Chapter 1

Introduction

It has long been the goal of computer vision researchers to be able to develop a system that can recognize objects in a scene. Achieving this unlocks a huge range of applications that can benefit society as a whole. A recently realized example is in autonomous vehicles, which are starting to be able to detect lane markers and vehicles on the road, and act as a supervisor for human drivers who, while highly intelligent, are fallible. Perhaps it could also be used to label some of the billions of images that get uploaded to the internet daily, making them searchable. Or be a hypervisor for surveillance and security systems, forwarding only an important subset of the feed to human experts, reducing strain and tedium.

These are only simple examples, and there are many more; the challenges that face computer vision are not finding applications for it, but 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 certainly drastically change a 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 somewhat simple, habitual collecting of photons in the eye. An excerpt from a recent Neurology paper sums up the problem well[1]:

> 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. Because of a limited number of axons in the optic nerves (approximately 1 million axons in each) 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, a subject of more than passing interest to those interested in the processing of visual information[4].

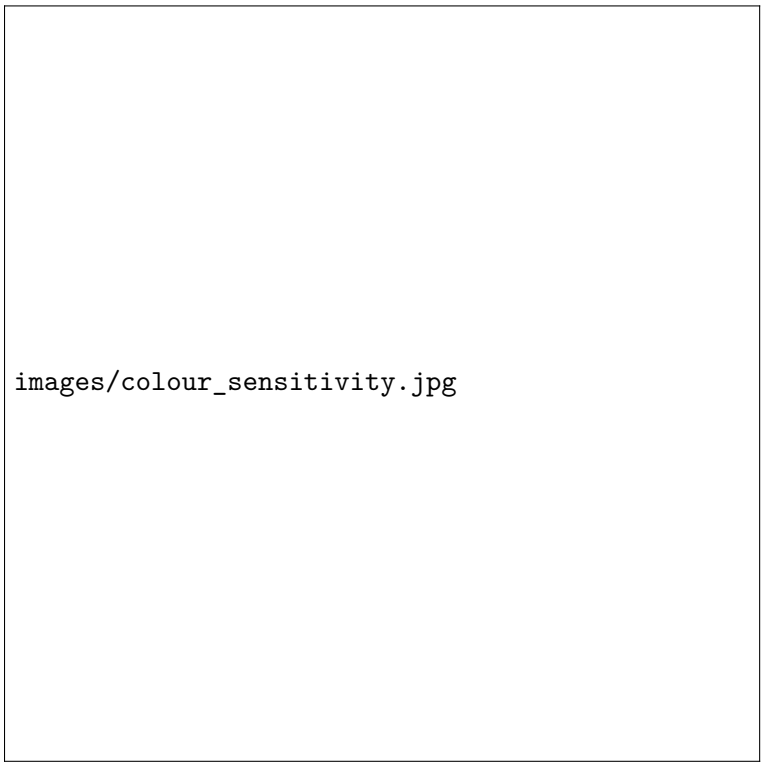

Figure 1.1: Wavelength responsiveness of the different photoreceptors in the eye. S, M, and L are short, medium, and long cones, compared to R — rods. Taken from bowmaker_visual_1980

> Parenthetically, 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).

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 [5], [6], macaques and spider monkeys [7]. 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 blakemore_development_1970 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 The Current State of the Art - Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have recently become the de facto standard tool to build any type of vision model. They are seemingly inspired by the visual cortex in the way that they are hierarchically connected, progressively compressing the information into a richer representation.

They are a supervised learning model that stacks many learned convolutional filters on top of each other to extract features, before using a densely connected Neural Network to combine learned features and provide the desired classification or detection output.

They have moved spectacularly quickly from the doldrums to the spotlight, once they were able to nearly halve (from 26% to 16%) classification error rates on the ImageNet Large Scale Visual Recognition Competition in 2012 [8]. In just three years since then, their complexity has grown exponentially, from eight layers to over 150[1], and their error rate has dropped to 3.6%.

Despite their success, they are often criticized for being 'black box' methods. Indeed, once you train a CNN, you can view the first layer of filters quite easily — see Figure 1.2 — as they exist in RGB space. Beyond that things get trickier, as the filters have a third, 'depth' dimension typically much larger than its two spatial dimensions.

1.2 Problems with CNNs and Project Motivation

One of the key benefits of CNNs is their ability to take raw images and learn features — i.e., they do not require you to construct feature vectors. The paradigm that they are built on is really quite elegant — you set up a network, you feed in your images, you define a cost function, and then you propagate error gradients to learn your representation. Unfortunately, the process rarely goes so smoothly. There are entire sets of hyperparameters that must be searched to find optimal solutions. Some hyperparameters (such as learning rate and initialization weights), can be so sensitive as to cause the network to diverge if they are set outside of a small range. Others, such as network depth and width often have counter-intuitive effects on the result. A commonly known one is that increasing the depth of a network often causes the accuracy to decrease, even though it should at worst remain constant, as the network could learn the identity mapping for deeper layers.

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] Research groups have even started designing networks 1000+ layers deep, but have had difficulty getting improvements from them.

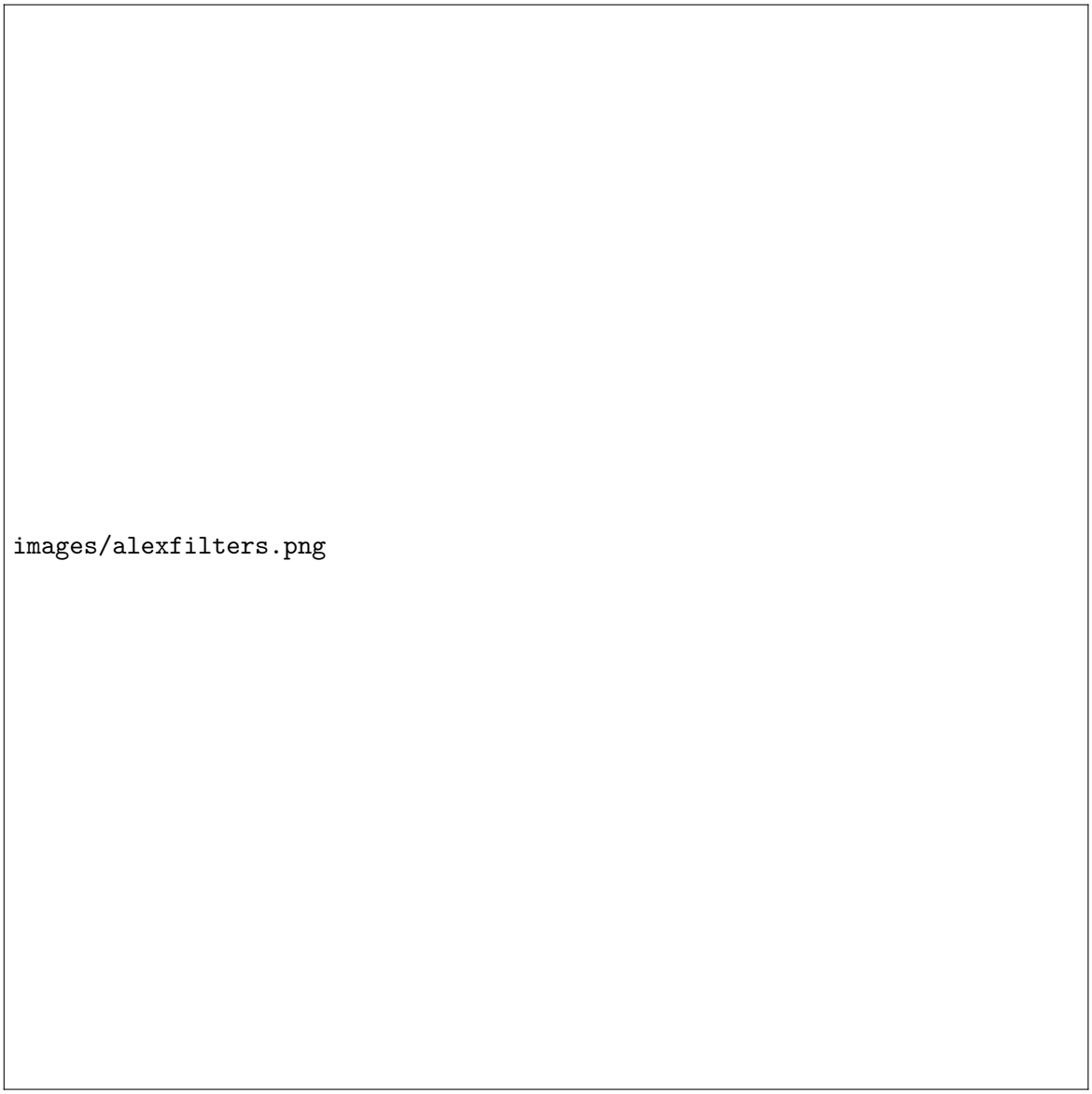

Figure 1.2: The first layer filters learned by AlexNet. The architecture was learned on 2 GPUs with limited cross-connectivity. The colour filters in the top half were learned one GPU and the edge filters in the bottom half were learned on the other. Weights taken from krizhevsky_imagenet_2012

Figure 1.3: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are 'Electric Guitar' (p=0.32), 'Acoustic Guitar' (p=0.24) and 'Labrador' (p=0.21). The regions which contributed to these predictions are shown in (b) through (d)

As an example[2], it is not hard to imagine a deep network that could be used to assess whether giving a bank loan to an applicant is a safe investment. It could compare a vector of their current and past financial situation to a dataset of others and choose a simple yes/no answer. Trusting a black box solution is deeply unsatisfactory in this situation. Not only from the customer's perspective, who, if declined, has the right to know why goodman_european_2016, but also from the bank's — before lending large sums of money, most banks would like to know why the network has given the all clear. 'It has worked well before' is a poor rule to live by.

A recent paper titled 'Why Should I Trust You?' ribeiro_why_2016 explored this concept in depth. They rated how highly humans trusted machine learning models. Unsurprisingly, a model with an interpretable methodology was trusted more than those which did not have one, even if it had a lower prediction accuracy on the test set. One of their measures of 'Interpretability' for image classifiers involved finding the regions of the input image that caused a high output for that particular class — see Figure 1.3.

1.3 Project Aims

The overarching goal is to achieve better image understanding systems. Presently, CNNs appear to be the best approach to achieving this but a lot of their power and expressibility is not clear, or at least not well understood. Even worse — there are few heuristics or methodologies for designing them. In a conversation the author had with Yann LeCun, the attributed father of CNNs, at a Computer Vision Summer School, 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.

We are not the first research group to be unsatisfied with not knowing the mechanics of CNNs. There has been a lot of very impressive work done recently on trying to visualize the response a CNN has to a given input, in particular the work done by zeiler_visualizing_compact_2014, which builds 'Deconvolutional Neural Networks' (deconvnets) to view the regions of the input image that cause large responses at deeper layers of a

[2]While we limit the scope of our project to Imaging problems, CNNs can and have been used successfully in time series and language tasks.

CNN. This was the key tool in the previously mentioned 'Why Should I Trust You?' paper ribeiro_why_2016 to find the regions of the image that make up Figure 1.3.

So far these tools, while useful, stop short of turning visualization around into an improved strategy for designing networks. This brings us to the main goal of our research:

We hope to further research into understanding *how* and *what* deep networks learn by building a well understood and well-defined network that mimics their operation. Intuition is the primary goal, and with that, we believe improved performance will follow.

We do not wish to focus only on purely handcrafted methods, nor on purely learned methods. Neither is discounted, and a hybrid of the two seems to be a good way to achieve our goal.

In particular, the starting point for our project is clear — the filters learned by the first layer of a CNN, shown in Figure 1.2, look like oriented wavelets. This is not surprising either from a biological point of view — as it matches the earlier mentioned results by Hubel and Wiesel, or from a signal processing point of view — as the wavelet transform is a powerful and stable way to split up an image into areas of the frequency domain.

With this, the above goal can then be refined to a more targeted one:

Starting with a wavelet transform, we want to look at how to develop a higher order system by drawing inspiration from the second and third layers of a CNN. Achieving this will shed light on what is learned and what is needed for a good image recognition system.

The second inspiration for a starting point is the recent work in developing *Scatternets* by Stephané Mallat, one of the forefathers of the wavelet transform, and his research group. Their work attempts to do precisely what we want — start with a wavelet transform that is sensitive to edges, and then build deeper layers on top of this that are sensitive to larger shapes, while being insensitive to uninformative variations such as translation, rotation, and scale. Their Scatternets are purely deterministic.

1.4 Layout

The layout of the report is as follows. **??** 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.

?? covers the wavelet transform used by Mallat, and compares it to the preferred Dual-Tree Complex Wavelet Transform (DTCWT) by Kingsbury kingsbury_complex_2001.

?? reviews in depth the Scatternet designs by Mallat et. al. This is the last chapter of literature review, before **??** which starts to explore our work and analysis done on these Scatternets. In particular, we swap out the Morlet wavelets from Mallat's Scatternet to the faster, separable DTCWT wavelets, but we also make changes to the design of the Scatternet and look at how to apply the principles of visualizations, like those in the work by zeiler_visualizing_compact_2014.

?? then explores our recent work in attempting to combine Scatternets with CNNs. The inspiration for this being the idea that a well designed tool like the Scatternet, followed by a shallower CNN, should be equivalent to or better than a deeper CNN.

chapter 2 then summarizes our findings so far, discusses and analyses the results, and lays out the plan for the remainder of the project.

This work is stimulated by the intuition that wavelet decompositions, in particular complex wavelet transforms, are good building blocks for doing image recognition tasks. Their well understood and well defined behaviour as well as the similarities seen in learned networks, implies that there is potential gain for thinking about CNN layers in a new light.

To explore and test this intuition, we begin by looking at one of the most popular current uses of wavelets in image recognition tasks, in particular the Scattering Transform.

1.5 Series Expansions of Signals

Look at the intro to Vetterli's book. Want to make a statement about expanding signals in some form or another.

1.6 Contributions

The contributions and layout 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)**.

1.6.1 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] B. A. Olshausen and D. J. Field, "How close are we to understanding v1?", eng, *Neural Computation*, vol. 17, no. 8, pp. 1665–1699, Aug. 2005.

[5] 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.

[6] ——, "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.

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

[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.