Unsupervised Representation Learning Via Information Compression

Zezhen Zeng, Jonathon Hare, and Adam Prügel-Bennett

University of Southampton, Southampton, United Kingdom
{zz8n17, jsh2, apb}@ecs.soton.ac.uk

Abstract. This paper explores a new paradigm for decomposing an image by seeking a compressed representation of the image through an information bottleneck. The compression is achieved iteratively by refining the reconstruction by adding patches that reduce the residual error. This is achieved by a network that is given the current residual errors and proposes bounding boxes that are down-sampled and passed to a variational auto-encoder (VAE). This acts as the bottleneck. The latent code is decoded by the VAE decoder and up-sampled to correct the reconstruction within the bounding box. The objective is to minimise the size of the latent codes of the VAE and the length of code needed to transmit the residual error. The iterations end when the size of the latent code exceeds the reduction in transmitting the residual error. We show that a very simple implementation is capable of finding meaningful bounding boxes and using those bounding boxes for downstream applications. We compare our model with other unsupervised object discovery models.

Keywords: Unsupervised representation learning · VAE · Object discovery · Information bottleneck

1 Introduction

In the last few years there has been a significant research effort in developing unsupervised techniques in deep learning. A very prominent example of these methods is the variational auto-encoder (VAE) [12, 16] that attempts to find latent representations to efficiently encode a dataset. A drawback of VAEs is that they represent the entire image. This is unlikely to lead to an efficient representation for many real-world images that depict multiple objects. Following the development of VAEs there have been a number of attempts to use unsupervised techniques for object location and segmentation within an image [1, 3, 4, 9, 13, 14].

In this paper we will explore the use of a minimum description length cost function together with an information bottleneck to achieve unsupervised image understanding. The evidence lower bound (ELBO) of a VAE can be interpreted as a description length where the KL-divergence corresponds to a code length of the latent representation and the log-probability of the reconstruction error as the code length of the residual error (i.e. the error between the reconstruction and the original image). In our approach we will use multiple glimpses of an image

corresponding to a sequence of bounding boxes. These are resized to 8×8 patches and passed to a variational auto-encoder. The full reconstruction is built up from adding together the reconstructions from the VAE. This is done iteratively with each patch providing a correction between the current reconstruction and the true image. A spatial transformer is fed the current residual error and used to select the next bounding box. The overall cost function is the cost of the latent codes for all the bounding boxes together with the cost of the final residual error. We stop when the cost of transmitting the latent code is higher the reduction in the cost of transmitting the residual error. The spatial transformer and VAE is trained end-to-end by minimising the description cost of the images in a dataset.

Although we expect our approach to be very different to human eye movement nevertheless, there is a rough correspondence due to the restricted size of the fovea requiring multiple fixations of an image around areas of high interest and possible interpretational ambiguity [18]. We deliberately avoid building in any bias towards glimpsing complete objects, however, as we will see later, at least, in simple scenes this behaviour emerges. Our aim is not to build a state-of-the-art unsupervised object detector, but rather to investigate how a minimal implementation using minimum description length and an information bottleneck will glimpse images. As we will demonstrate these glimpses can sometimes be used to solve downstream tasks that have competitive results with much more sophisticated approaches.

2 Related Work

There are several works on unsupervised object-centric representation learning and scene decomposition. MONet [1] applies a recurrent attention mechanism that produces deterministic soft masks for each component in the scene. They also use a VAE following their attention mechanism. Genesis [4] is similar to MONet and employs an RNN network after the encoder to infer the mask of objects and then uses another VAE to infer the object representations. Unlike MONet, all the modules in Genesis can be processed in parallel. IODINE [8] employs an amortized iterative refinement mechanism for the latent code, which is computationally expensive. Genesis-V2 [3] is the upgraded version of Genesis which replaces the RNN network with a semi-convolution embedding method. Other scene-mixture methods based on self-attention mechanism can also perform image decomposition and object representation learning [14, 19].

The Attend-Infer-Repeat (AIR) [5] and the following work SPAIR[2] and SuPAIR [17] infer the object representation as "what", "where" and "pres" variables, where SPAIR infer an additional variable "depth". The "what" variable represents the shape and appearance of objects, the "where" contains the position and scale of objects and the "pres" variable is slightly different in the two models. In AIR, "pres" is a unary code which is formed of 1 and 0, where 0 means the termination of inference, while in SPAIR, "pres" is a binary variable that can be sampled from a Bernoulli distribution, where 0 represents no object in the corresponding cell. The image is encoded as a feature map which can be the same size as the original image or smaller size. Each cell in the feature map is processed with the nearby cells that have already been processed before.

Thus, the whole process is sequential. However, such a sequential operation is time-consuming. The SPACE network [13] discards the nearby cells and processes all the cells fully parallel. The authors also add another network to infer the background components.

PermaKey [7] is a model that aims to extract object keypoints from images that take the error map between two feature maps as input. While our model takes the error map between two images as input directly to inference the position of objects.

3 Model

In this section we introduce our model in details.

3.1 Glimpsing Network

In our approach we iteratively build up a reconstruction. We use a glimpsing network that consists of a spatial transformer network that proposes the location of a bounding box and then resamples the image within that bounding box to create (in our case) a low-resolution patch of the original image. In our network at each iteration, the spatial transformer network is given the residual error between the current reconstruction and the input image. The glimpsing network selects a bounding box. Then the residual error, $\boldsymbol{\Delta}(t) = \boldsymbol{x} - \hat{\boldsymbol{x}}(t)$, within the bounding box is down-sampled to an 8×8 patch (with 3 colour channels) and fed to a VAE. The VAE produces a latent code $q(\boldsymbol{z}|\boldsymbol{\Delta}(t))$. This is used to create a reconstruction using the standard reparameterisation trick, which is then resized to the size of the original bounding box. This results in a reconstructed correction, $\hat{\boldsymbol{\Delta}}(t)$, which is then added to reconstruction to obtain an new reconstruction $\hat{\boldsymbol{x}}(t+1) = \hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{\Delta}}(t)$ (note that $\hat{\boldsymbol{\Delta}}(t)$ only has non-zero values within the bounding box selected by the glimpsing network).

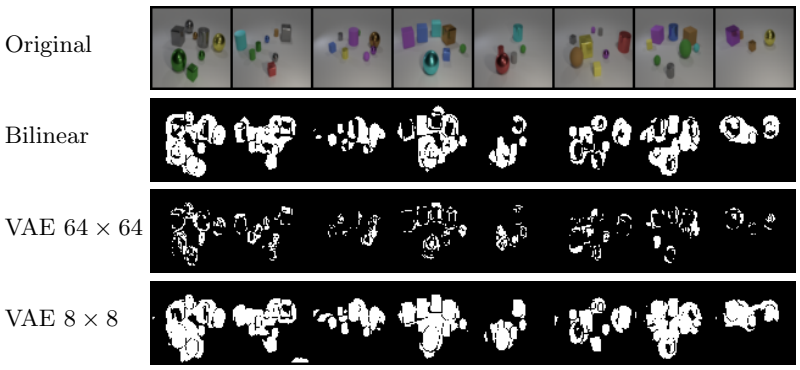

Fig. 1: The error maps of different compression methods. The first row is the original image, the next three rows are the error maps of bilinear interpolation, a VAE for 64×64 image and a VAE for 8×8 image which takes the downsampled version of the original images as input

We use the standard information theoretic result that the cost of transmitting a random variable with a distribution $q(\boldsymbol{z}|\boldsymbol{\Delta}(t))$ relative to a distribution $p(\boldsymbol{z})$ is given by the KL-divergence (or relative entropy) $D_{KL}(q\|p)$. In our case we use the standard latent encoding $q(\boldsymbol{z}|\boldsymbol{\Delta}(t)) = \mathcal{N}\big(\boldsymbol{z}|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma})\big)$ and standard prior $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$. The cost of communicating the residual error is given by $-\log\big(p(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)$. We use the standard assumption that the residual errors are independent at each pixel and colour channel and normally distributed with mean 0 and standard deviation σ. To minimise the communication cost we choose σ^2 to be the empirical variance. In this case the cost of communicating the residual errors is, up to a constant, equal to $N\log(\sigma)$, where N is the number of pixels times the number of colour channels.

Note that provided both the sender and receiver have the same VAE decoder we can communicate an image by sending the set of latent codes (plus the position of the bounding box) and the residual error. (We assume that the dataset we are sending is so large that the cost of transmitting the VAE decoder is negligible). To train our spatial transformer and VAE we attempt to minimise this communication cost for a dataset of images.

A critical component of our approach is that we use an information bottleneck. That is, we down-sample our bounding box and feed this to a VAE. We illustrate the effect of this for images taken from the CLEVR [10] dataset in Figure 1. In the first row we show the original images. In the second row we show the reconstruction error after down-sampling the whole image to an 8×8 image and then up-sampling using bilinear interpolation to the original size (64×64). In the third row we show the reconstruction loss if we use a vanilla VAE without down-sampling. Finally, we show the reconstruction loss when we down-sample to 8×8 encode that through a VAE and then up-sample the VAE reconstruction.

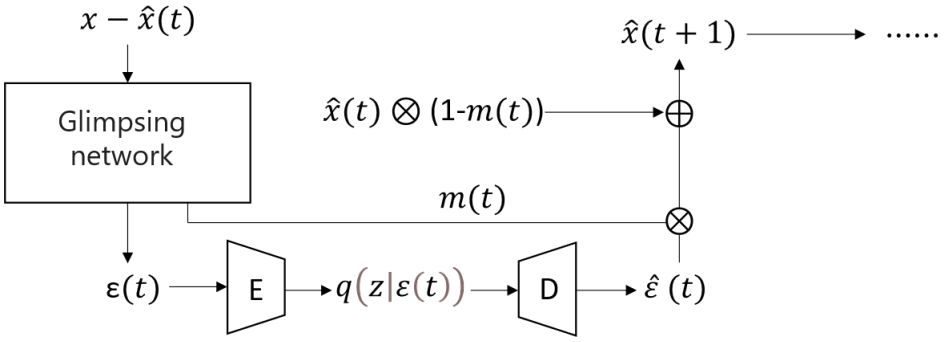

Fig. 2: The diagram of one iteration in SAID, where we omit the operation from position code $\boldsymbol{u}$ to $\boldsymbol{\epsilon}(t)$.

Figure 1 illustrates that due to the bottleneck we have high reconstruction error around the objects in the image. This error will drive the glimpse model towards parts of the scene of high complexity.

3.2 Model Architecture and Training

In keeping with our philosophy or keeping our model simple. Our glimpsing network consists of 2 layers of CNN and 2 layers of MLP. Each image is presented K times. At each presentation the input to the glimpsing network is the residual error, $\boldsymbol{\Delta}(t) = \boldsymbol{x} - \hat{\boldsymbol{x}}(t)$. The output of the glimpsing network provides the coordinates of the bounding box that is then resampled to create an 8×8 RGB image patch, $\boldsymbol{\epsilon}(t)$ that is used as the input to a standard VAE. In Figure 2, we illustrate the structure of one iteration in our model. In the experiment, $\boldsymbol{\epsilon}(t)$ is the original image within the bounding box rather than the residual error (which is better for the downstream tasks). This also makes $\hat{\boldsymbol{x}}(t+1) = (1-\boldsymbol{m}(t)) \otimes \hat{\boldsymbol{x}}(t) + \boldsymbol{m}(t) \otimes \hat{\boldsymbol{\epsilon}}(t)$. where $\boldsymbol{m}(t)$ is a mask equal to 1 in the bounding box and 0 otherwise that obtained from the position code, and $\otimes$ denotes elementwise multiplication. Additionally, $\boldsymbol{m}(t)$ can be an alpha mask produced by the decoder of the VAE, the importance of the alpha mask will be investigated in the ablation study. The VAE reconstruction, $\hat{\boldsymbol{\epsilon}}(t)$, is reshaped to the original bounding box to create a correction to the reconstruction. Both encoder and decoder contain 4 layers of CNN with ReLU and 2 layers of MLP. We used a 10-dimensional latent representation. To train the VAE we minimise the standard ELBO loss function

$$\mathcal{L}_{vae} = -\log(p(\boldsymbol{\epsilon}(t)|\hat{\boldsymbol{\epsilon}}(t))) + D_{KL}\left(q(\boldsymbol{z}(t)|\boldsymbol{\epsilon}(t))\middle\|\mathcal{N}(\boldsymbol{x}|\mathbf{0},\mathbf{I})\right) \tag{1}$$

where $q(\boldsymbol{z}(t)|\boldsymbol{\epsilon}(t))$ is the distribution describing the latent. Note that the reconstruction $\hat{\boldsymbol{\epsilon}}$ is generated by sampling a latent vector from $q(\boldsymbol{z}(t)|\boldsymbol{\epsilon}(t))$ and feeding this to the decoder of the VAE.

Recall the glimpsing network predicts the position of the bounding box. We encode this through a position parameter $\boldsymbol{\mu} = (\mu_x, \mu_y)$ and width parameters $\boldsymbol{\sigma} = (\sigma_x, \sigma_y)$. Adding a bounding box at iteration t reduces the cost of communicating the residual error by

$$\mathcal{L}_{res} = -\log(p(\boldsymbol{\Delta}(t)) + \log(p(\boldsymbol{\Delta}(t-1)) \tag{2}$$

but requires an additional cost

$$\mathcal{L}_{kl} = D_{KL}\left(q(\boldsymbol{z}(t)|\boldsymbol{\epsilon}(t))\middle\|\mathcal{N}(\boldsymbol{x}|\mathbf{0},\mathbf{I})\right). \tag{3}$$

Summing these two terms provide a loss function for the network that measures (up to an additive constant) the reduction in cost of communicating the image using the new latent code describing the correction to the residual error. Note that in using the trained network when this difference becomes positive then we stop glimpsing. In practice, training the glimpsing network with just these term leads to poor performance. To improve this we assume the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ represent parameters of a normal distribution where we regularise them with an additional loss term

$$\mathcal{L}_{pos} = D_{KL}\left(\mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu},\boldsymbol{\sigma})\middle\|p(\boldsymbol{u})\right). \tag{4}$$

This acts as a regularisation term for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. We call the whole network *Spatial Attention for Information Detection* (SAID).

We also considered two modified network architectures. In the first we learn an alpha channel so that the corrections are only applied to particular regions within the bounding box. In the second case we include an additional channel as the input, which we called *scope*. This is motivated by MONet [1] that uses the same idea. This scope channel can force the network to look at the area that has not been discovered even when there are areas that have been discovered before contain high error pixels. Initially we set $\boldsymbol{s}(0) = \mathbf{1}$. Recall that the mask, $\boldsymbol{m}(t)$, is defined to be 1 in the bounding box and 0 elsewhere. Then the scope is updated as

$$\boldsymbol{s}(t+1) = \boldsymbol{s}(t) \otimes (\mathbf{1} - \boldsymbol{m}(t)) \tag{5}$$

so the scope will be 0 where a bounding box has been proposed and 1 otherwise. Note that in our approach we learn after every iteration rather than build up a gradient over multiple iterations. This makes the learning problem for our system much simpler than methods such as MONet that applies the loss function only after making a series of bounding box proposals. We investigate the role of the alpha channel and scope in ablations studies described in Section 4.4.

4 Experiments

In this section we attempt to quantify the performance. Recall that the objective is to find glimpses that allows an image to be efficiently encoded through a bottleneck, so it will not necessarily find glimpses that correspond to objects. However, as we will show this is an emergent property of the network, at least for simple scenes. We therefore compare our model to two models, SPACE [13] and SPAIR [2] designed to find multiple objects in an image. To evaluate our model, we use three commonly used datasets in unsupervised object-centric representation learning models. The first dataset is Multi-dSprites, which is developed from dSprites [15] and each image consists of different shapes with different colour, the maximum number of objects in this dataset is 4. The second dataset is Multi-MNIST. For Multi-MNIST, we render MNIST digits of size 20×20 on a 84×84 canvas. The maximum number of objects in this dataset is 5. The last dataset is CLEVR [10]. For all the datasets we resize images into 64×64.

We trained our network, SAID, using the ADAM [11] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train our model for 200 epochs for all the datasets.

4.1 Quantitative Comparison

As a first test we consider object location on the Multi-MNIST dataset. Although object location is not an objective of our model, for the Multi-MNIST dataset the hand-written characters are well separated and so a natural choice of bounding box would be around each character. There are however situations where the bounding box only covers part of the objects or covers more than one object, as shown in Figure 5. We perform a quantitative comparison of results on Multi-MNIST using two metrics with SPACE and SPAIR. The two metrics are the Average Precision (AP) and the Object Count Error. AP is a commonly used metric in object detection task [6] and the Object Count Error is the difference

Table 1: Comparison with respect to the quality of the bounding boxes in the Multi-MNIST. Results are averaged over 5 random seeds.

	AP	Object
	IoU Threshold $\in$ [0.1:0.1:0.9]	Count Error
SPAIR	0.501 $\pm$ 0.004	0.261 $\pm$ 0.032
SPACE	0.310 $\pm$ 0.041	0.031 $\pm$ 0.005
SAID	0.503 $\pm$ 0.005	0.810 $\pm$ 0.040

between the number of objects predicted by the models and the true number of digits [2]. For SPACE and SPAIR, we set the grid size as 3×3. For our model, we use the index of the iteration that the KL term of the VAE is smaller than improvements on the mean squared error as the number of objects, and we set $K = 5$ in training and $K = 9$ in AP measurement.

As shown in table 1, our model achieves similar AP with SPAIR, SPACE has the worst AP. However, this result on Multi-MNIST does not reflect the ability of object detection. The reason is the ground truth bounding box we are using for MNIST in the AP calculation is larger than the digits, which degrades the AP result when the model returns a smaller bounding box while it still detects the objects well. In Figure 3, the first row is the ground truth bounding box we can obtain in Multi-MNIST dataset, this brings disadvantages to the SPACE model in the AP calculation since the third row shows the bounding box of SPACE model is tighter than the ground truth and still maintain the accuracy. Our model does not perform well on the Object Count Error, since there is no $\mathbf{z_{pres}}$ in our model and the objective of our model is to find high error areas rather than find objects. Objects of high complexity are often selected more than once leading to a count error. We note that our stop criteria is applicable to any image and was not chosen to given an accurate object counts.

4.2 Downstream Task

Obviously a glimpsing model is only of value if the glimpses can be used in some downstream task. Here we consider the task of returning the sum of all the digits in a Multi-MNIST image. Each image contains 5 digits. This is a task that has previously been used to test unsupervised multi-object detection. We show the results on 80k training set and 20k test set. We compare our results to the SPACE and SPAIR models. We run implementation of all three models to ensure consistency.

For SPACE and our model, we use the same architecture of the encoder, and the channel of the latent space is 10. For SPAIR, we observered that the model tended to collapse at an early stage when we using these parameter setting. Thus, we maintain the original architecture of the encoder, but increase the channel of the latent space to 50 and the input size of the encoder is 15×15 rather than 8×8, which potentially brings benefits to the capacity of the encoder.

To compute the sum of the digits we construct a 3 layer MLP using the latent codes, $\boldsymbol{z}(t)$, as inputs. The output of the MLP has a single output which

Fig. 3: Qualitative comparison between the bounding boxes found for different models. GT is the ground truth bounding boxes, SPAIR and SPACE are models developed by other authors while SAID is our model.

we train to have the same numerical value as the digit (that is, the digit 3 should have an output of 3). When testing we add the outputs from each glimpse and then round that number to the nearest integer. We trained the MLP for 100 epochs. This is considerably simpler than the set up described in [2] who use an LSTM to perform the addition. We note that our method explicitly treats the glimpse for this problem as a set, where the result is invariant to the ordering of the glimpses. We also demonstrate the results by feeding the ground truth bounding box into the encoder rather than the predicted bounding box, which we note as GT. GT provides an estimated upper-bound on the performance we could achieve.

Table 2 shows the results of four models in 2 different conditions. Fixed represents a frozen encoder during the classifier training while Unfixed represents an encoder tuning with the classifier. Due to the architecture issue, SPAIR performs best under Fixed but worst under Unfixed. Our model performs better than SPACE in both situations but there is still a huge gap between our model and the ground truth bounding box.

4.3 Generalization

Our model uses a very general principle that we believe can be widely used in different contexts. To explore this we look at out-of-distribution generalisation. That is, when we train on one dataset (here we use CLEVR) and use the model on a different dataset. We test the network on the Multi-dSprites and Multi-

Table 2: The performance on the downstream task of summing the digits in the images in Multi-MNIST is shown using the ground truth (GT) bounding boxes and bounding boxes found by SPAIR, SPACE and our network SAID. The results are computed by averaging over 5 runs with different random seeds.

	Fixed		Unfixed	
	Train	Test	Train	Test
GT	30.3% ± 1.2%	29.2% ± 1.1%	97.5% ± 0.9%	92.3% ± 1.1%
SPAIR	25.8% ± 2.2%	24.0% ± 2.1%	24.6% ± 1.5%	22.0% ± 1.1%
SPACE	15.1% ± 2.5%	14.4% ± 2.2%	42.3% ± 1.5%	30.1% ± 1.2%
SAID	22.3% ± 1.8%	21.3% ± 2.0%	57.8% ± 1.6%	31.9% ± 1.5%

MNIST datasets. We set the maximum number of iteration $K = 10$ which the same as we used when training CLEVR, but we stop the iteration when the KL is larger than the reduction in code length of the reconstruction error. Results are shown in Figure 4. The first row is the result for Multi-dSprites, the model trained on CLEVR can stop at reasonable iteration. But the model tends to infer more times on Multi-MNIST, we assume it is because the binary images are simpler to be transmitted than the RGB images, the VAE trained on CLEVR can transmit binary images efficiently no matter if the bounding box covers the digits correctly. The model can still locate the area of objects although some of the bounding boxes failed at covering one object.

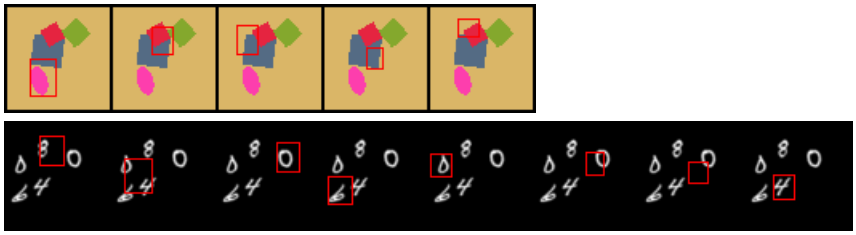

Fig. 4: Examples of bounding boxes found on the Multi-dSprites and Multi-MNIST dataset are shown for a network trained of the CLEVR dataset.

4.4 Ablation Study

The results we show in the previous section is trained with a scope channel and the input is the difference between the original image and a lossy image that has been downsampled to 8×8 and then upsampled to the original size through bilinear interpolation. Also, we have used an alpha mask instead of a binary mask when we blend the images. In this section, we show the importance of the scope channel and the alpha channel. We also show how the lossy image can affect the results when we use different methods to get the lossy version.

Table 3 shows the results on AP and Object Count Error, no alpha and no scope mean we remove the alpha channel and scope channel respectively. VAE8 means we use a 8×8 VAE to reconstruct the image after the downsample interpolation, as shown in Figure 1. It can be observed that the alpha channel does not make a huge difference to the model while the model gets a degeneration performance when we remove the scope channel. Also, the model performs worse when we use a VAE to obtain the lossy version of images. This is because after using a 8×8 VAE, the error map tends to cover more background.

Table 3: The performance of ablation studies carried out on the Multi-MNIST dataset. The average precision in the detection of bounding boxes is presented together with the error in the count of the numberof objects. Different versions of SAID are compared.

	AP IoU Threshold ∈ [0.1:0.1:0.9]	Object Count Error
SAID (no alpha)	0.490 ± 0.008	0.731 ± 0.040
SAID (no scope)	0.341 ± 0.006	1.710 ± 0.110
SAID (VAE8)	0.452 ± 0.008	1.101 ± 0.031
SAID	0.503 ± 0.005	0.810 ± 0.040

4.5 CLEVR and Multi-dSprites

In the Multi-MNIST the objects are of approximately the same size and do not suffer from occlusion. Clearly, this is very different to real images. To explore these issues we tested our models on CLEVR and Multi-dSprites dataset.

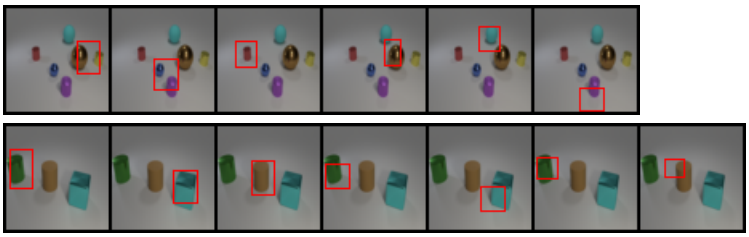

Fig. 5: Examples of bounding boxes found by SAID on the CLEVR dataset.

Figure 5 shows the results on CLEVR dataset and Figure 6 shows the results on Multi-dSprites dataset, we set $K = 10$ for CLEVR and $K = 4$ for Multi-dSprites respectively. We stop the iteration when the KL divergences is greater than the reduction in transmitting the residual error. In Figure 5, the first row, the size of objects is close to 8×8, the model can stop at the correct iteration and all the bounding boxes covers different objects, although the bounding boxes are

less accurate. For the last row, the size of objects is much larger than 8×8. Our model tends to infer more than the number of objects, this is due to the limited bottleneck failing at transmitting the whole object at the first transmission. But for those big objects, the model returns more accurate bounding boxes compared to the first two rows. Since big objects tend to show a big error. In Figure 6, our model does not stop after in a reasonable number of iterations. It has the same issue as CLEVR dataset that the bounding box tends to cover parts of shapes rather than the whole object. Also, our model cannot deal with overlap properly. In part we attribute this failure to the weakness of the attention network which struggles with finding bounding boxes of very different sizes.

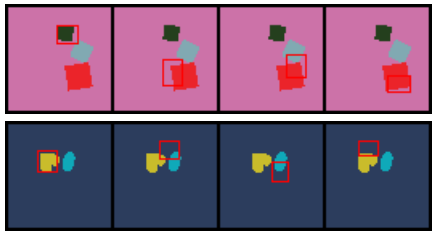

Fig. 6: Examples of bounding boxes found on the Multi-dSprites dataset.

5 Conclusion

Information compression provides a powerful tool for recognising structure in complex data sources. In this paper we have combined this with an information bottleneck to produce a glimpsing network that encodes images through a series of glimpses. By feeding the network the current residual error we can generate a series of bounding box proposals around parts of the image with high uncertainty in its reconstruction. We combine this with a VAE that can learn the common structures within an image (e.g. objects, or rather typical residual errors associated with objects). As the bounding boxes are rescaled, the structures being learned by the VAE are translation and scale invariant. We have shown that following these principles it is possible to train a very simple network that has comparable performance on object detection tasks to much more complex networks designed for multi-object detection. Our objective was to test as simple a network as possible to prove the power of this learning paradigm.

References

1. Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390 (2019)
2. Crawford, E., Pineau, J.: Spatially invariant unsupervised object detection with convolutional neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 3412–3420 (2019)

3. Engelcke, M., Jones, O.P., Posner, I.: Genesis-v2: Inferring unordered object representations without iterative refinement. arXiv preprint arXiv:2104.09958 (2021)
4. Engelcke, M., Kosiorek, 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)
5. Eslami, S., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., Hinton, G.E.: Attend, infer, repeat: Fast scene understanding with generative models. In: Advances in Neural Information Processing Systems (2016)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
7. Gopalakrishnan, A., van Steenkiste, S., Schmidhuber, J.: Unsupervised object keypoint learning using local spatial predictability. In: International Conference on Learning Representations (2021)
8. Greff, K., Kaufman, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., Lerchner, A.: Multi-object representation learning with iterative variational inference. In: International Conference on Machine Learning. pp. 2424–2433. PMLR (2019)
9. Greff, K., Van Steenkiste, S., Schmidhuber, J.: Neural expectation maximization. In: Advances in Neural Information Processing Systems (2017)
10. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2901–2910 (2017)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of International Conference on Learning Representations (2015)
12. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of International Conference on Learning Representations. (2013)
13. Lin, Z., Wu, Y.F., Peri, S.V., Sun, W., Singh, G., Deng, F., Jiang, J., Ahn, S.: Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In: Proceedings of International Conference on Learning Representations. (2020)
14. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: Advances in Neural Information Processing Systems (2020)
15. Matthey, L., Higgins, I., Hassabis, D., Lerchner, A.: dsprites: Disentanglement testing sprites dataset (2017)
16. 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 (2014)
17. Stelzner, K., Peharz, R., Kersting, K.: Faster attend-infer-repeat with tractable probabilistic models. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5966–5975. PMLR (09–15 Jun 2019)
18. Stewart, E.E., Valsecchi, M., Schütz, A.C.: A review of interactions between peripheral and foveal vision. Journal of vision **20**(12), 2–2 (2020)
19. Van Steenkiste, S., Kurach, K., Schmidhuber, J., Gelly, S.: Investigating object compositionality in generative adversarial networks. Neural Networks **130**, 309–325 (2020)