

porocilo

January 10, 2023

1 Optični pretok (poročilo)

```
[1]: import matplotlib.pyplot as plt
import numpy as np

import torch
import torchinfo
import torchvision

from mreza import *
```

1.1 Učni primeri

```
[2]: data_gen = nalagalnik_mpi_sintel(looped=False, shuffle=True, nivoji=6)
```

```
[3]: def plot_flow_data(img_data, flow_ref_data, flow_est_data=None):
    img_cur = img_data[0, :3, :, :].detach().numpy().transpose(1, 2, 0)
    img_next = img_data[0, 3:, :, :].detach().numpy().transpose(1, 2, 0)

    fig, ax = plt.subplots(1, 2)
    ax[0].imshow(img_cur)
    ax[0].set_axis_off()
    ax[0].set_title('trenutna')
    ax[1].imshow(img_next)
    ax[1].set_title('naslednja')
    ax[1].set_axis_off()

    if flow_est_data is not None:
        fig, ax = plt.subplots(len(flow_ref_data), 2, squeeze=False)
        ax_data_pack = (ax[:, 0], flow_ref_data), (ax[:, 1], flow_est_data)
    else:
        fig, ax = plt.subplots(len(flow_ref_data), 1, squeeze=False)
        ax_data_pack = (ax[:, 0], flow_ref_data),

    for ax_list, flow_data in ax_data_pack:
        for n, f_img in enumerate(flow_data):
            f_img = torchvision.utils.flow_to_image(f_img[0])
```

```
f_img = f_img.numpy().transpose(1, 2, 0)
ax_list[n].imshow(f_img, interpolation='nearest')
ax_list[n].set_axis_off()
ax_list[n].set_title(f'pretok, locljivost {f_img.shape[:2]}')
```

```
[4]: img_data, flow_data = next(data_gen)
      plot_flow_data(img_data, flow_data)

      img_data, flow_data = next(data_gen)
      plot_flow_data(img_data, flow_data)

      img_data, flow_data = next(data_gen)
      plot_flow_data(img_data, flow_data)

      img_data, flow_data = next(data_gen)
      plot_flow_data(img_data, flow_data)
```

trenutna



naslednja



pretok, locljivost (384, 1024)



pretok, locljivost (192, 512)



pretok, locljivost (96, 256)



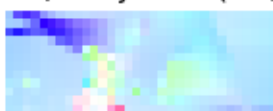
pretok, locljivost (48, 128)



pretok, locljivost (24, 64)



pretok, locljivost (12, 32)



trenutna



naslednja



pretok, ločljivost (384, 1024)



pretok, ločljivost (192, 512)



pretok, ločljivost (96, 256)



pretok, ločljivost (48, 128)



pretok, ločljivost (24, 64)



pretok, ločljivost (12, 32)



trenutna



naslednja



pretok, locljivost (384, 1024)



pretok, locljivost (192, 512)



pretok, locljivost (96, 256)



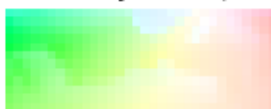
pretok, locljivost (48, 128)



pretok, locljivost (24, 64)



pretok, locljivost (12, 32)



trenutna



naslednja



pretok, locljivost (384, 1024)



pretok, locljivost (192, 512)



pretok, locljivost (96, 256)



pretok, locljivost (48, 128)



pretok, locljivost (24, 64)



pretok, locljivost (12, 32)



1.2 Pregled mreže

```
[7]: #of_model = torch.load('./flow_net_simple_tiny.pt', map_location=torch.  
    ↪device('cpu'))  
of_model = torch.load('./flow_net_simple_small_2.pt', map_location=torch.  
    ↪device('cpu'))  
torchinfo.summary(of_model)
```

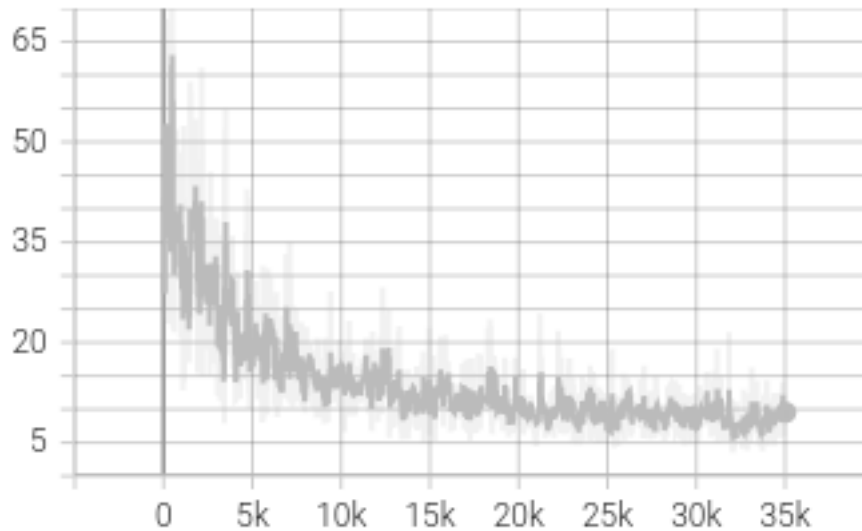
```
[7]: =====  
Layer (type:depth-idx)                   Param #  
=====
```

FlowNetSimple	--
DownBlock: 1-1	--
Conv2d: 2-1	1,760
BatchNorm2d: 2-2	64
ReLU: 2-3	--
Conv2d: 2-4	9,248
BatchNorm2d: 2-5	64
ReLU: 2-6	--
MaxPool2d: 2-7	--
DownBlock: 1-2	--
Conv2d: 2-8	18,496

BatchNorm2d: 2-9	128
ReLU: 2-10	--
Conv2d: 2-11	36,928
BatchNorm2d: 2-12	128
ReLU: 2-13	--
MaxPool2d: 2-14	--
DownBlock: 1-3	--
Conv2d: 2-15	73,856
BatchNorm2d: 2-16	256
ReLU: 2-17	--
Conv2d: 2-18	147,584
BatchNorm2d: 2-19	256
ReLU: 2-20	--
MaxPool2d: 2-21	--
DownBlock: 1-4	--
Conv2d: 2-22	295,168
BatchNorm2d: 2-23	512
ReLU: 2-24	--
Conv2d: 2-25	590,080
BatchNorm2d: 2-26	512
ReLU: 2-27	--
MaxPool2d: 2-28	--
ConvTranspose2d: 1-5	131,200
ConvTranspose2d: 1-6	18
Conv2d: 1-7	514
ConvTranspose2d: 1-8	65,600
ConvTranspose2d: 1-9	18
Conv2d: 1-10	518
ConvTranspose2d: 1-11	16,416
ConvTranspose2d: 1-12	18
Conv2d: 1-13	262
Conv2d: 1-14	134
=====	
Total params: 1,389,738	
Trainable params: 1,389,738	
Non-trainable params: 0	
=====	

1.2.1 Učenje

Izguba (povprečje evklidske razadlje, PEP) v času učenja.

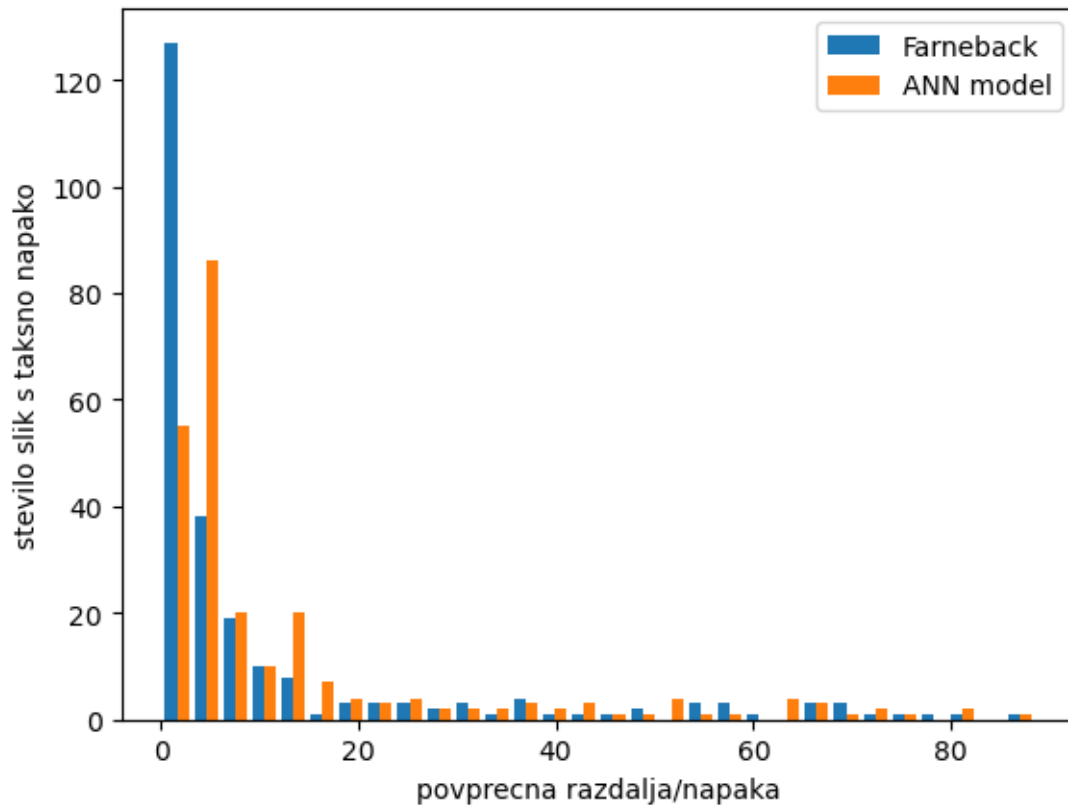


1.3 Rezultati

1.3.1 Rezultati na testnih podatkih

```
[124]: fb_errs = np.load('./farneback_test_errors.npy')  
       of_model_errs = np.load('./flow_net_simple_small_2_test_errors.npy')
```

```
[130]: plt.hist([fb_errs, of_model_errs], bins=30, label=['Farneback', 'ANN model'])  
       plt.legend()  
       plt.xlabel('povprecna razdalja/napaka')  
       plt.ylabel('stevilo slik s taksno napako');
```

```
[126]: print(f'farneback: {np.mean(fb_errs):.2f} +- {np.std(fb_errs):.2f}')
       print(f'ANN model: {np.mean(of_model_errs):.2f} +- {np.std(of_model_errs):.2f}')
```

```
farneback: 10.98 +- 18.26
ANN model: 13.45 +- 18.30
```

1.3.2 Primeri

```
[66]: import cv2
```

```
[110]: # testni podatki
data_gen = nalagalnik_mpi_sintel(looped=False, shuffle=True, nivoji=4,
                                posnetki=[
                                    'cave_2', 'cave_4',
                                    'mountain_1',
                                    'shaman_2', 'shaman_3',
                                ],
                                )
```

```
[111]: img_data, flow_ref = next(data_gen)
       flow_est = of_model(img_data)
```

```

plot_flow_data(img_data, flow_ref, flow_est)

img_cur = np.uint8(img_data[0, :3, :, :].detach().numpy().transpose(1, 2, 0).
    ↳mean(2)*255)
img_next = np.uint8(img_data[0, 3:, :, :].detach().numpy().transpose(1, 2, 0).
    ↳mean(2)*255)
flow_fb = cv2.calcOpticalFlowFarneback(img_cur, img_next, None, 0.5, 6, 15, 3,
    ↳5, 1.2, 0)
plot_flow_data(img_data, [flow_ref[0]], [torch.from_numpy(flow_fb.
    ↳transpose(2,0,1))[np.newaxis,...]])

```

trenutna



naslednja



pretok, locljivost (432, 1024)



pretok, locljivost (216, 512)



pretok, locljivost (108, 256)



pretok, locljivost (54, 128)



pretok, locljivost (432, 1024)



pretok, locljivost (216, 512)



pretok, locljivost (108, 256)



pretok, locljivost (54, 128)



trenutna



naslednja



pretok, locljivost (432, 1024)



pretok, locljivost (432, 1024)



```
[112]: img_data, flow_ref = next(data_gen)
flow_est = of_model(img_data)
plot_flow_data(img_data, flow_ref, flow_est)

img_cur = np.uint8(img_data[0, :3, :, :].detach().numpy().transpose(1, 2, 0).
    ↳mean(2)*255)
img_next = np.uint8(img_data[0, 3:, :, :].detach().numpy().transpose(1, 2, 0).
    ↳mean(2)*255)
flow_fb = cv2.calcOpticalFlowFarneback(img_cur, img_next, None, 0.5, 6, 15, 3,
    ↳5, 1.2, 0)
plot_flow_data(img_data, [flow_ref[0]], [torch.from_numpy(flow_fb.
    ↳transpose(2,0,1))[np.newaxis,...]])
```

trenutna



naslednja



pretok, locljivost (432, 1024)



pretok, locljivost (216, 512)



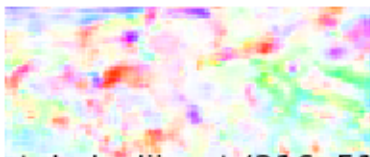
pretok, locljivost (108, 256)



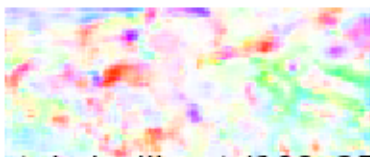
pretok, locljivost (54, 128)



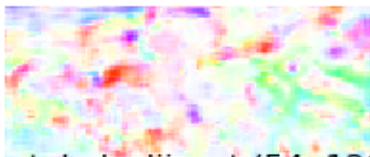
pretok, locljivost (432, 1024)



pretok, locljivost (216, 512)



pretok, locljivost (108, 256)



pretok, locljivost (54, 128)



trenutna



naslednja



pretok, locljivost (432, 1024)



pretok, locljivost (432, 1024)



```
[113]: img_data, flow_ref = next(data_gen)
flow_est = of_model(img_data)
plot_flow_data(img_data, flow_ref, flow_est)

img_cur = np.uint8(img_data[0, :3, :, :].detach().numpy().transpose(1, 2, 0).
    ↳mean(2)*255)
img_next = np.uint8(img_data[0, 3:, :, :].detach().numpy().transpose(1, 2, 0).
    ↳mean(2)*255)
flow_fb = cv2.calcOpticalFlowFarneback(img_cur, img_next, None, 0.5, 6, 15, 3,
    ↳5, 1.2, 0)
plot_flow_data(img_data, [flow_ref[0]], [torch.from_numpy(flow_fb.
    ↳transpose(2,0,1))[np.newaxis,...]])
```

trenutna



naslednja



pretok, locljivost (432, 1024)



pretok, locljivost (432, 1024)



pretok, locljivost (216, 512)



pretok, locljivost (216, 512)



pretok, locljivost (108, 256)



pretok, locljivost (108, 256)



pretok, locljivost (54, 128)



pretok, locljivost (54, 128)





1.3.3 Lasten posnetek

Pripravljen video posnetek za optični pretok ocenjen z nevronske mrežo (my_video_OF.avi) ter z Farneback metodo (my_video_FB.avi).

```
[149]: vid_reader = cv2.VideoCapture('./my_video-1.mkv')
f, img_cur = vid_reader.read()
H, W = img_cur.shape[:2]
vid_writer = cv2.VideoWriter('./my_video_FB.avi', fourcc=cv2.
    ↳VideoWriter_fourcc(*'MJPG'), fps=30, frameSize=(W, H))
while True:
    f, img_next = vid_reader.read()

    if not f:
        break

    img_data = np.zeros((*img_cur.shape[:2], 6), dtype=np.float32)
    img_data[:, :, :3] = img_cur/255.
    img_data[:, :, 3:] = img_next/255.
    img_data = img_data.transpose(2, 0, 1)[np.newaxis, ...]

    flow_est = of_model(torch.from_numpy(img_data))[0]

    #flow_est = cv2.calcOpticalFlowFarneback(np.uint8(img_cur.mean(2)), np.
    ↳uint8(img_next.mean(2)), None, 0.5, 6, 15, 3, 5, 1.2, 0)
    #flow_est = torch.from_numpy(flow_est.transpose(2, 0, 1)[np.newaxis, ...])
```

```
of_img = torchvision.utils.flow_to_image(flow_est)[0]
of_img = of_img.numpy().transpose(1, 2, 0)

vid_writer.write(of_img)

vid_reader = None
vid_writer = None
```

1.4 Zaključek

Rezultati učenja niso bili preveč dobri. Metoda Farneback, implementirana v OpenCV, je dajala bistveno boljše rezultate. Morda je težava v prekomernem prileganju učnih vzorcev (ang. overfitting). Uporaba validacijske zbirke bi bila tukaj zelo pomembna.