





Learning Scale-Invariant Object Representations with a Single-Shot Convolutional Generative Model

Piotr Zieliński^(✉)  and Tomasz Kajdanowicz 

Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27,
50-370 Wrocław, Poland
p.zielinski@pwr.edu.pl

Abstract. Contemporary machine learning literature highlights learning object-centric image representations' benefits, i.e. interpretability, and the improved generalization performance. In the current work, we develop a neural network architecture that effectively addresses the task of multi-object representation learning in scenes containing multiple objects of varying types and sizes. In particular, we combine SPAIR and SPACE ideas, which do not scale well to such complex images, and blend them with recent developments in single-shot object detection. The method overcomes the limitations of fixed-scale glimpses' processing by learning representations using a feature pyramid-based approach, allowing more feasible parallelization than all other state-of-the-art methods. Moreover, the method can focus on learning representations of only a selected subset of types of objects coexisting in scenes. Through a series of experiments, we demonstrate the superior performance of our architecture over SPAIR and SPACE, especially in terms of latent representation and inferring on images with objects of varying sizes.

Keywords: Deep autoencoders · Representation learning · Generative models · Scene analysis

1 Introduction

The ability to discriminate and reason about individual objects in an image is one of the important tasks of computer vision, which is why object detection and instance segmentation tasks have drawn vast attention from researchers throughout the years. The latest advances in artificial intelligence require a more insightful analysis of the image to provide more profound reasoning about its contents. It can be achieved through representation learning, which facilitates extracting useful information about objects, allowing transferring more general knowledge to other tasks [2]. One can see multi-object representation learning as a natural extension to the aforementioned computer vision tasks. Here, the objective is to produce a valuable abstract feature vector of each of the inferred

objects and hence produce a structured representation of the image, allowing for its more insightful understanding.

Recently, the most successful methods are based on the variational autoencoder (VAE) framework [16, 21], with structured latent space, which includes individual objects' representations. The original approach consists in extracting object latent vectors with a recurrent network [1, 3, 7–9]. Alternatively, each object's representation can be produced with a single forward pass through the network by employing a convolution-based single-shot approach [4, 18]. However, these methods are limited by a single feature map utilized to create objects' latent vectors and hence cannot be used when object sizes vary.

In this paper, we propose a single-shot method for learning multiple objects' representations, called *Single-Shot Detect, Infer, Repeat* (SSDIR¹). It is a convolutional generative model applying the single-shot approach with a feature pyramid for learning valuable, scale-invariant object representations. By processing multi-scale feature maps, SSDIR can attend to objects of highly varying sizes and produce high-quality latent representations directly, without the need of extracting objects' glimpses and processing them with an additional encoder network. The ability to focus on individual objects in the image is improved by leveraging knowledge learned in an SSD [19] object detection model. In experiments, we compare the SSDIR model on multi-scale scattered MNIST digits, CLEVR [15] and WIDER FACE [23] datasets with other single-shot approaches, proving the ability to focus on individual objects of varying sizes in complicated scenes, as well as the improved quality of objects' latent representations, which can be successfully used in other downstream problems, despite the use of an uncomplicated convolutional backbone.

We summarize our contributions as follows. We present a model that enhances multi-object representation learning with a single-shot, feature pyramid-based approach, retaining probabilistic modeling of objects. We provide a framework for generating object representations directly from feature maps without extracting and processing glimpses, allowing easier scaling to larger images. We compare the method with other single-shot multi-object representation learning models and show its ability to attend to objects, the improved latent space quality, and applicability in various benchmark problems.

2 Related Works

Multi-object representation learning has recently been tackled using unsupervised, VAE-based models. Two main approaches include sequential models, attending to a single object or part of the image at a time, and single-shot methods, which generate all representations in a single forward pass through the network.

The original approach to this problem was presented by Ali Eslami *et al.* in [1]. The *Attend, Infer, Repeat* (AIR) model assumes a scene to consist of objects,

¹ Code available at: <https://github.com/piotlinski/ssdir>.

represented with *what* vector, describing the object’s appearance, *where* vector indicating its position on the image and *present* vector, describing if it is present in the image, controlling termination of the recurrent image processing. The model attends to a single object at a time, generating representations sequentially with a recurrent network until a non-present object is processed. Other studies, including [10] and [22] proposed a different approach, where objects representations are learned using Neural Expectation-Maximization, without structuring the latent representations explicitly. These methods suffer from scaling issues, not being able to deal with complex scenes with multiple objects.

Alternatively, an image might be described with a scene-mixture approach, as in MONet [3], IODINE [9] and GENESIS [7, 8]. Here, the model does not explicitly divide the image into objects but instead generates masks, splitting the scene into components, which the model encodes. In the case of MONet and GENESIS, each component is attended and encoded sequentially, while IODINE uses amortized iterative refinement of the output image. However, these methods are not a good fit for learning object representations in an image, as scene components usually consist of multiple objects. Furthermore, masks that indicate particular objects limit the model’s scalability due to this representation requiring more memory than bounding box coordinates.

GENESIS belongs to a group of methods, which focus on the ability to generate novel, coherent and realistic scenes. Among them, one should notice recent advances with methods leveraging generative adversarial networks (GANs), such as RELATE [6] or GIRAFFE [20]. Compared to VAE-based methods, they can produce sharp and natural images, which are more similar to original datasets. However, these models do not include an explicit image encoder, and therefore cannot be applied for multi-object representation learning directly. What is more, the process of training GANs tends to be longer and more complicated than in the case of VAEs.

Recently, methods such as GMAIR [24] postulate that acquiring valuable *what* object representations is crucial for the ability to use objects encodings in other tasks, such as clustering. Here, researchers enhanced the original *what* encoder with Gaussian Mixture Model-based prior, inspired by the GMVAE framework [11]. In our work, we also emphasize the importance of the *what* object representation and evaluate its applicability in downstream tasks.

One of the promising methods of improving model scalability of VAE-based multi-object representation learning models was presented in SPAIR [4], where the recurrent attention of the original AIR was replaced with a local feature maps-based approach. In analogy to single-shot object detection models like SSD [19], the SPAIR first processes image with a convolutional backbone, which returns a feature map with dimensions corresponding to a fixed-sized grid. Each cell in the grid is then used to generate the locations of objects. Objects representations’ are inferred by processing these cells sequentially, generating *what*, *depth* and *present* latent variables, describing its appearance, depth in the scene, and the fact of presence. This approach has recently been extended in SPACE [18], which fixes still existing scalability issues in SPAIR by employing parallel

latent components inference. Additionally, the authors used the scene-mixture approach to model the image background, proving to be applicable for learning objects’ representations in more complex scenes. However, both methods rely on a single grid of fixed size, which makes it difficult for this class of models to attend to objects of highly varying sizes. What is more, both of them employ glimpse extraction: each attended object is cut out of the input image and processed by an additional encoder network to generate objects’ latent representations; this increases the computational expense of these methods.

Latest advances in the field of multi-object representation learning try to apply the aforementioned approaches for inferring representations of objects in videos. SQAIR [17] extends the recurrent approach proposed in AIR for sequences of images by proposing a propagation mechanism, which allows reusing representations in subsequent steps. A similar approach was applied to single-shot methods by extending them with a recurrent network in SILOT [5] and SCALOR [14]; here, the representations were used in the object tracking task. An interesting approach was proposed by Henderson and Lambert [12]. Authors choose to treat each instance within the scene as a 3D object; the image is then generated by rendering each object and merging their 2D views into an image. This allows for a better understanding of objects’ representations, at the cost of significantly higher computational complexity.

3 Method

SSDIR (**S**ingle-**S**hot **D**etect, **I**nter, **R**epeat) is a neural network model based on a variational autoencoder architecture [16, 21] as shown in Fig. 1; its latent space consists of structured objects’ representations \mathbf{z} , enhanced by leveraging knowledge learned in a single-shot object detection model SSD [19], both sharing the same convolutional backbone.

3.1 The Proposed Model: SSDIR

Our model extends the idea of single-shot object detection. Let \mathbf{x} be the image representing all relevant (i.e. detected by the SSD) objects present in the image. SSDIR is a probabilistic generative model, which assumes that this image is generated from a latent representation \mathbf{z} according to a likelihood distribution. This representation consists of a set of latent vectors assigned to each grid cell in the feature pyramid of SSD’s convolutional backbone and is sampled from a prior distribution $p(\mathbf{z})$. Since the likelihood distribution is unknown, we approximate it using the decoder network θ , which parametrizes the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. Then, the generative model can be described as a standard VAE decoder (1).

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} \quad (1)$$

To do inference in this model, SSDIR applies variational method and approximates the intractable true posterior with a function $q_\phi(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$,

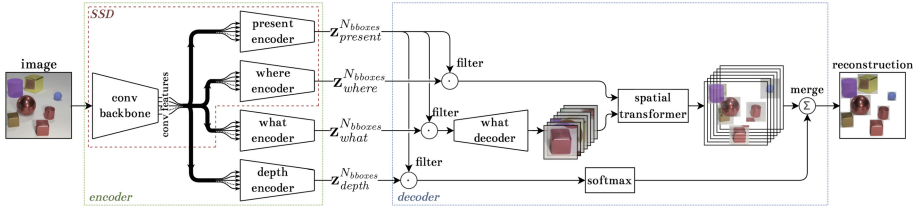


Fig. 1. Illustration of the SSDIR model. It consists of two fully-convolutional neural networks: an *encoder* and a *decoder*. The *encoder* uses a convolutional backbone as a feature extractor, which builds a pyramid of multi-scale features processed by each latent component encoder. Each object’s position z_{where} and presence $z_{present}$ latent vectors are computed using a trained object detection model *SSD*, indicating grid cells, which refer to detected objects; z_{what} and z_{depth} are computed with additional convolutional encoders, which process the feature maps from the pyramid in a similar manner to *SSD*. In the decoder, all latents are filtered to include only present objects for reconstructions. *What* decoder reconstructs appearances of each present object, which are then put in their original place with an affine transformation in the *spatial transformer* module. Finally, object reconstructions are merged using weighted sum, created by applying *softmax* on objects’ *depth* latents.

parametrized by ϕ (encoder parameters). This allows us to use ELBO (Evidence Lower Bound) as the loss function (2):

$$\mathcal{L}(\theta, \phi) := \mathbb{E}_{z \sim q_\phi(z|\mathbf{x})} [\log p_\theta(\mathbf{x}|z)] - D_{KL}(q_\phi(z|\mathbf{x}) \| p(z)) \quad (2)$$

where D_{KL} is the KL divergence.

Object Representation. SSDIR extends the grid-based approach with a feature pyramid for object detection proposed in SSD to produce objects’ latent representations. We assume each object can be described by four latent variables:

- $z_{where} \in \mathbb{R}^4$ – the object’s bounding box position and size,
- $z_{present} \in \{0, 1\}$ – a binary value indicating if given cell contains any object,
- $z_{what} \in \mathbb{R}^D$ – D -sized vector describing the object appearance,
- $z_{depth} \in \mathbb{R}$ – a real number indicating how deep in the scene the given object was observed (we assume, that objects with a bigger value of z_{depth} appear in front of those with a lower value).

To simplify the process of objects discovery, we reuse a trained SSD model to get bounding box position and size, as well as the detected object class. SSDIR utilizes detections to produce z_{where} and $z_{present}$ as shown in (3) and (4).

$$z_{where}^i = [cx_i \ cy_i \ w_i \ h_i] \quad (3)$$

$$z_{present}^i \sim \text{Bernoulli}(\beta^i) \quad (4)$$

where:

i refers to the cell in the feature pyramid,
 cx , cy are the bounding box' center coordinates,
 w , h are the bounding box' width and height dimensions,
 $\beta^i = \begin{cases} \arg \max_k c_i & \text{if an object detected in the cell,} \\ 0 & \text{otherwise,} \end{cases}$
 c are the object's predicted class confidences.

The two remaining latent components: \mathbf{z}_{what} and \mathbf{z}_{depth} are modeled with Gaussian distributions, as shown in (5) and (6).

$$\mathbf{z}_{what}^i \sim \mathcal{N}(\boldsymbol{\mu}_{what}^i, \boldsymbol{\sigma}_{what}^i) \quad (5)$$

$$\mathbf{z}_{depth}^i \sim \mathcal{N}(\boldsymbol{\mu}_{depth}^i, \boldsymbol{\sigma}_{depth}^i) \quad (6)$$

where:

$\boldsymbol{\mu}_{what}$, $\boldsymbol{\mu}_{depth}$ are means, encoded with *what* and *depth* encoders,
 $\boldsymbol{\sigma}_{what}$, $\boldsymbol{\sigma}_{depth}$ are standard deviations, which are treated as model's hyperparameters.

SSDIR Encoder Network. To generate the latent representation of objects contained in an image, we apply the feature pyramid-based object detection approach. The function of the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ is implemented with a convolutional backbone (VGG11) accepting images of size $300 \times 300 \times 3$, extended with a feature pyramid, and processed by additional convolutional encoders, as shown in Fig. 1. Specifically, *where*, *present* and *depth* encoders contain single convolution layer with 3×3 kernels (1 in case of *present* and *depth* and 4 for *where* encoder) per each feature map in the pyramid, whereas *what* encoder may include sequences of convolution layers with ReLU activations, finally returning D -sized vector for each cell in each feature pyramid grid. The outputs of these encoders are used to generate latent vectors \mathbf{z}_{where} , $\mathbf{z}_{present}$, \mathbf{z}_{what} and \mathbf{z}_{depth} .

The backbone's, as well as *where* and *present* encoders' weights are transferred from an SSD model trained with supervision for detection of objects of interest in a given task and frozen for training; *what* and *depth* encoders, which share the same pretrained backbone, are trained with the decoder network. Such architecture allows parallel inference, since neither latent component depends on any other, without the need of extracting glimpses of objects and processing them with a separate encoder network – in SSDIR latent representations are contained within feature maps directly, improving its scalability.

SSDIR Decoder Network. Latent representations of objects in the picture are forwarded to the decoder network to generate reconstructions of areas in the input image that contain objects of interest, i.e. those detected by the SSD network. First, the latent variables are filtered according to $\mathbf{z}_{present}$, leaving only those objects, which were found present in the image by the SSD network.

Next, per-object reconstructions are generated by passing filtered \mathbf{z}_{what} vectors through a convolutional *what* decoder, producing M images of size $64 \times 64 \times 3$, representing each detected object’s appearance. These images are then translated and scaled according to the tight bounding box location \mathbf{z}_{where} in the *spatial transformer* module [13]. The resulting M $300 \times 300 \times 3$ images are merged using a weighted sum, with softmaxed, filtered \mathbf{z}_{depth} as the weights. The output of the model might then be normalized with respect to the maximum intensity of pixels in the reconstruction to improve the fidelity of the reconstruction.

SSDIR does not require special preprocessing of the image, apart from the standard normalization used widely in convolutional neural networks. Originally, the background is not included in the reconstruction phase, since its representation is not crucial in the task of multi-object representation learning; we assume that this way SSDIR learns to extract the key information about all objects from the image. The background might however be reconstructed as well by including an additional \mathbf{z}_{what} encoder and treating the background as an extra object, which is transformed to fill the entire image and put behind all other objects.

The parallel nature of the model is preserved in the decoder. The operations of filtering, transforming, and merging are implemented as matrix operations, allowing good performance and scalability.

Training. The SSDIR model is trained with a modified ELBO loss function. We extend the original form (2), which intuitively includes reconstruction error of an entire image and KL divergence for latent and prior distributions with a normalized sum of each detected object’s reconstruction error. This allows the model to reach high quality of reconstructions (and as a result – high quality of \mathbf{z}_{what} latent representations) and correct order of objects’ \mathbf{z}_{depth} , preserving transformation function continuity thanks to KL divergence-based regularization. The final form of the loss function is shown in (7).

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \theta, \phi) &= \alpha_{obj} \mathbb{E}_{\mathbf{z}} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] + \alpha_{rec} \frac{1}{M} \sum_i^M \mathbb{E}_{z_i} [\log p_{\theta}(x_i|z_i)] \\ &\quad - \alpha_{what} D_{KL}(q_{\phi}(\mathbf{z}_{what}|\mathbf{x}) \| p(\mathbf{z}_{what})) \\ &\quad - \alpha_{depth} D_{KL}(q_{\phi}(\mathbf{z}_{depth}|\mathbf{x}) \| p(\mathbf{z}_{depth})) \end{aligned} \quad (7)$$

where:

$\mathbb{E}_{\mathbf{z}} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]$ is the likelihood of the reconstruction generated by the decoder, $\mathbb{E}_{z_i} [\log p_{\theta}(x_i|z_i)]$ is the likelihood of an i -th detected object reconstruction, α_{obj} , α_{rec} , α_{what} , α_{depth} are loss components coefficients, modifying the impact of each one on the learning of the model, M is the number of objects detected by the SSD model in a given image.

In case of both \mathbf{z}_{what} and \mathbf{z}_{depth} we assume the prior to be a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The training objective is described by (8) for each image \mathbf{x}_i in the training dataset. The model is trained jointly with gradient ascent

using Adam as the optimizer, utilizing the reparametrization trick for back-propagating gradients through the sampling process. The process of learning representations is unsupervised, although the backbone’s and *where* and *present* encoders’ weights are transferred from a pretrained SSD model.

$$\theta^*, \phi^* = \arg \max_{\theta, \phi} \sum_i \mathcal{L}(\mathbf{x}_i, \theta, \phi) \quad (8)$$

Table 1. Differences between **SSDIR** and baseline methods. “*semi*–” indicates that the object detection model is trained with supervision, while the representation learning procedure is unsupervised. “*glimpses*” refers to the process of learning object’s z_{what} by extracting a sub-image containing the object (based on its z_{where} latent vector) and encoding it with a separate VAE; “*single-shot*” is the approach adopted in SSDIR.

Criterion	Basic VAE	SPAIR [4]	SPACE [18]	SSDIR
Unsupervised	Semi-	✓	✓	Semi-
Inferring representations	Glimpses	Glimpses	Glimpses	Single-shot
Varying sizes	✓	✗	✗	✓
Particular objects type	✓	✗	✗	✓
Parallel encoding	✗	✗	✗	✓

4 Experiments

In this section, we evaluate the performance of SSDIR and compare it with two baseline methods: SPAIR [4] and SPACE [18]. We focus on verifying the ability to learn valuable representations of objects, which sizes vary; this is conducted by analyzing the quality of reconstructions produced by the decoder of each method and applying the produced representations in a downstream task. Besides, we conduct an ablation study to analyze the influence of the dataset characteristics on SSDIR performance.

Our implementation of SPAIR is enhanced with a convolutional encoder instead of the original, fully-connected network, which should improve its performance on more complicated datasets. Since in this work we focus on learning objects’ representation, we consider models without background: SPAIR does not explicitly model it, whereas in SPACE we analyze the foreground module outputs, which tries to reconstruct individual objects in the image. In Table 1 we included a comparison between the analyzed methods, together with an approach employing an object detector, a spatial transformer for extracting glimpses, and a VAE for learning their representations (denominated as *SSD+STN+VAE*).

The datasets used in the research were chosen to resemble common choices among recent multi-object representation learning methods. Among them, we decided to include datasets of various complexity, providing the ability to validate the model on simple images and prove its performance on complex, realistic images. Therefore, we conducted our experiments using three datasets: 1) multi-scale, scattered MNIST digits (with configured minimum and maximum digit

size, as well as grids for scattering digits), 2) CLEVR dataset [15] (containing artificially generated scenes with multiple objects of different shape, material, and size, used widely in the field of scene generation and multi-object representation learning), 3) WIDER FACE [23] (face detection benchmark dataset, with images containing multiple people; the dataset was used to demonstrate the ability of SSDIR to focus on objects of a particular type).

4.1 Per-object Reconstructions

In this section, we present a comparison of images' and objects' reconstructions for the proposed model and the baseline methods. In Fig. 2 we show inputs and reconstructions of representative images from each dataset (test subset, i.e. images not used for training), as well as some individual object reconstructions. Note, that due to the number of objects presented in the image and the nature of the models, it would not be possible to show all reconstructed objects.

Both SPAIR and SPACE can reconstruct the scattered MNIST dataset's image correctly. However, looking at the *where* boxes inferred by these models it is visible, that due to their limited object scale variability they are unable to attend to individual objects with a single latent representation, often reconstructing one digit with multiple objects. This is confirmed by the analysis of object reconstructions: SPAIR builds object reconstructions by combining reconstructed parts of digits, whereas SPACE can reconstruct digits of sizes similar to its preset, but divides bigger ones into parts. SSDIR is able to detect and reconstruct the MNIST image accurately: the use of a multi-scale feature pyramid allows for attending to entire objects, creating scale-invariant reconstructions, which are then mapped to the reconstruction according to tight *where* box coordinates.

SPAIR did not manage to learn object representations in the other two datasets. Instead, it models the image with rectangular boxes, containing a bigger part of an image. The aberrations visible in CLEVR dataset with SPAIR are caused by a transparency mask applied in this model and the fact, that these objects are heavily transformed when merging into the reconstruction. The tendency to model the image with rectangles is even more visible in the WIDER FACE dataset, where SPAIR divides the image in almost equal rectangles, aligned with the reconstruction grid. This effect allows for a fair quality of overall image reconstructions but does not yield valuable object representations.

In the case of SPACE, the model was not able to learn objects' representation in the CLEVR dataset, despite an extensive grid search of the hyperparameters relevant to the foreground module (especially the object's size). Instead, it models them using the background module, which cannot be treated as object representations since they gather multiple objects in one segment (this lies in line with problems reported in the GitHub repository²). Hence, objects reconstructions visible in Fig. 2 for this dataset contain noise. When applied to the WIDER FACE dataset, SPACE tends to approach image reconstruction in the

² <https://github.com/zhixuan-lin/SPACE/issues/1>.

same way as SPAIR, dividing the image into rectangular parts, reconstructed as foreground objects. Similarly, this leads to an acceptable reconstruction quality but does not provide a good latent representation of the image’s objects.

SSDIR shows good performance on the CLEVR dataset: it can detect individual objects and produce their latent representations, which results in good quality reconstructions. Similarly, in the case of the WIDER FACE dataset, the model is able to reconstruct individual faces. However, due to the simple backbone design and low resolution of object images, the quality of reconstructed faces is low. Additionally, as a result of using a multi-scale feature pyramid, SSDIR returns multiple image reconstructions for individual objects.

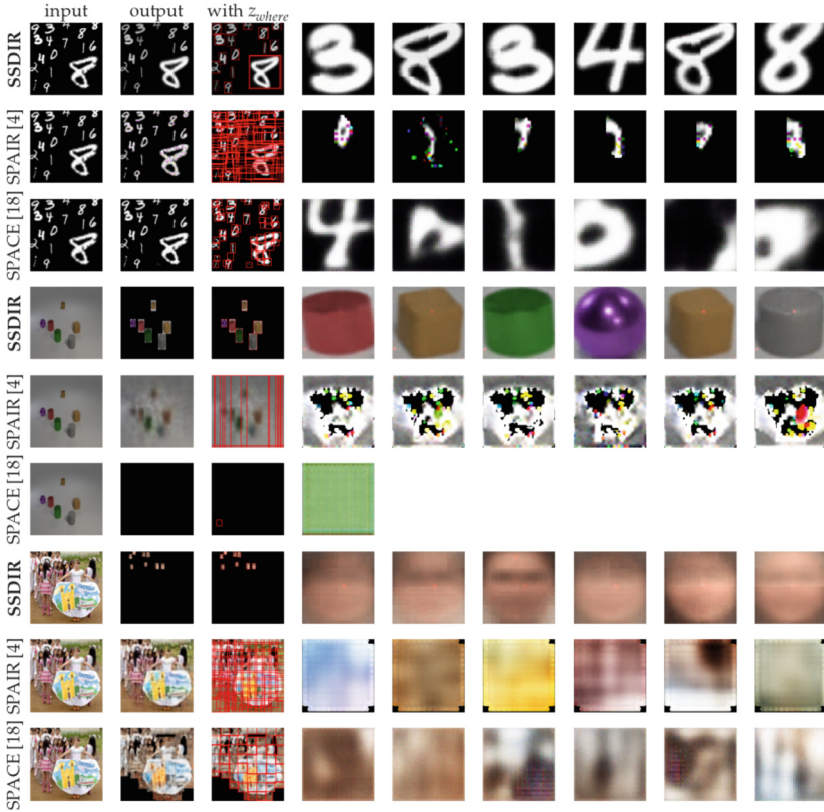


Fig. 2. Model inference comparison between **SSDIR**, SPAIR [4], and SPACE [18] for three typical images from each dataset. The first column presents the input image, the second and third contain image reconstruction without and with inferred bounding boxes; the remaining columns include some of the reconstructed individual objects. The number of images is limited due to the number of objects reconstructed by each model; for SSDIR, objects are meaningful and visually sound, while SPAIR and SPACE tend to divide bigger objects into smaller ones, or, in case of more complicated scenes, reconstruct them by dividing into rectangles, returning a redundant number of latents.

4.2 Latent Space

In this section, we present the analysis of the SSDIR model’s latent space and compare it with the latent space of SPAIR and SPACE. Figure 3 visualizes latent spaces for the scattered multi-scale MNIST dataset. For each model, we process the test subset to generate latent vectors of each image. Then, individual objects’ z_{where} vectors were compared with ground truth bounding boxes, and labels were assigned to latent representations by choosing the maximum intersection over union between predicted and true boxes. Each z_{what} vector was then embedded into two-dimensional space using t-SNE.

Table 2. Comparison of metrics for digit classification task using latent objects’ representations and logistic regression. Results are averaged over 3 random seeds.

Method	Accuracy	Precision	Recall	F1-Score
SSDIR	0.9789 \pm 0.0016	0.9787 \pm 0.0017	0.9786 \pm 0.0016	0.9786 \pm 0.0016
SPAIR [4]	0.1919 \pm 0.0073	0.1825 \pm 0.0087	0.2019 \pm 0.0092	0.1803 \pm 0.0102
SPACE [18]	0.2121 \pm 0.0432	0.2020 \pm 0.0431	0.2158 \pm 0.0435	0.1992 \pm 0.0462

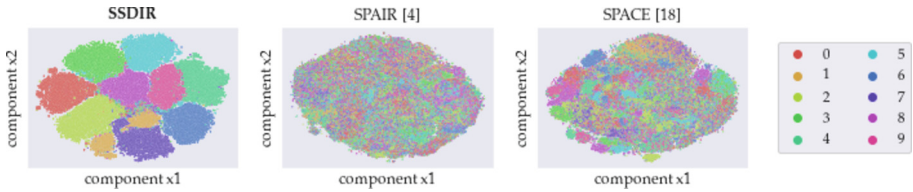


Fig. 3. Visualization of z_{what} latent space for scattered MNIST test dataset. Each object representation was converted using t-SNE to a two-dimensional space and plotted; the labels were inferred by choosing maximum intersection over union of predicted z_{where} and the ground truth bounding box and label. SSDIR shows a structured latent space, allowing easier distinguishing between digits.

Comparing the latent spaces, it is visible that SSDIR embeds the objects in a latent space, where digits can be easily distinguished. What is more, the manifold is continuous, without visible aberrations. The baseline methods’ latent spaces are continuous as well, but they do not allow easy discrimination between each object class. The main reason is probably the fact, that both SPAIR and SPACE tend to divide large objects into smaller parts, according to the preset object size, as shown in Sect. 4.1.

Next, we tried to use the latent representations of objects in images for a downstream task of digit classification. For each of the methods, we trained models on the scattered MNIST dataset using three random seeds and produced latent representations for both train and test subset, assigning labels to each object’s z_{what} based on intersection over union between z_{where} and ground truth boxes. Then, for each model and seed, we trained a logistic regression model to

classify the digits based on their latent representations. Test subset classification metrics are gathered in Table 2. SSDIR latent space proves to be more valuable than the baseline methods’, reaching high values of each metric.

4.3 Ablation Study

To test the influence of the dataset’s characteristics on the model performance, we performed an ablation study. The scattered MNIST dataset is generated by drawing random cells in a preset grid and inserting a random-sized MNIST digit inside it with a random offset. The number and size of grids, as well as the minimum and maximum size of a digit, are the hyperparameters of the dataset generation researched in the ablation study.

An SSDIR model was trained on each of the generated datasets and evaluated on a test subset with regard to the mean square error of reconstructions. The results of the study are shown in Fig. 4.

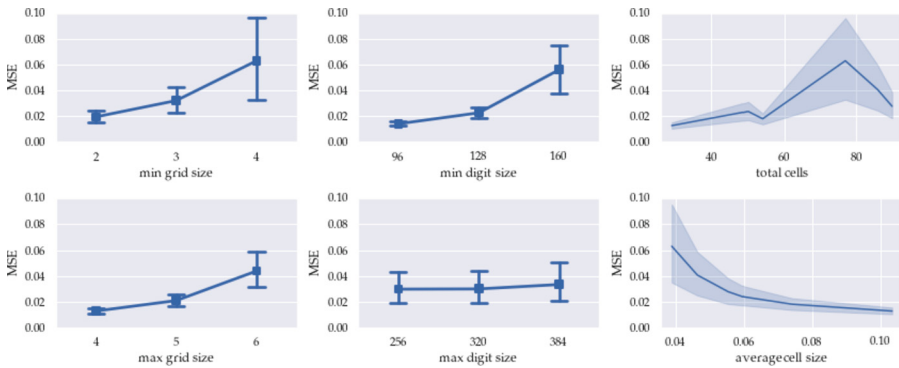


Fig. 4. Influence of the dataset generation parameters on the model performance. Parameters generating a dataset with larger or more occluded digits causes the model’s performance to mitigate. SSDIR works best for non-occluded, small digits.

It is visible, that the model is sensitive to the size of objects in images. Bigger objects cause the mean square error to rise, mainly due to the transformation of small-sized reconstructions to the output image. Another factor that causes the error to increase is the number of digits in the image, which usually leads more occlusions to appear in the final image. The upturn is visible with increasing the minimum and maximum grid size, as well as the total number of cells.

5 Conclusions

In this paper, we proposed SSDIR, a single-shot convolutional generative model for learning scale-invariant object representations, which enhances existing solutions with a multi-scale feature pyramid-based approach and knowledge learned

in an object detection model. We showed the improved quality of latent space inferred by SSDIR by applying it in a downstream task and proved its ability to learn scale-invariant representations of objects in simple and complex images.

Among the method’s drawbacks, one should mention limited input image size, which makes it struggle with very complicated scenes, especially in case of occlusions. What is more, learning representations of objects in complex scenes could be improved by more advanced modeling of objects’ interactions. These issues will be addressed in future works, which include applying a more advanced convolutional backbone and larger input images for improving the ability to detect objects and the quality of their representations. The latent vectors inferred by SSDIR could potentially be used in other advanced tasks, such as object tracking or re-identification. In such a case, the model could benefit from the increased sophistication of the model architecture. Additionally, SSDIR could be extended for processing videos by utilizing a recurrent network to consider temporal dependencies between subsequent frames.

References

1. Ali Eslami, S.M., et al.: Attend, infer, repeat: fast scene understanding with generative models. In: *Advances in Neural Information Processing Systems (Nips)*, pp. 3233–3241 (2016)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1798–1828 (2013). <https://doi.org/10.1109/TPAMI.2013.50>
3. Burgess, C.P., et al.: MONet: Unsupervised Scene Decomposition and Representation, pp. 1–22 (2019). <http://arxiv.org/abs/1901.11390>
4. Crawford, E., Pineau, J.: Spatially invariant unsupervised object detection with convolutional neural networks. In: *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 3412–3420 (2019)
5. Crawford, E., Pineau, J.: Exploiting spatial invariance for scalable unsupervised object tracking. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34(04), pp. 3684–3692, April 2020. <https://doi.org/10.1609/aaai.v34i04.5777>, <https://ojs.aaai.org/index.php/AAAI/article/view/5777>
6. Ehrhardt, S., et al.: RELATE: physically plausible multi-object scene synthesis using structured latent spaces. *NeurIPS* (2020)
7. Engelcke, M., Kosiosek, A.R., Jones, O.P., Posner, I.: Genesis: generative scene inference and sampling with object-centric latent representations. In: *International Conference on Learning Representations* (2020). <https://openreview.net/forum?id=BkxfaTVFwH>
8. Engelcke, M., Parker Jones, O., Posner, I.: GENESIS-V2: inferring unordered object representations without iterative refinement. *arXiv preprint arXiv:2104.09958* (2021)
9. Greff, K., et al.: Multi-object representation learning with iterative variational inference. In: *36th International Conference on Machine Learning, ICML 2019 2019-June*, pp. 4317–4343 (2019)

10. Greff, K., Van Steenkiste, S., Schmidhuber, J.: Neural expectation maximization. *Adv. Neural Inf. Process. Syst.* **2017**-Decem(Nips), 6692–6702 (2017)
11. Gu, C., Xie, H., Lu, X., Zhang, C.: CGMVAE: Coupling GMM prior and GMM estimator for unsupervised clustering and disentanglement. *IEEE Access* **9**, 65140–65149 (2021). <https://doi.org/10.1109/ACCESS.2021.3076073>
12. Henderson, P., Lampert, C.H.: Unsupervised object-centric video generation and decomposition in 3D. In: *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020)
13. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS 2015*, pp. 2017–2025. MIT Press, Cambridge (2015)
14. Jiang*, J., Janghorbani*, S., Melo, G.D., Ahn, S.: Scalor: generative world models with scalable object representations. In: *International Conference on Learning Representations* (2020). <https://openreview.net/forum?id=SJxrKgStDH>
15. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997 (2017). DOI: <https://doi.org/10.1109/CVPR.2017.215>
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings* (2014)
17. Kosiorek, A.R., Kim, H., Posner, I., Teh, Y.W.: Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems 2018-Decem (NeurIPS)*, pp. 8606–8616 (2018)
18. Lin, Z., et al.: Space: unsupervised object-oriented scene representation via spatial attention and decomposition. In: *International Conference on Learning Representations* (2020). <https://openreview.net/forum?id=rkl03ySYDH>
19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016. LNCS*, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
20. Niemeyer, M., Geiger, A.: Giraffe: representing scenes as compositional generative neural feature fields. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
21. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML 2014*, pp. II-1278-II-1286. JMLR.org (2014)
22. Van Steenkiste, S., Greff, K., Chang, M., Schmidhuber, J.: Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–15 (2018)
23. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: a face detection benchmark. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5525–5533 (2016). <https://doi.org/10.1109/CVPR.2016.596>
24. Zhu, W., Shen, Y., Yu, L., Sanchez, L.P.A.: Gmair: unsupervised object detection based on spatial attention and gaussian mixture (2021)