# Dataset for Image Inpainting:

1. **[Given by prof] NVlabs/ffhq-dataset** <https://github.com/NVlabs/ffhq-dataset>
2. **[Given by prof]** CityScapes dataset <https://www.cityscapes-dataset.com/>
3. <https://paperswithcode.com/task/image-inpainting>
4. **Cifar-10 dataset**: <https://www.cs.toronto.edu/~kriz/cifar.html>
5. Art Image : <https://www.kaggle.com/ikarus777/best-artworks-of-all-time>

Have downloaded

1. **[Thumbnails at 128×128]**

<https://drive.google.com/drive/folders/1tg-Ur7d4vk1T8Bn0pPpUSQPxlPGBlGfv>

1. **[vehicle\_trainvaltest.zip]**

<https://www.cityscapes-dataset.com/downloads/>

# Information/papers Links for Image Inpainting:

1. **A Gentle Introduction to Generative Adversarial Networks (GANs)** <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>
2. **Opportunities of scale: texture synthesis, multi-view reconstruction, im2gps, tiny images, etc.** Slide 95 - related to link #6 <https://courses.engr.illinois.edu/cs445/fa2020/lectures/Lecture%2021%20-%20Opportunities%20of%20Scale%20-%20Online.pdf>
3. **[USEFUL] Introduction to Generative models for Image Inpainting and Review: Context Encoders** <https://medium.com/analytics-vidhya/introduction-to-generative-models-for-image-inpainting-and-review-context-encoders-13e48df30244>
4. **[USEFUL] Review: High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis** <https://medium.com/analytics-vidhya/review-high-resolution-image-inpainting-using-multi-scale-neural-patch-synthesis-4bbda21aa5bc>
5. **[Given by prof] A Milestone in Deep Image Inpainting - Review: Globally and Locally Consistent Image Completion** [key words: **Dilated Convolution**] <https://towardsdatascience.com/a-milestone-in-deep-image-inpainting-review-globally-and-locally-consistent-image-completion-505413c300df>
6. **[Given by prof] Revision for Deep Image Inpainting and Review: Patch-Based Image Inpainting with Generative Adversarial Networks** <https://towardsdatascience.com/revision-for-deep-image-inpainting-and-review-patch-based-image-inpainting-with-generative-4197d29c5468>
7. **Encoder -Decoder** <https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726>

# Code for Image Inpainting

1. **Image Fine-grained Inpainting** <https://github.com/Zheng222/DMFN?fbclid=IwAR2vRKzx8Z3YcdbUn5lpAspNjvY3bfBjdkVY3yj2ECUXMQqxow5JXOqMCWQ>

# Paper + code for Image Inpainting

1. **[paper (Using GANs)]** https://arxiv.org/abs/2002.02609

**[code]**<https://github.com/Zheng222/DMFN>

**[code: someone practice again]**<https://github.com/HannH/DMFN>

1. [Huong’s Journal Club paper] **Texture Memory-Augmented Deep Patch-Based Image Inpainting**

28 Sep 2020 • open-mmlab/mmediting • Pytorch

Paper link: <https://arxiv.org/pdf/2009.13240v1.pdf>

Code: <https://github.com/open-mmlab/mmediting>

1. [Yunting’s Journal Club paper] **Pluralistic Image Completion**

Paper link: <https://openaccess.thecvf.com/content_CVPR_2019/papers/Zheng_Pluralistic_Image_Completion_CVPR_2019_paper.pdf>

Code: <https://github.com/lyndonzheng/Pluralistic-Inpainting>

Website: <http://www.chuanxiaz.com/publication/pluralistic/>

Poster:

<http://www.chuanxiaz.com/publication/pluralistic/Images/poster_picnet_cvpr19.pdf>

Running Successfully

1. **Pluralistic Image Completion**

Paper link: <https://openaccess.thecvf.com/content_CVPR_2019/papers/Zheng_Pluralistic_Image_Completion_CVPR_2019_paper.pdf>

Code: <https://github.com/lyndonzheng/Pluralistic-Inpainting>

Website: <http://www.chuanxiaz.com/publication/pluralistic/>

Colab: <https://colab.research.google.com/drive/1bHfSYVymDUmFVH3Zla7fVlGYJeeBDjX3?usp=sharing>

# Evaluation metrics

1. Structural Similarity Index (SSIM) - the higher the better
   1. <https://ourcodeworld.com/articles/read/991/how-to-calculate-the-structural-similarity-index-ssim-between-two-images-with-python>
   2. <https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e>
   3. <https://scikit-image.org/docs/0.12.x/api/skimage.measure.html#skimage.measure.compare_ssim>
   4. Changed in version 0.16: This function was renamed from skimage.measure.compare\_ssim to skimage.metrics.structural\_similarity
2. Total variation loss (TV loss) - the lower the better -
   1. tf.image.total\_variation | TensorFlow Core v2.4.1
   2. loss = tf.reduce\_sum(tf.image.total\_variation(images)) https://www.tensorflow.org/api\_docs/python/tf/image/total\_variation
   3. Total variation loss is the sum of the absolute differences for neighboring pixel-values in the input images. This measures how much noise is in the images.
3. Peak Signal-to-Noise Ratio (PSNR - the higher the better)
   1. <https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/>
   2. <https://dsp.stackexchange.com/questions/38065/peak-signal-to-noise-ratio-psnr-in-python-for-an-image>
   3. Concept: <https://www.ni.com/en-us/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html>

Other Links:

**Finetune AlexNet with Tensorflow** <https://github.com/kratzert/finetune_alexnet_with_tensorflow?fbclid=IwAR2_muqkKspQ5pYm1_qDyeyTVCpyOvlKHZywBGYit6eqfNLtV56TJuzi_TA>

**L1 and L2 Loss Functions**

<https://afteracademy.com/blog/what-are-l1-and-l2-loss-functions>