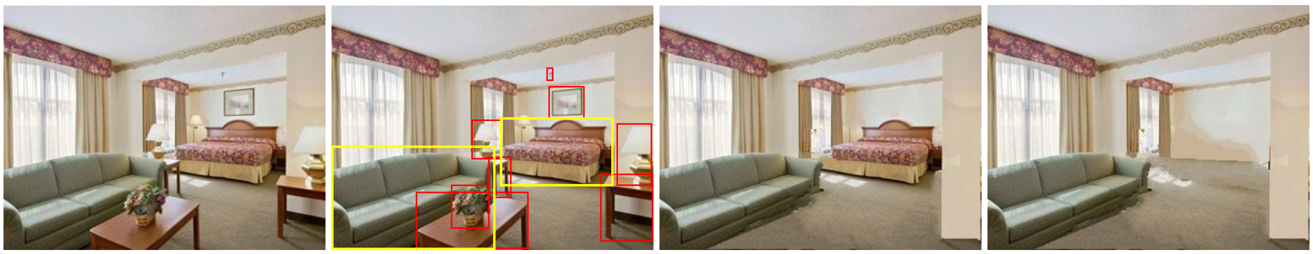
Scene Essence

Jiayan Qiu[1], Yiding Yang[2], Xinchao Wang[3,2], Dacheng Tao[1]
[1]The University of Sydney, [2]Stevens Institute of Technology, [3]National University of Singapore
jiayan.qiu.1991@outlook.com, yyang99@stevens.edu, xinchao@nus.edu.sg, dacheng.tao@sydney.edu.au



(a) (b) (c) (d)

Figure 1: Given an input image of a hotel room (a), we detect its scene objects in (b) and learn to identify the *Scene Essence* that comprises a collection of essential elements for recognizing the scene, as labeled by the yellow bounding boxes. The image with essential elements preserved but minor ones inpainted are shown in (c), which, still, would be visually recognized as a hotel room. Should we further wipe off elements from the Scene Essence, in this case the bed, the scene will be interpreted as a living room.

Abstract

What scene elements, if any, are indispensable for recognizing a scene? We strive to answer this question through the lens of an exotic learning scheme. Our goal is to identify a collection of such pivotal elements, which we term as Scene Essence, to be those that would alter scene recognition if taken out from the scene. To this end, we devise a novel approach that learns to partition the scene objects into two groups, essential ones and minor ones, under the supervision that if only the essential ones are kept while the minor ones are erased in the input image, a scene recognizer would preserve its original prediction. Specifically, we introduce a learnable graph neural network (GNN) for labelling scene objects, based on which the minor ones are wiped off by an off-the-shelf image inpainter. The features of the inpainted image derived in this way, together with those learned from the GNN with the minor-object nodes pruned, are expected to fool the scene discriminator. Both subjective and objective evaluations on Places365, SUN397, and MIT67 datasets demonstrate that, the learned Scene Essence yields a visually plausible image that convincingly retains the original scene category.

1. Introduction

Looking at the image in Fig. 1(a), we may effortlessly tell that it is a scene of a hotel room. But if we are asked to pinpoint a few indispensable objects in the scene, if any, that dedicate our recognition, it might take us some effort to figure them out: maybe the sofa, or the table, or a combination of both? If we human observers find this to be a non-trivial task, shall we expect deep networks to be competent?

In this paper, we target at learning to extract a collection of such scene objects, which, together with the scene background, are coined as *Scene Essence*. In other words, Scene Essence comprises the scene background and pivotal scene objects, if any, that jointly make a scene a scene, and hence serves as a scene signature. We show an example of the learned Scene Essence, in Fig. 1(c), where only the sofa and the bed are preserved while all other objects are wiped off by an off-the-shelf image inpainter [91]. This derived Scene Essence image successfully fools a state-of-the-art scene recognizer [99], since it is still categorized as a hotel room; in fact, even when we human observers look at this image, likely we will not even doubt it is being a hotel-room image. Should we, however, take one more object from the Scene Essence, for example the bed as shown in Fig. 1(c), the scene recognizer will immediately alter its prediction, in this case to a living room, which indeed appears to be such

Figure 2: (a) and (b) respectively show the original dorm scene image and its corresponding Scene Essence; (c) shows the learned Scene Essence if provided with a label of bedroom, and (d) shows the one learned with a label of office.

for human.

Despite prior efforts on attribution maps [65, 73] and activation maps [101, 54] also aim to interpret the scene recognition rationale, the proposed Scene Essence distinguishes itself from the perspective that it reasons at object level and meanwhile delivers a minimum set of objects to ensure the image being recognized as the original category. Furthermore, Scene Essence comes with other unique and interesting properties, such as generating images of other categories and hence enabling scene transfer. For example, an image of the dorm category is shown in Fig. 2(a); when trained with the original label, Scene Essence will remove the dispensable objects like the books on the bed, and keep the essential ones as in Fig. 2(b). If, however, we train our network with other labels, such as bedroom or office, Scene Essence would consequently produce images displayed respectively in Fig. 2(c) and Fig. 2(d), which are indeed visually convincing scenes from the two categories and hence offer an exotic and inexpensive way of conducting scene transfer.

We devise a novel approach to learning Scene Essence, by explicitly accounting for both object-level semantics and visual evidences. The core idea here is to learn a partition of scene objects into two groups, *essential* ones and *minor ones*, such that if the minor ones are erased by an image inpainter while the essential ones are preserved, a scene classifier would not alter its predicted label. To this end, we propose an innovative network architecture that first takes an image as input and conducts object detection using an off-the-shelf detector module. Each detected object is modeled as a node in a scene graph, which is then fed into a learnable hierarchical Graph Neural Network (GNN) for labeling each node as essential or minor. Next, an off-the-shelf image inpainter is introduced to erase the minor objects and produce the Scene Essence image, whose visual feature is concatenated with the features learned from GNN and afterwards fed into a scene discriminator. The GNN module, therefore, learns to update its parameters from the supervision back-propagated from the scene discriminator and the image inpaitner, and eventually specializes in identifying essential objects.

In sum, our contribution is an exotic scene signature, termed as Scene Essence, that maintains a minimum set of scene objects to preserve its predicted label and meanwhile offers an inexpensive way for scene transfer. Scene Essence is derived via a novel network architecture, in which a GNN learns to categorize scene objects under the supervision that if only the essential ones are kept while the rest ones are wiped off, a scene recognizer will stick to its original prediction. We conduct extensive objective and subjective experiments to evaluate Scene Essence in terms of recognition accuracy, visual quality, and inter-category transferability, and showcase that it may readily serve as a new option for interpreting scene recognition rationale at the object level.

2. Related Work

We briefly review here prior works related to ours, including scene recognition, discriminative region detection, graph convolutional network, and image inpainting.

Scene Recognition. Earlier scene recognition methods learn to understand the scene from the spatial correlation between handcrafted features of random regions [60, 30, 55, 38, 56]. Due to the development of deep learning [36, 66, 25], data-driven based feature learning methods are proposed and have achieved promising performance on scene recognition [102, 48, 68, 100, 29, 84, 43]. More recently, the embedded structural information in the image is utilized for scene understanding [11, 6, 69, 59]. However, none of the existing methods tried to learn to find the Scene Essence that maintains the minimum set of elements to preserve the predicted category.

Discriminative Region Detection. Traditional discriminative region detection methods rely on handcrafted features to locate the discriminative areas [67, 32]. In recent years, deep-learning-based methods have dominated discriminative region detection. They can be broadly categorized into three classes. Methods in the first class focus on the dense prediction [62, 45, 76], those in the second estimate activation maps for locating the discriminative regions [82, 13], and the ones in the third class explore the convolutional responses from CNNs [96, 98, 95, 81]. More recently, the structural attention mechanism is implemented to extract discriminative regions in scene image [11]. However, none of the existing methods explored the object-level discrimination.

Graph Neural Network. Earlier works on graph-related tasks either assume the node features to be predefined [58, 79, 80, 51, 50, 37], or apply iterative schemes for learning node representations, which are time consuming [17, 63, 21, 72]. Recently, graph neural networks have been proposed to learn graph features. They can be coarsely categorized into two types: spectral-based approaches, which aim to develop graph convolution based on the spectral theory [42, 40, 34, 78, 15, 26, 9], and spatial-based ones, which investigate information mutual dependency [75, 28, 12, 3, 18, 52, 22, 19, 53, 2, 1, 39, 49, 89, 88, 87, 94, 74]. More recently, the hierarchical GCN [90] is proposed to strengthen the learning capability and has achieved promising results.

Image Inpainting. Traditional image inpainting methods utilize the cross image correlation to inpaint the masked area [7, 4, 41, 5, 8, 14]. Thanks to the development of deep learning, especially the generative adversarial networks [20], many deep learning based inpainting algorithms have been proposed and delivered visually realistic results [61, 35, 23, 57, 92, 44, 71]. The more recent methods utilize both intra-image information and learning from large datasets, and gain significant improvement in terms of semantic continuity and visual authenticity [86, 91, 85, 93, 31, 101, 70, 92, 97]. However, image inpainting concerns only the inpainting process, but not the explicit object-level inference as done in our approach.

3. Method

In this section, we show the working scheme of the proposed approach in detail. As depicted in Fig. 3, our approach comprises four stages. In Stage 1, we utilize a detection module on the input image to detect the objects in the scene and extract their semantic and spatial information. In Stage 2, we model the scene image as a graph, in which each node corresponds to an object in the scene. Afterwards, we apply the GNN module to cluster the detected objects into two groups, Essential ones and Minor ones. The clustering result is then utilized for the inpainting mask generation and the structural feature extraction. In Stage 3, we first feed the scene image and the inpainting mask into the inpainting module for wiping the Minor objects off. The erased image, then goes through the visual scene recognition (VSR) module for visual feature extraction. In Stage 4, we concatenate the structural feature and the visual feature, and then feed the concatenated feature into the scene classifier.

3.1. Stage 1: Detection

We adopt a detection module in Stage 1 to detect the objects in the scene image so as to derive object-level features. Specifically, we implement the pretrained Mask-RCNN model [24] on the scene images. For each detected object, we construct a 1028-dimension feature vector that encodes both the semantic and spatial information. Features in first 1024 dimensions are taken directly from the last layer of Mask-RCNN to embrace the semantics, while features in the last four dimensions, namely upper-left and lower-right coordinates of the detection bounding box, are adopted to encode its spatial information. The 1028-dimension vectors are further fed to Stage 2, and taken to be the features of the corresponding node in the scene graph.

3.2. Stage 2: GNN Module

The second stage of our approach takes as input the instance semantics obtained in Stage 1, and models the interplays between the scene objects using a scene graph. Let N denotes the number of detected objects in Stage 1. We then construct a graph of N nodes and link all the pairs of the N nodes to form a complete graph. Each node in the graph holds a 1028-dimension feature.

We then feed the graph into the GNN module for clustering the objects into two groups, the Essential and Minor ones. Specifically, we represent the graph G as (A, F), where $A \in \{0,1\}^{N\times N}$ denotes the adjacency matrix, and $F \in \mathbb{R}^{N\times d}$ denotes the feature matrix with d-dimension node feature.

For basic GNN layers, the general "message-passing" architecture is employed for structural information aggregation:

$$X^{(l)} = E(A, X^{(l-1)}; \theta^{(l)}), \tag{1}$$

where $X^l \in \mathbb{R}^{N\times d}$ denotes the node embedding (*i.e.* "message") computed after l steps of the GNN, the input node embedding $X^{(0)}$ for the first step is initialized as the feature matrix F; E denotes the message propagation function, which takes the adjacency matrix, the trainable parameter $\theta^{(l)}$, and the node embedding $X^{(l-1)}$ generated from the previous step as input. Specifically, we implement the E using the combination of linear transformation and $\mathbf{ReLU}$ activation:

$$\begin{aligned} X^{(l)} &= E(A, X^{(l-1)}; \theta^{(l)}) \\ &= \mathbf{ReLU}(\tilde{B}^{-\frac{1}{2}}\tilde{A}\tilde{B}^{-\frac{1}{2}}X^{(l-1)}W^{(l)}), \end{aligned} \tag{2}$$

where $\tilde{A} = A + I$, $\tilde{B} = \sum_j \tilde{A}_{ij}$, and $W^l \in \mathbb{R}^{d\times d}$ denotes the trainable parameter matrix.

We then implement the DIFFPOOL layer [90] on the node embedding for nodes clustering. Specifically, $S^{(l)} \in \mathbb{R}^{n_l\times n_{l+1}}$ is defined as the assignment matrix for clustering the n_l nodes in layer l into the n_{l+1} groups in layer $l+1$. Each row of $S^{(l)}$ corresponds to one of the nodes or groups at layer l, and each column of $S^{(l)}$ corresponds to one of the n_{l+1} groups in layer $l+1$. Thus, the node embedding and the adjacency matrix in layer $l+1$ are computed as:

$$X^{(l+1)} = {S^{(l)}}^T X^{(l)} \in \mathbb{R}^{n_{l+1}\times d}, \tag{3}$$

$$A^{(l+1)} = {S^{(l)}}^T A^{(l)} S^{(l)} \in \mathbb{R}^{n_{l+1}\times n_{l+1}}. \tag{4}$$

The assignment matrix S^l is computed as:

$$\begin{aligned} S^{(l)} &= Sigmoid(\alpha * (S^{(l)}_{init} - \beta)), \\ S^{(l)}_{init} &= softmax(\mathbf{GNN}_{l,pool}(A^{(l)}, X^{(l)})), \end{aligned} \tag{5}$$

where the $\mathbf{GNN}_{l,pool}$ denotes the GNN layer for computing the assignment matrix from the node embedding, α is a hand-setting threshold and β is a learnable parameter. $S^{(l)}$ is normalized to be converged to $\{0,1\}$ for ensuring that each node is assigned to one of the groups.

In the last DIFFPOOL layer of the GNN module, the number of clustered groups, n_{l+1}, is set as 2 for clustering the nodes into two groups, the Essential and Minor ones. Let K denote the total number of implemeneted DIFFPOOL layers in the GNN module and let $L_D =$

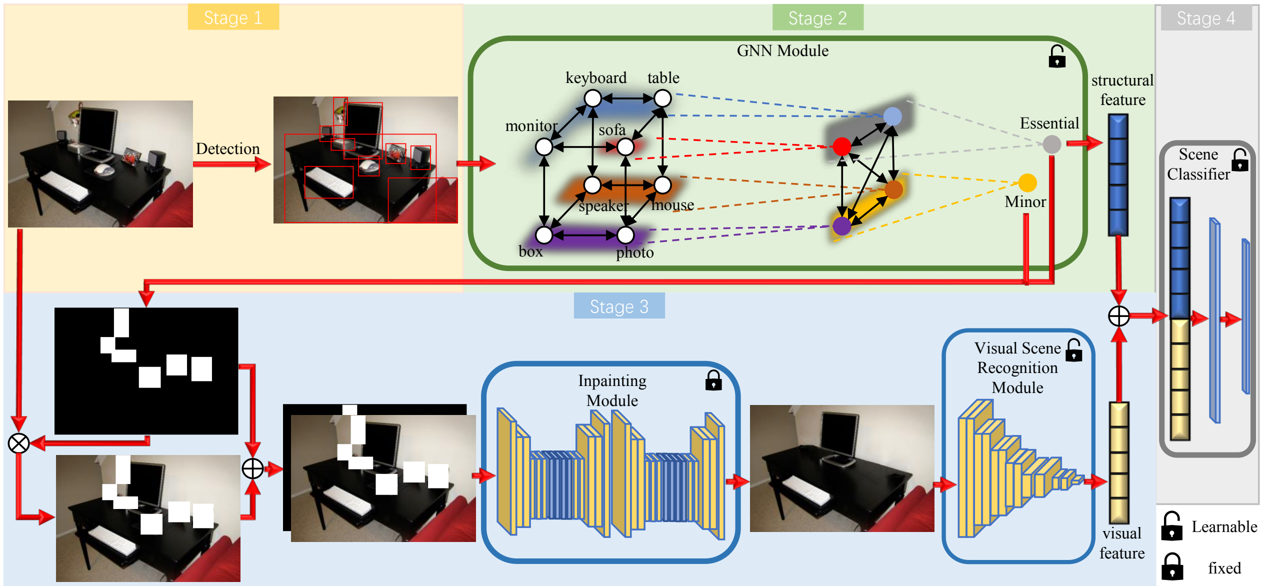


Figure 3: Illustration of the proposed approach. $\oplus$ denotes concatenation, and $\otimes$ denotes element-wise multiplication. Note that, the white regions in the mask denote the ones to be erased. In Stage 2, the structural feature is extracted only from the sub-graph formed by the Essential objects.

$[l_{D_1}, l_{D_2}, ..., l_{D_K}]$ denote the DIFFPOOL layer indexes, the structural feature of the scene graph is computed with the assignment matrix of the last DIFFPOOL layer:

$$E^{struct} = S^{(l_{D_K})}[:, 1]^T X^{l_{D_K}} \in \mathbb{R}^{1\times d}, \tag{6}$$

where the $S^{(l_{D_K})}[:, 1]$ denotes the assignment column for the Essential group and $S^{(l_{D_K})}[:, 2]$ denotes the column for the Minor one.

3.3. Stage 3: Visual Understanding

Given the clustering results from the previous stage, we erase the Minor objects and then extract the visual features from the derived image. This is achieved by our inpainting module and VSR module: the former takes care of the erasing process and the latter carries out feature extraction.

Specifically, we first compute the assignment score $S_{obj} \in \mathbb{R}^{N\times 1}$ for objects to be Essential:

$$S_{obj} = \prod_{i=1}^{K-1} S^{(l_{D_i})} * S^{(l_{D_K})}[:, 1]. \tag{7}$$

With the assignment score $s_i \in S_{obj}$ of the i-th object and its corresponding detected location $L_i = (x_{ul}, y_{ul}, x_{lr}, y_{lr})$ from Stage 1, the inpainting mask M is updated as:

$$M[x_{ul} : x_{lr}, y_{ul} : y_{lr}] = s_i, \tag{8}$$

where M is initialized as an all-ones matrix. The masked image is then derived as follows:

$$P_m = P \otimes M, \tag{9}$$

where the P denotes the input image and $\otimes$ denotes element-wise multiplication.

Next, we concatenate the inpainting mask M and the masked image P_m, and feed it into the inpainting module shown in Fig. 4. Here, we adopt the generative inpainting network proposed by Yu *et al.* [91]. The erased image is thus obtained:

$$P_I = (1 - M) \otimes GI(P_m, 1 - M) + M \otimes P, \tag{10}$$

where the P_I denotes the erased image and the GI denotes the generative inpainting network.

Finally, the erased image P_I is fed into the VSR module for visual feature extraction. Specifically, we adopt the VGG-16 [66] to achieve this task. The visual feature from the second last layer of VGG-16 is extracted:

$$E^{visual} = VSR(P_I). \tag{11}$$

3.4. Stage 4: Scene Classifier

Once the structural feature E^{struct} and the visual feature E^{visual} are collected, we stack them together and feed the concatenation into the scene classifier:

$$\tilde{Y} = N_{sc}(E^{struct}, E^{visual}; \theta_{sc}), \tag{12}$$

where the $\tilde{Y}$ denotes the predicted class of the erased scene image, N_{sc} denotes the scene classifier network, and θ_{sc} denotes the trainable parameters.

We then compute the cross entropy loss for the prediction:

$$\mathcal{L}_{CE} = \frac{1}{T}\sum_{i=1}^{T} \mathcal{H}(Y_i, \tilde{Y}_i), \tag{13}$$

where T denotes the number of input samples, $\mathcal{H}$ denotes cross entropy function, and Y_i denotes the ground-truth scene class.

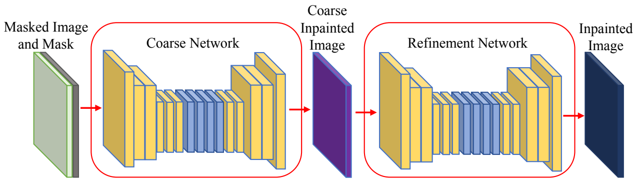


Figure 4: The architecture of the inpainting module. The yellow filters denote the standard convolutional operation and the blue ones denote the dilated convolutional operation.

Moreover, to encourage all Minor objects to be erased, we introduce a l_1-norm term to penalize the number of kept objects. We write,

$$\mathcal{L}_{norm} = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{N_i}||S_{obj}^i||_1, \tag{14}$$

where N_i denotes the number of detected objects in an image. The final objective function for the proposed approach is taken to be

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{norm}, \tag{15}$$

where λ denotes the balancing weight.

4. Implementation Details

We show here the details of training settings and module implementations.

Training Settings. Our networks are implemented using PyTorch and with 4 Tesla V-100 SXM2 GPUs. In the training process, the batch size is 192. The loss balancing weight λ is set to be 0.5, α is set to be 10^3, and the learning rate is manually reduced from 0.0001 to 0.00001.

GNN module. We implement two DIFFPOOL layers in the GNN module, each of which follows a three-layer neural network with residual connections. The feature dimensions of the neural network layers are set to be 512, 256, and 512 respectively. For the last neural network layer, we adopt ReLU as its activation function. The learning rate of the GNN module is set to be 0.003 for obtaining the best performance.

For the first DIFFPOOL layer in the GNN module, we set its number of clustered groups as 4. Therefore, the N detected objects are clustered into 4 groups, meaning that the number of nodes in the scene graph should be at least 4. For images which contains fewer than 4 detected objects, the self-connected virtual nodes are inserted so as to form a 4-node scene graph. Specifically, the virtual node is set to be with all zeros semantic and spatial information.

Inpainting Module. We adopt a popular generative inpainting network [91] as the inpainting module in our approach. Specifically, we pretrain it on MS-COCO dataset [46] and fix it in the training process. This module is implemented to produce visually realistic and reasonable inpainted contents. If we simply replace the areas of Minor objects with mean pixel value, the visual quality will be poor, especially for large area erasing, thus affecting the scene understanding and decreasing the scene recognition performance.

In the inpainting mask generation process, the areas of detected objects may overlap. If the overlapped objects belong to the same group, for example Essential objects, we average their assignment scores to be the mask value for the overlapped area. Otherwise, we average the assignment scores of the Minor objects to be the mask value.

VSR Module. We adopt the VGG-16 network [66] as our VSR module. Specifically, we pretrain it on the selected scene datasets. The pretrained VSR module is adopted to ensure that the Essential objects of the scene are kept. Since the VSR module takes the erased image as its input, it is expected to tell whether the input image belongs to the ground-truth scene category or not. If not, the Essential objects are incorrectly erased, through which supervision is back-propagated to update the network parameters.

Scene Classifier. We implement a three-layer neural network as the scene classifier. The feature dimensions of each neural network layer are set to be 1512, 1024, N_c, where N_c denotes the number of categories for each dataset. We adopt the ReLU as the activation function for the first two layers.

5. Experiments

In this section, we provide our experimental setups and show the results. Since we are not aware of any existing work that performs exactly the same task as we do here, we mainly focus on showing the promise of the proposed approach. We also compare part of our approach with other popular models. Our goal is, again, to show the possibility of learning Scene Essence, rather than trying to beat the state-of-the-art scene recognition, GNN, and inpainting models. Other modules with the same functionality, as long as end-to-end trainable, can be adopted in our approach to achieve potentially better performances.

5.1. Datasets

We adopt three datasets, Places365 [99], SUN397 [83], and MIT67 [60] to validate the proposed least scene subgraph learning approach.

Places365 Dataset [99]. It is one of the largest scene-centric datasets, which comprises two subsets, Places365-standard and Places365-challenge. In our experiments, the Places365-standard, which consists of around 1.8 million images from 365 scene classes, is used for training and validation. The validation set of Places365-standard, which comprises 100 images per class, is used for testing. Also, 10-fold validation is used during the training process.

SUN397 Dataset [83]. It is one of the most commonly used scene recognition datasets, which comprises around 109k images from 397 scene classes. In our experiments, we randomly chose 50 images as test ones, 20 as validation ones for each scene class. In total, we use around 81k images for training, 8k for validation, and 20k for testing.

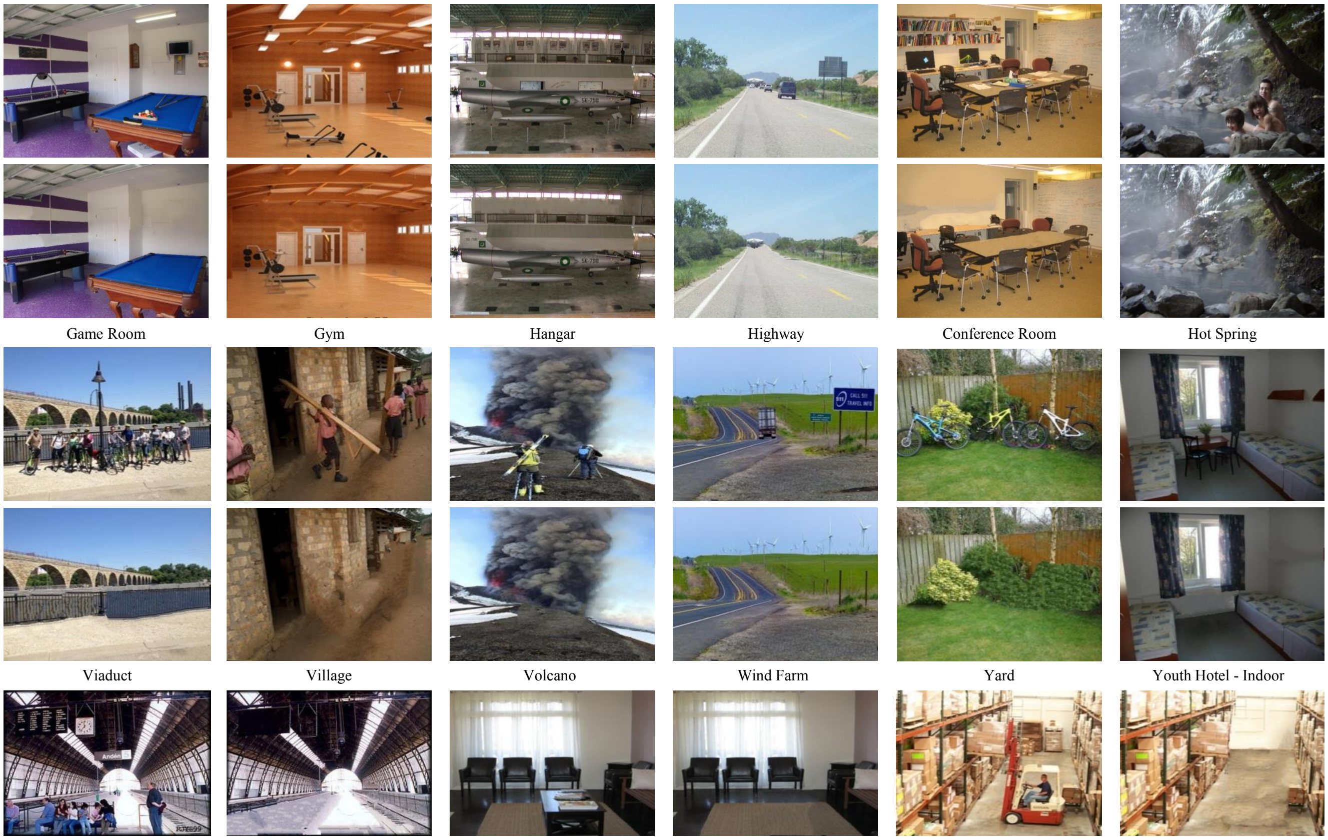


Figure 5: Scene Essence examples. The first two rows show the scenes in Places365 and their corresponding Scene Essence; the third and fourth row show the ones in SUN397; the last row shows the ones in MIT67. Within each pair, the upper/left one is the original image and the lower/right one is the corresponding Scene Essence.

Term	MIT67 Acc (%)	SUN397 Acc (%)
MFA-FS [16]	79.57	61.71
HSCFVC [47]	79.50	-
MFAFVNeT [43]	80.30	62.51
LSO-VLADNet [10]	81.70	61.60
Three [27]	80.90	66.23
S-HunA [64]	83.70	-
SpecNet [33]	84.30	67.60
CNN-DL [48]	82.86	67.90
LGN [11]	**85.37**	**69.48**
Ours	83.92	68.31

Table 1: Scene recognition accuracy of our approach and the state-of-the-art ones on the MIT67 and SUN397 datasets.

MIT67 Dataset [60]. It comprises around 16k images from 67 real-world indoor scenes. We adopt 20 images of each category for testing, 20 for validation, and the rest for training. In total, we use around 13.4k images for training, 1.3k for validation, and 1.3k for testing.

5.2. Scene Recognition

The scene recognition accuracy of our proposed approach and the state-of-the-art scene recognition methods on SUN397 and MIT67 datasets are shown in Tab. 1. As can be seen, although the aim of our approach targets at learning the Scene Essence, which keeps only few objects in the scene, it still achieves performance on par with the state of the art.

Term	Top-1 Acc (%)	Top-5 Acc (%)
CNN-SMN [68]	54.30	-
Places365-ResNet [99]	54.74	85.08
Places365-VGG [99]	55.24	84.91
Deeper BN-Inception [77]	56.00	86.00
LGN [11]	**56.50**	**86.24**
Ours	55.21	80.42

Table 2: Scene recognition accuracy of our approach and the state-of-the-art ones on the Places365-standard datasets.

We present the recognition accuracy on Places365 dataset in Tab. 2, where both the top-1 and top-5 accuracy are reported. The top-5 accuracy of our approach is considerably lower than those of other methods, as compared to difference on top-1 accuracy. This can be explained by that, the Scene Essence only keeps the Essential objects for its predicted category, thus reducing the inter-category affinities and further resulting in the lower top-5 accuracy.

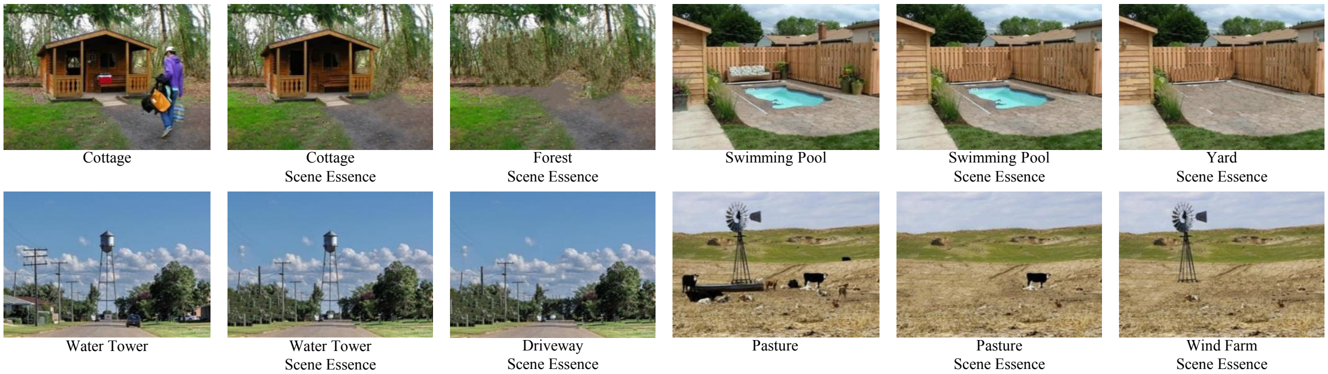


Figure 6: Scene transfer. The left image of each group shows the original scene, the middle one shows the Scene Essence of the ground-truth category, the right one shows the transferred Scene Essence of the second-top predicted category for the original scene.

λ	0.1	0.3	0.5	0.7	0.9
Acc (%)	55.25	55.22	55.21	9.57	6.09
Erasing Ratio	0.33	0.46	0.58	0.64	0.71

Table 3: Effect of λ on scene recognition accuracy and erasing ratio. Results are obtained on Places365.

Term	Places UE1	SUN UE1	MIT UE1	Places UE2	SUN UE2	MIT UE2
Score	98.23	98.74	97.59	99.52	99.39	99.15
Std	0.013	0.011	0.015	0.016	0.017	0.012

Table 4: Score and standard deviation of the first and second visual results validation user-study.

5.3. Erasing Ratio

We introduce a l_1-norm term as a part of the objective function, to penalize the number of objects left in the scene, and hence encourage all Minor objects to be wiped off. A balancing weight λ is used to trade-off the scene recognition accuracy and the ratio of erased objects. In this experiment, we show the effect of λ on the erasing ratio and its corresponding recognition accuracy. Specifically, we compute the erasing ratio of each image using $ER = \frac{N_m}{N}$, where N_m denotes the number of Minor objects and N denotes the total number of detected objects. As can be seen from Tab. 3, when λ is small (*i.e.* ≤ 0.5), the Minor objects are erased and the scene recognition accuracy stays stable. However, Essential objects tend to be erased with the increasing of λ and thus the recognition accuracy decreases dramatically.

5.4. Visual Results Validation

To validate the authenticity of the Scene Essence, we conduct two user-study experiments, where 112 users are involved to evaluate the quality of the erased images. In the first user-study experiment (UE1), we send each user 100 randomly selected image pairs, where one of them is the ground-truth image and the other is its corresponding Scene Essence, and ask the user whether or not these two images belongs to the same scene category. As can be seen from Tab. 4, the proposed method achieves 98.23% (same class) on Places365, 98.74% on SUN397, and 97.59% on MIT67.

In the second user-study experiment (UE2), we send each user 100 randomly selected Scene Essences with their ground-truth category label, and ask the user whether the Scene Essence belongs to the category or not. The proposed method achieves 99.52% same class on Places365, 99.39% on SUN397, and 99.15% on MIT67, as shown in Tab. 4.

It is interesting to note that the score is higher in the second experiment when compared with the first one. This can be explained that, in the first experiment, the attention of users is driven to seek the visual difference between the scene image and its corresponding Scene Essence. Such visual differences are explicitly taken into account and hence influence the final decision. In the second experiment, however, the attention of users focuses on the entire image, and is thus less affected.

The results of these two experiments show that our proposed approach indeed achieves promising and stable performances in terms of the visual quality.

5.5. Essence Validation

To validate that our Scene Essence maintains the minimum set of objects to preserve its predicted scene category, we conduct a subjective experiment as well as an objective one. In the subjective experiment (EV1), we involve 112 users, and then send each user 100 randomly selected *defective* Essence and their corresponding ground-truth category. The defective Essence is generated by randomly erasing one of the kept objects in our Scene Essence. Then we ask each user whether or not the further-erased image belongs to the ground-truth scene category. The results are shown in Tab. 5, in which we see the scores drop dramatically when compared to ones of our Scene Essence.

In the objective experiment, we train a ResNet-18 network [25] on Places365, and then use it as the classifier to predict the category of the original scene images, our Scene Essences, and the defective Essence. As can be seen from Tab. 6, our Scene Essence achieves recognition accuracy similar to the original images, while accuracy of defective

Term	Places365-EV1	SUN397-EV1	MIT67-EV1
Score	23.98	23.31	21.43
Std	0.122	0.127	0.114

Table 5: Score and standard deviation of the subjective essence validation experiment.

Term	Original Scene	Scene Essence	Defective Essence
Acc(%)	54.74	53.95	17.11

Table 6: Accuracy of a ResNet-18 classifier obtained using the original scene, Scene Essence, and the defective Essence on Places365.

Term	Ours	GSM [65]	GBP [73]	Original Scene
Acc (%)	53.95	32.65	34.04	54.74

Table 7: Scene recognition accuracy for Scene Essence obtained by our approach, GSM, and GBP on Places365.

Essence reduces significantly.

The results from these two experiments show that our Scene Essence truly maintains a minimum set of objects to preserve its predicted scene category; in other words, the kept objects in Scene Essence are indeed indispensable.

5.6. Visual Results Comparison

Since there is no existing method that aims to learn Scene Essence, we modified two state-of-the-art discriminative region detection methods, the gradient saliency map (GSM) [65] and the guided back-propagation (GBP) [73], and compare their results with ours. Specifically, as shown in Fig. 7, we assign the importance of each detected object based on its area-averaged importance score produced by the comparison methods. We then keep k objects with top importance scores as Essential objects, in which k denotes the number of Essential objects derived in our Scene Essence approach.

To compare our approach and the modified ones, we train a ResNet-18 network on Places365, and then use it as the classifier to predict the category of our Scene Essence and those from the modified methods. As can be seen from Tab. 7, our approach outperforms the other methods by a large margin, demonstrating that our explicit object-level reasoning yields a better performance in terms of interpreting recognition rationale.

5.7. Visual Results and Ablation Study

Examples of the derived Scene Essence are shown in Fig. 5, where the proposed method generates visually pleasing results. In Fig. 6, we showcase several scene transfer examples enabled by Scene Essence. Specifically, here we use the *second-top predicted category* of the original scene to be the training label in order to derive Scene Essence.

We conduct an ablation study to compare our GNN module with a GAT network [74] on the scene recognition accuracy, to demonstrate its capability to partition scene objects into Essential and Minor ones. As can be seen from Tab. 8, our GNN model outperforms GAT on all datasets. This can

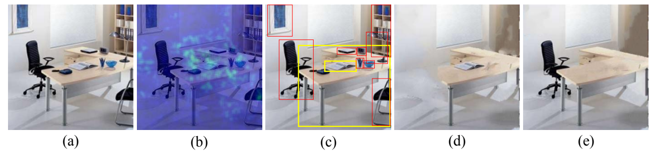


Figure 7: (a) shows the original office scene image, (b) shows the saliency map obtained from GSM, (c) shows the ranked object detections from the saliency map, (d) shows the Scene Essence obtained from GSM method, and (e) shows our Scene Essence.

Term	Places365 Acc(%)	SUN397 Acc(%)	MIT67 Acc(%)
Ours	55.21	68.31	83.92
GAT [74]	50.64	62.91	76.83

Table 8: Scene recognition accuracy of our GNN module and the GAT network on Places365, SUN397 and MIT67.

Term -full	Ours -full	GAT -full	Ours -without-S	GAT -without-S
Places365	55.21	50.64	43.84	40.66
SUN397	68.31	62.91	54.58	49.14
MIT67	83.92	76.83	60.37	54.77

Table 9: Results of our GNN and GAT under different setups. We compare the recognition accuracies of the two networks trained with full settings in our paper (Ours-full/GAT-full), and those of the two trainings without spatial information (Ours-without-S/GAT-without-S).

be in part explained by that, the GAT network lacks the subgraph level understanding (*i.e.* the hierarchical clustering).

We next conduct an experiment to study the impact of the spatial information encoded in our GNN module. If we remove the bounding-box coordinates from the node features, and hence reduce the feature dimension to 1024, the performances of both our GNN and GAT decrease significantly as shown in Fig. 9, indicating that the spatial coordinates of objects play a crucial role.

6. Conclusion

In this paper, we introduced Scene Essence, a novel scene signature that maintains a minimum set of objects with pivotal roles in scene recognition. We also proposed an innovative network to learn Scene Essence, in which a GNN is trained to partition the scene objects into Essential and Minor ones, and the latter are erased by an inpainter so as to fool the scene discriminator. Subjective and objective experiments demonstrate that, Scene Essence indeed captures key elements and hence is capable of interpreting scene recognition at object level, which has been largely overlooked by prior works. We also showcase that Scene Essence offers an inexpensive way to realize scene transfer.

Acknowledgement This research was supported by Australian Research Council Projects FL-170100117, DP-180103424, IH-180103424, IC-190100031.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A Alemi. Watch your step: Learning node embeddings via graph attention. In *Advances in Neural Information Processing Systems*, pages 9180–9190, 2018.

[2] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1993–2001, 2016.

[3] Davide Bacciu, Federico Errica, and Alessio Micheli. Contextual graph markov model: A deep and generative approach to graph processing. In *ICML*, 2018.

[4] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001.

[5] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.

[6] Daniel M Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li Fei-Fei, Jiajun Wu, Joshua B Tenenbaum, et al. Learning physical graph representations from visual scenes. *arXiv preprint arXiv:2006.12373*, 2020.

[7] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[8] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003.

[9] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[10] Boheng Chen, Jie Li, Gang Wei, and Biyun Ma. A novel localized and second order feature coding network for image recognition. *Pattern Recognition*, 76:339–348, 2018.

[11] Gongwei Chen, Xinhang Song, Haitao Zeng, and Shuqiang Jiang. Scene recognition with prototype-agnostic scene layout. *IEEE Transactions on Image Processing*, 29:5877–5888, 2020.

[12] Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017.

[13] Xiaojuan Cheng, Jiwen Lu, Jianjiang Feng, Bo Yuan, and Jie Zhou. Scene recognition with objectness. *Pattern Recognition*, 74:474–487, 2018.

[14] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II. IEEE, 2003.

[15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[16] Mandar D Dixit and Nuno Vasconcelos. Object based scene representations using fisher scores of local subspace projections. In *Advances in Neural Information Processing Systems*, pages 2811–2819, 2016.

[17] Claudio Gallicchio and Alessio Micheli. Graph echo state networks. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[18] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1416–1424. ACM, 2018.

[19] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[21] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[22] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[23] James Hays and Alexei A Efros. Scene completion using millions of photographs. *Communications of the ACM*, 51(10):87–94, 2008.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[27] Luis Herranz, Shuqiang Jiang, and Xiangyang Li. Scene recognition with cnns: objects, scales and dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 571–579, 2016.

[28] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in Neural Information Processing Systems*, pages 4558–4567, 2018.

[29] Yuanjun Huang, Xianbin Cao, Xiantong Zhen, and Jungong Han. Attentive temporal pyramid network for dynamic

scene classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8497–8504, 2019.

[30] Hamid Izadinia, Fereshteh Sadeghi, and Ali Farhadi. Incorporating scene context and object layout into appearance modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–239, 2014.

[31] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[32] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 923–930, 2013.

[33] Salman H Khan, Munawar Hayat, and Fatih Porikli. Scene categorization with spectral features. In *Proceedings of the IEEE international conference on computer vision*, pages 5638–5648, 2017.

[34] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[35] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[37] Long Lan, Xinchao Wang, Shiliang Zhang, Dacheng Tao, Wen Gao, and Thomas S. Huang. Interacting tracklets for multi-object tracking. *IEEE Transactions on Image Processing*, 27(9):4585–4597, 2018.

[38] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[39] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1666–1674. ACM, 2018.

[40] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[41] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *null*, page 305. IEEE, 2003.

[42] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] Yunsheng Li, Mandar Dixit, and Nuno Vasconcelos. Deep scene image classification with the mfafvnet. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5746–5754, 2017.

[44] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017.

[45] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems*, pages 1853–1863, 2018.

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[47] Lingqiao Liu, Peng Wang, Chunhua Shen, Lei Wang, Anton Van Den Hengel, Chao Wang, and Heng Tao Shen. Compositional model based fisher vector coding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2335–2348, 2017.

[48] Yang Liu, Qingchao Chen, Wei Chen, and Ian Wassell. Dictionary learning inspired deep network for scene recognition. 2018.

[49] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4424–4431, 2019.

[50] Andrii Maksai, Xinchao Wang, Francois Fleuret, and Pascal Fua. Non-markovian globally consistent multi-object tracking. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[51] Andrii Maksai, Xinchao Wang, and Pascal Fua. What players do with the ball: A physically constrained interaction modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[52] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[53] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[54] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[55] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *2011 International Conference on Computer Vision*, pages 1307–1314. IEEE, 2011.

[56] Sobhan Naderi Parizi, John G Oberlin, and Pedro F Felzenszwalb. Reconfigurable models for scene recognition. In

2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 2775–2782. IEEE, 2012.

[57] 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*, pages 2536–2544, 2016.

[58] Jiayan Qiu, Xinchao Wang, Pascal Fua, and Dacheng Tao. Matching seqlets: An unsupervised approach for locality preserving sequence matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[59] Jiayan Qiu, Yiding Yang, Xinchao Wang, and Dacheng Tao. Hallucinating visual instances in total absentia. In *European Conference on Computer Vision*, 2020.

[60] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.

[61] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2015.

[62] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[63] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[64] Ronan Sicre, Yannis Avrithis, Ewa Kijak, and Frédéric Jurie. Unsupervised part learning for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6271–6279, 2017.

[65] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[67] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, pages 73–86. Springer, 2012.

[68] Xinhang Song, Shuqiang Jiang, and Luis Herranz. Multi-scale multi-feature context modeling for scene recognition in the semantic manifold. *IEEE Transactions on Image Processing*, 26(6):2721–2735, 2017.

[69] Xinhang Song, Shuqiang Jiang, Bohan Wang, Chengpeng Chen, and Gongwei Chen. Image representations with spatial object-to-object relations for rgb-d scene recognition. *IEEE Transactions on Image Processing*, 29:525–537, 2019.

[70] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[71] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018.

[72] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[73] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[74] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[75] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[76] Anran Wang, Jianfei Cai, Jiwen Lu, and Tat-Jen Cham. Modality and component aware feature fusion for rgb-d scene classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5995–6004, 2016.

[77] Limin Wang, Sheng Guo, Weilin Huang, Yuanjun Xiong, and Yu Qiao. Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns. *IEEE Transactions on Image Processing*, 26(4):2055–2068, 2017.

[78] Xinchao Wang, Zhu Li, and Dacheng Tao. Subspaces indexing model on grassmann manifold for image search. *IEEE Transactions on Image Processing*, 20(9):2627–2635, 2011.

[79] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision and Pattern Recognition (ECCV)*, pages 17–32, 2014.

[80] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects using intertwined flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2312–2326, 2016.

[81] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Transactions on Image Processing*, 26(6):2868–2881, 2017.

[82] Ruobing Wu, Baoyuan Wang, Wenping Wang, and Yizhou Yu. Harvesting discriminative meta objects with deep cnn features for scene classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1287–1295, 2015.

[83] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[84] Guo-Sen Xie, Xu-Yao Zhang, Shuicheng Yan, and Cheng-Lin Liu. Hybrid cnn and dictionary-based models for scene recognition and domain adaptation. *IEEE Transactions on*

Circuits and Systems for Video Technology, 27(6):1263–1274, 2015.

[85] 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)*, pages 1–17, 2018.

[86] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6721–6729, 2017.

[87] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[88] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[89] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. SPAGAN: shortest path graph attention network. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[90] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pages 4800–4810, 2018.

[91] 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*, pages 5505–5514, 2018.

[92] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[93] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[94] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018.

[95] Ke Zhang, Na Liu, Xingfang Yuan, Xinyao Guo, Ce Gao, Zhenbing Zhao, and Zhanyu Ma. Fine-grained age estimation in the wild with attention lstm networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[96] Zhengyu Zhao and Martha Larson. From volcano to toyshop: Adaptive discriminative region discovery for scene recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1760–1768, 2018.

[97] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019.

[98] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217, 2017.

[99] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[100] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[101] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016.

[102] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983–2996, 2016.