Deep Learning for Image Inpainting

CS337 PROJECT PRESENTATION

Images taken from Places2 dataset, for depiction

Agenda

- Problem at Hand
- Datasets
- Baselines and Models
- Results
- Limitations

Images taken from IMAGENET dataset, for depiction

Image Inpainting

Task: Fill the missing regions in the image by understanding the content of the entire image and producing a plausible hypothesis for the missing part.

Related Work

- Traditional methods solve the problem using ideas of texture and patch synthesis by matching and copying background patches into holes. These approaches work well especially in background inpainting tasks.
- These approaches assume that missing patches can be found somewhere within valid regions, and hence lack the capability to hallucinate or generate entirely new image content (e.g. , face)
- Recent successes of Deep Learning and Generative Adversarial Learning for Image related tasks inspires us.



Datasets Used

Images taken from ANIMALS dataset, for depiction

ANIMALS – 5400 images of 90 different animals, 60 images in each class. 4950/450 Train Test Split.

IMAGENET – Subset of 50K images chosen for training.

Baseline Methods and Models

NAVIER STOKES

- The method is directly based on the Navier-Stokes equations for fluid dynamics.
- Idea is to propagate isophote lines continuously from the exterior into the region to be inpainted.
- Available in OpenCV [1].

FAST MARCHING

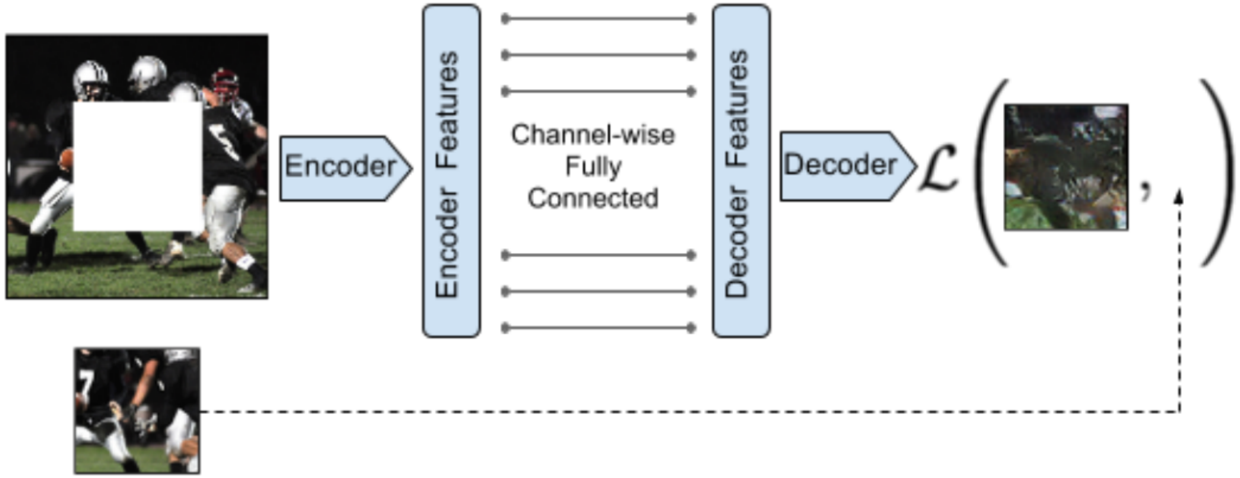
- Algorithm is based on Level Set method.
- Iteratively propagates information from known parts of image to unknown parts minimizing the difference.
- Available in OpenCV [2].

AUTOENCODER

- The very first thought for an Image-to-Image task. We use standard MSE Reconstruction Loss.

Context Encoder

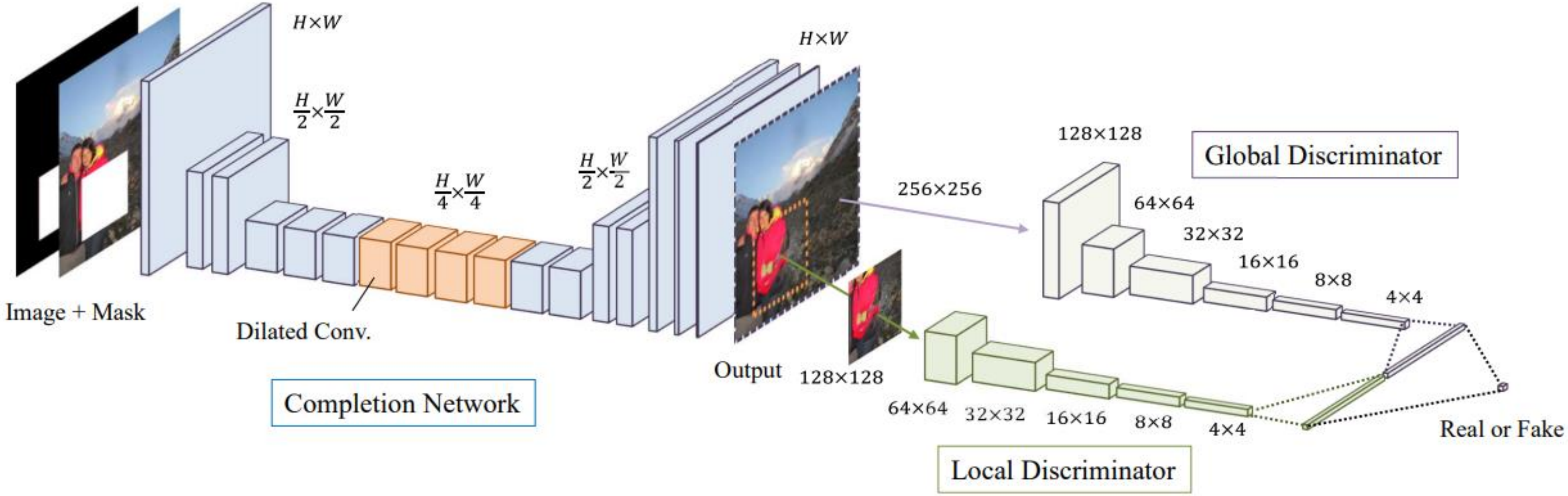
- Encoder here takes a masked image and produces a latent feature representation, which the decoder takes as input and produces the missing image content.
- Uses a channel wise fully connected layer to connect encoder with decoder.
- Employed Joint Loss of Reconstruction(MSE) and Adversarial Loss(On Missing Patch only)



Reference: [3] : Context Encoders: Feature Learning by Inpainting, Pathak et al., 2016

Globally and Locally Consistent Inpainting

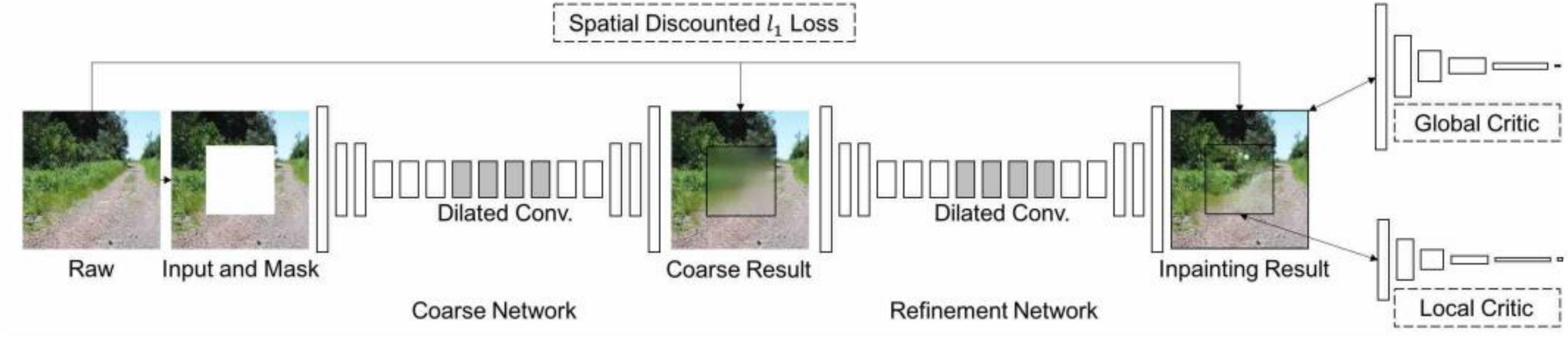
- Used Dilated Convolution Networks and Multi Scale Discriminators for Global and Local Consistency.



Reference: [4] : Globally and locally consistent image completion, Iizuka et al., 2017

Variant of GLCIC

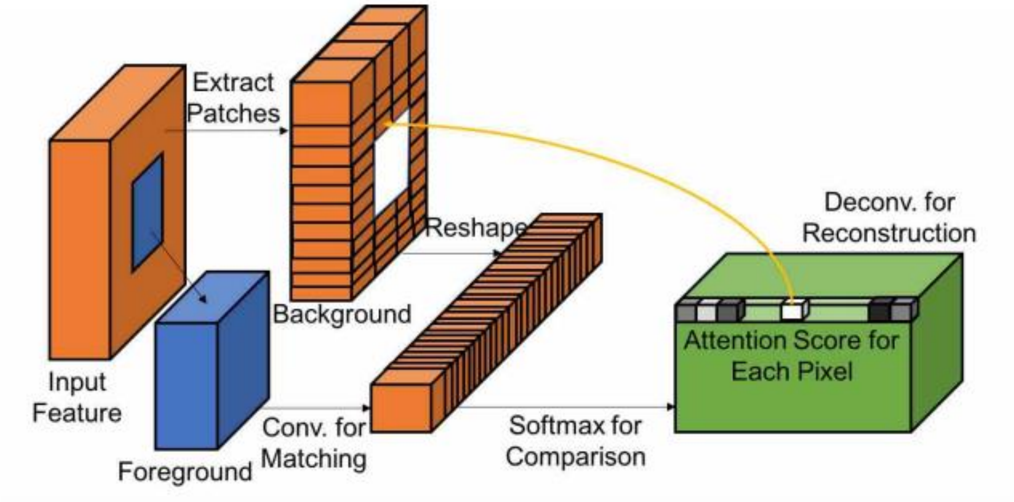
- Has two networks - Coarse Network and Refinement Network (inspired from deep supervision or residual learning), outperformed the GLCIC in terms of both time and image quality. Also here, Multi-Scale Critics(WGAN) are treated independently. Adds spatially discounted reconstruction loss to improve stability.



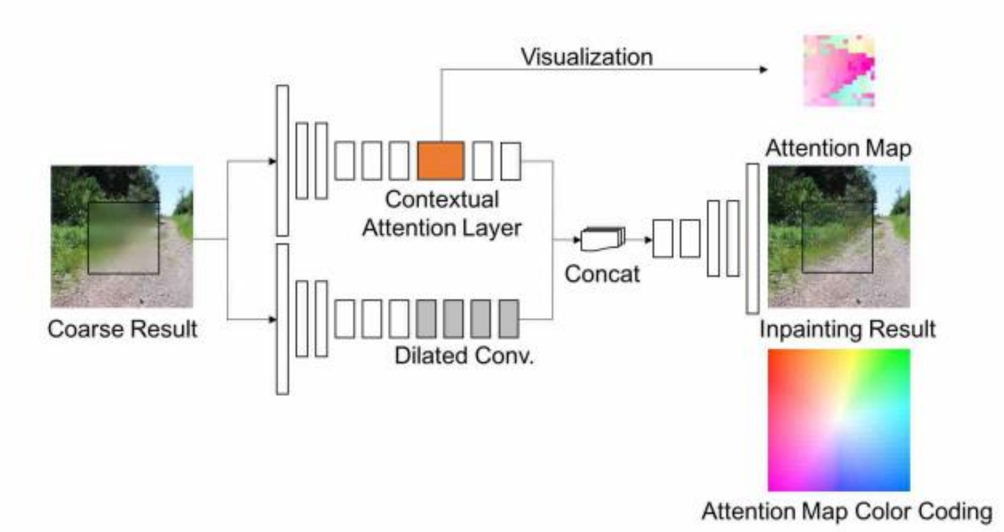
Reference: [5] Generative Image Inpainting with Contextual Attention, Jiahui et al., 2018

Contextual Attention

- Proposes a novel contextual attention layer to explicitly attend on related feature patches at distant locations.



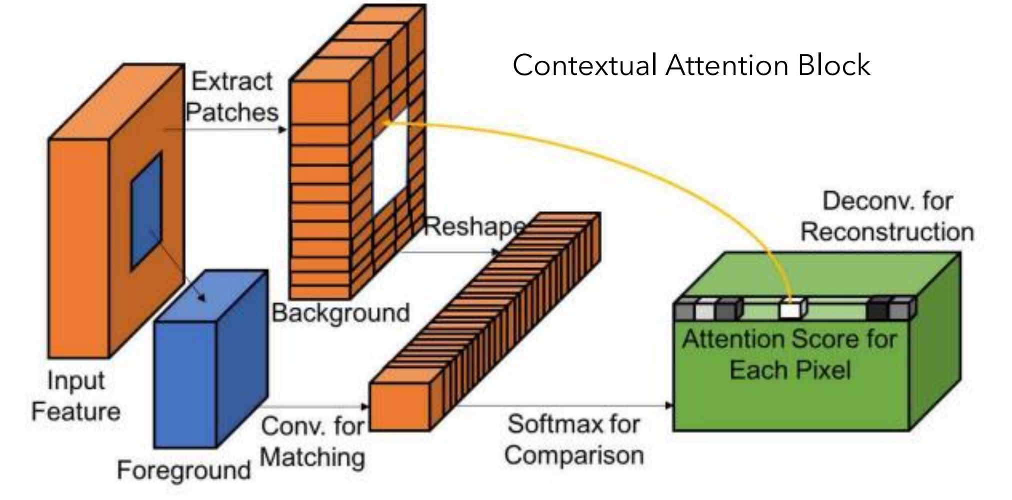
Contextual Attention Block



CA Block integrated into Refinement Layer as a parallel network

Reference: [5] Generative Image Inpainting with Contextual Attention, Jiahui et al., 2018

Contextual Attention



- Match and Attend $s_{x,y,x',y'} = \langle \frac{f_{x,y}}{||f_{x,y}||}, \frac{b_{x',y'}}{||b_{x',y'}||} \rangle$

Attention score for each pixel DeConv reconstructs foregrounds using (x',y'). Attention score can be calculated using convolution and channel-wise softmax. $s^*_{x,y,x',y'} = \mathit{softmax}_{x',y'}(\lambda s_{x,y,x',y'})$

- Attention Propagation

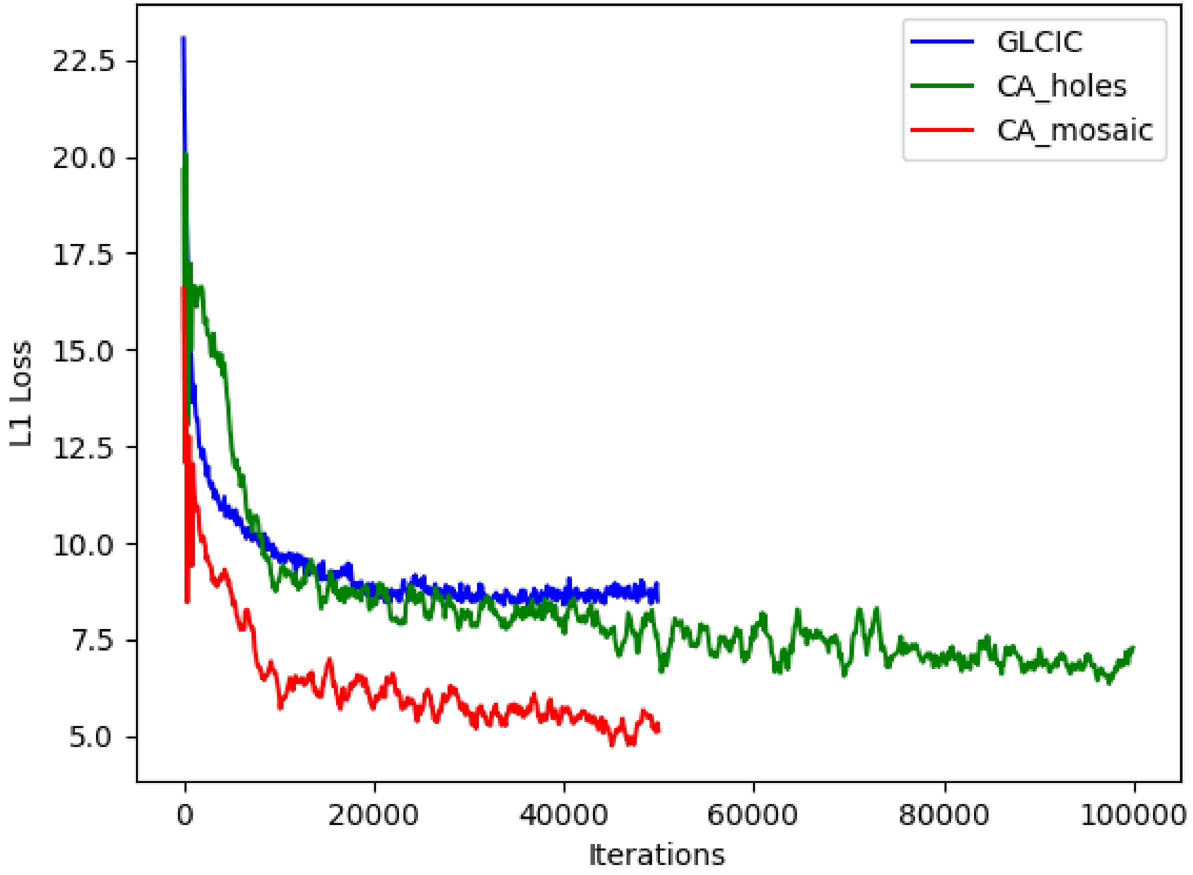
  This encourages coherency of attention. The idea of coherency is that a shift in foreground patch is likely corresponding to an equal shift in background patch for attention. We perform a left-right AP followed by top-down. Can be implemented using convolution matrix with identity matrix as kernels.

$$\hat{s}_{x,y,x',y'} = \sum_{i \in \{-k,\ldots,k\}} s^*_{x+i,y,x'+i,y'}.$$

Reference: [5] Generative Image Inpainting with Contextual Attention, Jiahui et al., 2018

Results and Performance

Training Progression

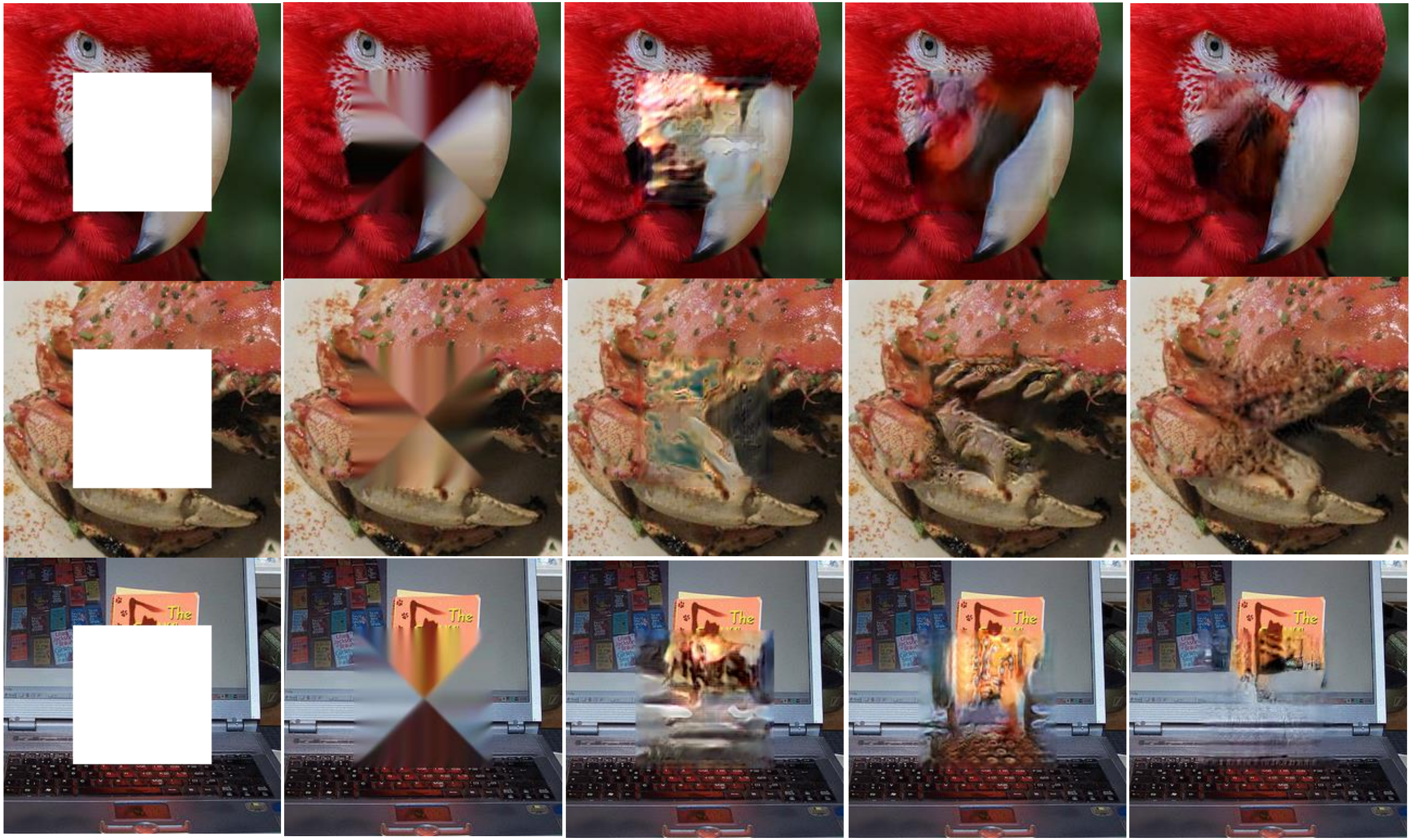


- GLCIC Variant trained on ANIMALS for 50K iterations.
- CA – Trained with Single squared hole on ANIMALS for 100K iterations.
- CA – Trained with Single squared mosaic blur on ANIMALS for 50K iterations.

Performance Comparison

MODE LS	L1	MSE	PSNR	LPIPS	TV
NS	9.68	810.47	19.61	0.164	8.14
TELEA	9.65	821.72	19.63	0.168	7.96
AUTOENCODER(100K)	9.34	1472	**26.25**	0.18	2.18
GLCIC(50K)	9.25	720.49	20.16	0.13	1.70
CA – ANIMALS - HOLE(100K)	8.45	659.44	20.65	0.11	**0.79**
CA – IMAGENET – HOLE(300K)	**7.20**	**505.84**	21.90	**0.10**	1.43
CA – ANIMALS-MOSAIC(50K)	5.31	270.44	24.68	0.06	1.25

***Losses are calculated on test dataset of ANIMALS**



Input Image | Navier Stokes | GLCIC(ANIMALS) | CA (ANIMALS) | CA (IMAGENET)

Random Images chosen from ImageNet test dataset

Input Image | Navier Stokes | GLCIC(ANIMALS) | CA (ANIMALS) | CA (IMAGENET)

Random Images chosen from ImageNet test dataset

Original Image Masked Image CA(IMAGENET)

Random Images chosen from ImageNet Test Set

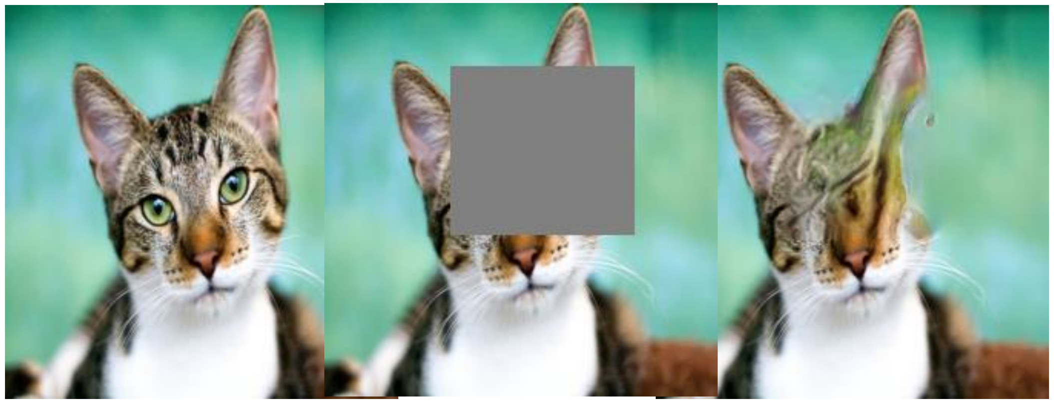
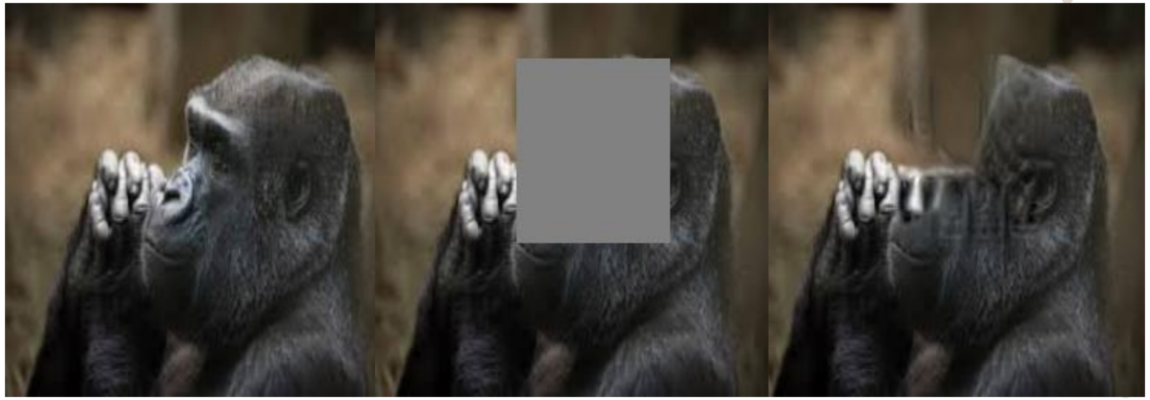
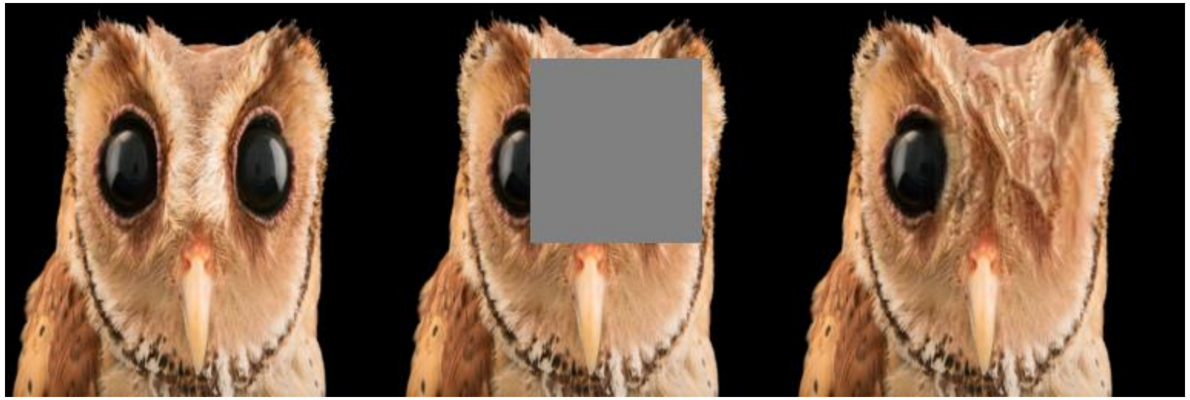
Masked Image CA(IMAGENET)

Masked Image CA(IMAGENET)

Random Images chosen from Places2 Dataset

Limitations

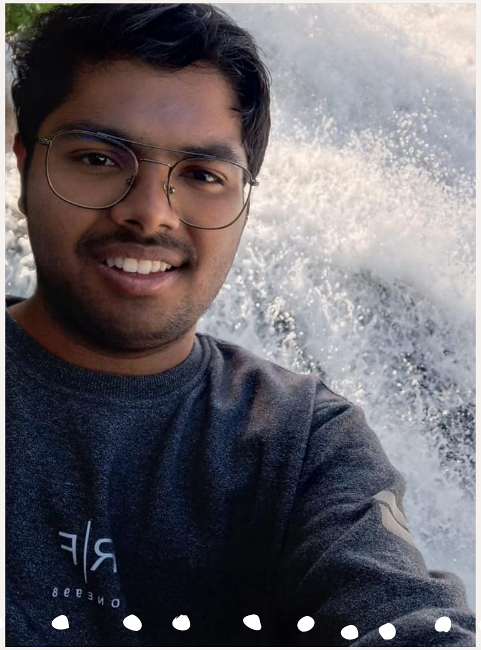
- Images with highly structured objects like faces do not satisfactorily get inpainted.

Our Team

Guramrit

210050061



Sabyasachi

210050138

Nikhil

210050035

Omm

2100050110

Work and Contribution

1. CodeWork : (Whole Team)
    1. **Model Components – Omm (Gen, Local and Global Dis), Sabyasachi (CA)**
    2. **Training Pipeline – Nikhil, Guramrit**
    3. **Utilities – Sabyasachi, Nikhil**
    4. **Dataset and Log Outputs – Guramrit, Omm**
2. Experiments and Evaluation : (Nikhil, Guramrit)
3. Documentation : (Sabyasachi)

References

[1] M. Bertalmio, A. L. Bertozzi and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990497.

[2] Telea, Alexandru Cristian. "An Image Inpainting Technique Based on the Fast Marching Method." Journal of Graphics Tools 9 (2004): 23 - 34.

[3] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2536-2544, doi: 10.1109/CVPR.2016.278.

[4] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. ACM Trans. Graph. 36, 4, Article 107 (August 2017), 14 pages. https://doi.org/10.1145/3072959.3073659.

[5] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu and T. S. Huang, "Generative Image Inpainting with Contextual Attention," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 5505-5514, doi: 10.1109/CVPR.2018.00577.

[6] https://github.com/JiahuiYu/generative_inpainting/tree/v1.0.0 , Tensorflow code for Generative Inpainting using Contextual Attention

THANKYOU