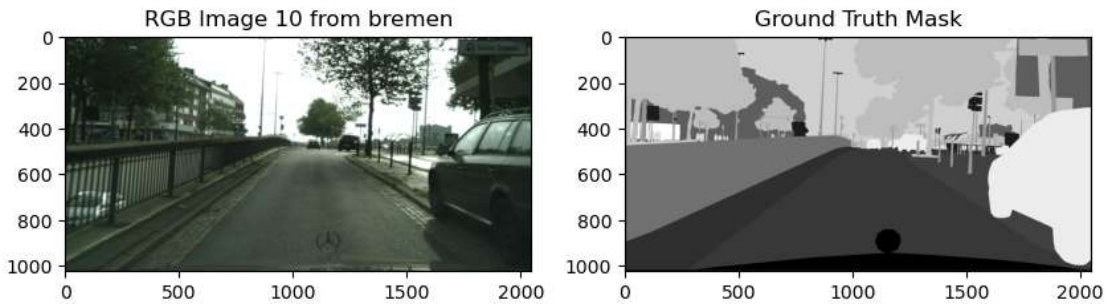



```
Finished processing city: bochum

Processing city: bremen
```

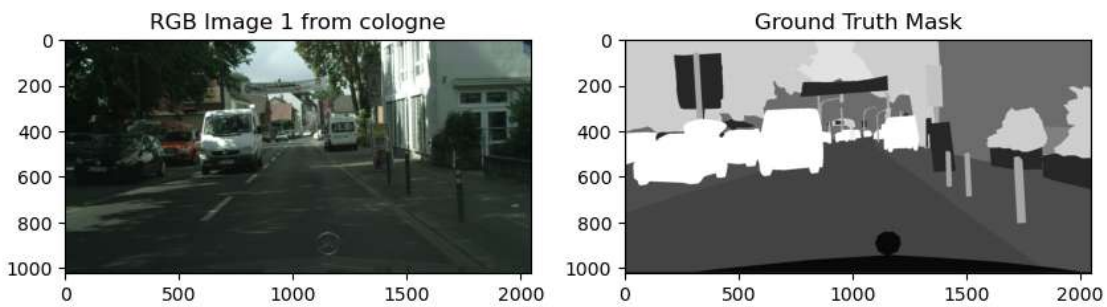

RGB Image 1 from bremen
Ground Truth Mask
RGB Image 2 from bremen
Ground Truth Mask
RGB Image 3 from bremen
Ground Truth Mask
RGB Image 4 from bremen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

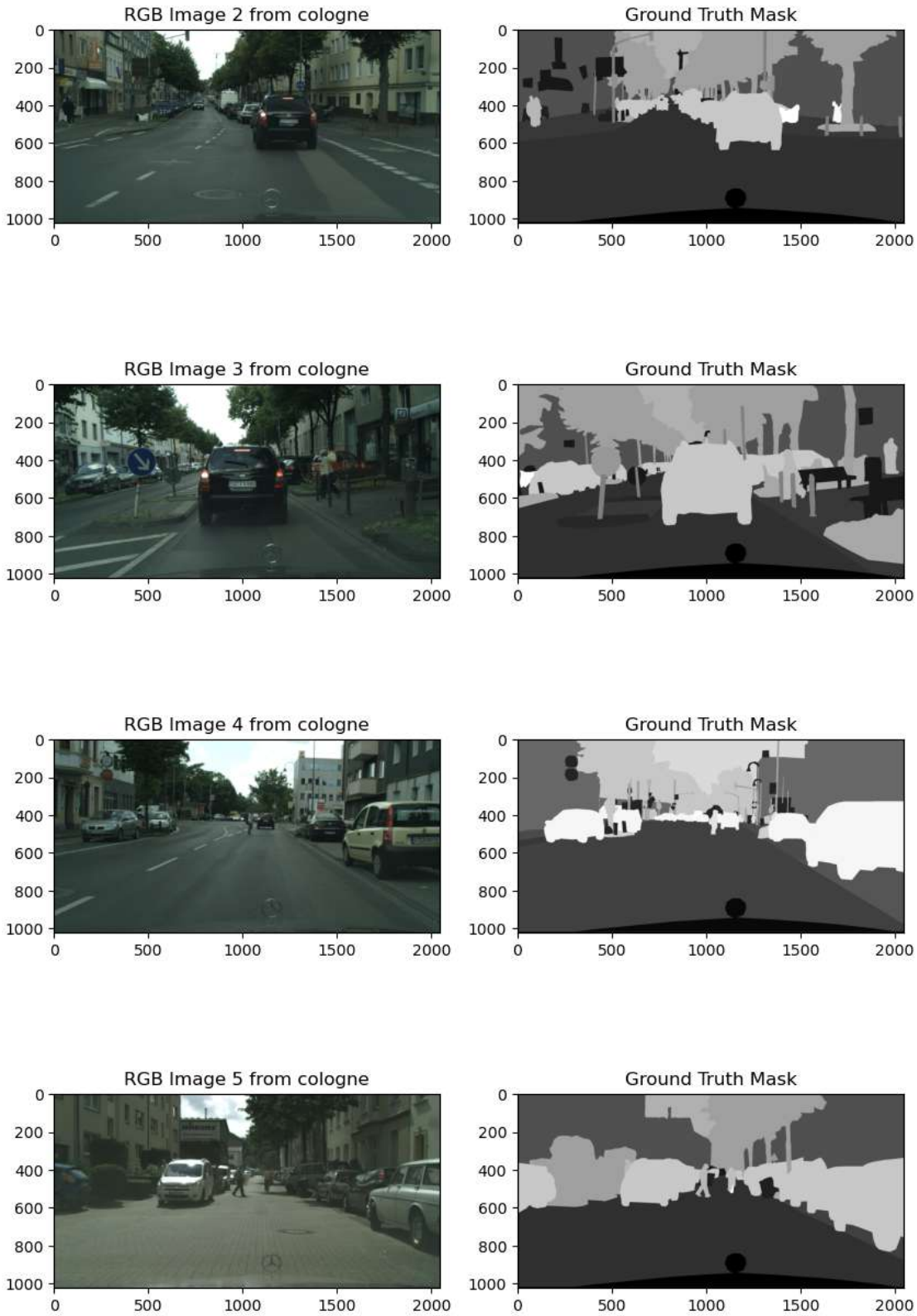

RGB Image 5 from bremen
Ground Truth Mask
RGB Image 6 from bremen
Ground Truth Mask
RGB Image 7 from bremen
Ground Truth Mask
RGB Image 8 from bremen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

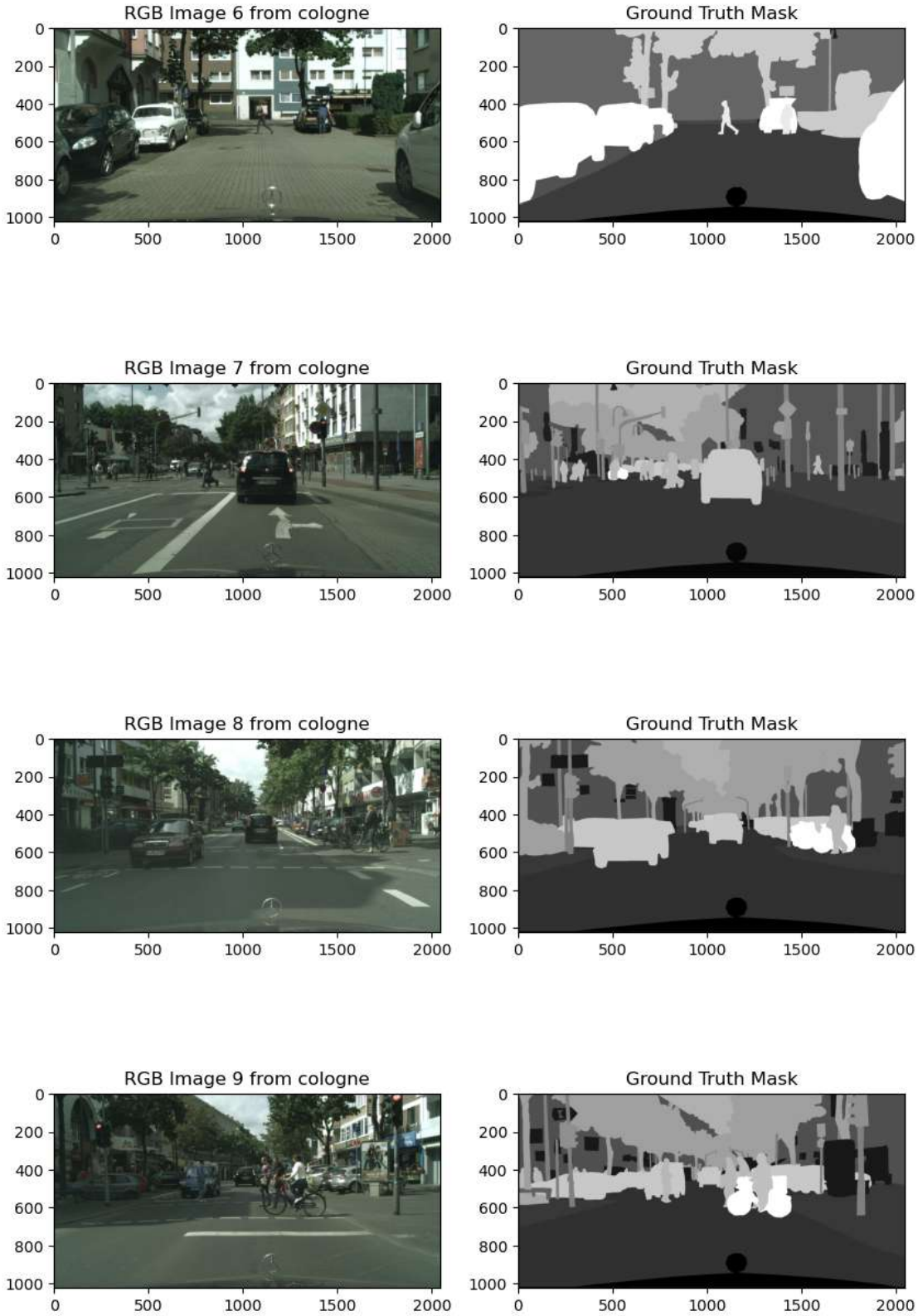


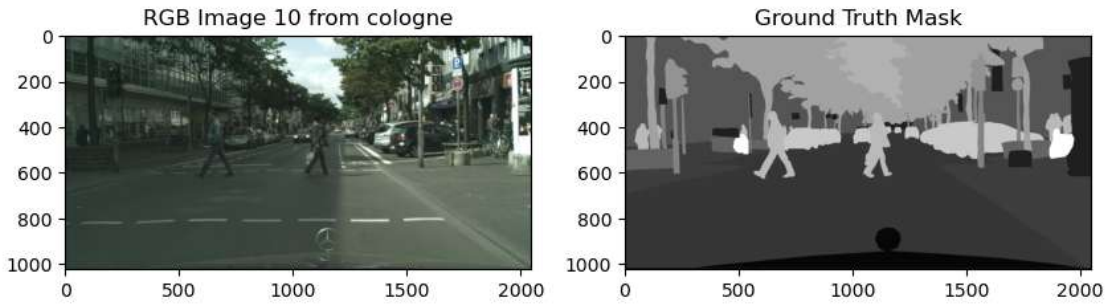



```
Finished processing city: bremen

Processing city: cologne
```

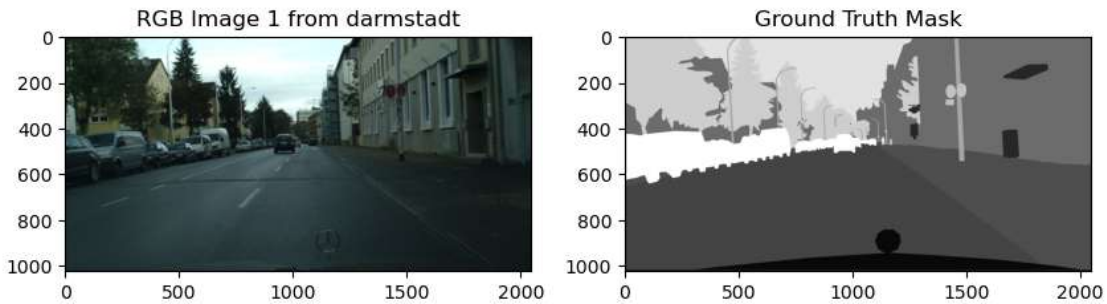


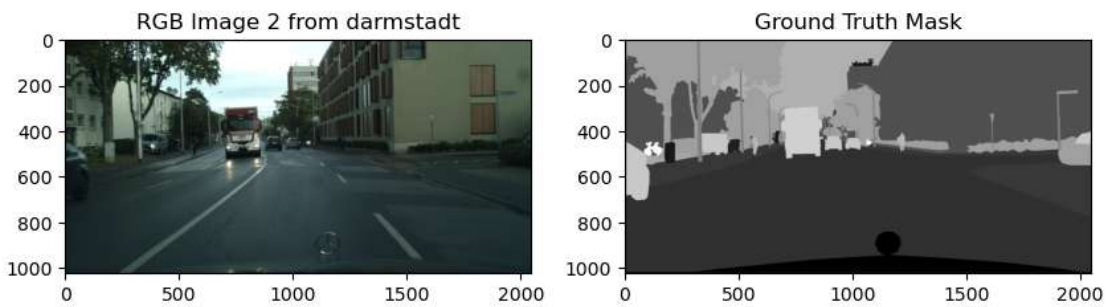

RGB Image 2 from cologne
Ground Truth Mask
RGB Image 3 from cologne
Ground Truth Mask
RGB Image 4 from cologne
Ground Truth Mask
RGB Image 5 from cologne
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 6 from cologne
Ground Truth Mask
RGB Image 7 from cologne
Ground Truth Mask
RGB Image 8 from cologne
Ground Truth Mask
RGB Image 9 from cologne
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

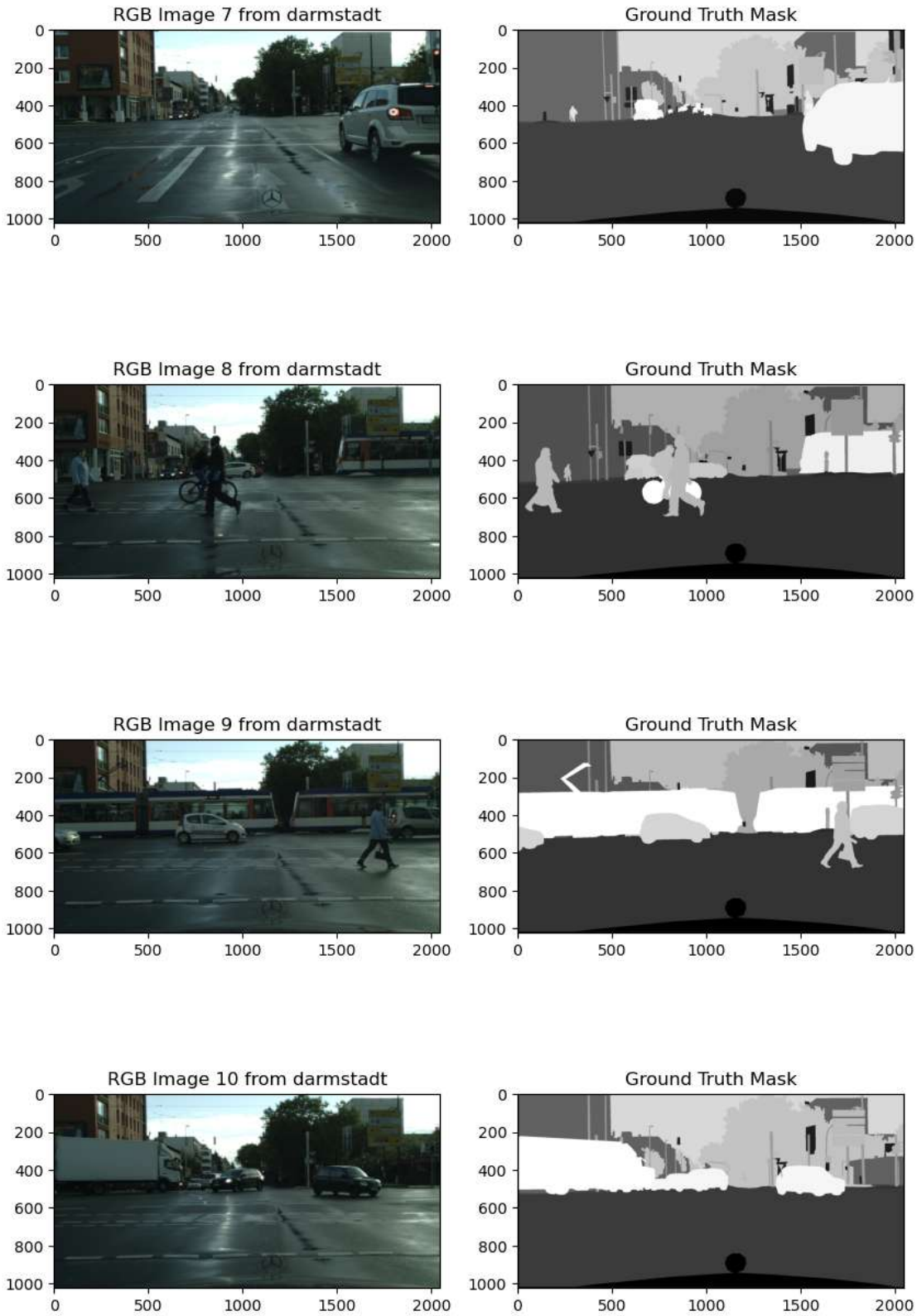



```
Finished processing city: cologne

Processing city: darmstadt
```

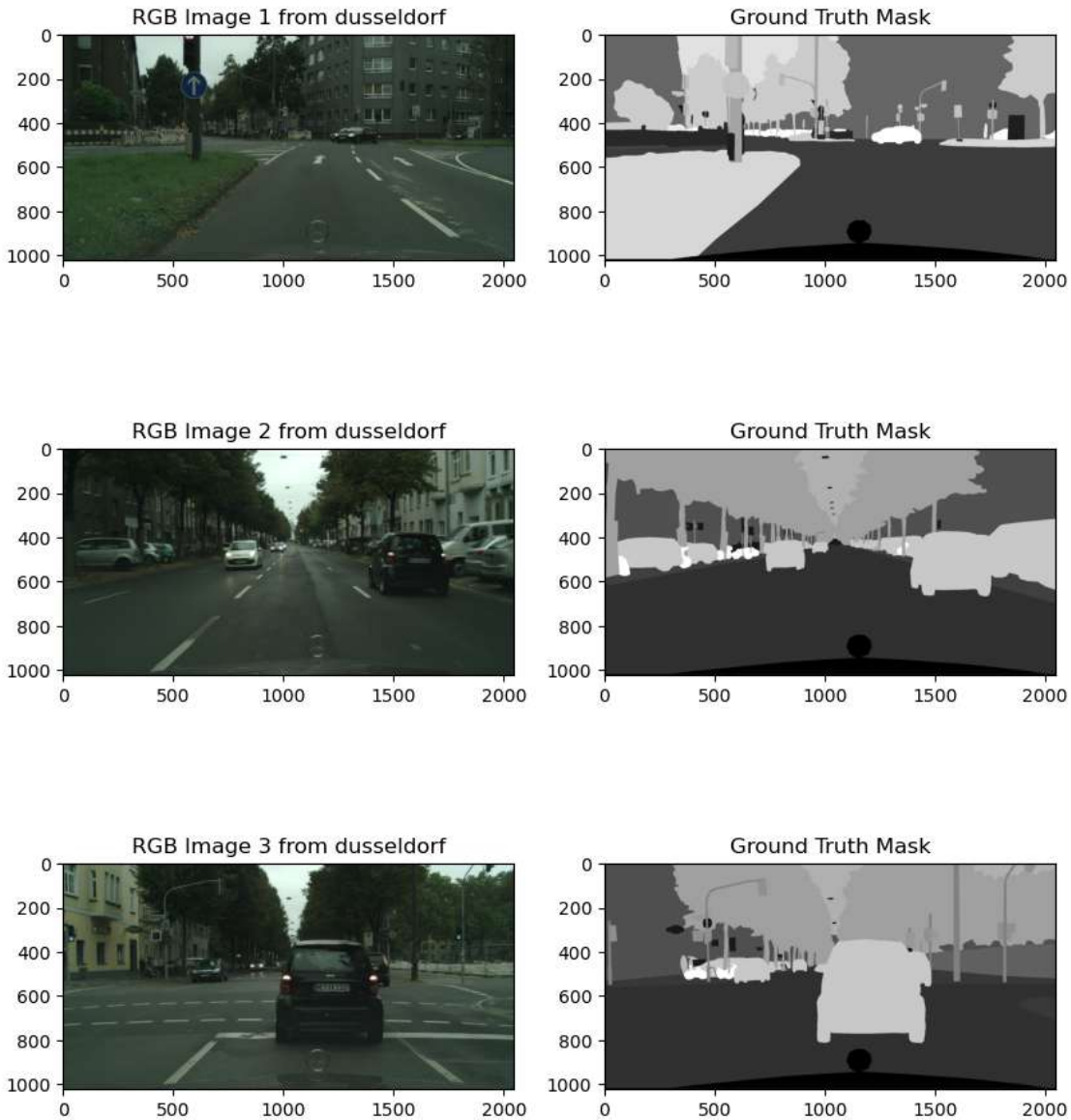


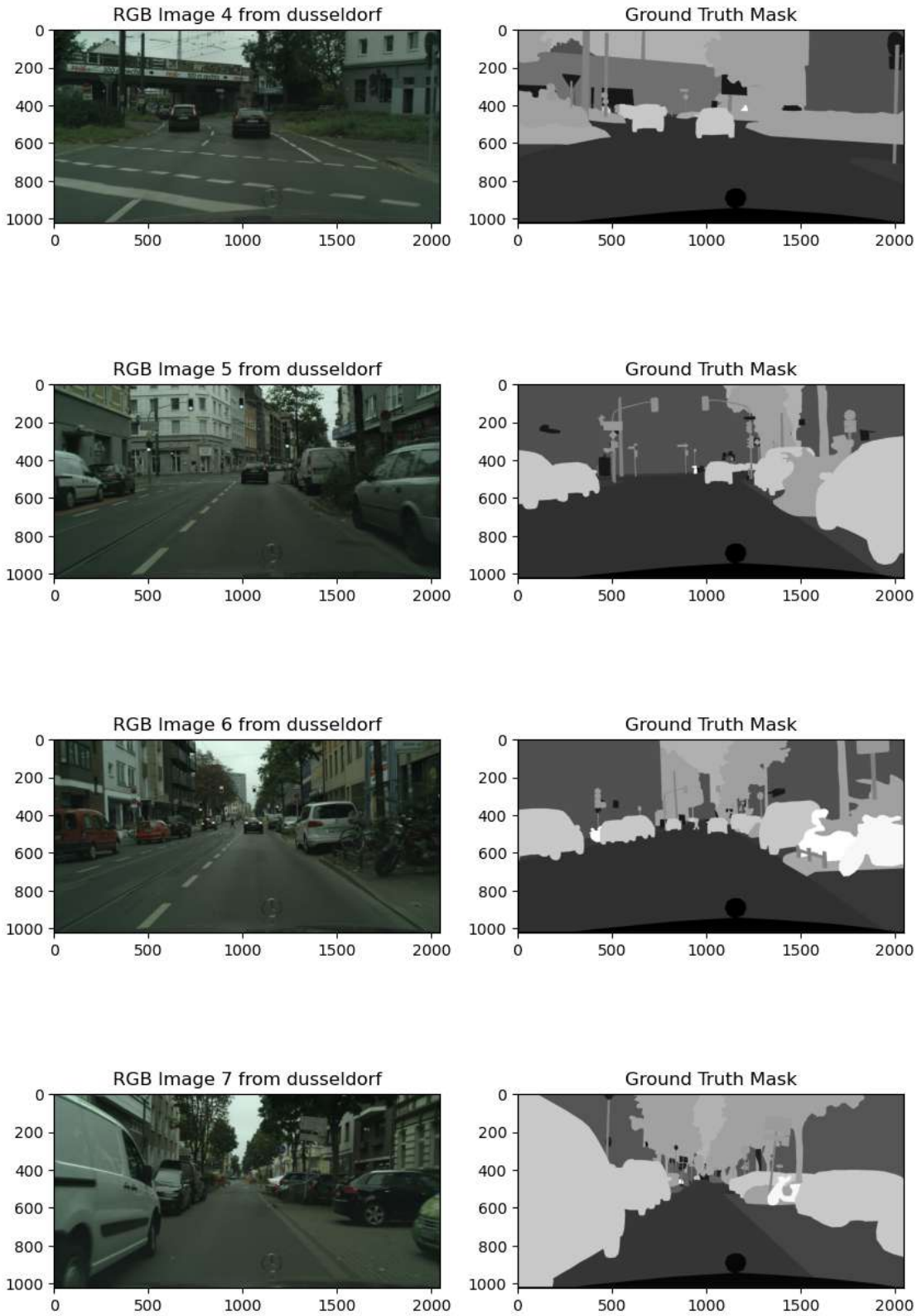


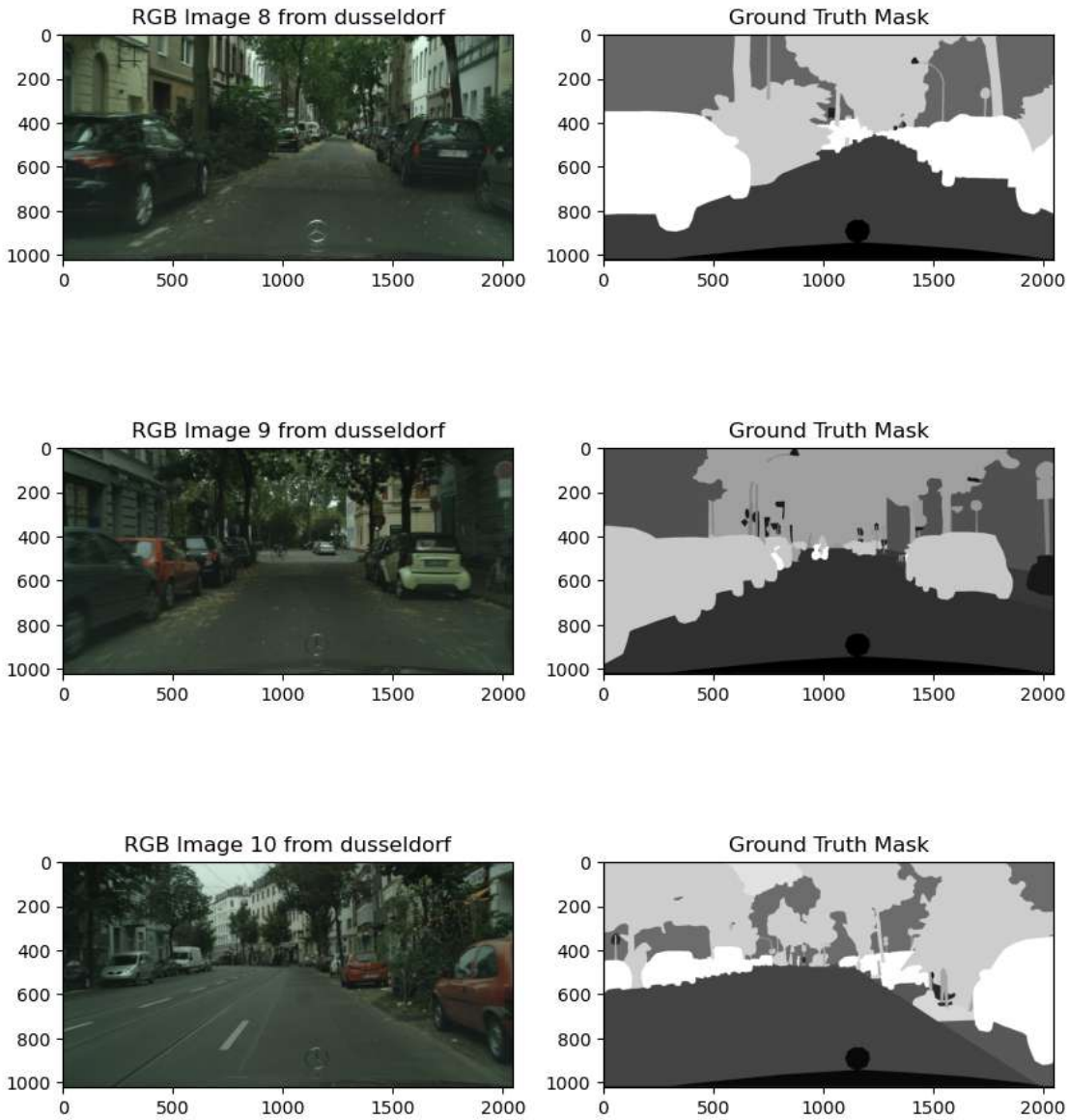
RGB Image 3 from darmstadt
Ground Truth Mask
RGB Image 4 from darmstadt
Ground Truth Mask
RGB Image 5 from darmstadt
Ground Truth Mask
RGB Image 6 from darmstadt
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 7 from darmstadt
Ground Truth Mask
RGB Image 8 from darmstadt
Ground Truth Mask
RGB Image 9 from darmstadt
Ground Truth Mask
RGB Image 10 from darmstadt
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

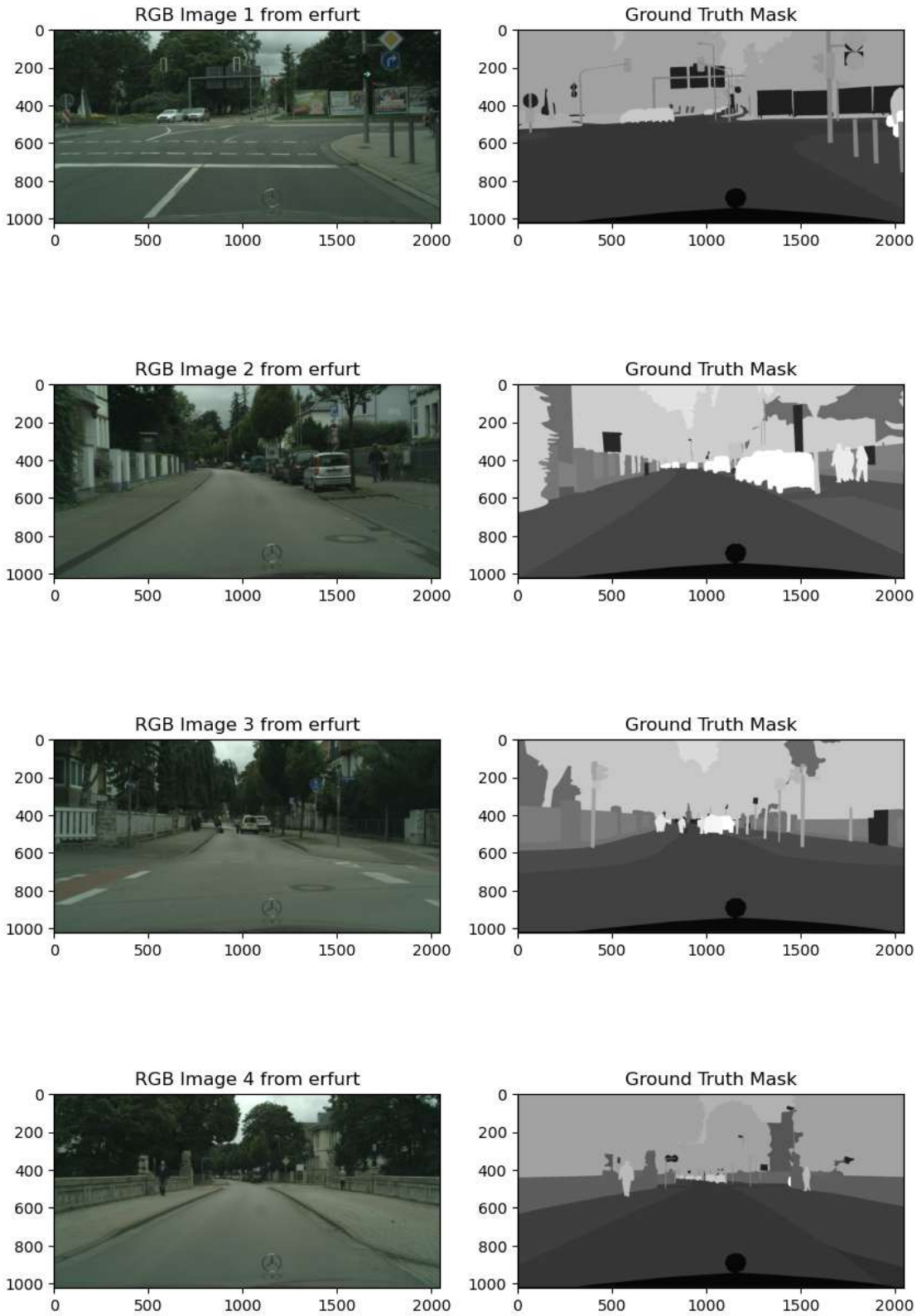
```
Finished processing city: darmstadt

Processing city: dusseldorf
```

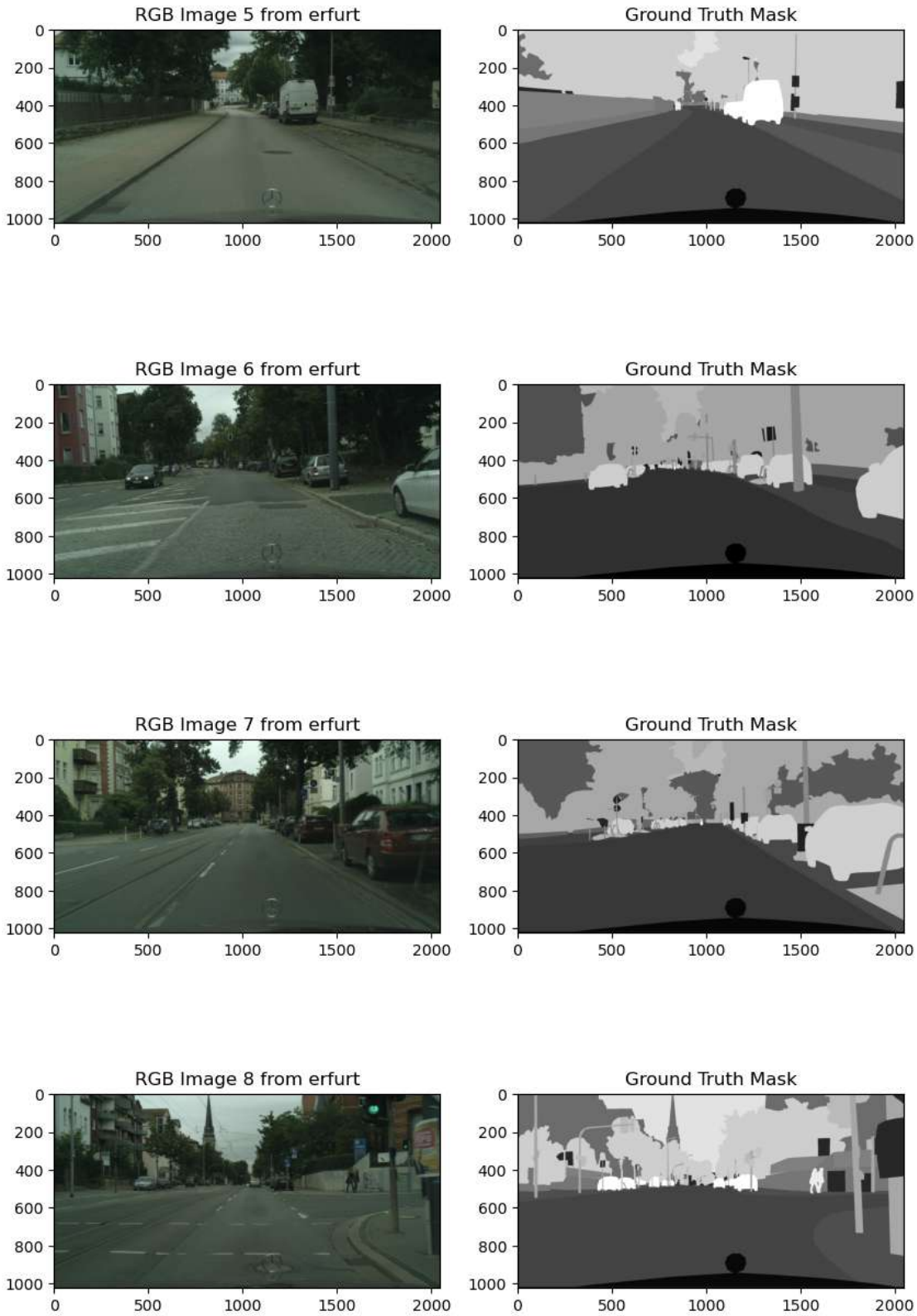


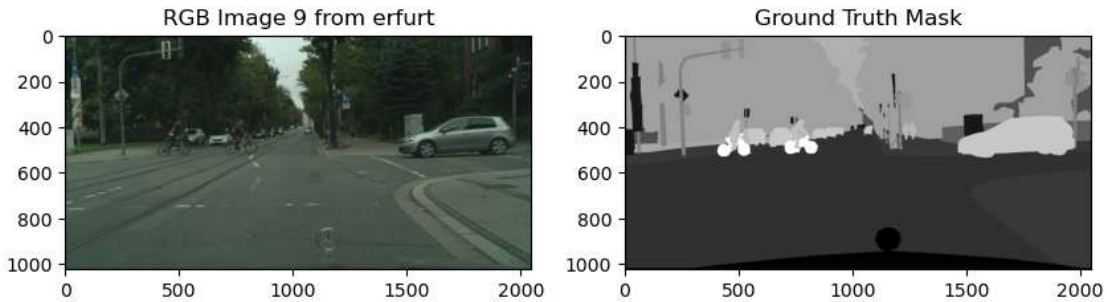

RGB Image 4 from dusseldorf
Ground Truth Mask
RGB Image 5 from dusseldorf
Ground Truth Mask
RGB Image 6 from dusseldorf
Ground Truth Mask
RGB Image 7 from dusseldorf
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

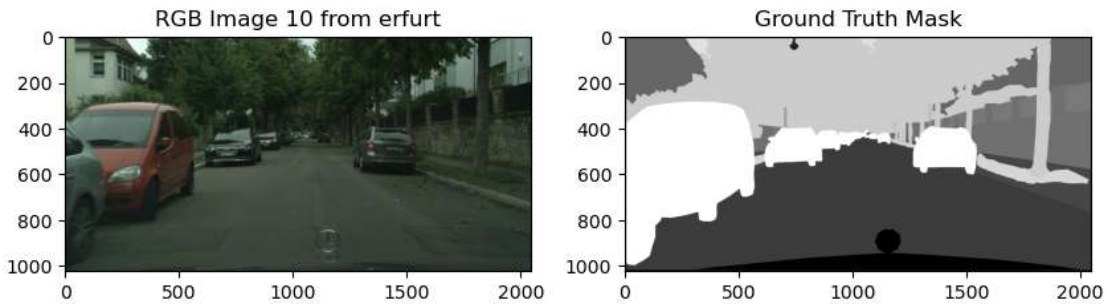



```
Finished processing city: dusseldorf

Processing city: erfurt
```

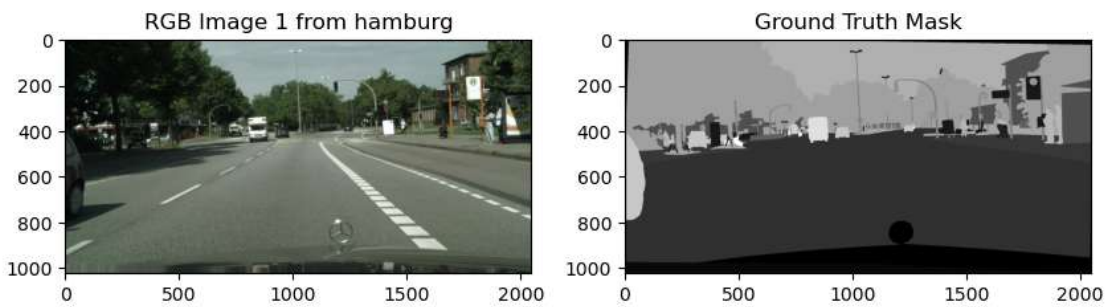

RGB Image 1 from erfurt
Ground Truth Mask
RGB Image 2 from erfurt
Ground Truth Mask
RGB Image 3 from erfurt
Ground Truth Mask
RGB Image 4 from erfurt
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

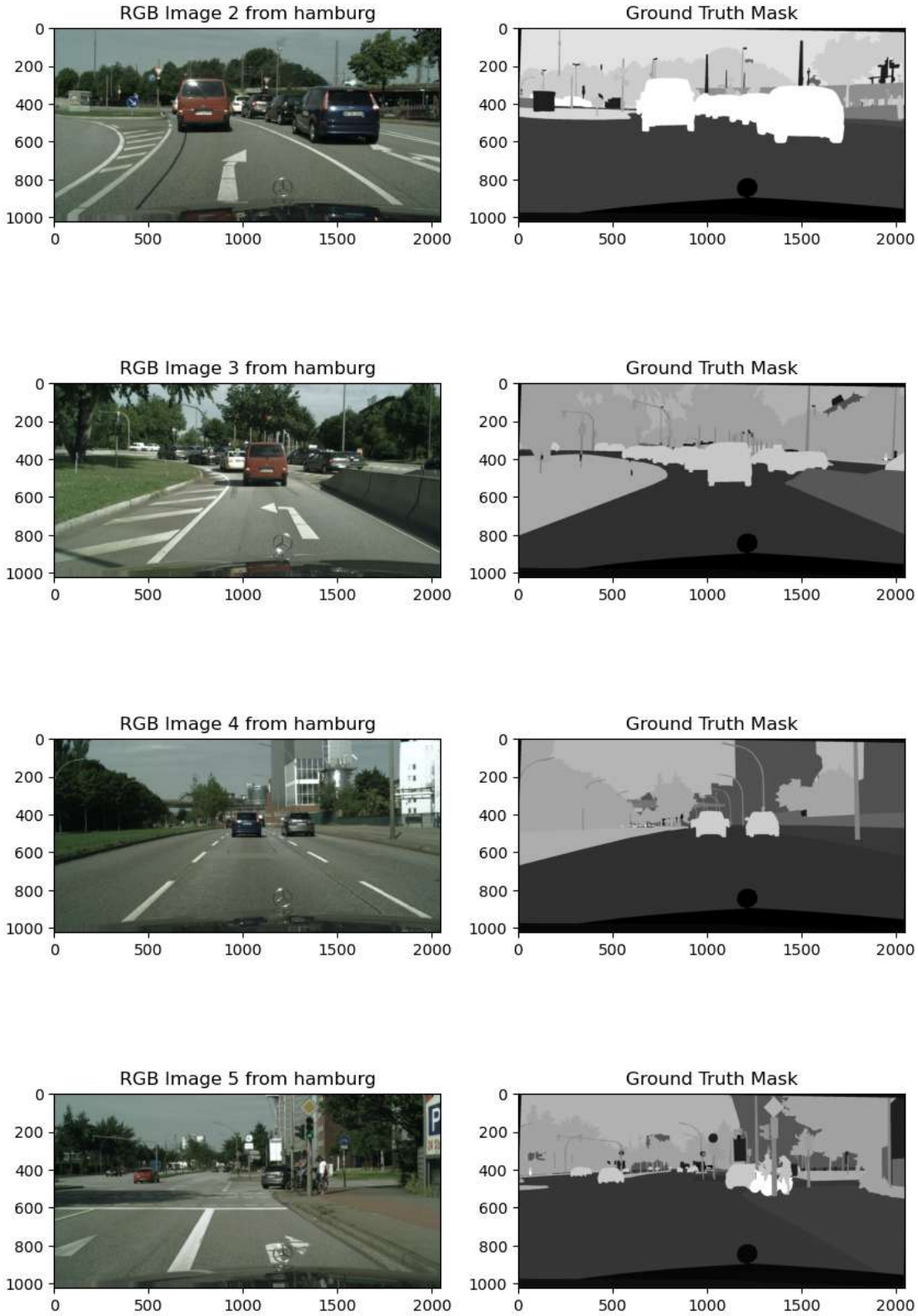

RGB Image 5 from erfurt
Ground Truth Mask
RGB Image 6 from erfurt
Ground Truth Mask
RGB Image 7 from erfurt
Ground Truth Mask
RGB Image 8 from erfurt
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

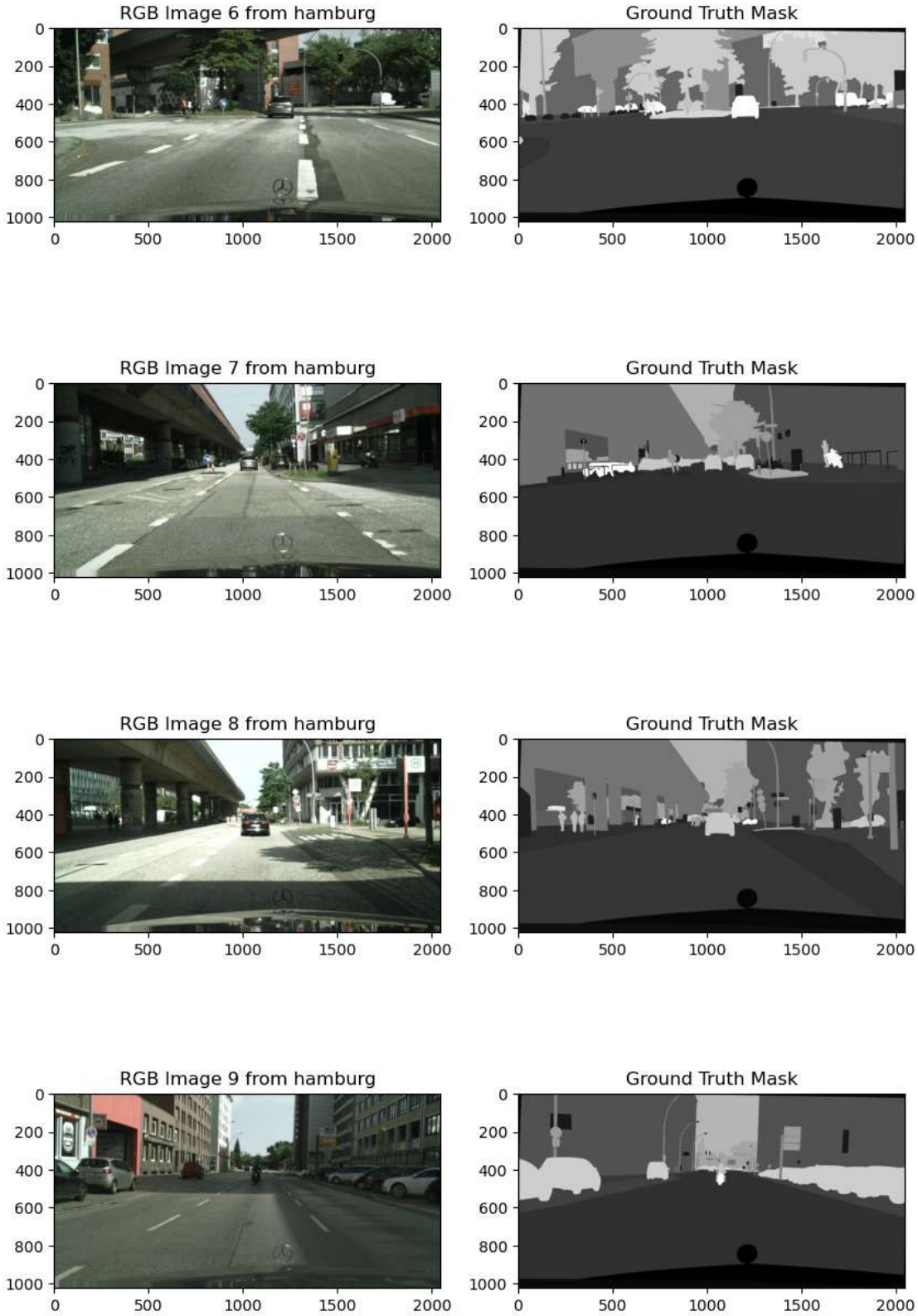


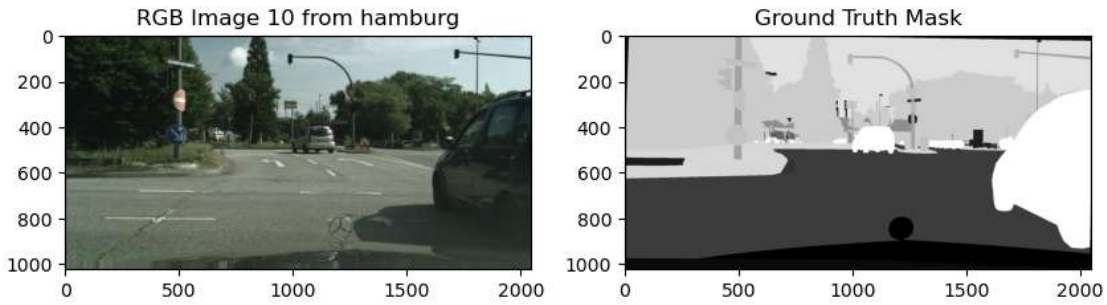



```
Finished processing city: erfurt

Processing city: hamburg
```

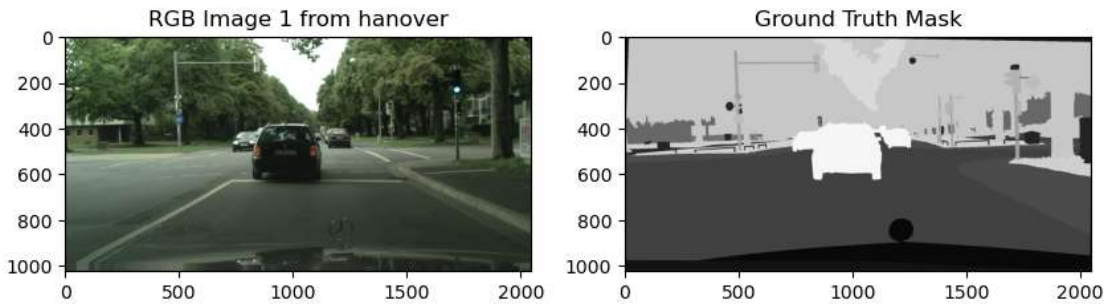


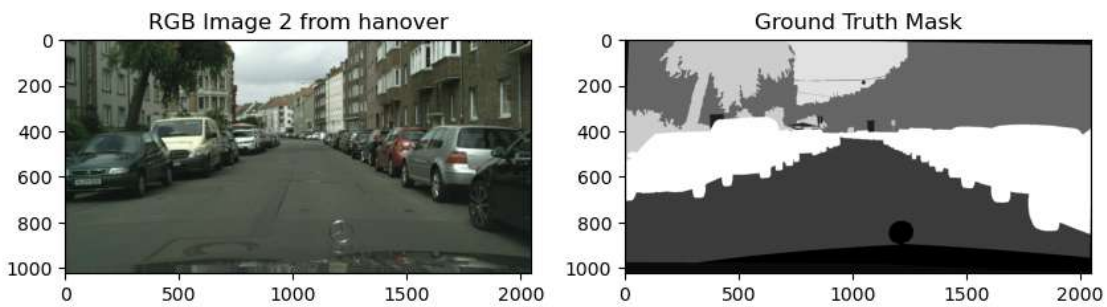

RGB Image 2 from hamburg
Ground Truth Mask
RGB Image 3 from hamburg
Ground Truth Mask
RGB Image 4 from hamburg
Ground Truth Mask
RGB Image 5 from hamburg
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

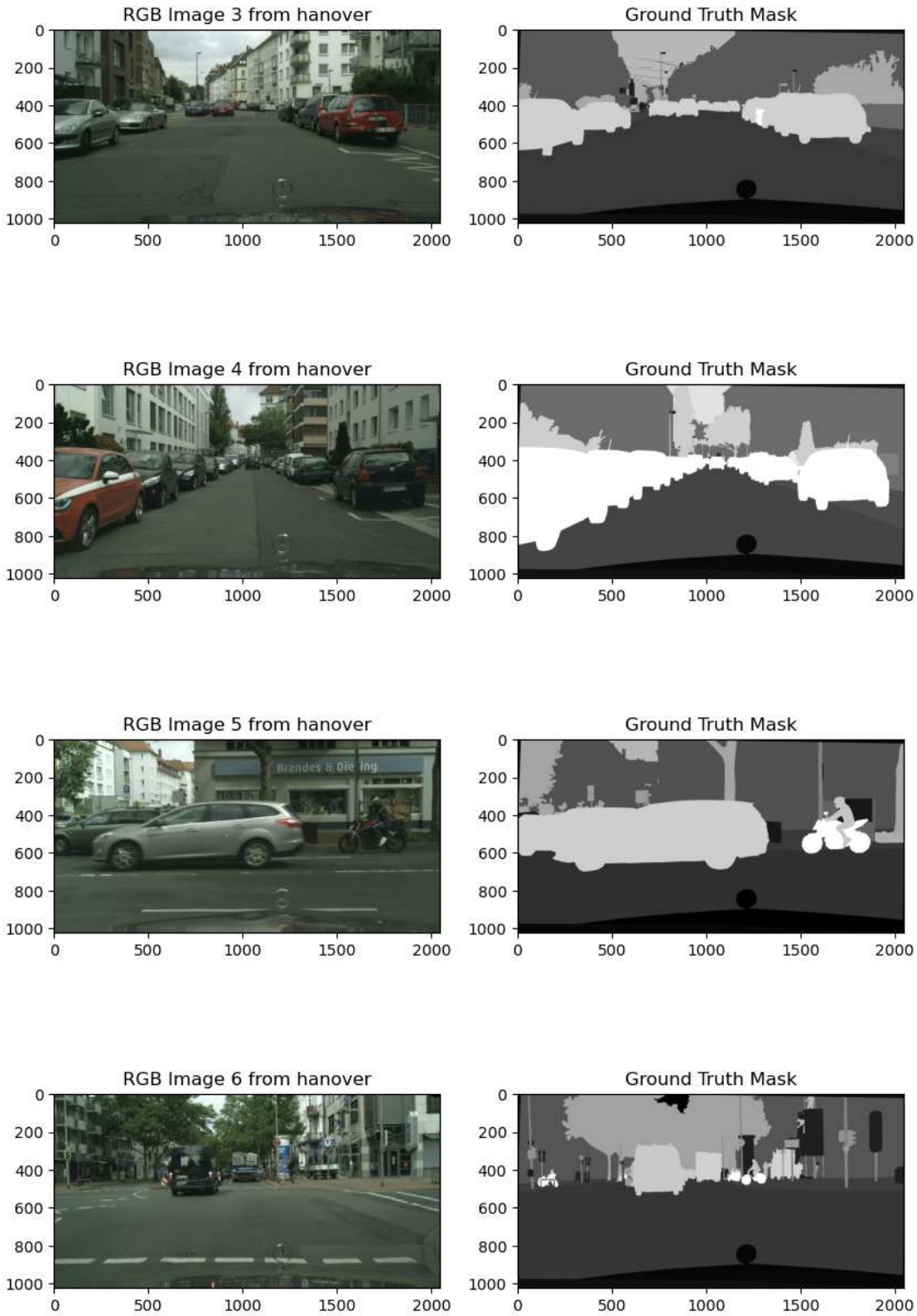

RGB Image 6 from hamburg
Ground Truth Mask
RGB Image 7 from hamburg
Ground Truth Mask
RGB Image 8 from hamburg
Ground Truth Mask
RGB Image 9 from hamburg
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

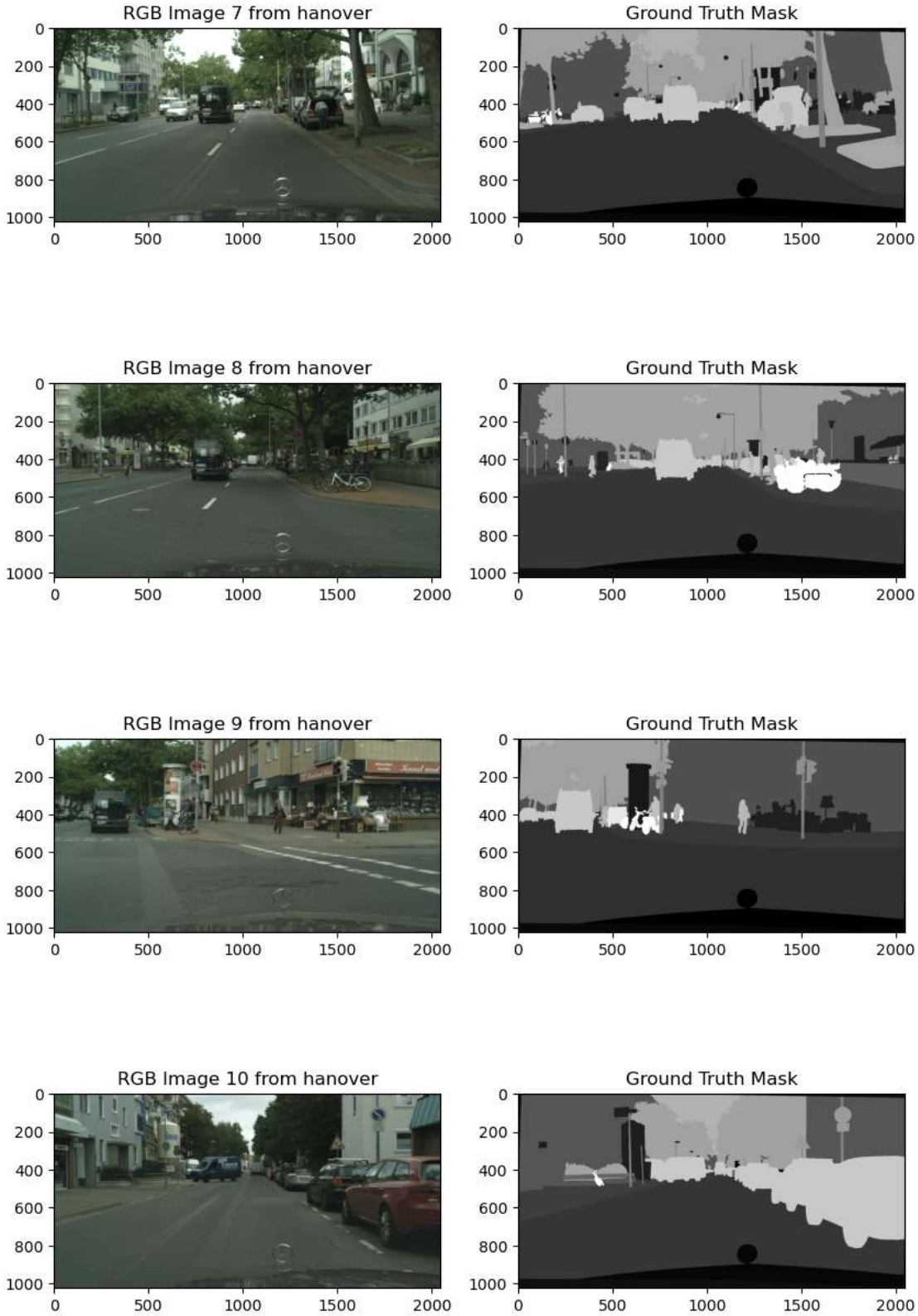



```
Finished processing city: hamburg

Processing city: hanover
```

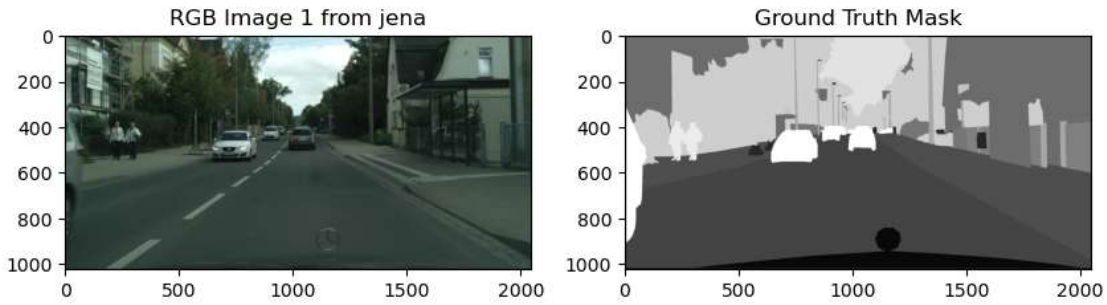




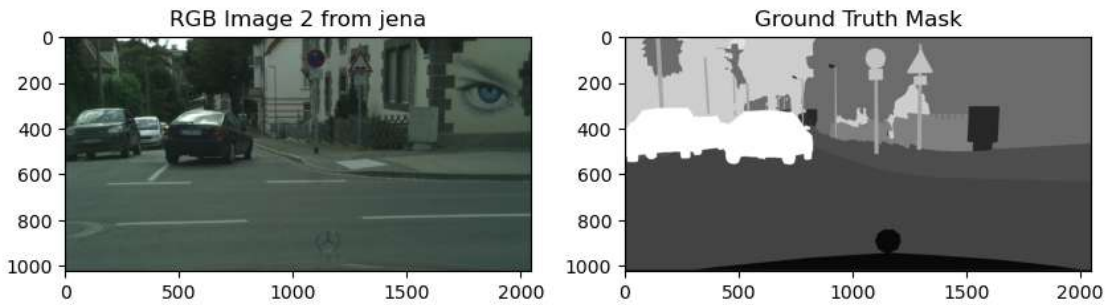

RGB Image 3 from hanover
Ground Truth Mask
RGB Image 4 from hanover
Ground Truth Mask
RGB Image 5 from hanover
Ground Truth Mask
RGB Image 6 from hanover
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

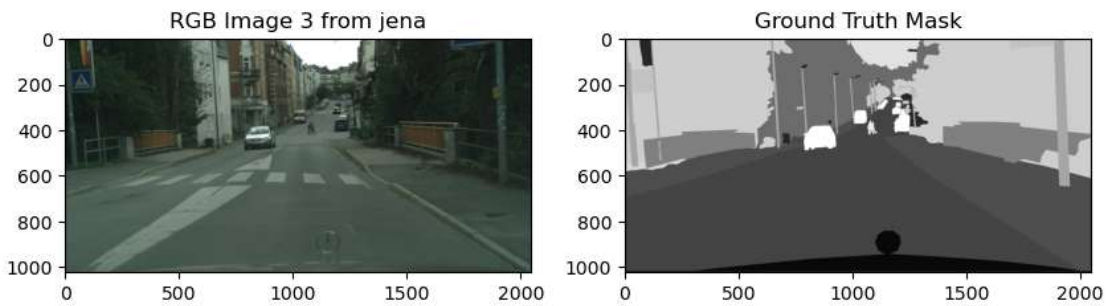

RGB Image 7 from hanover
Ground Truth Mask
RGB Image 8 from hanover
Ground Truth Mask
RGB Image 9 from hanover
Ground Truth Mask
RGB Image 10 from hanover
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

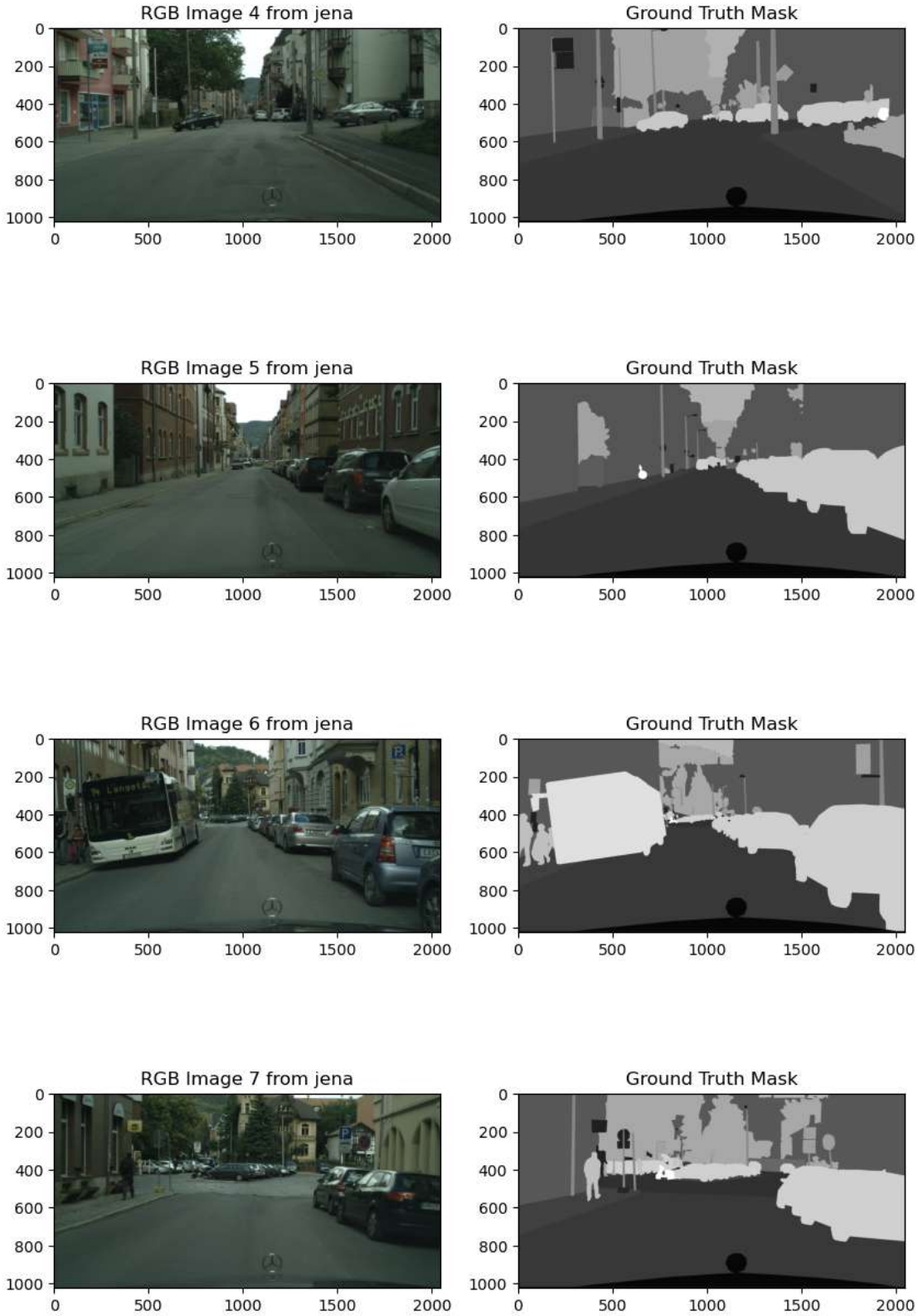
```
Finished processing city: hanover

Processing city: jena
```

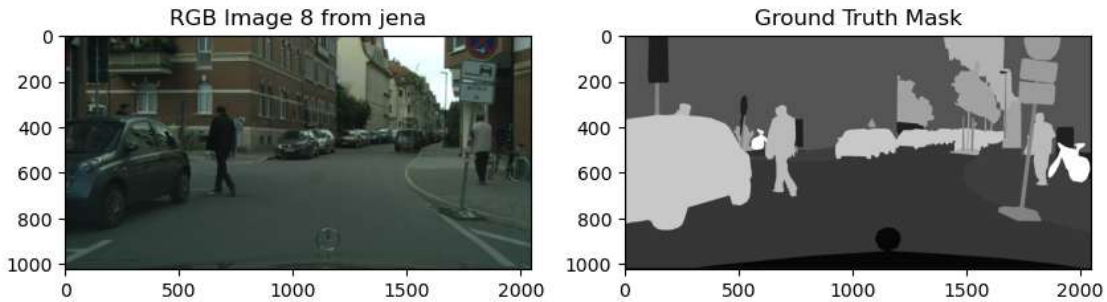


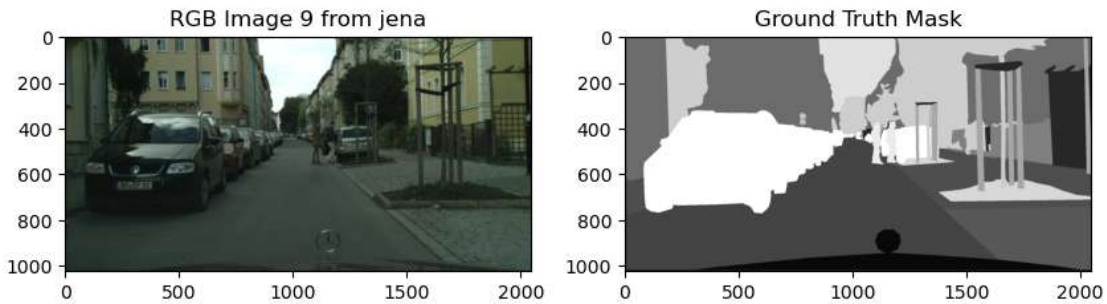

```
Ground truth file not found for: jena_000001_000019_leftImg8bit.png
Ground truth file not found for: jena_000002_000019_leftImg8bit.png
```



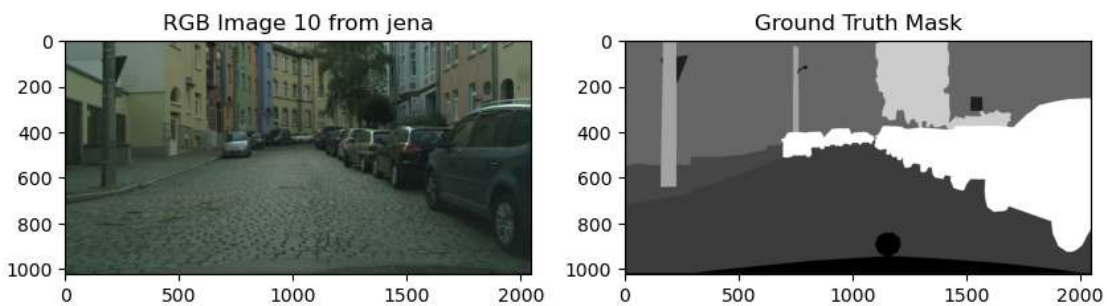



RGB Image 4 from jena
Ground Truth Mask
RGB Image 5 from jena
Ground Truth Mask
RGB Image 6 from jena
Ground Truth Mask
RGB Image 7 from jena
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

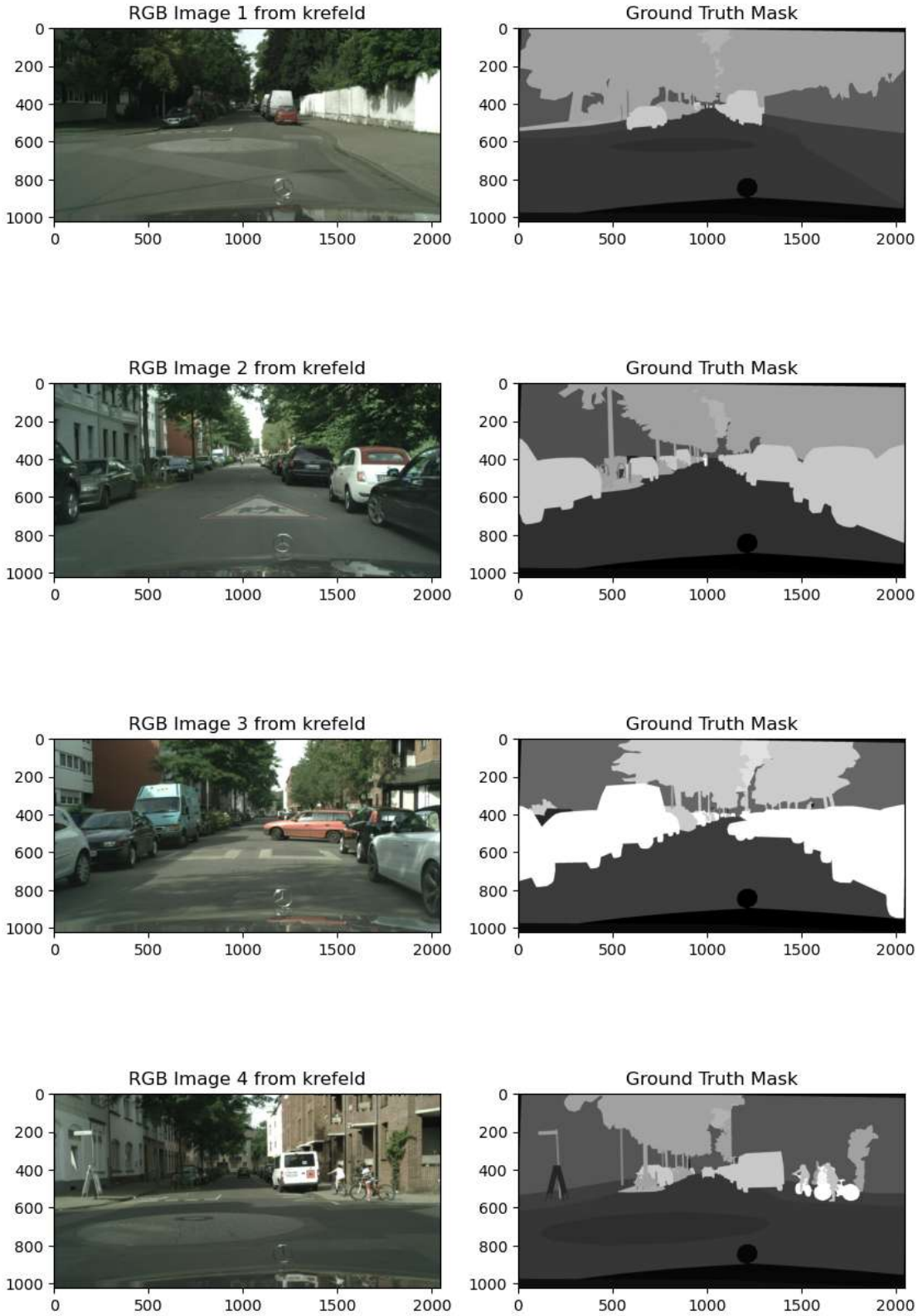


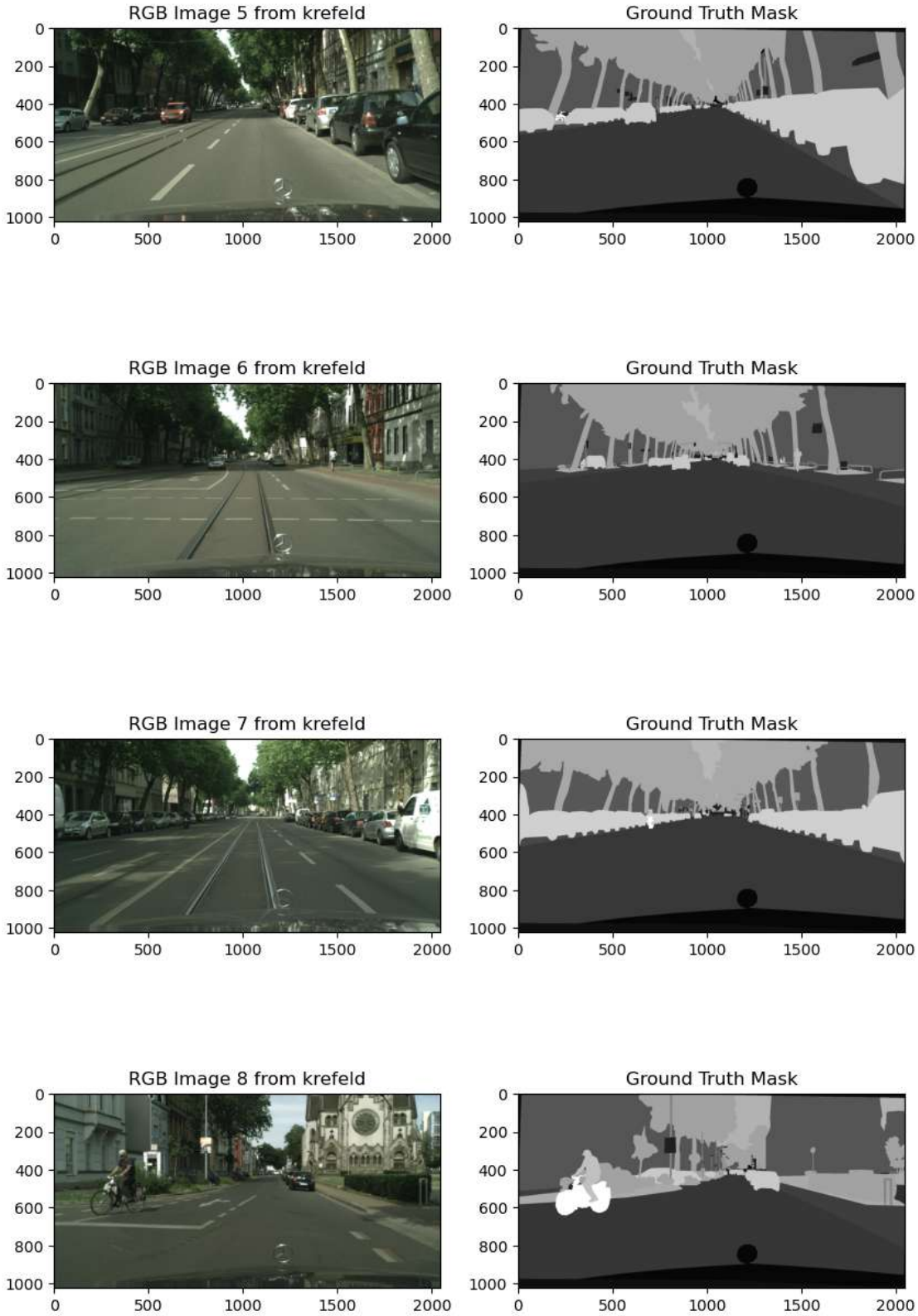


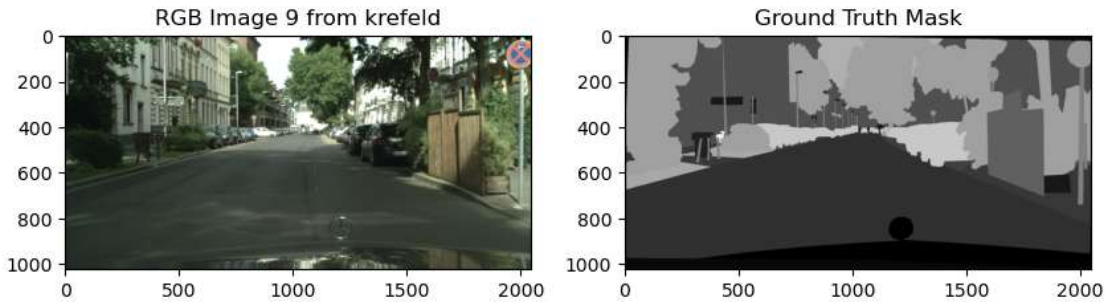

```
Ground truth file not found for: jena_000011_000019_leftImg8bit.png
```

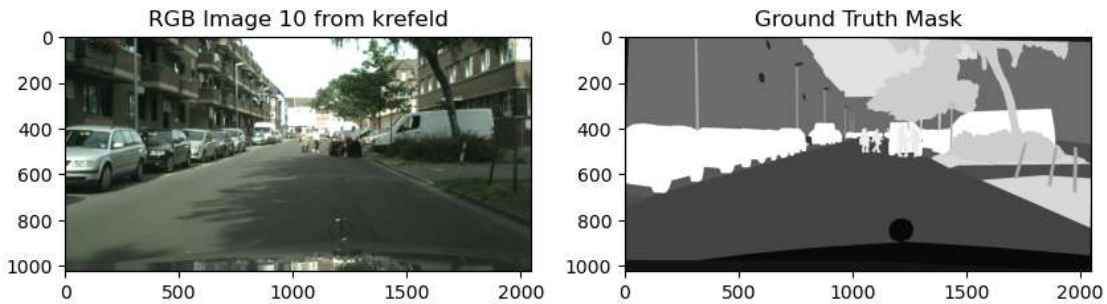



```
Finished processing city: jena

Processing city: krefeld
```

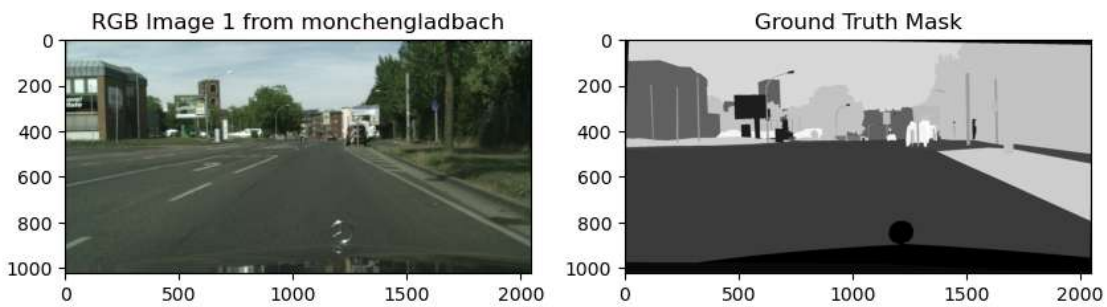

RGB Image 1 from krefeld
Ground Truth Mask
RGB Image 2 from krefeld
Ground Truth Mask
RGB Image 3 from krefeld
Ground Truth Mask
RGB Image 4 from krefeld
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

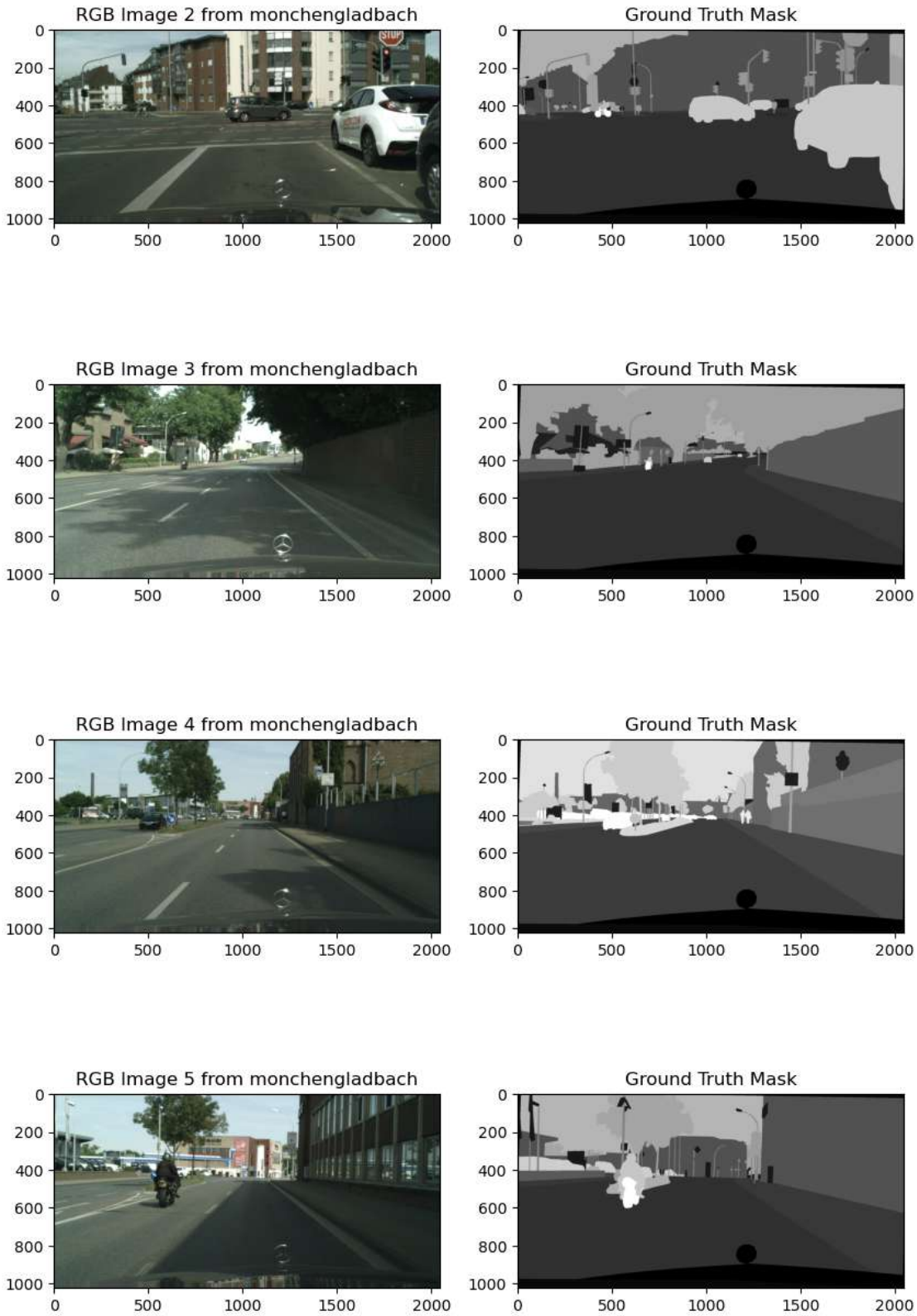

RGB Image 5 from krefeld
Ground Truth Mask
RGB Image 6 from krefeld
Ground Truth Mask
RGB Image 7 from krefeld
Ground Truth Mask
RGB Image 8 from krefeld
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

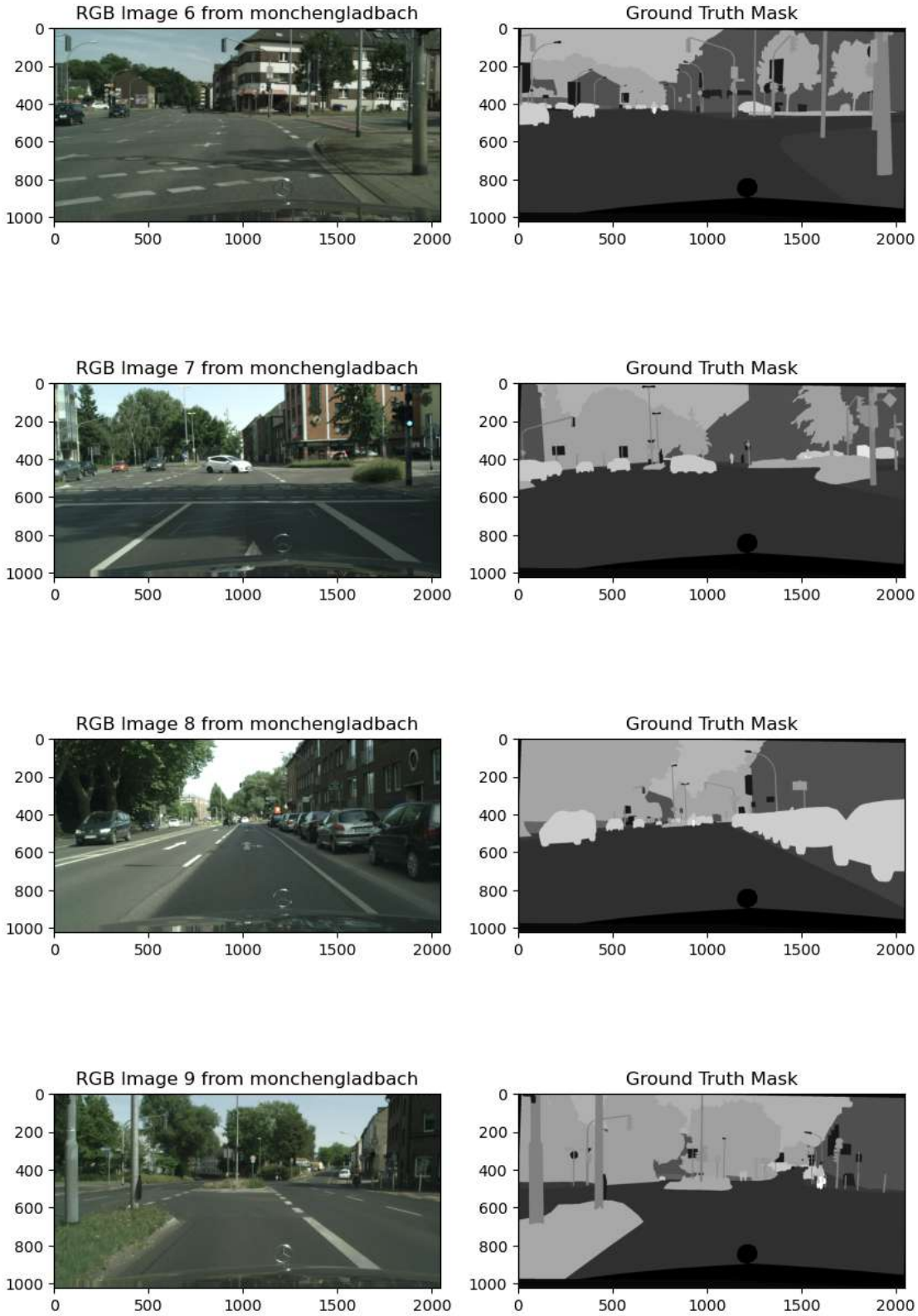


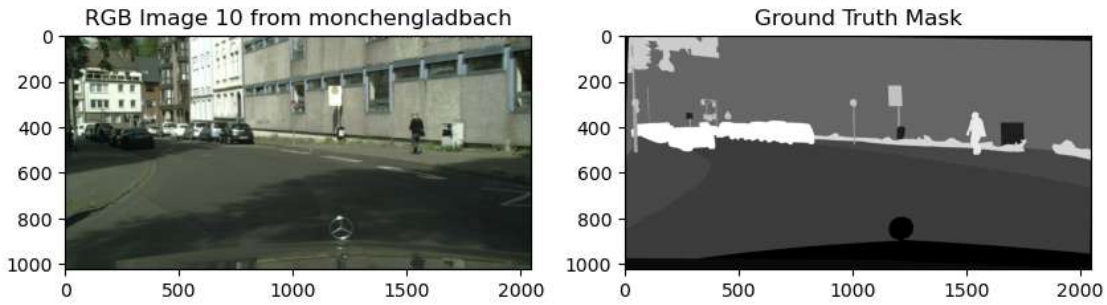



```
Finished processing city: krefeld

Processing city: monchengladbach
```

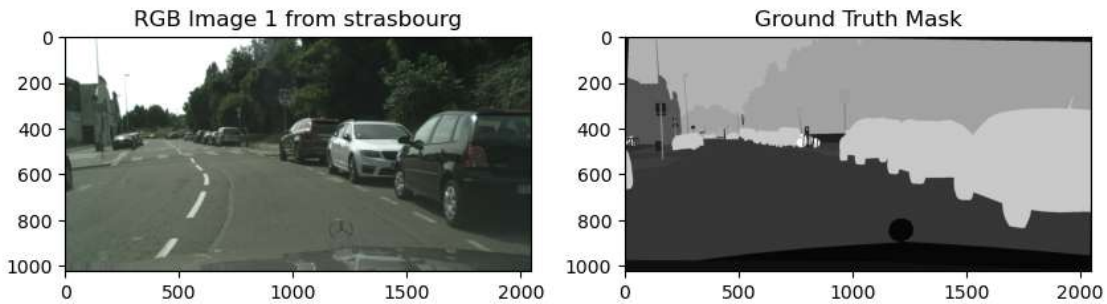


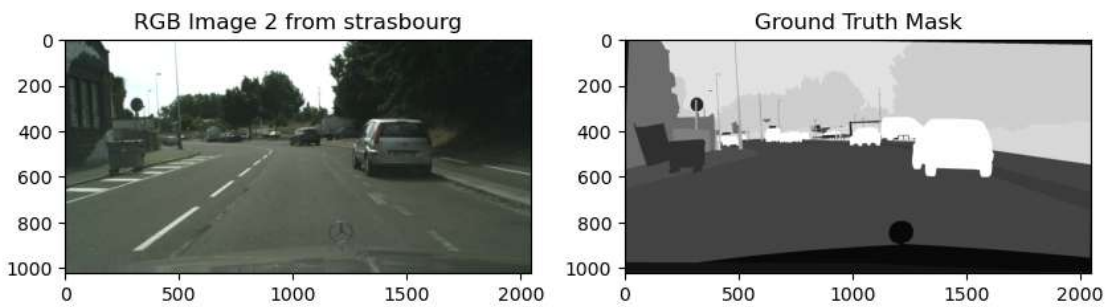

RGB Image 2 from monchengladbach
Ground Truth Mask
RGB Image 3 from monchengladbach
Ground Truth Mask
RGB Image 4 from monchengladbach
Ground Truth Mask
RGB Image 5 from monchengladbach
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

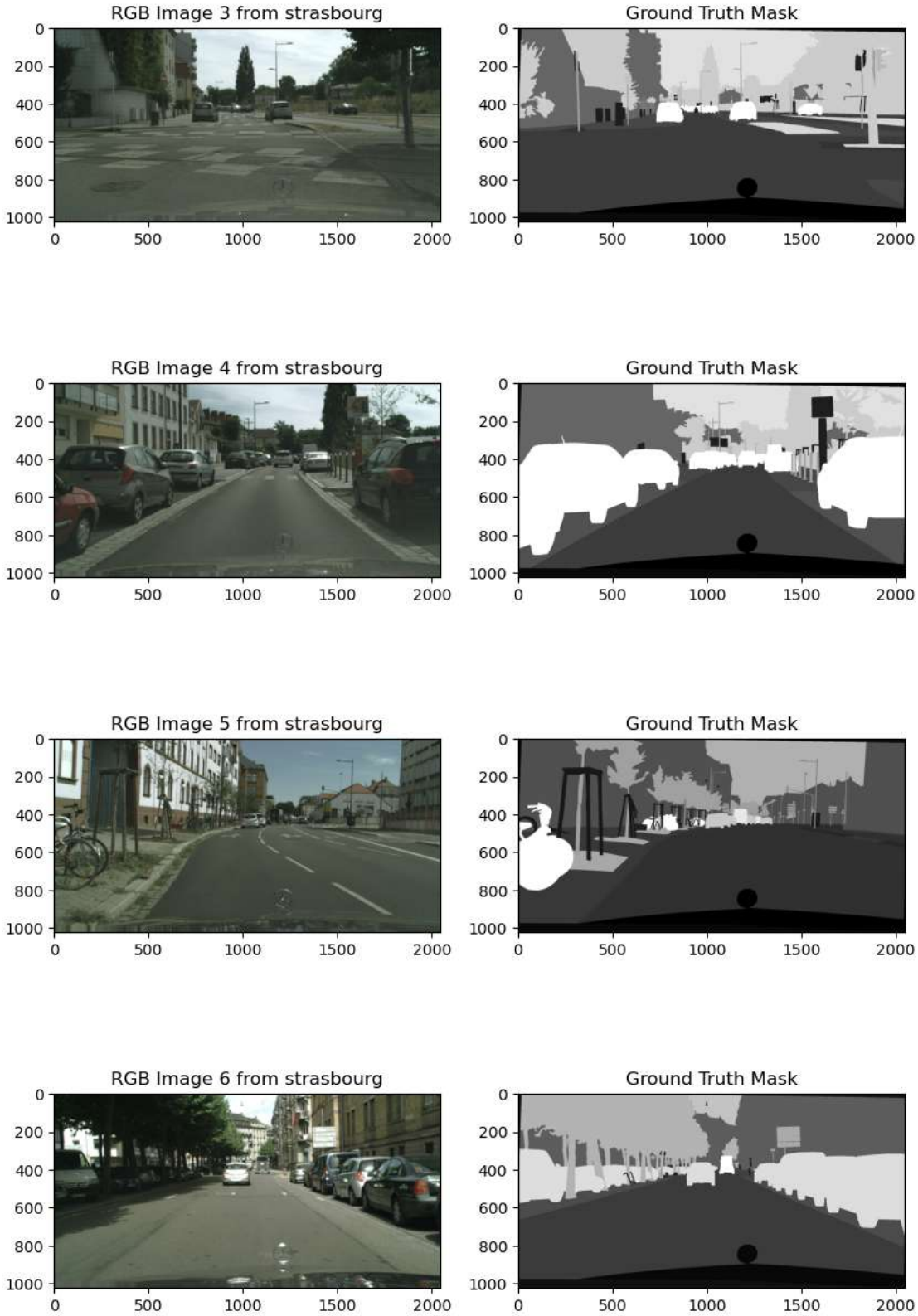

RGB Image 6 from monchengladbach
Ground Truth Mask
RGB Image 7 from monchengladbach
Ground Truth Mask
RGB Image 8 from monchengladbach
Ground Truth Mask
RGB Image 9 from monchengladbach
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

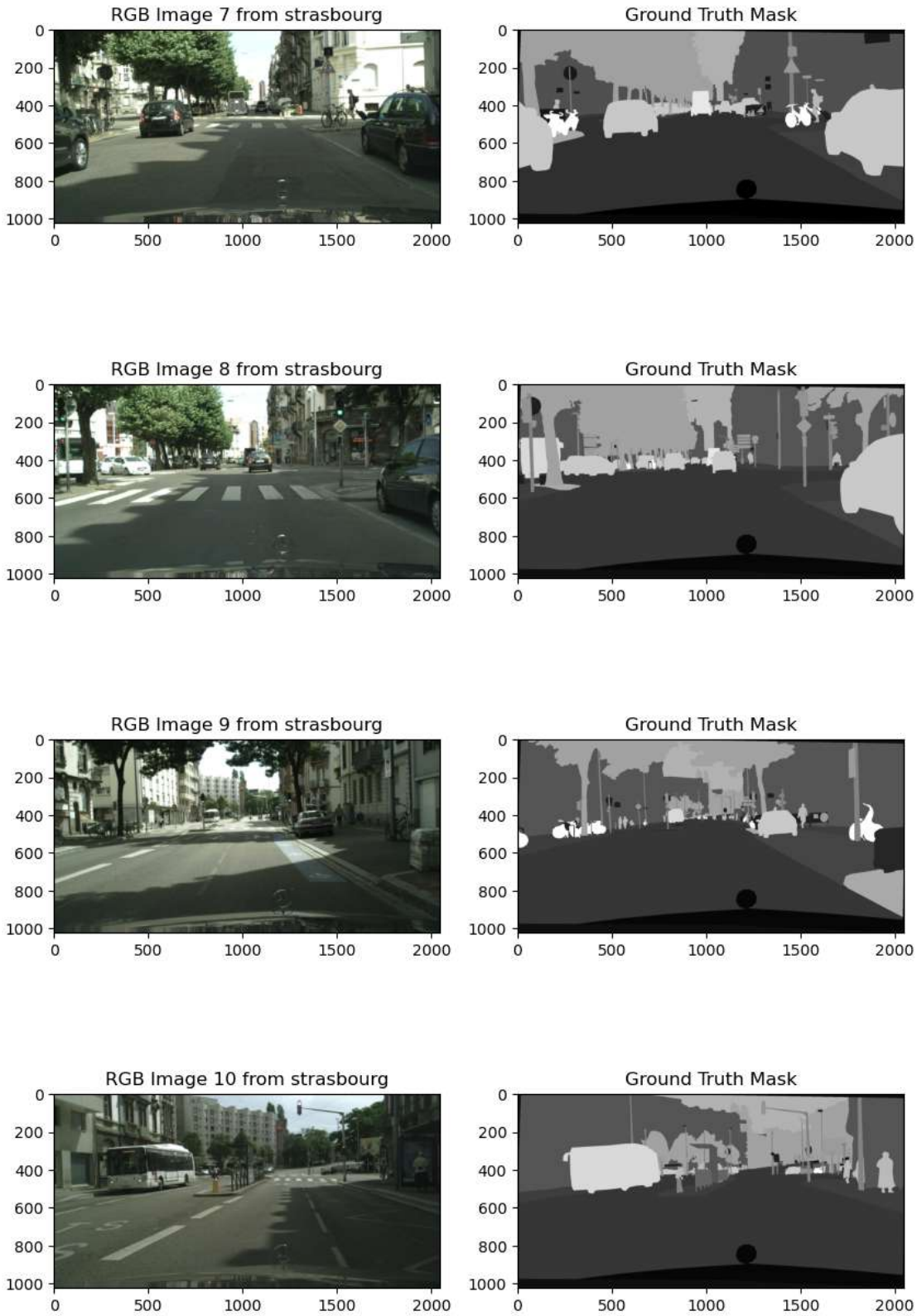



```
Finished processing city: monchengladbach

Processing city: strasbourg
```

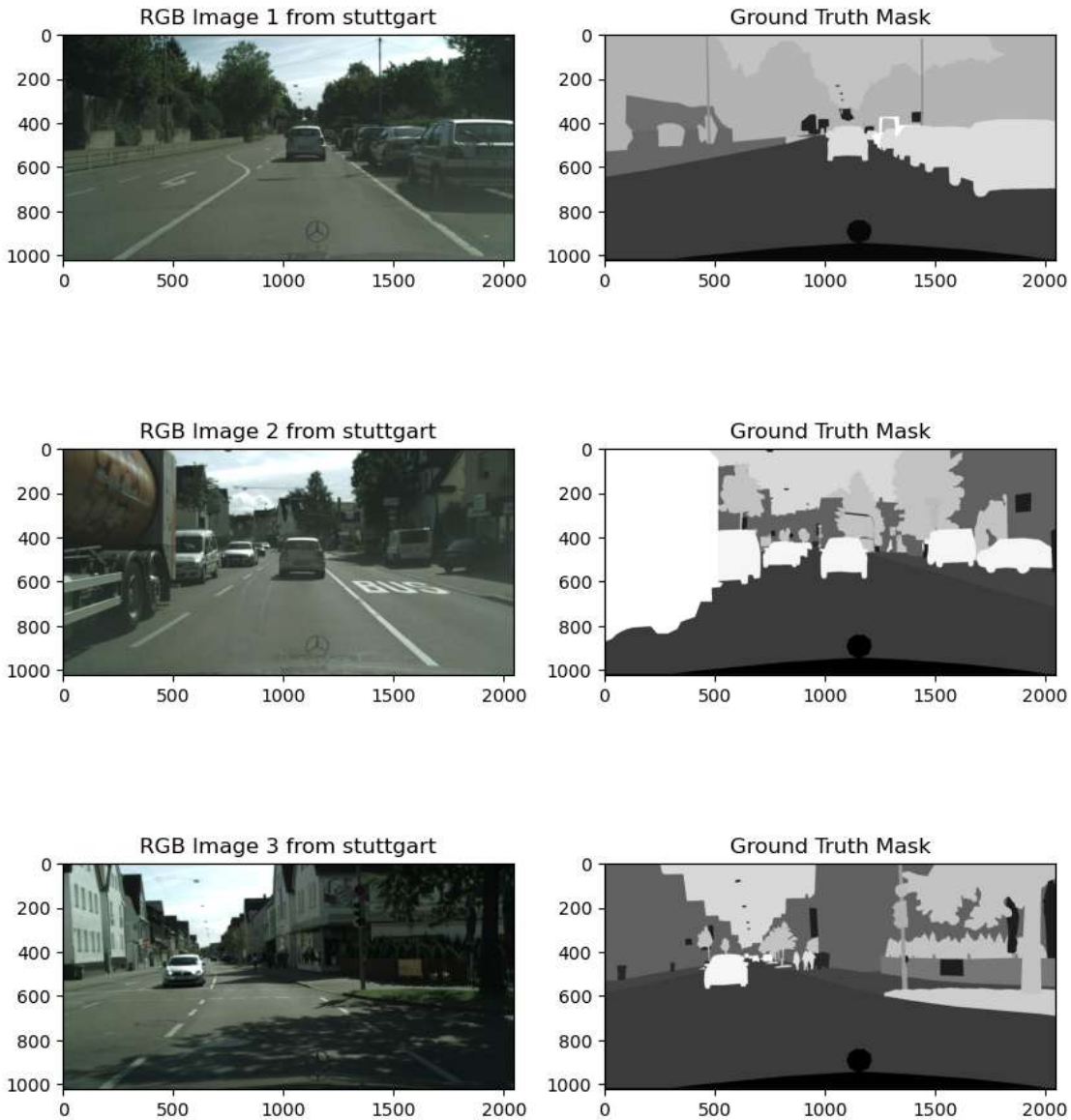


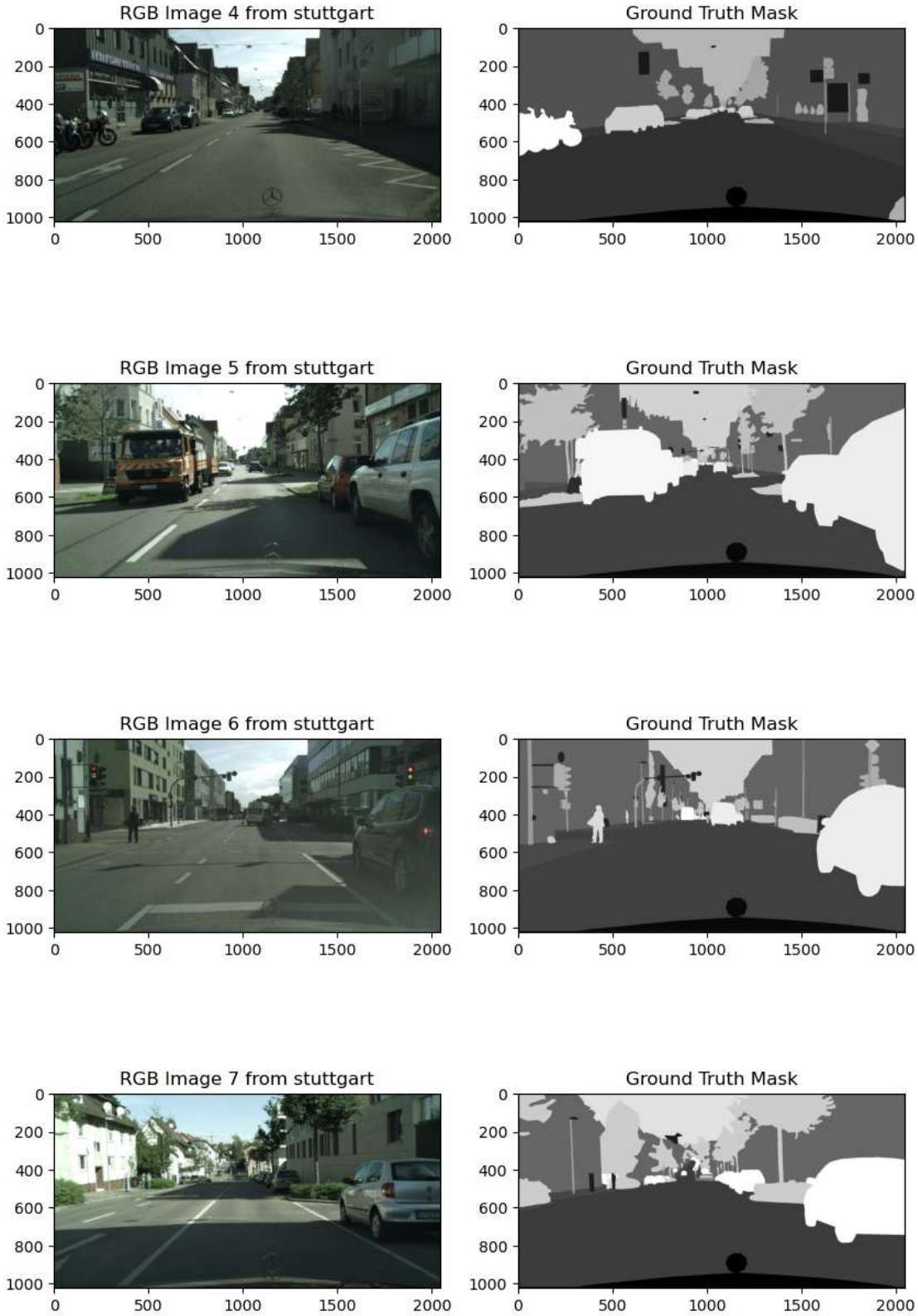


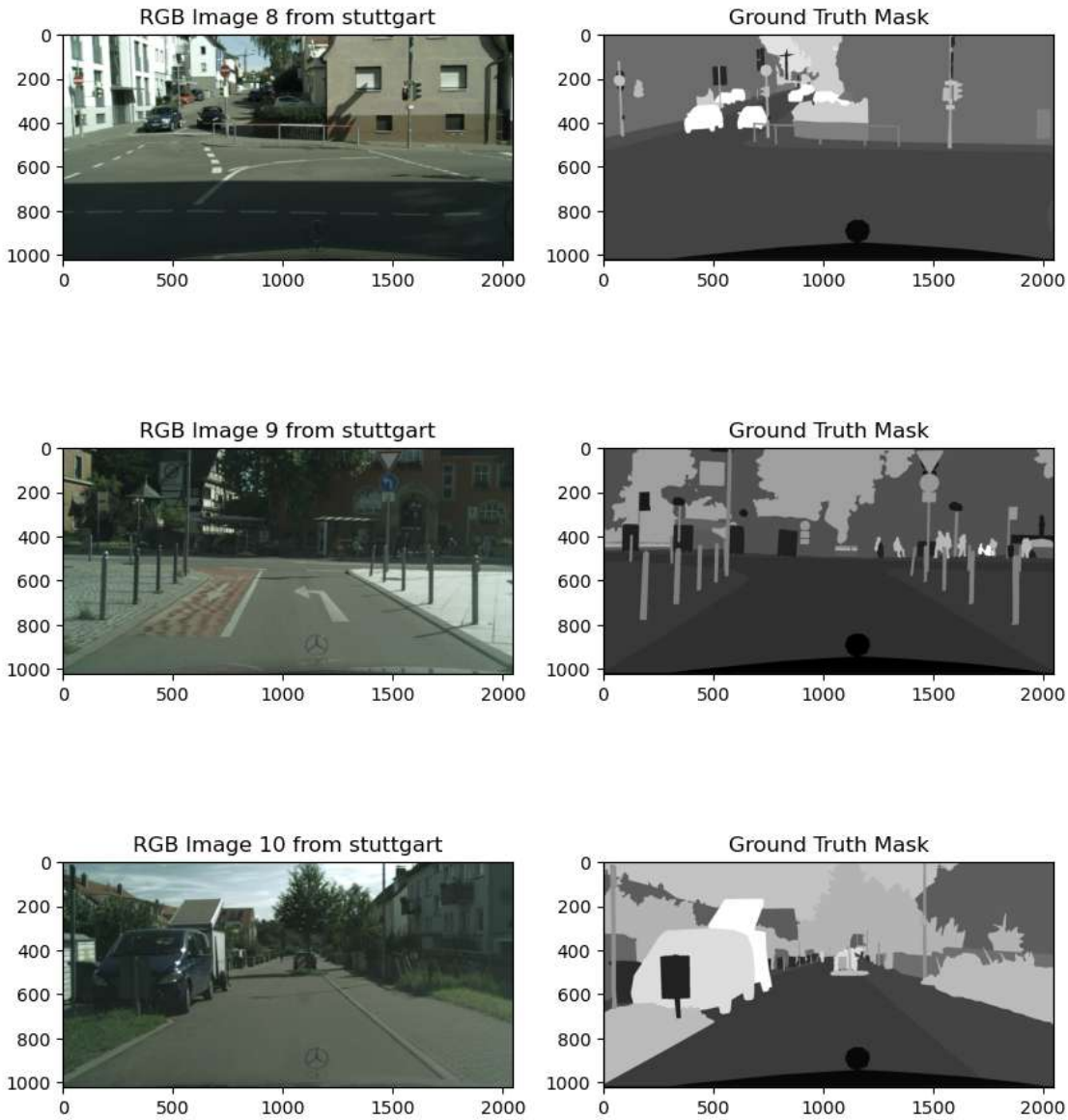

RGB Image 3 from strasbourg
Ground Truth Mask
RGB Image 4 from strasbourg
Ground Truth Mask
RGB Image 5 from strasbourg
Ground Truth Mask
RGB Image 6 from strasbourg
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 7 from strasbourg
Ground Truth Mask
RGB Image 8 from strasbourg
Ground Truth Mask
RGB Image 9 from strasbourg
Ground Truth Mask
RGB Image 10 from strasbourg
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

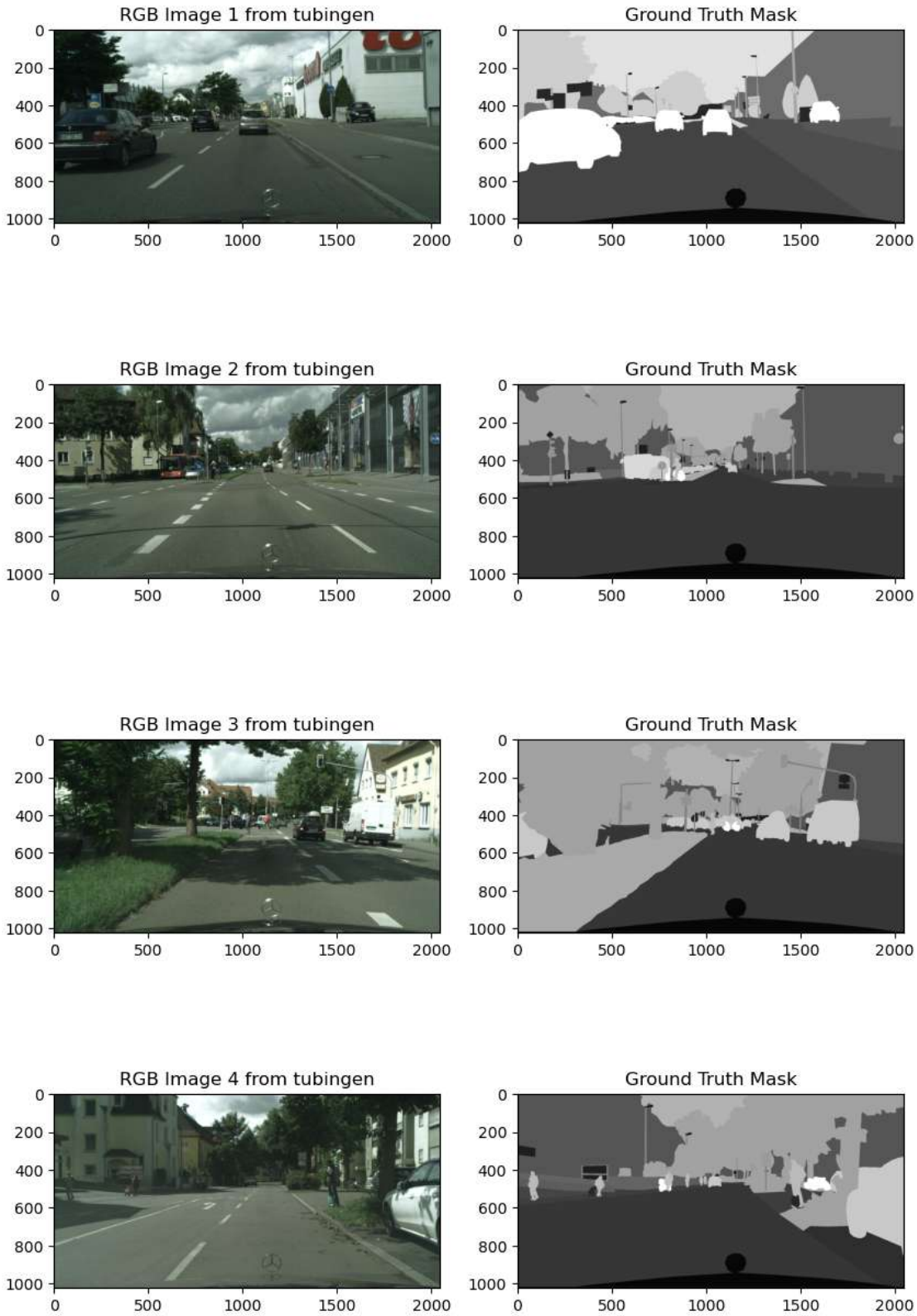
```
Finished processing city: strasbourg

Processing city: stuttgart
```

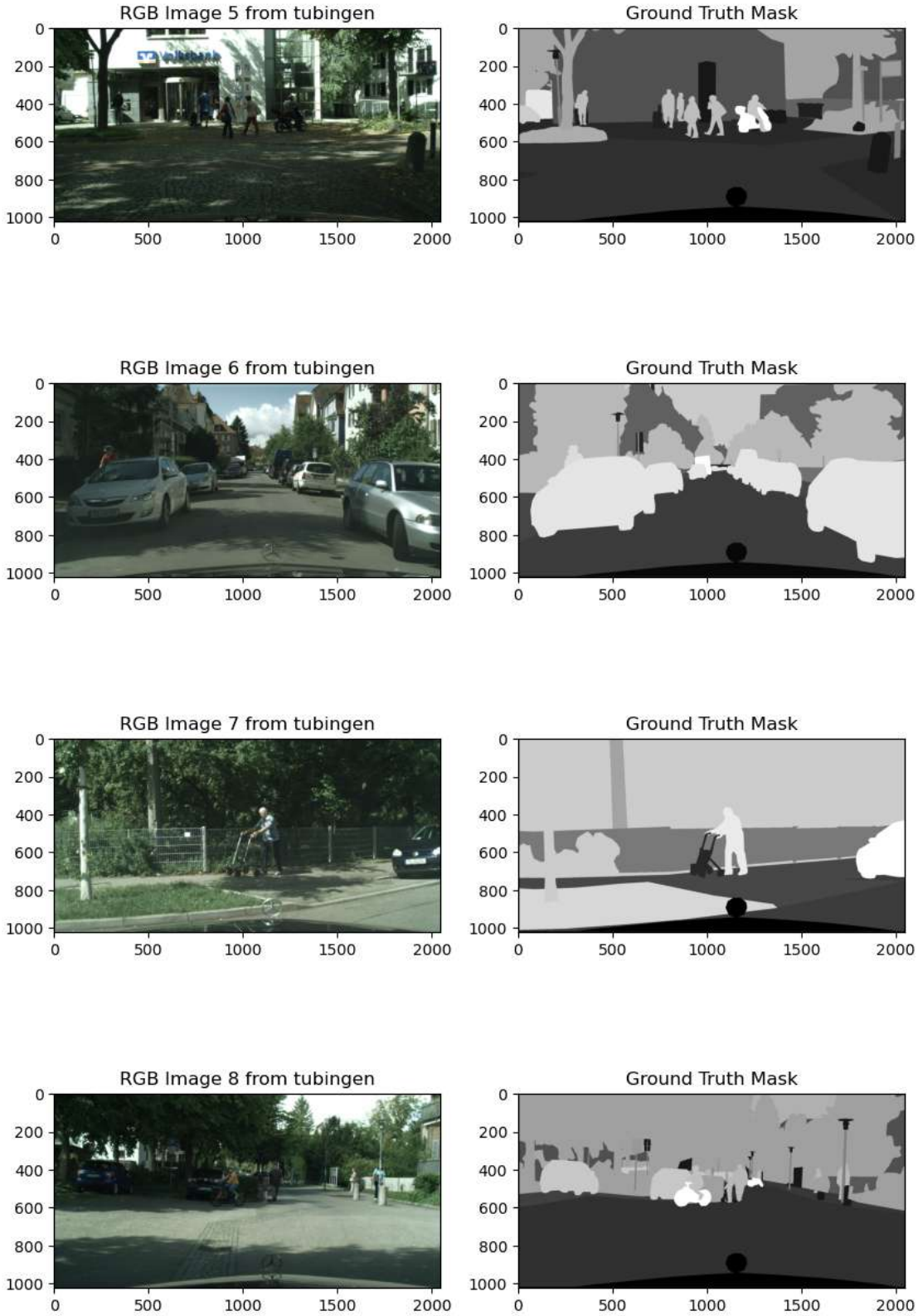


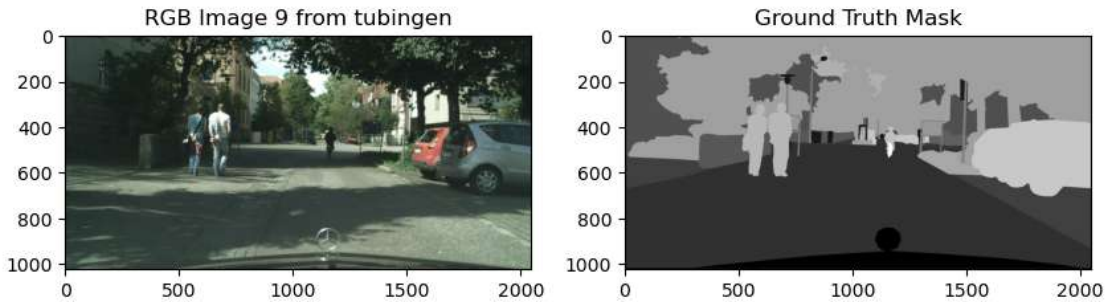

RGB Image 4 from stuttgart
Ground Truth Mask
RGB Image 5 from stuttgart
Ground Truth Mask
RGB Image 6 from stuttgart
Ground Truth Mask
RGB Image 7 from stuttgart
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

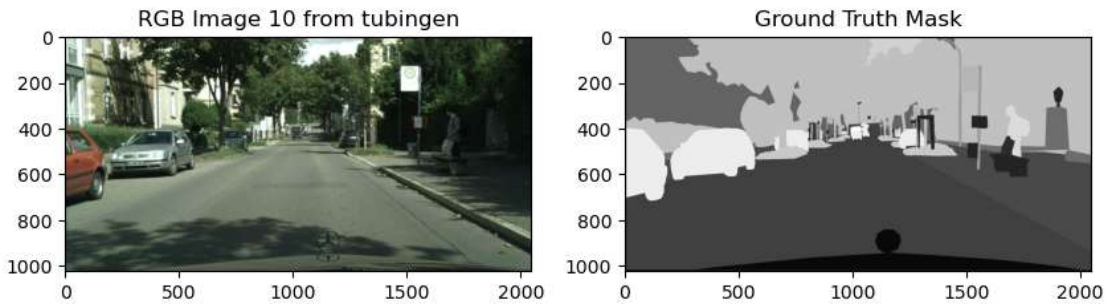



```
Finished processing city: stuttgart

Processing city: tubingen
```

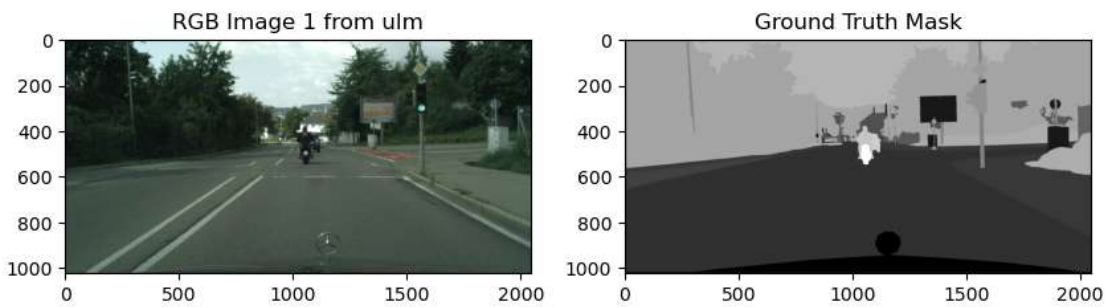

RGB Image 1 from tubingen
Ground Truth Mask
RGB Image 2 from tubingen
Ground Truth Mask
RGB Image 3 from tubingen
Ground Truth Mask
RGB Image 4 from tubingen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

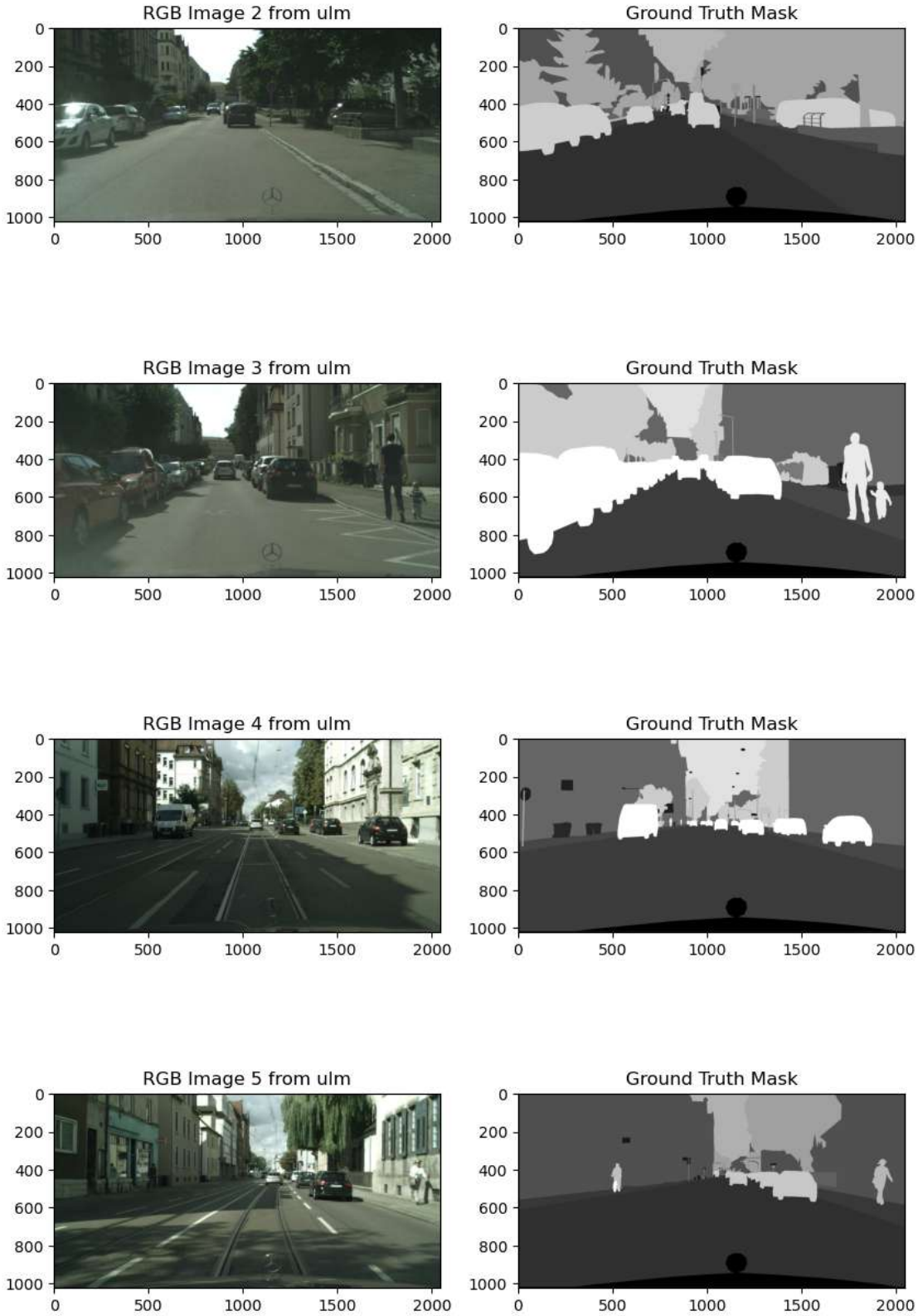

RGB Image 5 from tubingen
Ground Truth Mask
RGB Image 6 from tubingen
Ground Truth Mask
RGB Image 7 from tubingen
Ground Truth Mask
RGB Image 8 from tubingen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

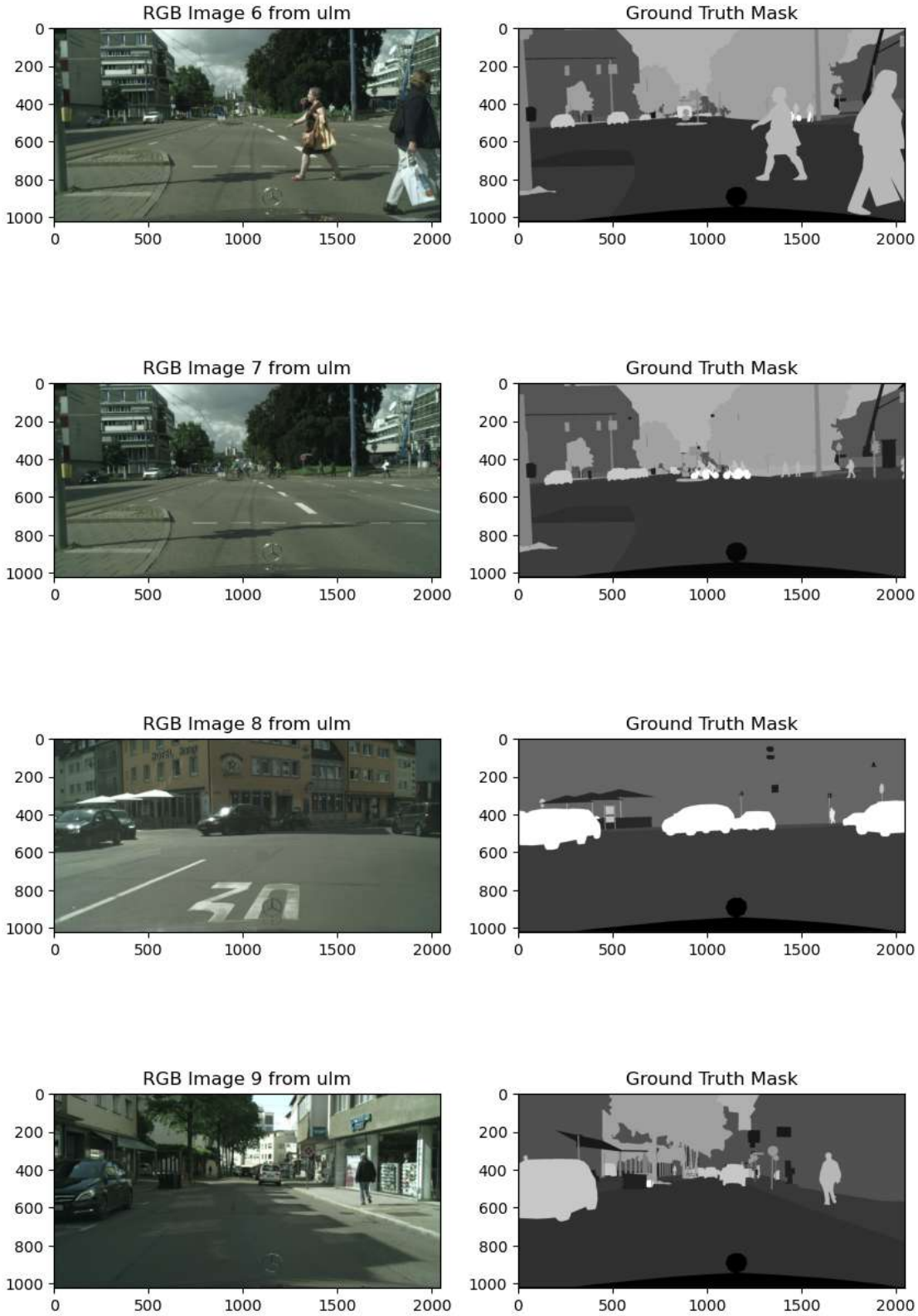


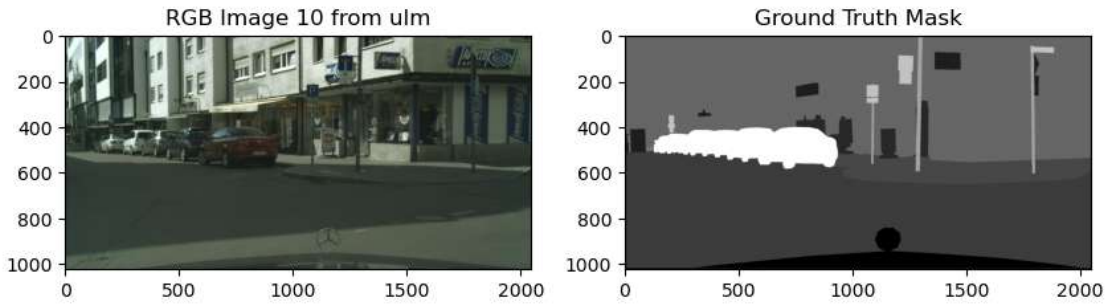



```
Finished processing city: tubingen

Processing city: ulm
```

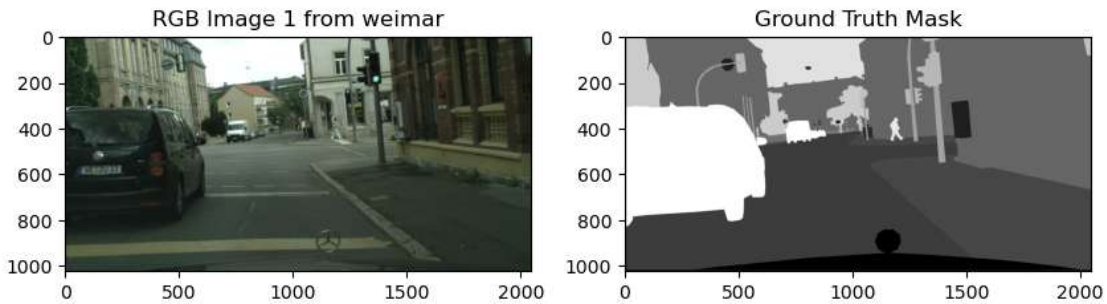


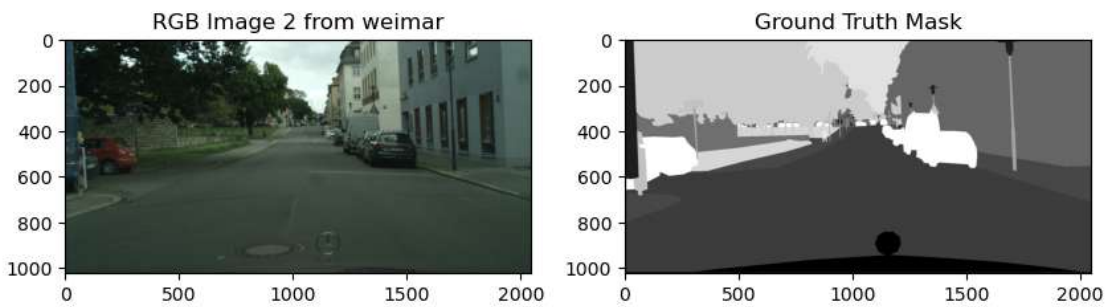

RGB Image 2 from ulm
Ground Truth Mask
RGB Image 3 from ulm
Ground Truth Mask
RGB Image 4 from ulm
Ground Truth Mask
RGB Image 5 from ulm
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

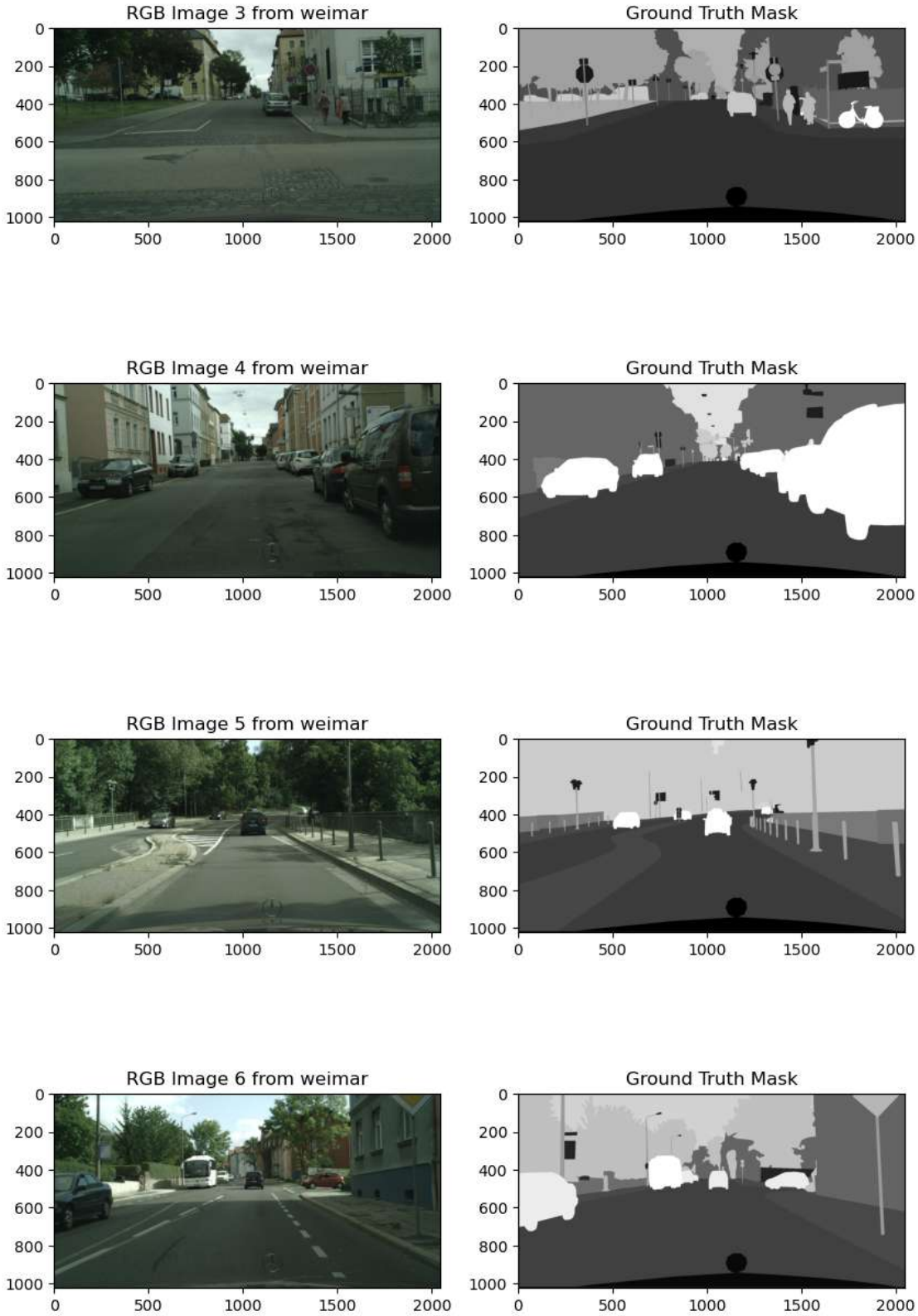

RGB Image 6 from ulm
Ground Truth Mask
RGB Image 7 from ulm
Ground Truth Mask
RGB Image 8 from ulm
Ground Truth Mask
RGB Image 9 from ulm
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

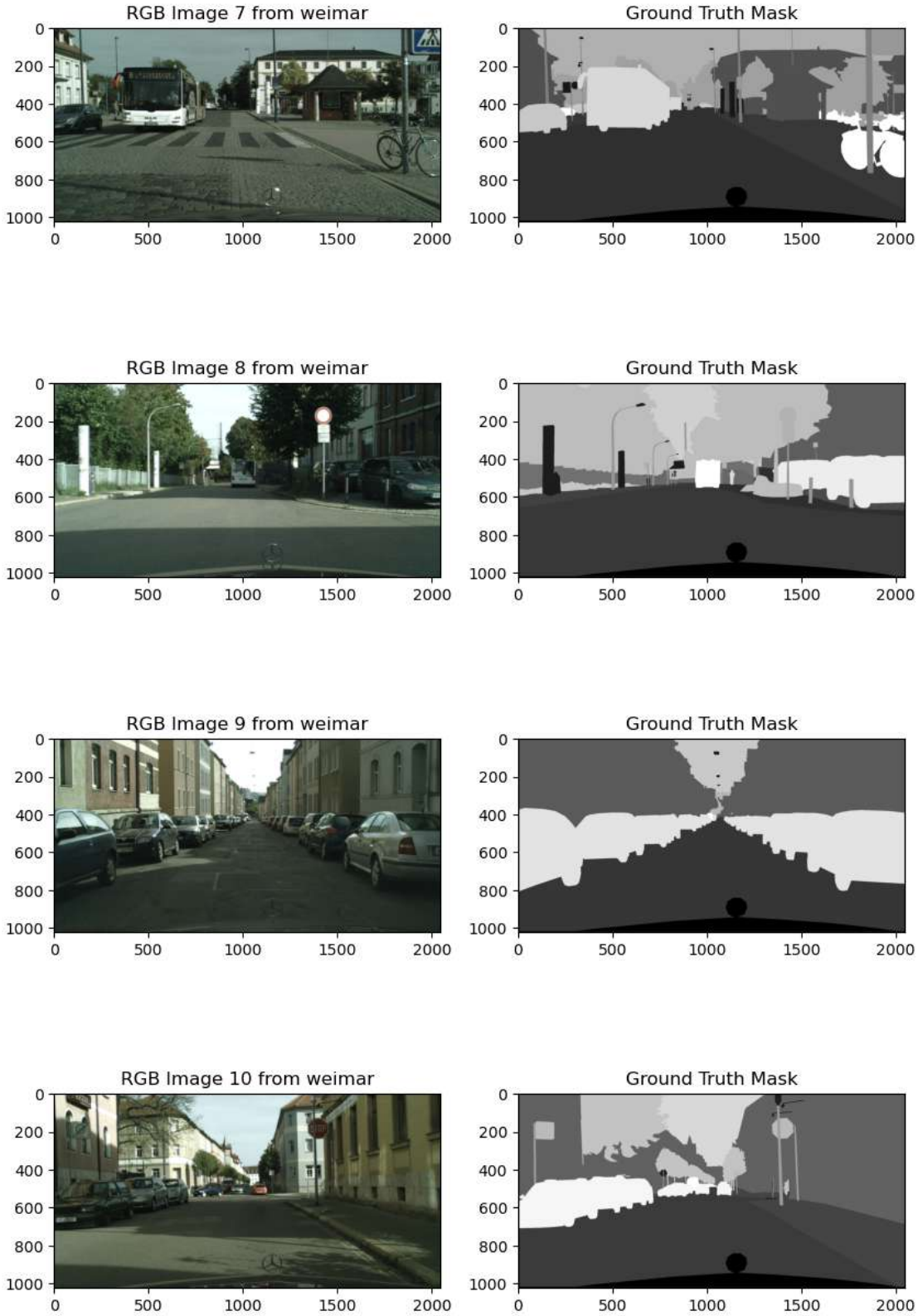



```
Finished processing city: ulm

Processing city: weimar
```

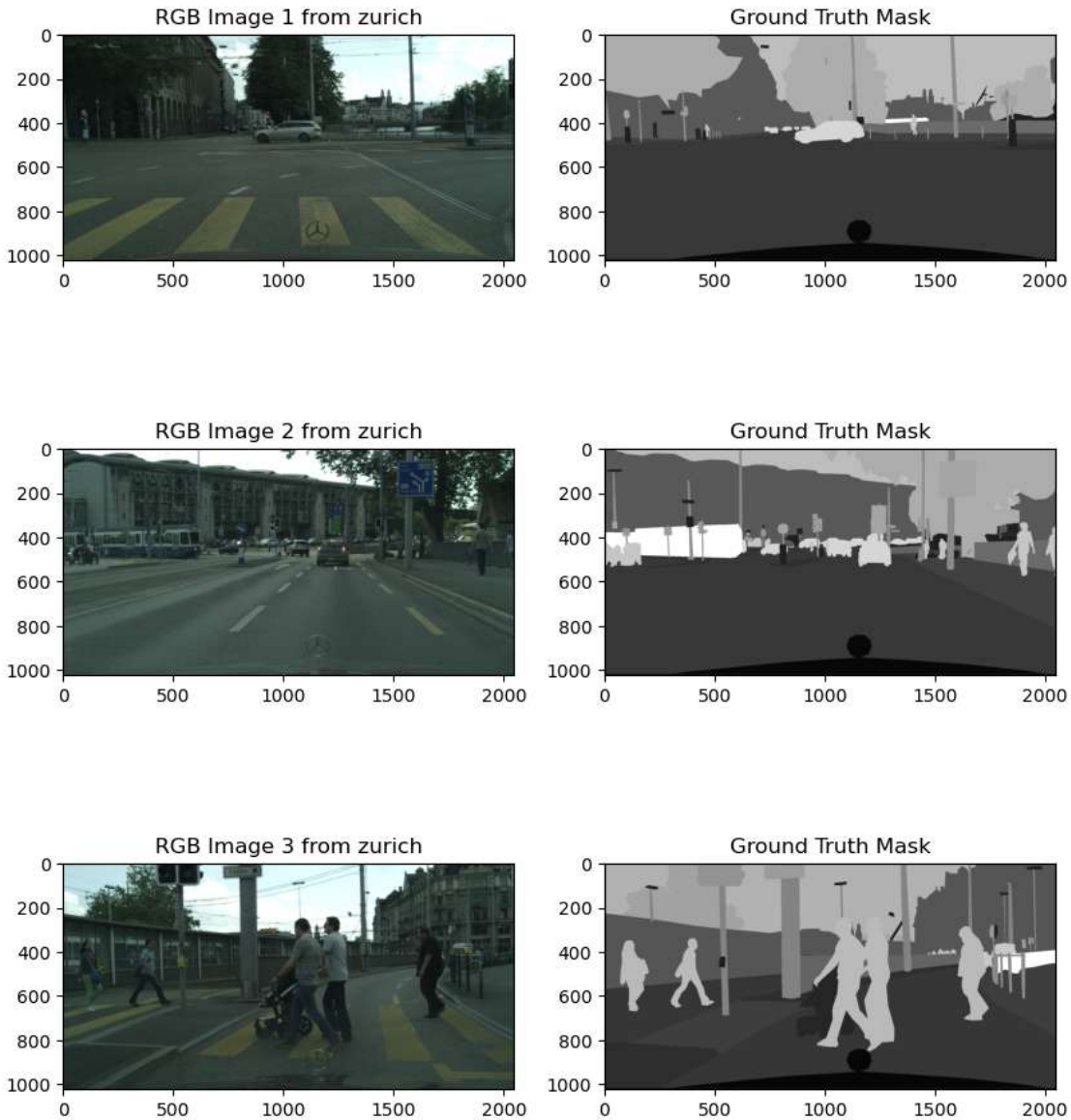


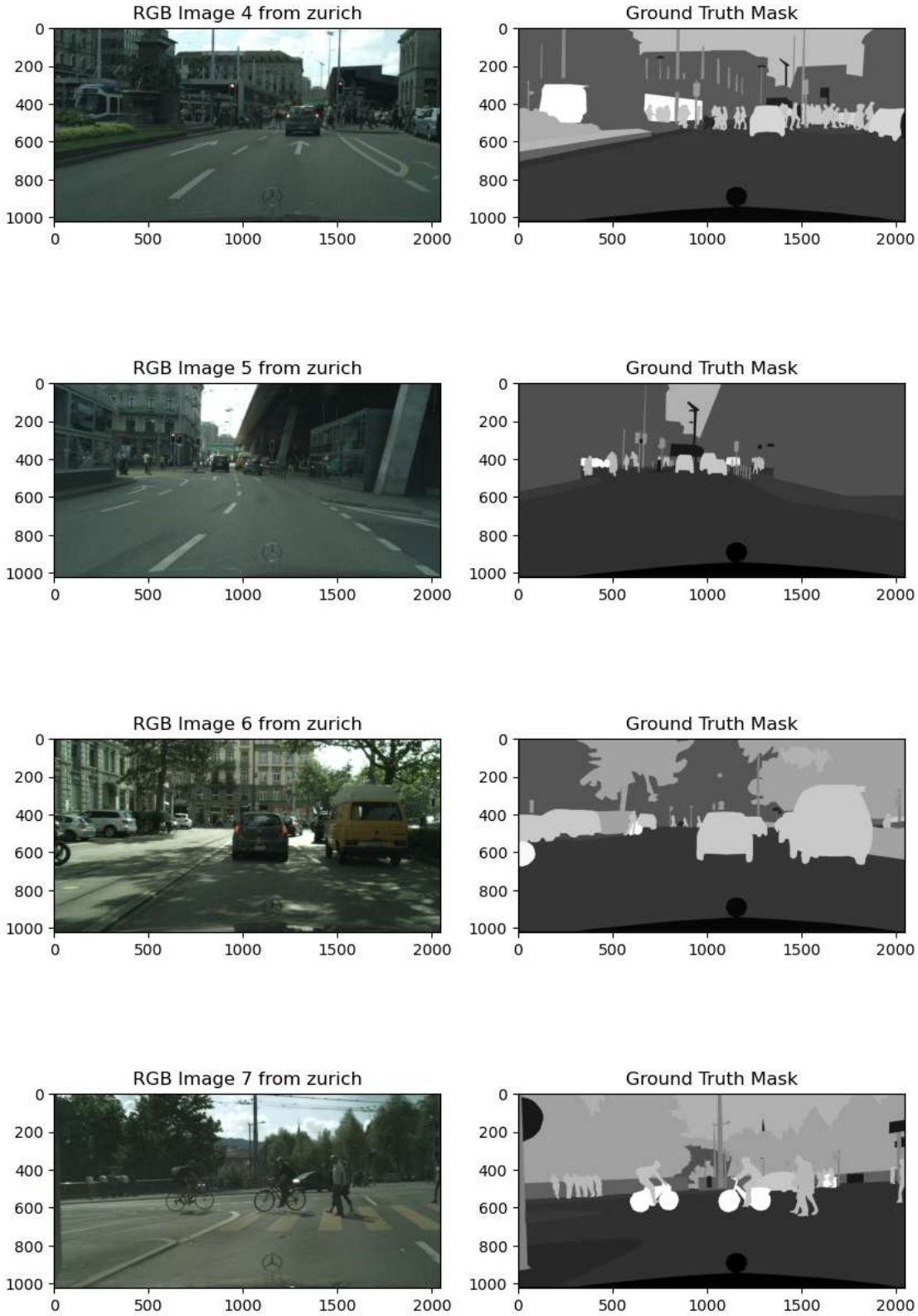


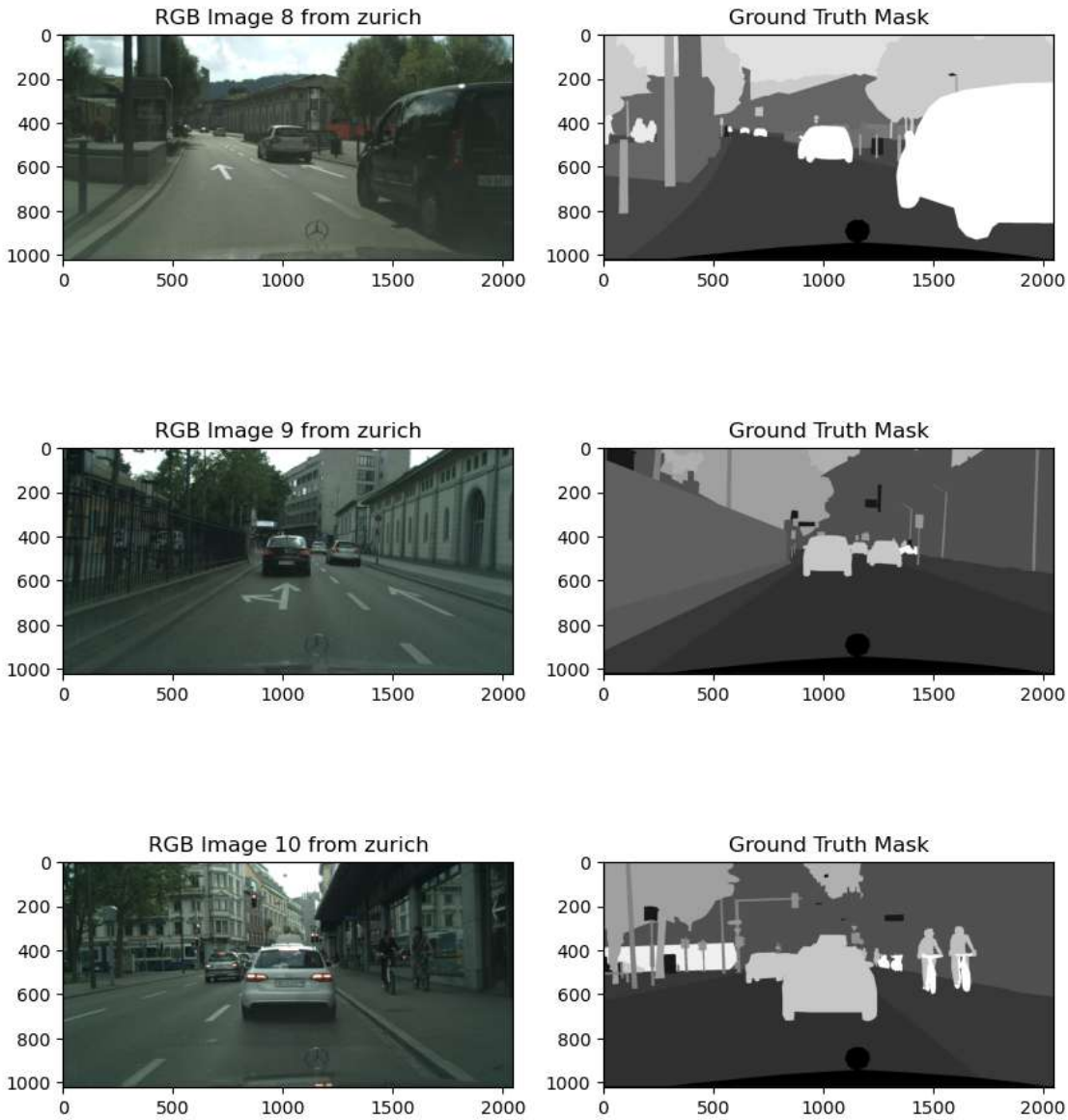

RGB Image 3 from weimar
Ground Truth Mask
RGB Image 4 from weimar
Ground Truth Mask
RGB Image 5 from weimar
Ground Truth Mask
RGB Image 6 from weimar
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 7 from weimar
Ground Truth Mask
RGB Image 8 from weimar
Ground Truth Mask
RGB Image 9 from weimar
Ground Truth Mask
RGB Image 10 from weimar
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

```
Finished processing city: weimar

Processing city: zurich
```




RGB Image 4 from zurich
Ground Truth Mask
RGB Image 5 from zurich
Ground Truth Mask
RGB Image 6 from zurich
Ground Truth Mask
RGB Image 7 from zurich
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

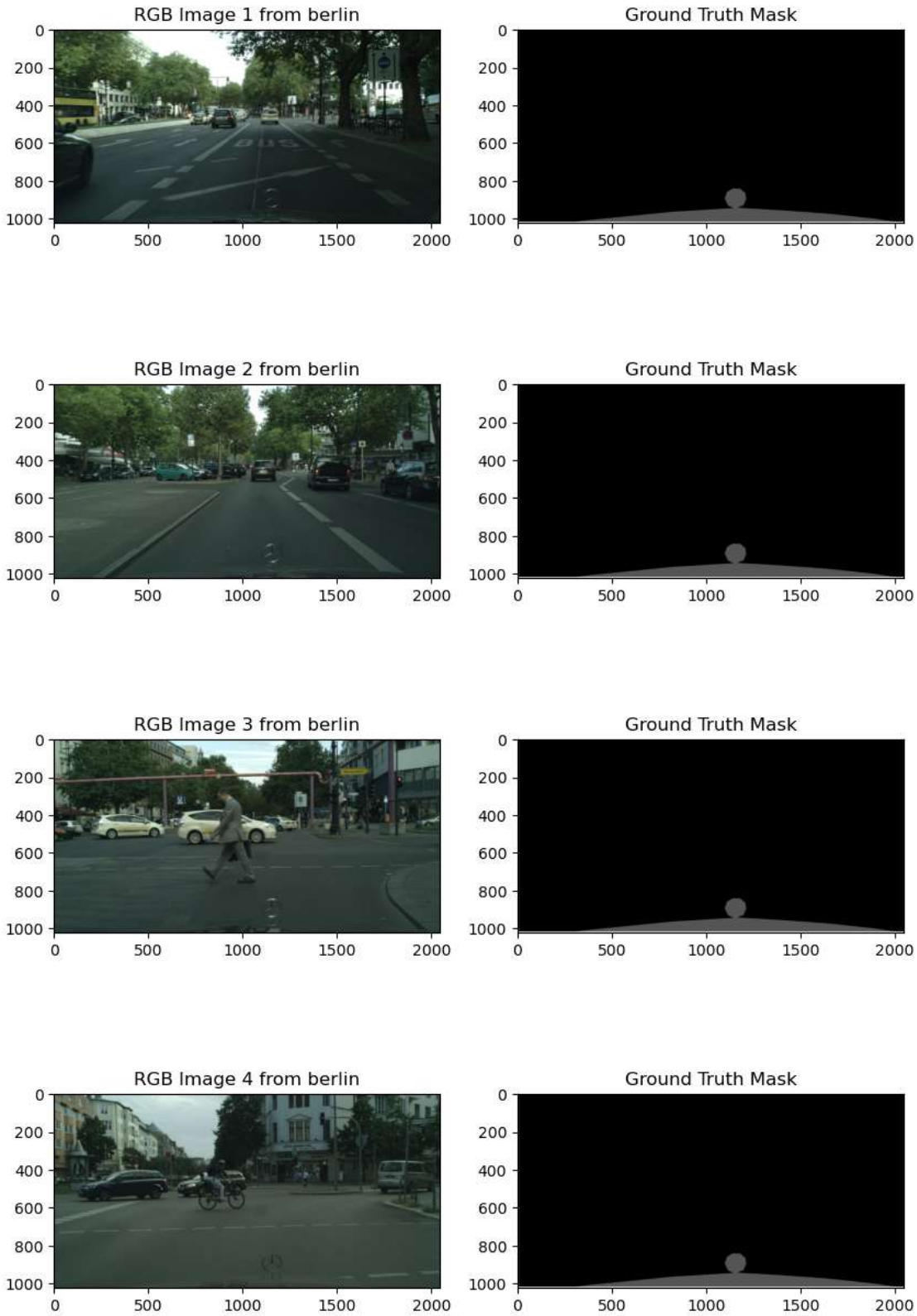



```
Finished processing city: zurich
```

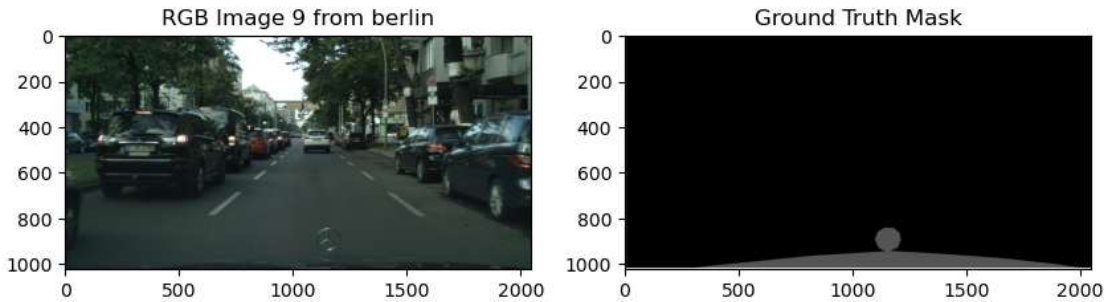
[3]:
```python
import os
from PIL import Image
import matplotlib.pyplot as plt

# Define paths to the datasets
img_path = r'D:
 ↪\infosys_springboard_project\leftImg8bit_trainvaltest\leftImg8bit\test'
gt_path = r'D:\infosys_springboard_project\gtFine_trainvaltest\gtFine\test'
```

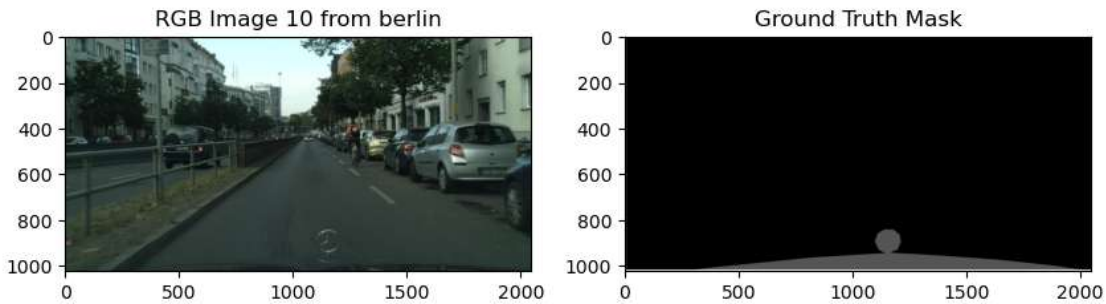
```python
# Get the list of cities (assuming cityscapes-style structure)
cities = os.listdir(img_path)

# Loop through all the cities in the dataset
for city in cities:
    print(f"Processing city: {city}")

    # Get the list of image files in the city folder
    img_files = os.listdir(os.path.join(img_path, city))

    # Initialize a counter for the number of images displayed
    displayed_images = 0

    # Loop through the images and find the corresponding ground truth files
    for img_file in img_files:
        if displayed_images >= 10:  # Only display 10 images per city
            break

        # Load the RGB image
        img = Image.open(os.path.join(img_path, city, img_file))

        # Load the corresponding ground truth file (gtFine has a different␣
↪naming convention)
        gt_file = img_file.replace('_leftImg8bit.png', '_gtFine_labelIds.png')
        gt_path_full = os.path.join(gt_path, city, gt_file)

        # Check if the ground truth file exists before loading
        if os.path.exists(gt_path_full):
            gt = Image.open(gt_path_full)

            # Display the image and the ground truth side by side
            fig, ax = plt.subplots(1, 2, figsize=(10, 5))
            ax[0].imshow(img)
            ax[0].set_title(f'RGB Image {displayed_images + 1} from {city}')
            ax[1].imshow(gt, cmap='gray')
            ax[1].set_title('Ground Truth Mask')
            plt.show()

            displayed_images += 1  # Increment the counter after displaying an␣
↪image
        else:
            print(f"Ground truth file not found for: {img_file}")

    print(f"Finished processing city: {city}\n")
```

```
Processing city: berlin
```

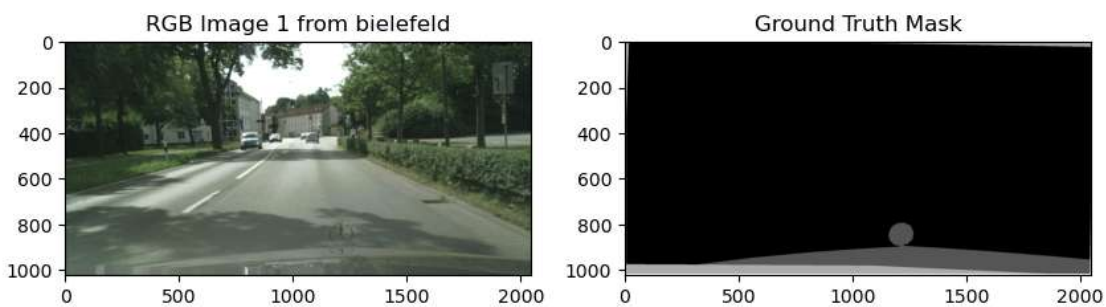

RGB Image 1 from berlin
Ground Truth Mask
RGB Image 2 from berlin
Ground Truth Mask
RGB Image 3 from berlin
Ground Truth Mask
RGB Image 4 from berlin
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

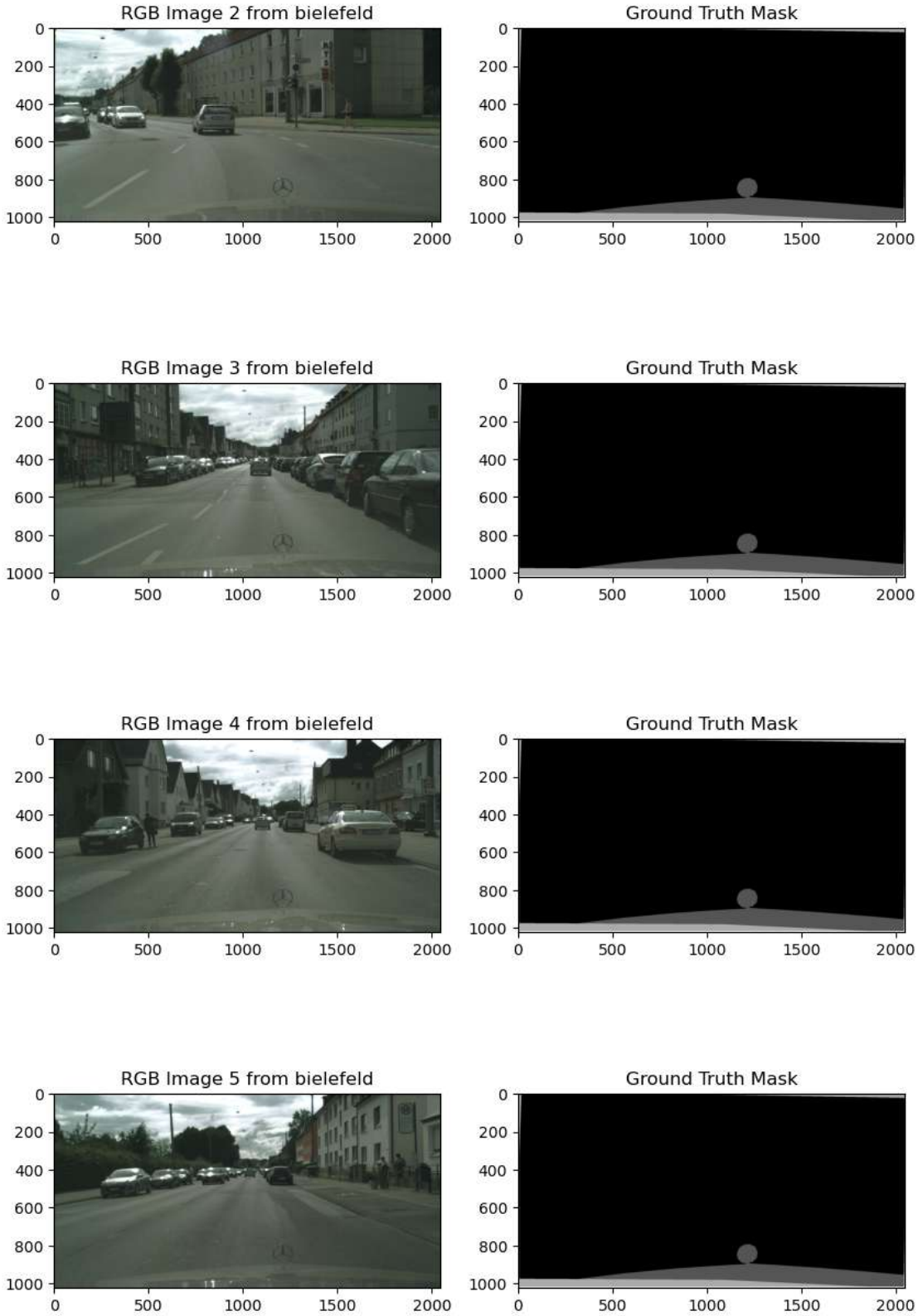
RGB Image 5 from berlin
Ground Truth Mask
RGB Image 6 from berlin
Ground Truth Mask
RGB Image 7 from berlin
Ground Truth Mask
RGB Image 8 from berlin
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

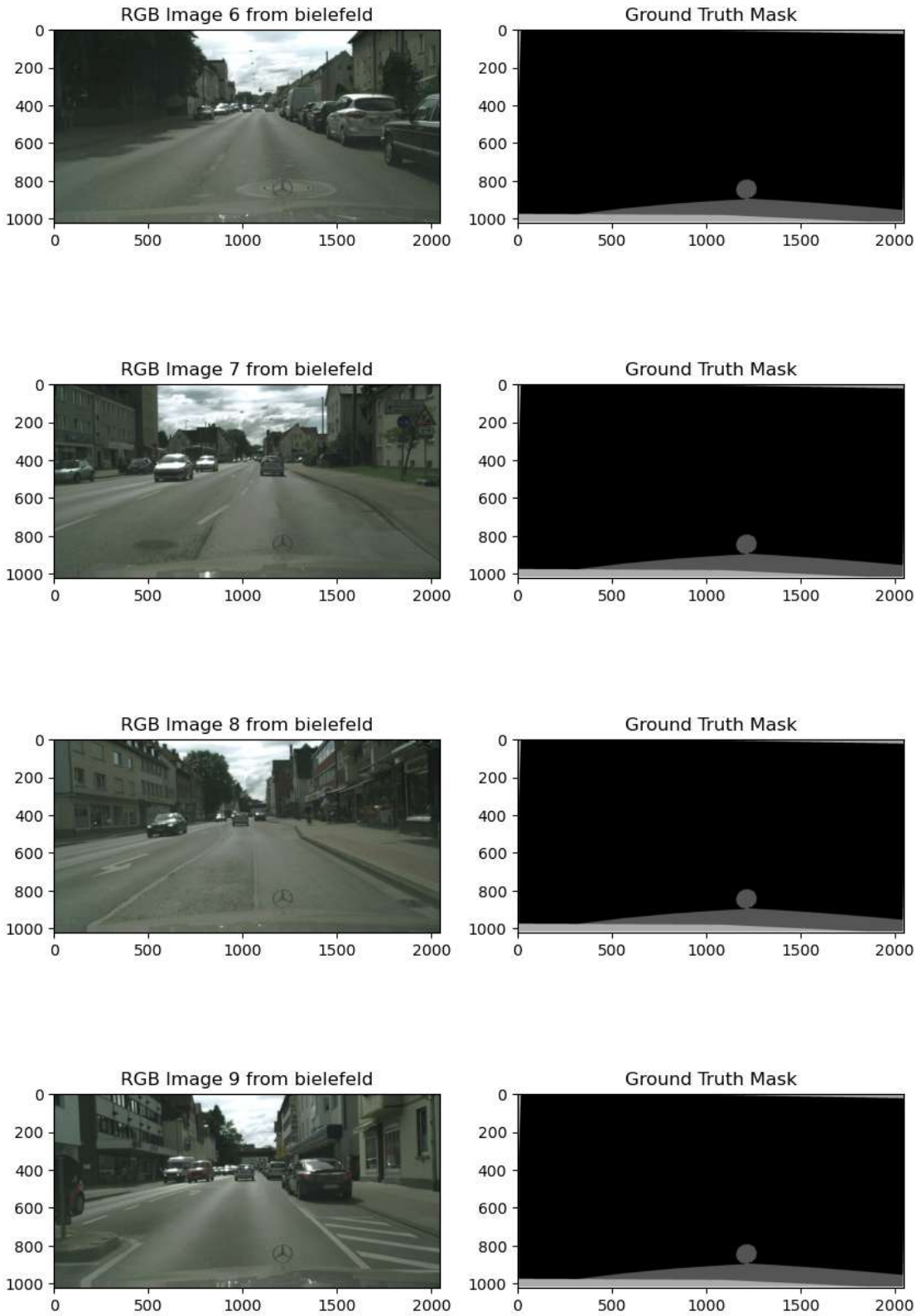


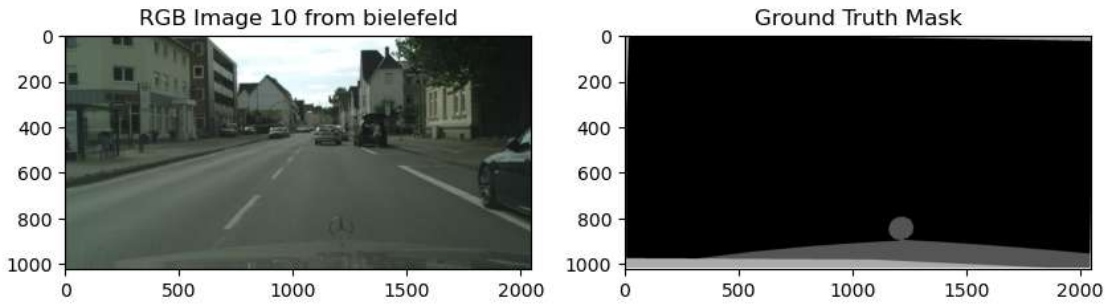



```
Finished processing city: berlin

Processing city: bielefeld
```

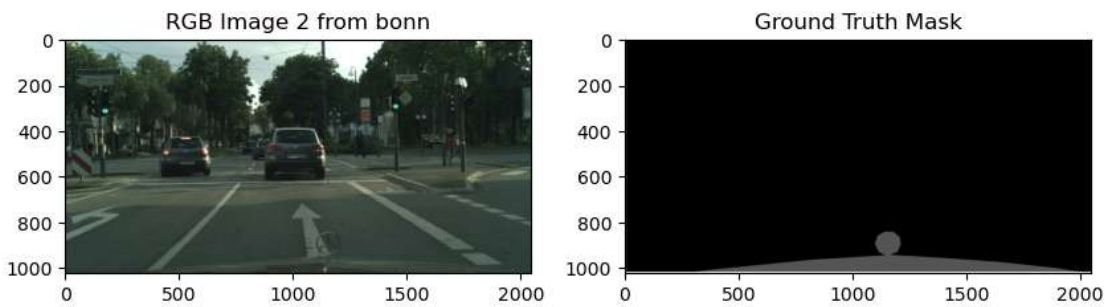



RGB Image 2 from bielefeld
Ground Truth Mask
RGB Image 3 from bielefeld
Ground Truth Mask
RGB Image 4 from bielefeld
Ground Truth Mask
RGB Image 5 from bielefeld
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

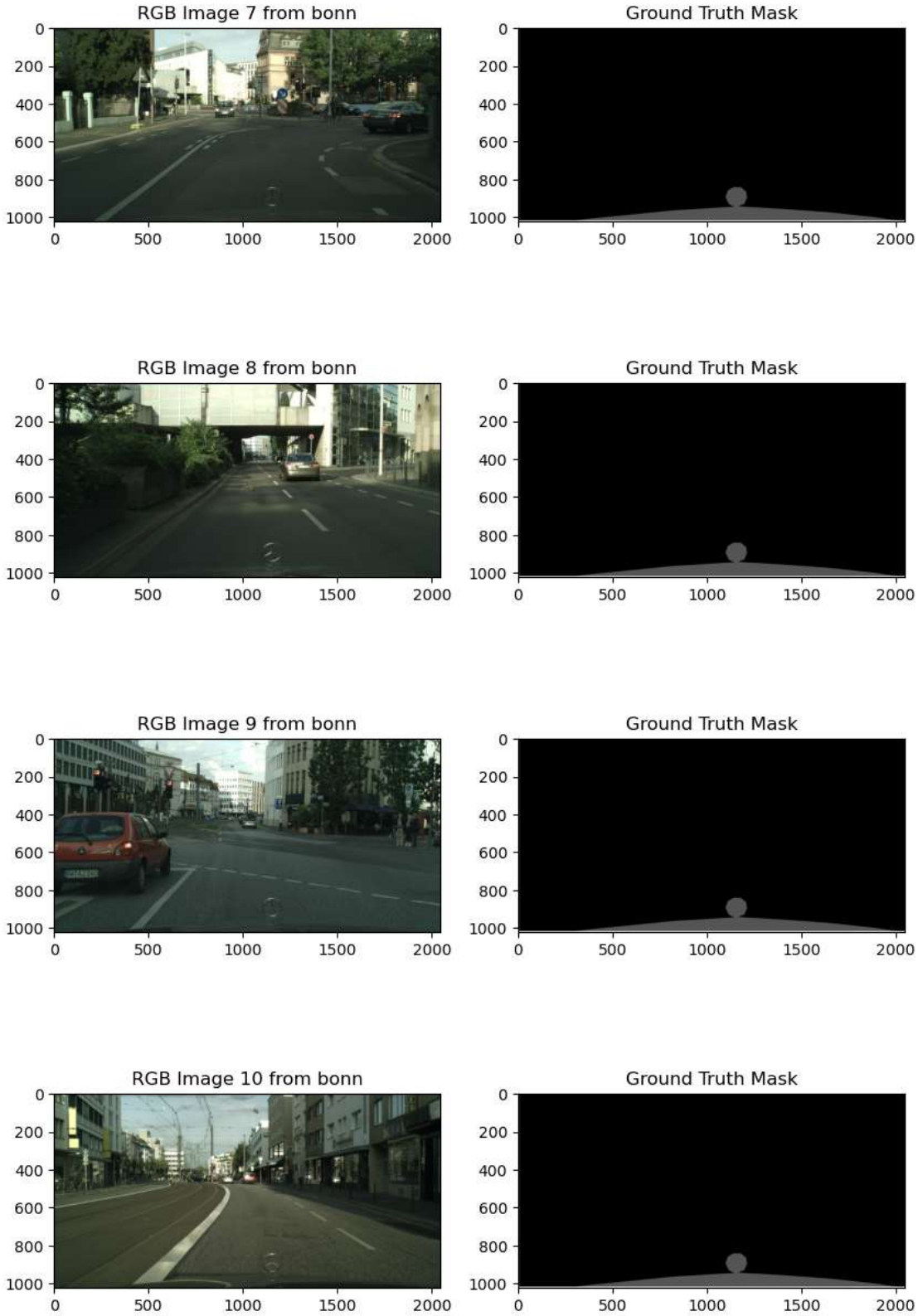

RGB Image 6 from bielefeld
Ground Truth Mask
RGB Image 7 from bielefeld
Ground Truth Mask
RGB Image 8 from bielefeld
Ground Truth Mask
RGB Image 9 from bielefeld
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000



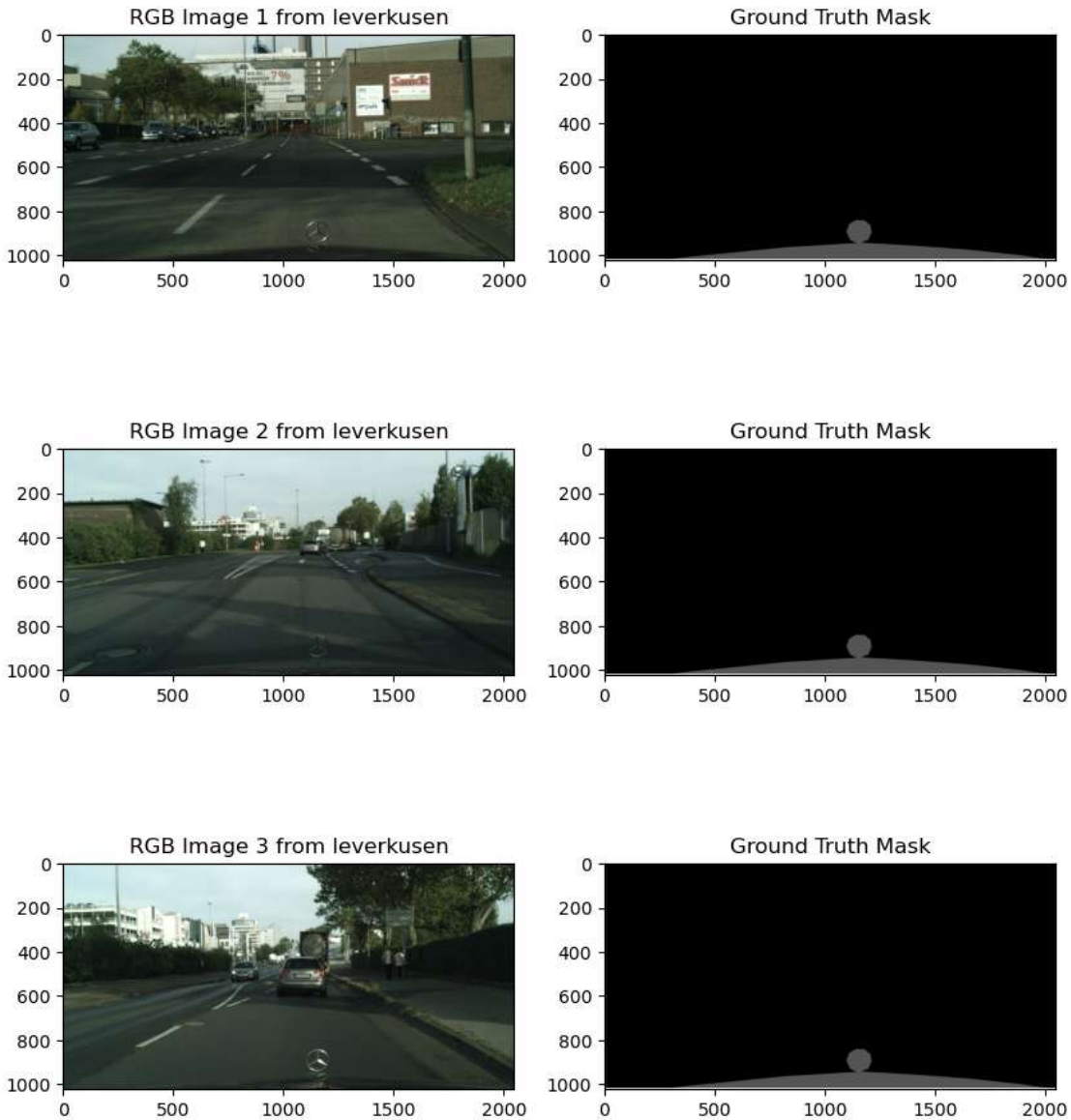

```
Finished processing city: bielefeld

Processing city: bonn
```

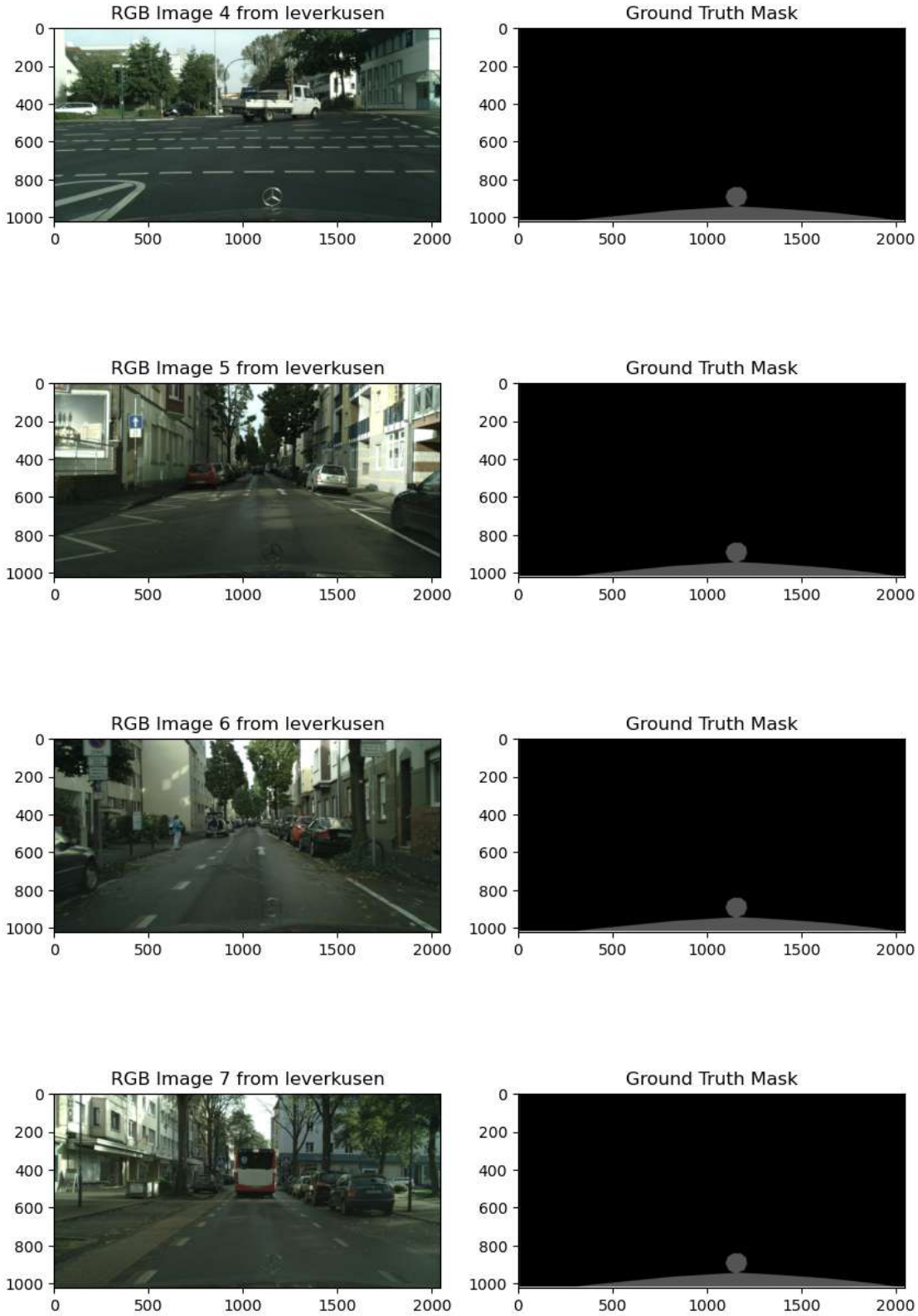


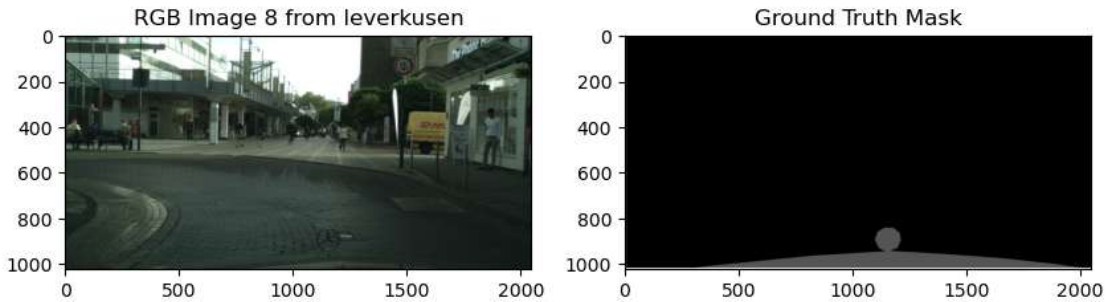
RGB Image 3 from bonn
Ground Truth Mask
RGB Image 4 from bonn
Ground Truth Mask
RGB Image 5 from bonn
Ground Truth Mask
RGB Image 6 from bonn
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

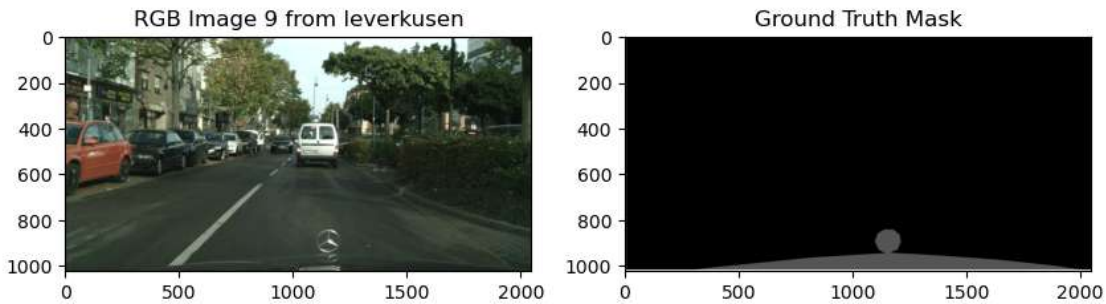

RGB Image 7 from bonn
Ground Truth Mask
RGB Image 8 from bonn
Ground Truth Mask
RGB Image 9 from bonn
Ground Truth Mask
RGB Image 10 from bonn
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

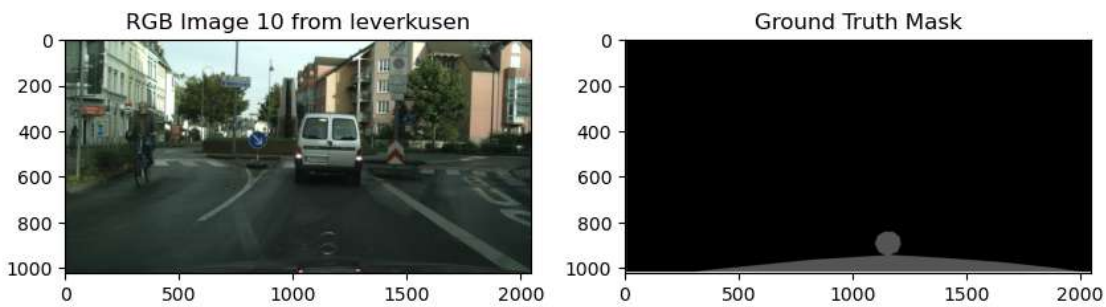
```
Finished processing city: bonn

Processing city: leverkusen
```



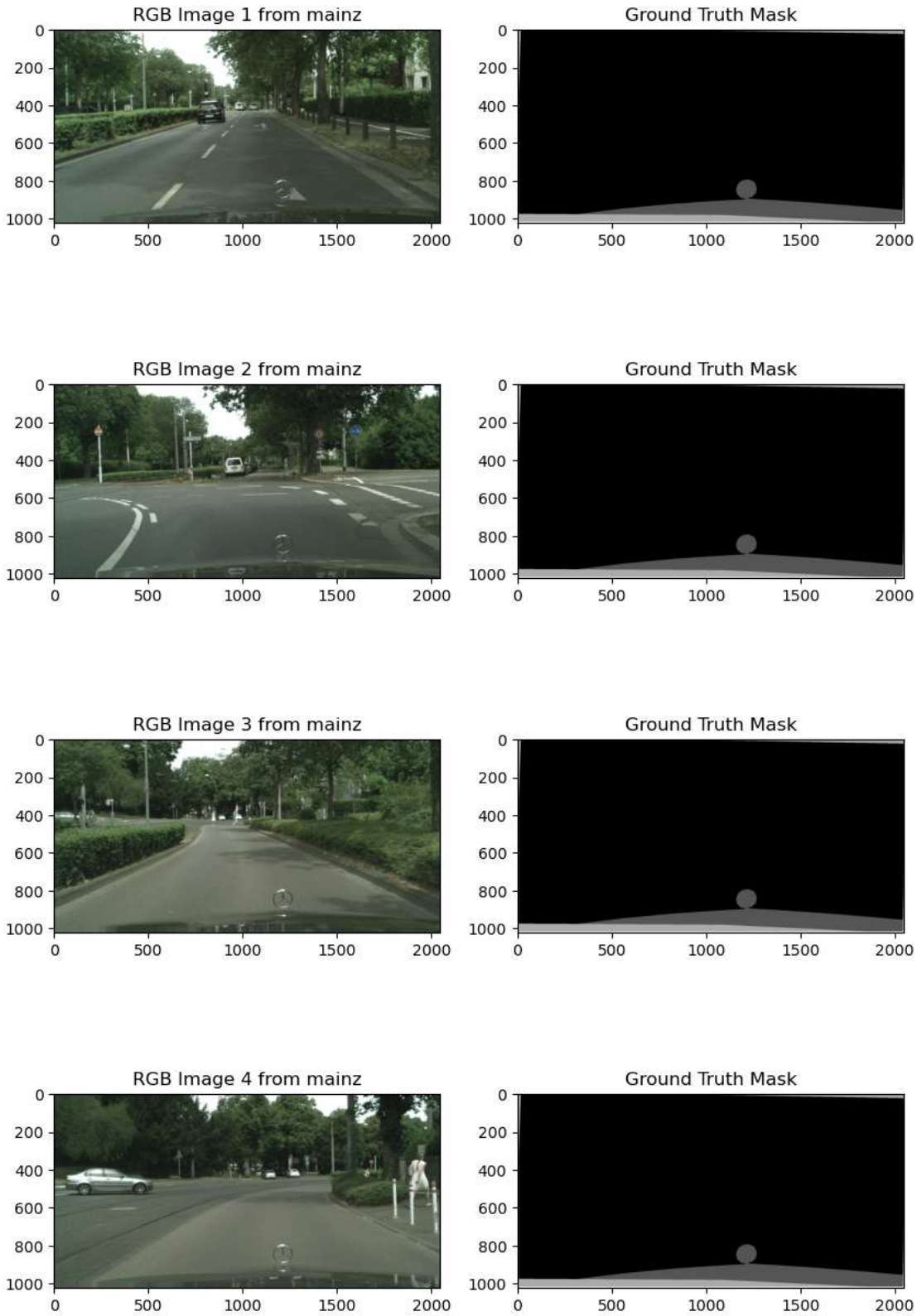

RGB Image 4 from leverkusen
Ground Truth Mask
RGB Image 5 from leverkusen
Ground Truth Mask
RGB Image 6 from leverkusen
Ground Truth Mask
RGB Image 7 from leverkusen
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

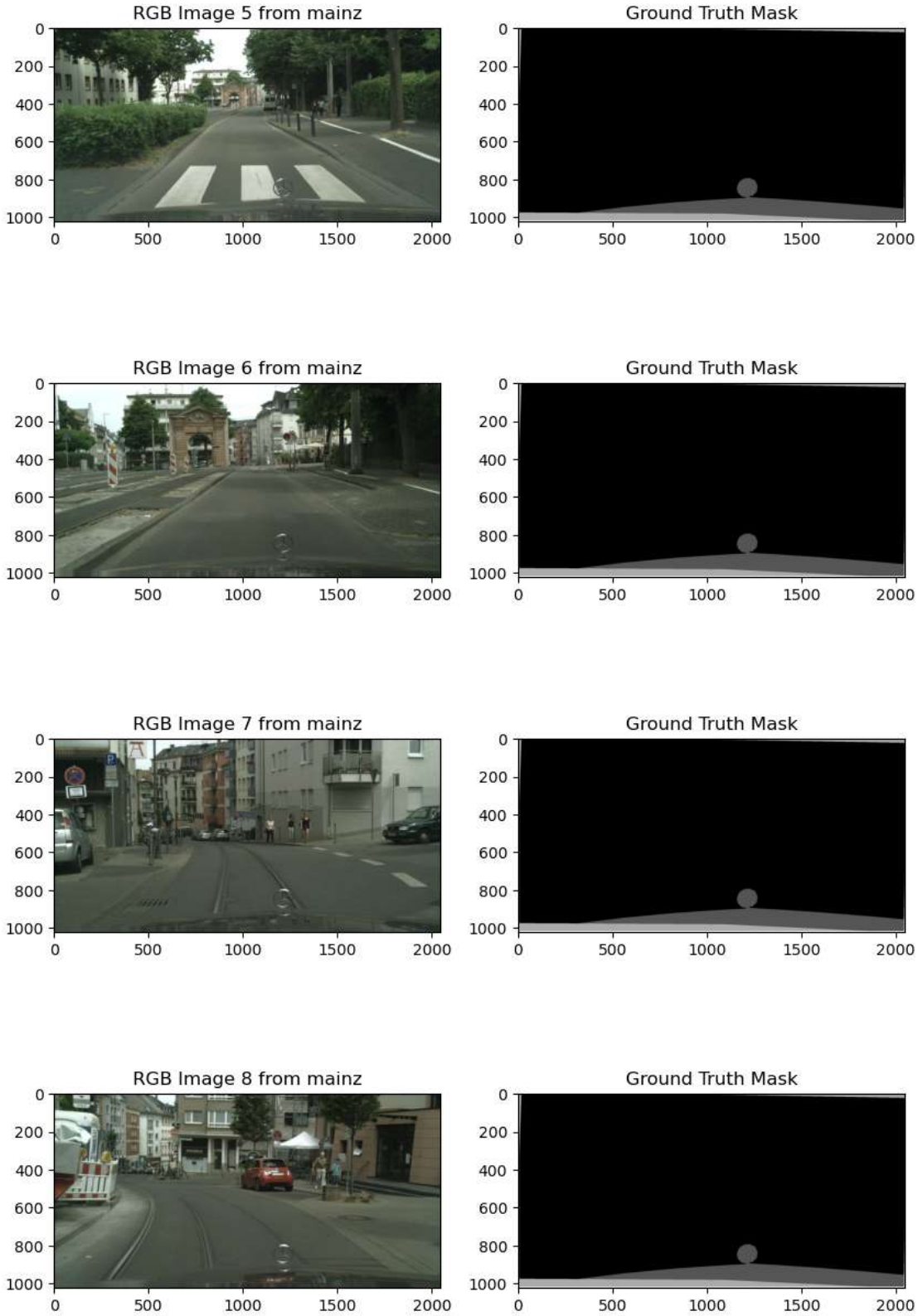


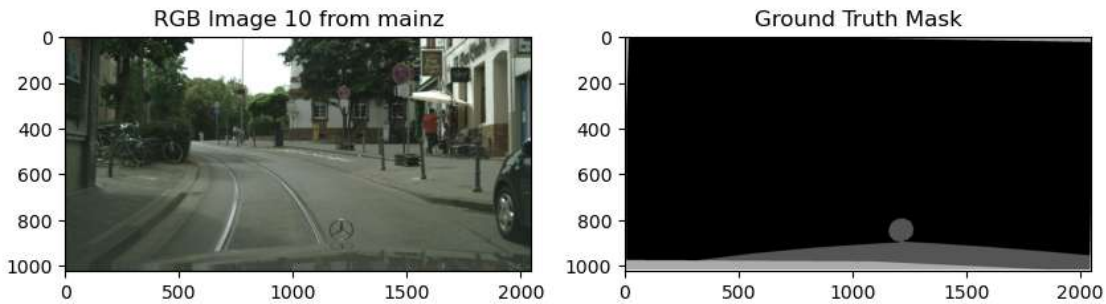




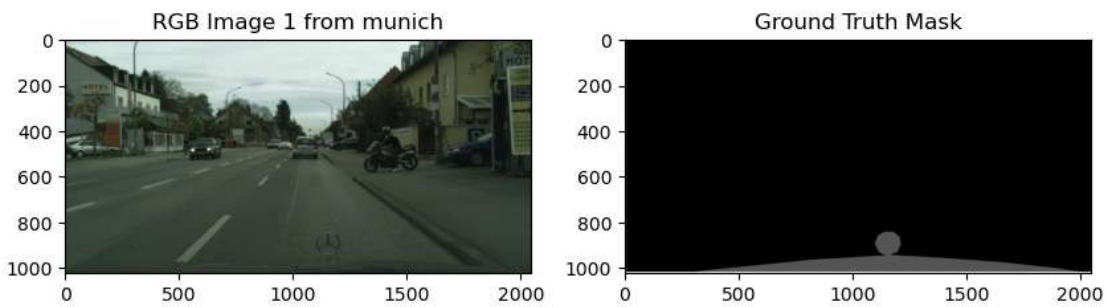

```
Finished processing city: leverkusen

Processing city: mainz
```

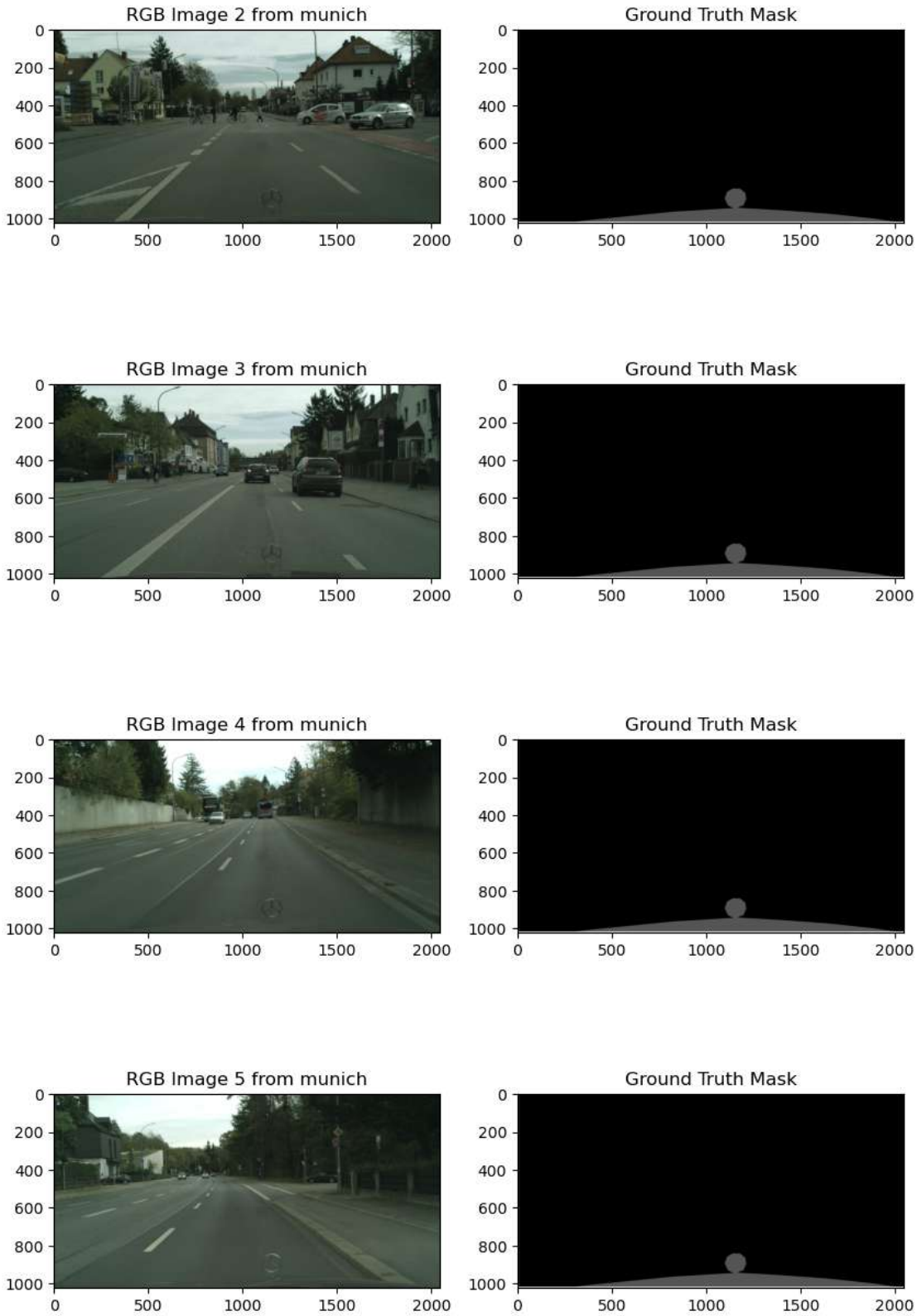

RGB Image 1 from mainz
Ground Truth Mask
RGB Image 2 from mainz
Ground Truth Mask
RGB Image 3 from mainz
Ground Truth Mask
RGB Image 4 from mainz
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 5 from mainz
Ground Truth Mask
RGB Image 6 from mainz
Ground Truth Mask
RGB Image 7 from mainz
Ground Truth Mask
RGB Image 8 from mainz
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

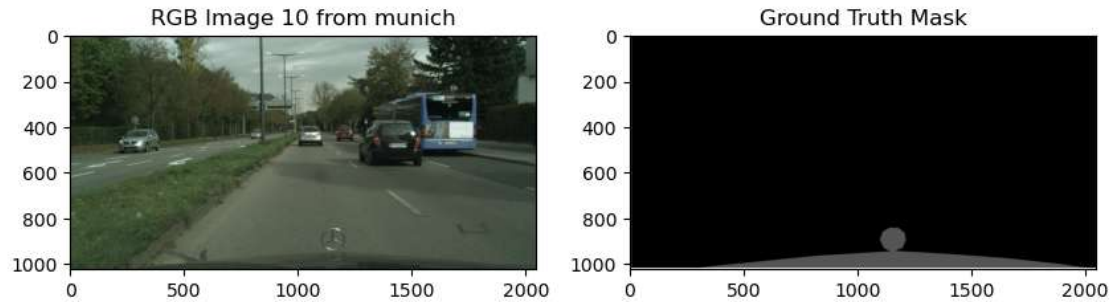



```
Finished processing city: mainz

Processing city: munich
```




RGB Image 2 from munich
Ground Truth Mask
RGB Image 3 from munich
Ground Truth Mask
RGB Image 4 from munich
Ground Truth Mask
RGB Image 5 from munich
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

RGB Image 6 from munich
Ground Truth Mask
RGB Image 7 from munich
Ground Truth Mask
RGB Image 8 from munich
Ground Truth Mask
RGB Image 9 from munich
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

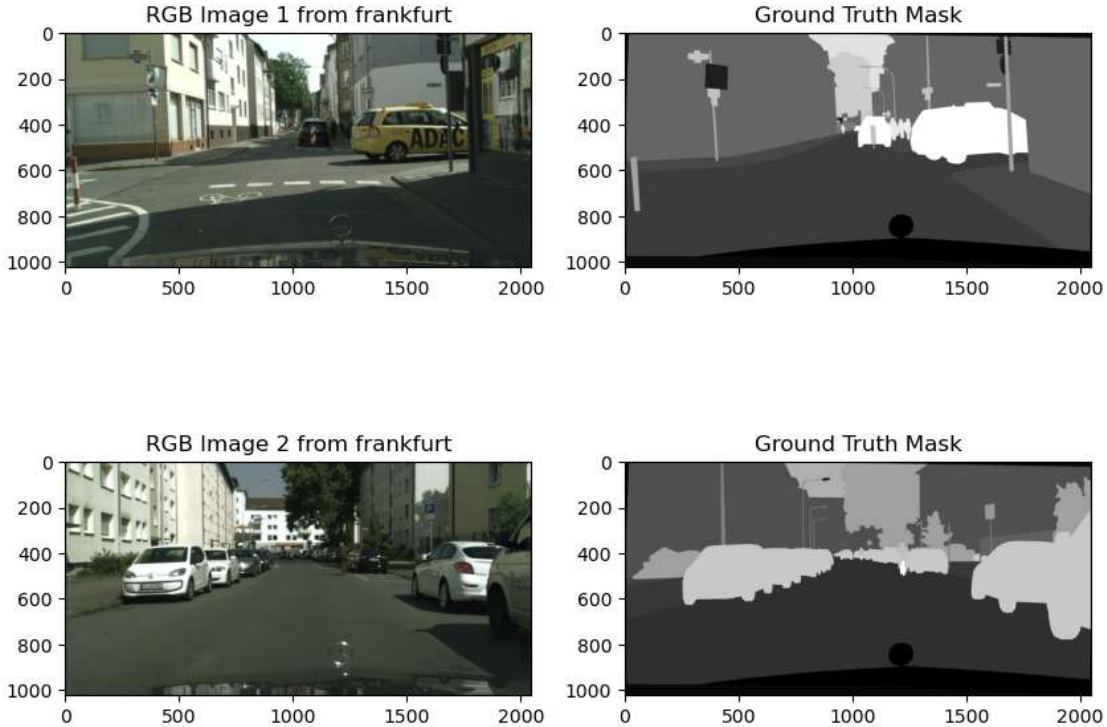


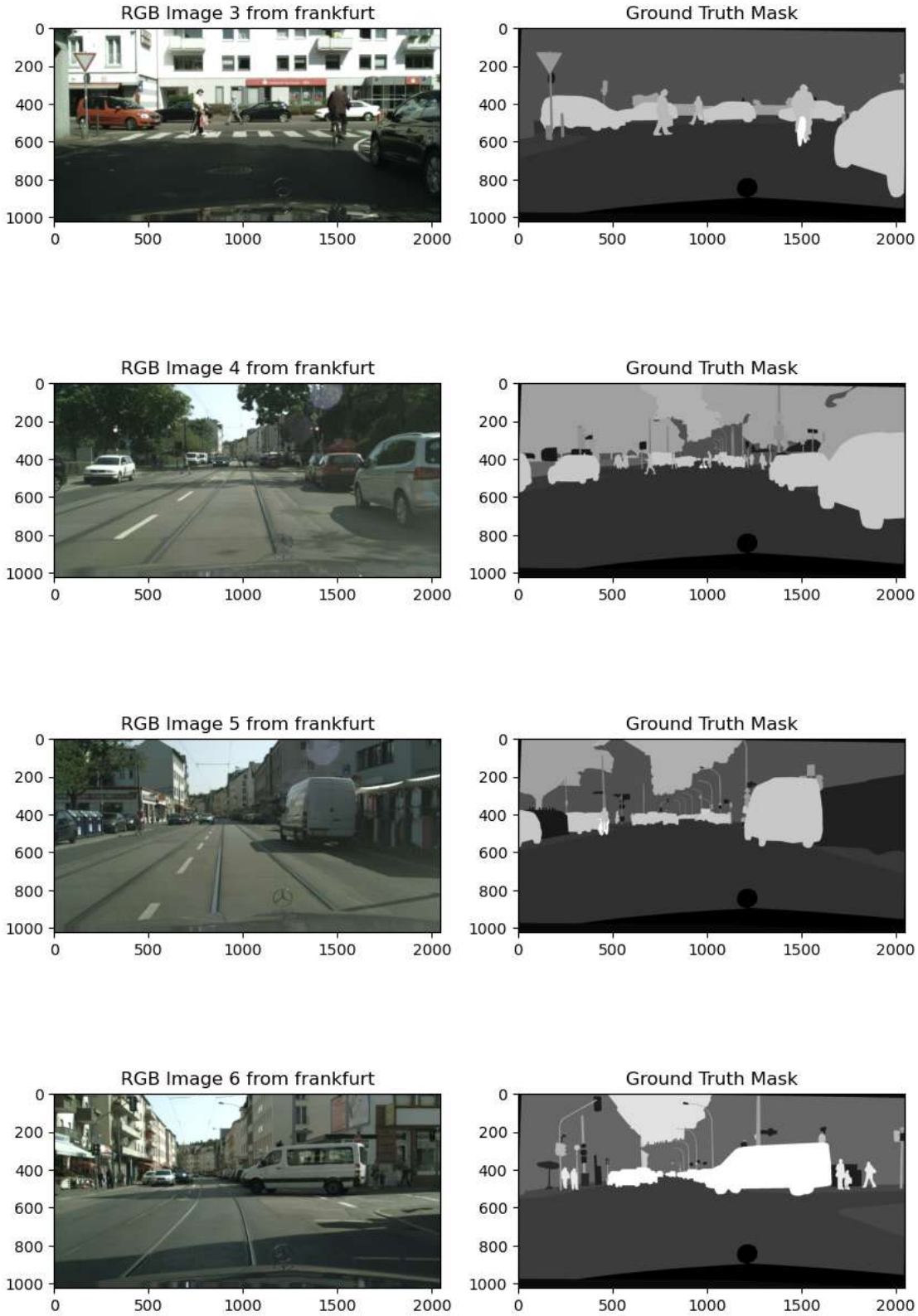

```
Finished processing city: munich
```

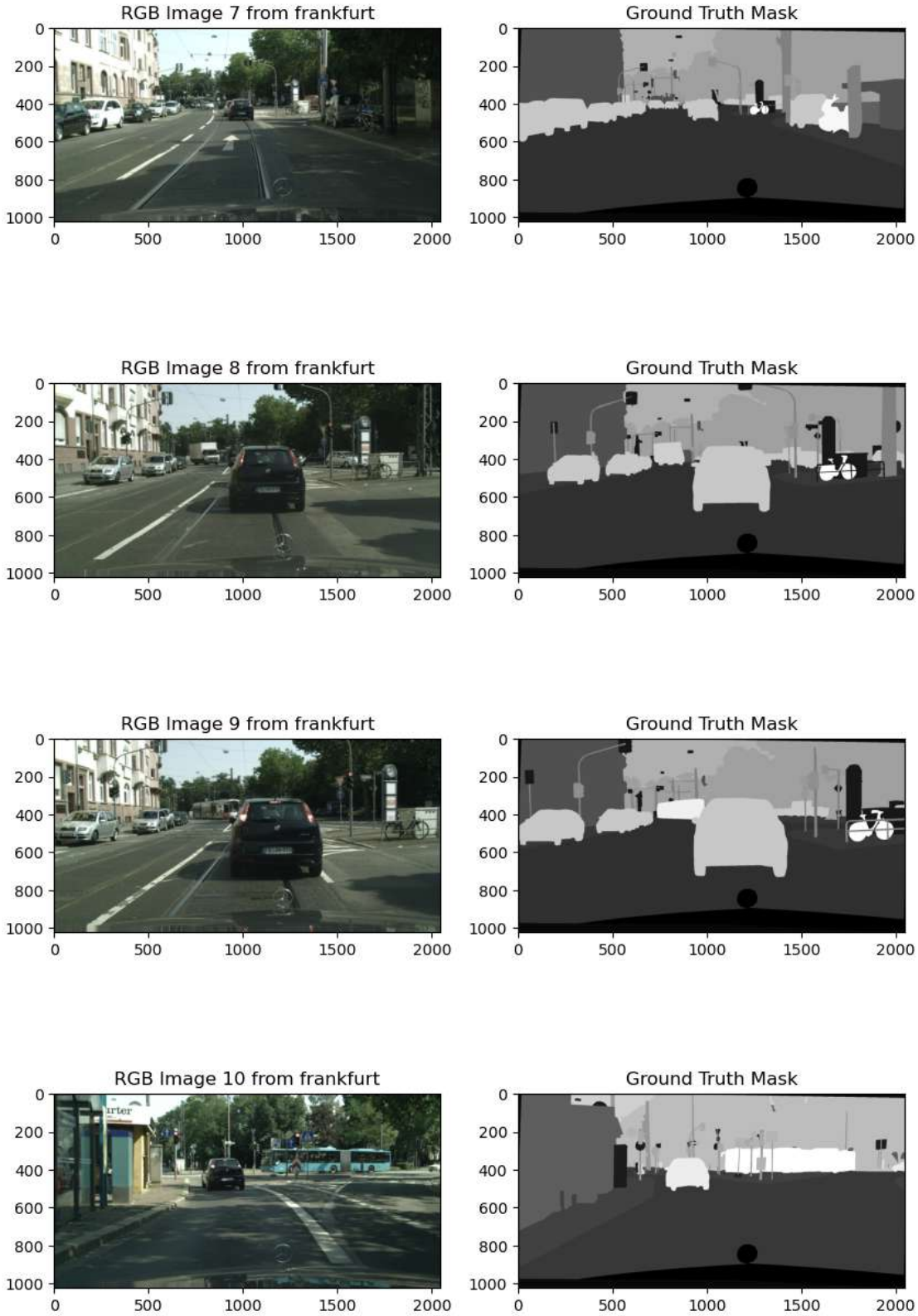
[4]:
```python
import os
from PIL import Image
import matplotlib.pyplot as plt

# Define paths to the datasets
img_path = r'D:
 ↪\infosys_springboard_project\leftImg8bit_trainvaltest\leftImg8bit\val'
gt_path = r'D:\infosys_springboard_project\gtFine_trainvaltest\gtFine\val'

# Get the list of cities (assuming cityscapes-style structure)
cities = os.listdir(img_path)

# Loop through all the cities in the dataset
for city in cities:
    print(f"Processing city: {city}")

    # Get the list of image files in the city folder
    img_files = os.listdir(os.path.join(img_path, city))

    # Initialize a counter for the number of images displayed
    displayed_images = 0

    # Loop through the images and find the corresponding ground truth files
    for img_file in img_files:
        if displayed_images >= 10:  # Only display 10 images per city
            break

        # Load the RGB image
        img = Image.open(os.path.join(img_path, city, img_file))

        # Load the corresponding ground truth file (gtFine has a different␣
 ↪naming convention)
```

```
        gt_file = img_file.replace('_leftImg8bit.png', '_gtFine_labelIds.png')
        gt_path_full = os.path.join(gt_path, city, gt_file)

        # Check if the ground truth file exists before loading
        if os.path.exists(gt_path_full):
            gt = Image.open(gt_path_full)

            # Display the image and the ground truth side by side
            fig, ax = plt.subplots(1, 2, figsize=(10, 5))
            ax[0].imshow(img)
            ax[0].set_title(f'RGB Image {displayed_images + 1} from {city}')
            ax[1].imshow(gt, cmap='gray')
            ax[1].set_title('Ground Truth Mask')
            plt.show()

            displayed_images += 1  # Increment the counter after displaying an␣
↪image
        else:
            print(f"Ground truth file not found for: {img_file}")

    print(f"Finished processing city: {city}\n")
```

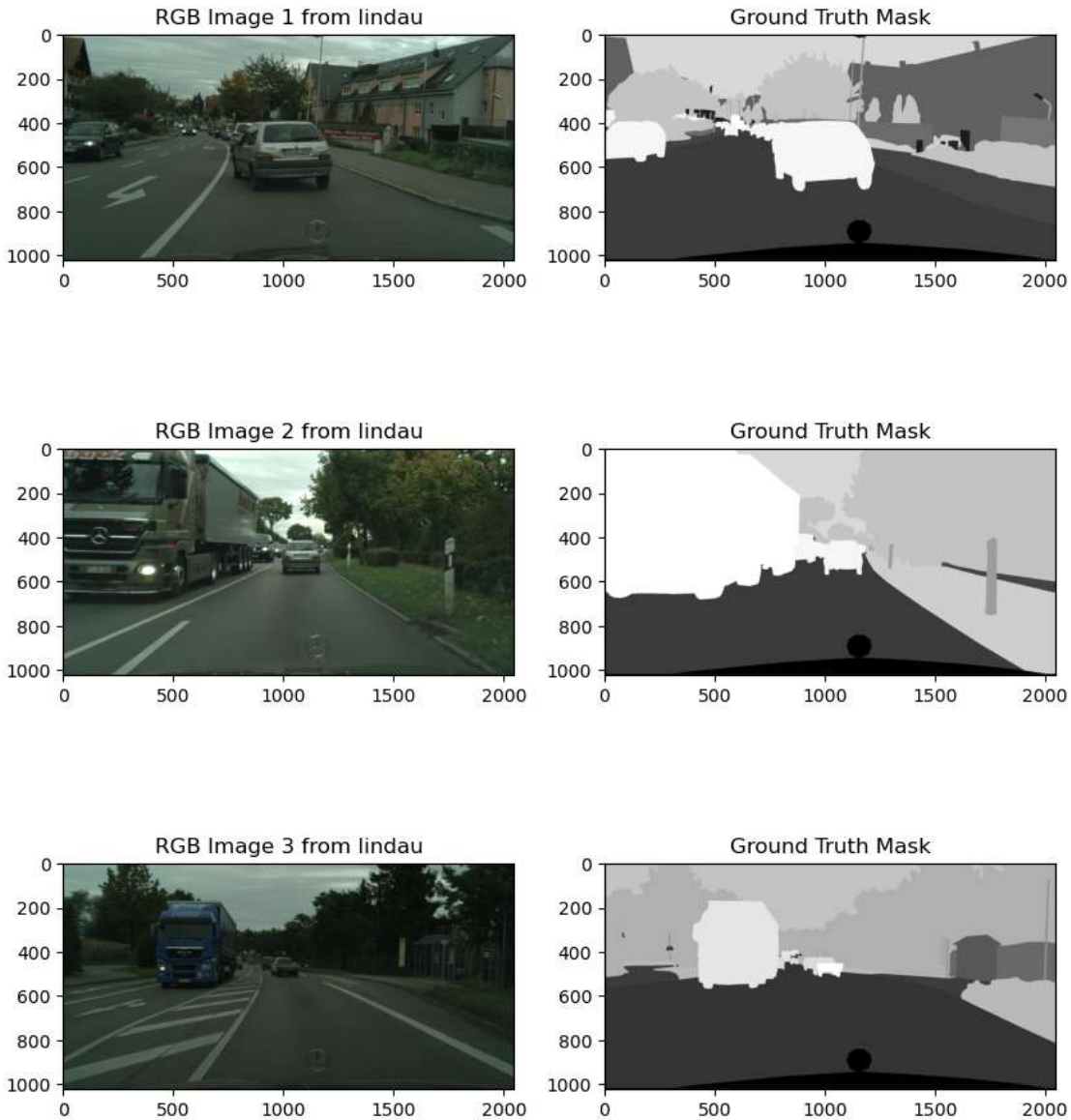
```
Processing city: frankfurt
```

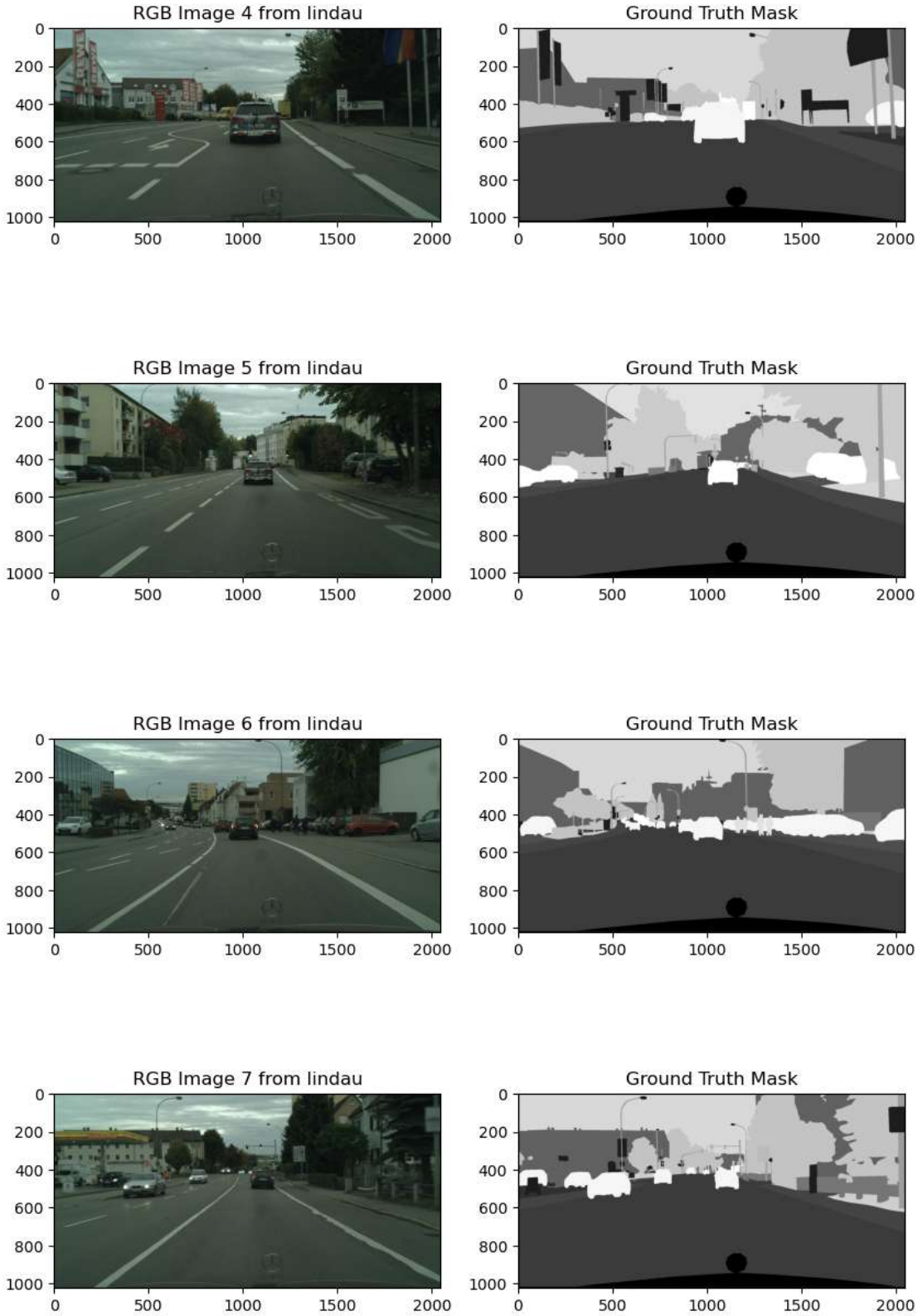


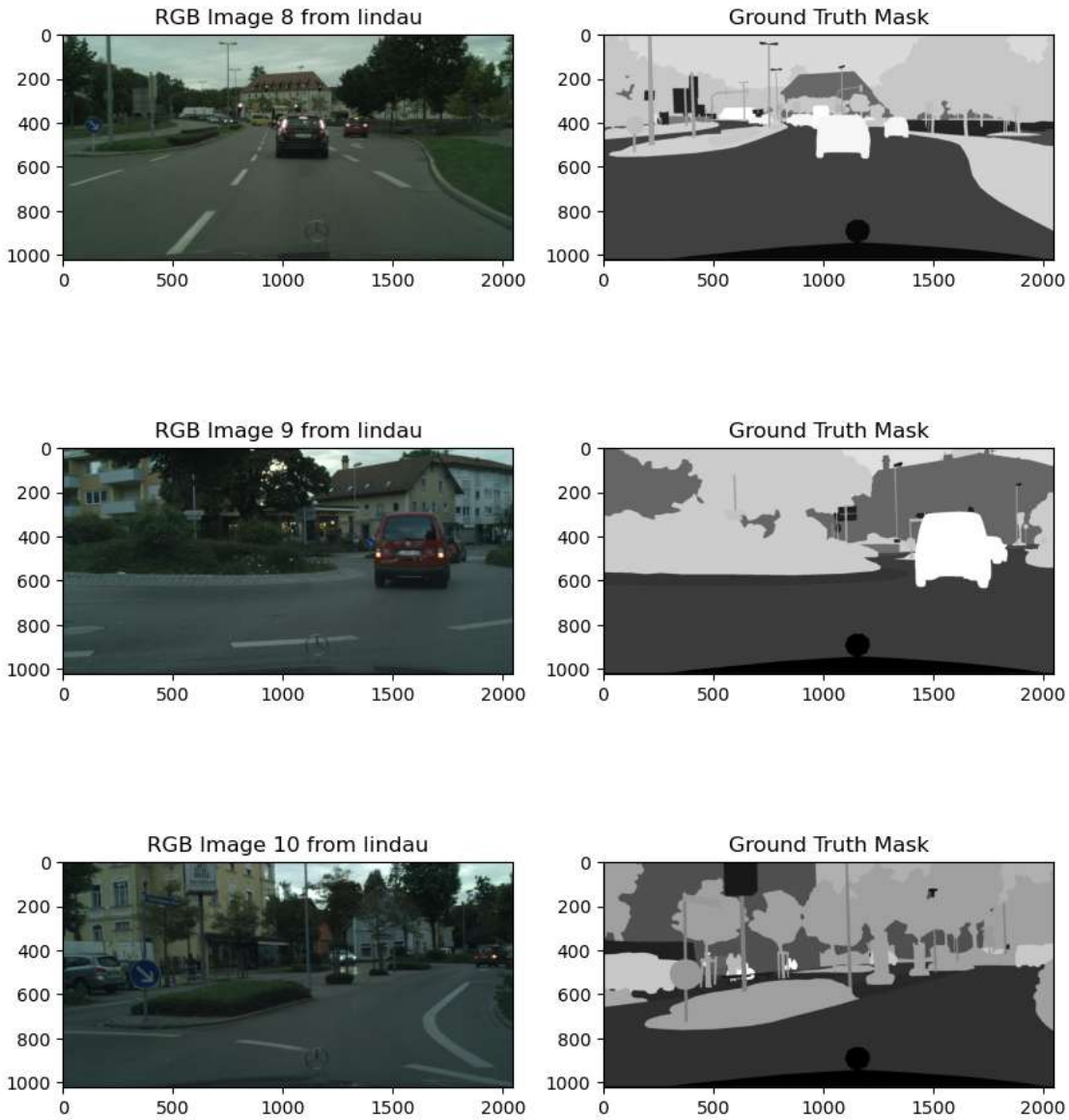

RGB Image 3 from frankfurt
Ground Truth Mask
RGB Image 4 from frankfurt
Ground Truth Mask
RGB Image 5 from frankfurt
Ground Truth Mask
RGB Image 6 from frankfurt
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 7 from frankfurt
Ground Truth Mask
RGB Image 8 from frankfurt
Ground Truth Mask
RGB Image 9 from frankfurt
Ground Truth Mask
RGB Image 10 from frankfurt
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000

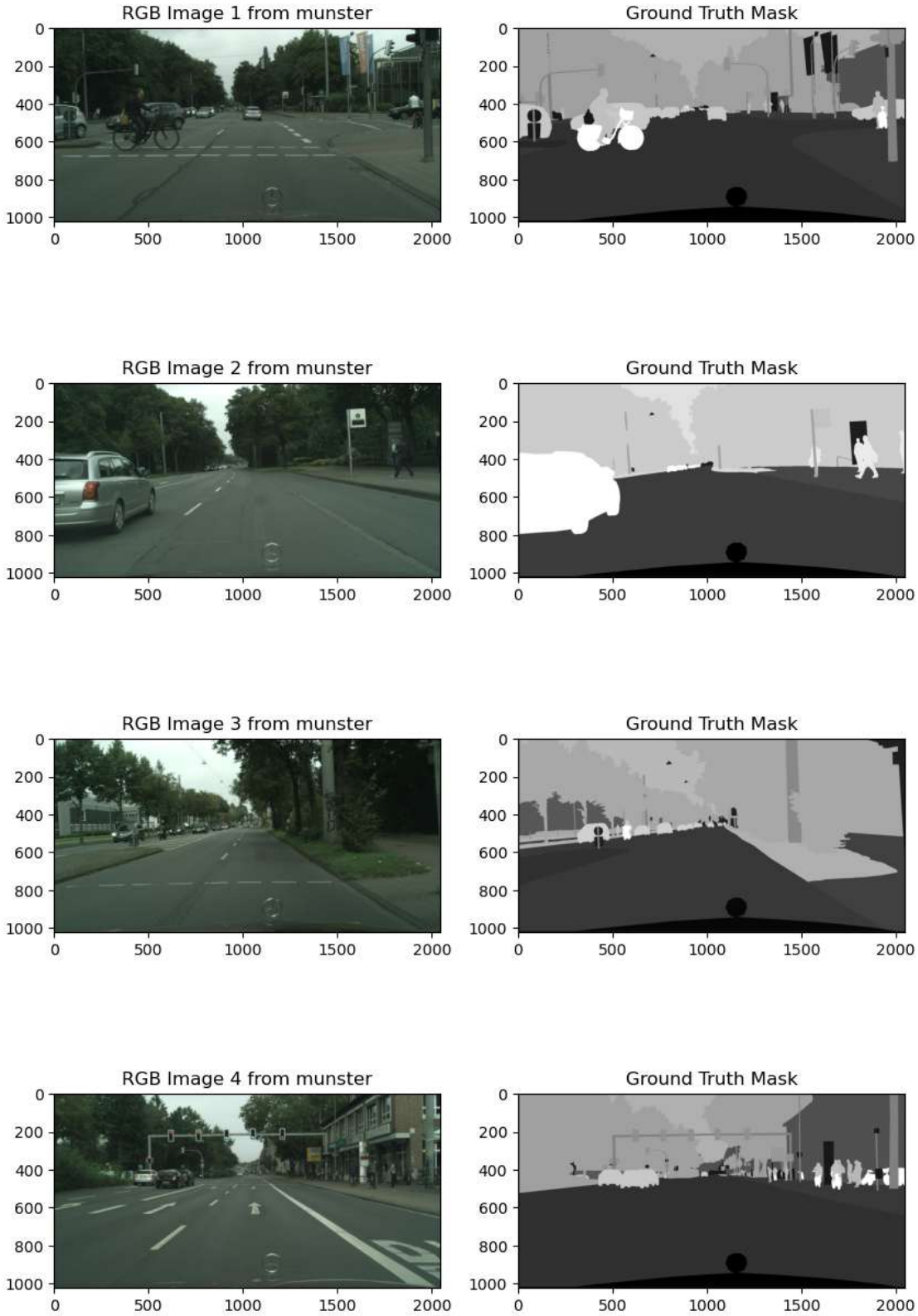
```
Finished processing city: frankfurt

Processing city: lindau
```

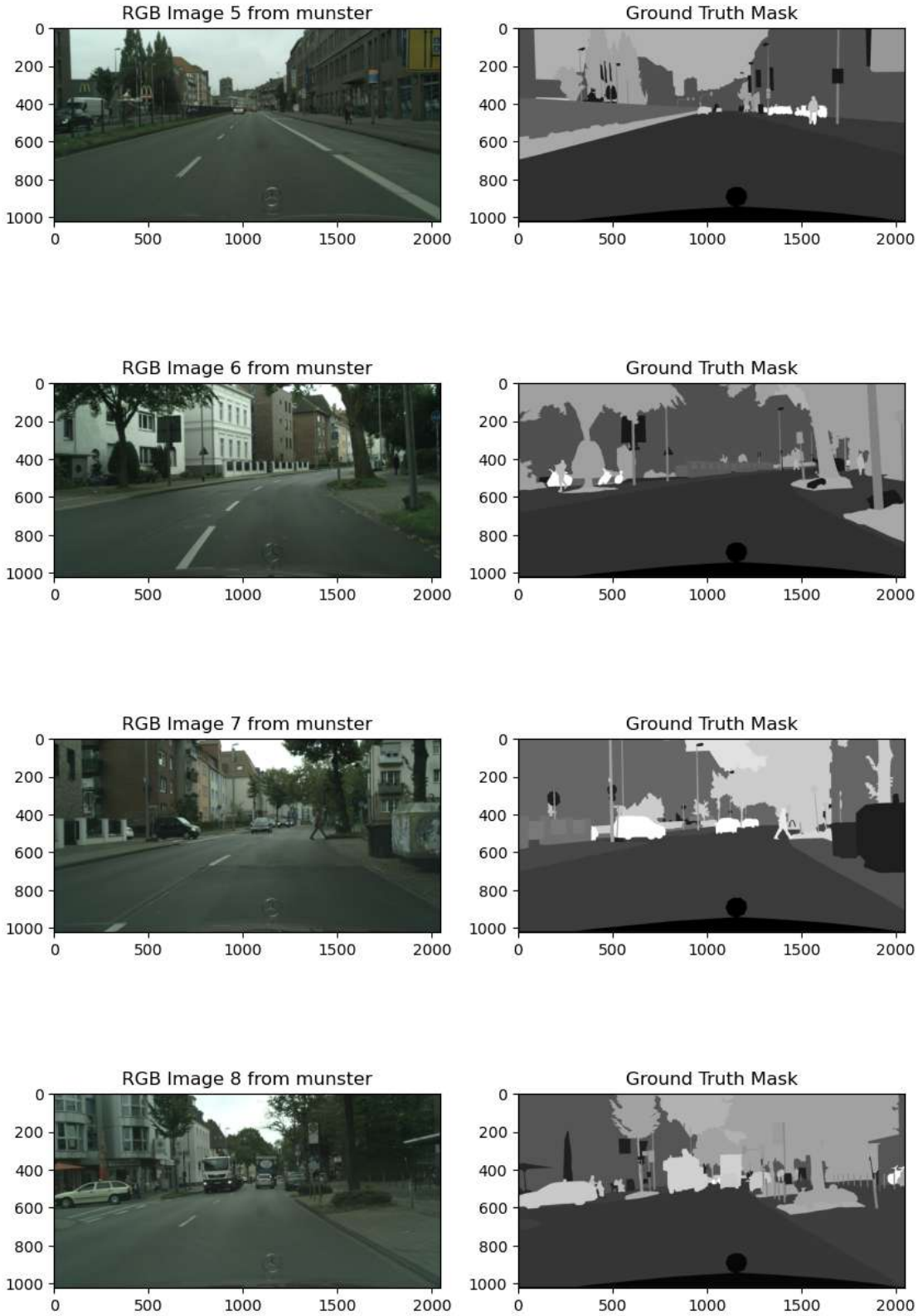


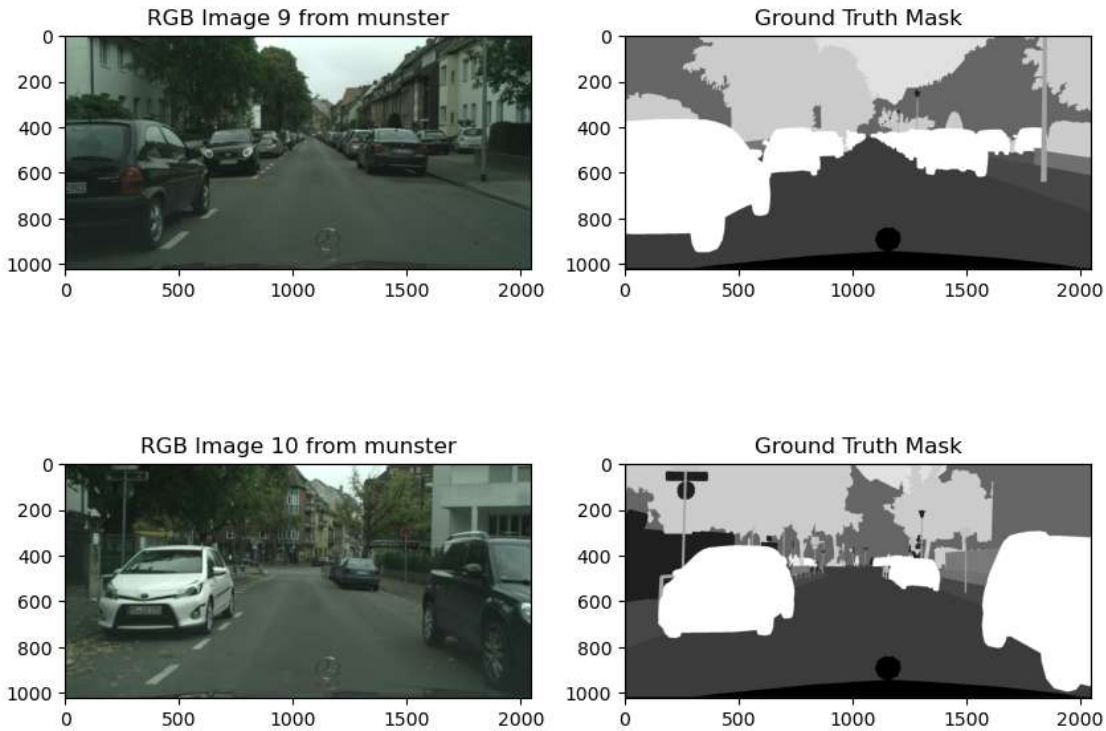

RGB Image 4 from lindau
Ground Truth Mask
RGB Image 5 from lindau
Ground Truth Mask
RGB Image 6 from lindau
Ground Truth Mask
RGB Image 7 from lindau
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000




```
Finished processing city: lindau

Processing city: munster
```


RGB Image 1 from munster
Ground Truth Mask
RGB Image 2 from munster
Ground Truth Mask
RGB Image 3 from munster
Ground Truth Mask
RGB Image 4 from munster
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000


RGB Image 5 from munster
Ground Truth Mask
RGB Image 6 from munster
Ground Truth Mask
RGB Image 7 from munster
Ground Truth Mask
RGB Image 8 from munster
Ground Truth Mask
0
200
400
600
800
1000
0
500
1000
1500
2000




```
Finished processing city: munster
```

[8]:
```python
import numpy as np
from tensorflow.keras.preprocessing.image import img_to_array
from PIL import Image
import os

# Preprocess image (resize and normalize)
def preprocess_image(img, target_size=(256, 256)):
    img = img.resize(target_size)  # Resize image to 256x256
    img = img_to_array(img) / 255.0  # Normalize pixel values between 0 and 1
    return img

# Preprocess mask (resize and binarize)
def preprocess_mask(mask, target_size=(256, 256)):
    mask = mask.resize(target_size, resample=Image.NEAREST)  # Resize to 256x256
    mask = img_to_array(mask)  # Convert mask to array
    mask = np.where(mask > 0, 1, 0)  # Binarize mask
    return mask

# Function to preprocess data for a given dataset
def preprocess_dataset(img_path, gt_path):
    preprocessed_images = []
```

```
    preprocessed_masks = []

    # Get the list of cities (assuming cityscapes-style structure)
    cities = os.listdir(img_path)

    # Loop through all cities
    for city in cities:
        print(f"Processing city: {city}")

        # Get list of image files for the city
        img_files = os.listdir(os.path.join(img_path, city))

        # Loop through each image in the city
        for img_file in img_files:
            # Load the RGB image
            img = Image.open(os.path.join(img_path, city, img_file))

            # Find the corresponding ground truth file
            gt_file = img_file.replace('_leftImg8bit.png', '_gtFine_labelIds.
↪png')
            gt_path_full = os.path.join(gt_path, city, gt_file)

            # Check if the ground truth file exists
            if os.path.exists(gt_path_full):
                # Preprocess the image
                preprocessed_img = preprocess_image(img)
                preprocessed_images.append(preprocessed_img)

                # Load and preprocess the mask
                gt = Image.open(gt_path_full)
                preprocessed_gt = preprocess_mask(gt)
                preprocessed_masks.append(preprocessed_gt)

                #print(f"Processed: {img_file}")

            else:
                # Skip the image if no corresponding ground truth mask is found
                print(f"Skipping {img_file}: ground truth file not found.")

        print(f"Finished processing city: {city}\n")

    # Convert preprocessed lists to numpy arrays
    preprocessed_images = np.array(preprocessed_images)
    preprocessed_masks = np.array(preprocessed_masks)

    return preprocessed_images, preprocessed_masks
```

```python
# Define dataset paths
train_img_path = 'D:
 ↪\infosys_springboard_project\leftImg8bit_trainvaltest\leftImg8bit/train'
train_gt_path = 'D:\infosys_springboard_project\gtFine_trainvaltest\gtFine/
 ↪train'

val_img_path = 'D:
 ↪\infosys_springboard_project\leftImg8bit_trainvaltest\leftImg8bit/val'
val_gt_path = 'D:\infosys_springboard_project\gtFine_trainvaltest\gtFine/val'

test_img_path = 'D:
 ↪\infosys_springboard_project\leftImg8bit_trainvaltest\leftImg8bit/test'
test_gt_path = 'D:\infosys_springboard_project\gtFine_trainvaltest\gtFine/test'

# Preprocess train, val, and test datasets
train_images, train_masks = preprocess_dataset(train_img_path, train_gt_path)
val_images, val_masks = preprocess_dataset(val_img_path, val_gt_path)
test_images, test_masks = preprocess_dataset(test_img_path, test_gt_path)

# Output the shapes of preprocessed datasets
print(f"Train Images Shape: {train_images.shape}")
print(f"Train Masks Shape: {train_masks.shape}")
print(f"Val Images Shape: {val_images.shape}")
print(f"Val Masks Shape: {val_masks.shape}")
print(f"Test Images Shape: {test_images.shape}")
print(f"Test Masks Shape: {test_masks.shape}")
```

```
Processing city: aachen
Finished processing city: aachen

Processing city: bochum
Finished processing city: bochum

Processing city: bremen
Finished processing city: bremen

Processing city: cologne
Finished processing city: cologne

Processing city: darmstadt
Finished processing city: darmstadt

Processing city: dusseldorf
Finished processing city: dusseldorf

Processing city: erfurt
Finished processing city: erfurt
```

```
Processing city: hamburg
Finished processing city: hamburg

Processing city: hanover
Finished processing city: hanover

Processing city: jena
Skipping jena_000001_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000002_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000011_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000015_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000020_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000028_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000032_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000041_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000045_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000048_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000064_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000071_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000078_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000080_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000084_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000092_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000093_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000097_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000098_000019_leftImg8bit.png: ground truth file not found.
Skipping jena_000105_000019_leftImg8bit.png: ground truth file not found.
Finished processing city: jena

Processing city: krefeld
Finished processing city: krefeld

Processing city: monchengladbach
Finished processing city: monchengladbach

Processing city: strasbourg
Finished processing city: strasbourg

Processing city: stuttgart
Finished processing city: stuttgart

Processing city: tubingen
Finished processing city: tubingen

Processing city: ulm
Finished processing city: ulm
```

```
Processing city: weimar
Finished processing city: weimar

Processing city: zurich
Finished processing city: zurich

Processing city: frankfurt
Finished processing city: frankfurt

Processing city: lindau
Finished processing city: lindau

Processing city: munster
Finished processing city: munster

Processing city: berlin
Finished processing city: berlin

Processing city: bielefeld
Finished processing city: bielefeld

Processing city: bonn
Finished processing city: bonn

Processing city: leverkusen
Finished processing city: leverkusen

Processing city: mainz
Finished processing city: mainz

Processing city: munich
Finished processing city: munich

Train Images Shape: (2955, 256, 256, 3)
Train Masks Shape: (2955, 256, 256, 1)
Val Images Shape: (500, 256, 256, 3)
Val Masks Shape: (500, 256, 256, 1)
Test Images Shape: (1525, 256, 256, 3)
Test Masks Shape: (1525, 256, 256, 1)
```

[11]:
```python
import numpy as np
import zipfile
import os

# Path where the zip file will be saved
zip_file_path = 'D:/infosys_springboard_project/preprocessed_data.zip'
```

```python
# Create a zip file
with zipfile.ZipFile(zip_file_path, 'w') as zipf:
    # Save train data
    np.save('train_images.npy', train_images)
    np.save('train_masks.npy', train_masks)
    zipf.write('train_images.npy')
    zipf.write('train_masks.npy')

    # Save validation data
    np.save('val_images.npy', val_images)
    np.save('val_masks.npy', val_masks)
    zipf.write('val_images.npy')
    zipf.write('val_masks.npy')

    # Save test data
    np.save('test_images.npy', test_images)
    np.save('test_masks.npy', test_masks)
    zipf.write('test_images.npy')
    zipf.write('test_masks.npy')

# Remove the .npy files after zipping
os.remove('train_images.npy')
os.remove('train_masks.npy')
os.remove('val_images.npy')
os.remove('val_masks.npy')
os.remove('test_images.npy')
os.remove('test_masks.npy')

print(f"Data saved and zipped at: {zip_file_path}")
```

```
Data saved and zipped at: D:/infosys_springboard_project/preprocessed_data.zip
```

[ ]:

```python

```