Mental Replay: Learning Diverse Object Placement by Inpainting for Compositional Data Augmentation

Anonymous CVPR submission

Paper ID 1943

Abstract

We study the problem of common sense placement of visual objects in an image. This involves multiple aspects of visual recognition: the instance segmentation of the scene, 3D layout, and common knowledge of how objects are placed and where objects are moving in the 3D scene. This seemingly simple task is difficult for current learning based approaches because of the lack of labeled training data which ideally should consist of a variety of foreground objects paired with cleaned background scenes with no objects with many demonstrated plausible object locations. Because of this challenge, many current solutions only work in synthetic environments or rely on dense supervision. We propose a self-learning framework that automatically generates the necessary training data without any manual labeling by detecting, cutting, and inpainting objects from an image. We learn a generative model that predicts a distribution of common sense locations when given a foreground object and a background scene. We show experimentally our object placement network can be used to augment training data to boost instance segmentation. In addition, the learned representation of our placement network displays strong discriminative power in image retrieval and transfer learning. Inspired by human's memory system, we call our self-supervised learning system mental replay.

1. Introduction

Studies in humans and animals suggest that the mental replay of past experiences is essential for enhancing visual procession as well as making action decisions [3]. We ask the question: can developing a computational mental replay model help to improve AI visual perception tasks such as recognition and segmentation? More specifically, would the mental replay of object placement and scene affordance boost visual recognition systems?

This is not only a scientific question, but also a highly practical one for training a deep learning network. Most AI systems based on deep learning have a large appetite for a vast quantity of human-labeled training dataset. Several recent works demonstrated 'copy-paste' like data argumentation by inserting objects into a background image in order to boost object recognition performance [5, 7, 6, 9, 24]. If the mental replay of object placement could be carried out reliably, these methods point to a new way of data augmentation by utilizing self-supervised learning of object placement.

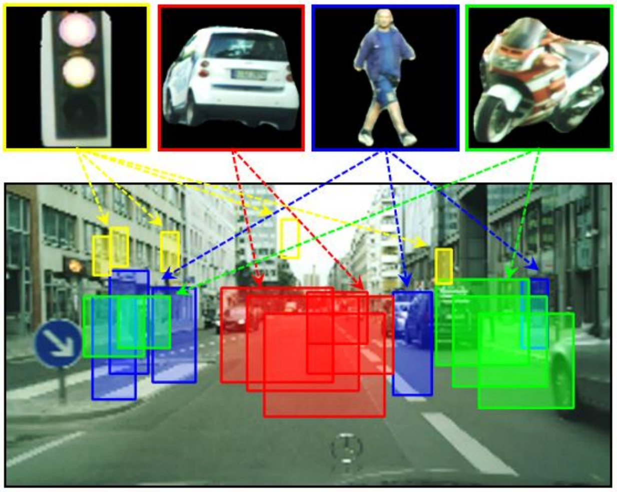

Figure 1: Given a foreground object and a background scene, our model is able to learn a set of reasonable and diverse locations for where to place the object into the scene.

Motivated by vast amount of driving scenes in public datasets, we create a self-supervised mental replay task of learning object placements into street scenes. Our system starts by observing many annotated street scenes along with psuedo ground truth annotated scenes. It learns to mental replay: transferring objects from one scene and composite them into plausible scenes at plausible new locations. This task has many useful side-effects: 1) it encourages the algorithm to discover functionality based object and scene features, and their contextual dependency; 2) it helps to create rare object-scene compositions to balance out biases in the training dataset.

The self-learning can also come for ‘free’ just by observing unlabeled scenes. Our insight is that we can generate ‘free’ labeled training data using an instance segmentation network [11] to cut out objects and fill in the holes using an image inpainting network [25]. The ‘free’ labeled object-background pairs tell us *what* the object looks like and *where* it is placed.

The ‘free’ labeled object-background pairs are then fed into our object placement network. The key challenge is to learn diverse yet precise object placements. There is a many-to-many mapping between the objects/scenes with plausible placement solutions. For example, one object-scene image pair can correspond to many different object placements (one-to-many). At the same time, similar object-scene pairs can correspond to the same object placement (many-to-one). The two key properties we want are 1) *diversity*: learns a many-to-many mapping, where images consisting of similar object-scene pairs can correspond to similar distributions of outputs; and 2) *modularity*: the objects and scenes are represented modularly to allow for maximal composition possibility for placing objects into scenes.

The diversity objective requires us to go beyond common approaches such as Variational Auto-Encoder (VAE) [12] and GAN [10] that suffer from mode collapse, which inhibits the many-to-many mapping we desire. We demonstrate our proposed network by building on the recent work of Normalized Diversification [17], and show its capability to generate rare compositions of objects and scenes.

In contrast with [14], we learn object placements using images of objects and backgrounds as input without compressing them to abstract category names. Using image appearances as input is much harder due to large feature dimensionality, but it allows us to create more contextually natural scenes compared to using GAN generated objects.

In contrast with [7], we can sample directly from the joint distribution of three disjoint variables: object appearance, scene appearance, and stochastic variations of object-scene interaction, without being forced into a compressed conditional distribution. This allows us to generate a far greater diversity of scene compositions.

We evaluate our object placement learning in terms of two applications. First, we use our learned object placement to augment datasets for instance segmentation. Our hypothesis is that by compositing scenes that model the distribution of any object, we are able to improve the performance on rare classes by placing them more naturally into scenes. Second, we test whether the encoding we learned for object placement can help with object recognition. Our hypothesis is that image features learned for the task of object placement are useful for retrieval and transfer learning.

2. Related Work

2.1. Learning Object Placements.

There have been several attempts to solve the task of object placement with deep learning. Lin et al [16] proposed Spatial Transformer Generative Adversarial Networks (ST-GAN) that iteratively warps a foreground instance into a background scene with a spatial transformer network via adversarial training against geometric and natural image manifolds. Azadi et al [1] proposed a self-consistent composition-by-decomposition network named Compositional GAN to composite a pair of objects. Their insight is that the composite images should not only look realistic in appearance but also be decomposable back into individual objects, which provides the self-consistent supervisory signal for training the composition network. Li et al [15] focused on predicting a distribution of locations and poses of humans in 3D indoor environments using Variational Auto-Encoders [12]. The work closest to ours is Lee et al [14], where they proposed a two-step model that predicts a distribution of possible locations where a specific class of objects could be placed and how the shape of the class of objects could look like using semantic maps.

Most of these works, while demonstrating the effectiveness of their approaches, fall short mainly due to their reliance on synthetic datasets [1] [15][16] or on semantic maps [14]. Thus, the existing approaches cannot generalize well to complex situations in the wild.

2.2. Data Augmentation for Object Recognition

There have been many efforts to improve performance of Object Recognition through data augmentation. One simple method to accomplish this is through geometric transformations of the images [11, 8, 18, 22] such as scale changes, horizontal flips, cropping, and rotations. By varying the levels of context around objects, the orientation, and the size of objects, their aim is to augment the data distribution that better matches the natural distribution of objects. Another method includes adjusting the signal-to-noise ratio to model the uncertainty in object boundaries and other possible sampling noises [8] by distorting the color information.

It has been demonstrated that context plays a key role in vision recognition systems [21, 23]. Having contextually related objects in a scene has more of an multiplicative effect than an additive one. That is, a scene composed of contextually sound objects is more than the sum of the constituent parts. Both [21, 23] validate that having contextually related objects provides more evidence for recognition than beyond just the local evidence of the object instance itself.

Instead of operating on the original data, one way to generate new images is to cut and paste object instances onto an image [5, 7, 6, 9, 24]. This has been shown to be effective

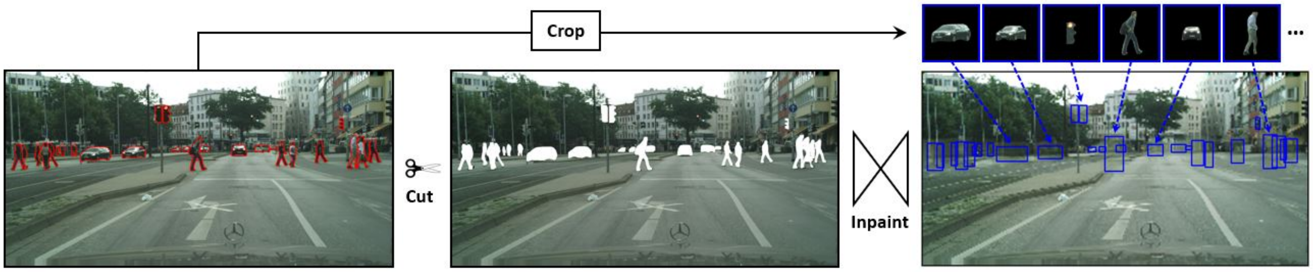

Figure 2: In our data acquisition pipeline, we first cut out the object region with the instance segmentation mask and obtain each individual object image with the corresponding ground truth, such as bounding box, at the same time. We then use an image inpainting network to fill the holes of the occluded region and generate the clean background.

for both object detection and instance segmentation.

The context-based cut and paste method most related to our work is [5], in that placement is learned based on context. But [5] does not condition the placement of the object on both the context and the appearance of the instance itself like ours. Instead the locations are classified on which class is most likely to be present in each location given the context. The method used is unable to distinguish if specific object instances of the same semantic class actually belong in the scene given the context.

Another work that is conditioned on both the appearance of the object and the context is [7]. But the only synthesis comes in the form of paired object-context scenes since an object can only be placed at different locations in the image based on a probability map.

3. Methods

Our work aims to learn to place foreground objects into background scenes without heavy human labeling. To do so, we first propose a novel data acquisition technique which leverages an image inpainting network. With generated training data, we propose a generative model named PlaceNet to predict a set of diverse but plausible locations and scales to place foreground objects into background scenes. With the learned object placement network, we further propose a data augmentation pipeline to composite different foregrounds into many different backgrounds for boosting instance segmentation.

3.1. Data Acquisition by Inpainting

What kind of data do we need in order to learn common sense object placements? Intuitively, our training set needs to contain paired examples of a foreground object, a cleaned background scene without objects, and labeled plausible locations to place the object. While such labeled data would be extremely difficult and expensive to obtain, we propose a novel data acquisition system. Our system leverages existing instance segmentation dataset and a self-supervised image inpainting network to generate the necessary training data for object placement learning.

Our insight is that we can generate such training data by removing objects from the background scenes. With an instance segmentation mask, we can first cut out the object regions and then fill in the holes with an image inpainting network. After that, we simultaneously obtain a clean background scene without objects in it, the corresponding ground truth locations, and scales for placing these objects into the scene. The overall process is described in Fig.(5). The instance segmentation could be obtained from labeled data or a pretrained Mask R-CNN network [11]. The inpainting network [25] is trained by randomly cropping out regions and letting the network fill the holes. After the training, the inpainting network learns a prior to fill the holes with background information even if the holes were previously occupied by some objects. Such GAN synthesis bias has been studied in [2]. Nevertheless, the instance segmentation network and the image inpainting network are used as building blocks in our data acquisition system, but the networks themselves are not the focus of this work.

3.2. Learning Object Placements

3.2.1 Overall Network Design

Most objects can have a multitude of possible placements in a given scene. For example, a pedestrian could stand on the left or right side of the street, walk across the street, or stand besides a car. To model such diverse and dense object placements is challenging, since object placements in real scenes are very sparse. Our insight is to share information across similar objects and similar backgrounds, where similar foregrounds and backgrounds would correspond to similar distribution of placements.

In order to share information across sparse samples, our insight is to encode foregrounds and backgrounds into a feature space, where foregrounds can be clustered by their semantic functionality and backgrounds can be clustered by layouts and affordance. We demonstrate the feature learn-

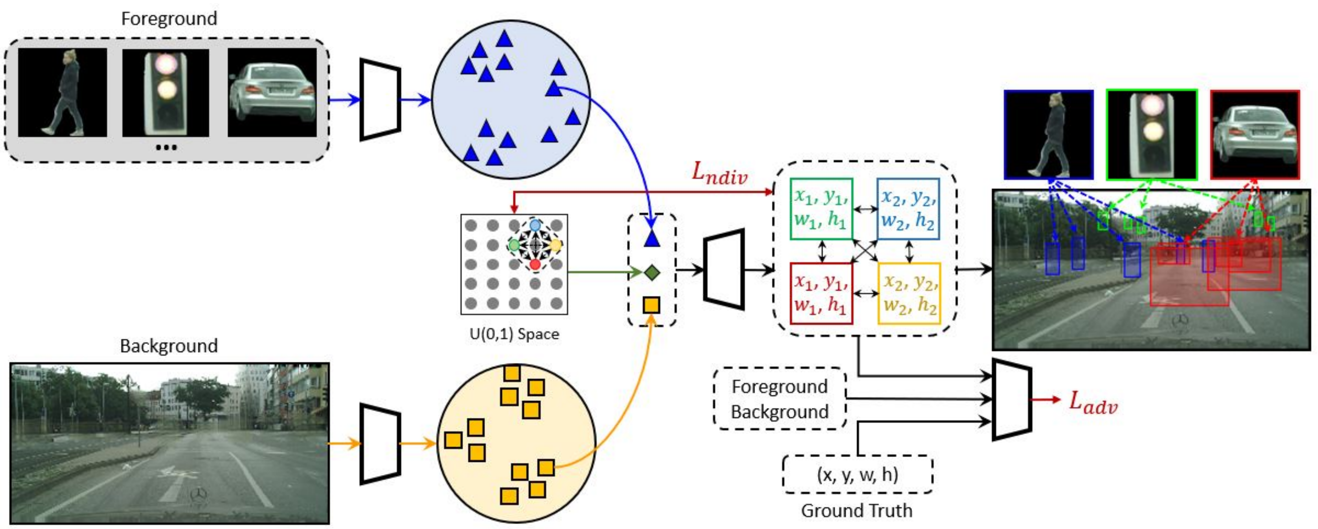

Figure 3: In our object placement network, we first extract features of foreground and background image, combined with a random variable sampled from a $U(1,0)$ uniform space, and decode to the predicted object location and scale. The predicted outputs are checked in terms of plausibility by a conditional discriminator and are diversified by preserving the pairwise distance between the sampled outputs and corresponding latent variables.

ing of our encoders in section 4.4. In our network, a foreground object and a background scene are first encoded into two feature vectors. The two feature vectors, combined with a sampled latent variable, are then decoded to the predicted object location and scale, which are parameterized as normalized center point, width, and height. In the encoding space, the foreground and background features provide necessary conditional information to determine the plausible placement regions, and the latent variable controls stochastic sampling of a specific predicted object placement. During training, we use a normalized diversification loss [17] to explicitly force the random variables to model the stochastic predictions and an adversarial loss [10] to check if the sampled object predictions are reasonable.

3.2.2 Model Diverse One-to-Many Distribution

Normalized Diversification (NDiv) [17] is proposed to relieve mode collapse in image synthesis. The main idea is to preserve the pairwise distance between the sampled output and the corresponding latent codes. Here, we borrow the idea of preserving the pairwise distance between outputs and latent variables in the context of our diverse conditional placement predictions.

In the latent encoding space, we sample a random variable from the $U(0,1)$ uniform space, combined with the foreground and background codes, and decode to the predicted object placements. One sampled random variable would correspond to a unique plausible object placement. This is realized by preserving the pairwise distance between sampled object placements and latent variables. The loss function is shown as follows,

$$\mathcal{L}_{ndiv}(y,z) = \frac{1}{N^2 - N} \sum_{i=1}^{N} \sum_{i \neq j}^{N} max(0, \alpha D_{ij}^{z} - D_{ij}^{y}) \quad (1)$$

$$D_{ij}^{z} = \frac{d_z(z_i, z_j)}{\sum_j d_z(z_i, z_j)} \quad , \quad D_{ij}^{y} = \frac{d_y(y_i, y_j)}{\sum_j d_y(y_i, y_j)} \quad (2)$$

where N is the number of sampled latent variables, z denotes the latent variable, y denotes the predicted placement location and scale, and i, j indicate the sample indices, D_{ij}^{z}, $D_{ij}^{y} \in \mathbb{R}^{N \times N}$ are normalized pairwise distance matrices, α is a relaxation hyperparameter in the hinge loss. Every pair of unnormalized distance is simply measured using Euclidean distances define as follows,

$$d_z(z_i, z_j) = ||z_i - z_j|| \quad , \quad d_y(y_i, y_j) = ||y_i - y_j|| \quad (3)$$

In our implementation, we sample $N = 4$ latent variables at each iteration, and aim to preserve the pairwise distance between the four sampled object placements with respect to the four latent variables in the uniform latent space. With this optimization purpose, we explicitly encourage random variables to actively model stochasticity in the placement predictions.

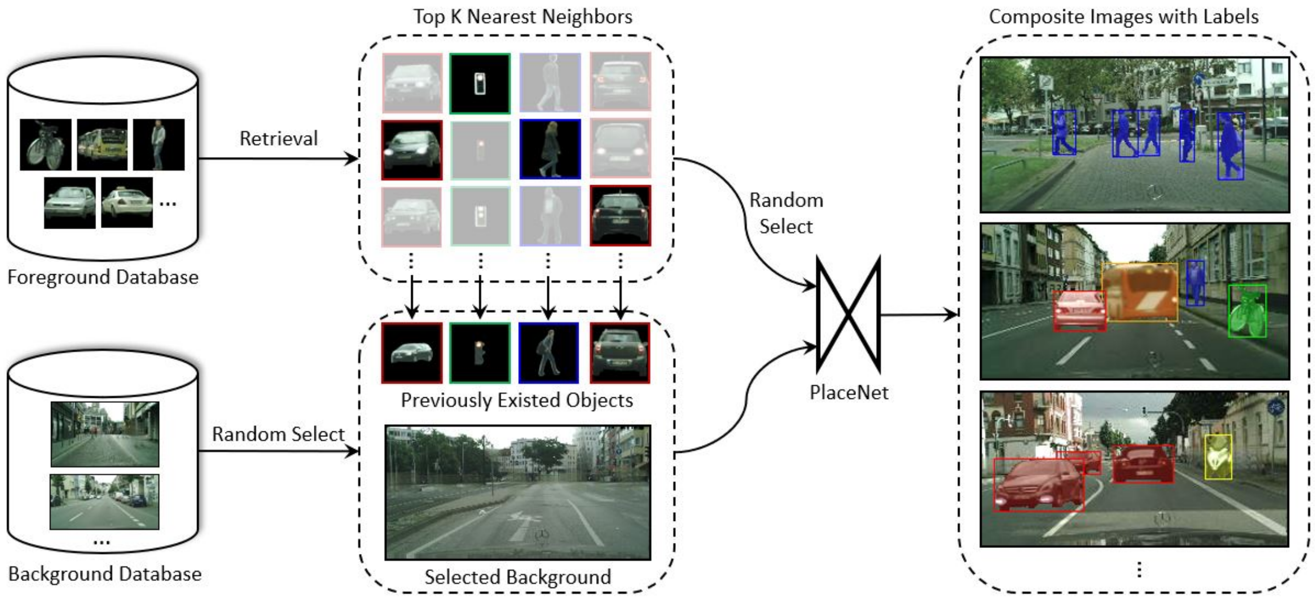

Figure 4: In our data augmentation pipeline, the foreground database contains masked foreground objects and the background database contains ”cleaned” backgrounds with no objects. To make sure the selected foregrounds semantically make sense to be placed into a background, we retrieve top K nearest neighbors of foregrounds with respect to the objects that were previously in the background scene. Then, we randomly select several foregrounds in the retrieved subset of foregrounds and copy-and-paste them into a selected background with predicted locations and scales from our PlaceNet.

3.2.3 Adversarial Check on Predictions

While normalized diversification encourages the network to sample diverse placements, we apply a conditional adversarial loss [19] to check whether the predicted placements are plausible. Our discriminator takes the predicted placement as input and tries to distinguish it from the ground truth placement conditioned on the foregrounds and background. The adversarial loss is defined as follows,

$$\begin{aligned}\mathcal{L}_{adv} =& E_{x\sim p_{\text{data}}(x)}\left[log(D(y|f,b))\right] \\ &+ E_{z\sim p(z)}\left[log(1-D(G(z|f,b)|f,b))\right]\end{aligned} \tag{4}$$

where f is the foreground, b is the background, y is real placement, z is the random variable, D and G are discriminator and generator respectively. In order to stabilize training, we use spectral normalization [20] to scale down the weight matrices in the discriminator by their largest singular values, which effectively restricts the Lipschitz constant of the network.

3.3. Data Augmentation

We randomly select a background to start placing objects, but the starting background could be completely empty and filled in with inpainting or only a few objects removed. This allows us to combine the natural distribution of the object placements with our own generated ones. This essentially can generate more contextually natural and more varied scenes around objects.

After selecting the background, we then choose objects that are semantically similar to the ones previously removed from the scene. This is done because there can be multiple reasons why two instances of the same class might not belong in the same scenes. The most obvious reason why an object might not belong is that some instances are occluded. For example there are many ”floating heads” in Cityscapes [4] because cars are in front of the person. This is done by selecting top K nearest neighbors from the foreground database for each of the previously existing objects in the scene. We use our pretrained encoder to extract features of foregrounds and use cosine similarity as a distance metric to find top K neighbors. The overall pipeline is shown in fig (4). Basically, the K nearest neighbors search finds a plausible subset of foregrounds to add into a specific background.

From there, we randomly select an object from a retrieved foreground subset and feed them into PlaceNet together with a background image one at a time. From the predicted locations and scales, we simply cut and paste to synthesize the new image. This method of synthesizing has been demonstrated by previous works [24, 5, 7] to not be detrimental for detection or instance segmentation despite the visual flaws at the borders.

Due to the diversity property of the placement network, we are better able to model the probability distribution map

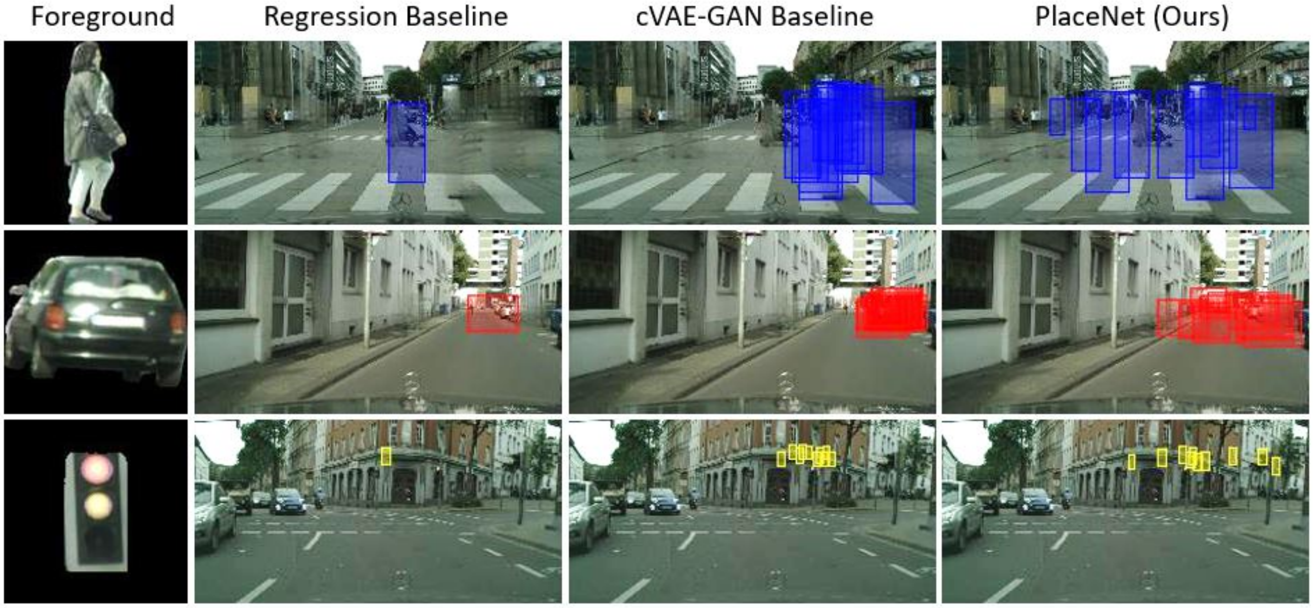

Figure 5: This is a qualitative comparison of object placement predictions between two baseline models and our PlaceNet. The first column is the input foreground and the right three columns are the predicted placements in a specific background. As seen, the regression model could only produce a single solution and cVAE-GAN model can sample a set of solutions but collapse at certain regions compared to our results.

of objects in a scene. The modularity of our design allows us to generate any pair of object-context images. These two properties combine to generate novel scenes with contextually related objects that appear sufficiently different yet natural. Another effect of this is that we can decorrelate instances from a specific scene and location using diversity and modularity since objects can naturally move around.

4. Experiments

In the experiment section, we first evaluate the plausibility and diversity of our placement predictions compared with strong baseline models. We then evaluate how much our data augmentation technique could boost instance segmentation performance. Finally, we also show that our foreground and background encoders learn meaningful representations in terms of image retrieval and classification.

Models	Plausibility	Diversity
Regression Baseline	73.1% / 37.3%	0
cVAE-GAN Baseline [13]	75.9% / 38.6%	0.0289
PlaceNet (Ours)	**76.4% / 39.5%**	**0.0392**

Table 1: Evaluations on object placements. In the first column, the left numbers indicate the percentage of plausible results from the user study, and the right numbers indicate the percentage of predicted placements is indistinguishable.

4.1. Object Placements

4.1.1 Baseline Models

We compare our placement network with two baseline models, which are a deterministic model and a variational generative model.

Regression Model is a baseline model that directly minimize the mean square error between the predicted object locations and scales. It is a determinisitic model meaning that it can only predict one solution for one pair of foreground and background inputs.

cVAE-GAN is a combination of conditional Variational Auto-Encoder (cVAE) and Generative Adversarial Newtork (GAN). This model encodes conditional inputs into a Gaussian latent space, where it can sample a distribution of solutions for a given conditional input.

4.1.2 Plausibility and Diversity

We evaluate the predictions of object placement in terms of plausibility and diversity. For plausibility evaluation, we conduct two types of human evaluations. First, we ask evaluators to decide whether the predicted location is reasonable given the object and the scene, and then compute the percentage of user selected examples. The higher the percentage the better the plausibility. Second, we give the user a ground truth placement image and a network predicted image at the same, and let the user select which one is more

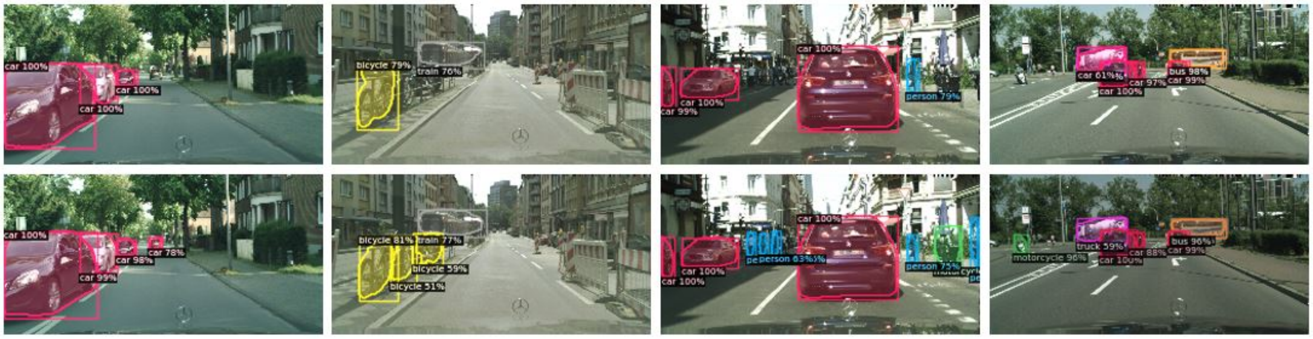

Figure 6: The top row comes from the baseline, and the bottom row comes from our best model (Ours + InstaBoost). In the first image (from left to right), there is small car clearly ahead of the ego vehicle yet the baseline fails to capture it. In the second picture we detect all the bikes on the bike rack. For the third picture, the baseline has a difficult time in crowded scenes, and it misses multiple people. The motorcycle in the last image is completely missed probably due to its low class appearance. We can detect more highly occluded and small instances where context is more important for identifying them.

Methods	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mAP	mAP_{50}
Baseline	0.202	0.069	0.620	**0.495**	0.493	0.156	0.133	0.118	0.286	0.634
Random Placement	0.210	0.097	0.619	0.449	0.471	0.143	0.128	0.118	0.279	0.610
InstaBoost [7]	0.202	**0.164**	0.627	0.465	0.479	0.196	**0.143**	**0.121**	0.300	0.684
Ours	0.198	0.080	0.621	0.487	**0.512**	0.264	**0.143**	0.109	0.302	0.664
Ours + InstaBoost [7]	**0.212**	0.150	**0.630**	0.448	0.506	**0.324**	0.139	0.119	**0.316**	**0.695**

Table 2: Instance segmentation results using Mask R-CNN (Res-50-FPN) under different types of data augmentation.

realistic. The best score for this experiment would be 50%, which indicates the predicted results are indistinguishable from labeled results. Overall, we evaluate 200 testing examples across ten users and average the results. The results indicate that our model can produce similar or better plausibility compared to the baseline approaches.

To evaluate how diverse the predicted object placements are, we compute the pairwise absolute distance between sampled outputs for each conditional input and average the results across 1,000 testing examples. The predicted outputs are normalized horizontal and vertical location, width and height in range between 0 and 1. The higher the pairwise absolute distance indicate more diversity in the predicted outputs. The results indicate that our method can outperform cVAE-GAN baseline by an obvious margin.

4.2. Data Augmentation

Even though our placement model could be trained on either labeled instance masks or coarse masks generated from Mask R-CNN, we use only the Cityscape labeled masks to train our placement network for data augmentation. This ensures that we use the exact same amount of labeled data as the non-augmented baseline for fair comparison.

One thing we noticed is the class imbalance in the Cityscapes dataset [4] in that there are very few trains, rider, bicycles, and buses. In order to improve the performance of rare and hard to detect classes, we biased some of augmented data to sample inversely proportional to the frequency of the class. And from Table (2) we can see that biasing the synthesized images towards rare classes does improve the results of some of the classes. We can see the greatest improvement in trains which was the rarest class, accounting for 0.67% of all annotated instances.

4.2.1 Comparison with baselines and state-of-the-art

Baseline and Random Placement. To test our data augmentation, we started with the baseline gtFine annotated of 3475 images. The baseline model is trained with the labeled images only. Then we generated data in which the instance object and the location of that instance were randomly selected. We can see that the random placement of objects hurts the performance which is consistent with [7, 5].

Comparison to the State-of-the-Art. Then as a comparison to a state-of-the-art data augmentation technique that has demonstrated a performance boost, we utilized [7] in the Cityscape dataset [4]. As expected, both InstaBoost and our method can boost the performance by obvious and similar margins compared to the baseline.

Our last test combined both [7] and our method to check if the augmentation we perform forces the network to learn different attributes. We can see a larger overall performance increase from combining both methods. This combination allows for higher detection rate of small and occluded objects. Seeing as how the combination performs much better

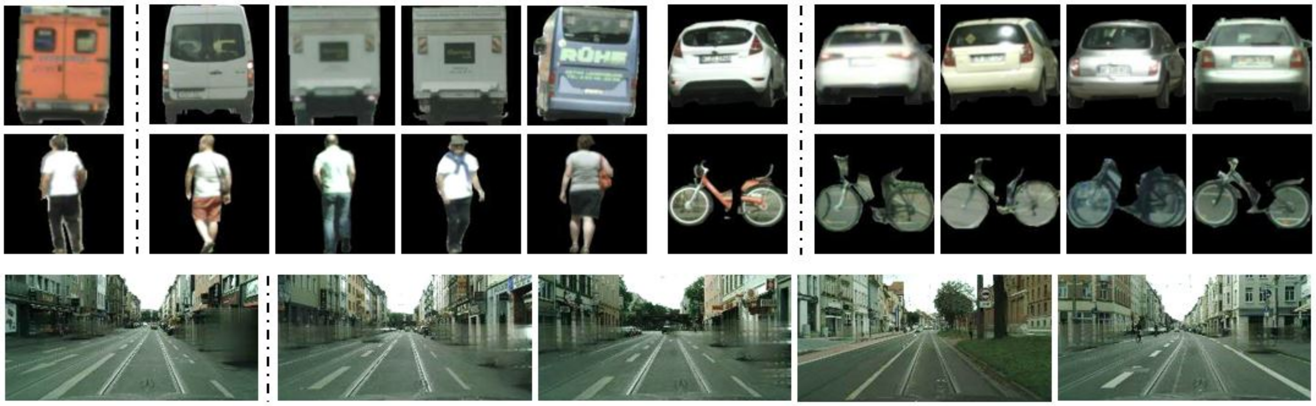

Figure 7: We show that our learned foreground encoder can retrieve images with similar semantic categories and poses, and our learned background encoder can retrieve images with similar layouts. The figures on the left side of dashed line are query images and the figures on the right side of dashed line are top retrieved images. We can see that for the background the buildings all vary while the road layout and sidewalks remain similar. Through the object placement task, our system learns to cluster scene affordance.

than each one alone, we can be confident that the distribution of information in our synthetic data is different enough from [7].

4.2.2 Implementation Details

For network training on the Cityscapes dataset [4], we used facebook's detectron2. For the data preparation, we scale images to be 128x256, which is the same size as the inputs/outputs of the placement network. All experiments ran for 200k iterations with an evaluation every 1000 iteration. Two classes, trailer and caravan, are removed due to their extreme rarity. The anchors used were scaled down 8x from the default anchors to match the down sampled images. The learning rate used is 0.0025 with a learning rate step at every 50k iterations. The batch size used is 8.

4.3. Feature learning

A key property of our network is to share information across sparse observations of samples such that we can learn a dense distribution of diverse placements between a foreground and a background. This is achieved in that our foreground and background are able to learn a feature representation such that the encoding space can cluster foregrounds and backgrounds based on their semantics and functionality. We demonstrate that our encoders learn such features through image retrieval for both foregrounds and backgrounds. As shown in Fig. (7), the foregrounds can be clustered with consistent semantics and poses and the street scenes can be clustered with same semantic layout regardless of building appearances.

In addition, we show that our pretrained features can be used to classify foreground objects by fine-tuning on few labeled training data. As seen in table. (3), the classifier with our pretrained features consistently outperforms the classifier trained from scratch with different training settings.

Training Data	From Scratch	Learned Encoder
1000	34.3	**46.5**
5000	52.5	**74.8**
10000	67.7	**86.3**

Table 3: This table shows that the classification performance between a classifier trained from scratch and a classifier fine-tuned on our pretrained foreground encoder. The first column indicates total number of training data for eight classes of foregrounds.

5. Conclusion

We formulated the task of learning object placement as a problem of mental replay that can be decomposed into two separate properties: diversity and modularity. The diversity aspect properly models the distribution of plausible object locations in a single scene, and the modularity models the distribution of plausible scenes that an object can exist in. In keeping with these properties we have demonstrated the ability to synthesize contextually natural novel data that can boost vision recognition systems. In addition to the generative ability, the encoding of both the object and the background display strong discriminative power in that they are clustered according to their semantics and functionality. In the future we want to extend our system into more complex scenes and temporal contextual relationships.

References

[1] Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, and Trevor Darrell. Compositional gan: Learning image-conditional binary composition. *arXiv preprint arXiv:1807.07560*, 2019. 2

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4502–4511, 2019. 3

[3] Margaret F Carr, Shantanu P Jadhav, and Loren M Frank. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature neuroscience*, 14(2):147, 2011. 1

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 7, 8

[5] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018. 1, 2, 3, 5, 7

[6] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, 2017. 1, 2

[7] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. *arXiv preprint arXiv:1908.07801*, 2019. 1, 2, 3, 5, 7, 8

[8] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 2

[9] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*, 2017. 1, 2

[10] 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. 2, 4

[11] 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. 2, 3

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[13] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 6

[14] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *Advances in Neural Information Processing Systems*, pages 10393–10403, 2018. 2

[15] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12368–12376, 2019. 2

[16] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018. 2

[17] Shaohui Liu, Xiao Zhang, Jianqiao Wangni, and Jianbo Shi. Normalized diversification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10306–10315, 2019. 2, 4

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 5

[20] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[21] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 2

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[23] Antonio Torralba, Kevin P Murphy, William T Freeman, and Mark A Rubin. Context-based vision system for place and object recognition. 2003. 2

[24] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching, 2019. 1, 2, 5

[25] 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. 2, 3