Visualizing and Understanding CNN From A Biological Point Of View

SYDE 552/BIOL 487 Project Report

Ding (Henry) Tong 20870647

Namitra Kalicharran 20674483

Karan Chopra 20573139

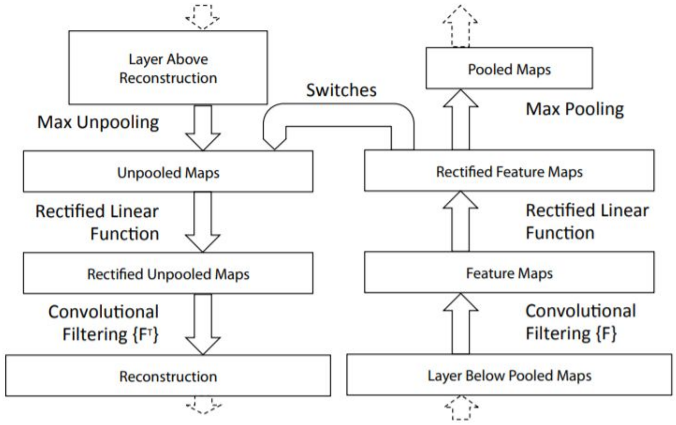
Introduction

The goal of our project is to visualize and compare the feature maps that a convolutional neural network (CNN) learns, after training, to features used in visual processing in the brain. In lectures, we learnt the computational and the mathematical mechanisms of a CNN, which are widely used for computer vision. Additionally, we also learnt about visual processing in our brain. However, we didn't go into the relationship between visual processing in a CNN and visual processing in the brain. CNNs were initially inspired by early findings in biological vision, and so we believed that there would be many similarities with their visual processing (Lindsay, 2020). Zeiler and Fergus (2014) were interested in why CNNs perform so well at classification, on the ImageNet benchmark, so they trained a CNN and created a visualization network to investigate the contribution of different layers of the model and thus improve the performance of the model further. This inspired us to use a similar visualization technique to see what types of patterns in an image caused a feature map to activate for each layer of CNN. We used these visualizations to compare the feature maps generated by each layer of a CNN to biological features. Another paper we used as reference how the LGN of a cat's brain responded to lines of different orientations, as well as contours (Hubel & Wiesel, 1959). We expect to see similar patterns emerge in the feature maps of the early layers of a CNN. In general, we expect to find that there are many similarities between the biological features and the feature maps of a CNN. This would explain why CNNs perform so well from a biological point of view.

Methods

In class we learned about using tensorflow to create convolutional networks and train them to perform image classification. In this project these skills were used and elaborated on to train two CNNs. The first model, VGG16 (Simonyan & Zisserman, 2014) is a famous CNN known for performing very highly in the ImageNet image classification challenge. Each of it's the layers consist of a series of convolution operations each with a ReLU activation, followed by a max pooling operation to condense the image even further. We used the already trained ImageNet weights for the VGG16 model since we used a subset of the ImageNet dataset, provided through tensorflow, for our project. The CNN model had to be slightly tuned since the dataset was significantly smaller than the original ImageNet dataset.

To understand the operation of the CNN, we used a deconvolutional neural network (DCNN) proposed by Zeiler, et al. (2011). The purpose of this DCNN is to reproject the feature activities in the intermediate layers of the CNN back into the input image shape. This helps us understand what features in the input image initially caused the given activation in the feature maps. A DCNN can be thought of as the opposite of a CNN, it uses the same operations such as pooling and filtering but in the reverse direction (Zeiler, et al., 2011). A DCNN was attached to each layer in the CNN to each of the layers in convnet as shown in Fig. 1. The input image is first presented to the CNN model so that it can generate feature maps. These feature maps were then fed into the DCNNs to try and reconstruct the original input image. The DCNN performs this reconstruction by repeatedly deconvoluting, activating, and unpooling the feature map until it is finally projected into input image size. Once the DCNN is trained to reconstruct the images from a feature map, you can feed in feature maps where every channel is zeroed, except for the feature map you're interested in visualizing. The DCNN will then reconstruct the image and show what activates that channel in the original image. The only difference in our implementation compared to the implementation in the original paper is that tensorflow doesn't require switches to unpool a feature map, it instead has an upsampling layer that scales the features maps to a larger size.



(Adapted from Zeiler & Fergus, 2014).

Figure 1. An illustration of the reconstruction of the feature map from a CNN using a DCNN.

Results

We trained a DCNN to reconstruct the feature maps after the first pooling layer in the VGG16 CNN (early DCNN) and another DCNN to reconstruct the features in the last feature map of the VGG16 CNN (late DCNN). Initially we wanted to train DCNNs for every pooling layer in the VGG16 model, but google colab barely allowed for these two models to be trained without throwing resource exhaustion errors. The reconstruction of the images by the early and late DCNN models is shown in Fig. 2. The early DCNN is able to more faithfully reconstruct the image compared to the late DCNN. To examine certain features of interest, we set all other channels to zero and passed the feature maps to the trained DCNN models. Figure 3 shows four different features that were visualized by the early DCNN and Figure 4 shows features that were visualized by the late DCNN. Based on these visualizations. We see in Figure 3 that the CNN uses early layers to detect basic features such as lines, contours, colours and textures within an image, while the later layers encode more obscure colour maps to an image.

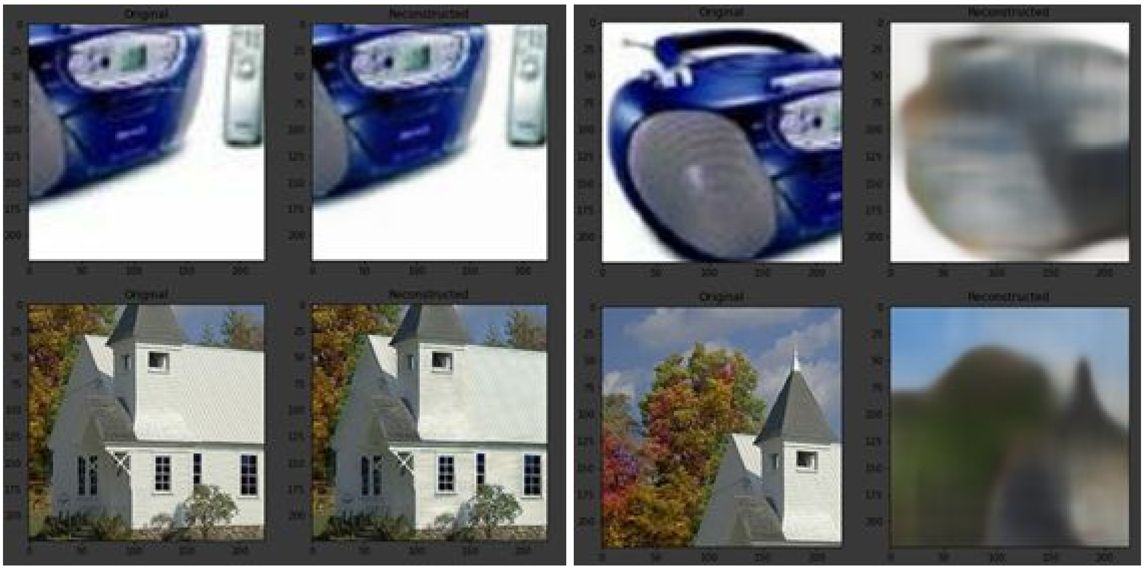
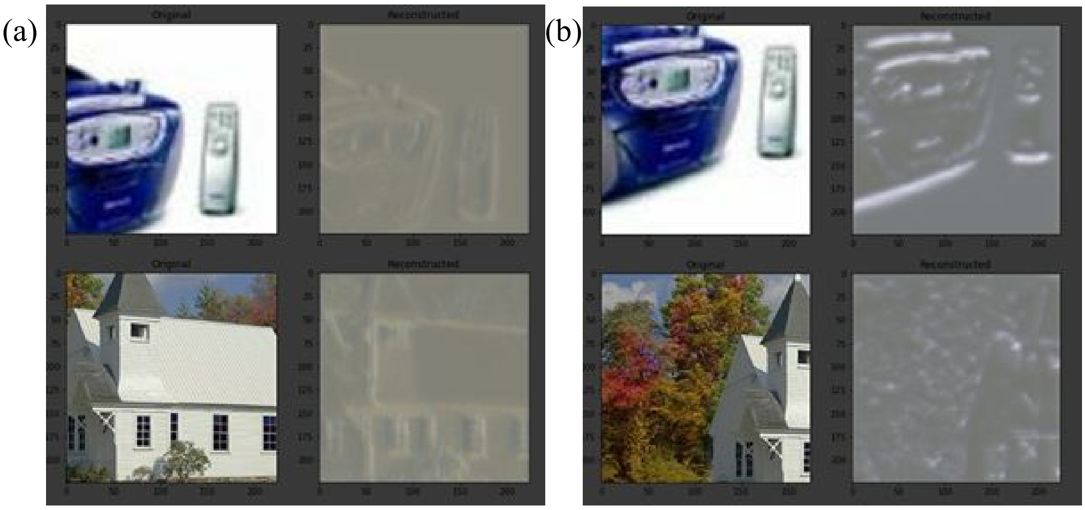


Figure 2. Reconstruction of the images by early (left) and late (right) DCNN models. In each of the reconstruction figures, the left two are the original images, and the right two are the reconstructed images.



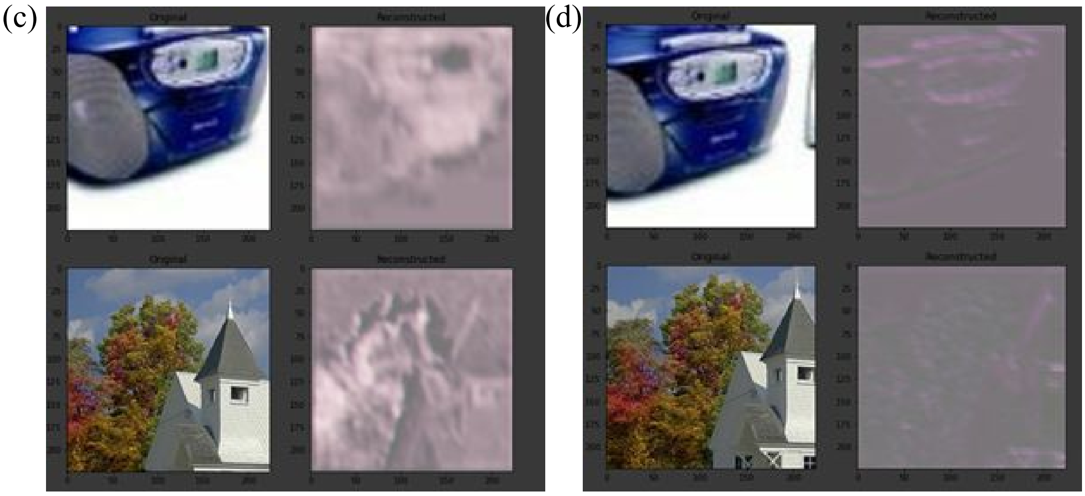


Figure 3. Visualizations of four different features by the early DCNN. In each of the figures, the original images are on the left, and the reconstructed images are on the right. These CNN features detect edges in (a), horizontal lines in (b), the colour green in (c), and textures in (d).

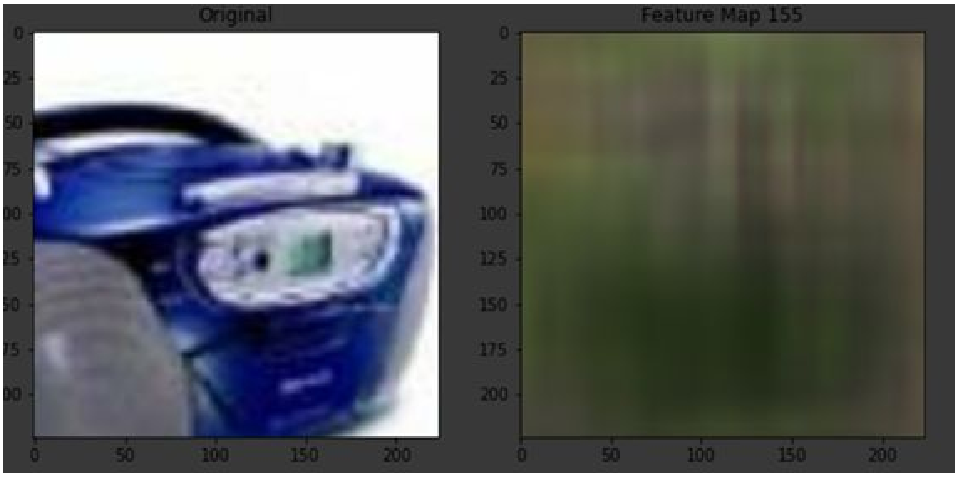


Figure 4. Visualizations of a feature generated by the late DCNN.

Discussion

Earlier results showed that earlier convolution layers of the CNN responded to things like edges or contours, textures and colours (Figure 3), but features in later layers of a CNN encode abstract colour maps to an image. Also, the reconstructions of images, from earlier features of the CNN had a higher quality than those from the features in later layers in the CNN. This could be the case, since the earlier feature maps are less processed, therefore closer to the raw data, when compared to the later feature maps. The later feature maps may be so compressed that they no longer represent the full detail of the image, but rather they encode a semantic representation of the data.

It was found in previous studies, a feature that the LGN in cats detects are lines and contours that are in specific orientations (Hubel & Wiesel, 1959). They did this by passing lines across a blank screen and measured the firing rates of neurons in different regions of the LGN (Hubel & Wiesel, 1959). With this they were able to test more complicated contours to see what caused greater firing rates in the certain regions of the cat LGNs (Hubel & Wiesel, 1959). Some of the early layers of a CNN do appear to detect these features. We can see in Figures 3a and 3b that the CNN is able to detect lines and contours to highlight the edges within an image (Figure 3a) and that these detected contours are sensitive to orientation (Figure 3b).

Color processing in the human brain relies on the three types of cone photoreceptors, which are short, medium, and long wavelength sensitive. Low level visual processing plays an important role in signal processing of color vision. The LGN neurons, whose axons project to layers of V1, provide significant signals for color vision (Chatterjee & Callaway, 2003). Neurons in V1 hold different properties of receptive fields. While the majority of the cortical neurons respond well to the orientation of edges, only a few of the cortical neurons respond to the spatially uniform stimulation (Solomon, S. G., & Lennie, 2007). Approximately 5 - 10% of neurons in V1 respond weakly to colorless modulation, but strongly to pure color modulation, which are important for color vision (Solomon, S. G., & Lennie, 2007). Also, previous study (Porter, et al., 2012) revealed that the earliest eyes of the common ancestors of metazoans are able to be selective to the wavelength information. Additionally, it was found that the wavelength processing coexists with the color processing in some crustaceans (Porter, et al., 2012). At the same time, earlier study reported response to colors by the neurons that are responsible for tracking objects through wavelength processing in dragonflies (Horridge, et al., 1990). Compared to the color detection in CNN (Fig. 3c), we can see that CNN is also able to respond to certain colors, which is similar to the selection of wavelengths. In this case, the green color was highlighted in the reconstruction.

We successfully implemented the DCNN and used that to visualize the CNN. However, due to the large number of models we have and the simultaneous training of the models, google colab kept throwing memory errors, which happened a lot and we always had to reset the colab and did the training again. We could have gone better if we had a more stable server as the trouble from google colab cost lots of time. To further improve it, we want to use several more different feature visualization techniques and see if we'll have some new findings by comparing them to the biological vision.

Reference

Chatterjee, S., & Callaway, E. M. (2003). Parallel colour-opponent pathways to primary visual cortex. *Nature,* 426(6967), 668-671.

Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, *148*(3), 574-591.

Horridge, G. A., Wang, X., & Zhang, S. W. (1990). Colour inputs to motion and object vision in an insect. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *329*(1254), 257-263.

Krauskopf, J., & Karl, G. (1992). Color discrimination and adaptation. *Vision research,* 32(11), 2165-2175.

Lindsay, G. (2020). Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *Journal of Cognitive Neuroscience,* 1-15.

Porter, M. L., Blasic, J. R., Bok, M. J., Cameron, E. G., Pringle, T., Cronin, T. W., & Robinson, P. R. (2012). Shedding new light on opsin evolution. *Proceedings of the Royal Society B: Biological Sciences*, *279*(1726), 3-14.

Solomon, S. G., & Lennie, P. (2007). The machinery of colour vision. *Nature Reviews Neuroscience,* 8(4), 276-286.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011, November). Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision* (pp. 2018-2025). IEEE.