Supplementary material for iCaps: An Interpretable Classifier via Disentangled Capsule Networks

Dahuin Jung, Jonghyun Lee, Jihun Yi, and Sungroh Yoon

Electrical and Computer Engineering,
ASRI, INMC, and Institute of Engineering Research
Seoul National University, Seoul 08826, South Korea

S1 Experimental Details

Our code uses TensorFlow and Keras libraries. For the underlying DeepCaps [10] code, we used the official Keras code uploaded by the authors of the original paper. All experiments were implemented on one Titan V with a batch size of 64.

We resized the images as 64 for MNIST, SVHN, and CelebA datasets for experimental convenience. We obtained the reported results within epochs 10–20 for the MNIST dataset, and epochs 20–30 for the SVHN and CelebA datasets. For λ_M, λ_{recon}, λ_G, λ_{D_G}, λ_{LGP}, and λ_{KL}, we set the values to 100, 100, 1, 1, 10, and 0.0001, respectively, for all the three datasets. For λ_{CS}, λ_{concept}, and λ_{CR}, we increased the value from 3 to 5 at epoch 10, from 1 to 10 in the first ten epochs, and from 0 to 1 at epoch 1, respectively, for the three datasets. For λ_{C_G}, we set the value to 1 in the case of CelebA and 1 for the original and 10 for swapping in the case of MNIST and SVHN. For λ_{RS}, we set the value to 1 for CelebA and 0.001 for MNIST and SVHN. In addition, for $\mathcal{L}_{\text{recon}}$, the structural similarity index [6] value was clearer than the mean squared error in the cases of SVHN and CelebA datasets. Lastly, for batches 200-400 just before the end of the training, we set λ_{concept} as 0 to allow images to have concept values, regardless of class labels.

We ran the code three times with the same hyperparameter setting to determine whether the same concepts consistently appear. Several concepts appeared to be a little unclear; however, we saw consistent concepts in general.

In experiments, we investigate whether our method performs on par with non-interpretable (ResNet-18) and base (DeepCaps) methods. We confirmed that iCaps provide a prediction along with proper rationales behind it with no performance degradation. Architecture details are given in Table. S1 and Fig. S1. The architectures for MNIST, SVHN, and CelebA datasets are the same. One difference is that for MMIST and SVHN, the capsule dimension of C_C remains at 4 till the final layer. We measured the number of model parameters needed for each methods as shown in Table. S2. The number of parameters of our framework is 2-3 times more than that of ResNet-18 and DeepCaps. However, we think that this is not a big increase because our framework can additionally offer an explanation behind its own prediction.

Table S1. Architectures of the proposed method except for C_C. Stride = 2. H = height, W = width, and C = channel.

G	D_G and C_G	E	D_{CR}
Input: $\mathbb{R}^{c+r}$	Input: H x W x C	Input: H x W x C	Input: H x W x 2C
FC, 4 x 4 x 512 BN, ReLU	5 x 5 conv, 64 LReLU	5 x 5 conv, 64, LReLU	5 x 5 conv, 64, LReLU
5 x 5 deconv, 256 BN, ReLU	5 x 5 conv, 128, LReLU	5 x 5 conv, 128, LReLU	5 x 5 conv, 128, LReLU
5 x 5 deconv, 128 BN, ReLU	5 x 5 conv, 256, LReLU	5 x 5 conv, 256, LReLU	5 x 5 conv, 256, LReLU
5 x 5 deconv, 64 BN, ReLU	5 x 5 conv, 512, LReLU	5 x 5 conv, 512, LReLU	5 x 5 conv, 512, LReLU
5 x 5 deconv, 3 Tanh	FC, 1 / FC, $\mathbb{R}^y$	FC, 32 / FC, 32	FC, $\mathbb{R}^c$

Table S2. Number of model parameters for the datasets

Architecture	# of Parameters		
	MNIST	SVHN	CelebA
ResNet-18	11M	11M	11M
DeepCaps	12M	12M	12M
Ours	24M	24M	38M

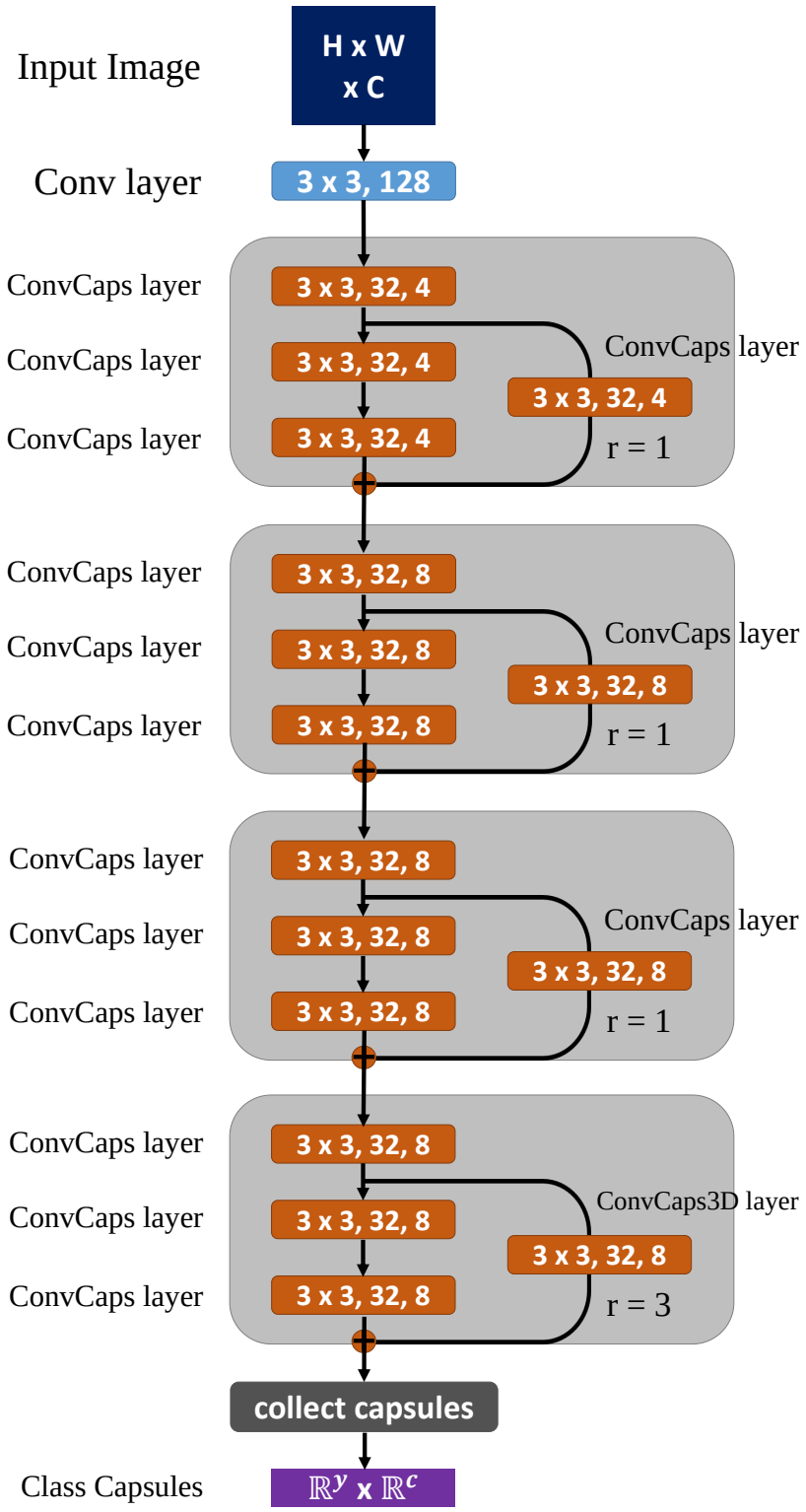


Fig. S1. Architecture of C_C [10]. For first convcaps layers in every block, the stride is 1. Otherwise, the stride is 2.

S2 Ablation Studies

We additionally confirmed a suitable performance improvement in the process of adding components one by one on top of DeepCaps (C_C and G) on the CelebA dataset. As shown in Table. S3, each component encourages proper performance increase which we expected.

First, the FID score [5] indicates the quality of the generated image, and when the value is smaller, it is better. MI refers to mutual information, specifically, I(z; y) in Table. S3(2) and I(c; y) in Table. S3(3, 4, and 5). We also measured the importance of D_{CR} using Mutual Information Gap (MIG) which is a quantitative metric suggested in the β-TCVAE [2] paper. A large MIG represents that a ground-truth factor has high mutual information (MI) with only one element.

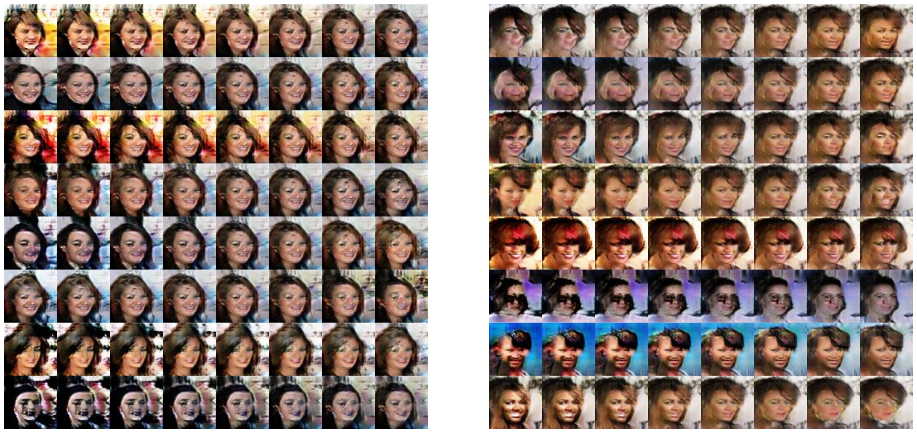
As mentioned in Table. 1 in the main manuscript, DeepCaps (Table. S3(1)) already shows high accuracy. Our method aims at providing explanations about its performance with maintained accuracy. For it, four more components are added. In turn, we added a discriminator, D_G, on the reconstruction stage, such as generative adversarial networks. Improving reconstructed image quality was essential for disentanglement. Through D_G, the FID score is improved from 101.53 to 87.00 as shown in Table. S3. Next, we added E to divide the features as class-relevant and -irrelevant. However, to completely achieve it, C_G is needed. When E and C_G are added in turn, MI is increased from 0.02 to 0.08 and from 0.08 and 0.56. As last, we added D_{CR} for distinctness. We qualitatively showed that the semantic overlap between the elements of the class capsule is reduced by D_{CR}. In our experiments, the MIG value is increased from 0.03 to 0.2 when D_{CR} is added. That is, we quantitatively confirmed that D_{CR} helps less semantic overlap between the elements. In addition, the qualitative result is shown in Fig. S2.

Table S3. Results of ablation studies. FID = Frechet Inception Distance [5] (lower is better), MI = Mutual Information (higher is better), and MIG = Mutual Information Gap [2] (higher is better).

	Architecture	FID↓	MI↑	MIG↑
(1)	C_C and G	**101.53**	-	-
(2)	C_C, G, and D_G	**87.00**	**0.02**	-
(3)	C_C, G, D_G, and E	87.02	**0.08**	-
(4)	C_C, G, D_G, E, and C_G	86.83	**0.56**	**0.03**
(5)	C_C, G, D_G, E, C_G, and D_{CR}	86.72	0.56	**0.20**

For the MNIST and SVHN datasets, the importance of D_{CR} was very clear and easy to identify. This is because MNIST and SVHN do not have many factors of variations that should be considered when classifying an input. Therefore, the concepts are prone to overlap when we do not enforce distinctness.

CelebA tended to overlap less, however, D_{CR} was still important. As shown in Fig. S2(a), the changes in $c1$, $c3$, $c4$, and $c5$ are very similar without D_{CR}. However, in (b), the change in each element is clearly distinct from each other.



(a) w/o D_{CR} (b) w/ D_{CR}

Fig. S2. Ablation study of D_{CR} (CelebA). For (a), changes in rows overlap ($c0$-$c7$), whereas changes in rows are distinct in (b).

S3 Comparison with SENN

We tested the performance of SENN [9] on the CelebA dataset and compared with ours. SENN also aims to provide similar types of explanations as we provide. For SENN, the test accuracy is 96.90, which is similar to ours and non-interpretable classifiers. However, the learned concepts of SENN do not provide a proper explanation about their classifications as shown in Fig. S3. It is hard to recognize the learned concepts of any dimension by analyzing the top and bottom 7 prototypes. In the case of $c1$, $c3$, $c4$, and $c6$, all images have equal value, so that we could not obtain prototypes. We chose dimension 8 for a fair comparison with our method.

In the case of our method, most concepts can be induced by analyzing the top and bottom 7 prototypes as shown in Fig. S4. However, for $c6$, we had to check more images having high or low values in order to obtain the learned concept. The reason that the images which do not fit the concept are selected as prototypes is that disentanglement is based on the reconstructed image, not the real image. As given in Fig. S4(b), the reconstructed image seems to have a beard. This makes inappropriate images be selected as prototypes. To overcome this problem, more clear and accurate reconstruction is important. For it, we still work on its performance.

Fig. S3. Top and bottom 7 prototypes of SENN [9] (Female vs. Male). It is difficult to recognize what concept each dimension represents.

Fig. S4. (a): Top and bottom 7 prototypes of our method (Female vs. Male). To some extent, we can recognize the learned concepts by only analyzing the given prototypes. (b): Reconstructed images of the bottom 7 of the beard ($c6$) concept. The reconstructed images seem to have a beard. In the case of $c6$, we had to analyze more images having values closer to 1 or -1 in order to get a clear concept.

S4 Replacement of CapsNet C_C

We conducted an experiment by replacing C_C with ResNet-18. All settings are the same with original experiments except for margin loss (L_M) because of their structural difference. In this experiment, we checked that the represented concepts are ambiguous, or different concept per class is detected in one element so that we can't obtain clear or consistent concepts. It makes hard to understand for human. This is due to the absence of L_M. As explained in [35], only L_M + L_{recon} help a single concept to be represented in a single element, independent with class (not just L_{recon}). C_C is an essential component of our framework.

S5 Concepts for SVHN

We observed that MNIST and SVHN learn very similar concepts. For SVHN, case $c0$ creates a small circle in the bottom part, $c1$ is being a line. Cases $c2$ and $c3$ change the upper part of the digit to a line and to a circle, respectively, as shown in Fig. S5.

Furthermore, we saw that some elements have additional changes in the background in addition to the change in the digit. For case $c1$, when the value becomes -1, the circle next to the digit becomes thicker or darker. For case $c3$, the pattern next to the digit becomes darker or larger. As such, c seems to contain classification-relevant intra-class information even though it is difficult to analyze in the case of SVHN.

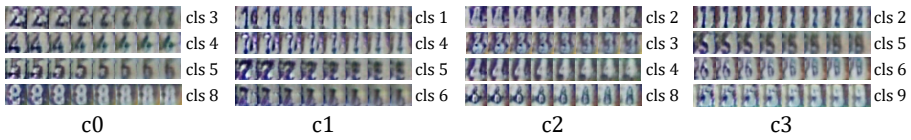


Fig. S5. Concepts learned for SVHN. $c0$: creating a small circle in the bottom part, $c1$: being a line, $c2$: changing the upper part of the digit to a line, and $c3$: changing the upper part of the digit to a circle (round).

S6 Size of c

When the size of c is reduced by half, the accuracies of MNIST and CelebA are preserved. However, in the case of SVHN, the accuracy decreased and did not exceed 87%. The accuracies of MNIST and CelebA were not degraded significantly; however, it was found that r contained much more class-relevant information than before, as illustrated in Fig. S6.

For MNIST and SVHN, when the size of c is reduced by half, more than one concept tends to appear in a single element. It seems that MINST and SVHN separate the range of a single element, and contain several concepts in ranges. For CelebA, rather than containing several concepts in a single element, only four concepts among the total of eight appeared.

When the size of c is doubled, there is no disadvantage in accuracy. However, perceiving the concept represented by each element becomes challenging. When the appropriate number of elements is assigned, identifying the concept represented by each element is comparably easy. However, if the number of elements becomes too large, each element seems to have a smaller concept than humans can perceive, or to have a similar concept between each other. Therefore, it is important to assign an appropriate size to c. We will work on this problem in future work.

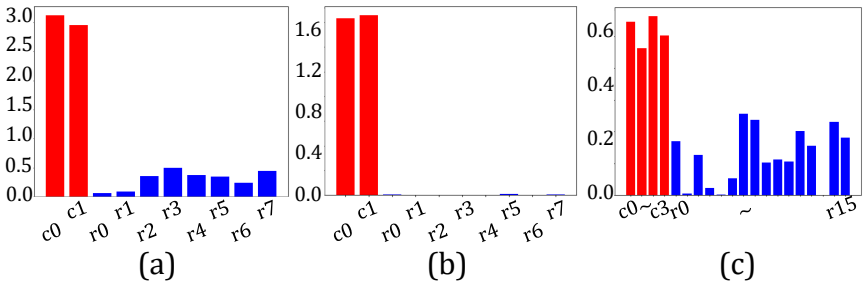
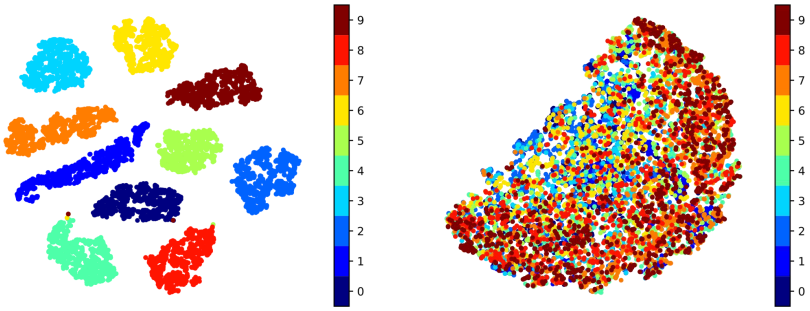


Fig. S6. Mutual information between y and c, r of (a) MNIST, (b) SVHN, and (c) CelebA when the size of c becomes half of the reported one (higher is better for c; lower is better for r).

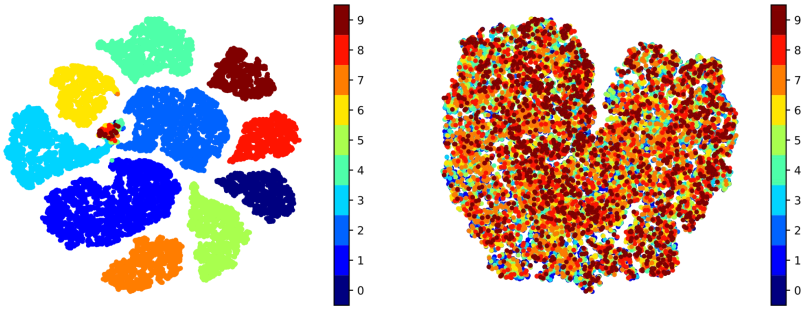
S7 T-SNE

As shown in Sec. 4.1, our method disentangles the latent feature of x to the two complementary spaces: class-relevant and class-irrelevant latent. Here, we additionally demonstrate it with T-SNE [8]. As shown in Figs. S7, S8, and S9, c shows ten or two separate clusters, and r shows a single cluster for all classes.



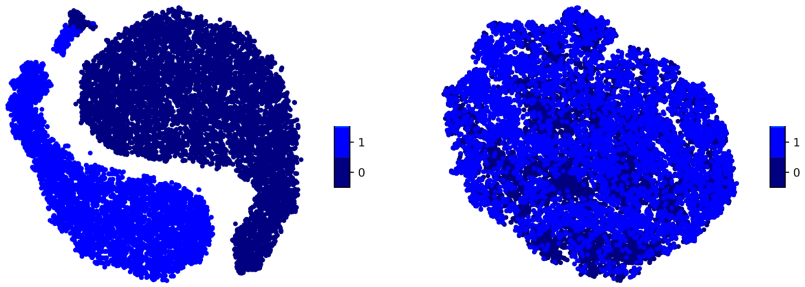
(a) MNIST: c (b) MNIST: r

Fig. S7. T-SNE for MNIST. (a) class-relevant feature c and (b) class-irrelevant feature r.



(a) SVHN: c (b) SVHN: r

Fig. S8. T-SNE for SVHN. (a) class-relevant feature c and (b) class-irrelevant feature r.



(a) CelebA: c (b) CelebA: r

Fig. S9. T-SNE for CelebA. (a) class-relevant feature c and (b) class-irrelevant feature r.

S8 FID Score

We measured the FID score of CelebA. As shown in Table. S4, for reconstruction, our method shows a better FID score [5] than the original DeepCaps. Given that DeepCaps [10] consists of a decoder, it cannot swap and randomly generate new images; therefore, we could not measure the generation and swapping performance of DeepCaps. Although it is not a fair comparison, the performance of our swapping and random generation is comparable to that of reconstruction of DeepCaps. As shown in Fig. S10, our method shows good quality even for images created by swapping.

Table S4. FID score of CelebA (lower is better).

	Ours			DeepCaps [10]
	Reconstruction	Swapping	Generation	Reconstruction
FID	86.72	103.22	103.35	101.53

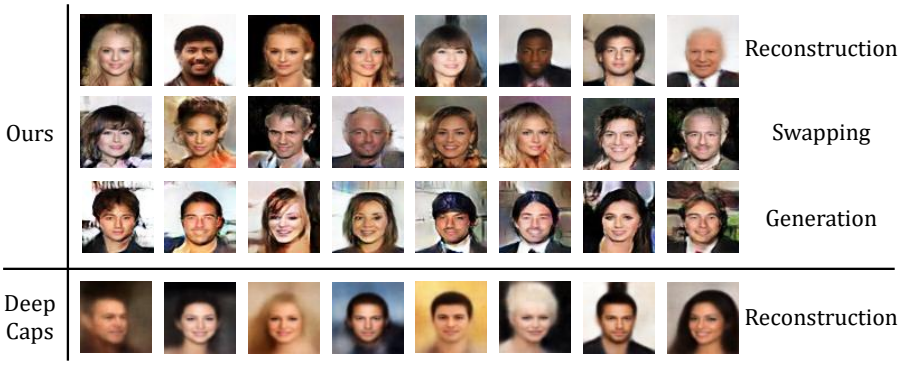


Fig. S10. The generated images by ours and DeepCaps [10]. We can see that the generated images by DeepCaps are blurry, compared to the images by ours.

S9 Comparison Methods

For the comparison methods, we reported the average of the test accuracies of five runs. In the case of Cycle-VAE [7], for MNIST, we directly copied the experimental results of the original paper since it was reported. Except for MNIST, we experimented on the SVHN and CelebA (female vs. male) datasets because these experiments were not included in the original paper. Given that the complexity of the SVHN dataset is similar to that of the MNIST dataset, we followed each dimension of c and r used for MNIST (256, 64). For CelebA, we tested on a set of (64, 64), (256, 64) and (64, 512) for (c, r) and selected and reported the best results of each experiment (64, 512). We ran the code with the reported number of epochs in this study.

In the case of ML-VAE [1], for MNIST, we ran the code with epoch 200 and followed the structure and hyperparameter setting ($c = 10$, $r = 10$) given by the official GitHub code of the authors. For SVHN, based on the experiments of the original paper, we ran the code with epoch 200 and the two hyperparameter settings : (256, 64) and (64, 64). Since (64, 64) shows better results, we reported the results of (64, 64).

In the case of LORD [4], we set the size of c to 256 and r to 128 in all experiments in accordance with the original paper.

S10 Similarity with InfoGAN

Our work is structurally similar to the unsupervised disentanglement method, InfoGAN [3], in that it uses Generative Adversarial Networks. However, InfoGAN additionally uses a factorized code predictor to learn factored representations of given observations. Unlike, our work uses an encoder and capsule network to disentangle class-relevant and -irrelevant features.

References

1. Bouchacourt, D., Tomioka, R., Nowozin, S.: Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
2. Chen, R.T.Q., Li, X., Grosse, R., Duvenaud, D.: Isolating sources of disentanglement in variational autoencoders. In: Advances in Neural Information Processing Systems (2018)
3. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems. pp. 2172–2180 (2016)
4. Gabbay, A., Hoshen, Y.: Demystifying inter-class disentanglement. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=Hyl9xxHYPr
5. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in neural information processing systems. pp. 6626–6637 (2017)
6. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369. IEEE (2010)
7. Jha, A.H., Anand, S., Singh, M., Veeravasarapu, V.: Disentangling factors of variation with cycle-consistent variational auto-encoders. In: European Conference on Computer Vision. pp. 829–845. Springer (2018)
8. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
9. Melis, D.A., Jaakkola, T.: Towards robust interpretability with self-explaining neural networks. In: Advances in Neural Information Processing Systems. pp. 7775–7784 (2018)
10. Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., Rodrigo, R.: Deepcaps: Going deeper with capsule networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10725–10733 (2019)