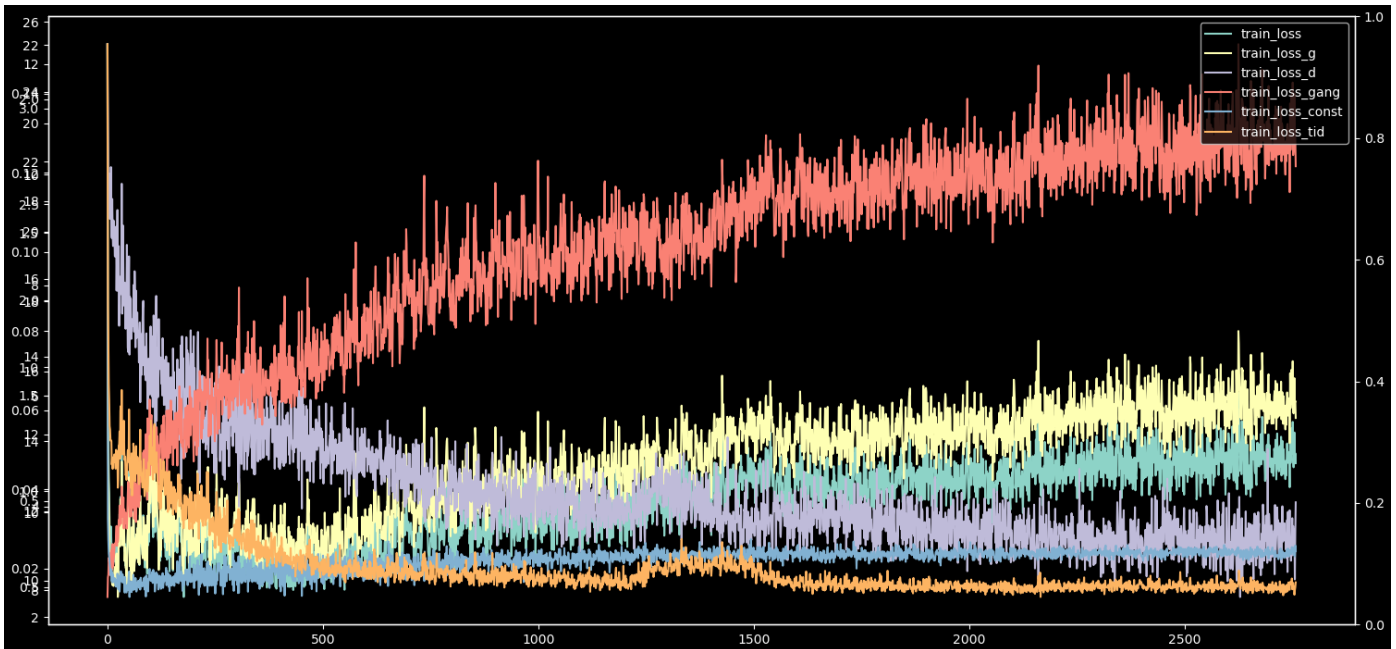


Meeting #10 Report

DTN-GAN Horse2Zebra:

Labākie rezultāti ko pagaidām varēja dabūt ar dtn-gan horse-to-zebra:





HDF5 Data processor priekš FER datu kopas.

```

1 import pandas as pd
2 import numpy as np
3 import h5py
4
5 ### Prepare data to be stored
6 df = pd.read_csv('./fer/fer2013/fer2013.csv', sep=',', header=None)
7

```

```

8 values = df.values
9 count = len(values)
10 col_name = values[0]
11
12 labels = []
13 imgs = []
14 usage = []
15
16 for i in range(1, count):
17     # filter out Happiness (3) label
18     label = int(values[i][0])
19     if label == 6:
20         img_flat = np.fromstring(values[i][1], dtype=int, sep=' ')
21         img = np.reshape(img_flat, (48, 48))
22         imgs.append(img)
23         labels.append(label)
24         usage.append(values[i][2])
25
26 # Sort into training and testing usages
27 train_imgs = []
28 train_labels = []
29 test_imgs = []
30 test_labels = []
31 for idx, u in enumerate(usage):
32     if u == 'Training':
33         train_imgs.append(imgs[idx])
34         train_labels.append(labels[idx])
35     else:
36         test_imgs.append(imgs[idx])
37         test_labels.append(labels[idx])
38
39 train_imgs = np.array(train_imgs)
40 train_labels = np.array(train_labels)
41 test_imgs = np.array(test_imgs)
42 test_labels = np.array(test_labels)
43
44 ### Create h5py object
45 train_shape = (len(train_imgs), 48, 48)
46 test_shape = (len(test_imgs), 48, 48)
47
48 # open a hdf5 file and create arrays
49 hdf5_path = './fer_48_s.hdf5'
50 f = h5py.File(hdf5_path, mode='w')
51

```

```

52 ### Store data in .hdf5 file
53 f.create_dataset("train_labels", (len(train_labels),), np.uint8)
54 f["train_labels"][...] = train_labels
55 f.create_dataset("test_labels", (len(test_labels),), np.uint8)
56 f["test_labels"][...] = test_labels
57 f.create_dataset("train_imgs", train_shape, np.float32)
58 f["train_imgs"][...] = train_imgs
59 f.create_dataset("test_imgs", test_shape, np.float32)
60 f["test_imgs"][...] = test_imgs
61
62 f.close()

```

Dataset class:

Šī datu kopa nav tik liela tāpēc uzreiz ieliek atmiņā.

```

1 import h5py
2 import numpy as np
3 import torch
4 from torch.utils import data
5
6
7 class Dataset(data.Dataset):
8     def __init__(self, file_path, mode, img_size, transform):
9         super().__init__()
10        self.data = []
11        self.mode = mode
12        self.img_size = img_size
13        self.file_path = file_path
14        self.transform = transform
15
16        file = h5py.File(self.file_path)
17
18        labels = file[f"{self.mode}_labels"]
19        imgs = file[f"{self.mode}_imgs"]
20        self.data = list(zip(imgs, labels))
21
22    def __getitem__(self, index):
23        x, y = self.data[index]
24        y = np.expand_dims(y, axis=0)
25
26        x = np.reshape(x, (1, self.img_size, self.img_size))
27        x = torch.FloatTensor(x)
28        x = self.transform(x)

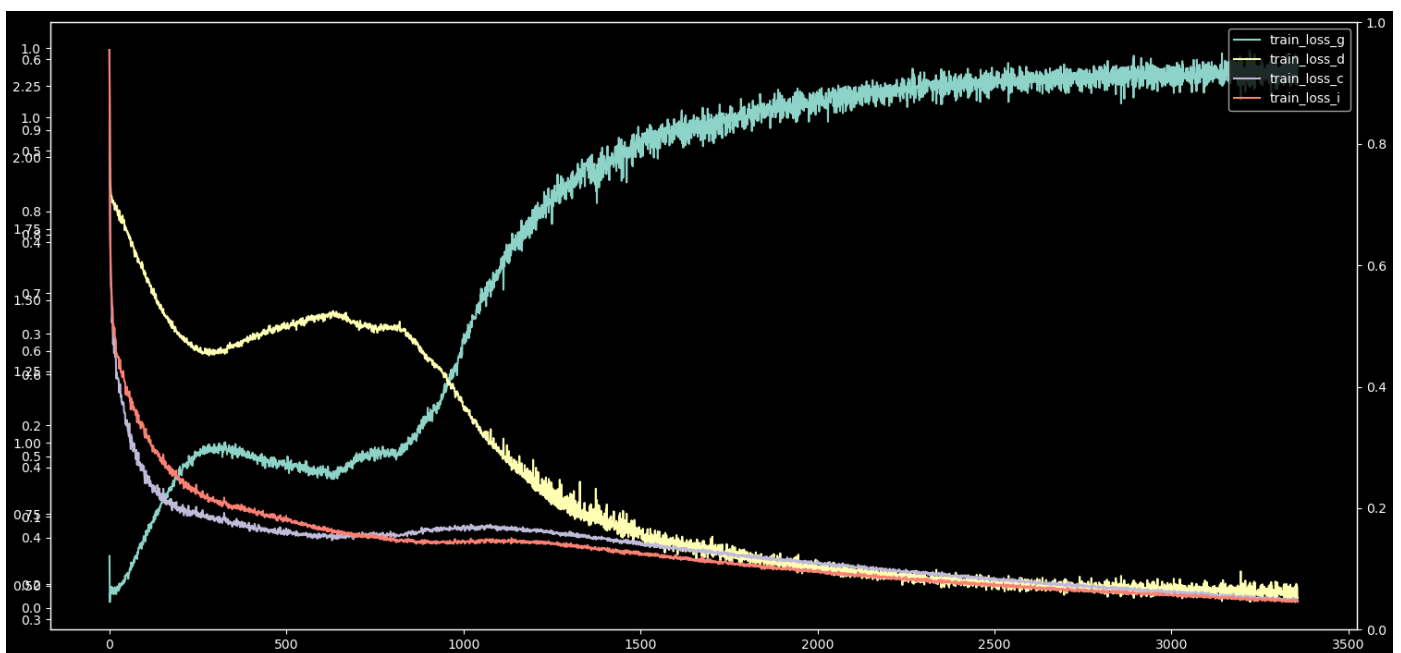
```

```

29     mid = 255.0 / 2
30     x = (x - mid) / mid
31
32     y = torch.from_numpy(y)
33
34     return x, y
35
36 def __len__(self):
37     return len(self.data)

```

CycleGAN rezultāti pārnesot Happiness emociju uz neutral:



1.rinda Source; 2.rinda Fake Target; 3.rinda Recovered Source

Epoch: 3354 samples:



Epoch: 3356 samples:



Epoch: 2734 samples:



Problēmas pieteikties RaFD seju datu kopai (nav nekādas atbildes uz epasta apstiprināšanu). Alternatīva: Flickr Faces Dataset. (FFHQ), kuram ir emotion labels.

<https://github.com/NVlabs/ffhq-dataset>

<https://openprojectrepo.com/project/DCGM-ffhq-features-dataset-python-deep-learning>
