Semantic Image Inpainting Using Self-Learning Encoder-Decoder and Adversarial Loss

Nermin M. Salem
Department of Electrical Engineering
Future University in Egypt
Cairo, Egypt
nfawzy@fue.edu.eg

Hani M. K. Mahdi
Department of Computer Engineering
Ain Shams University
Cairo, Egypt
hani.mahdi@eng.asu.edu.eg

Hazem M. Abbas
Department of Computer Engineering
Ain Shams University
Cairo, Egypt
hazem.abbas@eng.asu.edu.eg

***Abstract*—Images are exposed to deterioration over years due to many factors. These factors may include, but not limited to, environmental factors, chemical processing, improper storage, etc. Image inpainting has gained significant attention from researchers to recover the deteriorated parts in images. In this paper, two new techniques for image inpainting techniques using Deep Convolution Neural Networks (CNN) are proposed. In the first technique, a self-tuned Encoder-Decoder architecture based on a Fully Convolution Network (FCN) is used to generate different sized blocks from non-deteriorated image dataset with L2 being used as a loss measure. On the other hand, the second technique is a two-step technique inspired from Context Encoders. In the first step, Context Encoders are trained on non-deteriorated image dataset to select blocks from training images with minimum L2 loss. In the second step, the selected block is applied to Generative Adversarial Networks (GAN) in order to improve the quality of the recovered image. Several simulation examples were made to proof that the performance self-tuned Encoder-Decoder and GAN is the same. Simulations have also shown that the proposed methods have superior performance in recovering missing regions in deteriorated images over other state-of-art techniques. Paris Street View dataset was used for training and validation to validate our results.**

***Keywords*—Image Inpainting, L2 loss, GAN, Context Encoder.**

I. Introduction

Image inpainting is one of the challenging topics in computer vision fields. It aims to recover the missing or deteriorated regions in an unnoticeable manner to human eye. Many techniques have been proposed to overcome this problem using structural or textural or hybrid methods [1]. In [2], the proposed technique trained an Encoder-Decoder structure using CNN with L2 reconstruction loss and adversarial loss (adv loss). They inpainted a fixed center missing region of an image. The results were good, but they failed to produce the high-level texture details due to the presence of aircrafts around the mask edges. Also, they failed to handle high-resolution images in terms of adv loss. In [3], the proposed technique was proposed for high-resolution images using pretrained model obtained from [2] trained with L2 loss only. The results are then passed to VGG19 [4] network to obtain the high-level texture details. The results were satisfied only for center-symmetric images but failed for asymmetric images. In [5], the proposed technique is a restoration model for inpainting, deblurring, denoising and pixel interpolation. The inpainting model is a self-generation feedback mechanism based on L2 loss. Although the results were good for small regions, they were not satisfactory when the missing region is large. In [6], the proposed technique used a classification pretrained-model to obtain the image that has maximum score then construct a graph-based regularizer for producing smooth results. The used technique, however, failed to produce the texture details of the images. In [7], the proposed technique used an image generative network that takes high-level features into consideration while inpainting missing region, the drawback was that it requires long training time, as well as, powerful GPUs.

All the above mentioned techniques have failed to give a natural visual to the image when the missing region is relatively large. Hence, a technique that handles inpainting relative large missing regions regardless to its size or location was needed.

In this paper, two techniques are proposed for semantic region filling (i.e. large region is missing from the original image). The first technique is based on active learning approach that uses an automated self-tuned Encoder-Decoder architecture trained with L2 loss. The proposed technique arranges training samples in an unsophisticated training order according to the size of the region to be inpainted (up to 50% of the image size). Since the area to be inpainted is large, a more complex task to perform is required. The common problem of using L2 loss only is that, the results are blurry restored images as in Fig.1. The proposed system overcomes blurry results by dividing the training dataset into pre-defined categories (i.e. equal to the size of missing region to be inpainted) with initially the same number of samples-per-category. In the end of each training epoch, the system self-evaluates its performance by calculating the L2 loss over the validation set. If the obtained L2 loss of the category is large, the system will need to produce more samples for this category in the next training epoch and vice versa. As a result, a better feature representation for the nearest neighbors patches computations is obtained giving more natural visual images.

Fig. 1: Blurry results of Context Encoders with L2 loss.

The second technique is a two-step technique. In the 1st step, the same Encoder-Decoder architecture of the first method is used. However, it is trained on a fixed square cropped region at a fixed location in the image (i.e. top-left, bottom-right). This produces an initial recovered block for the missing region. In the 2nd step, the recovered block is used as an input for GAN [8] to produce the final recovered image. Using a GAN network helps in fully understanding the high-level texture details which corresponds to natural restored images. This technique can inpaint up to 37% of the whole image size but with a fixed location and deep recognition of image context, as well as, its high-level texture features. To inpaint a missing region in any other location, the network must be retrained.

The two proposed techniques aim to inpaint any arbitrary square missing region regardless to its location and size within an image. In addition, they maintain reasonable predictions that results in a smooth natural looking images without any overhead of blending or post-processing steps. We used Paris Street View dataset [9] for training models and evaluation. Experiments showed that, our two approaches achieved superior improved consistent images in terms of texture details, as well as, the filling size and location of the semantic cropped region.

We made the following contributions:

1) We proposed a self-tuned Encoder-Decoder network technique that uses fully convolutions along with automatically updated random square mark to achieve state-of-art image inpainting.
2) We proposed another technique, inspired by context encoders, and succeeded in inpainting any square mask at any position of the image with very natural look results. However, the original Context Encoder has failed to inpaint any arbitrary square mask other than the center of the image.
3) We concluded that a self-tuned Encoder-Decoder can be used for producing very smooth consistent images without the need to use an additional GAN network.

II. Related Work

Image inpainting techniques have been tackled a lot in literature. However, we will not be able to discuss all these methods. Therefore, we will only discuss the approaches that are based on unsupervised, active and semantic learning. Furthermore, we will review the work that has been done in these three fields.

A. Unsupervised Learning

One of the pioneers of unsupervised deep learning methods were the methods that employed the use of autoencoders [10, 11]. Unfortunately, Autoencoders inpaint an image by mapping it to a low dimension bottleneck layer, obtaining a compact feature representation but without understanding the semantics of the image which results in poor-quality images. Another approach is denoising autoencoders [12], where it deploys unsupervised learning of a representation to reconstruct the image to reconstruct a partially corrupted image using the learned representation. Context Encoders [2] used autoencoders concept but upgraded it to be able to handle highly corrupted input images they trained sensitive model to obtain good restored results but only for center-missing region images based on L2 and adversarial loss. An alternative of using adversarial loss, the authors in [3], used a texture network to jointly infer the missing center region.

B. Active Learning

An Active Learning algorithm implies the ability of studying and modifying the system without any external help, relying only on a little amount of information. These algorithms have a huge contribution for computer vision researches [5], [13, 14, 15, 16, 17].

C. Semantic Learning

Semantic image inpainting [7, 17] is considered as a difficult task as it is concerned with restoring big missing regions depending on the visual neighborhood data. the new methods of semantic image inpainting involves the use a fully convolution approach. In [18], a back-propagation approach is used to map the corrupted image to a smaller featured vector. The mapped vector is then passed through the GAN [8] network to predict the missing region.

Our two techniques overcome the drawbacks of the existing inpainting techniques mentioned above in this section; in terms of realistic complete scene images and decreased computation time as neither images have to be optimized nor a different set of hyper parameters for each single image is required. Also, our techniques can inpaint any random mask in shape and size while maintaining consistent images. Our trained models are flexible enough to detect mismatches in a wide range of higher order statistics between the model predictions and the ground-truth, without the need to pre-define them.

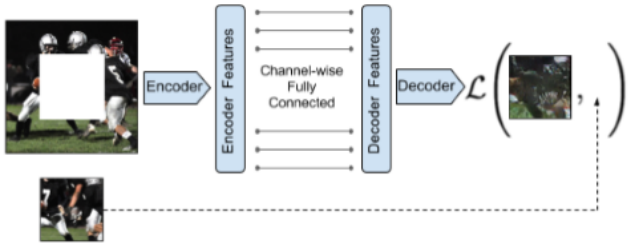

Fig. 2: Context Encoder Architecture.

III. Network Architecture

The image inpainting network as depicted in Fig.2, is a simple Encoder-Decoder. The Encoder is based on the AlexNet architecture [19]. The encoder composed of six fully convolution layers with kernel sizes of 4 and stride size of 2. The decoder is also composed of six up-sampling layers. The last layer of the decoder is responsible for combining the predicted block with the rest of the image. We used a symmetric encoder-decoder pipeline that is for efficient and effective learning and training.

The Encoder-Decoder modules are connected through a fully channel-wise layer. The input for the encoder is a 128x128 distorted image R_d with cropped region distributed randomly across the image. It is found that scaling R_d from 227x227 to 128x128 results in a better quality recovered images, as mentioned in [2]. The encoder represents the image context by a learned feature vector. Next, the decoder applies this learned feature vector to produce the recovered image $\hat{R}$. To validate the quality of the recovered image $\hat{R}$, a normalized L2 reconstruction loss is used as:

$$L2_{rec} = \left|\left|R_i - \hat{R}\right|\right|_2^2 \tag{1}$$

where R_d is the real original image, while $\hat{R}$ is the recovered image.

IV. Proposed Techniques

The Encoder-Decoder described in sec. III suffers from poor image quality of the recovered image especially when the cropped region is large. To overcome this drawback, two new techniques are proposed namely: 1) Self-Tuned Encoder-Decoder system; 2) L2-GAN system. These techniques can inpaint any large cropped regions regardless its size or location. These techniques are described as follows:

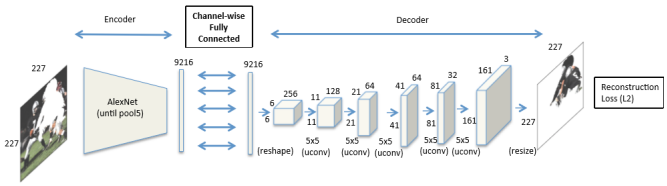

Fig. 3: Self-Tuned Encoder-Decoder System.

N sub-instances with increasing difficulty: $T_1, T_2 \ldots, T_N$.

Batch Size: B

Initialization: $B_i = B/N$

While not converge do

Continue training for one epoch and snapshot;

If end of epoch, then

i = 1

for $i \leq N$ do

validate snapshot model on sub-instances T_i;

get mean squared L2;

end.

update $B_i = \frac{1/p_i}{\sum_{i=1}^{N} 1/p_i} B$;

end.

end.

Fig. 4: Self-Tuned Encoder-Decoder pseudocode.

A. Self-Tuned Encoder-Decoder System

This technique starts with a predefined complexity levels N yields to different block sizes according to the percentage of cropped region in image R_d.

This complexity levels, denoted by N, correspond to the percentage of the cropped region in images and it ranges from 1% to 50% of the total image size. The training begins by dividing the dataset into categories according to the size of the cropped region to be inpainted. Initially all categories have the same number of images. As the training goes on, L2 is measured at end of each epoch and the number of images-per-category is increased or decreased according to this loss. In case the L2 loss measured for a certain category was big, the number of training samples-per-category is increased and vice versa. There is a threshold for selecting the number of complexity levels N. The higher the N, the more need to generate more categories. This leads to excessive computation time, and hence, complexity levels N optimizes the image accuracy and training computation time.

The architecture of the used system is shown in Fig.3 and the pseudocode is described in Fig.4 to express the steps mentioned above.

After setting up the complexity levels N, the image is applied to the Encoder-Decoder system mentioned in Section 3 for training and validation. To improve the image quality, we minimize L2 measure in Eq. (1) by maximizing the conditional probability between the real image , denoted by R_i, and

distorted image, denoted by R_d, by applying Bayes theorem.

$$P\left(R_d, R_i\right) = P\left(R_d\right) P\left(R_i | R_d\right) \tag{2}$$

However, this is not feasible because $P\left(R_i\right)$ is unknown. Therefore, we use a point estimate to yield new $\hat{R} = f\left(R_d, w\right)$ through the Encoder-Decoder system by minimizing the following mean squared error objective:

$$E_{R_i, R_d} = \left|\left|R_i - \hat{R}\right|\right|_2^2 \tag{3}$$

The Encoder-Decoder system is trained to learn its weights w by minimizing the following Monte-Carlo estimate of the mean squared error objective:

$$\hat{w} = argmin_w \sum_i \left|\left|R_i - \hat{R}\right|\right|_2^2 \tag{4}$$

The network takes distorted image R_d as an input and pass it within the network to output $\hat{R}$ as the restored image.

B. L2-GAN System

This technique consists of two steps:

1) the recovered image $\hat{R}_{initial}$ is obtained through the basic normalized L2 reconstruction loss in Eq. (1).
2) $\hat{R}_{initial}$ is applied to GAN system [7] with adversial loss (adv loss).

Adv loss produces recovered realistic textures as well as the fine-grained details from images that have been down sampled. As a result, it is more like an encouragement for the output images to be similar to the original images.

GAN's idea is to train two models simultaneously, a generative model G and a discriminative model D. G is responsible for capturing the data distribution while D evaluates the probability of sample to determine whether it is from training set or from G's prediction. It is like a two-game player where 4 takes G's prediction and real images as inputs and tries to distinguish between them. G tries to confuse D by providing samples that looks similar to the real images. D's function to decide whether it is real or not.

G's distribution p_g is learned from data x. $D(x)$ is the probability that an image x belongs to training data and not from p_g. D is trained to maximize the probability of providing the right match to both training examples and samples from G which is trained to minimize $\log(1 - D(G(x)))$.

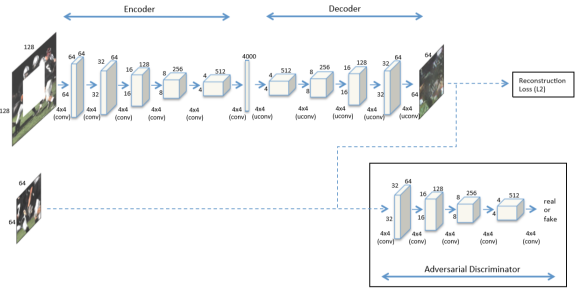

Fig. 5: L2-GAN System.

Therefore, GAN learns an adversarial D model to provide loss gradients to the G model as shown in Eq. (5).

$$L_{adv} = max_d \, E_{x \in x}[\log(D(x)) + \log(1 - D(x))] \tag{5}$$

For our network architecture, the decoder is considered as generator as shown in Fig. 5. Therefore, the overall loss function can be given from the following equation:

$$Overall\ Loss = \mu_{rec} L_{rec} + \mu_{adv} L_{adv} \tag{6}$$

Our technique can inpaint any large missing square region at one of location of the image (top right, top left, bottom left and bottom right). These four locations correspond to each quarter of the image. Results are shown in Fig.6 (b and c).

V. Experiments

Our inpainting network was carried out in 'Caffe' and 'Torch'. The optimization algorithm ADAM [20] is applied for optimization as a stochastic gradient descent solver. Default solver parameters are used as suggested in [1, 21] and batch size $B = 100$ and the bottleneck is of 4000 units during the training. All pooling layers of the Encoder are being replaced with fully convolutions layers of the same kernel size and stride for to yield better end-to-end learning through all experiments.

A. Dataset

The Paris Street View dataset [9] is used for comparison and simulation. It has a training set composed of 14900 images and a validation set composed of 100 images.

B. Comparison with existing image inpainter

The proposed techniques were compared with techniques proposed in [2], [3] and [5]. The technique proposed in [2] showed good results however it seemed blurry and can only inpaint center cropped region.

In [3], they used a combination between Context Encoders and texture network for center inpainting of images, but results were not consistent and did not look natural to human eye. In [5], they trained using random cropped regions, but also our proposed techniques have better performance. Results of [2], [3] and [5] are shown in Fig. 6 (6a and 6b).

For our first technique (Self-Tuned Encoder-Decoder System), we used five predefined complexity levels $N = 5$ during training. the choice of 5 levels was to demonstrate the idea of the technique with reasonable training time.

We used cropped square blocks of size 11 to 3030 at various locations scattered all over the image during training. We used five complexity levels as mentioned in section IV, 1x1 to 6x6, 7x7 to 12x12, 13x13 to 18x18, 19x19 to 24x24 and 25x25 to 30x30.

We inpaint random blocks of different sizes and locations using only one model, restoring distorted images with varying percentage of unknown pixels, however we experimented with images of size 128x128 for efficient results.

Our technique still shows a competitive behavior even for even the larger regions without the use of adversarial loss as during training.

For our second technique (L2-GAN System), we trained with images of size 350x350 with a cropped region of size 128x128. However, we must train the network on this location from the beginning as we could not train GAN on the whole image.

Results of our proposed techniques are shown in Fig. 7 (7a, 7b and 7c).

Our proposed techniques show a very good performance for semantic image inpainting in terms of the inpainting quality of arbitrary random square missing blocks (size and location) in an efficient visual manner. However, we did not consider haphazard shapes for this random mask and noise is not considered while training.

We trained using GeForce 940M GPU. Table I shows that our first technique yields the lowest L2 loss compared to the other techniques. The second technique, however, results achieved the highest L2 loss while looking superior, in the meanwhile, due to the presence of GAN.

Fig. 6a: Inpainting results of center masked-images of Context Encoders [2] and High resolution [3]

Fig. 6b: Inpainting results of On-Demand Learning for Deep Image Restoration [5]

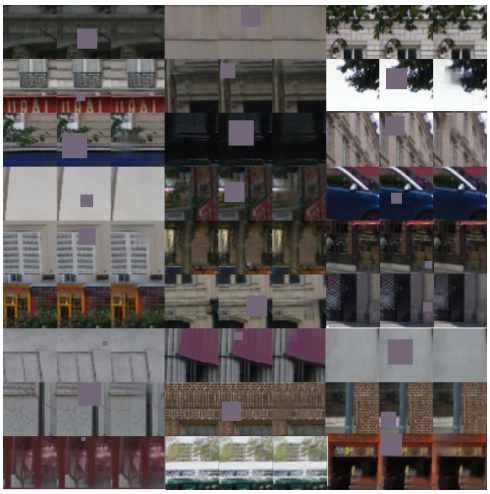

Fig. 7a: Self-Tuned Encoder-Decoder result.

TABLE I: Performance summary of all algorithms.

Method	Mean L2 Loss	D_Loss
Context Encoder (center region) [2]	1.96%	...
On-Demand learning [5]	2%	...
High-resolution (center region) [3]	2.21%	...
Self-Tuned Encoder/Decoder System	**0.79%**	...
L2-GAN System	**3.19%**	**1.96%**

Fig. 7b: L2-GAN System results for bottom_left cropped region.

VI. Conclusion

Image inpainting is indeed considered one of the most important emerging research areas in computer vision. We tackled it by using two different techniques that are based on a symmetric fully convolutional Encoder-Decoder network connected through a full channel. To prove the effectiveness of our techniques, our models were trained using Paris Street View dataset for inpainting images with random square masks varying in location and size. Experiments showed that our self-tuning Encoder-Decoder algorithm, based on L2 reconstruction loss only, was able to maintain superior, natural and consistent results similar to those obtained from GAN with less training time.

References

[1] Raluca Vreja and Remus Brad. "Image Inpainting Methods Evaluation and Improvement". In: *The Scientific World Journal* vol. 2014 (2014).

Fig. 7c: L2-GAN System results for top_right cropped region.

[2] D. Pathak et al. *Context encoders: Feature learning by inpainting*. In CVPR.

[3] Chao Yang et al. *High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis*. arXiv:1611.09969. 2016.

[4] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. In ICLR, 2014.

[5] Ruohan Gao and Kristen Grauman. *On-Demand Learning for Deep Image Restoration*. In ICCV, 2017.

[6] Alhussein Fawzi and Horst Samulowitz. *Deepak Turaga, Pascal Frossard. Image Inpainting Through Neural Network Hallucinations*. 12th IEEE Image, Video, and Multidimensional Signal Processing Workshop, 2016.

[7] Pauline Luc et al. *Semantic Segmentation using Adversarial Networks*. NIPS Workshop on Adversarial Training, 2016.

[8] I. Goodfellow et al. *Generative adversarial nets*. In NIPS, 2014.

[9] C.Doersch et al. *What makes paris look like paris?* ACM Transactions on Graphics, 2012.

[10] Y. Bengio. *Learning deep architectures for ai. Foundations and trends in Machine Learning*. 2009.

[11] G. E. Hinton and R. R. Salakhutdinov. *Reducing the dimensionality of data with neural networks*. Science, 2006.

[12] P. Vincent et al. *Extracting and composing robust features with denoising autoencoders*. In ICML, 2008.

[13] K. He et al. *Deep residual learning for image recognition*. In CVPR, 2016.

[14] A. Radford, L. Metz, and S. Chintala. *Unsupervised representation learning with deep convolutional generative adversarial, networks*. In ICLR, 2016.

[15] G. Larsson, M. Maire, and G. Shakhnarovich. *Learning representations for automatic colorization*. In ECCV, 2016.

[16] J. L. Elman. *Learning and development in neural networks: The importance of starting small Cognition*. 1993.

[17] V. Jain and S. Seung. *Natural image denoising with convolutional networks*. In NIPS, 2009.

[18] Raymond A. Yeh et al. *Semantic Image Inpainting with Perceptual and Contextual Losses*. In CoRR, 2016.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet classification with deep convolutional neural networks*. In NIPSR, 2012.

[20] Y. Jia et al. *Caffe: Convolutional architecture for fast feature embedding*. In ACM Multimedia, 2014.

[21] A. L. Maas, A. Y. Hannun, and A. Y. Ng. *Rectifier nonlinearities improve neural network acoustic models*. In ICML, 2013.