# Hyperrealistic Image Inpainting with Hypergraphs

Gourav Wadhwa[1] Abhinav Dhall[2,1] Subrahmanyam Murala[1] Usman Tariq[3]
Indian Institute of Technology, Ropar[1] Monash University[2] American University of Sharjah[3]
{2017eeb1206, murala}@iitrpr.ac.in abhinav.dhall@monash.edu utariq@aus.edu

**Figure 1:** Image Inpainting results by our method based on hypergraph convolution on spatial features. Each pair shows the input image and predicted image by our method. White pixels represent the missing data that needs to completed. [Best viewed in color]

## Abstract

*Image inpainting is a non-trivial task in computer vision due to multiple possibilities for filling the missing data, which may be dependent on the global information of the image. Most of the existing approaches use the attention mechanism to learn the global context of the image. This attention mechanism produces semantically plausible but blurry results because of incapability to capture the global context. In this paper, we introduce hypergraph convolution on spatial features to learn the complex relationship among the data. We introduce a trainable mechanism to connect nodes using hyperedges for hypergraph convolution. To the best of our knowledge, hypergraph convolution have never been used on spatial features for any image-to-image tasks in computer vision. Further, we introduce gated convolution in the discriminator to enforce local consistency in the predicted image. The experiments on Places2, CelebA-HQ, Paris Street View, and Facades datasets, show that our approach achieves state-of-the-art results.*

## 1. Introduction

Image inpainting is the task of filling the missing regions such that modifications in the image are semantically plausible and can be further used in real-world applications such as restoring damaged or corrupted parts, removing distracting features from images, and completing occluded regions. There have been many learning and non-learning methods proposed in the past few decades. However, due to its inherent equivocalness and complexity in the natural images, image inpainting remains a challenging task.

To create a semantically plausible and realistic image, generally there are two requirements, $(a)$ global semantic structure, and $(b)$ fine detailed texture around the holes. Capturing of global semantic structure is non-trivial as a trained model can be easily biased towards producing blurred content. Current image inpainting methods can be broadly divided into two categories: 1. content or texture copying approaches [8, 8, 11], and 2. generative networks based approaches [34, 47, 57].

The first method, content or texture copying, borrows the content or textures from the non-hole pixels to fill the miss-

ing regions. An example is total variations (TV) [46, 40] based approaches, which exploit the smoothness property in the image to fill in the missing regions. The patch matching approach borrows content from the surroundings to fill the missing regions. Patch Match algorithms [8, 3, 11, 12, 17] iteratively fill the missing pixels by searching the similar patches from the non-holes pixels in the image. These methods can effectively fill even the high-frequency missing content, however, are unable to identify the global semantic structure of the image producing improbable results.

Generative networks are being used in many computer vision tasks such as Image super-resolution [4, 37], image de-blurring [60, 41], image colorization [6, 56] etc. The generative network based approaches [39, 20, 42, 51, 53, 54, 34, 52] use these generative networks to predict the missing region in an image. These approaches learn to model distribution for the missing region conditioned on the image's available surrounding regions. In [20], the idea of using global and local discriminators to improve the local consistency of the completed image was proposed. These methods worked well when there are similar images in the training and test sets. However, these methods may not able to produce satisfying results for a totally different test image. Moreover, these methods produced artifacts for large irregular holes. In [42], a patch swap mechanism between the Image2Feature network and Feature2Image network was used. This helped in combining the copying and deep learning approaches to map the uncompleted image with the completed images. [51, 53, 54] used novel contextual attention mechanisms to borrow the patches from a distant location.

Inspired by the *hyperrealism art* genre of painting, which resembles the high-resolution images, we propose a novel image inpainting method using the hypergraphs structure. The proposed hypergraph structure enables the network to find matching features from the background to fill in the missing regions. We use a two-stage network (coarse and refine network) for image inpainting. Firstly, the coarse network roughly fills the missing region and then the refine network uses this coarse output to produce finer results. We introduce a novel data-dependent method for developing incidence matrix for hypergraph convolution. To the best of our knowledge, this is one of the first works to propose use of hypergraph convolution networks on spatial features for any image to image tasks in computer vision. We also show that our proposed method obtains substantially better results for both center and irregular mask for image inpainting. The proposed hypergraphs convolutional layer can easily be used for the other computer vision tasks such as image super-resolution, image de-blurring, to get a global context in the image. Further we introduce gated convolution in discriminator to enforce the local consistency in the predicted image. Our major contributions can be summarized as,

- We propose a novel Image inpainting network using hypergraphs to produce globally semantic completed images.
- We propose a trainable method to compute data-dependent incidence matrix for hypergraph convolutions.
- We introduce gated convolution instead of regular convolutions in the discriminator, enabling it to enforce local consistency in the completed image.

Further, we train our network using a simple yet effective incremental strategy, which enables completion of the irregular holes. We also test our network on four publicly available datasets and show that our method performs significantly better than the previous state-of-the-art-methods.

## 2. Related Work

**Free Form Image Inpainting:** One of the major problem with CNNs for image inpainting is that they provide equal weight to each spatial pixel in the image and hence are unable to discriminate between the hole pixels and non-hole pixels. To go around this problem, [34] introduced a partial convolution that would allow different weightage for the hole and non-hole pixels. They applied a convolutional operation only on the hole pixels and then followed it by a rule-based update of the mask for the following layers. In [52], the authors improved upon the idea of masked convolutions by introducing gated convolutions. Instead of the rule-based update of the mask, they introduced a trainable approach to find the mask values, where the masks are calculated using convolution operation and then multiplied by the spatial features to assign different weights for hole and non-hole pixels. In this work, we build upon this gated convolution framework to propose our method.

**Graph Neural Networks (GNN):** Recently there has been a growing interest [16, 30, 26] to extend the deep learning approaches for the graph-related data. The conventional CNNs can be seen as a special case of graph data in which each spatial pixel is connected by its surrounding pixels. Graph neural networks can increase the network's overall receptive field and hence enforce global consistency in the predictions [35]. Despite the significant improvement in these methods, there has been a limited use of GNNs in image inpainting. GNNs have been used in some of the related fields such as image super-resolution [59], semantic segmentation [31, 55], image de-noising [44, 45] etc. [59] models the correlation between the cross-scale similar patches as a graph and introduce and patch aggregation module to build the high-resolution image from the low-resolution counterpart. In [44, 45], the authors present a non-local aggregation block that uses a graph neural network for aggregating the features from far away pixels. To

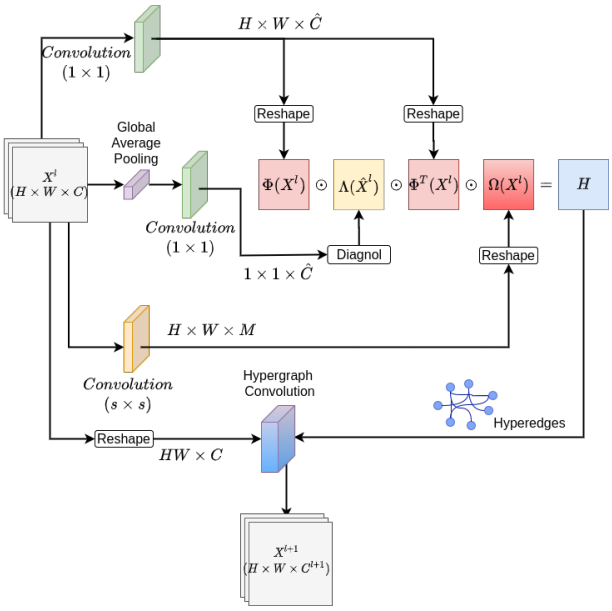


**Figure 2:** Overview of our proposed hypergraph convolution on the spatial features. First, we compute the incidence matrix $H$ using the information gathered from the input features and then we compute the hypergraph convolution using the calculated incidence matrix as given by Eq - (5).

build the graph, they find the distance between the spatial features and chooses k-nearest neighbors. In [31], the authors introduce a method for finding the similarity matrix for the graph, which is formed using each spatial pixel as vertex, using a data-dependent technique. They further use it in a pyramid based structure for the task of semantic segmentation. We extend this technique for the hypergraph neural networks to model a much more complex relation between the pixels using hyperedges, connecting more than two nodes using a single edge.

GNNs are efficient and can easily handle the long-range contextual information in the image, but they cannot accurately represent the non-pair relations among the data. Hypergraphs are a more generalized version of the graph in which a hyperedge can connect any number of vertices. Recently many researchers are using hypergraphs to represent their data in the deep learning approaches [50, 25]. In [13], the authors proposed hypergraph neural network (HGNN) which introduces spectral convolution on hypergraphs, using the regularization framework introduced in [58]. In [2], the authors introduce a hypergraph attention module. The hypergraph attention module further exerts an attention mechanism to learn the dynamic connections of hyperedges. Both of the previous methods cannot handle the dynamic structure of the hypergraphs. [22] introduces an idea of dynamic hypergraph construction, using the k-NN clustering method. This method is able to manage the dynamic nature of the input data, but it limits the number of nodes that can be connected. We propose a hypergraph inspired image inpainting method which can learn the hypergraph structure from the input data.

## 3. Methodology

We start the method discussion with brief introduction to spectral convolution on hypergraphs and then present the details oftrainable hypergraph module (Figure 2). Later, we discuss details of the inpainting network.

### 3.1. Hypergraph Convolution

Hypergraphs structure is used in many computer vision tasks to model the high-order constraints on the data which cannot be accommodated in the conventional graph structure. Unlike the pairwise connections in graphs, hypergraph contains hyperedge which can connect two or more vertices. A hypergraph is defined as $G = (V, E, \mathbf{W})$, where $V = \{v_1, \ldots, v_N\}$ is the set of vertices, $E = \{e_1, \ldots, e_M\}$ represents the set of hyperedges, and $\mathbf{W} \in \mathbb{R}^{M \times M}$ is a diagonal matrix containing the weight of each edge. The hypergraph $G$ can be represented by the incidence matrix $\mathbf{H} \in \mathbb{R}^{N \times M}$. For a vertex $v \in V$, and an edge $e \in E$ the incidence matrix is defined as,

$$h(v, e) = \begin{cases} 1 & \text{if } v \in e \\ 0 & \text{if } v \notin e \end{cases} \tag{1}$$

For a given hypergraph $G$, the vertex degree, $\mathbf{D} \in \mathbb{R}^{N \times N}$, and hyperedge degree $\mathbf{B} \in \mathbb{R}^{M \times M}$ are defined as $D_{ii} = \sum_{e=1}^{M} W_{ee} H_{ie}$, and $B_{ee} = \sum_{i=1}^{N} H_{ie}$ respectively,

Next, the incidence matrix $H$, vertex degree $D$, and hyperedge degree $B$, are used to compute the normalized hypergraph Laplacian matrix $\mathbf{\Delta} \in \mathbb{R}^{N \times N}$ as, $\mathbf{\Delta} = \mathbf{I} - \mathbf{D^{-1/2} H B^{-1} H^T D^{-1/2}}$. It is a symmetric positive semi-definite matrix [58] and the eigen decomposition $\mathbf{\Delta} = \mathbf{\Phi \Lambda \Phi^T}$ can be used to get the complete set of the orthonormal eigenvectors $\mathbf{\Phi} = \{\phi_1, \ldots, \phi_N\}$ and a diagonal matrix $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_N)$ containing the corresponding non-negative eigenvalues. We can define the hypergraph Fourier Transform, $\hat{\mathbf{x}} = \mathbf{\Phi^T x}$, which transforms a signal $\mathbf{x} = (x_1, \ldots, x_N)$ into the spectral domain spanned by the basis of $\Phi$, also known as Fourier basis. Generalizing the convolutional theorem into structured space of hypergraphs, the convolution on the signal $x \in \mathbb{R}^N$ can be defined as:

$$g \circledast x = \mathbf{\Phi} g(\mathbf{\Lambda}) \mathbf{\Phi^T} x \tag{2}$$

where $g(\mathbf{\Lambda}) = diag(g(\lambda_1), \ldots, \lambda_N)$ is a function of the Fourier coefficients [13]. However, to compute the convolution on the signal $x$ it would be required to compute the eigenvectors of the Laplacian matrix. So, Defferrand et al. [9] parameterized $g(\mathbf{\Lambda})$ with truncated chebyshev polynomials up to $K^{th}$ order, hence defining the convolutional

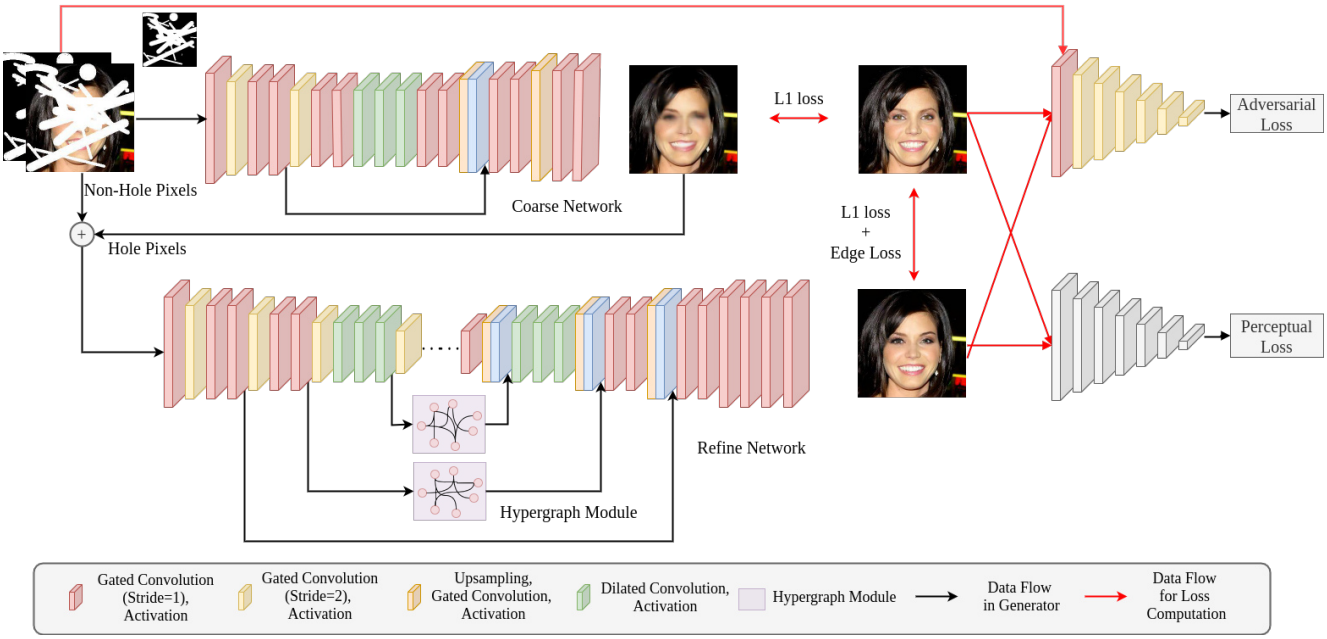


**Figure 3:** Overview of our proposed network for Image inpainting. The Coarse network roughly completes the missing holes. Later, the hypergraph convolution based Refine network generates the final high quality completed image.

operation on the hypergraph signal as,

$$g \circledast x = \sum_{k=0}^{K} \theta_k T_k(\mathbf{\Delta}) x \tag{3}$$

where $\theta_k$ is a vector of chebyshev polynomial coefficients, and $T_k$ is the chebyshev polynomial. In Eq - (3), we excluded the calculation of eigenvectors of the Laplacian matrix. We can further simplify the formulation by limiting $K = 1$. In [28], it is approximated $\lambda_{max} \approx 2$ because of the scale adaptability of neural networks. Therefore, our convolutional operation on the hypergraph signal becomes,

$$g \circledast x \approx \theta \mathbf{D}^{-1/2}\mathbf{H}\mathbf{W}\mathbf{B}^{-1}\mathbf{H}^T\mathbf{D}^{-1/2}x \tag{4}$$

where $\theta$ is the only chebyshev coefficient left after taking $K = 1$ chebyshev polynomials.

For a given hypergraph signal $X^l \in \mathbb{R}^{N \times C_l}$, where $C_l$ is the dimension of the feature vector of input at layer $l$, we can generalize the convolution operation in multi-layer hypergraph convolutional network as,

$$\mathbf{X^{l+1}} = \sigma(\mathbf{D}^{-1/2}\mathbf{H}\mathbf{W}\mathbf{B}^{-1}\mathbf{H}^T\mathbf{D}^{-1/2}\mathbf{X^l}\mathbf{\Theta}) \tag{5}$$

where $\Theta \in \mathbb{R}^{C_l \times C_{l+1}}$ is the learnable parameter, and $\sigma$ is the non-linear activation function.

In Eq - (5), the incidence matrix $H$ encodes the hypergraph structure, which is further used to propagate information among the hypergraph nodes. Hence, it can be easily seen that better hyperedges' connections would lead to better information sharing among the nodes, further improving the completed image. Currently, the formation of these incidence matrices is limited to non-trainable methods.

### 3.2. Hypergraphs Convolution on spatial features

To overcome the limited receptive field of CNN architectures, recent studies transform the spatial feature maps into the graph-based structure and perform graph convolution to capture the global relationship between the data [5, 32, 33]. It can be easily observed that simple graphs are a special case of hypergraphs where each hyperedge connects only two node. These simple graph can easily represent the pair-wise relationship among data but it is difficult to represent the complex relationship among the spatial features of the image because of which we use hypergraphs instead of graphs. To transform the spatial features $F^l \in \mathbb{R}^{h \times \times w \times c}$ into the graph-like structure, we consider each spatial feature as a node having a feature vector of dimension $c$, $X^l \in \mathbb{R}^{hw \times c}$.

In the recent studies [13, 2, 49], for the visual classification problem, the incidence matrix $H$ is formed using the euclidean distance between features of the images [13, 49]. To better capture the image's intra-spatial structure, we propose an improved incidence matrix that can learn to capture long-term intra-spatial dependencies. Instead of the euclidean distance between the spatial features, we use the cross correlation of the spatial features to calculate each node's contribution in each hyperedge. The incidence matrix $H$ contains the information regarding each node's contribution in each hyperedge and it is expressed as,

$$H = \Psi(X)\Lambda(X)\Psi(X)^T\Omega(X) \tag{6}$$

where $\Psi(X) \in \mathbb{R}^{N \times \hat{C}}$, is the linear embedding of the in-

| | | CelebA-HQ | | | | | Places2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| % | Metrics | PICNet[57] | GMCNN[47] | DeepFill[52] | SN[51] | Ours | PICNet[57] | GMCNN[47] | DeepFill[52] | Ours |
| 0.1-0.2 | PSNR ↑ | 30.29 | 30.98 | 31.21 | 30.16 | **33.34** | 29.6 | 30.35 | 29.87 | **32.21** |
| | SSIM ↑ | 0.971 | 0.977 | 0.9744 | 0.969 | **0.985** | 0.953 | 0.964 | 0.960 | **0.974** |
| | FID ↓ | 6.223 | 6.487 | 3.786 | 7.143 | **2.177** | 13.269 | 8.687 | 9.567 | **6.465** |
| | L1 ↓ | 2.004 | 1.9193 | 1.622 | 2.203 | **0.683** | 1.340 | 1.192 | 1.240 | **0.745** |
| | L2 ↓ | 0.234 | 0.203 | 0.187 | 0.235 | **0.124** | 0.309 | 0.273 | 0.306 | **0.184** |
| 0.2-0.3 | PSNR ↑ | 28.10 | 28.84 | 28.52 | 28.55 | **30.23** | 26.54 | 27.35 | 26.89 | **29.13** |
| | SSIM ↑ | 0.951 | 0.961 | 0.955 | 0.954 | **0.970** | 0.911 | 0.932 | 0.924 | **0.950** |
| | FID ↓ | 8.343 | 8.931 | 6.013 | 9.342 | **4.026** | 21.496 | 14.250 | 15.007 | **11.175** |
| | L1 ↓ | 2.508 | 2.303 | 2.179 | 2.560 | **1.250** | 2.230 | 1.938 | 2.030 | **1.390** |
| | L2 ↓ | 0.375 | 0.329 | 0.339 | 0.339 | **0.240** | 0.603 | 0.523 | 0.588 | **0.357** |
| 0.3-0.4 | PSNR ↑ | 26.38 | 26.80 | 26.62 | 27.00 | **28.22** | 24.50 | 25.37 | 24.93 | **27.17** |
| | SSIM ↑ | 0.927 | 0.941 | 0.933 | 0.934 | **0.954** | 0.862 | 0.897 | 0.885 | **0.923** |
| | FID ↓ | 10.334 | 15.840 | 8.650 | 11.930 | **5.991** | 29.340 | 19.900 | 21.566 | **16.116** |
| | L1 ↓ | 3.070 | 2.819 | 2.784 | 3.047 | **1.834** | 3.190 | 2.710 | 2.850 | **2.048** |
| | L2 ↓ | 0.551 | 0.524 | 0.516 | 0.476 | **0.374** | 0.963 | 0.804 | 0.902 | **0.545** |
| 0.4-0.5 | PSNR ↑ | 24.92 | 24.49 | 25.08 | 25.39 | **26.69** | 22.95 | 23.79 | 23.40 | **25.68** |
| | SSIM ↑ | 0.898 | 0.906 | 0.906 | 0.903 | **0.935** | 0.806 | 0.853 | 0.838 | **0.890** |
| | FID ↓ | 13.015 | 33.358 | 11.410 | 15.960 | **7.942** | 37.399 | 25.589 | 27.624 | **21.211** |
| | L1 ↓ | 3.730 | 3.588 | 3.469 | 3.827 | **2.454** | 4.218 | 3.574 | 3.750 | **2.750** |
| | L2 ↓ | 0.773 | 0.914 | 0.738 | 0.690 | **0.530** | 1.360 | 1.148 | 1.260 | **0.760** |
| 0.5-0.6 | PSNR ↑ | 23.47 | 21.33 | 23.67 | 22.90 | **25.27** | 21.51 | 22.37 | 22.08 | **24.35** |
| | SSIM ↑ | 0.86 | 0.842 | 0.873 | 0.838 | **0.911** | 0.742 | 0.802 | 0.785 | **0.851** |
| | FID ↓ | 16.13 | 64.449 | 14.852 | 25.440 | **10.087** | 45.630 | 33.239 | 34.387 | **28.237** |
| | L1 ↓ | 4.529 | 5.120 | 4.270 | 5.530 | **3.146** | 5.380 | 4.559 | 4.746 | **3.530** |
| | L2 ↓ | 1.074 | 2.008 | 1.010 | 1.245 | **0.730** | 1.900 | 1.580 | 1.710 | **1.030** |

**Table 1:** Quantitative comparison of our method with state-of-the-art methods on CelebA-HQ and Places2 datasets wrt different hole percentages. (↑ Higher is better, ↓ Lower is better)

put features followed by a non-linear activation function (in our case ReLU function), and $\hat{C}$ is the dimension of the feature vector after the linear embedding, $\Lambda(X) \in \mathbb{R}^{\hat{C}\times\hat{C}}$ is a diagonal matrix, which helps in learning a better distance metric among the nodes for the incidence matrix $H$, and $\Omega(X) \in \mathbb{R}^{N\times M}$ helps to determine the contribution of each node for each hyperedge, and $m$ is the number of hyperedges in the hypergraph. All $\Psi(X)$, $\Lambda(X)$, and $\Omega$ are data-dependent matrices. $\Psi(X)$ is implemented by $1 \times 1$ convolution on the input features, $\Lambda(X)$ is implemented by channel-wise global average pooling followed by a $1 \times 1$ convolution as used in [19], and $\Omega(X)$ is used to capture the global relationship of the features to develop better hyperedges (implemented using $s \times s$ filter, we keep $s = 7$).

We compute the incidence matrix $H^l$ as shown in the (Figure-2), and it can be formulated as,

$$\begin{aligned} \mathbf{H}^l &= \mathbf{\Psi}(X^l)\mathbf{\Lambda}(X^l)\mathbf{\Psi}(X^l)^T\mathbf{\Omega}(X^l) \\ \mathbf{\Psi}(X^l) &= conv(X^l, W^l_\Psi) \\ \mathbf{\Lambda}(X^l) &= diag(conv(\hat{X}^l, W^l_\Lambda)) \\ \mathbf{\Omega}(X^l) &= conv(X^l, W^l_\Omega) \end{aligned} \tag{7}$$

where $\hat{X}^l \in \mathbb{R}^{1\times1\times\hat{C}}$ is the feature map produced after global pooling of the input features, $W^l_\Psi$, $W^l_\Lambda$ and $W^l_\Omega$ are the learnable parameters for the linear embedding. To avoid the negative values in the incidence matrix $H$, which could lead to imaginary values in the degree matrices, we use the absolute values in the incidence matrix. Then we formulate our hypergraph convolution layer on spatial features as,

$$\mathbf{X^{l+1}} = \sigma(\mathbf{\Delta X}^l\mathbf{\Theta}) \tag{8}$$

where $\Theta \in \mathbb{R}^{C_l\times C_{l+1}}$ is the learnable parameters, $\sigma$ is the non-linear activation, (we use Exponential Linear Unit (ELU) [7]), $X^l$ are the input features and $X_{l+1}$ are the output features.

### 3.3. Inpainting Network Architecture

The network architecture of our proposed method is given in (Figure-3). We use a two-stage coarse-to-fine network architecture. The coarse network roughly fills the missing region, which is naively blended with the input image, then refine network predicts the finer results with sharp edges. In the refine network, we use the hypergraph layer with high-level feature maps to increase the receptive field of our network and obtain distant global information of the image. We use dilated convolutions [20] for our coarse and refine network to further expand our network's receptive field. We also used gated convolutions [52] to improve our

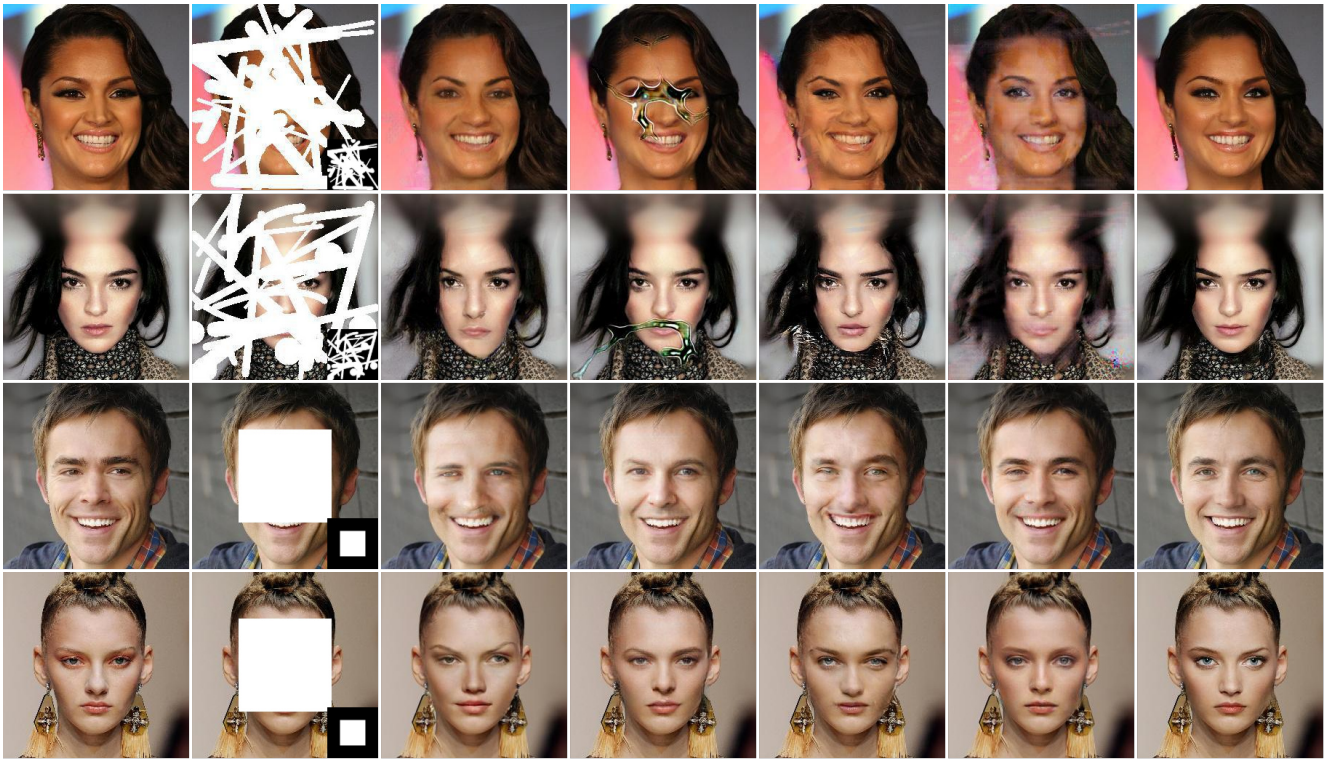


**Figure 4:** Qualitative Comparison on the CelebA-HQ dataset. From left to right $a$) Ground Truth, $b$) Input Image, $c$) Pluralistic (PICNet) [57], $d$) GMCNN [47], $e$) DeepFill-V2 [52], $f$) Shift-Net (SN) [51], $g$) Ours. All the images are scaled to the size $256 \times 256$.

performance on an irregular mask which can be defined as,

$$\begin{aligned} Gating &= Conv(W_g, I) \\ Features &= Conv(W_f, I) \\ O &= \phi(Features) \odot \sigma(Gating) \end{aligned} \tag{9}$$

where $W_g$ and $W_f$ are two different learnable parameters for convolution operation, $\sigma$ is the sigmoid activation function, and $\phi$ is a non-linear activation function, such as ReLU, ELU, and LeakyReLU. We also remove batch normalization from all our convolutional layers as they can deteriorate the color coherency of the completed image [20]. The architecture of the discriminator used is our method similar to the PatchGAN [21]. We remove all the batch normalization layers and replace all the convolution layers with the gated convolution using which enforces local consistency in the completed image. We provide the discriminator with both mask and completed/original image.

### 3.4. Loss Functions

Given an input image $I_{in}$ with holes, and a binary mask $R$ (1 for holes), our network predicts $I_{coarse}$, and $I_{refine}$ from the coarse and refine network respectively. For the given ground truth $I_{gt}$, we train our network on the combination of losses consisting of content loss, adversarial loss, perceptual loss, and edge-loss.

To force the pixel level consistency we use the $L1$ loss on both coarse $I_{coarse}$ and refine $I_{refine}$ outputs. We define content loss as,

$$\begin{aligned} \mathcal{L}_{hole} &= ||R \odot (I_{refine} - I_{gt})||_1 \\ &+ \frac{1}{2}||R \odot (I_{coarse} - I_{gt})||_1 \\ \mathcal{L}_{valid} &= ||(1-R) \odot (I_{refine} - I_{gt})||_1 \\ &+ \frac{1}{2}||(1-R) \odot (I_{coarse} - I_{gt})||_1 \end{aligned} \tag{10}$$

where $L_{hole}$ is the loss for the hole pixels values, and $L_{valid}$ is the loss for the non-pixels values.

The adversarial loss has been shown to be effective to generate realistic and globally consistent images [15, 29, 36]. The adversarial loss can be formulated as a min-max problem,

$$\begin{aligned} \mathcal{L}_{GAN} = \max_D \min_G \mathbb{E}[log(D(I_{gt}, R))] \\ + \mathbb{E}[log(1 - D(G(I_{in}), R)] \end{aligned} \tag{11}$$

where $G$ is our image inpainting network which predicts the final completed image $I_{refine}$, and $D$ is the Discriminator. Perceptual loss has been used in many applications such as image super-resolution [4, 37], image deblurring [60, 41], and style transfer [23, 14]. For a given input $x$, let $\phi_l(x)$ denote the high-dimension features of $l^{th}$ activation layer


**Figure 5:** Qualitative Comparison on the Places2 dataset. From left to right $a$) Ground Truth, $b$) Input Image, $c$) Pluralistic (PICNet) [57], $d$) GMCNN [47], $e$) DeepFill-V2 [52], $f$) Ours. All the images are scaled to the size $256 \times 256$

of the pre-trained network, then the perceptual loss can be defined as,

$$\mathcal{L}_p = \sum_l ||\phi_l(G(I_{in})) - \phi_l(I_{gt})||_1 \tag{12}$$

We compute perceptual loss for final prediction $I_{refine}$, and $I_{comp}$, where $I_{comp}$ is the final prediction but the non-hole pixels directly set to ground-truth [34].
To maintain the edges in the predicted images, we use the edge-preserving loss [38]. It can be defined as,

$$\mathcal{L}_{edge} = ||E(I_{refine}) - E(I_{gt})||_1 \tag{13}$$

where $E(\cdot)$ is the sobel filter. So our total loss $\mathcal{L}_{total}$ can be written as,

$$\begin{aligned}\mathcal{L}_{total} =& \lambda_{hole}\mathcal{L}_{hole} + \lambda_{valid}\mathcal{L}_{valid} + \lambda_{adv}\mathcal{L}_{adv} \\ &+ \lambda_p\mathcal{L}_p + \lambda_{edge}\mathcal{L}_{edge}\end{aligned} \tag{14}$$

where $\lambda_{hole}, \lambda_{valid}, \lambda_{adv}, \lambda_p$ and $\lambda_{edge}$ are the weights to balance the hole, valid, adversarial, perceptual and edge loss respectively.

### 3.5. Incremental Training

Training the deep learning approach for image inpainting on a random mask is an arduous task because of the random hole size in the testing phase. To handle this issue, we introduce a simple yet effective training technique for image inpainting. Initially, we start training our training with a very small hole percentage. Therefore initially, the network can learn to output the non-hole pixels accurately. Then gradually, we increase the hole percentage so that the network can learn a better mapping for large holes. Specifically, we train our network for K iterations and then increase the hole size.

## 4. Experiments

### 4.1. Implementation Details

During training, we linearly scale the image's values in the range $[0, 1]$. We trained our model on NVIDIA 1080Ti GPU with the image resolution of $256 \times 256$ with a batch size of 1. We use Adams optimization [27] with $\beta_1 = 0.9$, and $\beta_2 = 0.999$ and the initial learning rate of $1 \times 10^{-4}$ decreasing by a factor of $0.96$ after each epoch.[1]

### 4.2. Datasets

We evaluate our proposed network on four publicly available datasets, including CelebA-HQ [24], Paris Street View [10], Facades Dataset [43], and ten scenes in Places2 dataset [1]. The CelebA-HQ dataset contains $30,000$ images of faces, we randomly sample $28,000$ images for training and $2,000$ images for testing. There are $14,900$ images for training and $100$ images testing in the Paris Street View Dataset. The Facades Datasets contains $400$ training images, $100$ validation images, and $106$ testing images(For Facades Dataset we fine-tune the model trained on Paris

[1]The code is available at `https://github.com/GouravWadhwa/Hypergraphs-Image-Inpainting`

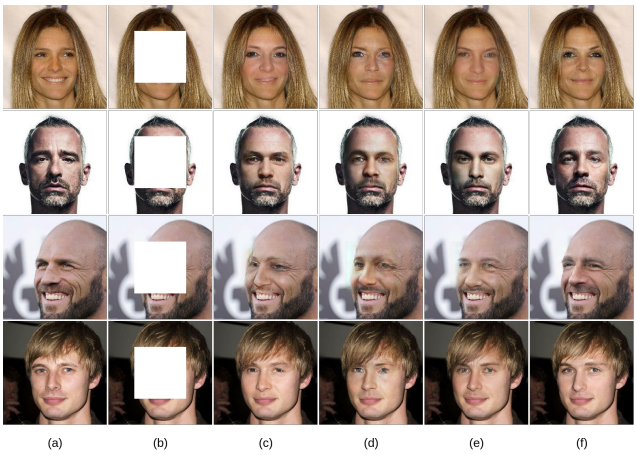


**Figure 6:** Comparison of different variants of our proposed method on CelebA-HQ dataset. From left to right $a$) Ground Truth (GT), $b$) Input Image, $c$) w/o hypergraph attention and w/o gated convolution in discriminator, $d$) w/o gated convolution in discriminator, $e$) w/o hypergraph attention, and $f$) Original method.

Street View dataset). Places365-Standard Dataset contains 1.6 million training images from 365 scenes. We choose ten different scenes including canyon, field_road, field-cultivated, field-urban, synagogue-outdoor, tundra, valley, canal-natural, and canal-urban. Each category contains $5,000$ training images, 100 validation images, and 900 testing images. We also use data augmentation, such as flipping and rotating for the Places2 and Paris Street View datasets. To evaluate our results, we train our model both on the center-fixed hole and random hole. We remove $25\%$ of the center pixels for the center-fixed hole, and for the random hole, we simulate spots, tears, scratches, and manual erasing with brushes.

### 4.3. Comparison with Existing Work

We compare our method against the existing works, including multi-column image inpainting (GMCNN)[47], DeepFill [52], Pluralistic Image Inpainting (PICNet) [57] (As PICNet can generate multiple results we choose the best result based on the discriminator scores for fair comparison), and Shift-Net (SN) [51], for both centre masks, and irregular masks.

**Quantitative Comparison:** As mentioned by [53], there is no good numeric method for evaluating the image inpainting results because of the existence of many plausible results for the same image and mask. Nevertheless, we report, in Table-1, our evaluation results in terms of $l_1$ error, $l_2$ error, PSNR, MS-SSIM [48], and Frechet Inception Distance (FID) [18] on the validation set of places2 and testing set CelebA-HQ datasets. In the table, we compare different approaches on random mask of different hole percentages for the both datasets. As shown in the table, our method outperforms all the existence methods in terms of $l_1$ loss, $l_2$ loss, PSNR, MS-SSIM, and FID.

**Qualitative Comparison:** Figure-4, and Figure-5 shows the comparison between our method and the other existing methods on CelebA-HQ and Places2 datasets respectively. We observe that our method produce much more semantically plausible and globally consistent results even for for much larger mask region. Earlier methods performs good en enough for small mask percentage but there performance deteriorates as the mask size increases. Especially, GM-CNN, and DeepFill-V2 produces severe artifacts when the hole size increases beyond $50\%$. The outputs of Shift-Net (SN) algorithm does not produce color consistent outputs. PICNet produces semantically plausible and clear results but the outputs produced are not globally consistent, this is because PICNet of the discriminator which constraints the the network to produce clearer image but looses the structural consistency of the image and hence produces artifacts in the predicted image. Our method is able to produce much more plausible and realistic outputs because of the hypergraphs, which helps the generator to learn the global context of the image, and the gated convolutions used in discriminator helps the generator learn the local contents of the image.

### 4.4. Ablation Study

We further perform experiments on CelebA-HQ dataset to study the effects of different components of our introduced methodology. In figure-6, we show the comparison between different variants of our method, including $a$) w/o hypergraph attention mechanism with normal convolution in discriminator $b$) Replacing gated convolution with normal convolution in the discriminator, and $c$) w/o hypergraph attention mechanism. Using normal convolution in discriminator affects the local consistency of the image and produces artifacts in the completed image. Not using hypergraph attention mechanism disturbs the global color consistency of the completed image because hypergraph convolution provides the global structure of the image.

## 5. Conclusion

In this paper, we proposed a Hypergraph convolution based image inpainting technique, where the hypergraph incidence matrix $H$, is data-dependent and enables us to use a trainable mechanism to connect nodes using hyperedges. Using hypergraphs helps the generator to get the global context of the image. We also propose the use of gated convolution in discriminator which helps the discriminator enforce a local consistency in the image. Our proposed method produces a final image which is semantically plausible and globally consistent. Our experimental results indicate that our method gets better performance than any of the state-of-the-art methods and improves the quality of the completed image. Further, we can easily extend the idea of hypergraph convolution on spatial features in any other application to learn the global context of the image.

## References


[1] Zhou B., Lapedriza A., Khosla A., A. Oliva, and Torralba A. Places: A 10 million image database for scene recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] Song Bai, Feihu Zhang, and Philip H.S. Torr. Hypergraph convolution and hypergraph attention. *arXiv preprint, arXiv:1901.08150*, 2019.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (TOG)*, 2009.

[4] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019.

[6] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *ICCV*, 2015.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[8] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. In *IEEE Transactions on image processing*, 2004.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

[10] Doersch, A C. Singh, S. Gupta, Sivic J., and A Efros. What makes paris look like paris? In *ACM Transactions on Graphics 31(4)*, 2012.

[11] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *ACM Transactions on graphics (TOG)*, 2003.

[12] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, 1999.

[13] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, 2019.

[14] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[16] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[17] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *European Conference on Computer Vision, (ECCV)*, 2012.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[20] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. In *ACM Transactions on Graphics (ToG)*, 2017.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[22] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[23] J. Johnson, A. Alahi, and L Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, 2016.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[25] Eun-Sol Kim, Woo Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. Hypergraph attention networks for multimodal learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[27] D.P. Kingma and Ba J.L. Adam: A method for stochastic optimization. In *international conference on learning representations*, 2015.

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[29] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[30] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[31] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[32] Yin Li, Abhinav Gupta, and Beyond grids. Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems 31*, 2018.

[33] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *NIPS*, 2018.

[34] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, 2018.

[35] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. Geniepath: Graph neural networks with adaptive receptive paths. *arXiv preprint arXiv:1802.00910v3*, 2018.

[36] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.

[37] Kamyar Nazeri, Harrish Thasarathan, and Mehran Ebrahimi. Edge-informed single image super-resolution. In *ICCV-Worksop*, 2018.

[38] Ram Krishna Pandey, Nabagata Saha, Samarjit Karmakar, and A G Ramakrishnan. MSCE: An edge preserving robust loss function for improving super-resolution algorithms. In *International Conference on Neural Information Processing, ICONIP*, 2018.

[39] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[40] J. Shen and T. F. Chan. Mathematical models for local nontexture inpaintings. In *SIAM booktitle on Applied Mathematics*, 2002.

[41] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[42] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C.-C. Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. *arXiv preprint arXiv:1711.08590v5*, 2018.

[43] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013.

[44] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Deep graph-convolutional image denoising. *arXiv preprint arXiv:1907.08448v1*, 2019.

[45] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Image denoising with graph-convolutional neural networks. *arXiv preprint arXiv:1905.12281v1*, 2019.

[46] Tianyang Wang, Zhengrui Qin, and Michelle Zhu. An elu network with total variation for image denoising. *arXiv preprint arXiv:1708.04317*, 2017.

[47] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 331–340, 2018.

[48] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems and Computers*, 2003.

[49] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, pages 1509–1520. Curran Associates, Inc., 2019.

[50] Yichao Yan, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Ying Tai, and Ling Shao. Learning multi-granular hypergraphs for video-based person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[51] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[52] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[53] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[54] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high quality image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[55] Kaihua Zhang, Tengpeng Li, Shiwen Shen, Bo Liu, Jin Chen, and Qingshan Liu. Adaptive graph convolutional network with attention graph clustering for co-saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[56] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *arXiv preprint arXiv:1603.08511*, 2016.

[57] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[58] D. Zhou, J. Huang, and Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *NIPS*, 2007.

[59] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. *arXiv preprint arXiv:1907.08448v1*, 2019.

[60] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy Ren. DAVANet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.