Image Regeneration with Generative Models

Abhijith C.(1MV14CS004), Raghava G Dhanya(1MV14CS077) and Shashank S.(1MV14CS131)
Guide: Smt Sushila Shidnal
Sir M Visvesvaraya Institute of Technology

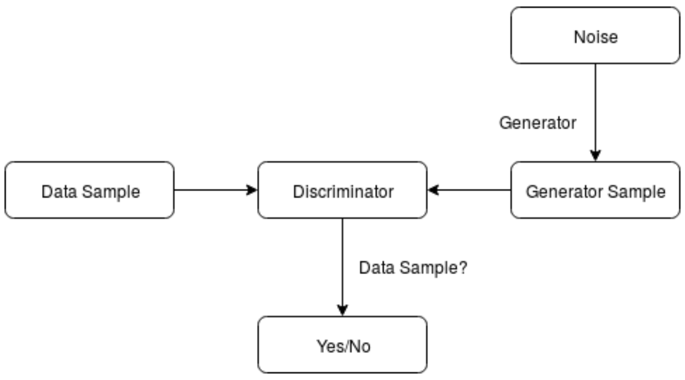
Abstract

Current advances in Generative Adversarial Networks allow us to obtain near realistic images but it is still quite distinguishable from actual photographic images. The technology is also not very amiable to changes in the orientation of images in Convolutional Neural Networks(CNN). Additionally, the amount of data required to train the network must be exhaustible, for example, in case different perspectives of a face are required the various perspectives must be explicitly present in the training data to achieve the result. Thus the network requires humongous amounts of data.

In this project we propose a novel approach to accomplish the same results using CapsNet. CapsNet employs a dynamic routing algorithm which replaces the scalar-output feature detectors of the CNN with vector-output capsules. A capsule is essentially a group of neurons describing a specific part of object or image. Active capsules at one level make predictions, via transformation matrices, for the instantiation parameters of higher-level capsules. In essence, the CapsNet is the reverse of the common Computer Graphics pipeline where we convert objects to their renders. The CapsNet works from the pixel level and works up towards the object.

Generative Adversarial Networks

Generative Adversarial Networks[1] pose the training process as a game between two distinct networks: A neural network, called the generator, generates new instances of data, while the other, the discriminator, evaluates their authenticity; discriminator network tries to classify samples as either coming from the true distribution $p(x)$ or the model distribution $\hat{p}(x)$. Every time the discriminator notices a difference between the two distributions the generator adjusts its parameters slightly to make it go away, until at the end (in theory) the generator exactly reproduces the true data distribution and the discriminator is guessing at random, unable to find a difference.



This can be represented minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \quad (1)$$

Capsule Networks

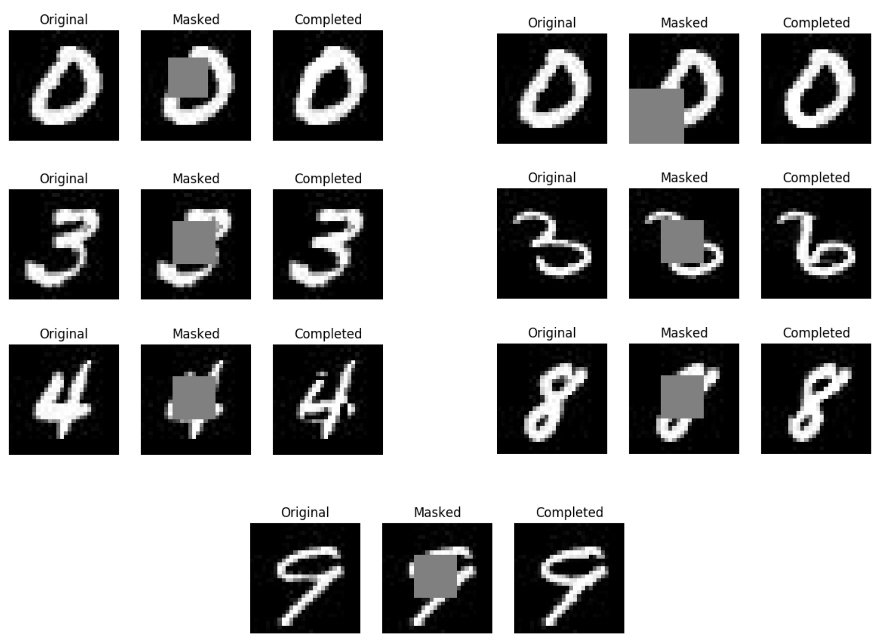
Capsule is a nested set of neural layers. Capsules are like cortical columns in human brains. Deep neural nets learn by back-propagation of errors over the entire network. In contrast real brains supposedly wire neurons by Hebbian principles: "units that fire together, wire together".

Capsules mimic Hebbian learning in the way that: "A lower-level capsule prefers to send its output to higher level capsules whose activity vectors have a big scalar product with the prediction coming from the lower-level capsule". Capsules, combination of capsules encodes objects parts AND their relative positions, so an object instance can be accurately derived from the presence of the parts at the right locations, and not just their presence. Capsules produce equivariant features. Capsules predict the activity of higher-layer capsules to route information to right higher-layer capsules, this is called "Dynamic routing"[2].

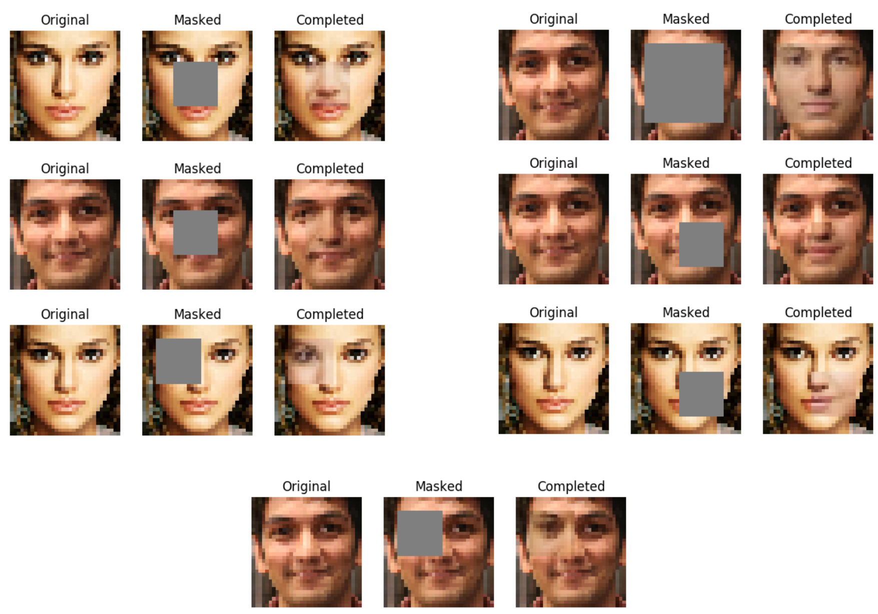
Semantic Inpainting

To demonstrate the application of our modified GAN, we will be using Semantic Inpainting. Inpainting is the process of reconstructing lost or deteriorated parts of images and videos. In the museum world, in the case of a valuable painting, this task would be carried out by a skilled art conservator or art restorer. In the digital world, inpainting (also known as image interpolation or video interpolation) refers to the application of sophisticated algorithms to replace lost or corrupted parts of the image data (mainly small regions or to remove small defects). Here we will be using GAN to implement semantic inpainting.

Results 1



Results 2



Conclusion

During the course of this project, we wished to replicate the results of the existing state-of-the-art in Generative Models. We implemented a few different versions of GANs with CapsNet. Our motivating assumption was that CapsNet would provide a performance improvement. We based this on the idea that it is more capable of understanding the variances in objects. This in turn should lead to lower data requirements during training of the model and consequently lower power consumption.

In conclusion, we can confidently state that augmenting the GANs with CapsNet was a fruitful endeavor. The CapsNet helped to reduce th training overhead considerably when compared to classical networks while providing remarkably similar results. Our research shows that embedding CapsNet into the GAN does not degrade its performance and, in certain cases, improves upon it.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3856–3866. Curran Associates, Inc., 2017.