Image Similarity using the TLL dataset

(Totally-Looks-Like)

# Done

* Obtained the Totally-Looks-Like Dataset
* Read more into the paper’s setup <https://arxiv.org/pdf/1803.01485v3.pdf>
* Made a custom Dataloader class that returns a triplet of images: 2 positives and a negative
* Created a feature extractor from a pretrained resnet18 to be used in the Siamese Network
* Found in the following paper that I could learn (through fine-tunning) the similarity function instead of using cosine similarity: <https://paperswithcode.com/paper/learning-an-adaptation-function-to-assess>
* Read the following paper about how to visualize what the network focuses on when deciding whether 2 input images are similar <https://arxiv.org/pdf/1312.6034.pdf>

# ToDo

* Filter the dataset to have only pairs of images with no faces (i.e. obtain the TLLobj dataset from the paper <https://arxiv.org/pdf/1803.01485v3.pdf> )
* Evaluate how effective is using directly (without fine-tuning) the pre-trained resnet as feature extractor to find matching images.
* Research more into how to change the pixels from the input image in order to see what the network focuses on.
* Investigate how to obtain the feature extractor from a vgg, alexnet, densenet networks as well
* Decide either to use a cosine similarity metric or to train the last dense layer to predict the similarity since I came across this paper: <https://paperswithcode.com/paper/learning-an-adaptation-function-to-assess>

# Blockers

* Load the weights into the resnet given in the code submitted with the paper and try to duplicate their results. (I tried everything to read their pickle file but failed ... I think I will just train one anew)