

Dokumentation für klassische Methoden der Augmentation

Erstellung der Pfade

Zum ersten Schritt muss die *path_creating.ipynb* Datei ausgeführt werden. Es ist die Pfade anzupassen, die für die Datensätze (Bilder (und Masken) für Vorder- und Hintergrund) und csv-Dateien (Abb. 1) sind, die später erstellt werden.

Creating of data path list with objects

```
json_directory = '/home/admin1/Document/CITYSCAPES_DATASET/gtFine_trainvaltest/gtFine/'  
fgPaths = '/home/admin1/Programming/HIWI_MRT/image_data_augmentation/basic_approaches/citysc_fgPaths.csv'  
bgPaths = '/home/admin1/Programming/HIWI_MRT/image_data_augmentation/basic_approaches/citysc_bgPaths.csv'
```

Abb. 1: Pfade für die Datensätze und zukünftige csv-Dateien

Nach der Programmausführung werden zwei csv-Dateien erstellt, die aus der Liste der Pfade bestehen. Jede Pfad ist die Adresse, wo das jeweilige Bild platziert ist. Im ersten Fall sind Bilder und Masken mit dem Vordergrund als *citysc_fgPaths.csv* benannt. Sie haben “person” Klasse. Darüber hinaus bestehen diese Bilder aus Personen, mindestens eine davon grösser als *obj_bg_ratio* Verhältnis.

Im zweiten Fall besteht die zweite Datei aus den Bildern und Masken, die den Hintergrund bezeichnen und “ground”, “road” und/oder “sidewalk” Klasse/-n haben.

Erstellung des Datensatzes

1. Allgemeine Beschreibung des Programms

Zum zweiten Schritt wird der Datensatz aus den klassisch augmentierten Bildern und entsprechenden Masken erstellt. Die Anzahl der Datensatzbilder muss festgelegt werden, die als *dataset_size* initialisiert ist. Zum Weiteren ist es die Pfade anzupassen, die für den zu erstellenden Datensatz (Bilder und Masken) und csv-Dateien (Abb. 2) sind.

Data path list creating

Paths in csv. file reading

```
fgPaths = '/home/admin1/Programming/HIWI_MRT/image_data_augmentation/basic_approaches/citysc_fgPaths.csv'  
bgPaths = '/home/admin1/Programming/HIWI_MRT/image_data_augmentation/basic_approaches/citysc_bgPaths.csv'  
  
save_directory = '/home/admin1/Programming/HIWI_MRT/cityscapes_data/created_dataset/'
```

Abb. 2: Pfade für den zukünftigen Datensatz und csv-Dateien

Dann müssen die Listen mit Pfaden zu den Vordergrund- und Hintergrundbildern erstellt werden. Es gibt zwei Möglichkeiten:

- Das Programm kann die csv-Dateien einlesen, die aus Pfaden zum Datensatz bestehen.

In diesem Fall führen wir den Programmteil in Abb. 3 aus.

Paths in csv. file reading

```
fgPaths = '/home/admin1/Programming/HIWI_MRT/image_data_augmentation/basic_approaches/citysc_fgPaths.csv'
bgPaths = '/home/admin1/Programming/HIWI_MRT/image_data_augmentation/basic_approaches/citysc_bgPaths.csv'

save_directory = '/home/admin1/Programming/HIWI_MRT/cityscapes_data/created_dataset/'

def pathReader(path):
    # Read paths of a CSV file
    with open(path, newline='') as fg_bg_data:
        reader = csv.reader(fg_bg_data)
        data = list(reader)
    return data

fg_path_list = pathReader(fgPaths)
bg_path_list = pathReader(bgPaths)
```

Abb. 3: Einlesen der csv-Dateien

- Die Pfade zu jedem Bild und jeweiliger Maske kann per Hand eingetippt, Kommentare weggelassen und ausgeführt (Abb. 4) werden.

Program testing

! Second variante !

```
# Activate for the testing of the particular images, masks
"""
fg_dir = '/home/admin1/Programming/HIWI_MRT/cityscapes_data/foreground/'
bg_dir = '/home/admin1/Programming/HIWI_MRT/cityscapes_data/background/'

save_directory = '/home/admin1/Programming/HIWI_MRT/cityscapes_data/created_dataset/'

fg_path_list = [fg_dir+'img'+'/aachen_000020_000019_leftImg8bit.png', fg_dir+'mask'+'/aachen_000020_000019_gtFine_l
[fg_dir+'img'+'/aachen_000031_000019_leftImg8bit.png', fg_dir+'mask'+'/aachen_000031_000019_gtFine_l
[fg_dir+'img'+'/aachen_000114_000019_leftImg8bit.png', fg_dir+'mask'+'/aachen_000114_000019_gtFine_l
[fg_dir+'img'+'/cologne_000056_000019_leftImg8bit.png', fg_dir+'mask'+'/cologne_000056_000019_gtFine
[fg_dir+'img'+'/cologne_000100_000019_leftImg8bit.png', fg_dir+'mask'+'/cologne_000100_000019_gtFine
[fg_dir+'img'+'/cologne_000108_000019_leftImg8bit.png', fg_dir+'mask'+'/cologne_000108_000019_gtFine
[fg_dir+'img'+'/cologne_000121_000019_leftImg8bit.png', fg_dir+'mask'+'/cologne_000121_000019_gtFine

bg_path_list = [bg_dir+'img'+'/bremen_000000_000019_leftImg8bit.png', bg_dir+'mask'+'/bremen_000000_000019_gtFine_l
[bg_dir+'img'+'/bremen_000001_000019_leftImg8bit.png', bg_dir+'mask'+'/bremen_000001_000019_gtFine_l
[bg_dir+'img'+'/bremen_000009_000019_leftImg8bit.png', bg_dir+'mask'+'/bremen_000009_000019_gtFine_l
"""
```

Abb. 4: Eintippen der Pfade manuell

Bei der Ausführung der *main*-Funktion werden neue Bilder erstellt, die aus dem ursprünglichem Datensatz bestehen.

Besonderheiten und Funktionen

Das Programm wählt ein zufälliges Bild und eine entsprechende Maske als Vordergrund-Vordergrundbild (Abb. 5) und ein anderes Bild und seine Maske als Hintergrund-Hintergrundbild (Abb. 6).



Abb. 5: Vordergrundbild (oben), Vordergrundmaske (unten)



Abb. 6: Hintergrundbild (oben), Hintergrundmaske (unten)

Zum nächsten Schritt werden die Funktionen für die Augmentierung eingesetzt. Das gewählte Vordergrund-, Hintergrundbild und die entsprechenden Masken werden zufällig horizontal oder vertikal gespiegelt.

Auf dem Vordergrundbild werden die Objekte mit der Klasse "Person" erkannt und eine zufällige Person wird gewählt. Auf dem Hintergrundbild wird eine zufällige Position (innerhalb der

Klassen “road”, “ground” und “sidewalk”) gefunden, worauf die ausgewählte Person angelegt wird. Vor dem Anlegen wird die Person entsprechend verkleinert oder vergrößert.

Die Größe der Person wird folgendermaßen ausgerechnet: Es wurde gemerkt, dass die Größen der Personen an den bestimmten Standorten auf dem Bild durchschnittlich einen Wert betragen. Es wurde angenommen, dass die Größe einer Person im ersten Fall bei $y = 1000 \text{ px}$ 800 px beträgt. Im zweiten Fall ist eine Person bei $y = 540 \text{ px}$ 170 px groß. Das Gleichungssystem macht das deutlich:

$$h = y \cdot a + b$$

h – Größe einer Person (px)

y – Position (px)

$$800 = 1000 \cdot a + b$$

$$170 = 540 \cdot a + b$$

Bei der Lösung des Gleichungssystem können die Unbekannten a und b gefunden werden. Dann kann die Größe der Person für jede Position auf dem Bild ausgerechnet werden.

Da nicht jedes Bild (Abb. 7) im *cityscapes* Datensatz perfekt annotiert ist, wird die Funktion *mattig* eingesetzt, die die entsprechenden Masken selbst generiert.

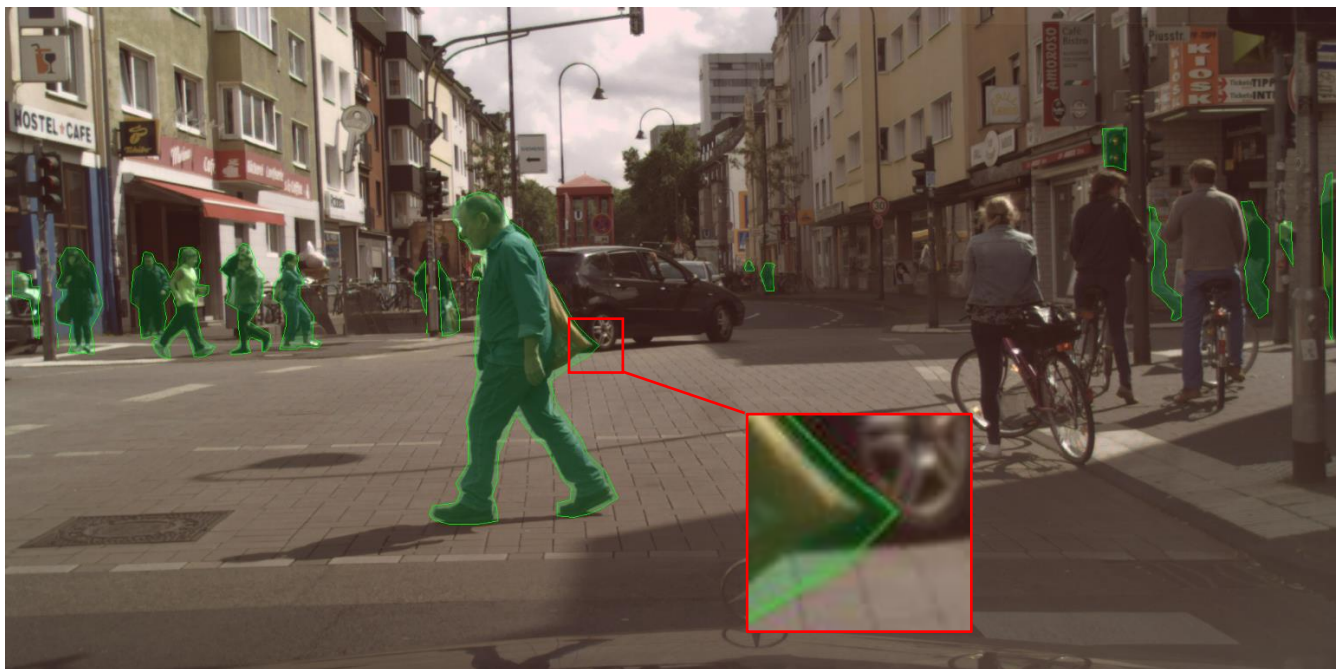


Abb. 7: Darstellung des *cologne_000056_000019_leftImg8bit.png* Bildes mit der angelegten Maske

Die *matting* Funktion ermöglicht es, die genaueren Masken zu erstellen. Darüber hinaus werden die schrägen Kanten zwischen des Objekts und Hintergrunds mittels des *alpha*-Kanals beim Anlegen des Objekts (Vordergrundbild) auf den Hintergrund (Hintergrundbild) (Abb. 8) vermieden.



Abb. 8: Darstellung des Bildes, nachdem das Objekt auf den Hintergrund angelegt und die *matting* Funktion eingesetzt wurde

Herausforderung und noch zu lösende Schwierigkeiten

Momentan generiert das aktuelle Programm oft die Bilder und jeweilige Masken, die nicht realistisch aussehen.

Größe der Person nicht realistisch

In einigen Fällen sehen die Personen nicht realistisch (Abb. 9) aus, nachdem das Programm sie verkleinert oder vergrößert hat.

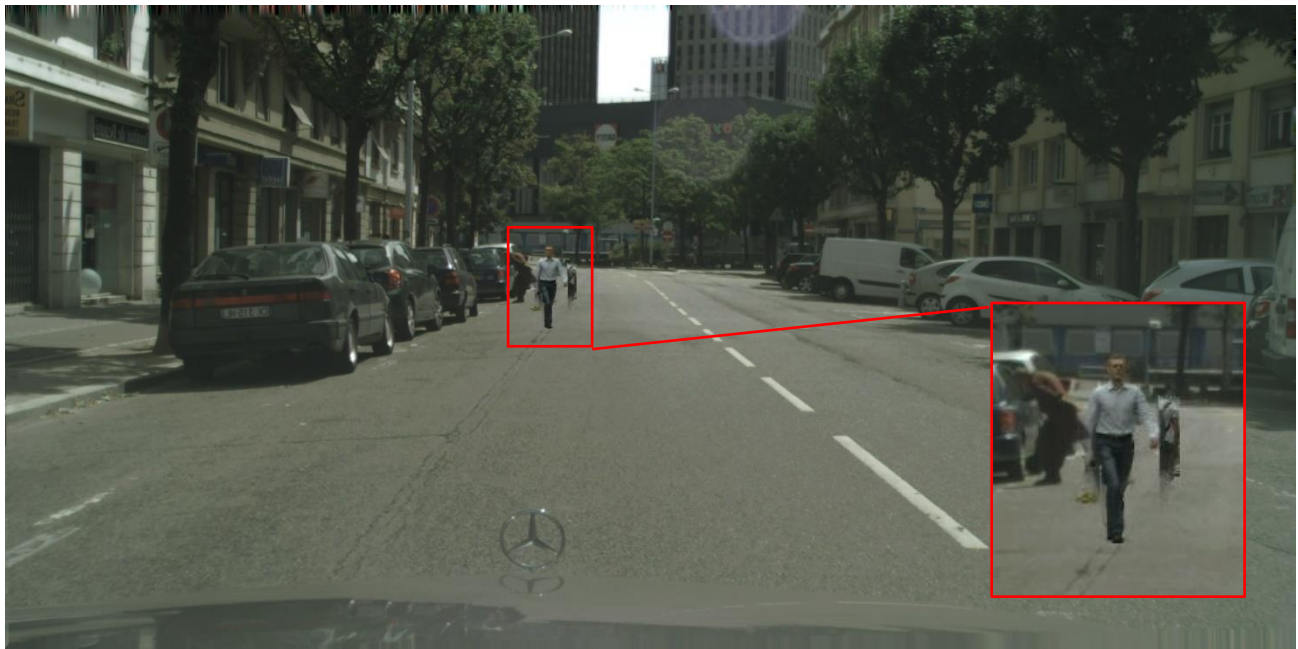


Abb. 9: Wegen ihrer Größe sieht die Person nicht realistisch aus

Erstellung der ungenauen Masken

Ab und zu erstellt die *matting* Funktion ungenaue Masken (Abb. 10). Aus diesem Grund sieht das angelegte Objekt nicht realistisch aus. Um dies Problem zu beheben, muss die Funktion `def border_blender(img, mask)` in `geometric_transformations.py` verbessert werden, die die *trimap* Masken im Programm erstellt.

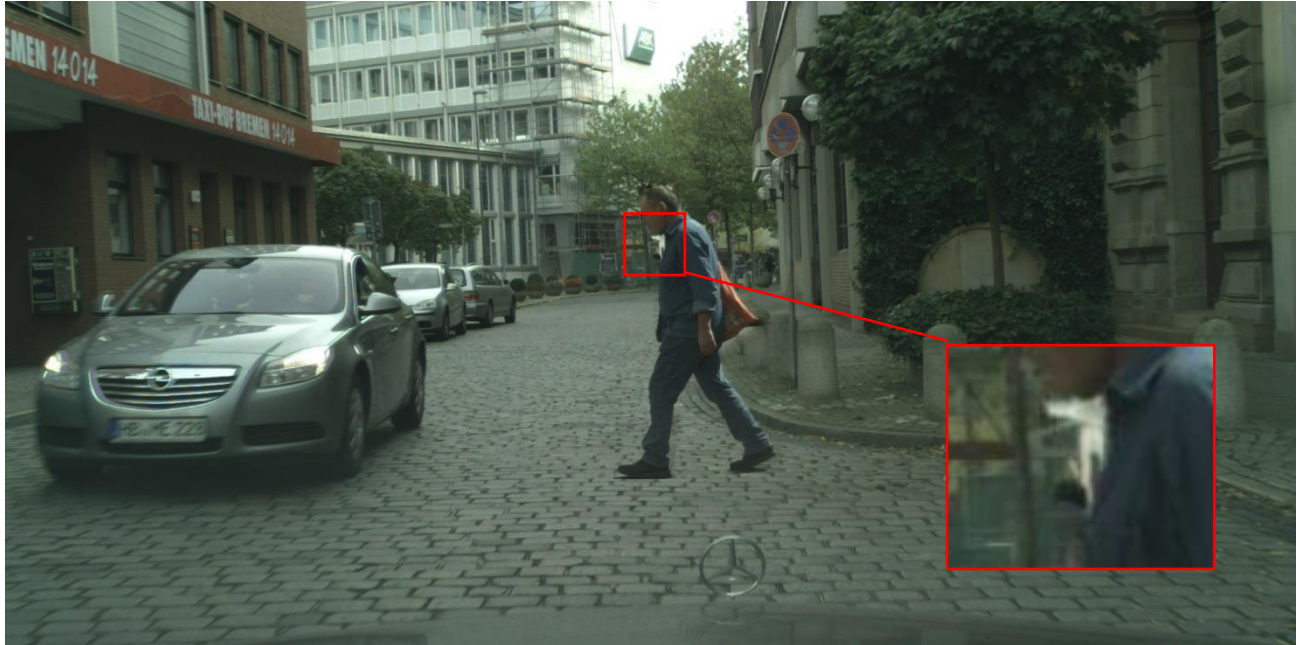


Abb. 10: Ungenaue Masken

Verdeckung

Die angelegten Personen können die Objekte (Abb. 11) verdecken.

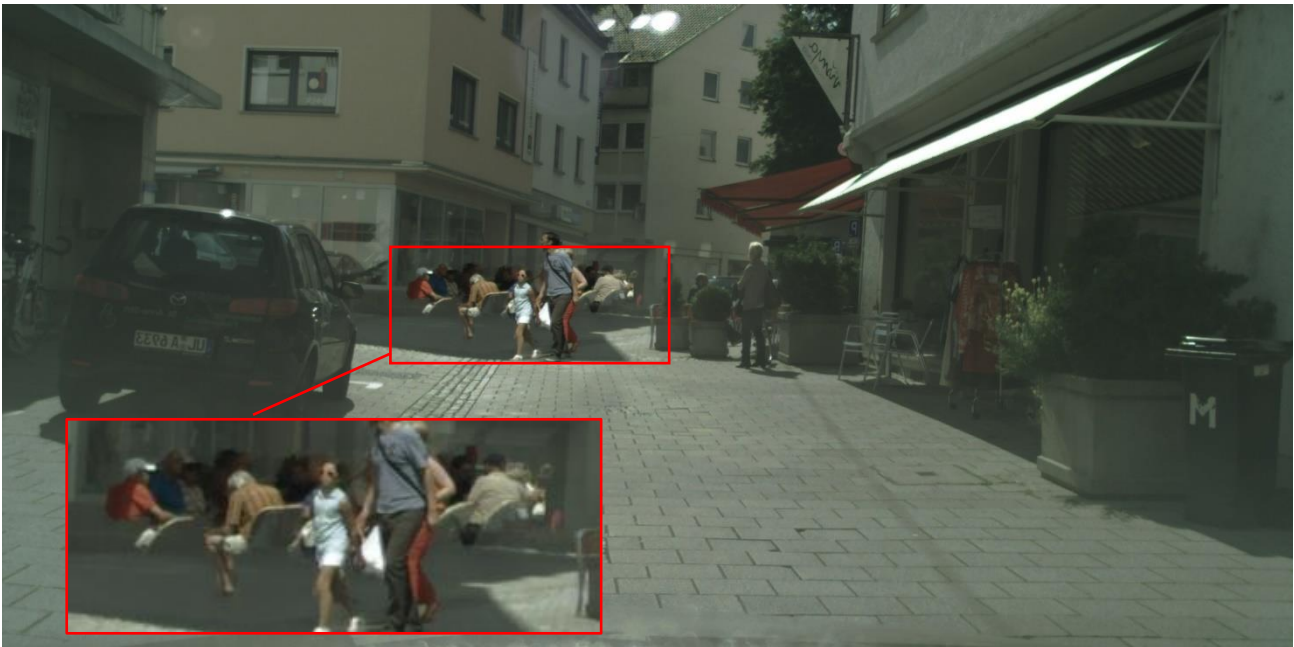
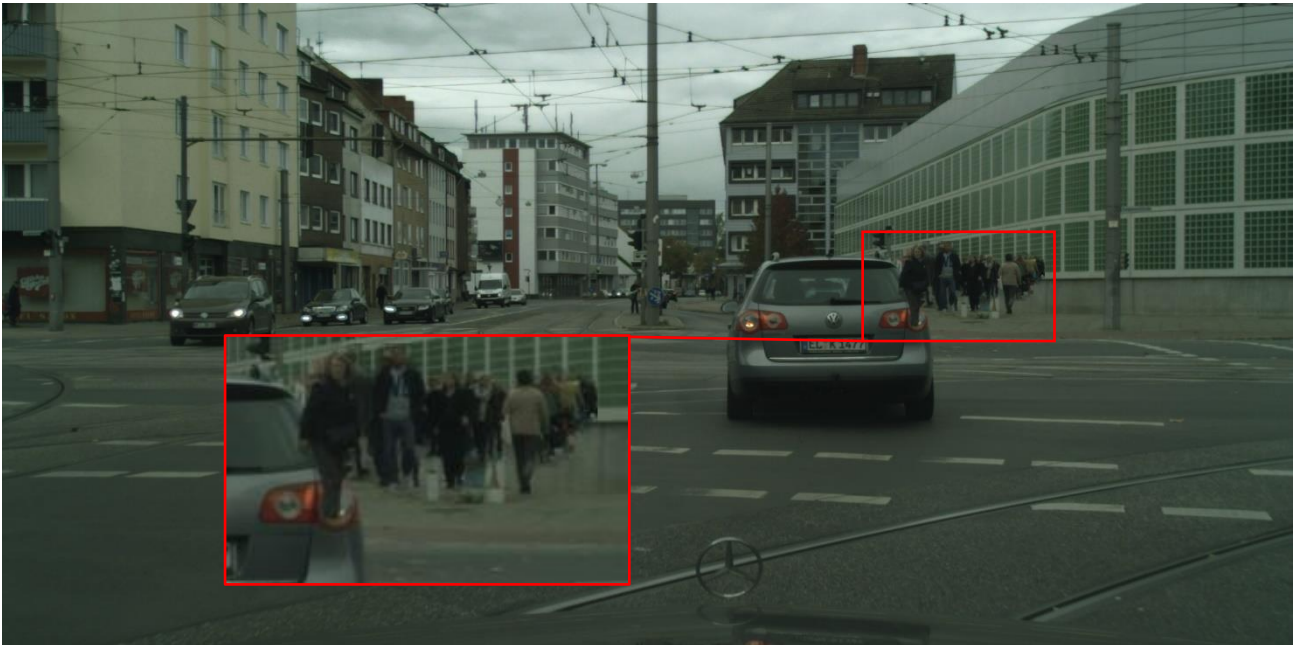


Abb. 11: Verdeckende Objekte (oben), verdeckte Objekte (unten)

Geschnittene Objekte

Die Personen sind geschnitten. Das liegt darauf an, dass die Person nicht vollständig auf dem ursprünglichen Bild (Abb. 12) dargestellt ist

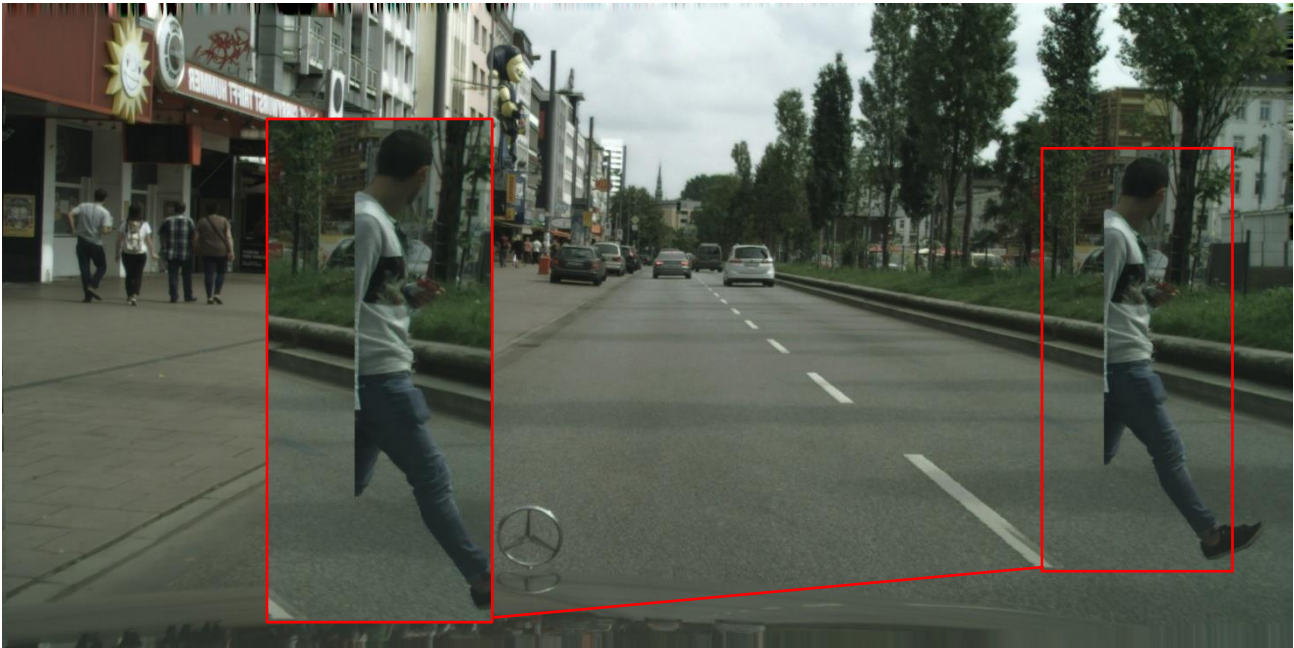


Abb. 12: Das Objekt ist geschnitten

Kontrast

Die Kontraste unterscheiden sich zwischen des Hintergrundes und angelegten Objektes
(Abb. 13)



Abb. 13: Unterschiedliche Kontraste