Supervised vs. Unsupervised: A Systematic Comparison of Deep Classification Networks

Upal Mahbub
Department of Electrical and Computer Engineering
University of Maryland
College Park, Maryland 20742
UID: 113101908

Ankan Bansal
Department of Electrical and Computer Engineering
University of Maryland
College Park, Maryland 20742
UID: 114236586

***Abstract*—In this work we explore the differences between deep classification networks which have been trained either with or without supervision. We use the discriminator of a generative adversarial model as the unsupervised classification network. These discriminator networks have previously been shown to perform very well on extracting features for the classification task even though they have not been given supervision in the form of class labels during training. We compare one such unsupervised discriminator network against a classification network trained under supervision with class labels. For this experiment, we trained both of these networks, analyzed the features learnt by them, and presented the results of our comparative studies in a systematic way to demonstrated the differences in the learnt representations of the two networks.**

I. Introduction

Generative adversarial networks (GAN) [1] are being used in several domains in computer vision. GANs consist of two networks: one for generating an image from random noise and the other for discriminating between a natural image and an artificial image generated by the generator network. These two networks are trained simultaneously in an adversarial manner where the objective of the discriminative network is to correctly identify natural and artificial images, and the objective of the generator is to prevent the discriminator from doing so. This forces the generator network to learns how to generate natural-like images, while the discriminator learns to differentiate more precisely between the natural and generated images.

The networks trained in this adversarial manner have been shown to have very nice properties and are being used for various applications such as scene understanding [2], semantic segmentation [3], and conditional image generation [4], [5], [6] etc. Recently, in [7], the authors trained a GAN model on the Imagenet dataset [8] and observed that the representations learnt by the discriminator network can be used to train image classifiers which perform well on the CIFAR-10 dataset [9]. This is despite of any class information given to the discriminator network during training. The authors claimed competitive performance of their network with other unsupervised methods for classification. Clearly, the discriminator network is able to learn some features which are useful not only for discriminating natural and artificial images, but also for classifying images into multiple categories. Given the completely different objective of the GAN network, this result if very surprising.

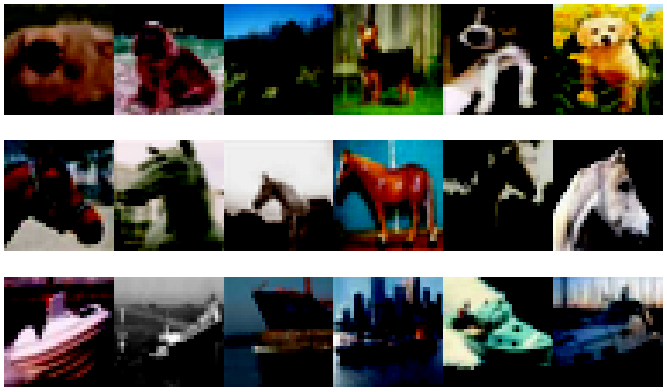

Fig. 1. Sample images from the dog (Top), horse (middle) and the ship (bottom) classes from the CIFAR-10 dataset.

In this work, we analyze the differences between the representations learnt by such GANs and a network trained with supervision of ground truth labels. For comparison, we train a classification network for CIFAR-10 dataset. This network has the same architecture as the discriminator network from [7]. We compare the features learnt by our network with the features learnt by the discriminator network from [7].

The report is organized as follows. In section II we briefly describe the most relevant previous works on this topic. We explain our experiments in detail and present the results in sections III and IV, respectively. Finally, our observations and key insights are described in section V.

II. Related Work

In this section we will first discuss generative adversarial networks [1] and then move on to DCGAN [7].

GANs estimate generative models via an adversarial process, in which two models are trained simultaneously: (a) a generative model (G) that models the data distribution, and (b) a discriminative model (D) that estimates the probability of a given image being natural (i.e. came from the training data) than from G. Adversarial training means that the objective of G is to maximise the probability of D making a mistake.

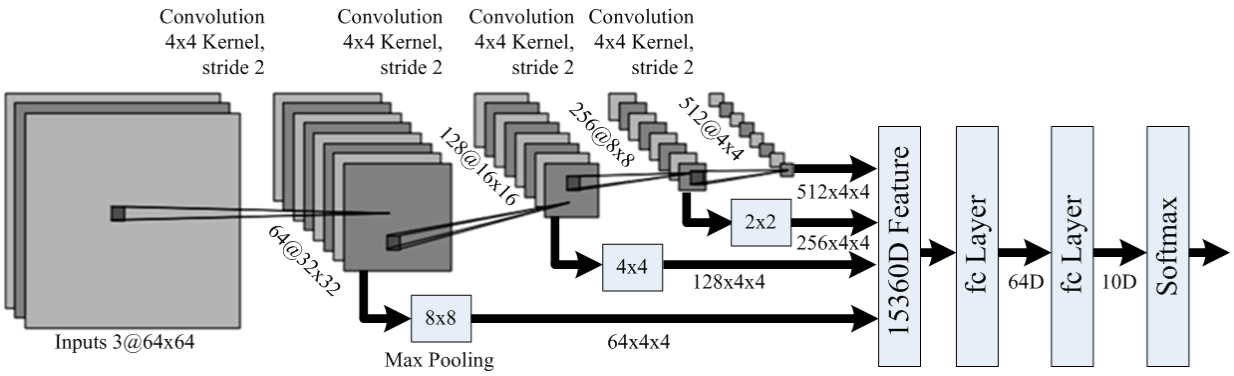

Fig. 2. The network architecture for both the disciminator network and the classification network. We train the discriminator network for classification by keeping all the convolution layers fixed and training only the fully connected layers. For training the classification network we use the same architecture but train the whole network end-to-end.

On the other hand, the objective of D is to maximise the probability of assigning the correct labels to both natural training examples and generated samples from G. This framework can be thought of as a two-player minimax game. An unique solution to this problem exists, which, for G, is to perfectly recover the underlying distribution of the training data and, for D, is to output a probability of 0.5 for both natural and generated images. The authors of [1] approximate both these models as multilayer neural networks and trained the whole system using back-propagation.

Our work mainly follows from [7]. In that paper, the authors propose and evaluate some constraints on the architecture of GANs which make them stable to train. They call this class of architectures Deep Convolutional GANs (DCGAN). They observed that the deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. The authors also observed that the trained discriminator networks can be used for image classification tasks showing competitive performance with other unsupervised algorithms. Also, these representations could be used for novel tasks like vector arithmetic on face samples. Some of the constraints introduced in [7] are: (a) using strided convolutions instead of pooling layers; (b) using batch normalisation [10] in both generator and discriminator; (c) using LeakyReLU activation [11], [12] in the discriminator for all layers except the last Sigmoid layer etc. Using strided convolutions instead of pooling layers allows the network to learn its own spatial down-sampling rather than specifying a fixed one. Batch normalisation stabilises training by normalising the input to each layer. This helps gradient flow in deeper models. Our classification network uses the same network architecture used by the discriminator network in [7] which was trained on the LSUN bedrooms dataset.

The CIFAR-10 datasset [9], that is being used here for evaluation, is a collection of 32×32 colour images of 10 categories - airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. It has $50,000$ images for training and $10,000$ images in the test set. Figure 1 shows some sample images for three categories from the dataset. Before passing the images through the networks, the images were resized to 64×64.

In the following sections we discuss our methodology and observations.

III. Problem Formulation and Setup

Our primary goal in this work is to compare the discriminator network from DCGAN trained in an unsupervised manner with a supervised network of the same architecture in terms of the features learned by the two networks.

A. Unsupervised Training of the Discriminator Network (DN)

The authors of [7] were kind enough to make their code available publicly. They also provide the network models of the generator. However, we wanted to compare their discriminator network against a network trained with supervision. So, we had to train the adversarial model ourselves. We used the codes provided by the authors to train an adversarial generative and discriminative networks on Imagenet [8]. Note that the generator network outputs a 64×64 image. So the input to the discriminator network is a 64×64 image. To use the CIFAR-10 images, we up-sample them to this size. (We use the same size for our classification network too.) We trained the networks till convergence (25 epochs). This took almost one week on two Titan X GPUs . All the results in the report are based on the discriminator network obtained at the end of the 25^{th} epoch in our experiments.

Figure 2 shows the architecture of the discriminator network (from [7]) that was used for training the GAN.

B. Supervised Training of the Classification Network (CN)

Because there is a $tanh$ non-linearity at the output of the generator network in [7], we normalise the images such that each channel of the image lies in $[-1, 1]$. We use the architecture shown in 2. We max-pool the features from each convolutional layer to a spatial dimension of 4×4 and concatenate the resulting features. We reshape these features to get a $15,360$-D vector $(64*4*4+128*4*4+256*4*4+512*4*4)$. We add a fully connected layer (output size 64) on top of

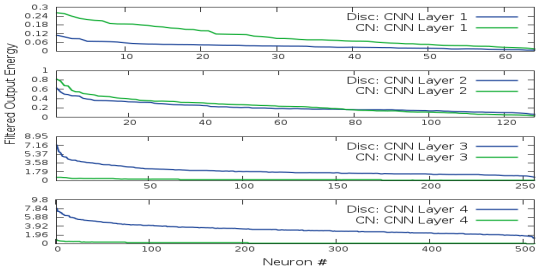

Fig. 3. Sorted energies dog. Both networks correctly classified the input image of dog (fig. 1, top row, first image from the left).

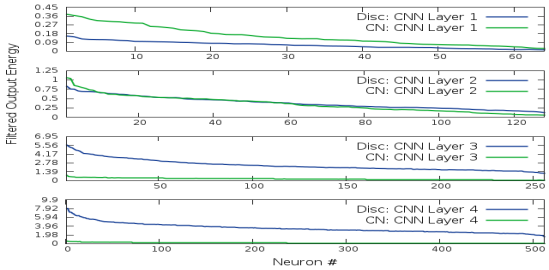

Fig. 4. Sorted energies horse. Both CN and DN correctly classified the input image of horse (fig. 1, middle row, first image from the left).

this vector. Another fully connected layer on top of this with Softmax gives us the probabilities of the 10 classes. We use batch-normalisation [10] and dropout [13] in this layer. We train this network on the training set of CIFAR-10. The network is trained for 200 epochs achieving a training accuracy of 98.63% and an accuracy of 80.87% on the test set.

C. Using the Discriminator Network for classification

We use the discriminator network trained using adversarial approach to extract the $15,360$-D features (in a similar manner as for the classification network) for the CIFAR-10 training set. We use these features to train a multi-layer perceptron for classification consisting of a 64-D fully connected layer with ReLU activation and an output layer with Softmax (see figure 2). The architecture of this top MLP block is exactly the same as the top portion (starting from the fully connected layer towards the softmax) of the supervised classification network. When tested on the CIFAR-10 test set, this approach gives an accuracy of 75.19%. From now on, we refer to the combined discriminator network and the added mlp as DN.

Notably, both the networks achieve comparable performance. This is really interesting since the convolutional layers of the discriminator network was not trained using any class information. Yet, it still learnt discriminating features that are suitable for classification.

IV. Results

In this section, we compare the filters and filter responses of the two networks and try to gain some insight into the similarities and differences between the two.

A. Comparing the energies of activations

We calculate the energy of each activation in each layer, and sort these energies in decreasing order of magnitude. We analyse the energy profiles for the following four cases:

1) Correctly classified by both: Figure 3 shows the energy profiles for an image of a dog for both networks and figure 4 shows the profiles for an image of a horse. Both of these networks correctly classified the images.

2) Correctly classified by DN but incorrectly by CN: Figure 5 shows the energy profiles for an image of a dog for both networks and figure 6 shows the profiles for an image of a horse. The DN correctly classified the images but CN did not.

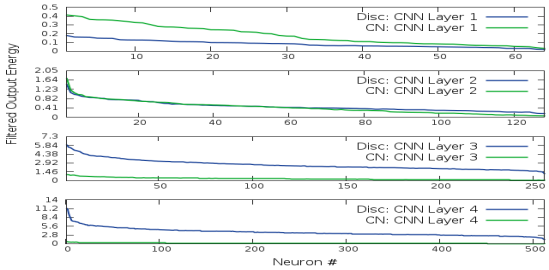

Fig. 5. Sorted energies dog. DN correctly classified the the input image of dog (fig. 1, top row, sixth image from the left) which CN misclassified as truck.

3) Correctly classified by CN but incorrectly by DN: Figure 7 shows the energy profiles for an image of a dog for both networks. Figure 8 shows the profiles for another image of a dog. These images were correctly classified by the CN but the DN did not classify these correctly.

4) Incorrectly classified by both: Figures 9 and 10 show the energy profiles for two different images of dogs for both networks. None of these networks correctly classified the images.

Note that for all cases, the total energy (area under the energy curve) and the maximum energy increase as we move up the layers for DN but decrease for CN. For each case (figures 3 - 10), CN has more energy in the first layer than DN. Both networks have similar energy in layer 2, and then DN takes over and has much higher energy in layers 3 and 4 than the corresponding layers in CN. This can also be seen from the visualisations of filter activations (figures 11 - 22) where we see that the activation magnitudes in the higher layers of CN are very low compared to activations of the corresponding layers in DN.

B. Cosine distance between the 64-D feature vectors

For studying the structure of the 64-D space where the features lie, we calculated the cosine distances between in-class and inter-class pairs of features for both networks. Table I lists the average in-class and inter-class distances for a few classes and class-pairs for the two networks.

A very interesting thing to note here is the magnitudes of distances for the two networks. The CN seems to project

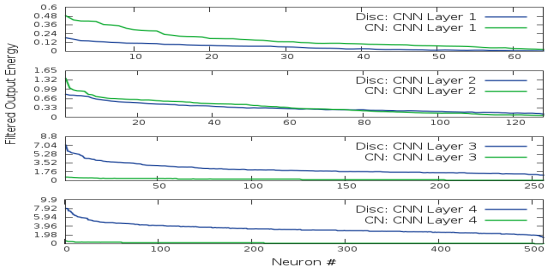

Fig. 6. Sorted energies horse. DN correctly classified the the input image of horse (fig. 1, middle row, fifth image from the left) which CN misclassified as cat.

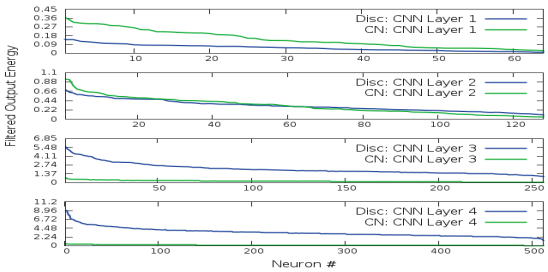

Fig. 7. Sorted energies dog. CN correctly classified the the input image of dog (fig. 1, top row, second image from the left) which DN misclassified as deer.

everything very close to each other. Though, there is still a difference for within-class and between-class distances. The DN projections are quite spread out. And the inter-class distances are much higher than within-class distances too. This is interesting because ideally you would assume that the DN projects all natural images to the same part of the space and the CN projects different classes to different parts. However, it seems that in practice the opposite is true. Note that the ratio of distances for different classes and same classes is higher for CN, which explains the reason for its superior performance over DN.

C. Visual comparison of activations at different layers

We visualize the neuron activation outcomes of the two networks at the four convolutional layers for different images. Since the number of neurons are large, we sort the neurons based on activation energy at the output in descending order and plot the heat map for the top 16 most active neurons for any input image in a 4×4 layout. In those plots, the top

TABLE I
AVERAGE COSINE DISTANCES BETWEEN CLASSES OVER 50 PAIRS

Class-pair	Avg. Cosine distance for DN	Avg. Cosine distance for CN
Dog-Dog	0.7119	0.0015
Horse-Horse	0.6993	0.0020
Ship-Ship	0.5853	0.0026
Dog-Horse	0.7424	0.0023
Dog-Ship	1.2079	0.0076
Horse-Ship	1.1692	0.0068

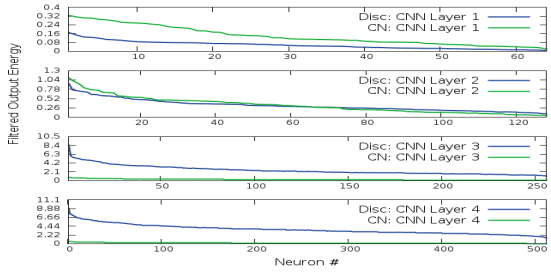

Fig. 8. Sorted energies horse. CN correctly classified the input image of dog (fig. 1, top row, third image from the left) which DN misclassified as horse.

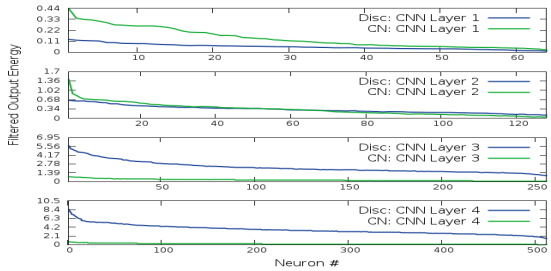

Fig. 9. Sorted energies dog. Both of these networks incorrectly classified the input image of dog (fig. 1, top row, fourth image from the left) as deer.

left subplot corresponds to the highest activation energy and the bottom right corresponds to the lowest activation energy among the 16 neurons, if otherwise not specified.

1) Correctly classified by both networks: First let's compare the activations of the networks for the case when both were able to correctly classify the image. Figure 11 shows the network activations for both networks for an image of a dog (fig. 1, top row, first image from the left) and figure 12 shows the activations for an image of a horse ((fig. 1, middle row, first image from the left).

In figure 11, let us compare the highest energy activations for DN and CN for the lowest layer. A close examination reveal several pairs of neurons between DN and CN network that have similar activations, such as, 4 and 25, 62 and 41 and 14 and 27. This observation can easily be verified by figure 12. For this image, finding neuron 4, 62 and 14 among the most active neurons of layer-1 automatically led us to assume that neuron 25, 41 and 27 will be present in the list of most active neurons of layer-1 of CN, which they are. This shows that the DN learns something at least partially similar to what the CN learns in CNN layer-1, although they were trained for different objectives. We know for classification networks, the first CNN layer usually reveals edge information. Hence, we can assume that DN also tries to extract edge information from the input image. Now, as we move towards higher CNN layer activations, the output images gradually becomes more abstract and finding similar activations for the CN and DN architectures become very difficult. For CNN layer-2 of 11, 38-94 and 107-77 pairs seem to match, and they are found in figure 12 for the horse image as well. Just by looking at the patterns, one might say that the two networks are learning

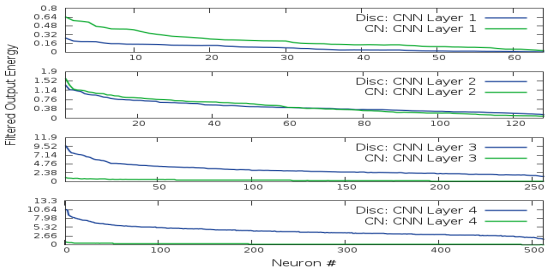

Fig. 10. Sorted energies horse. Both of these networks incorrectly classified the input image of dog (fig. 1, top row, fifth image from the left). DN classified it as cat and CN as bird.

something similar, especially at the lower layers.

We show other cases in the Appendix of the report.

V. Discussion and Conclusion

A notable point we would like to discuss is the reduction in energy of the higher layers of the CN. We believe that this is because the network was able to fit (over-fit, in fact (we will come to that part in a bit)) the data without using the upper layers and made everything in the upper layers very close to zero. The first two layers contained most of the energy in the network and the upper two convolution layers had little energy.

As we've mentioned earlier, both of these networks seem to overfit the training data and therefore could not reach the training accuracy on the test set. We attribute this phenomenon to the very large feature dimension of 15,360 that was obtained from the four convolutional layers. This feature is fed to the first fully connected layer to obtain a 64 dimensional feature vector. This fully connected layer has $15,360 \times 64$ parameters which is a huge number. Though we added batch-normalization [10] and drop-out [13] with probability 0.5 afterwards, those proved to be insufficient regularisation given the very large number of trainable parameters. In [7], the authors claimed to have achieved $\approx 82\%$ accuracy with the DN. However, the finer details of their methods are not apparent from their paper. Given our goal of analyzing the DN, it was not essential to achieve state-of-the-art performance with the network and therefore, we refrained from fine-tuning the hyper-parameters rigorously.

We compared the 16 most activated neurons for two dog images (refer to figs. 11 and 15) and a horse image (refer to fig. 12) for DN. Our analysis showed that the neurons in different layers that are common in the top 16 of the two dog images are as follows:

- CNN Layer 1: $14, 4, 6, 1, 62, 53, 58$
- CNN Layer 2: $64, 66, 107, 33, 57$
- CNN Layer 3: $72, 91, 95, 45$
- CNN Layer 4: $495, 305, 34, 69, 476, 404, 382$

And, the common neurons between the first dog image and the horse image are:

- CNN Layer 1: $4, 6, 19, 23, 53, 58, 48, 56, 18, 32$
- CNN Layer 2: $71, 64, 66, 107, 33, 17, 26, 114, 38, 36, 82$
- CNN Layer 3: $72, 217, 203, 91, 95, 45, 92, 26$
- CNN Layer 4: $354, 305, 114, 475, 229, 34, 69, 476, 382$

It is very interesting to see that the horse image has more in common in terms of the active neurons, although, the network correctly classified all these images. The reason behind this similarity for the DN networks is probably the fact that it was trained with an objective to discriminate between natural and generated images. Since, all these input images are natural, they share much similarity in the representation by DN. However, for the classification task, all the responses were combined together and fed to the fc layers. When evaluated together with an objective to discriminate between classes, the fc layers does a fine job of finding separating hyperplanes. As we have discussed before, similar to CN, the DN also shows the tendency to learn hierarchical features, a combination of all of them provides the fc layers ample opportunity to find features that would separate different classes from each other. This argument is backed by figs. 13 and 14 where we show pairwise cosine distance between activation energies per layer for DN and CN, respectively. It can be seen that it is hard to discriminate between same class and different class distances at any of the four convolutional layers, although, we already reported that both networks are capable of discriminate between the classes reasonably. Therefore, we we assume that none of the two networks are specifically learning class discriminating properties at the convolutional layers, rather they learn to discriminate effectively at the fc layers.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[2] S. Eslami, N. Heess, T. Weber, Y. Tassa, K. Kavukcuoglu, and G. E. Hinton, "Attend, infer, repeat: Fast scene understanding with generative models," *arXiv preprint arXiv:1603.08575*, 2016.

[3] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[4] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Advances In Neural Information Processing Systems*, pp. 217–225, 2016.

[5] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[6] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune, "Plug & play generative networks: Conditional iterative generation of images in latent space," *arXiv preprint arXiv:1612.00005*, 2016.

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

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[9] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[11] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013.

[12] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

Fig. 11. Activations for each layer of the DN (top) and the CN (bottom). Both of these networks correctly classified the input image (fig. 1, top row, first image from the left) as dog .

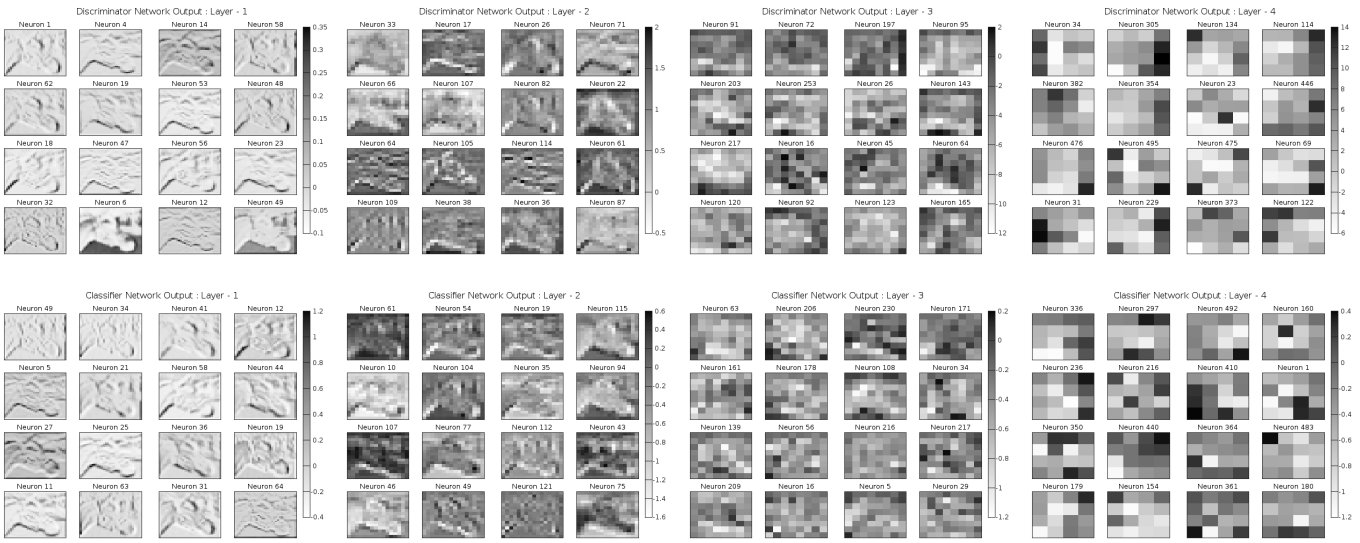

Fig. 12. Activations for each layer of the DN (top) and the CN (bottom). Both of these networks correctly classified the input image (fig. 1, middle row, first image from the left) as horse.

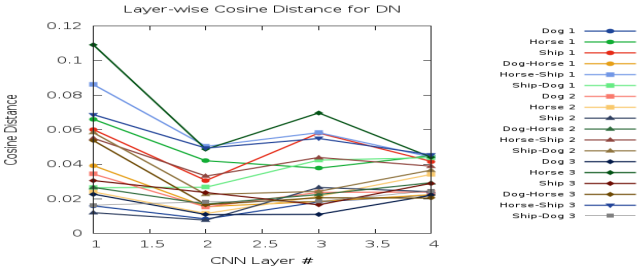

Fig. 13. Pairwise cosine distance betweeen neuron activation enargies between several similar and dissimilar class samples for DN.

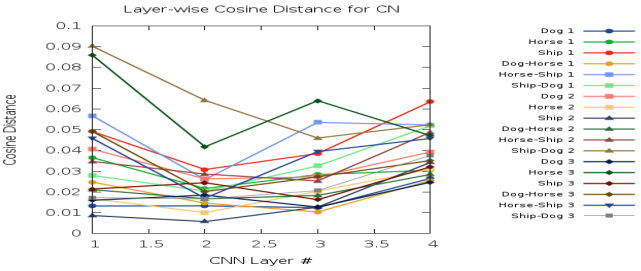

Fig. 14. Pairwise cosine distance betweeen neuron activation enargies between several similar and dissimilar class samples for CN.

[13] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

APPENDIX

A. Visual comparison of activations at different layers

We continue from section IV-C and show and discuss feature visualisations for some more cases. We note that the features learnt in the first layer are very similar to each other, with some of the features looking exactly the same for DN and CN. However, for the upper layers, the features are not similar. Also, traditional neural networks learn hierarchical features, but our CN does not seem to be doing that. We believe that this is because our networks are not hierarchical. Traditional feed-forward CNNs can be thought of as Markov chains where information about one layer eliminates the need for knowledge about lower layers. On the other hand, the CN in our case feeds information from multiple layers to the classifier and the training is not hierarchical.

1) Correctly classified by DN but incorrectly by CN: Next, let's compare the activations of the networks for the case when DN was able to correctly classify the image, while CN failed. Figure 11 shows the network activations for both networks for an image of a dog (fig. 1, top row, sixth image from the left) and figure 12 shows the activations for an image of a horse ((fig. 1, middle row, fifth image from the left).

2) Correctly classified by CN but incorrectly by DN: In this part we compare the activations of the two networks for the case when CN correctly classifies the images but DN does not. Figure 17 shows the highest activations for an image of a dog which was classified correctly as a dog by CN but incorrectly as a deer by DN. Similarly, figure 18 shows the activations for an image incorrectly classified by DN as a horse.

3) Incorrectly by both: Now we compare the activations for the case when both the networks incorrectly classify the image. Figure 19 shows the highest activations for an image of a dog which was classified as a deer by both networks. Figure 20 shows the activations for an image of a horse classified as a truck by both networks.

Figure 21 shows the activations for an image of a dog classified as a cat by DN and as a bird by CN. Figure 22 shows the activations for an image of a horse classified as an airplane by DN and as a deer by CN.

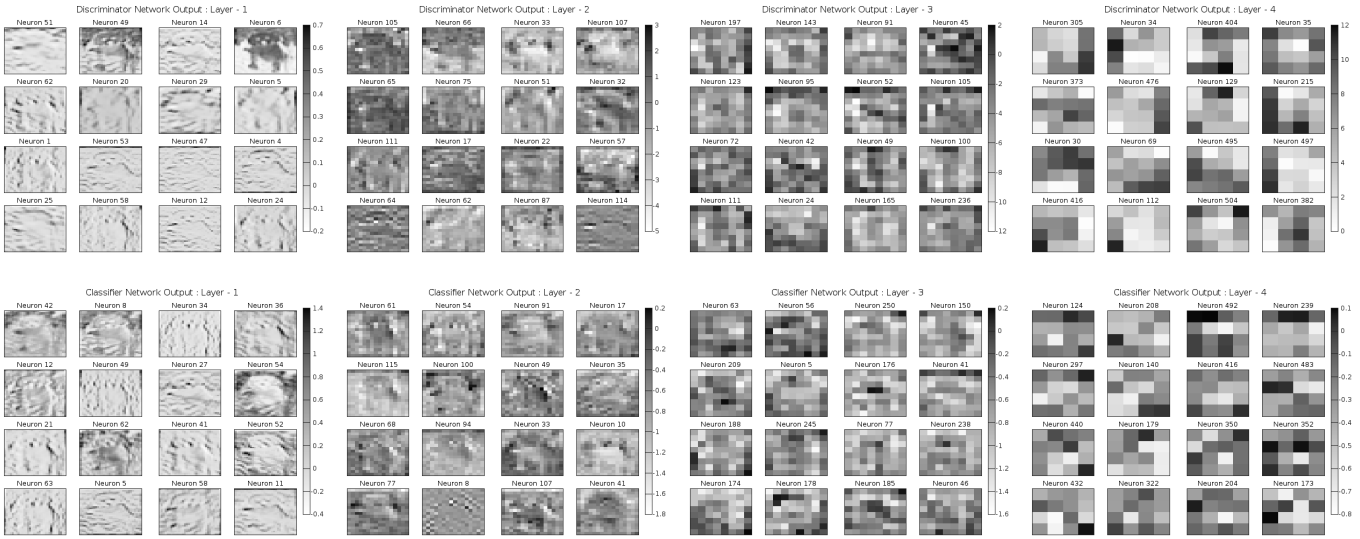

Fig. 15. Activations for each layer of the DN (top) and the CN (bottom). DN correctly classified the input image (fig. 1, top row, sixth image from the left) as dog, while CN misclassified it as a truck.

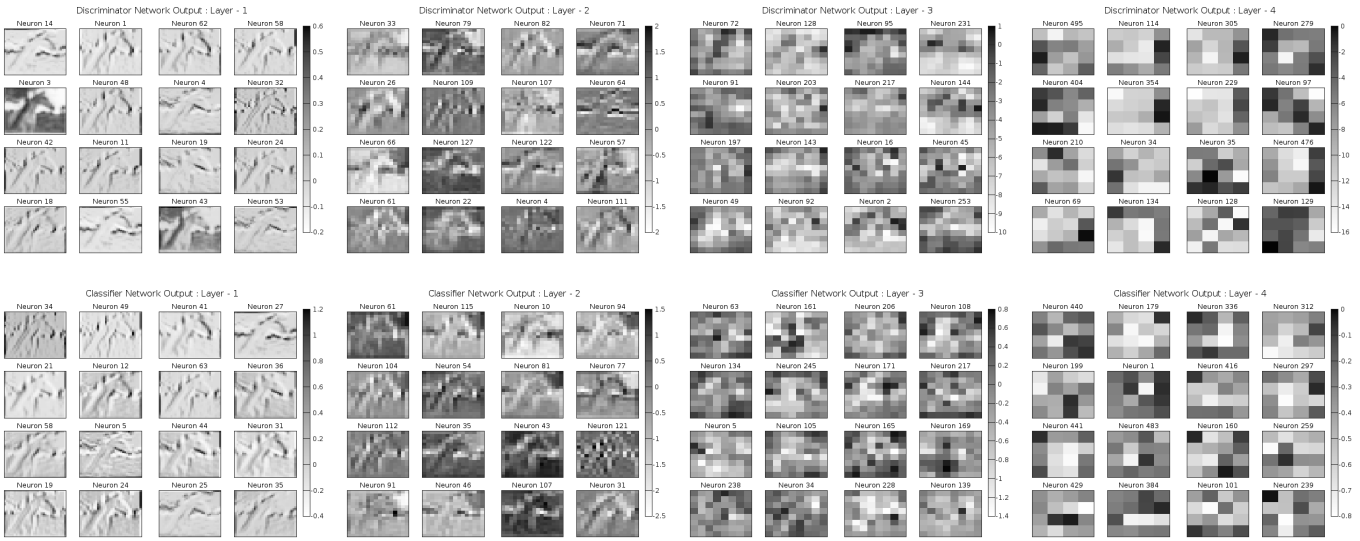

Fig. 16. Activations for each layer of the DN (top) and the CN (bottom). DN correctly classified the input image (fig. 1, middle row, fifth image from the left) as horse, while CN misclassified it as a cat.

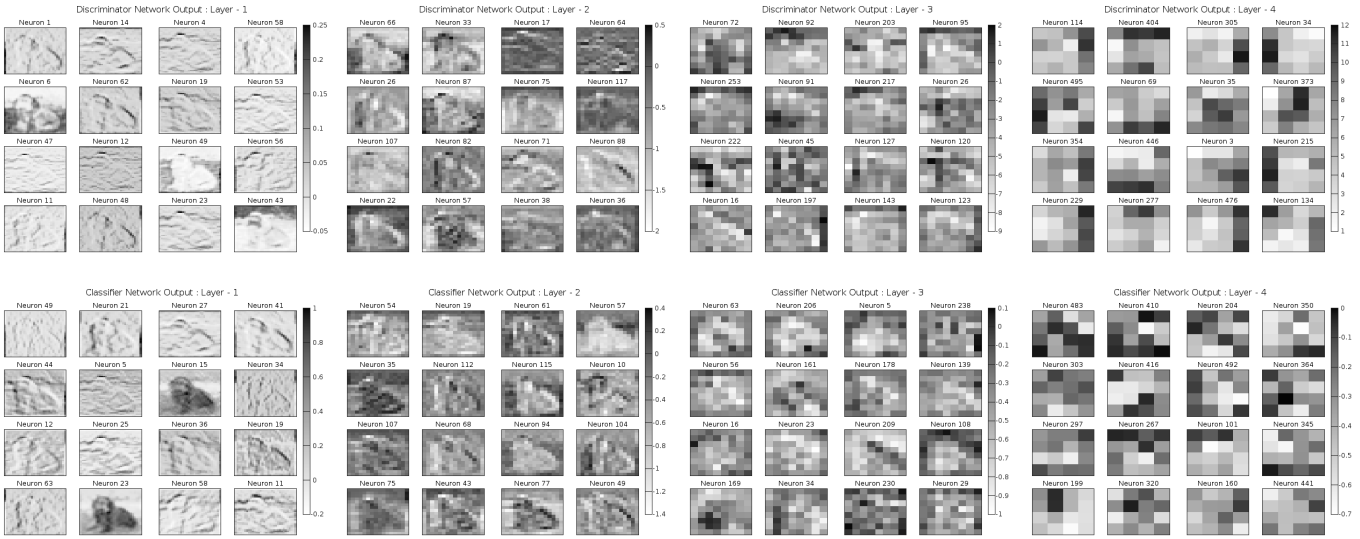

Fig. 17. Activations for each layer of the DN (top) and the CN (bottom). CN correctly classified the input image (fig. 1, top row, second image from the left) as dog, while DN misclassified it as deer.

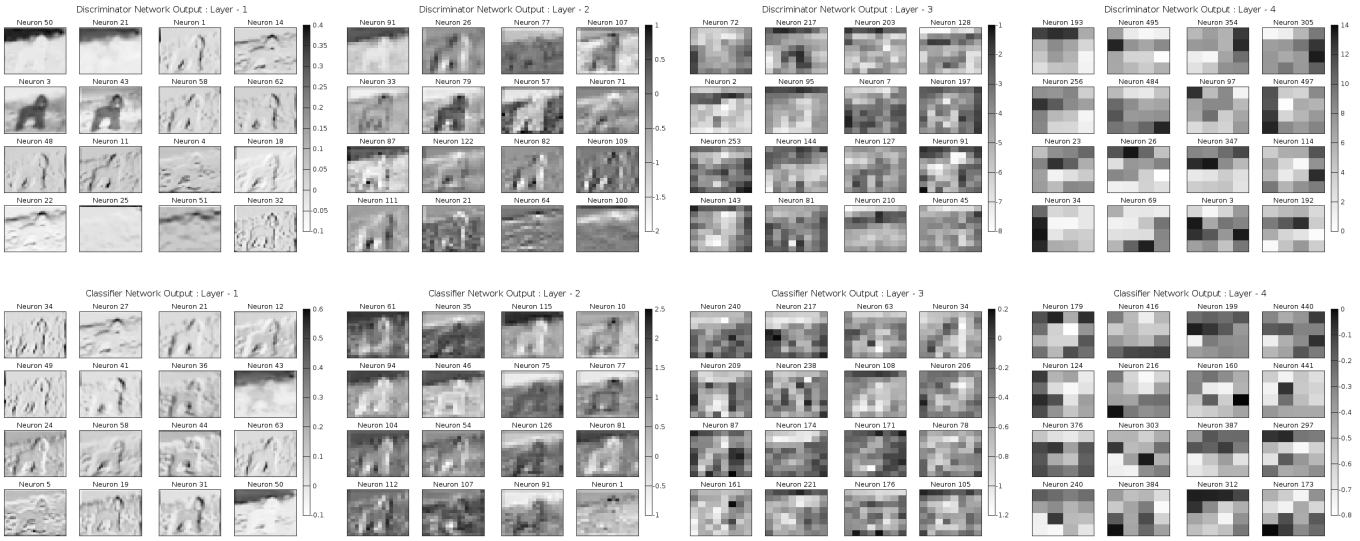

Fig. 18. Activations for each layer of the DN (top) and the CN (bottom). CN correctly classified the input image (fig. 1, middle row, third image from the left) as dog, while DN misclassified it as horse.

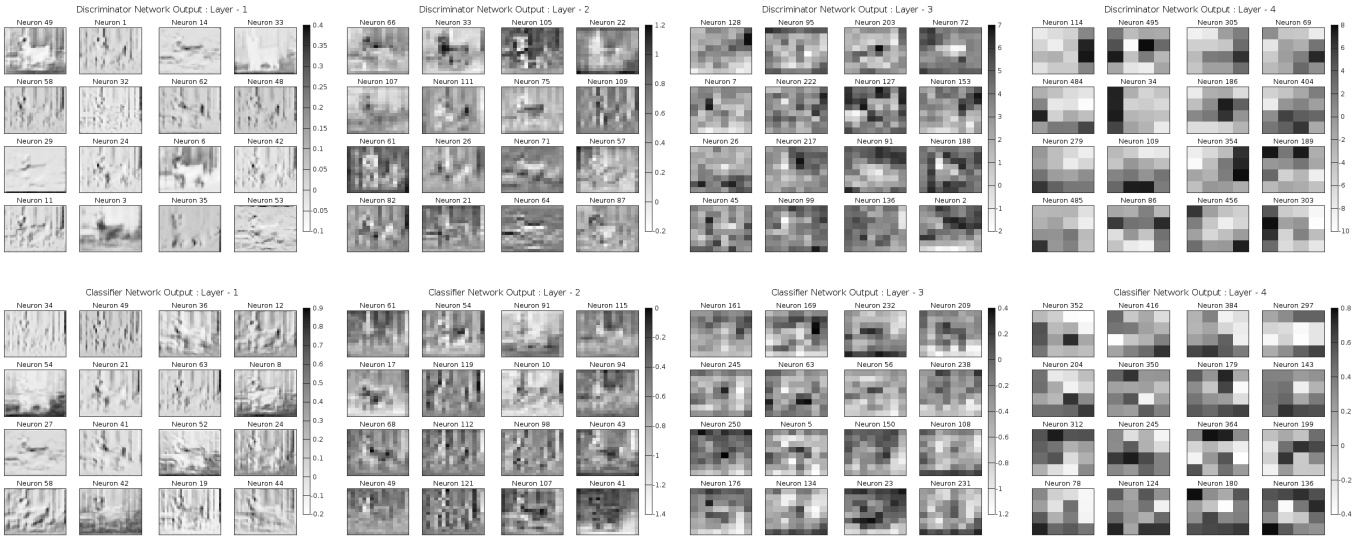

Fig. 19. Activations for each layer of the discriminator network (top) and the classification network (bottom). Both of these networks incorrectly classified the input image of dog (fig. 1, top row, fourth image from the left) as deer.

Fig. 20. Activations for each layer of the discriminator network (top) and the classification network (bottom). Both of these networks incorrectly classified the input image of horse(fig. 1, middle row, second image from the left) as truck.

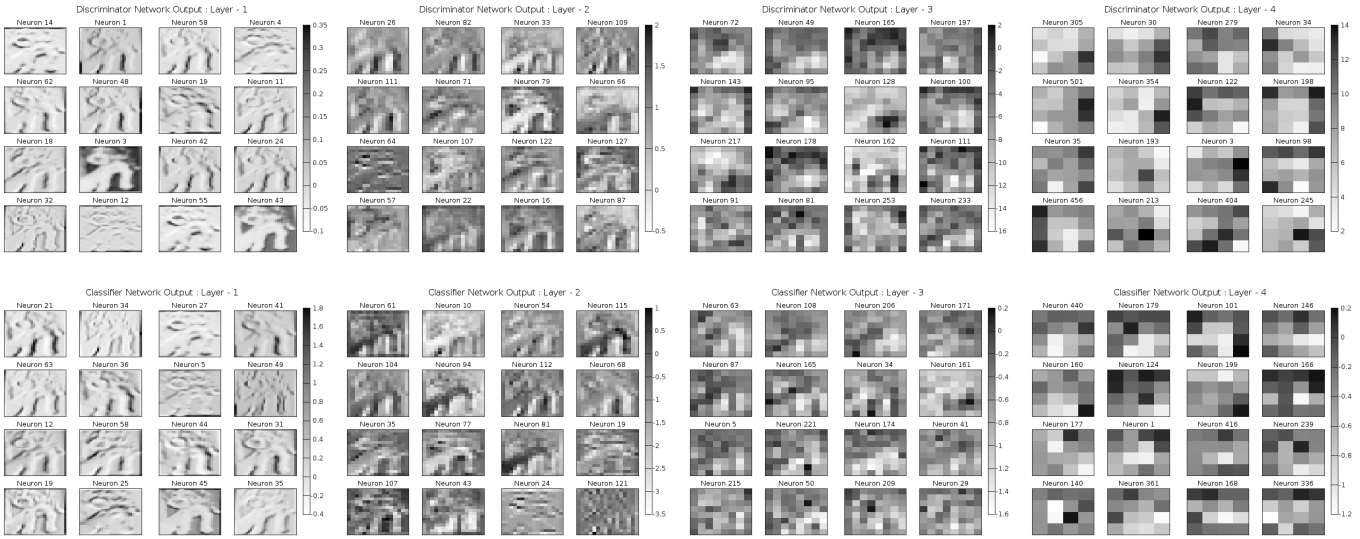

Fig. 21. Activations for each layer of the discriminator network (top) and the classification network (bottom). The DN incorrectly classified the input image of dog (fig. 1, top row, fifth image from the left) as cat, while CN misclassified it as bird.

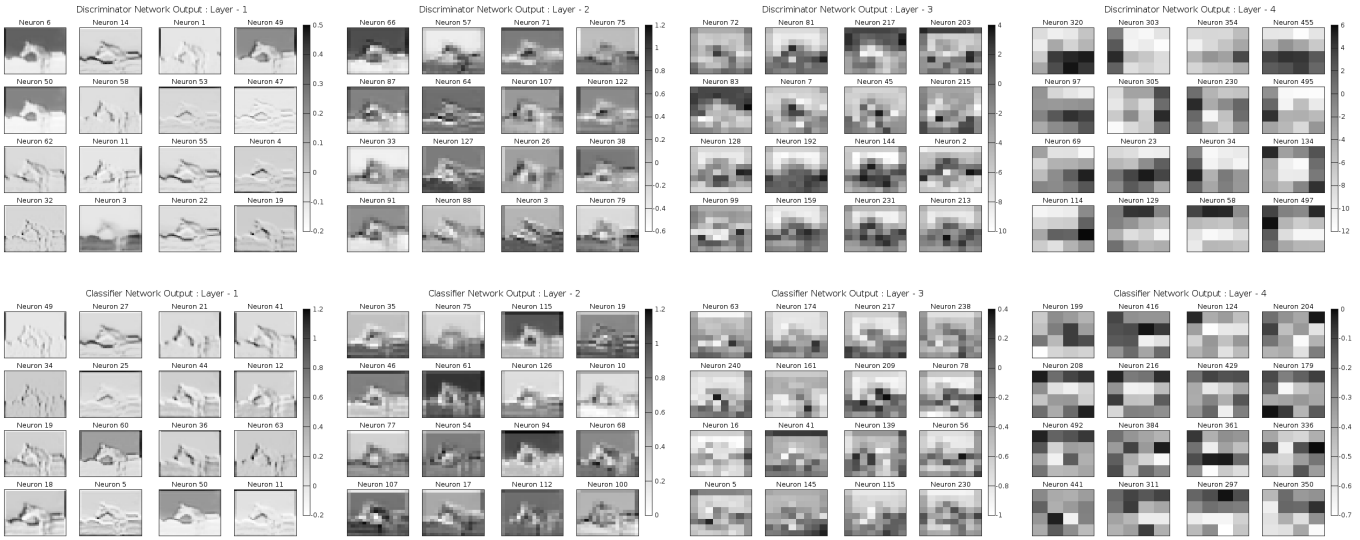

Fig. 22. Activations for each layer of the discriminator network (top) and the classification network (bottom). The DN incorrectly classified the input image of horse (fig. 1, middle row, third image from the left) as airplane, while CN misclassified it as deer.