The Missing Data Encoder: Cross-Channel Image Completion with Hide-And-Seek Adversarial Network

Arnaud Dapogny[1], Matthieu Cord[1, 2], and Patrick Perez[2]

[1]LIP6, Sorbonne Université, 4 Place Jussieu, Paris, France
[2]Valeo.ai, Paris, France

Abstract

Image completion is the problem of generating whole images from fragments only. It encompasses inpainting (generating a patch given its surrounding), reverse inpainting/extrapolation (generating the periphery given the central patch) as well as colorization (generating one or several channels given other ones). In this paper, we employ a deep network to perform image completion, with adversarial training as well as perceptual and completion losses, and call it the "missing data encoder" (MDE). We consider several configurations based on how the seed fragments are chosen. We show that training MDE for "random extrapolation and colorization" (MDE-REC), i.e. using random channel-independent fragments, allows a better capture of the image semantics and geometry. MDE training makes use of a novel "hide-and-seek" adversarial loss, where the discriminator seeks the original non-masked regions, while the generator tries to hide them. We validate our models both qualitatively and quantitatively on several datasets, showing their interest for image completion, unsupervised representation learning as well as face occlusion handling.

1 Introduction

In this paper, we investigate the problem of image completion, *i.e.* the one of generating a complete image from RGB or single-channel parts of an original image. From a representation learning standpoint, learning to perform image completion amounts to encoding the underlying structures of the visual objects. A number of approaches have been proposed in the literature that try to learn this structure in an unsupervised fashion, in the hope that the representations learned by doing so could help other (mostly supervised) tasks, such as image classification, object detection or semantic segmentation. Indeed, for a number of these tasks, performing a supervised pre-training on a large database such as ImageNet is beneficial to the accuracy. Yet, collecting such vast amounts of data is tedious, if not impractical.

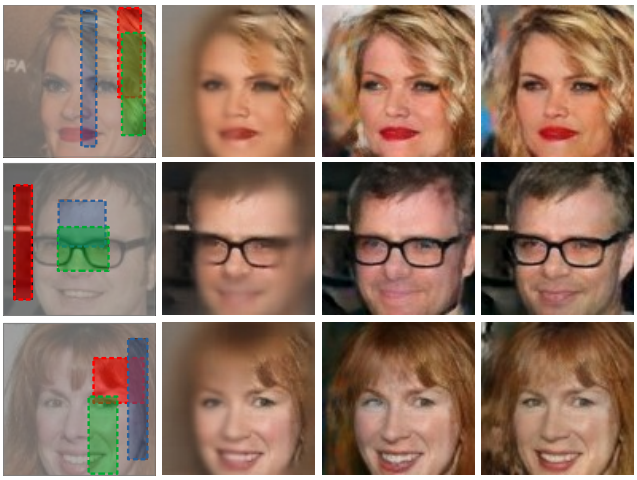

Figure 1: **Image completion from a small fragment in each color channel**. From left to right: Original image masked channel-wise; Images generated with proposed missing data encoder, trained respectively with *completion*, *perceptual+adversarial* and *perceptual+adversarial+hide-and-seek* losses (see text for details). In all cases, the image is completed using only the information within the boxes. The hide-and-seek loss ensures that there is no trace left of the generation process in the completed images.

Following recent advances in the field of text understanding [3], one can wonder if using the regularity of the images in an unsupervised fashion would yield such representations at virtually no cost. This echoes the ideas introduced in [10], where it is theorized that a strong artificial intelligence model should build an inner representation through unsupervised learning. A general idea for doing so is to design a proxy task for pretraining. The authors in [1] proposed to predict the relative position of two adjacent image patches from their content. In the same vein, the authors in [12] trained a network to solve jigsaw puzzles, created by shuffling a grid of patches. Intuitively, the network has to learn the structure of the objects to correctly predict the patches spatial layout and

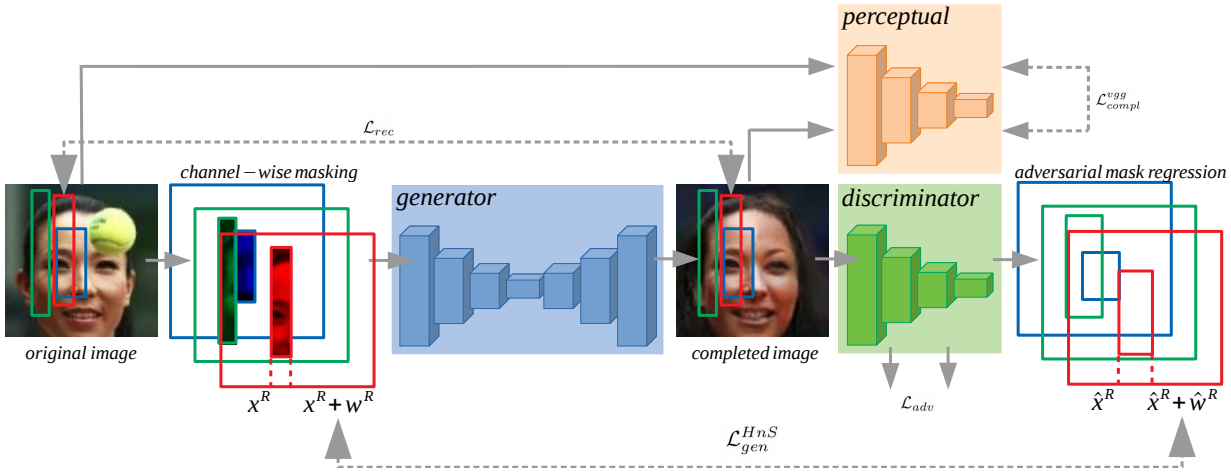

Figure 2: **Overview of the Missing Data Encoder approach**. At train time, random channel-wise masking is applied to the image, which then passes through the generator that completes it. To this end, MDE training uses perceptual, reconstruction and adversarial losses. The latter includes a novel mask regression term requiring the discriminator to "seek" the mask coordinates while the generator tries to"hide" them.

to solve the puzzle. Image colorization [6] has also been used as a proxy task: in [19], the authors introduce the split-brain autoencoder, where each encoder aims at reconstructing a specific channel (*e.g.* a color or a depth channel) given another one. Achieving such a completion task requires an even better capture of the visual structures by the trained network, as compared to predicting only loose spatial layout as in [1, 12]. Yet, colorization is a restricted form of completion where only low frequency chrominance information needs to be inferred. Other proxy tasks have been proposed, such as learning motion-based segmentation in videos [14]. Some approaches involve completing images given only a fraction of the original image. To this end, the authors of [13] use recurrent networks to encode the spatial dependency of pixels for image completion and generation. However, the learned representations cannot be easily transferred to other tasks, as most models now involve convolutional networks. A particular case of completion is inpainting, where a central patch is reconstructed given its context, as in [15]. Similarly, Li *et al.* [7] propose a generative face completion method. These approaches generally rely on adversarial training [2], where a discriminator network aims at distinguishing the fake data, provided by the generator network, from the true data.

Lastly, the problem of completion is related with the work in [10], where the authors generate a new frame given the past frames in a video. While the setups are different, we can draw a parallel between the temporal dependency between two events, and the spatial dependency between objects in an image. For instance, a man's trajectory is predictable in the short-term as it usually varies smoothly and in relation with a context. Similarly, if we see a mug in an image we are likely to also observe a desk, or a hand.

In this paper, we propose a framework for image completion using a deep neural network that we call the missing data encoder (MDE). We study several image completion scenarios with MDE: inpainting, reverse inpainting, colorization and the more general task of completing from random fragments in the different color channels – we call it the "random extrapolation and colorization" (REC). The latter proves to be the best at capturing the visual semantics for subsequent use. MDE uses skip-connections to ensure that the input image regions are not altered, and is trained with a combination of completion losses, adversarial discriminative loss, perceptual loss and a novel adversarial hide-and-seek loss, as shown on Figure 1. We demonstrate on multiple datasets that we can extrapolate high quality images from only small seed fragments, and that MDE-REC encodes semantic information as well as object geometry. The contributions of this paper are three-fold:

- We introduce MDE, a framework for image completion that uses a u-net-like architecture, adversarial training as well as perceptual and completion losses. We study several configurations and show that the best performing model, MDE-REC, uses a channel-wise random masking which encompasses inpainting, reverse inpainting and colorization as special cases.

- We introduce a novel adversarial hide-and-seek loss that complements the standard adversarial objective function for image completion tasks, by specifically ensuring that there is no trace left of the generation process in the completed images.

- We thoroughly validate our model on multiple datasets, showing that MDE-REC encodes image geometry and semantics. We show several applications of MDE-REC including image generation, representation learning, and face completion under targetted occlusions.

2 The missing data encoder

Figure 2 provides an overview of MDE-REC. As it was done in [15] for inpainting and in [10] for video frame prediction, we use GANs as our base architectural brick.

Given an RGB image Z of size $W \times H \times 3$, we mask it by element-wise multiplication with a random binary mask M of same size. As we will see in what follows, this mask can be generated in different ways. The generator G with parameters θ_g maps the masked image $M \odot Z$ to a *complete* image $G_{\theta_g}(M \odot Z)$. This new image can be decomposed as a *reconstructed* region, $M \odot G_{\theta_g}(M \odot Z)$ that should closely resemble the original fragment $M \odot Z$, and a *completed* one, $(1 - M) \odot G_{\theta_g}(M \odot Z)$. The discriminator D with parameters θ_d has to distinguish the generated images from the real ones. Given an image training set $\{Z_i\}_{1=1}^N$ and associated masks $\{M_i\}_{1=1}^N$, this is obtained by minimizing:

$$\mathcal{L}_{Disc}(\theta_d) = -\frac{1}{N}\sum_{i=1}^{N} \log D_{\theta_d}(Z_i) + \log[1 - D_{\theta_d}(G_{\theta_g}(M_i \odot Z_i))]. \tag{1}$$

The generator has to fool the discriminator by minimizing:

$$\mathcal{L}_{Gen}(\theta_g) = -\frac{1}{N}\sum_{i=1}^{N} \log D_{\theta_d}(G_{\theta_g}(M_i \odot Z_i)). \tag{2}$$

In practice, optimizing solely $\mathcal{L}_{adv}(\theta_g, \theta_d) = \mathcal{L}_{Gen}(\theta_g) + \mathcal{L}_{Disc}(\theta_d)$ at train time leads to unstable behaviors. To avoid this, a classic approach [15, 7] consists in adding an L_2 completion loss between the completed and the original regions:

$$\mathcal{L}_{compl}(\theta_g) = \frac{1}{N}\sum_{i=1}^{N} \left\| (1 - M_i) \odot (G_{\theta_g}(M_i \odot Z_i) - Z_i) \right\|_2^2. \tag{3}$$

However, optimizing $\mathcal{L}_{compl}(\theta_g) + \lambda_{adv}\mathcal{L}_{adv}(\theta_g, \theta_d)$ leads to bad results, as the discriminator network quickly wins against the generator, which generates unrealistic images. Also, nothing prevents the generated image to differ from the original one on the non-masked regions.

2.1 Preserving input information

The authors of [15] use an overlap between the inpainted region and the context, and apply a strong penalty for bad reconstructions of this region to "guide" training. In this vein, we add a reconstruction loss on the non-masked regions:

$$\mathcal{L}_{rec}(\theta_g) = \frac{1}{N}\sum_{i=1}^{N} \|M_i \odot (G_{\theta_g}(M_i \odot Z_i) - Z_i)\|_2^2. \tag{4}$$

Note that such a task merely consists in autoencoding the original element: it is way easier than the task of completion and thus effectively serves as a guide for the latter task. Note that it is crucial to reconstruct the original element with high fidelity. In practice, we observe that, even if we apply a large cost to bad reconstruction of the non-masked regions, these regions are often modified. This is problematic since, in that case, the extrapolated regions do not exactly match the input information at the mask boundary. To address this problem, we use a u-net-like architecture, with skip-connections between the encoder and decoder to help preserve further the input regions.

2.2 Perceptual loss

One way to better complement the adversarial loss is to add a completion loss not directly in the image space, which results in blurry images, but in the representation space of a pretrained network such as VGG-16. As it has been pointed out [5], the first layers of a VGG network trained on large databases such as ImageNet learn filters related to image structures at different scales. Comparing images through such deep features rather than pixel-wise intensities is thus more meaningful in terms of visual structure and semantics. This so-called "perceptual" loss can be written:

$$\mathcal{L}^{vgg}_{compl}(\theta_g) = \frac{1}{N\sum_{\ell=1}^{L}\lambda_\ell}\sum_{\ell=1}^{L}\sum_{i=1}^{N}\lambda_\ell \left\| \phi_l(G_\theta(M_i \odot Z_i)) - \phi_l(Z_i) \right\|_2^2, \tag{5}$$

where ϕ_ℓ denotes the output of the ℓ^{th} layer of VGG-16 and, classically, $\lambda_1 = 1$, $\lambda_2 = 0.5$, $\lambda_3 = 0.25$, $\lambda_4 = 0.125$, $\lambda_5 = 0.0625$ and $\lambda_\ell = 0$ for all the fully-connected layers.

2.3 Mask generation

During training, for each RGB image Z we generate a binary mask $M = (M^c)_{c=1}^3$ over the image channels. For each channel, M^c is defined by a rectangle $R^c(S) = (x^c, y^c, w^c, h^c)$ of size $w^c \times h^c$, lower-left corner (x^c, y^c) and area SWH, with $S \in (0, 1)$ the image masking ratio hyperparameter. Figure 3 summarizes different configurations for the mask generation process. Note that except for inpainting, in all of them, the rectangle interior defines the un-masked image region, the one that the network sees. The most general masking is used to perform "random extrapolation and colorization" (REC, Fig. 3(5)). This task amounts to the completion of the image over the intersection of the three channel-wise masked regions and the colorization of

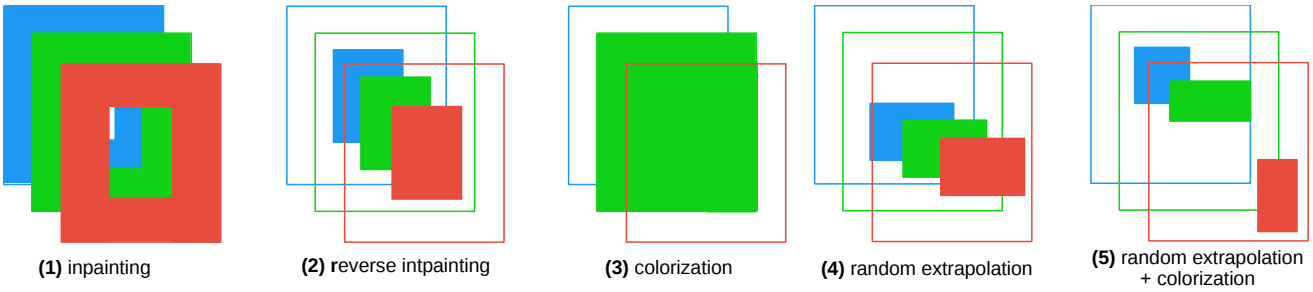

Figure 3: **Different masking methods for different image completion tasks**. (1) Inpainting (I): complete center given periphery; (2) Reverse Inpainting (RI): complete periphery given center: (3) Colorization: complete one or two color channels given the rest; (4) Random Extrapolation (RE): as RI but with a random known region; (5) Random Extrapolation and Colorization (REC): the most general task of completing image from independent random masking in the three channels.

remaining regions. Mask sampling is done as follows in each channel independently:

$$\begin{aligned} &x^c \sim U(0, W - w^c),\ y^c \sim U(0, H - h^c) \\ &h^c \sim U(SH, H),\ w_c = SWH/h^c \\ &M^c(x, y) = \mathbf{1}_{[x^c < x \leq x^c + w^c]} \mathbf{1}_{[y^c < y \leq y^c + h^c]}, \end{aligned} \tag{6}$$

where $U(a, b)$ denotes the uniform distribution over interval $[a, b]$.

The other completion tasks are special cases of REC. For instance, random extrapolation (RE, Fig. 3(4)) is the particular case where the masks are the same for all channels. For colorization (Fig. 3 (3)), the mask covers the entirety of one or two channels and nothing of remaining channels, Reverse Inpainting (RI, Figure 3(2)) is obtained from RE by fixing the mask coordinates and dimensions. Finally, inpainting (I, Figure 3(1)) is obtained from RI by switching the binary mask M to $1 - M$.

The proportion of dropped pixels (*i.e.* those for which all channels are missing) in the RI and RE tasks is exactly $1 - S$. In the general case of REC, when boxes are different across channels, an average proportion of $(1 - S)^3$ pixels is dropped and an average proportion of $1 - S^3$ is corrupted (at least one channel is missing). When $S = 0.1$ as in most of our experiments, this amounts to 72.9% (resp. 99.9%) of dropped (resp. corrupted) pixels in average.

2.4 Hide-and-seek loss

Despite the use of adversarial training and perceptual loss, the generator quickly learns to reconstruct the non-masked regions, which results in discontinuities on the boundaries of the masked regions. To avoid this, we design a novel adversarial mask coordinates regression loss for the discriminator, which shall estimate the locations of the original input masks by looking at the generated images (for MDE-RE and MDE-REC). Formally, we denote $r^c = (x^c/W, y^c/H, (x^c + w^c)/W, (y^c + h^c)/H)$ the normalized lower-left and upper-right coordinates of the ground truth box for channel c and $\hat{r}^c(\theta_g, \theta_d)$ a 4-dimensional sigmoid layer added at the end of the discriminator network (one for each channel). The adversarial mask regression loss reads:

$$\mathcal{L}^{HnS}_{disc}(\theta_d) = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \|r^c_i - \hat{r}^c_i(\theta_g, \theta_d)\|. \tag{7}$$

In case of a fake image, this loss makes the discriminator "seek" the original mask. On the other hand, the generator tries to "hide" it from the discriminator, *e.g.* by assigning to the regressed values $\hat{r}^c_i$ the coordinates of a randomly generated box (one per epoch) q^c_i:

$$\mathcal{L}^{HnS}_{gen}(\theta_g) = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \|q^c_i - \hat{r}^c_i(\theta_g, \theta_d)\|. \tag{8}$$

We refer to the sum of these losses as $\mathcal{L}^{HnS}(\theta_g, \theta_d)$. Note that in case of a real image, this loss is simply not used. In other words, this new game between the generator and discriminator networks ensures that there is no trace left of the generation process within the images, hence it helps reduce the artifacts caused by adversarial training. However, as pointed out in [8], regressing coordinates is a hard task for convolutional networks as their structure enforces translational invariance. Thus, we also experiment with concatenating 2 channels containing x and y-coordinates to the discriminator's inputs. We refer to this version as $\mathcal{L}^{HnS}_{coord}$. Our total loss is:

$$\begin{aligned} \mathcal{L}_{tot}(\theta_g, \theta_d) = \mathcal{L}_{rec}(\theta_g) + \lambda_{compl} \mathcal{L}^{vgg}_{compl}(\theta_g) \\ + \lambda_{adv} \mathcal{L}_{adv}(\theta_g, \theta_d) + \lambda_{HnS} \mathcal{L}^{HnS}_{coord}(\theta_g, \theta_d). \end{aligned} \tag{9}$$

2.5 Implementation details

As shown on Figure 4, the generator is composed of an encoder and a decoder. The encoder is similar to VGG network,

except it only uses one large fully-connected layer at the end. As it is a classical setup in the literature, the decoder mirrors the encoder, but here with the addition of skip-connections to explicitly preserve the non-masked regions.

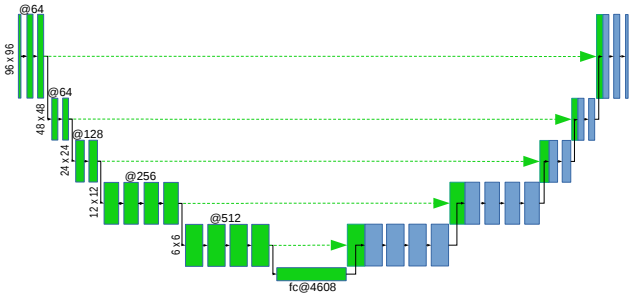

Figure 4: **Architecture of the MDE generator**. Green: encoder, Blue: decoder. The blocks indicate conv/batch-norm/ReLU layers and the descending/ascending arrows indicate downsampling (strided convolution) and upsampling operators (transposed convolution).

The discriminator is very similar to the encoder part of the generator, except that the fully-connected layer is replaced by a global average pooling: as discriminating between real and fake images is considered easier than generating images, it is assumed that the discriminator shall have fewer parameters. As in [16], we use leaky ReLU activations in the discriminator and strided convolutions everywhere instead of max-pooling. We also use a sigmoid layer as the last layer of the generator to better scale the outputs. We use ADAM optimizer with a learning rate of 2.10^{-4} for the generator and 2.10^{-5} for the discriminator. We train with a momentum of 0.5 and polynomial learning rate annealing. Finally, we apply $300\,000$ updates with batch size 24 to train the network.

3 Experiments

We validate our method both qualitatively and quantitatively on three datasets, to show its interest for image completion, representation learning as well as face occlusion handling. The **MNIST** database contains $55\,000$ train and $10\,000$ test images. As MNIST images are grayscale and low resolution, we upscale them to 96×96 and only apply MDE-RE on this dataset. The **Oxford-102** flowers dataset consists in 8187 images describing 102 classes of flowers. We train our models on 7167 images from the train and test partitions, and apply them on the 1020 validation images. We report results obtained with MDE-RI, MDE-RE and MDE-REC. The **CelebA** database [9] is a large-scale face attribute database which contains $202\,599$ 218×178 celebrity images coming from $10\,177$ identities, each annotated with 40 binary attributes (such as *gender*, *eyeglasses*, *smile*), and 5 landmarks (nose, left and right pupils, mouth corners). As in [20], we use the train partition that contains $162\,770$ images from $8k$ identities to train our models. The test partition contains $19\,962$ instances from $1k$ identities that are different from the training set identities. In all our tests, we use the aligned images, apply a constant rescaling factor (0.75) to crop the face region and resize it to 96×96. All evaluations are performed on the test sets for all datasets.

3.1 Qualitative evaluation

3.1.1 Image completion

Figure 5 shows images generated with MDE on the three datasets. In all cases, the images look very realistic: On MNIST, the generated digits usually match the ground truth ones. On Oxford-102, both the flowers and backgrounds are generated correctly. This implies that even with few data, MDE is able to capture the data distribution. Similarly, on CelebA, the generated images may present some alterations w.r.t. the ground truth images: the generator may suppress particularly low-probability patterns, such as beards, glasses, hats or a particular facial expression. Notice however that the quality of the completion is generally high, as there is no blurry pattern or artifact on the generated images. Figure 6 shows more results on CelebA. For each ground truth (GT) image, alternative completions can be generated by applying a new mask before passing the images to the generator. Depending on the mask position and dimensions, the generator can discard background information, or swap haircuts, remove beards or mustache, or change the facial expression.

3.1.2 MDE resampling

Figure 7 shows sequences of images generated by iteratively resampling an MDE-REC: for a ground truth image, we apply a random mask and generate a new image from it. We then pass the generated image in the generator once again with a new random mask, and repeat these steps 10 times. Note that at each step, more than 70% of the pixels are completely dropped in average, however the generator generally preserves a lot of semantic information, such as hair color or style, facial expression, gender or ethnicity. After a number of passes through the generator, such information is lost and the faces can be very different from the GT image. The images, however, are still highly realistic, indicating that MDE-REC learns a stable manifold of faces that encompasses face geometry and semantics, which we validate through quantitative evaluations.

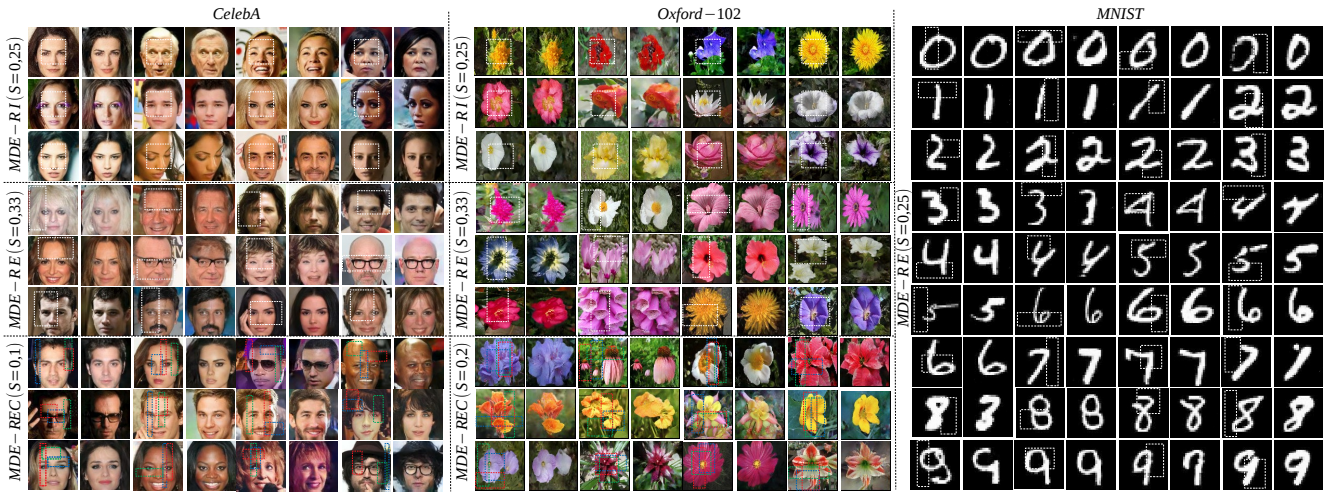

Figure 5: **Completing images with proposed Missing Data Encoders**. *Left*: Examples of images generated with MDE-RI (3 top rows, $S = 0.25$), MDE-RE (3 central rows, $S = 0.33$) and MDE-REC (3 bottom rows, $S = 0.1$) on CelebA. *Center:* images generated with MDE-RI (3 top rows, $S = 0.25$), MDE-RE (3 central rows, $S = 0.33$) and MDE-REC (3 bottom rows, $S = 0.2$) on Oxford-102. *Right:* Examples of images generated with MDE-RE ($S = 0.25$) on MNIST. Images with the dashed boxes are ground truth images and the boxes indicate the non-masked information. For MDE-REC on CelebA and Oxford-102, the red, green and blue boxes show preserved information in R,G,B channels, respectively. Best viewed in color.

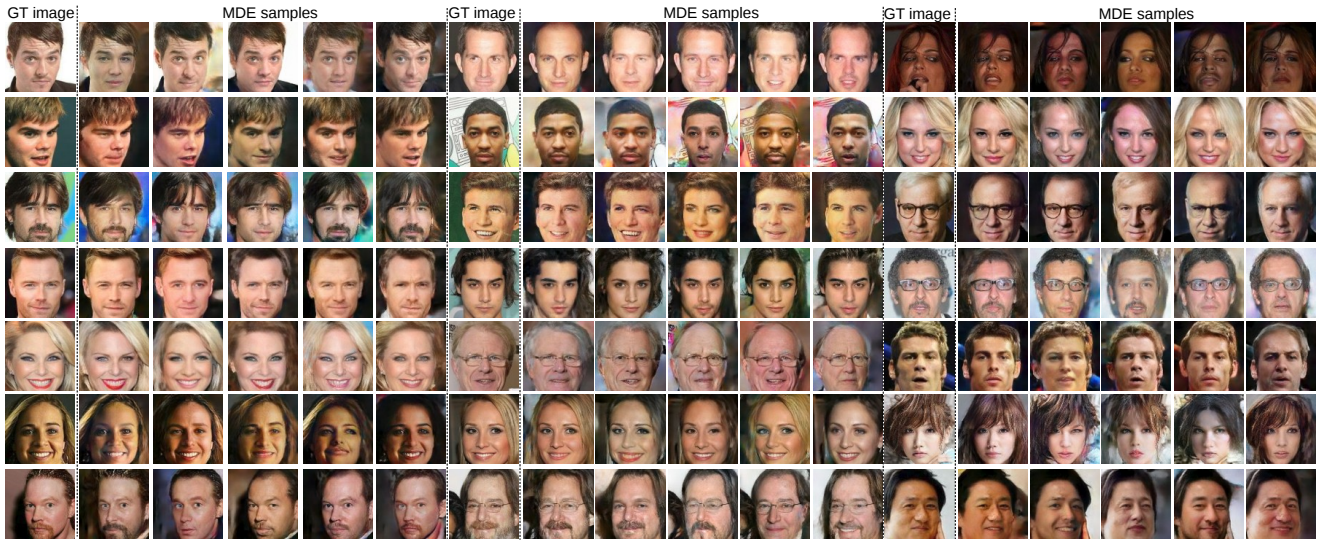

Figure 6: **Generating multiple image completions with MDE-REC**. For each GT image, a trained MDE (with $S = 0.1$) is sampled 5 times with different input masks.

3.2 Quantitative evaluation

3.2.1 Evaluation metrics

We use several metrics to assess the quality of the generated images. The peak signal-to-noise ratio (*pSNR*) quantifies the pixel-wise resemblance between the generated and ground truth images. The structural similarity (*SSIM*) index assesses the holistic visual quality of a completion. Lastly, we measure the *inception score* [17], which evaluates both the semantic relevance of the generated images as well as their diversity. As computing the inception score requires using a network pretrained on a similar distribution (in our case, a face database), we use VGG-face, as in [18]. For the same reason, we only perform quantitative evaluation on CelebA.

Figure 7: **Generating sequences of novel images through completion**. Sequences of ten images generated from an original GT one by successive MDE resampling (with $S = 0.1$).

3.2.2 Ablation study

Figure 8 shows pSNR and inception score for multiple train and test scenarios. First, we observe that models trained for colorization, inpainting or reverse inpainting have very low pSNR and inception score when tested in a mismatched scenario (*e.g.* training with inpainting and testing for colorization). On the contrary, MDE-RE also performs well for inpainting and reverse inpainting, as these scenarios can be viewed as special cases of random extrapolation. However, MDE-RE generalizes poorly to colorization as well as REC tasks. Conversely, MDE-REC performs very well on every task both in terms of pSNR and inception score. In terms of pSNR, MDE-RE and MDE-REC trained with high S tend to be better when S is also high in test, and vice-versa, for both RE and REC tasks. However, MDE-RE and MDE-REC trained with $S = 0.1$ always have higher inception score. Therefore, MDE-REC with $S = 0.1$ is a more generic model that has a better transferability to other completion tasks.

Second, we train a MDE-REC with $S = 0.1$ and completion loss, adversarial loss, perceptual loss as well as hide-and-seek loss. Furthermore, we always add a reconstruction loss to ensure that the input information is preserved. We set $\lambda_{rec}^{vgg} = 2.10^{-5}$, $\lambda_{adv} = 10^{-2}$ and $\lambda_{HnS} = 10^{-2}$. Figure 9 draws a comparison between those approaches. As it is classical in the GAN literature, optimizing only $\mathcal{L}_{rec}$ leads to high pSNR/SSIM, but results in blurry images, which have a low inception score. Using adversarial training and *a fortiori* perceptual loss leads to better quality. This is because not only pixel-level information is matched between the generated and ground truth images but also higher-order statistics such as edges intensities for lower VGG layers, and more semantically abstract information for downstream layers. Furthermore, using $\mathcal{L}_{rec}^{vgg}$, $\mathcal{L}_{adv}$ and $\mathcal{L}_{coord}^{HnS}$ yields the best results for every S.

3.2.3 What does MDE learn?

To study the representations learned by different MDE models, we quantify the transferability of the learned features for attribute prediction and landmark alignment. To do so, we truncate the pretrained MDE models after the fully-connected layer, and append two $4000 \rightarrow 40$ and $4000 \rightarrow 10$ fully-connected sigmoid layers to map the attributes and landmark coordinates, respectively. We then minimize a L_2-norm to map these outputs to the 40 attributes and 10 landmarks coordinates, respectively. We perform 5000 updates with batch size 16 (*i.e.* less than one epoch). We report in table 1 the average Euclidean distance between the landmarks as well as the average trace of the confusion matrices.

We observe that reverse inpainting as a pretraining transfers more efficiently to landmark localization and attribute prediction, as compared to inpainting. When compared with colorization, it is less accurate on the attribute prediction task, but better at predicting the face geometry. This stems from the fact that models trained with reverse inpainting only see a limited fraction of the input image. Conversely, MDE-RE models obtain high performance for predicting attributes but a slightly lower accuracy in landmark localization. Finally, MDE-REC models are significantly better for both landmark localization and attribute prediction. Through the channel-wise random region dropping and completion, they benefit from both completion and colorization pre-trainings at the same time. By doing so, they learn to encode the face geometry and high-level semantics in a more efficient way. Note that for both MDE-RE and MDE-REC, the models trained with lower S are not necessarily the best at predict-

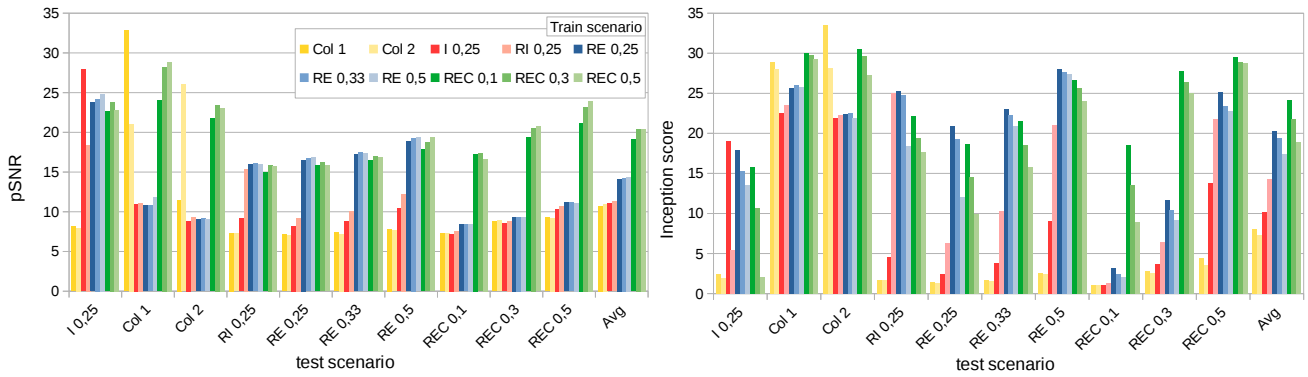

Figure 8: **Comparing MDE variants in different test setups**. pSNR and Inception score charts for models trained with various tasks and evaluated in different scenarios. I: inpainting. RI: reverse inpainting. Col 1-2: colorization (1-2 channels). RE: random extrapolation. REC: random extrapolation and colorization. For I, RI, RE and REC, appended number indicates the value of masking ratio S.

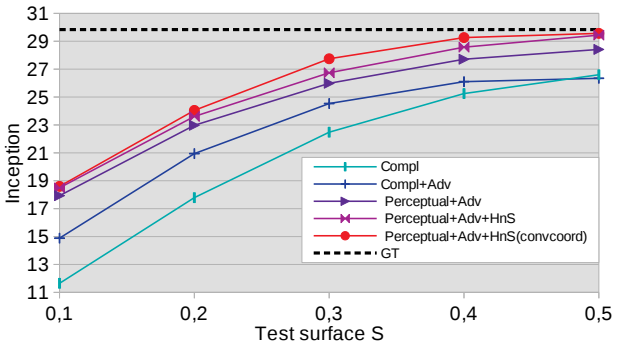

Figure 9: **Loss ablation**. Inception score for MDE-REC ($S = 0.1$) with different loss combinations.

ing attributes: this is due to fine-grained attributes such as the presence of earrings or lipstick not being successfully embedded within the generator.

Table 2 shows a comparison with recent state-of-the-art approaches, and MDE-REC trained with 50 000 updates. Our method is competitive with recent methods that use bigger architectures [11, 4] or pre-training involving large annotated dataset (350k face recognition dataset) [20]. This shows that MDE-REC learns useful representations in a completely unsupervised fashion.

3.2.4 Face completion under targeted occlusions:

We also study the application of MDE-REC ($S = 0.1$) to face completion under occlusions. We use the same protocol as in [7] and compare with state-of-the-art methods, CE [15] and GFC [7], without postprocessing. The results show that MDE-REC is more efficient than the random inpainting proposed by [7]. In addition, our method is agnostic to the nature of the dataset, as opposed to [7], where the authors use an auxiliary face parsing network. As shown in Table 3, results for MDE are significantly better nearly everywhere. Furthermore, high values of the inception score (which ranges from 18.80 to 27.28) indicates that the generated images are sharp and realistic.

Table 1: **Performance comparison for facial landmark localization and attribute recognition**. Comparison after only 5000 updates. "Landmarks": average point-to-point error. "Attributes": average trace of the confusion matrices obtained for each attribute.

Pretraining	Landmarks	Attributes
Random weights	9.753	19.79
Colorization (1c) [19]	2.358	11.69
Colorization (2c) [19]	2.278	10.92
Inpainting [15]	5.411	16.12
MDE-RI	1,496	13,73
MDE-RE(0.25)	2.039	12.72
MDE-RE(0.33)	1.719	11.41
MDE-RE(0.5)	1.759	10.39
MDE-REC(0.1)	1.509	10.49
MDE-REC(0.3)	**1.451**	10.30
MDE-REC(0.5)	1.498	**10.16**

4 Conclusion

In this paper, we introduced the Missing Data Encoder for image completion, unsupervised representation learning and face occlusion handling. The network is trained to complete an image from a rectangular region drawn at random in each channel independently, a task that subsumes to some

Table 2: **Facial attribute recognition**. Comparison of unsupervised MDE pre-training with state-of-the-art (% avg. error).

Method	attributes
Supervised pre-training [20]	13.4
Single-task baseline [4]	10.37
Multi-task baseline [4]	9.58
Parallel order [11]	10.21
Parallel order+landmarks [11]	10.29
Soft order+identity [11]	**8.64**
MDE-REC(0.5)	9.17

Table 3: **Comparison with state-of-the-art for face completion under targeted occlusions**. Results for context encoder (CE [15]) and generative face completion (GFC [7]) are excerpted from [7].

	pSNR			SSIM		
Occlusion	CE	GFC	MDE	CE	GFC	MDE
Right half	18.6	19.4	**21.6**	0.772	0.804	**0.814**
Left half	18.4	19.3	**21.8**	0.774	0.808	**0.815**
Both eyes	17.9	18.3	**21.8**	0.719	0.731	**0.839**
Right eye	19.0	19.1	**22.4**	0.754	0.759	**0.855**
Left eye	19.1	18.9	**22.6**	0.757	0.755	**0.860**
Mouth	19.3	19.7	**21.9**	0.818	**0.824**	0.818

extent inpainting, reverse inpainting and colorization. We showed on several datasets that the proposed method allows the generation of high quality images from only small seed fragments. By learning to do so, our architecture captures without supervision high level semantic information within its embedding. It also extends the state-of-the-art for face completion under occlusion. Future work involves using MDE pretraining for classification or semantic segmentation, as well as investigating the use of the proposed "hide and seek" adversarial loss for other applications such as object detection.

References

[1] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 1, 2

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[3] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 1

[4] K. He, Z. Wang, Y. Fu, R. Feng, Y.-G. Jiang, and X. Xue. Adaptively weighted multi-task deep network for person attribute classification. In *ACM Multimedia*, 2017. 8, 9

[5] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

[6] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017. 2

[7] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *CVPR*, 2017. 2, 3, 8, 9

[8] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv preprint:1807.03247*, 2018. 4

[9] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 5

[10] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 1, 2, 3

[11] E. Meyerson and R. Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. In *ICLR*, 2018. 8, 9

[12] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1, 2

[13] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2

[14] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, 2017. 2

[15] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2, 3, 8, 9

[16] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 5

[17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016. 6

[18] Z. Wang, X. Tang, W. Luo, and S. Gao. Face aging with identity-preserved conditional generative adversarial networks. In *CVPR*, 2018. 6

[19] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 2, 8

[20] Y. Zhong, J. Sullivan, and H. Li. Face attribute prediction using off-the-shelf cnn features. In *ICB*, 2016. 5, 8, 9