

# Interdisciplinary Project Big Data Analytics

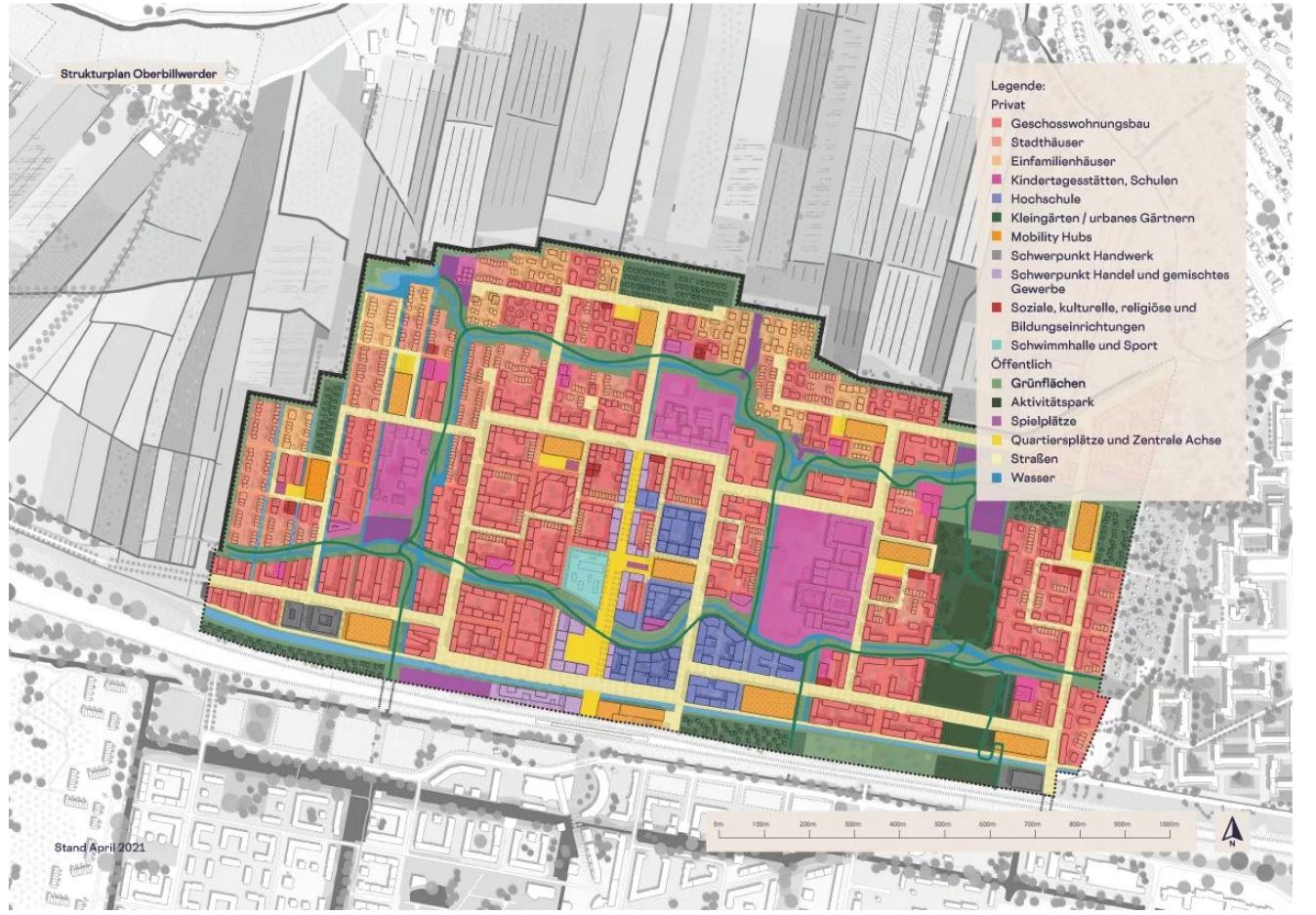
---

InstantCITY  
2nd presentation

Group 1  
Fabian Bloch & Christopher Mahn  
HafenCity University Hamburg

# Approach - Goal

- ▶ studying original repository
- ▶ exploring the provided examples
- ▶ digitize street layout of Oberbillwerder
- ▶ train GAN on new city developments in Hamburg
- ▶ apply GAN street layout of Oberbillwerder
- ▶ visually compare buildings with masterplan

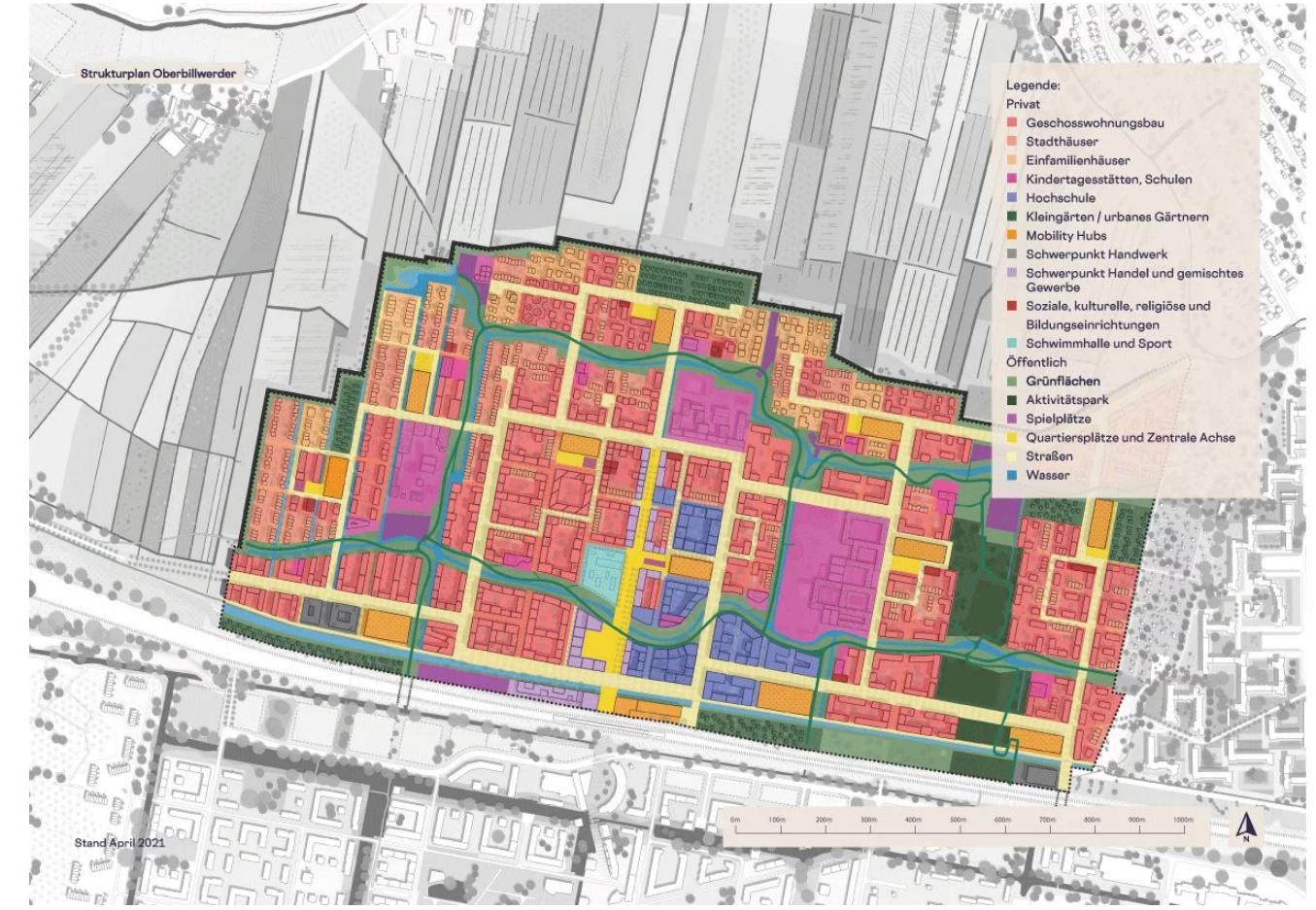


Example Oberbillwerder

[https://www.oberbillwerder-hamburg.de/projekt/  
masterplan-2022/](https://www.oberbillwerder-hamburg.de/projekt/masterplan-2022/)

# Approach - Goal

- ▶ studying original repository **done**
- ▶ exploring the provided examples
- ▶ digitize street layout of Oberbillwerder
- ▶ train GAN on new city developments in Hamburg
- ▶ apply GAN street layout of Oberbillwerder
- ▶ visually compare buildings with masterplan

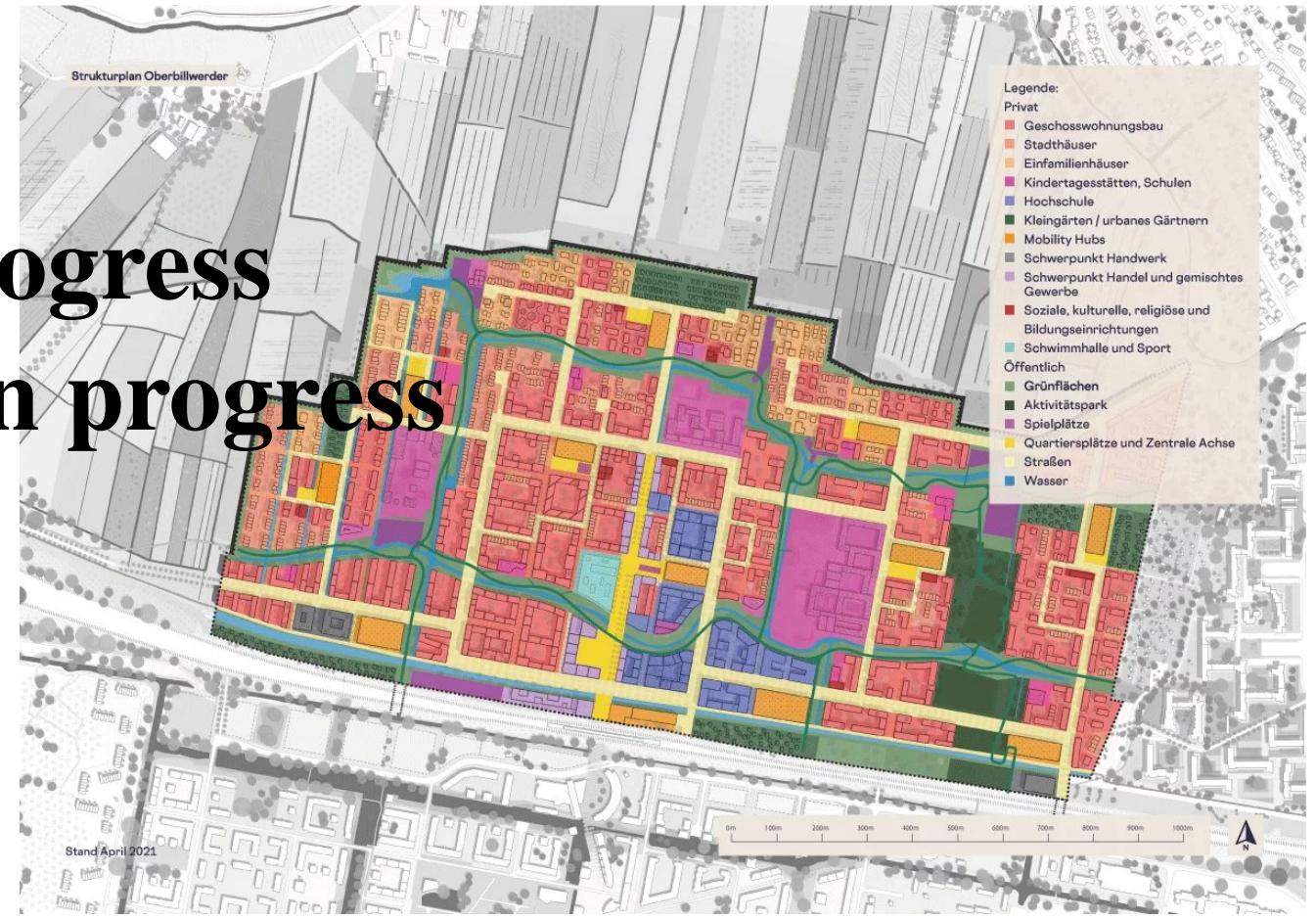


Example Oberbillwerder

[https://www.oberbillwerder-hamburg.de/projekt/  
masterplan-2022/](https://www.oberbillwerder-hamburg.de/projekt/masterplan-2022/)

# Approach - Goal

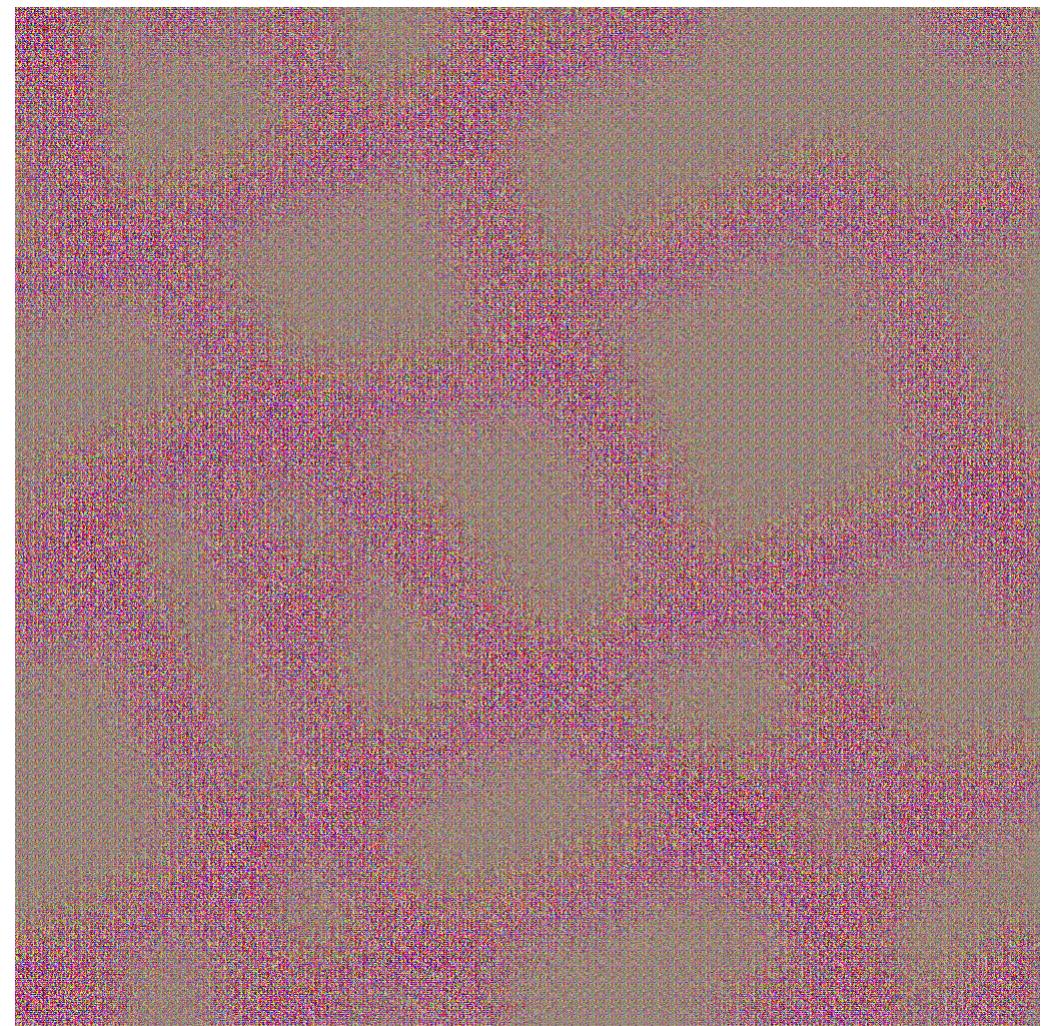
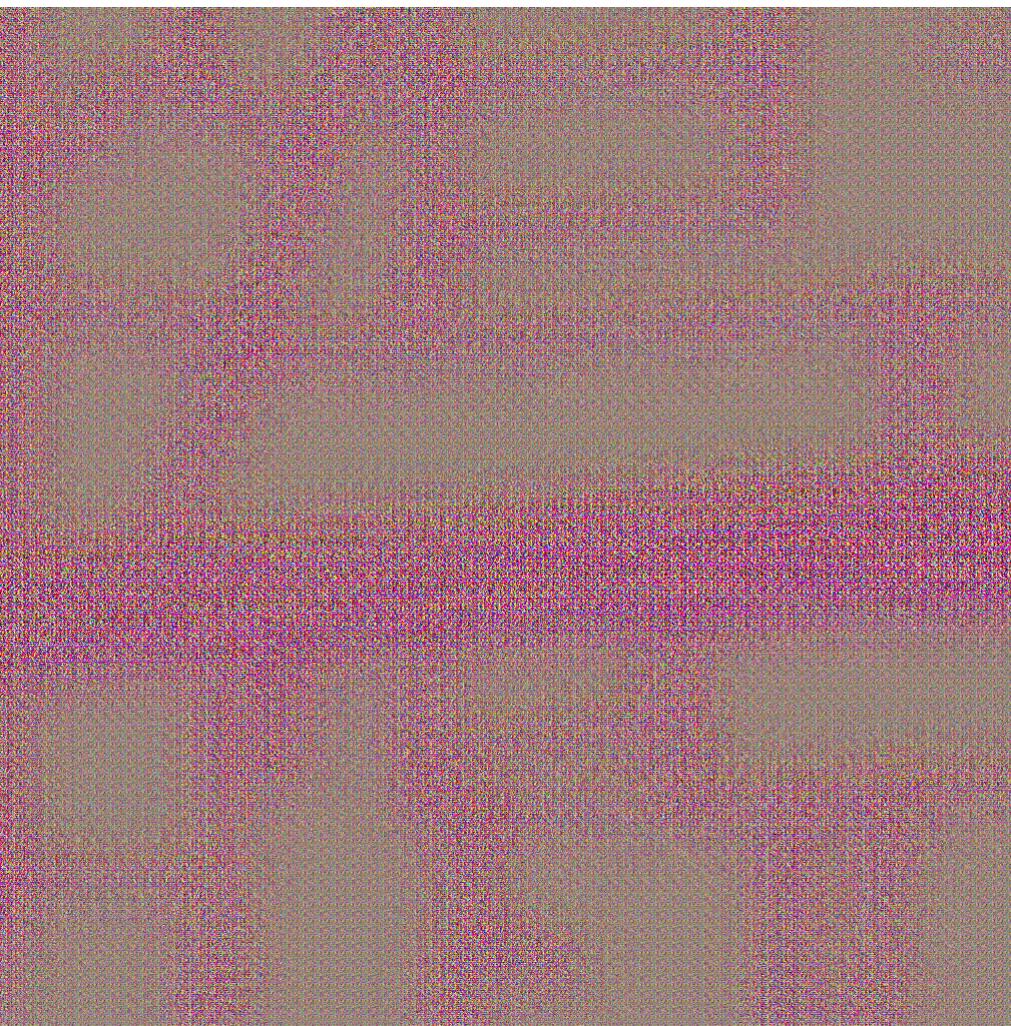
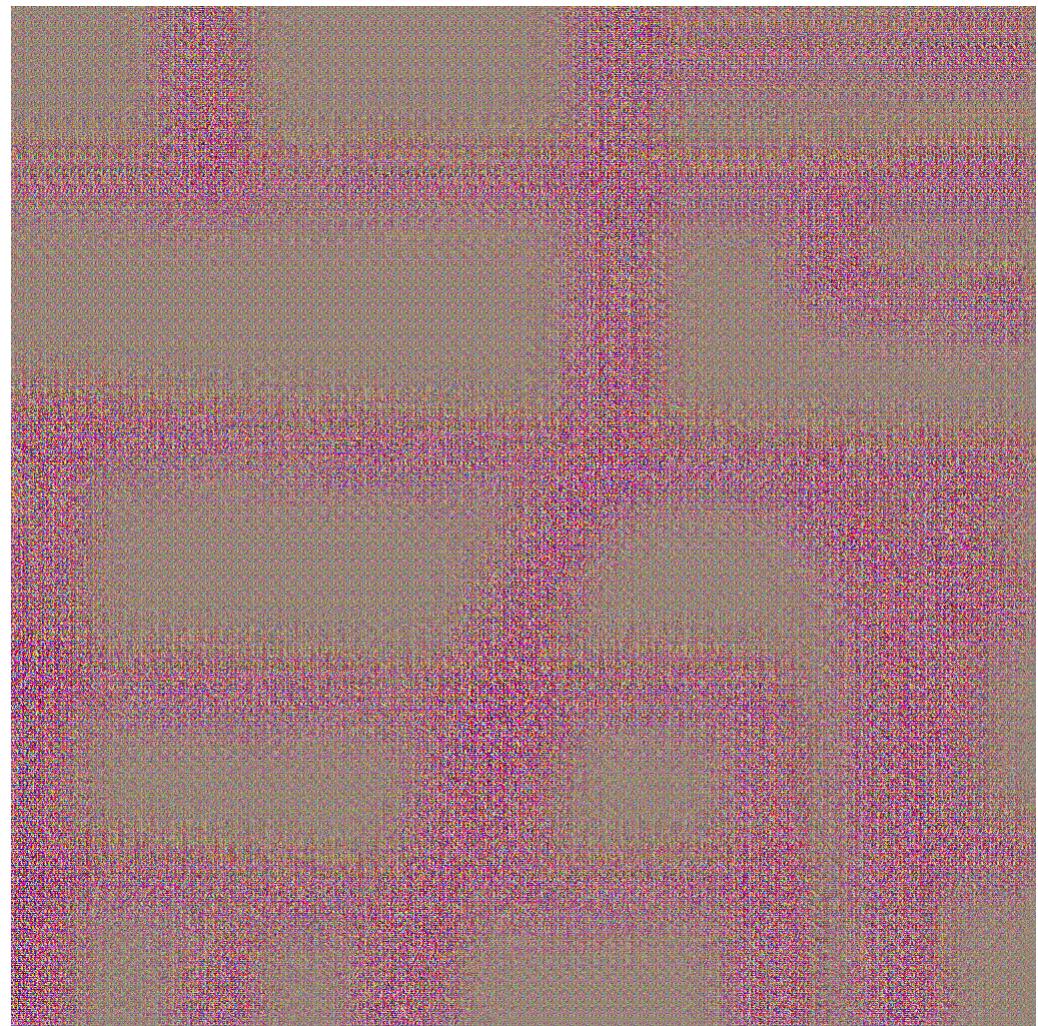
- ▶ studying original repository **done**
- ▶ exploring the provided examples **in progress**
- ▶ digitize street layout of Oberbillwerder **in progress**
- ▶ train GAN on new city developments in Hamburg
- ▶ apply GAN street layout of Oberbillwerder
- ▶ visually compare buildings with masterplan



Example Oberbillwerder

[https://www.oberbillwerder-hamburg.de/projekt/  
masterplan-2022/](https://www.oberbillwerder-hamburg.de/projekt/masterplan-2022/)

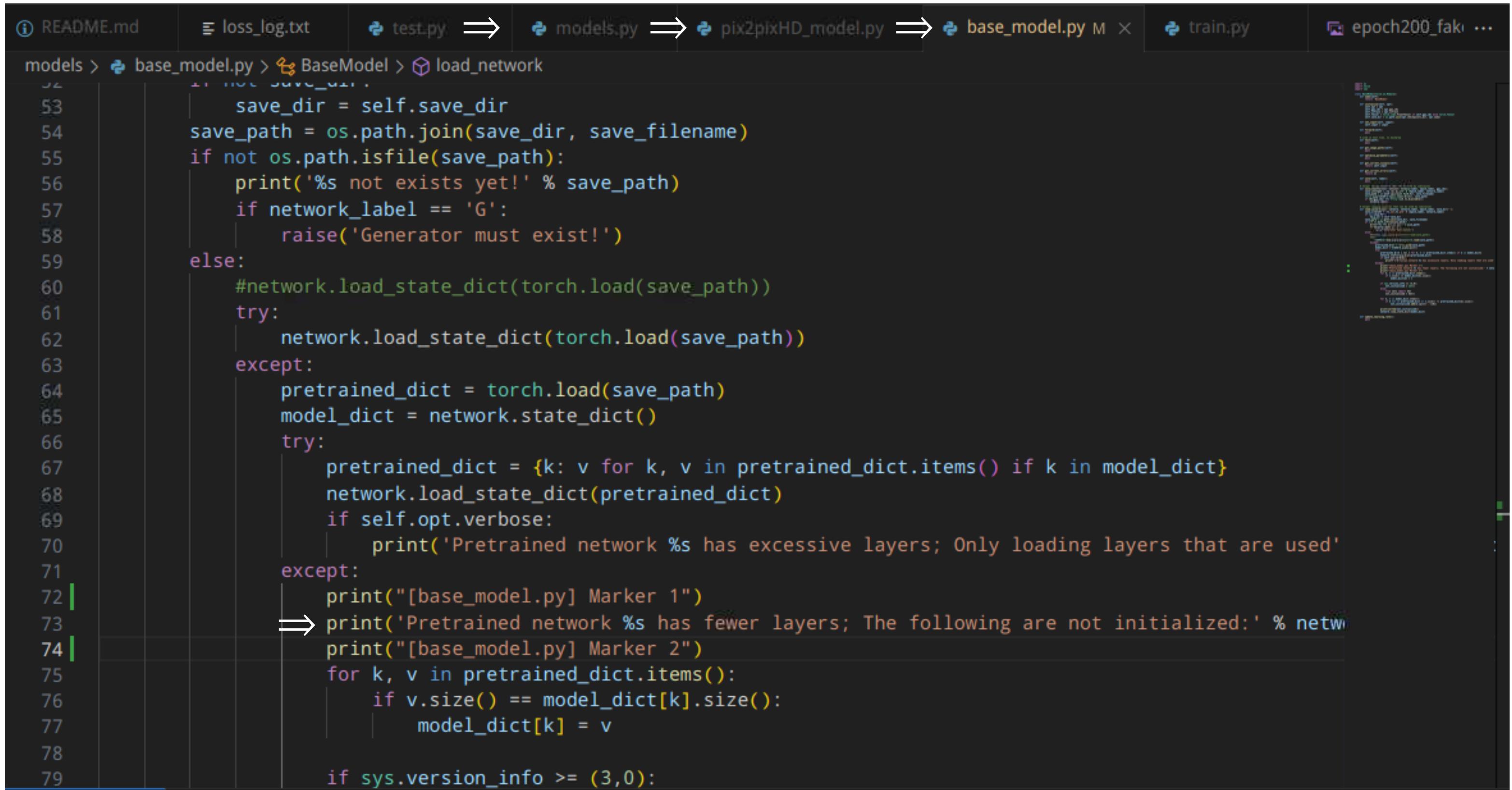
# Results



# Error - console output

```
(25): ConvTranspose2d(1024, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
(26): InstanceNorm2d(512, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
(27): ReLU(inplace=True)
(28): ConvTranspose2d(512, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
(29): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
(30): ReLU(inplace=True)
(31): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
(32): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
(33): ReLU(inplace=True)
(34): ConvTranspose2d(128, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
(35): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
(36): ReLU(inplace=True)
(37): ReflectionPad2d((3, 3, 3, 3))
(38): Conv2d(64, 3, kernel_size=(7, 7), stride=(1, 1))
(39): Tanh()
)
)
[pix2pixHD_model.py] Marker 1
[pix2pixHD_model.py] Marker 2
[base_model.py] Marker 1
Pretrained network G has fewer layers; The following are not initialized: ←
[base_model.py] Marker 2
['model']
[pix2pixHD_model.py] Marker 5
[models.py] Marker 2
[models.py] Marker 3
[models.py] Marker 4
[Test.py] Marker 2
[INFO] Done processing images via test.py
○ (IC) [christopher@ArchWorkstation InstantCITY]$ |
```

# Error



```
① README.md    loss_log.txt  test.py  ⇒  models.py  ⇒  pix2pixHD_model.py  ⇒  base_model.py M X  train.py  epoch200_fak...  
models > base_model.py > BaseModel > load_network  
52     save_dir = self.save_dir  
53     save_path = os.path.join(save_dir, save_filename)  
54     if not os.path.isfile(save_path):  
55         print('%s not exists yet!' % save_path)  
56         if network_label == 'G':  
57             raise('Generator must exist!')  
58         else:  
59             #network.load_state_dict(torch.load(save_path))  
60             try:  
61                 network.load_state_dict(torch.load(save_path))  
62             except:  
63                 pretrained_dict = torch.load(save_path)  
64                 model_dict = network.state_dict()  
65                 try:  
66                     pretrained_dict = {k: v for k, v in pretrained_dict.items() if k in model_dict}  
67                     network.load_state_dict(pretrained_dict)  
68                     if self.opt.verbose:  
69                         print('Pretrained network %s has excessive layers; Only loading layers that are used' % network_label)  
70                 except:  
71                     print("[base_model.py] Marker 1")  
72 ⇒ print('Pretrained network %s has fewer layers; The following are not initialized:' % network_label)  
73 ⇒ print("[base_model.py] Marker 2")  
74     for k, v in pretrained_dict.items():  
75         if v.size() == model_dict[k].size():  
76             model_dict[k] = v  
77  
78     if sys.version_info >= (3,0):  
79         if not save_dir:
```