

# Similar Image Retrieval

Similar image ranking can be implemented using an autoencoder. auto encoder consist of encoder and decoder. The goal is to compress our input data with the encoder, then decompress the encoded data with the decoder such that the output is a good reconstruction of our original input data. I have designed this model with some modification to existing works

1. Input data for training is read from current working directory(data/dataset/)
2. Input data for training is read from current working directory(data/test/)
3. Building auto encoder model- we create the architecture for our autoencoder network. This involves multiple layers of convolutional neural networks, max-pooling layers on the encoder network, and upscaling layers on the decoder network.
4. Resize image for training the data with available ram
5. Normalizing image- normalizing all train and test images(0,255)->(0,1)
6. Converting images to numpy array
7. Training our model
8. Reading input from user(k=no.of similar images)
9. Clustering of training data into k groups using KNN
10. Retrieving similar images and storing it in output folder

## Execution

1. Run image\_retrieval notebook with all the sub folders
2. set trainmodel=True for training,set trainmodel=False for loading pretrained model
3. set parallel=True for multiprocessing
4. Enter k for retrieving k number of inputs
5. I have executed the code in paperspace.com with 30 GB ram

## sample result



google drive link:

[https://drive.google.com/drive/folders/1MNW5XNTVbuRnsE-rLlkN\\_bKVNKVXOi3G?usp=sharing](https://drive.google.com/drive/folders/1MNW5XNTVbuRnsE-rLlkN_bKVNKVXOi3G?usp=sharing)

github:

[https://github.com/anandhu1436/similar\\_image\\_retrieval](https://github.com/anandhu1436/similar_image_retrieval)

(need to add full training dataset in git)