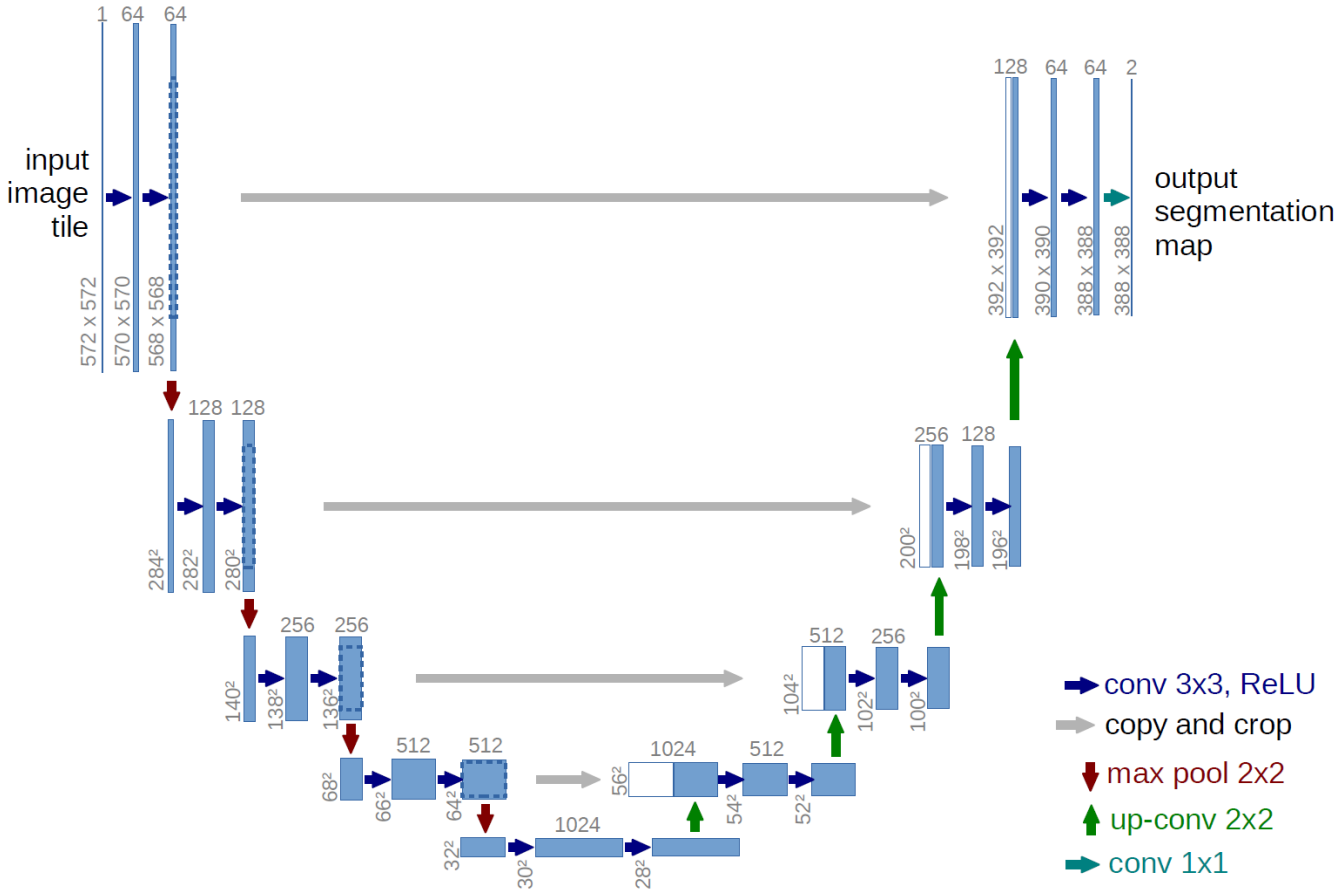
U-Net for cars segmentation

In this work I use `U-Net` to solve the problem of semantic segmentation: `0` — not a car, `1` — car.



In [ ]:

```
# !unzip /data/rvgorb/hw11/data.zip
```

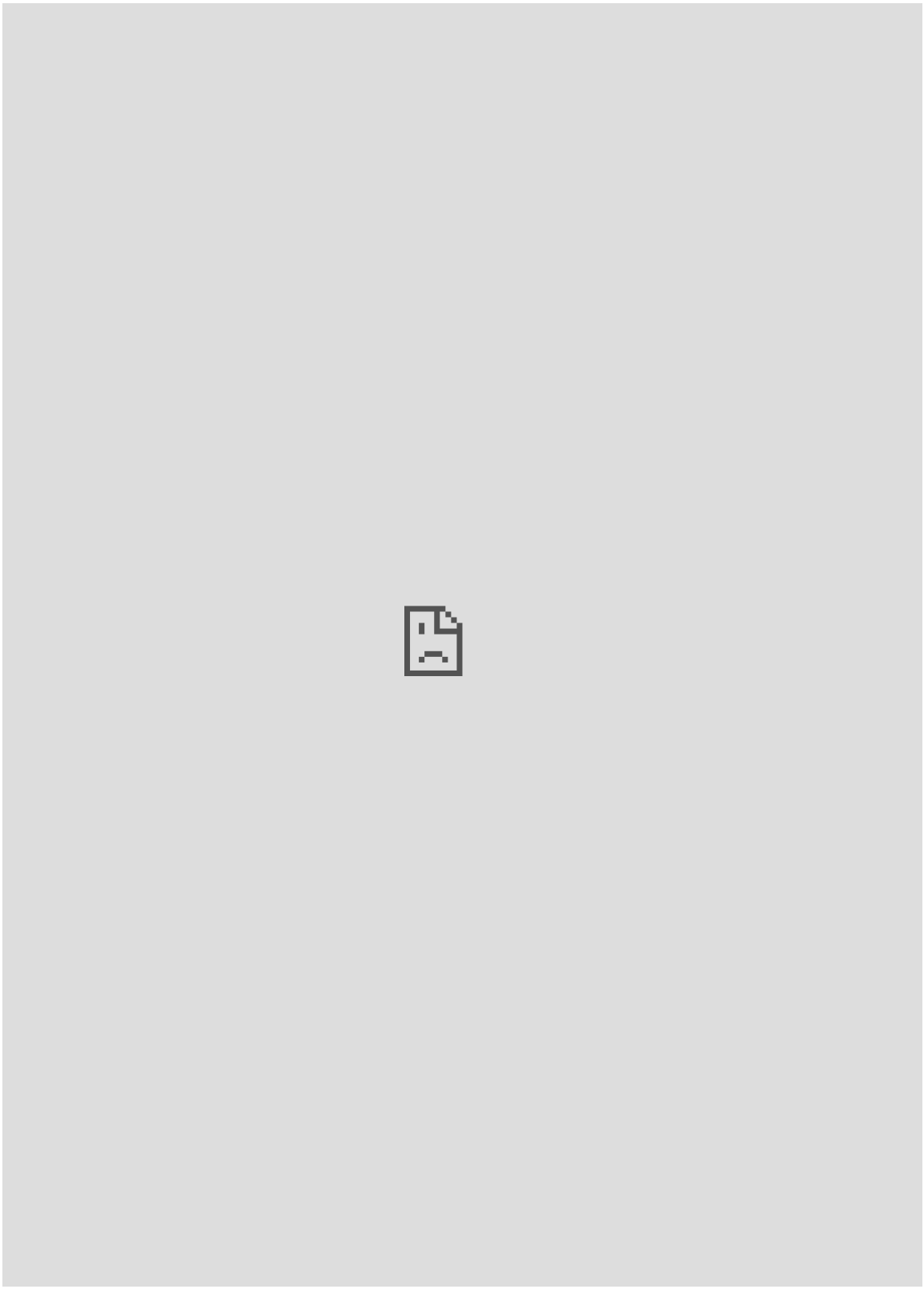
TensorBoard

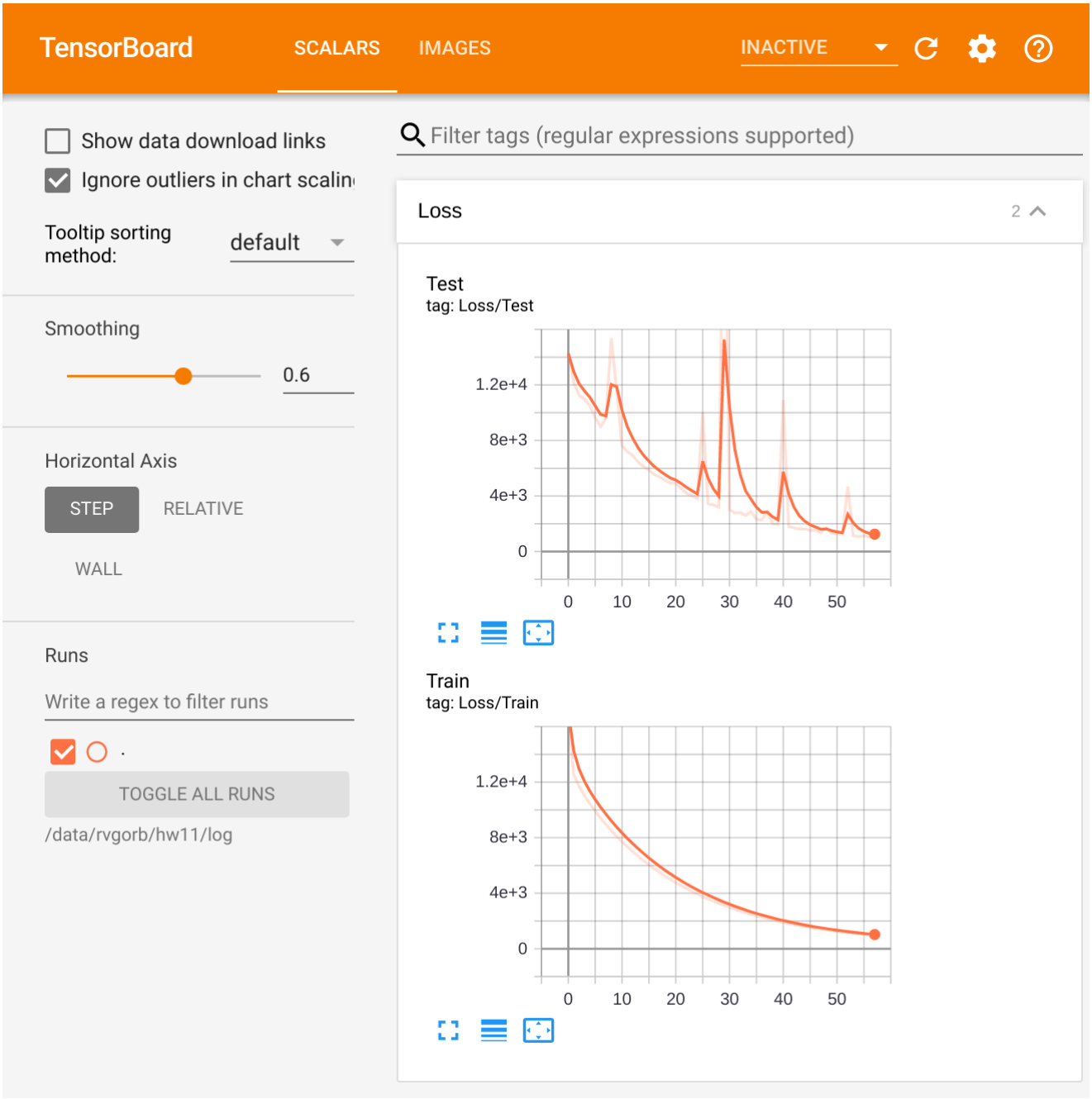
In [168]:

```
!mkdir /data/rvgorb/hw11/log
%load_ext tensorboard
%tensorboard --logdir /data/rvgorb/hw11/log --port 7773
```

```
mkdir: cannot create directory ‘/data/rvgorb/hw11/log’: File exists
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```

```
Reusing TensorBoard on port 7773 (pid 9891), started 0:00:08 ago. (U
se '!kill 9891' to kill it.)
```





Training model

In [85]:

```
!python3 train.py --n_epochs 300 --batch_size 20 --learning_rate 0.00001
```

```
Current epoch:  0
100%|████████████████████████████████████████| 229/229 [02:12<00:0
0,  1.72it/s]
Epoch loss 17139.0282716976
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.37it/s]
Test loss 14279.07421875
Current epoch:  1
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 12553.476886599345
Make test
100%|████████████████████████████████████████| 26/26 [00:08<00:0
0,  3.13it/s]
Test loss 12178.2054933563
Current epoch:  2
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 11687.493869405022
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.42it/s]
Test loss 11254.069451279527
Current epoch:  3
100%|████████████████████████████████████████| 229/229 [08:10<00:0
0,  2.14s/it]
Epoch loss 11027.155598389738
Make test
100%|████████████████████████████████████████| 26/26 [00:08<00:0
0,  3.05it/s]
Test loss 10961.254890501968
Current epoch:  4
100%|████████████████████████████████████████| 229/229 [06:26<00:0
0,  1.69s/it]
Epoch loss 10443.98858146834
Make test
100%|████████████████████████████████████████| 26/26 [00:14<00:0
0,  1.80it/s]
Test loss 10518.217658095473
Current epoch:  5
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.75it/s]
Epoch loss 9905.598956058951
Make test
100%|████████████████████████████████████████| 26/26 [00:09<00:0
0,  2.82it/s]
Test loss 9637.588659571851
Current epoch:  6
100%|████████████████████████████████████████| 229/229 [02:11<00:0
0,  1.75it/s]
Epoch loss 9413.685978438865
Make test
100%|████████████████████████████████████████| 26/26 [00:11<00:0
0,  2.33it/s]
Test loss 8997.551350270669
Current epoch:  7
100%|████████████████████████████████████████| 229/229 [02:11<00:0
0,  1.74it/s]
Epoch loss 8954.252746315502
Make test
```

```
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.33it/s]
Test loss 9596.471702755905
Current epoch:  8
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 8509.588659934498
Make test
100%|████████████████████████████████████████| 26/26 [00:10<00:0
0,  2.45it/s]
Test loss 15376.019454355315
Current epoch:  9
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 8088.47044555131
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.46it/s]
Test loss 11631.133919783464
Current epoch:  10
100%|████████████████████████████████████████| 229/229 [04:13<00:0
0,  1.11s/it]
Epoch loss 7715.233614219433
Make test
100%|████████████████████████████████████████| 26/26 [00:08<00:0
0,  2.98it/s]
Test loss 7628.0539800688975
Current epoch:  11
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 7347.467381959607
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.57it/s]
Test loss 7177.38019039124
Current epoch:  12
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 7004.710101664848
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.68it/s]
Test loss 6926.01711675689
Current epoch:  13
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 6670.103430335698
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.56it/s]
Test loss 6437.695220226378
Current epoch:  14
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 6349.797449849891
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.66it/s]
Test loss 6131.300796628937
Current epoch:  15
100%|████████████████████████████████████████| 229/229 [02:09<00:0
```

0, 1.76it/s]
Epoch loss 6047.6431188591705
Make test
100%|██| 26/26 [00:07<00:0
0, 3.57it/s]
Test loss 5876.022245632382
Current epoch: 16
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]
Epoch loss 5765.688455240174
Make test
100%|██| 26/26 [00:07<00:0
0, 3.61it/s]
Test loss 5548.768877645177
Current epoch: 17
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]
Epoch loss 5497.3382966020745
Make test
100%|██| 26/26 [00:07<00:0
0, 3.58it/s]
Test loss 5369.589367002953
Current epoch: 18
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]
Epoch loss 5235.329307109716
Make test
100%|██| 26/26 [00:07<00:0
0, 3.60it/s]
Test loss 5131.8424889271655
Current epoch: 19
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]
Epoch loss 4996.589031795852
Make test
100%|██| 26/26 [00:07<00:0
0, 3.48it/s]
Test loss 4928.618756151574
Current epoch: 20
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]
Epoch loss 4758.998260098253
Make test
100%|██| 26/26 [00:07<00:0
0, 3.62it/s]
Test loss 4926.532680241142
Current epoch: 21
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 4539.786901269104
Make test
100%|██| 26/26 [00:07<00:0
0, 3.66it/s]
Test loss 4533.99295644685
Current epoch: 22
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]
Epoch loss 4327.693449781659
Make test
100%|██| 26/26 [00:07<00:0
0, 3.62it/s]

```
Test loss 4201.564952940453
Current epoch:  23
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 4124.80806154476
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.52it/s]
Test loss 4023.7020907664864
Current epoch:  24
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.77it/s]
Epoch loss 3948.3663908979256
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.52it/s]
Test loss 3809.519577386811
Current epoch:  25
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 3757.456103302402
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.48it/s]
Test loss 10025.060162401574
Current epoch:  26
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 3590.6462830922487
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.64it/s]
Test loss 3451.1924827755906
Current epoch:  27
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 3420.9982831263646
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.55it/s]
Test loss 3360.088571142963
Current epoch:  28
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 3264.266545271561
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.42it/s]
Test loss 3195.0496585875985
Current epoch:  29
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.77it/s]
Epoch loss 3115.9707253002184
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.47it/s]
Test loss 32176.240926427166
Current epoch:  30
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 2986.154284934498
```

```
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.53it/s]
Test loss 3016.5042984128936
Current epoch:  31
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.77it/s]
Epoch loss 2840.8347792712884
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.58it/s]
Test loss 2787.6638817974904
Current epoch:  32
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.77it/s]
Epoch loss 2710.889703875546
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.40it/s]
Test loss 2815.934708722933
Current epoch:  33
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 2589.024857566867
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.51it/s]
Test loss 2608.8815821850394
Current epoch:  34
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 2474.8730903725436
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.47it/s]
Test loss 2878.3261488065946
Current epoch:  35
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 2368.1971197802945
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.47it/s]
Test loss 2331.8658995140254
Current epoch:  36
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 2262.934787800218
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.31it/s]
Test loss 2267.983013964075
Current epoch:  37
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 2165.288143934225
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.26it/s]
Test loss 2863.8117733452264
Current epoch:  38
```

```
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 2070.647650279749
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.58it/s]
Test loss 2004.1658464566929
Current epoch:  39
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 1980.331762588701
Make test
100%|████████████████████████████████████████| 26/26 [00:09<00:0
0,  2.89it/s]
Test loss 1960.6812215489665
Current epoch:  40
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 1897.90754639738
Make test
100%|████████████████████████████████████████| 26/26 [00:08<00:0
0,  3.10it/s]
Test loss 10936.80144254429
Current epoch:  41
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.75it/s]
Epoch loss 1815.8012810453056
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.45it/s]
Test loss 1809.693588136688
Current epoch:  42
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 1737.1326052469979
Make test
100%|████████████████████████████████████████| 26/26 [00:08<00:0
0,  3.21it/s]
Test loss 1698.5323688176672
Current epoch:  43
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.75it/s]
Epoch loss 1667.940561885917
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.40it/s]
Test loss 1622.9848440575788
Current epoch:  44
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 1598.4927521151747
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.35it/s]
Test loss 1625.4190644992618
Current epoch:  45
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 1530.7861217248908
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
```

0, 3.44it/s]
Test loss 1526.6940437376968
Current epoch: 46
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1474.3926190638647
Make test
100%|██| 26/26 [00:07<00:0
0, 3.50it/s]
Test loss 1536.06330931656
Current epoch: 47
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1409.4291109443232
Make test
100%|██| 26/26 [00:07<00:0
0, 3.52it/s]
Test loss 1356.9093104084645
Current epoch: 48
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1356.1122074576965
Make test
100%|██| 26/26 [00:07<00:0
0, 3.27it/s]
Test loss 1676.7773360605315
Current epoch: 49
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1304.5631004366812
Make test
100%|██| 26/26 [00:07<00:0
0, 3.49it/s]
Test loss 1303.0598778912401
Current epoch: 50
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1252.0678501978712
Make test
100%|██| 26/26 [00:07<00:0
0, 3.28it/s]
Test loss 1284.8295071819637
Current epoch: 51
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1207.565331178357
Make test
100%|██| 26/26 [00:07<00:0
0, 3.53it/s]
Test loss 1262.8276444082185
Current epoch: 52
100%|██| 229/229 [02:10<00:0
0, 1.76it/s]
Epoch loss 1162.28740234375
Make test
100%|██| 26/26 [00:07<00:0
0, 3.39it/s]
Test loss 4681.2366126353345
Current epoch: 53
100%|██| 229/229 [02:09<00:0
0, 1.76it/s]

```
Epoch loss 1116.808769019514
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.41it/s]
Test loss 1144.3880528727855
Current epoch:  54
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 1078.0093796909116
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.44it/s]
Test loss 1081.2053030419538
Current epoch:  55
100%|████████████████████████████████████████| 229/229 [02:09<00:0
0,  1.76it/s]
Epoch loss 1040.513224959061
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.40it/s]
Test loss 1128.466875692052
Current epoch:  56
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 1002.3978643558952
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.38it/s]
Test loss 1069.4753225732038
Current epoch:  57
100%|████████████████████████████████████████| 229/229 [02:10<00:0
0,  1.76it/s]
Epoch loss 968.9999164164847
Make test
100%|████████████████████████████████████████| 26/26 [00:07<00:0
0,  3.55it/s]
Test loss 1173.9578271099901
Current epoch:  58
 26%|██████████                                | 60/229 [00:35<01:3
4,  1.80it/s]^C
```

Results visualization

In [86]:

```
# Imports
%load_ext autoreload
%autoreload 2

import matplotlib.pyplot as plt
import seaborn as sns
import torch
import torch.utils.data as dt
from tqdm.auto import tqdm

from carvana_dataset import CarvanaDataset
from conf import *
from model import SegmenterModel
from stats import *
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [110]:

```
# Datasets
train_data = CarvanaDataset('/data/rvgorb/hw11/train/',
                            '/data/rvgorb/hw11/train_masks/')
test_data = CarvanaDataset('/data/rvgorb/hw11/test/',
                           '/data/rvgorb/hw11/test_masks/',
                           is_train=False)
```

Loading trained model:

In [111]:

```
unet = SegmenterModel(3, 1).cuda()
unet.load_state_dict(
    torch.load("/data/rvgorb/hw11/unet_dump_recent", map_location="cuda:6"))
unet.train(False)
```

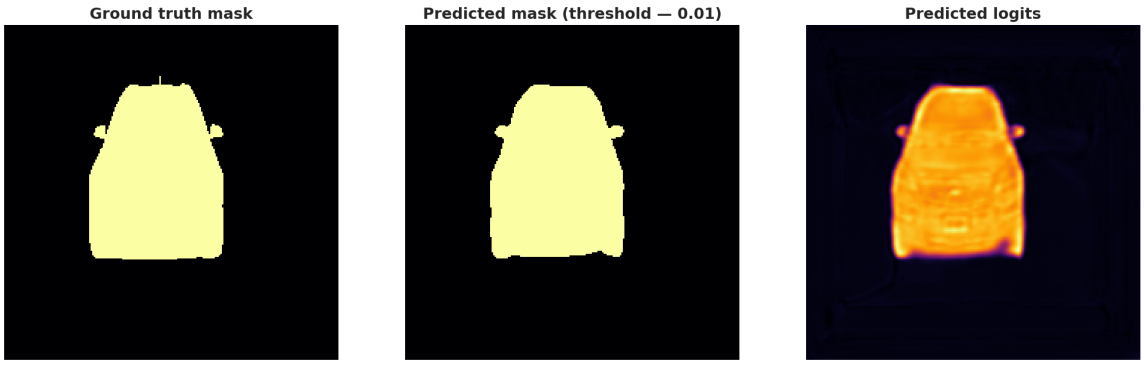
In [162]:

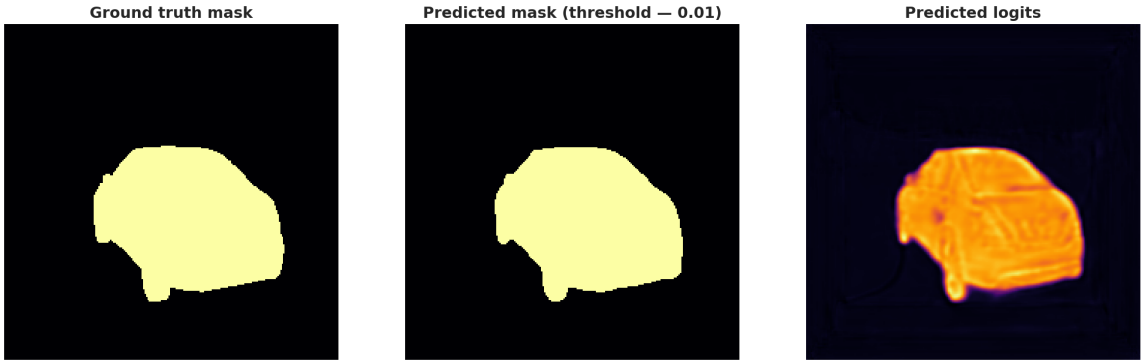
```
def plot_predictions_on_random_image(fitted_model: SegmenterModel,
                                     test_data: CarvanaDataset,
                                     seed: int = None):
    if seed is None:
        rvgen = np.random
    else:
        rvgen = np.RandomState(seed)

    n_img = len(test_data.files)
    img_idx = rvgen.randint(0, n_img)
    img = test_data[img_idx][0]
    true_mask = test_data[img_idx][1].squeeze(0)

    pred_logits = fitted_model.forward(img.unsqueeze(0).cuda())
    pred_logits = pred_logits\
                    .squeeze(0).squeeze(0)\
                    .detach().cpu().numpy()
    pred_mask = pred_logits > 0.01

    images_to_plot = [true_mask, pred_mask, pred_logits]
    ax_titles = ["Ground truth mask",
                 "Predicted mask (threshold — 0.01)",
                 "Predicted logits"]

    fig, axes = plt.subplots(1, 3, figsize=(20, 10))
    for i in range(len(axes)):
        ax = axes[i]
        to_plot = images_to_plot[i]
        ax.imshow(to_plot, cmap="inferno")
        ax.set_title(ax_titles[i], fontsize=15, fontweight="bold")
        ax.set_axis_off()
```

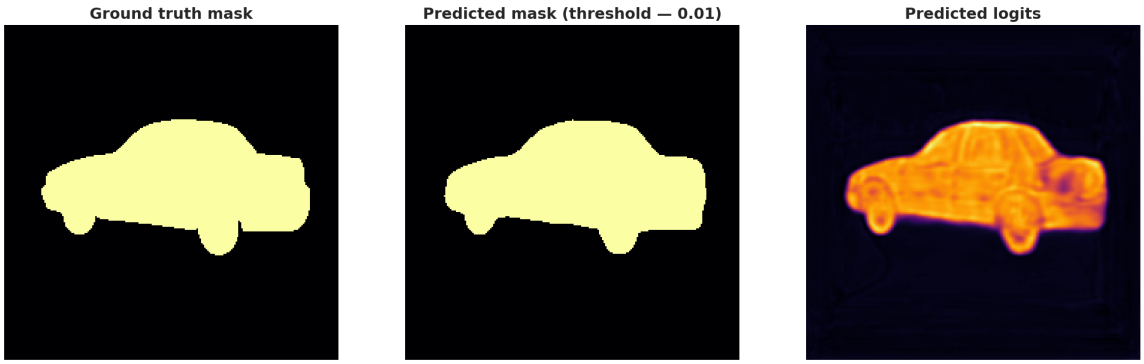
Predicted and ground truth masks comparison:

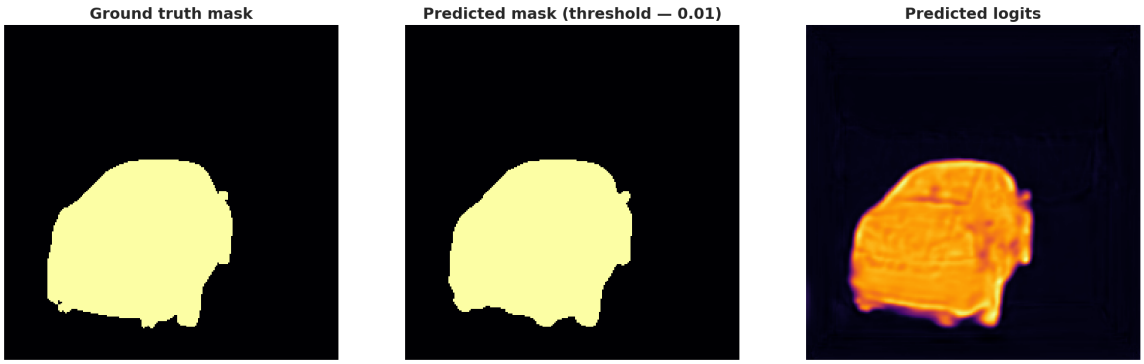
In [163]:

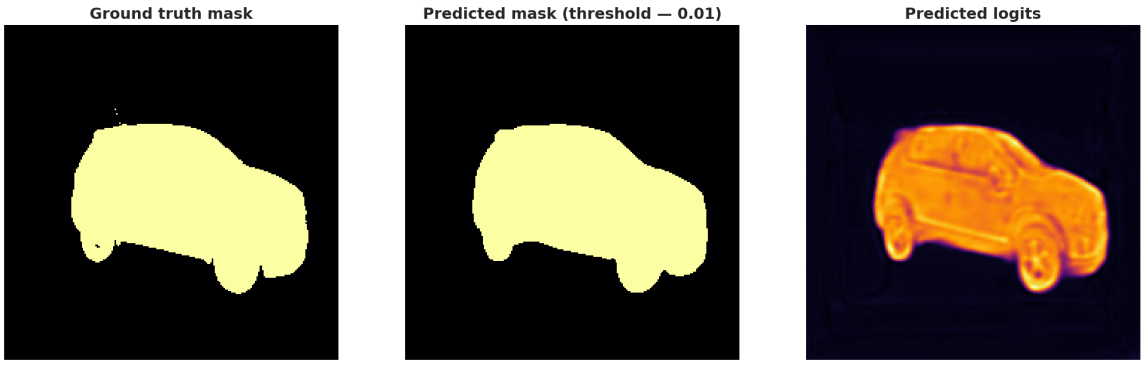
```
for i in range(5):
    plot_predictions_on_random_image(unet, test_data)
```


Ground truth mask
Predicted mask (threshold — 0.01)
Predicted logits


Ground truth mask
Predicted mask (threshold — 0.01)
Predicted logits

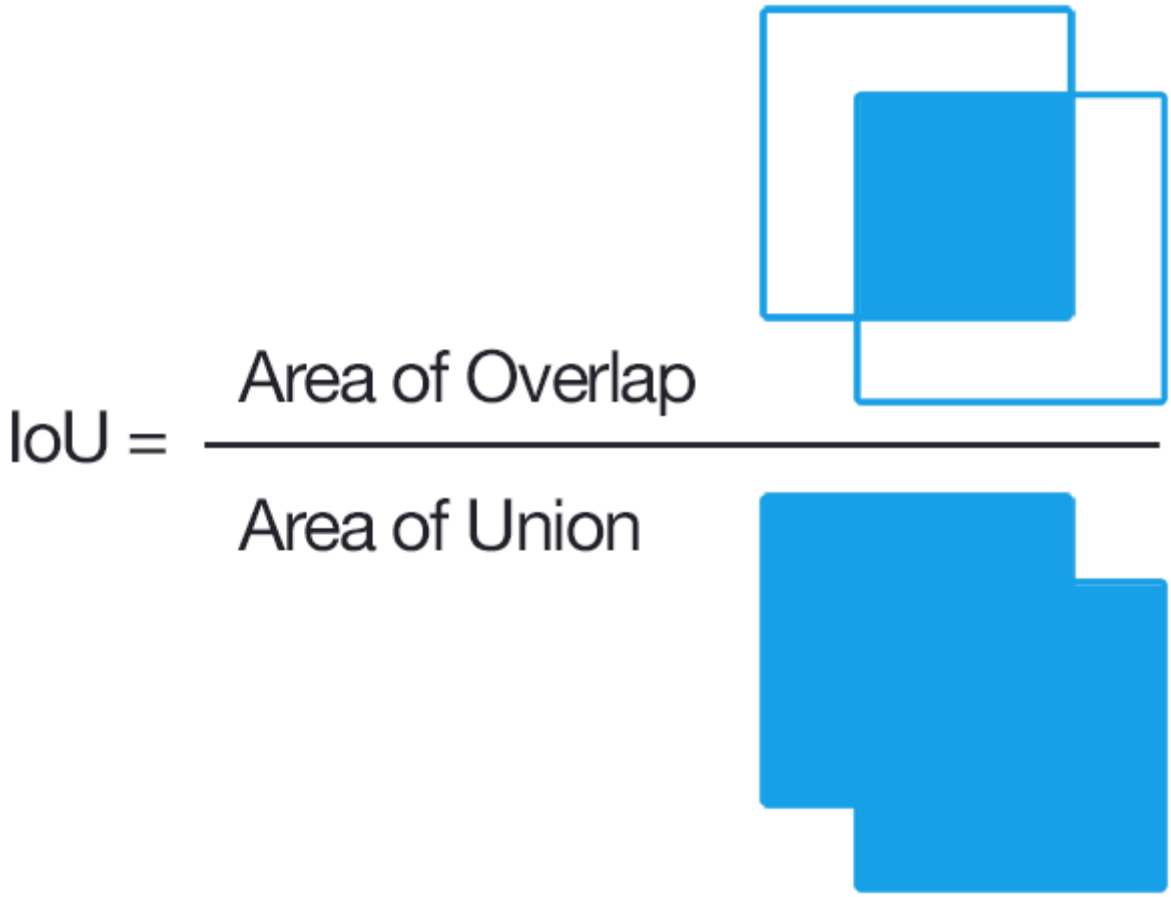

Ground truth mask
Predicted mask (threshold — 0.01)
Predicted logits


Ground truth mask
Predicted mask (threshold — 0.01)
Predicted logits



Quality estimation

`IoU` — *Intersection over Union* — is an evaluation metric used to measure the accuracy of an object detector on a particular dataset.



Examining this equation you can see that Intersection over Union is simply a **ratio**.

In the numerator we compute the **area of overlap** between the predicted bounding box and the ground-truth bounding box.

The denominator is the **area of union**, or more simply, the area encompassed by both the predicted bounding box and the ground-truth bounding box

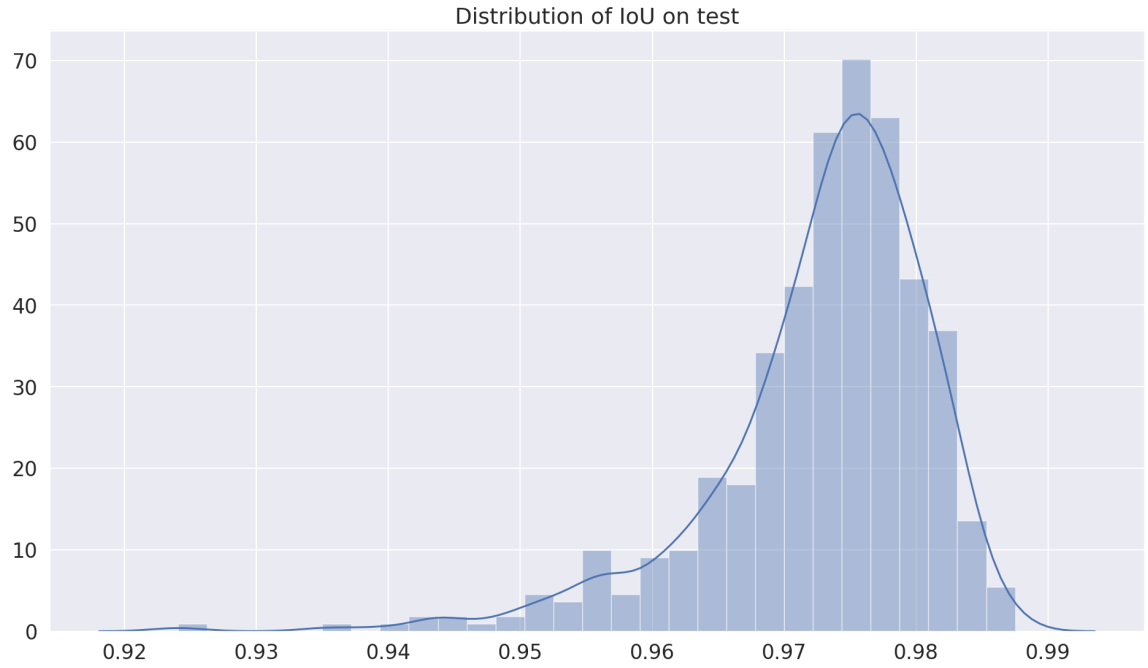
In [154]:

```
def iou_loss(prediction, ground_truth):
    """ Считает IoU по всем элементам батча """
    n_images = len(prediction)
    intersection, union = 0, 0
    for i in range(n_images):
        intersection += np.logical_and(
            prediction[i] > 0, ground_truth[i] > 0).astype(np.float32).sum()
        union += np.logical_or(prediction[i] > 0,
                               ground_truth[i] > 0).astype(np.float32).sum()
    return float(intersection) / union
```

In [159]:

```
# Quality on test
dl_test = dt.DataLoader(test_data, shuffle=False, num_workers=8, batch_size=1)

test_loss = []
for i, (input_batch, target_mask_batch) in enumerate(tqdm(dl_test)):
    with torch.no_grad():
        output_logits_batch = unet(input_batch.to("cuda")).cpu()
    output_mask_batch = output_logits_batch > 0.03
    loss = iou_loss(output_mask_batch.cpu().numpy(),
                    target_mask_batch.cpu().numpy() > 0)
    test_loss.append(loss)
test_loss = np.hstack(test_loss)

plt.figure(figsize=(16, 9))
sns.distplot(test_loss)
_ = plt.title("Distribution of IoU on test")

print(f"Mean value of of IoU on test: {np.mean(test_loss)}")
```

```
Mean value of of IoU on test: 0.9728439083072745
```



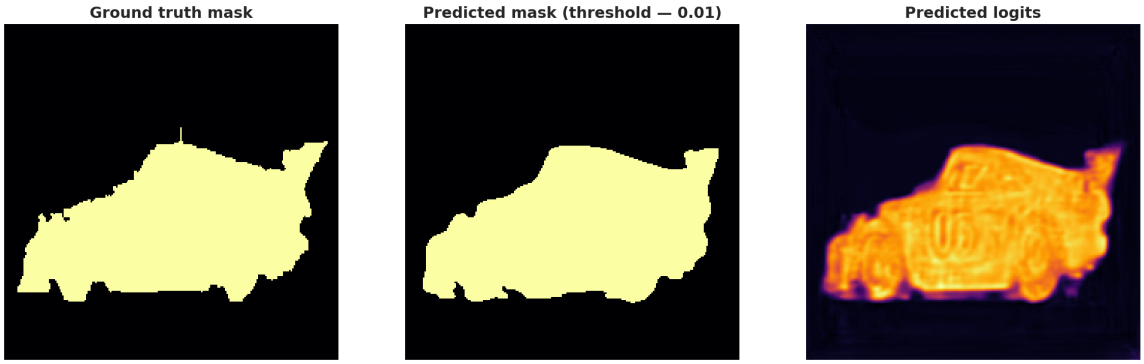
Quality on radom photo from web:

In [94]:

```
# !wget https://i.pinimg.com/originals/2e/49/11/2e49114884cf3d543454a687fb55b3e
7.jpg
# !mkdir example
# !mv 2e49114884cf3d543454a687fb55b3e7.jpg example/
# !mv 2e49114884cf3d543454a687fb55b3e7.gif example_mask/
```

In [164]:

```
example_data = CarvanaDataset('./example/', './example_mask/', is_train=False)
plot_predictions_on_random_image(unet, example_data)
```



In [166]:

```
# IoU computation
dl_example = dt.DataLoader(example_data, shuffle=False,
                           num_workers=8,
                           batch_size=1)


test_loss = []
for i, (input_batch, target_mask_batch) in enumerate(tqdm(dl_example)):
    with torch.no_grad():
        output_logits_batch = unet(input_batch.to("cuda:0")).cpu()
    output_mask_batch = output_logits_batch > 0.
    loss = iou_loss(output_mask_batch.cpu().numpy(), target_mask_batch.cpu().num
py() > 0)
    test_loss.append(loss)

test_loss = np.hstack(test_loss)

print(f"IoU: {np.mean(test_loss)}")
```

```
IoU: 0.9357152344268828
```

Conclusions:

- Achieved 98% `IoU` on test, which is not bad. On random image from web got a little bit worse, because the image contains unseen patterns like reflection from the floor which bothers the model to detect wheels of the car.