Chapter 1

Introduction

It has long been the goal of computer vision researchers to be able to develop systems that can reliably recognize objects in a scene. Achieving this unlocks a huge range of applications that can benefit society as a whole. From fully autonomous vehicles, to automatic labelling of uploaded videos/images for searching, or facial recognition for identification and security, the uses are far reaching and extremely valuable. The challenge does not lie in finding the right application, but in the difficulty of training a computer to *see*.

There are nuisance variables such as changes in lighting condition, changes in viewpoint and background clutter that do not affect the scene but drastically change the pixel representation of it. Humans, even at early stages of their lives, have little difficulty filtering these out and extracting the necessary amount of information from a scene. So to design a robust system, it makes sense to design it off how *our* brains see.

Unfortunately, vision is a particularly complex system to understand. It has more to it than the simply collecting photons in the eye. An excerpt from a recent Neurology paper [1] sums up the problem well:

> It might surprise some to learn that visual information is significantly degraded as it passes from the eye to the visual cortex. Thus, of the unlimited information available from the environment, only about 10^{10} bits/sec are deposited in the retina ... only $\sim 6 \times 10^6$ bits/sec leave the retina and only 10^4 make it to layer IV of V1 [2], [3]. These data clearly leave the impression that visual cortex receives an impoverished representation of the world ... it should be noted that estimates of the bandwidth of conscious awareness itself (i.e., what we 'see') are in the range of 100 bits/sec or less[2], [3].

Current digital cameras somewhat act as a combination of the first and second stage of this system, collecting photons in photosensitive sensors and then converting this to an image on the order of magnitude of 10^6 pixels (slightly larger but comparable to the 10^6 bits/sec travelling through the optic nerve).

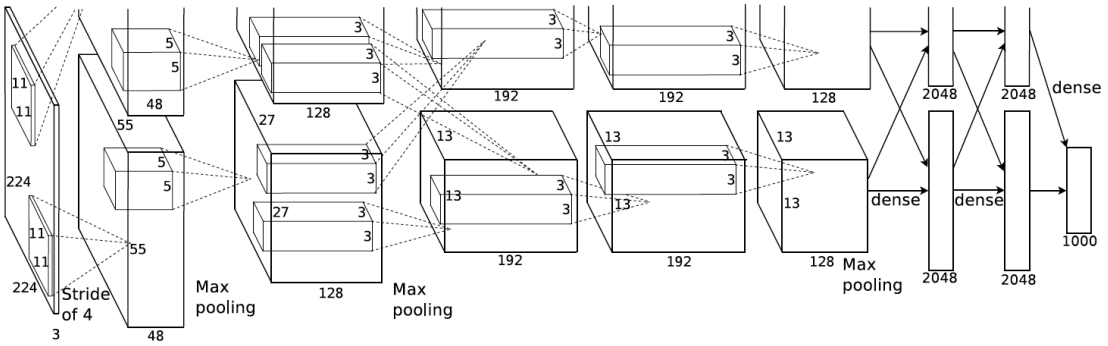


Figure 1.1: **Convolutional Architecture of [8].**

If we are to build effective vision systems, it makes sense to emulate this compression of information. The question now stands before us — what information is kept on entry to the V1 cortex? Hubel and Wiesel revolutionized our understanding of the V1 cortex in the 50s and 60s by studying cats [4], [5], macaques and spider monkeys [6]. They found that neurons in the V1 cortex fired most strongly when edges of a particular (i.e., neuron-dependent) orientation were presented to the animal, so long as the edge was inside the receptive field of this neuron. Continued work on their experiments by Blakemore and Cooper [7] showed, by exposing kittens to controlled environments in which they only saw horizontal and vertical lines, that these early layers of perception are in fact *learned*.

1.1 Convolutional Neural Networks

The current state of the art in image understanding sysems are Convolutional Neural Networks (CNNs). These are a learned model that stacks many convolutional filters on top of each other separated by nonlinearities. They are seemingly inspired by the visual cortex in the way that they are hierarchically connected, progressively compressing the information into a richer representation.

Figure 1.1 shows an example architecture for the famous AlexNet [8]. Inputs are resized to a manageable size, in this case 224×224 pixels. Then multiple convolutional filters of size 11×11 are convolved over this input to give 96 output *channels* (or *activation maps*). In the figure, these are split onto two graphics cards or GPUs for memory purposes. These are then passed through a a pointwise nonlinear function, or a *nonlinearity*. The activations are then pooled (a form of downsampling) and convolved with more filters to give 256 new channels at the second stage. This is repeated 3 more times until the 13×13 output with 256 channels is unravelled and passed through a fully connected neural network to classify the image as one of 1000 possible classes.

CNNs have garnered lots of attention since 2012 when the previously mentioned AlexNet nearly halved the top-5 classification error rate (from 26% to 16%) on the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [9][1]. In the years since then, their complexity has grown significantly. AlexNet had only 5 conovlutional layers, whereas the 2015 ILSVRC winner ResNet [15] achieved 3.57% top-5 error with 151 convolutional layers (and had some experiments with 1000 layer networks).

1.2 Issues with CNNs

Despite their success, they are often criticized for being *black box* methods. You can view the first layer of filters quite easily (see Figure 1.2a) as they exist in RGB space, but beyond that things get trickier as the filters have a third, *depth* dimension typically much larger than its two spatial dimensions. Additionally, it is not clear what the input channels themselves correspond to. For illustration purposes, we have also shown some example activations from the first three convolutional layers for AlexNet in Figure 1.2(b)-(d)[2]. We can see in Figure 1.2b that in the conv1 activations, some of the first layer channels are responding to edges or colour information, but as we go deeper to conv2 and conv3, it becomes less and less clear what each activation is responding to.

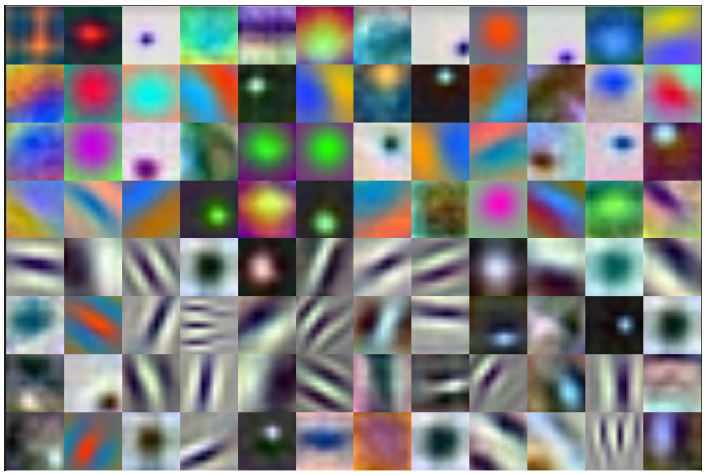
Aside from their lack of interpretability, it takes a long time and a lot of effort to train state of the art CNNs. Typical networks that have won ILSVRC since 2012 have had roughly 100 million parameters and take up to a week to train. This is optimistic and assumes that you already know the necessary optimization or architecture hyperparameters, which you often have to find out by trial and error. In a conversation the author had with Yann LeCun, the attributed father of CNNs, at a Computer Vision Summer School (ICVSS), LeCun highlighted this problem himself:

> There are certain recipes (for building CNNs) that work and certain recipes that don't, and we don't know why.

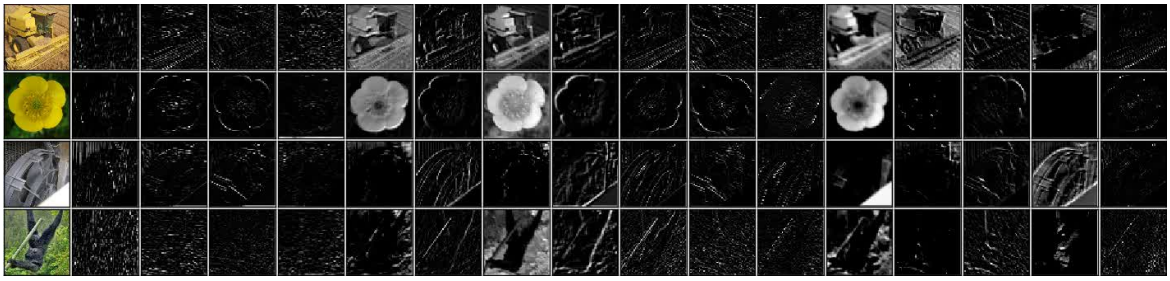
Considering the recent success of CNNs, it is becoming more and more important to understand *how* and *what* a network learns, which has contributed to it making its classification or regression choice. Without this information, the use of these incredibly powerful tools could be restricted to research and proprietary applications.

[1] The previous state of the art classifiers had been built by combining keypoint extractors like SIFT[10] and HOG[11] with classifiers such as Support Vector Machines[12] and Fisher Vectors[13], for example [14].

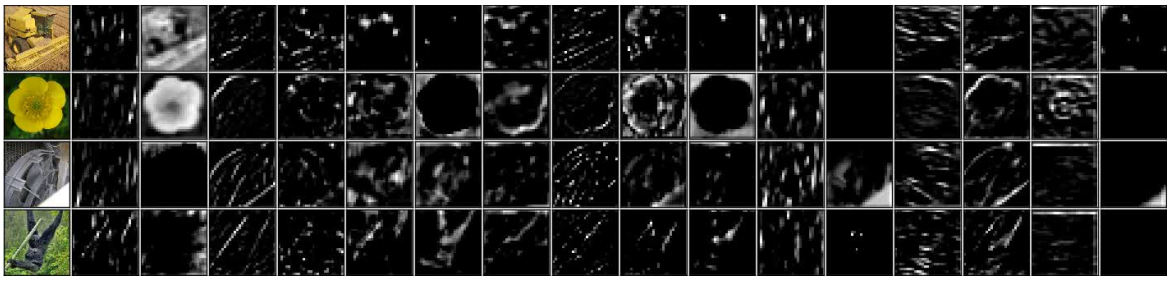
[2] These activations are taken after a specific nonlinearity that sets negative values to 0, hence the large black regions.



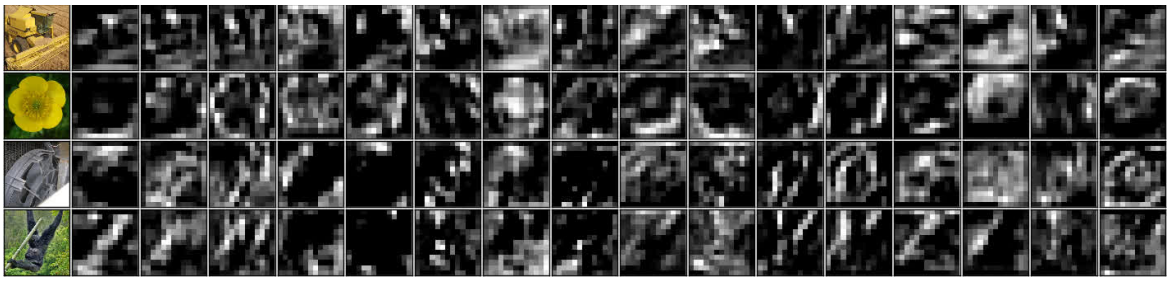
(a) conv1 filters



(b) conv1 activations



(c) conv2 activations



(d) conv3 activations

Figure 1.2: **The first layer filters learned by AlexNet and the first three layer's activations.** (a) The 11×11 filters for the first stage of AlexNet. Of the 96 filters, 48 were learned on one GPU and another 48 on another GPU. Interestingly, one GPU has learned mostly lowpass/colour filters and the other has learned oriented bandpass filters. (b) - (d) Randomly chosen activations from the output of the first, second and third convolutional layers of AlexNet (see Figure 1.1) with negative values set to 0. Filters and activation images taken from supplementary material of [8].

1.3 An Interesting Result - The Learned Wavelet Transform

The structure of convolutional layers is fairly crude in terms of signal processing - arbitrary taps of an FIR filter are learned typically via gradient descent to minimize either a mean-squared error or cross entropy loss. It is perhaps surprising and motivating then that the filters we saw earlier learned by the first layer of a CNN (Figure 1.2) look like oriented wavelets. From a biological point of view this makes sense, as it matches the earlier mentioned results by Hubel and Wiesel. From a signal processing point of view this also makes sense, as the wavelet transform is a powerful and stable way to split up an image into areas of the frequency domain. However, there was no prior placed on the filters to make them have this similarity to wavelets. There were no constraints on vanishing moments[16] or smoothness, or even for the 'conv1' activations to be sparse; the only constraint they had was an ℓ_2 penalty to avoid large filter taps.

1.4 Project Motivation

This leads us to ask a motivating question:

> Is it possible to learn convolutional filters as combinations of basis functions rather than individual filter taps?

However in achieving this, it is important to find ways to have an adequate richness to filtering while reducing the number of parameters needed to specify this. We want to contract the space of learning to a subspace or manifold that is more useful. In much the same way, the convolutional layer in a CNN is a restricted version of a fully connected layer in a Multilayer Perceptron, yet adding this restriction allowed us to train more powerful networks.

The intuition that we explore in this thesis is that *complex wavelets* are the right basis functions for convolutional filtering in CNNs. We have already seen in the previous section that they would do well in replacing the first layer of a CNN, but can they be used at deeper layers? Their well understood and well defined behaviour would help us to answer the above *how* and *why* questions. Additionally, they allow us to enforce a certain amount of smoothness and near orthogonality; smoothness is important to avoid sensitivity to adversarial or spoofing attacks [17] and near orthogonality allows you to cover a large space with fewer coefficients.

But first we must find out *if* it is possible to get the same or near the same performance by using wavelets as the building blocks for CNNs, and this is the core goal of this thesis.

1.5 ScatterNets

To explore this intuition, we begin by looking at one of the most popular current uses of wavelets in image recognition tasks, the Scattering Transform. The Scattering Transform, or the *ScatterNet*, was introduced in [18], [19] at the same time as AlexNet. It is a non black box network that can be thought of as a restricted complex valued CNN [20]. Unlike a CNN, it has predefined convolutional kernels, set to complex wavelet (and scaling) functions. Due to its well-defined structure, it can be analyzed and bounds on its stability to shifts, noise and deformations are found in [18].

For a simple task like identifying small handwritten digits the variablities in the data are simple and small and the ScatterNet can easily reduce the problem into a space which a Gaussian SVM can easily solve [19]. For a more complex task like identifying real world objects, the ScatterNet can somewhat reduce the variabilities and get good results with an SVM, but there is a large performance gap between this and what a CNN can achieve [21].

1.6 Layout

This thesis has one literature review chapter and four work chapters:

- **??** explores some of the background necessary for starting to develop image understanding models. In particular, it covers the inspiration for CNNs and the workings of CNNs themselves, as well as covering the basics of wavelets and ScatterNets.
- **??** proposes a change to the core of the ScatterNet. In addition to performance issues with ScatterNets, they are slow and memory expensive to run. This in itself is enough of an issue to prevent them from ever being used as part of deep networks. To overcome this, we change the computation to use the DT$\mathbb{C}$WT [22] instead of Morlet wavelets, achieving a 20 to 30 times speed up.
- **??** describes our *DeScatterNet*, a tool used to interrogate the structure of ScatterNets. We also perform tests to determine the usefulness of the different scattered outputs finding that many of them are not useful for image classification.
- **??** describes the learnable ScatterNet we have developed to address some of the issues found from the interrogation in **??**. This is the first
- In **??**, we step away from ScatterNets, and look at using the wavelet space as a latent space to learn representations. We find possible nonlinearities and describe how to learn in both the pixel and wavelet domain.

1.6.1 Contributions

The key contributions of this thesis are:

- **Software for wavelets and** DT$\mathbb{C}$WT **based ScatterNet (chapter 3)**
- **ScatterNet analysis and visualizations (chapter 4)**. Presented at MLSP2017, this chapter
- **Invariant Layer/Learnable ScatterNet (chapter 5)** Presenting at ICIP2019.
- **Learning convolutions in the wavelet domain (chapter 6)**.

1.6.2 Desirable Properties

Unlike CNNs introduced earlier which have little prior constraints (apart from the commonly used L_2 regularization), the scattering operator may be thought of as an operator S that imposes structural priors on learning by extracting features with manually chosen, desirable properties. The extracted features can be used In classical paradigms of image understanding, it makes sense to add these priors, but it remains yet to be shown that these help learning.

limit variability these properties areview on these properties are manually chosen with the ultimate goal of aiding image understanding.

Chapter 2

Conclusion

This chapter aims to logically tie together the results from the previous chapter, outlining what has been promising and what has not been, offering explanations as to why we think that is the case.

References

[1] M. E. Raichle, "Two views of brain function", eng, *Trends in Cognitive Sciences*, vol. 14, no. 4, pp. 180–190, Apr. 2010.

[2] C. H. Anderson, D. C. Van Essen, and B. A. Olshausen, "Directed Visual Attention and the Dynamic Control of Information Flow", en, in *Neurobiology of Attention*, Elsevier, 2005, pp. 11–17.

[3] Tor Nørretranders, *The User Illusion*. Viking, 1998.

[4] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex", *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, Oct. 1959.

[5] ——, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex", eng, *The Journal of Physiology*, vol. 160, pp. 106–154, Jan. 1962.

[6] ——, "Receptive fields and functional architecture of monkey striate cortex", eng, *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, Mar. 1968.

[7] C. Blakemore and G. F. Cooper, "Development of the Brain depends on the Visual Environment", en, *Nature*, vol. 228, no. 5270, pp. 477–478, Oct. 1970.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in *NIPS*, Curran Associates, Inc., 2012, pp. 1097–1105.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *arXiv:1409.0575 [cs]*, Sep. 2014. arXiv: 1409.0575 [cs].

[10] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", en, Jun. 2005.

[12] C. Cortes and V. Vapnik, "Support-vector networks", en, *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[13] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice", en, *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, Dec. 2013.

[14] J. Sanchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification", in *CVPR 2011*, Jun. 2011, pp. 1665–1672.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. arXiv: 1512.03385.

[16] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks", *arXiv:1312.6199 [cs]*, Dec. 2013. arXiv: 1312.6199 [cs].

[18] S. Mallat, "Group Invariant Scattering", en, *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, Oct. 2012.

[19] J. Bruna and S. Mallat, "Invariant Scattering Convolution Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[20] J. Bruna, S. Chintala, Y. LeCun, S. Piantino, A. Szlam, and M. Tygert, "A mathematical motivation for complex-valued convolutional networks", *arXiv:1503.03438 [cs, stat]*, Mar. 2015. arXiv: 1503.03438 [cs, stat].

[21] E. Oyallon and S. Mallat, "Deep Roto-Translation Scattering for Object Classification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2865–2873.

[22] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The dual-tree complex wavelet transform", *Signal Processing Magazine, IEEE*, vol. 22, no. 6, pp. 123–151, 2005.