

► Import

```
[ ] ↳ 2 cells hidden
```

► Functions and setups

```
[ ] ↳ 5 cells hidden
```

► Data setup

```
[ ] ↳ 7 cells hidden
```

► Model training

```
[ ] ↳ 7 cells hidden
```

► Evaluate Functions

```
[ ] ↳ 1 cell hidden
```

▼ Eval

```
1 # set up to predict every images we have
2 PATH = "./drive/MyDrive/github_capstone/msca_capstone/model/model.pt"
3 net = SiameseNetwork().cuda()
4 net.load_state_dict(torch.load(PATH))
5 net.eval()
6
7 datafolder = './drive/MyDrive/github_capstone/msca_capstone/data/manual-clusters/categoriz
8 testings = []
9 for label in os.listdir(datafolder):
10     for image in os.listdir(datafolder + label):
11         testings.append({
12             'dir': datafolder + label + "/" + image,
13             'true_label': label,
14         })
15 testings = pd.DataFrame(testings)
```

```
1 # Get the predicated and true label
2 pred = []
3 true = []
```

```

4
5 from tqdm import tqdm
6 for i, row in tqdm(testimgs.iterrows()):
7     test_image = row['dir']
8     pred.append(siamese_eval(test_image, datafolder, net, n=3, metric='as'))
9     true.append(row['true_label'])

0it [00:00, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py
  cpuset_checked))
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:1331: UserWarning: dropout
  warnings.warn(warn_msg)
143it [04:34, 1.92s/it]

```

```

1 # Number of images with each labels
2 top_3_pred = pd.DataFrame({'pred':pred,
3                             'true':true})
4
5 top_3_pred['true'].value_counts()

```

```

6      44
5      23
2      22
3      19
1      17
4      11
7       7
Name: true, dtype: int64

```

```

1 # Top 1 accuracy for each label
2 correct = []
3 for i, row in top_3_pred.iterrows():
4     if row['true'] in row['pred'][0]:
5         correct.append(1)
6     else:
7         correct.append(0)
8 top_3_pred['correct'] = correct
9 print(top_3_pred.groupby(['true'])['correct'].mean())
10
11 # Top 1 accuracy for overall
12 top_3_pred['correct'].mean()

```

```

true
1      0.4706
2      0.0000
3      0.0526
4      0.0909
5      0.7391
6      0.3864
7      0.1429
Name: correct, dtype: float64
0.3146853146853147

```

```
1 # Top 2 accuracy for each label
2 correct = []
3 for i, row in top_3_pred.iterrows():
4     if row['true'] in row['pred'][:2]:
5         correct.append(1)
6     else:
7         correct.append(0)
8 top_3_pred['correct'] = correct
9 print(top_3_pred.groupby(['true'])['correct'].mean())
10
11 # Top 2 accuracy for overall
12 top_3_pred['correct'].mean()
```

true

```
1    0.7647
2    0.2273
3    0.0526
4    0.0909
5    0.9130
6    0.9545
7    0.1429
```

Name: correct, dtype: float64
0.5874125874125874

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 4:10 PM

● ✕