

Кумпан Виктор

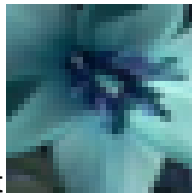
@SkylarPro - tg

AutoEncoder

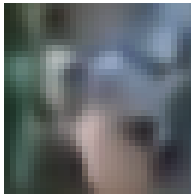
img2img without learning



predict -> gt



img2img 60 epoch learning

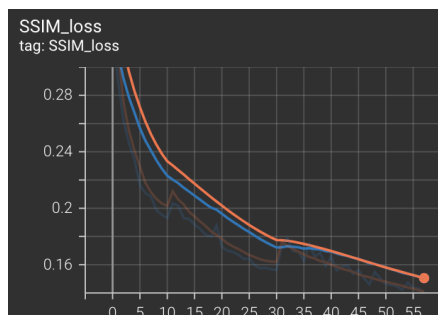
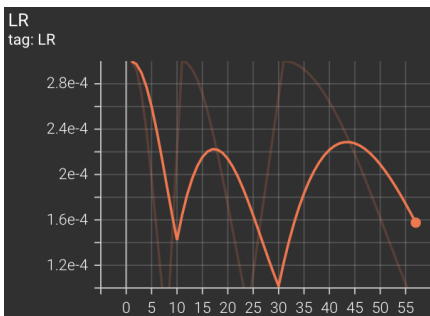
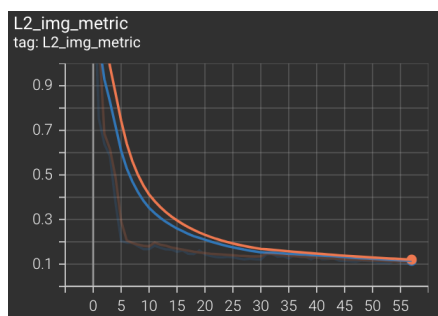
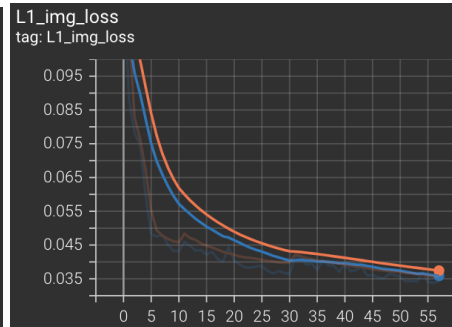
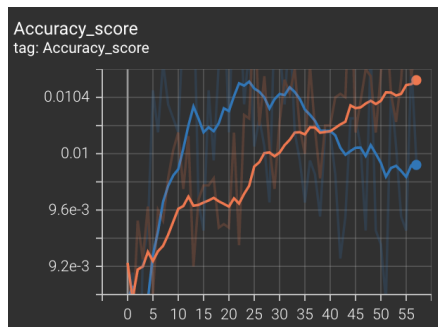


predict -> gt

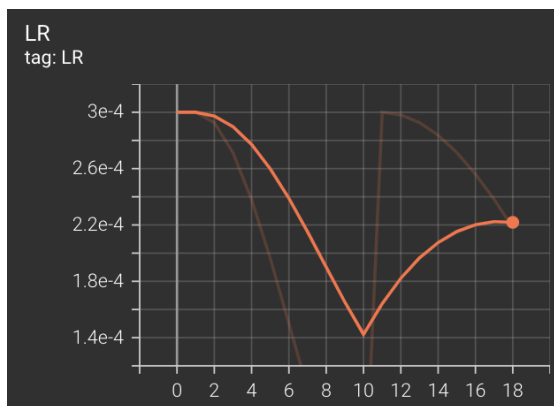
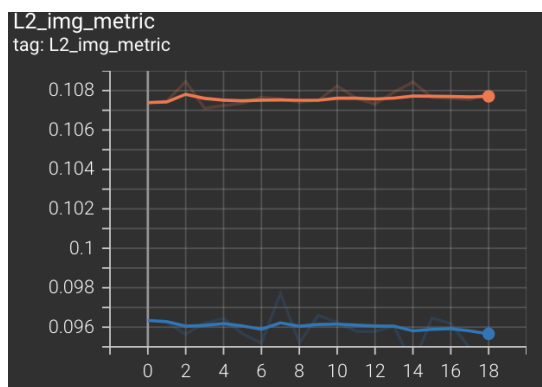
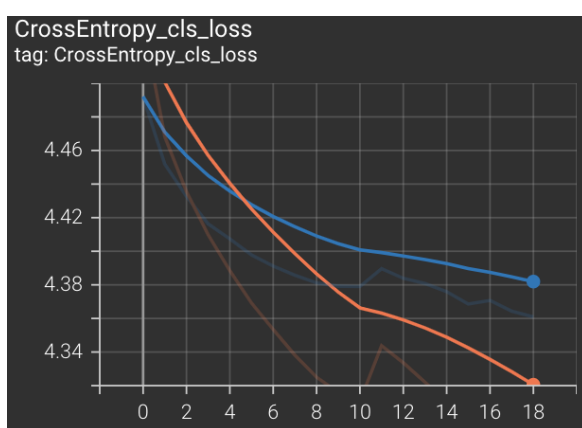
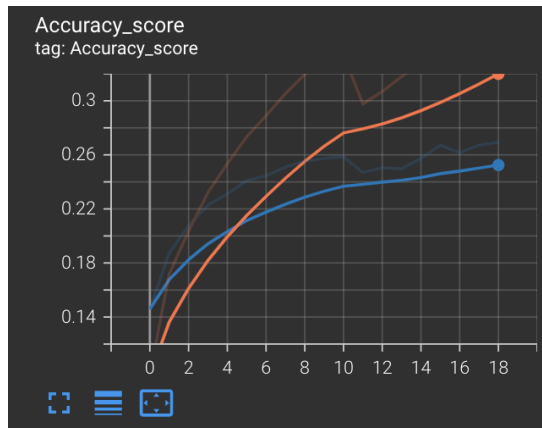


Обучение img2img (пункт 3 из гита)

60 эпох переобучения нет и оптимизация идет плавно, можно учить дальше



Обучение img2cls (пункт 5 из гита)



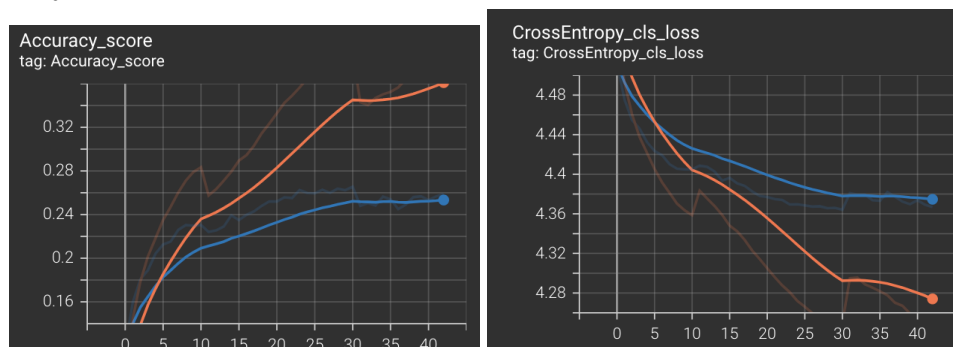
Inference

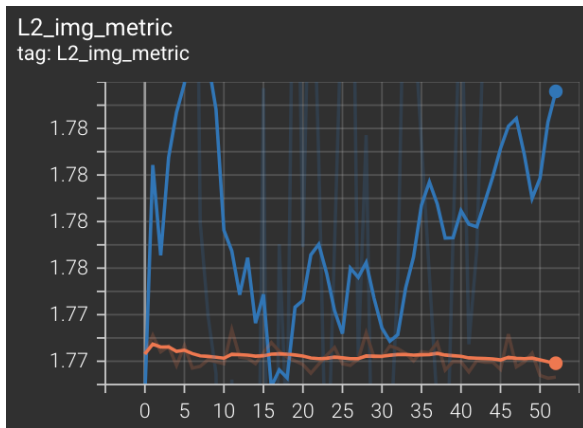
```
(base) → MILTestTasks git:(master) × python3 inference/inference.py
Load weight in model...
0it [00:00, ?it/s]/home/viktor.kumpan/.miniconda3/lib/python3.8/site-packages/torch/nn/modules/softmax.py:52: DeprecationWarning: The `size` attribute for softmax has been deprecated. Change the call to include dim=X as an argument.
input = module(input)
625it [00:03, 201.41it/s]
Create_confusion
2022-04-27 12:28:28: Epoch 0/0, Accuracy_score: 0.25300;
```

apple	58	2	1	0	0	1	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0
aquarium_fish	0	40	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0
baby	5	5	13	0	0	7	0	0	0	0	0	0	0	0	2	0	0	1	0	0	2	0
bear	0	0	0	0	0	0	0	3	2	1	0	0	2	1	6	0	0	1	1	5	0	14
beaver	0	0	0	0	0	0	0	3	0	0	0	0	0	0	2	1	0	0	4	4	1	6
bed	3	0	2	0	0	26	0	0	1	0	0	0	0	1	1	0	0	3	0	1	2	0
bee	0	1	1	0	0	2	0	5	0	0	0	0	0	0	9	0	0	1	4	2	1	0
beetle	0	0	0	0	0	0	0	28	0	1	0	0	0	0	6	0	0	0	4	2	0	4
bicycle	0	1	1	0	0	0	0	2	26	1	0	0	2	0	1	0	0	5	0	0	0	1
bottle	1	0	2	0	0	0	0	2	2	45	0	0	0	1	2	1	0	2	0	0	0	0
bowl	7	2	0	0	0	5	0	5	1	0	0	0	0	2	2	0	0	0	0	1	2	2
boy	1	3	6	0	0	1	0	0	5	2	0	2	0	0	2	0	0	0	0	0	2	2
bridge	0	0	0	0	0	1	0	0	1	0	0	0	32	0	0	0	0	6	1	0	0	1
bus	1	0	0	0	0	0	0	0	1	0	0	0	3	25	2	1	0	0	0	0	1	4
butterfly	1	2	1	0	0	2	0	1	0	0	0	0	0	0	29	0	0	0	7	3	1	0
camel	1	1	0	0	0	1	0	1	0	0	0	0	2	0	1	4	0	4	0	8	0	1
can	2	0	1	0	0	2	0	3	1	15	0	0	0	4	1	0	0	0	2	1	3	1
castle	0	0	0	0	0	1	0	0	2	0	0	0	5	0	0	4	0	53	0	2	0	0
caterpillar	0	5	1	0	1	1	0	2	0	3	0	0	0	1	7	0	0	0	36	0	0	0
cattle	0	0	0	0	1	1	0	1	1	0	0	0	0	0	2	4	0	1	2	29	2	0
chair	1	1	2	0	0	4	0	1	2	1	0	0	0	1	0	0	0	0	0	0	53	0
chimpanzee	0	0	0	0	0	0	0	2	2	0	0	0	0	1	3	0	0	0	0	2	0	45

Получаем оверфит, для борьбы с этим выкрутим `weight_decay` на $1e-05$ и добавим `dropout`

Результат:





```
((base) → MILTestTasks git:(master) × python3 inference/inference.py
Load weight in model...
0it [00:00, ?it/s]/home/viktor.kumpan/.miniconda3/lib/python3.8/site-pa
rch/nn/modules/container.py:119: UserWarning: Implicit dimension choice
max has been deprecated. Change the call to include dim=X as an argumen
input = module(input)
625it [00:03, 199.35it/s]
Create_confusion
2022-04-27 13:37:06: Epoch 0/0, Accuracy_score: 0.23400;
```

apple	63	3	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
aquarium_fish	1	38	1	0	0	1	0	0	2	0	0	0	0	2	0	0	0	0	1	1	0	0
baby	1	1	24	0	0	5	0	0	0	0	0	0	1	2	2	0	0	1	1	0	2	2
bear	0	0	2	0	0	1	0	5	1	1	0	0	1	0	2	0	0	1	4	4	0	10
beaver	1	0	1	0	0	0	0	4	0	0	0	0	0	0	4	0	0	1	1	2	0	3
bed	1	1	3	0	0	29	0	0	0	1	0	0	2	2	1	0	0	1	1	0	0	0
bee	0	1	2	0	0	0	0	7	0	1	0	0	0	0	9	0	0	0	6	1	2	0
beetle	0	1	0	0	0	1	0	36	1	0	0	0	0	0	7	0	0	0	8	1	0	1
bicycle	0	1	0	0	0	0	0	0	35	2	0	0	1	3	1	0	0	0	2	0	0	0
bottle	1	0	1	0	0	0	0	3	1	39	0	0	2	0	1	0	0	1	2	1	1	0
bowl	7	3	2	0	0	3	0	4	0	0	0	0	0	2	2	0	0	2	0	2	1	0
boy	1	4	10	0	0	5	0	1	0	2	0	0	2	1	3	0	0	0	0	0	5	1
bridge	0	0	0	0	0	0	0	0	1	0	0	0	35	1	0	0	0	4	0	0	2	2
bus	1	2	0	0	0	2	0	0	0	1	0	0	3	23	0	0	0	1	0	0	0	0
butterfly	1	4	1	0	0	1	0	3	0	1	0	0	0	0	21	0	0	1	3	2	2	0
camel	1	2	2	0	0	3	0	1	1	0	0	0	1	1	1	0	0	4	2	5	0	0
can	3	0	1	0	0	3	0	5	3	8	0	0	0	3	0	0	0	0	2	2	3	3
castle	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	33	0	1	0	0
caterpillar	0	3	1	0	0	1	0	5	0	0	0	0	0	2	5	0	0	0	19	1	1	0
cattle	0	0	0	0	0	1	0	3	1	0	0	0	0	3	5	0	0	3	1	24	1	1
chair	3	0	2	0	0	2	0	1	2	1	0	0	1	0	0	0	0	0	0	0	54	1
chimpanzee	0	0	0	0	0	1	0	1	0	0	0	0	1	1	2	0	0	0	1	2	0	43

Ловим все равно переобучение, для устранения этого нужны трюки которые я описал в ШАГ6

И если прокомментировать conf matrix то моделька не может отличить похожие объекты (bear от chimpanzee)

ResNetUnet

img2img with out learning



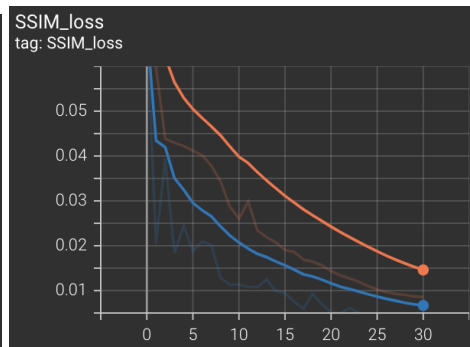
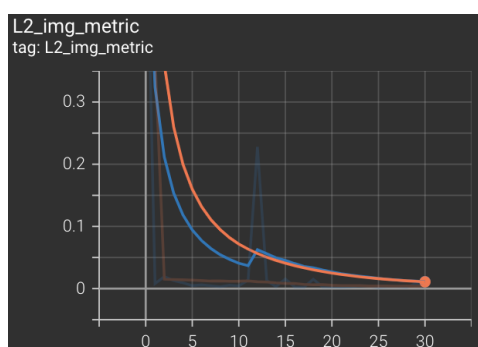
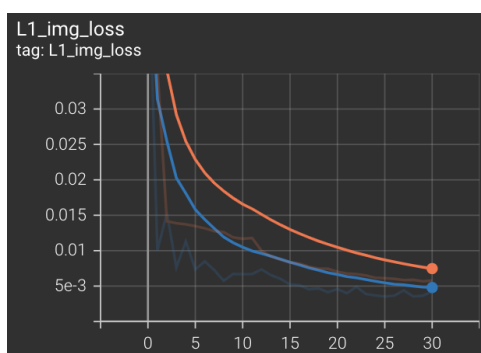
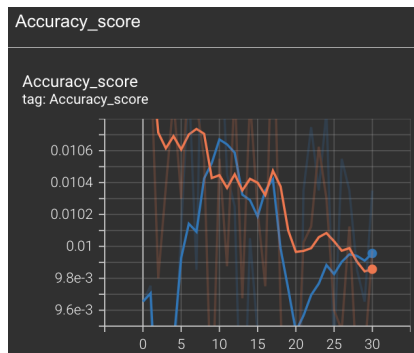
img2img 30 epoch learning



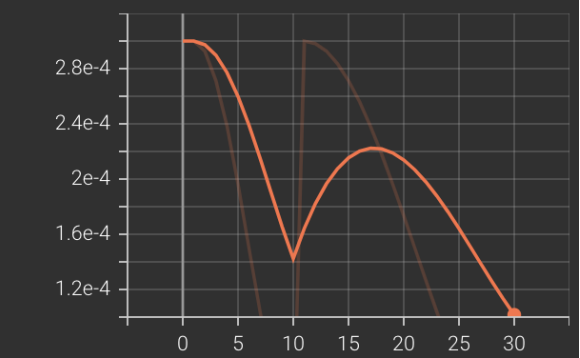
img2img обучение (30 эпох) результат намного лучше чем AutoEncoder благодаря скипам)

По графикам видно, что еще можно учить и учить. (учить можно всегда)

Важно отметить что l1,l2,SSIM на порядки ниже чем на AutoEncoder, скипы решают)

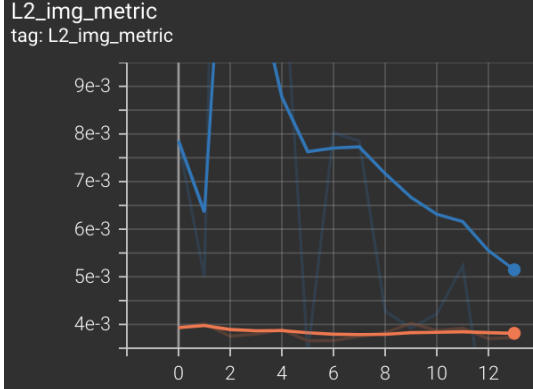
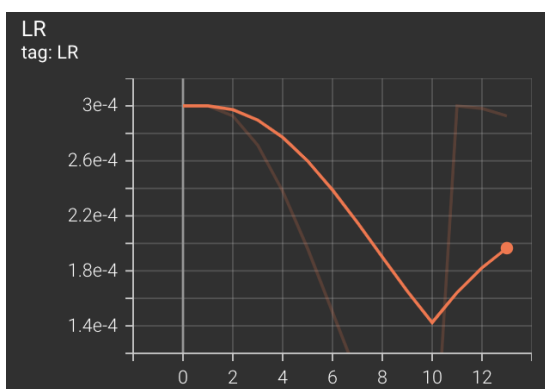
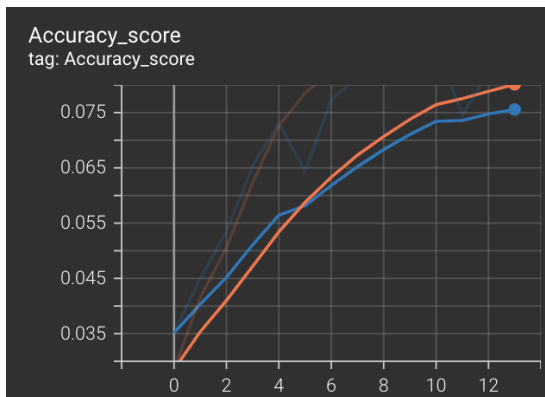


LR
tag: LR



Обучение img2cls (пункт 5 из гита)

Сразу выкрутим weight decay на $1e-05$ и добавим dropout



Прикрутить голову классификации на эмбединги encoder'a (АЕ использующий скипы нормально не удастся так как большое количество информации идет именно через них и перед сеткой нет особой необходимости делать репрезентативное латентное пространство с последнего слоя encoder'a. Это видно по графикам обучения. Точность крайне маленькая

```
00h_00.p
(base) → MILTestTasks git:(master) × python3 inference/inference.py
Load weight in model...
0it [00:00, ?it/s]/home/viktor.kumpan/.miniconda3/lib/python3.8/site-p
oice for softmax has been deprecated. Change the call to include dim=X
input = module(input)
625it [00:06, 102.76it/s]
Create_confusion
2022-04-27 13:30:50: Epoch 0/0, Accuracy_score: 0.08300;
```

apple	31	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	12	0
aquarium_fish	1	25	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
baby	1	4	12	0	0	0	0	0	0	1	0	0	0	0	0	0	3	1	0	0	9	3
bear	0	0	0	0	0	2	0	11	0	1	0	0	0	0	0	0	1	1	0	0	4	13
beaver	0	1	0	0	0	0	0	7	0	1	0	0	0	0	0	0	1	0	0	0	0	12
bed	1	2	2	0	0	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	17	1
bee	2	0	2	0	0	0	0	4	0	0	0	0	0	0	0	0	2	1	0	0	6	1
beetle	3	0	0	0	0	2	0	10	0	0	0	0	0	0	0	0	1	1	0	0	7	6
bicycle	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	11	1
bottle	4	3	2	0	0	4	0	1	0	2	0	0	0	0	0	0	1	2	0	0	17	5
bowl	4	7	6	0	0	3	0	2	0	1	0	0	0	0	0	0	1	0	0	0	5	3
boy	2	4	6	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8	4
bridge	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	11	0	0	1	3
bus	0	2	0	0	0	1	0	3	0	1	0	0	0	0	0	0	0	4	0	0	1	3
butterfly	0	9	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	5
camel	0	0	1	0	0	2	0	3	0	1	0	0	0	0	0	0	1	3	0	0	2	4
can	2	0	2	0	0	2	0	1	0	1	0	0	0	0	0	0	3	0	0	0	16	4
castle	2	1	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	22	0	0	7	1
caterpillar	0	8	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	4
cattle	0	2	0	0	0	0	0	6	0	2	0	0	0	0	0	0	0	0	0	0	3	10
chair	1	1	0	0	0	2	0	1	0	2	0	0	0	0	0	0	2	0	0	0	33	1
chimpanzee	2	1	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	29

Шаг 6

Улучшения, которые можно сделать

1. Для улучшения латентной репрезентации можно добавить в входное изображение еще один канал и его заполнить меткой класса (даст мощный буст)
2. Для тренировки `img2img` можно добавить аугментацию шумом и сдвиги (так как сетка не устойчива к данным преобразованиям), с разными скейлами, собирать мозаику из данных это все даст тоже мощный буст. Да и вообще можно изменять по контрасту, цвету и прочие ауги. Для классификации можно делать кропы, шифты
3. Можно переделать FPN подход чтобы лучше прокидывать градиенты по сетке (для "AutoEncoder" оч поможет)
4. Можно поиграть с коэффициентами перед потерями, но это уже больше про другую задачу ну и с самими потерями.
5. Погуглить оптимальный `lr` для этого сета))))
6. Можно добавить в `pipeline` PCA визуализацию латентного пр-ва для более глубокого анализа, больше метрик `img2img`

Подведение итогов

1. Написан полный `pipeline` для тренировки, валидации, теста, логи модели
2. Созданы две архитектуры АЕ и много всего)