Mathematisch-Naturwissenschaftliche Fakultät

Lernbasierte Computer Vision

Bachelorarbeit

Fancy Title

Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Autonomous Computer Vision
Marius Hobbhahn, marius.hobbhahn@student.uni-tuebingen.de, 2018/19

Bearbeitungszeitraum:	von-bis
Betreuer/Gutachter:	Prof. Dr. Andreas Geiger, Universität Tübingen
Zweitgutachter:	Dr. Benjamin Coors, Universität Tübingen

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Bachelorarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Marius Hobbhahn (Matrikelnummer 4003731), April 24, 2019

Abstract

TODO

Zusammenfassung

TODO

Contents

1 Introduction

Object detection describes the task of detecting instances of semantic objects in visual data, i.e. images and videos in two or three dimensions. Even though the task is very easy for humans in most situations, it is very hard for computers. However, in recent years object detection algorithms have gotten significantly better for many different applications like face recognition, object tracking (e.g. the ball in a football match) and especially semantic segmentation of traffic scenes, pedestrian and car tracking.

For most state of the art (SOTA) object detection algorithm convolutional neural networks (CNNs) are used. In many implementations the filters used in those CNNs are all trained during the training period. [BM12] introduced a new technique called the Scattering Transform that uses wavelet operations on the image and performs classification tasks on those. They also show that the technique is essentially equivalent to using CNNs with fixed weights for some or all filters. It has been applied successfully in a variety of tasks. [SM13] showed that the scattering transform is applicable to texture discrimination. [OM14] have demonstrated that the scattering transform also produces results similar to other SOTA algorithms for unsupervised learning. [ACC+17] improved the classification of diseases from neuroimages considerably. Lastly, [OBZ17] shows that substituting the first layer filters of CNN approaches with the scattering transform yields equivalent results compared to these filters being trained.
The reason why the scattering transform has proven so successful are the properties it provides. It is invariant to rotation, translation and scaling. These properties are important for image classification but also necessary for object detection. For example, when detecting pedestrians in real traffic situations, the object detection algorithm must be able to identify them independent of their location, size or rotation within the image.
This work tries to harvest the useful properties of the scattering transform and combine it with already established state of the art object detection algorithms. This will be done primarily in two ways. First, the techniques are combined sequentially,

i.e. the SOTA algorithms are applied only on the outputs of the scattering transform. [OBZ17] have already shown that sequential combination is able to produce SOTA results for image recognition. This is the attempt to reproduce these findings for object detection.
Second, the techniques are combined in parallel, i.e. the information of the scattering transform are used as additional inputs for the object detection algorithm or merged at later stages. This has not been tested yet and is the primary extension of the just described related work.
If the approaches are successful that has three specific advantages. First, the scattering transform might yield information that were currently not available to the network and therefore increasing its accuracy. Even if that might only be a marginal increase, it is meaningful for application. Every little reduction of the error in object detection, especially for autonomous driving, means a reduction of risk of self driving cars. This is directly translated to lives being saved in the longterm. Second, fixed weights imply no additional training time for them. If, for example, one layer can be substituted that would reduce the length of training and save cost and energy while creating access for people who currently do not own multiple GPUs. Third, fixed weights cannot be overfit and are very maximally general. This might produce more robust algorithms and protect against black box attacks or other malicious practices applied to CNNs. This, however, will not be tested within the scope of this work but might be interesting follow-up.

2 Theory

2.1 2D Image Processing

Image processing describes the application of different algorithms on images with the purpose of gaining certain information about it or changing its representation. Most of the time images are given as two dimensional pixel arrays where each entry denotes the intensity of that pixel. In the case of grayscale images the value is between 0 and 255 representing black and white respectively. When handling color images an additional 3rd dimension is added with three channels representing a red, green, blue (rgb) encoding. Each entry, again, has values between 0 and 255 representing color intensity.
Instead of imagining an image as a flat 2D object, it can also be seen as a terrain with surface, where the height of each coordinate is determined by the intensity of its value. An example of this is shown in figure 2.1.

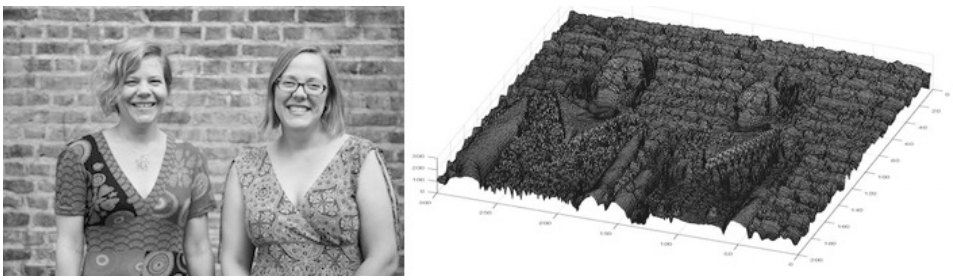

Figure 2.1: Left: image represented as 2D flat surface. Right: image as 3D terrain with uneven surface. [1]

Like every other surface, these images can now be approximated as the sum of many different two dimensional sine waves. 2D sine waves are defined as in equation 2.1, where f a is the amplitude and h,k are the frequencies in x and y direction respectively.

[1] Figure taken from `https://plus.maths.org/content/fourier-transforms-images`

$$f = a\sin(h \cdot x + k \cdot y) \tag{2.1}$$

To give an example of how this approximation looks like, figure 2.2 shows examples of three different two dimensional sine waves. It can be observes that higher amplitudes dominate the resulting wave, i.e. determine the direction of the wave stronger than the smaller amplitudes.

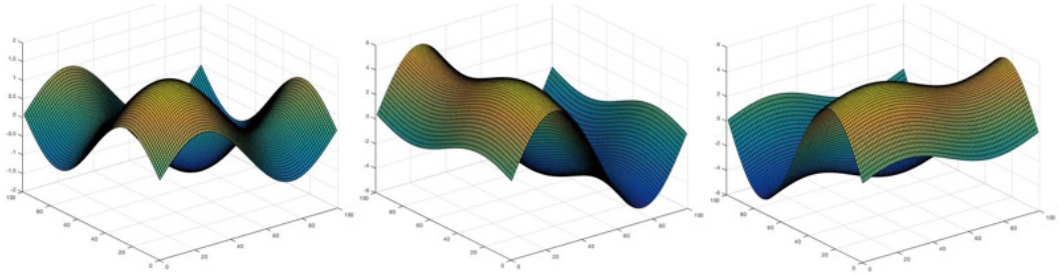

Figure 2.2: Left: $\sin(x)+\sin(y)$. Middle: $5\sin(x)+\sin(y)$. Right: $\sin(x)+5\sin(y)$. On the middle and right images the higher amplitudes of 5 dominate the resulting wave. [2]

2.2 Fourier Transformation

A Fourier Transform (FT) decomposes a signal into the frequencies that make it up.

2.2.1 One dimensional FT

In the case of one dimensional signals the decomposition are the coefficients of the sine waves representing the signal. A good example of this would be the decomposition of a The FT is defined by equation 2.2 for any real number ω and any integrable function $f : \mathbb{R} \to \mathbb{C}$.

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \omega} dx \tag{2.2}$$

To get back to the Fourier domain when the given a frequency, the inverse Fourier transform defined in equation 2.3 is used.

[2] Figure taken from `https://plus.maths.org/content/fourier-transforms-images`

$$f(x) = \int_{-\infty}^{\infty} \tilde{f}(\omega)\, e^{2\pi i x\omega} d\omega \tag{2.3}$$

When using discrete instead of continuous functions, the integrals in the definitions become sums. Then the definition of the forward FT is given in equation 2.4 and in equation 2.5 for the inverse FT.

$$\tilde{f}(\omega) = \sum_{x=1}^{n} f(x)\, e^{-2\pi i x\omega} \tag{2.4}$$

$$f(x) = \sum_{x=1}^{n} \tilde{f}(\omega)\, e^{2\pi i x\omega} \tag{2.5}$$

2.2.2 Two dimensional FT

Since images are two dimensional objects the Fourier transform needs to be extended. The Fourier transform then becomes a complex function of two or more real frequency variables ω_1, ω_2. Since images images are finite objects the discrete version of the two dimensional Fourier transform is given in equation 2.6 for the forward case and in equation 2.7 for the inverse case.

$$\tilde{f}(\omega_1, \omega_2) = \sum_{x=1}^{n} \sum_{y=1}^{m} f(x, y)\, e^{-2\pi i(\omega_1 \cdot x + \omega_2 \cdot y)} \tag{2.6}$$

$$f(x, y) = \sum_{x=1}^{n} \sum_{y=1}^{m} \tilde{f}(\omega_1, \omega_2)\, e^{2\pi i(\omega_1 \cdot x + \omega_2 \cdot y)} \tag{2.7}$$

2.3 Object detection

Object detection is a task within image processing where objects on a given image are supposed to be detected. These objects can be anything from buildings over cars to humans. The images are already annotated for training, i.e. a rectangle (or other representation) that approximates the object best is already placed over the picture

with the associated class attached. An example of such an annotated image can be found in figure 2.3.

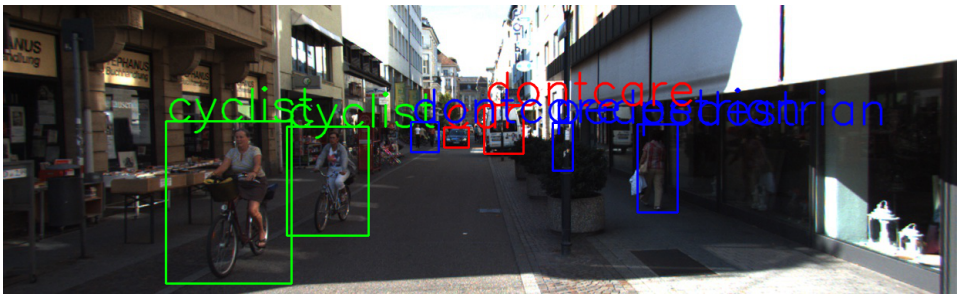

Figure 2.3: An example of a annotated image taken from the Kitti dataset. There are seven objects of four classes visible: cyclist, car, pedestrian and dontcare.

Many methods have been explored to increase the accuracy of object detection ranging from sliding window approaches (citation needed) over hand crafted feature extraction (citation needed) to artificial neural networks. All current state of the art results for object detection are achieved by using Convolutional Neural Networks (CNN) with minor or major adaptions (citation needed). Therefore this work also uses a CNN as the backbone of the object detection.

2.4 Convolutional Neural Networks

For most image-related tasks, i.e. classification or object detection, a picture is used as a collection of pixels. However, not all pixels are equally important and subsets of the entire image form meaningfully connected subcollections. This might be a face in a photo of a family gathering. For humans the ability to detect these features and contextualize them comes naturally, for computers it does not. Therefore convolutional neural networks (CNNs) [LBD+89] are used. Convolutions are essentially just the application of filters on an image. The filter is applied at every possible location in the image, as described in figure:

In CNNs there are multiple stages of filters in sequential order and multiple filters per layer. That means at every stage of the network different filters are applied on the outcome of an earlier step. The filters are assumed to learn different features of the images. The later the stage, the higher the level of complexity of the feature to be

[3]Figure taken from and animated version at `https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d`

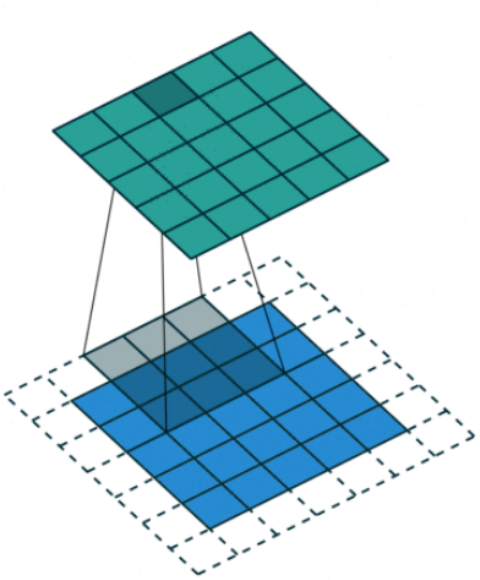

Figure 2.4: 3x3 convolution on 5x5 image. Resulting image is also 5x5 due to the padding of size 1 added on original image. [3]

learned. That means, that an early filter might learn simple attributes such as edges or colors while a later filter might learn more complex features such as an eye or a nose and the very last filters even more complex semantic objects such as a face. The weights that are adapted during training determine what each individual filter does. This means that every filter that is a applied on a given layer learns its specific function such that overall the best accuracy can be achieved.

As just explained in conventional CNNs the filters are trained. However, there are some approaches that use static filters. The Scattering Transform is one of those approaches. Given its unique properties the results might be comparable to the conventional approach. If weights are not trained a lot of time can be saved.

2.5 Scattering Transform

A transformation from the Fourier to the frequency domain cannot only be performed by using the sine but in principal with any given periodic function. Wavelets are wave-like oscillation with an amplitude that begin and end at zero. In most use cases wavelets are specifically crafted to have certain properties. The Scattering Transform is based on a Morlet wavelet, which is defined in equation 2.8.

$$\psi(u) = C_1(e^{iu.\xi} - C_2)e^{\frac{-|u|^2}{2\sigma^2}} \tag{2.8}$$

where C_2 is chosen such that $\int \psi(u)du = 0$. $u.\xi$ denotes the innerproduct of u and ξ and $|u|^2$ is the norm in $\mathbb{R}^2$. Figure 2.5 shows the 2 dimensional Morlet wavelet with parameters $\sigma = 0.85$ and $\xi = \frac{3\pi}{4}$. These parameters are taken from [BM12]. No additional fine tuning w.r.t. to these parameters is done in this work.

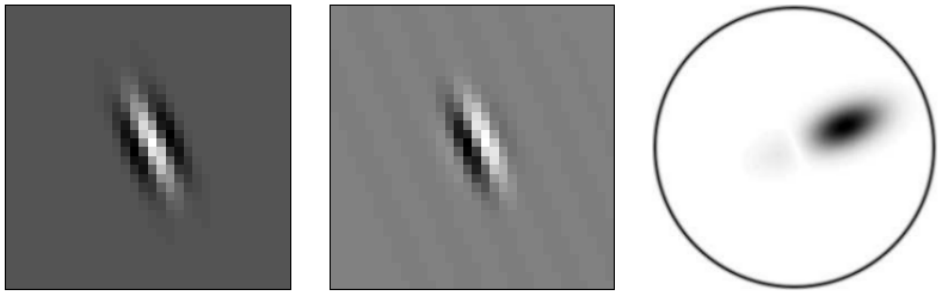

Figure 2.5: Complex morlet wavelet. a) Real part of ψ. b) Imaginary part of ψ. c) Fourier modulus $|\hat{\psi}|$. Image taken from [BM12].

2.5.1 Filter Bank

A wavelet transform filters x using a family of wavelets: $\{x \star \psi_\lambda(u)\}_\lambda$. It is computed with a filter bank of dilated and rotated wavelets having no orthogonality property. The filter bank has 4 parameters: M, N, J, L where M, N stand for the initial spatial size of the input, J is the scaling parameter and L is the number of angles used for the wavelet transform. J determines the size of the downsampling for the filters. The new output is downsampled by 2^{2*J}, i.e. An input image of size $(32, 32)$ is downsampled by $J = 1$ to be of size $(16, 16)$ or by $J = 2$ to be of size $(8, 8)$. It is important to note that the filter bank is just an accumulation of filters and is independent of the data.
A visualization of the filter bank used in this work can be found in figure 2.6. The filters are shown for $J = 0, 1, 2$ and $L = 8$ different angles. The red blurry dot in the bottom is the result of a Gabor filter which is a sinosoidal wave multiplied with a Gaussian function. The Gabor Filter is used as a low-pass filter.

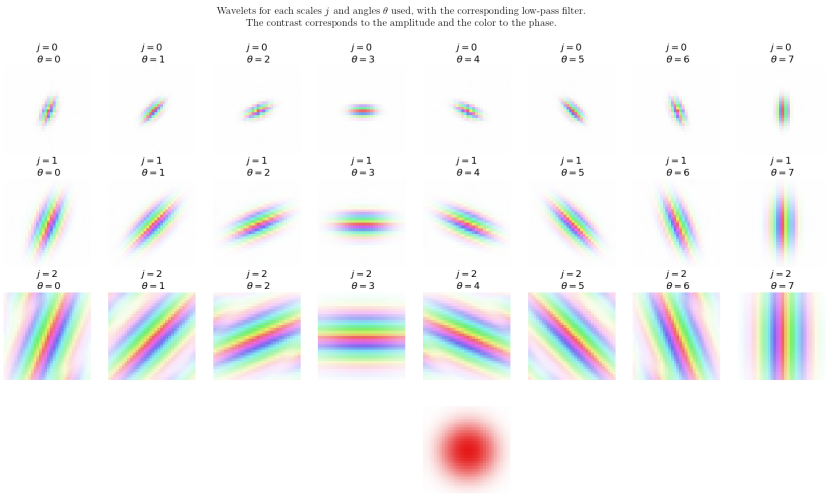

Figure 2.6: Visualization of the filter bank. The $j = 0,1,2$ describe the different downsample sizes. The $\theta = 0,...,7$ describe the different angles. For these images $N,M = 32, L = 8, J = 3$.

2.5.2 Scattering Paths

To compute more and higher order scattering coefficients we iteratively apply the scattering transform. Let $U[\lambda]x = |x \star \psi_\lambda|$. A sequence $p = (\lambda_1, \lambda_2, ...\lambda_m)$ defines a *path*, which is the ordered product:

$$U[p]x = U[\lambda_m]...U[\lambda_2]U[\lambda_1] = |||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|...| \star \psi_{\lambda_m}|.$$

A scattering transform along the path p is defined as an integral, normalized by the response of a Dirac:

$$\bar{S}x(p) = \mu_p^{-1} \int U[p]x(u)du)$$

with $\mu_p \int U[p]\delta(u)du$. From this it follows that each scattering coefficient $\bar{S}x(p)$ is invariant to a translation of x. The scattering is Lipschitz continuous to deformations as opposed to the Fourier transform modulus

2.5.3 Scattering Networks

Scattering Networks are the result of all previously explained concepts. There are m layers in the network, where every m describes the length of the scattering paths in that layer. m is also called order because it describes the number of consecutive scattering applications in that path. A scattering network is a tree that starts with a single root node in layer $m = 0$ and branches out for every further layer with branching factor L. Similar to the filter bank the Scattering network is a collection of filters. However, the scattering network is the first time when data is actually input into those filters. In distinction to conventional CNNs the final output is not only taken from the last layer, but from every single node in that network. An example of a scattering network with $L = 4$ and $m = 3$ is shown in figure 2.7. The nodes describe the filters that are independent of the data, i.e. $U[\lambda_1]f$ for the first layer. The black arrows indicate the outputs at every node, i.e. the scattering coefficients that result from applying this particular scattering path to data for example $S_J[\lambda_1]f$ for a node in the first layer. The root node $U_J[\theta]f = f \star \phi_J$ is the low-pass filter which is in this case a Gabor filter.
In [BM12] it is shown that using more than $m = 2$ produces a lot of unnecessary computation because most of the information from data is already captured in the second-order scattering coefficients. For practical purposes this paper from now on assumes that networks are at maximum $m = 2$ layers deep and in this paper for some applications $m = 1$ only. The total number of filters and therefore also the total number of outputs per datapoint are shown in

$$i \cdot (1 + JL) \tag{2.9}$$

$$i \cdot (1 + JL + \frac{1}{2}J(J-1)L^2) \tag{2.10}$$

2.9 for $m = 1$ and in 2.10 for $m = 2$. i denotes the number of input channels of the input which is 3 for most applications since RGB images are used. In the case of RGB images the scattering transformation is applied for every channel separately. Figure 2.7 is only showing a network for one abstract J. In a real network this J is an actual integer. Therefore it also has to be factored in the equations 2.9 and 2.10. Lastly, it should be noted that the output of the scattering network all have the same downsamples size determined by J even if the filters have different sizes. This is achieved by subsampling the current scattering coefficients in the Fourier domain such that the output is the desired one. To make this more clear an example is

provided: A scattering network with $N, M = 32; J = 2; L = 8$ is initialized. The network is applied on an RGB image. Therefore there are $3 * (1 + 2 * 8) = 51$ outputs of size $(8, 8)$ because of the downsampling factor $J = 2$.

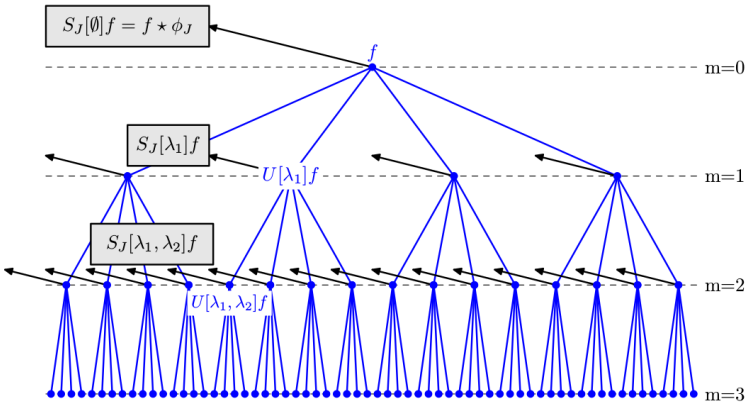

Figure 2.7: Representation of a scattering network. Each black arrow is an output of scattering coefficients of the specified path. The $m = 0, 1, 2, 3$ indicates the layer in the network, i.e. the number of iterations of the scattering transform.

2.5.4 Properties of the Scattering Transform

In this subsection the properties of the scattering transform as used in this work are layed out and the reasons for the properties are pointed out but not proven. For a more detailed explanation and references to proofs the original paper [BM12] must be conducted.

A wavelet is a localized waveform and therefore stable to deformations. This is an upgrade to the sinusoidal waves of the Fourier transform which do not have this property. A wavelet transform computes convolutions with wavelets. It is thus translation covariant not invariant. To achieve invariance a non-linearity must be added. One can then show that the only non-linearity that fulfills the $\mathbf{L}^2(\mathbb{R}^2)$ stability, is differentiable and commutes with translations is the $\mathbf{L}^1(\mathbb{R}^2)$ norm. Therefore it is chosen as the non-linearity.

As already stated in its respective subsection 2.5.2 a scattering transform along any scattering path is translation invariant. Compared to the Fourier transform modulus, which is also invariant to deformations, the scattering transform is Lipschitz continuous to deformations, i.e. the change in the scattering coefficients is bounded and determined by the change in the deformed object.

There are no theoretical bounds on the behavior of the scattering transform when confronted with scaled or rotated objects. However, there are some predictions one can make that will be layed out in the following. Given that scattering networks use rotated and dilated filters with L different angles with equidistant spacing it seems plausible that rotated objects should be captured in some of the filters. Scaled objects should also be captured by the scattering network, because scaling can be viewed as a specific subset of deformation. The Lipschitz continuity w.r.t deformation should also benefit the detection of scaled objects.

Discussion of the properties

In this work the scattering transform should be applied to object detection. In previous works the scattering transform has mainly been applied to image classification. In the following a short distinction between the two tasks is presented. For image classification invariance w.r.t. rotation, translation, scale and deformation are all positive attributes because every image corresponds to exactly one category.

2.5.5 Hybrid Networks

TODO

3 Experiments

3.1 Setup

4 Results

TODO

5 Conclusion

TODO

Bibliography

[ACC+17] Tameem Adel, Taco Cohen, Matthan Caan, Max Welling, On behalf of the AGEhIV study group Initiative, and the Alzheimer's Disease Neuroimaging. 3d scattering transforms for disease classification in neuroimaging. *NeuroImage: Clinical*, 14:506–517, 2017. Exported from https://app.dimensions.ai on 2018/10/21.

[BM12] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *CoRR*, abs/1203.1513, 2012.

[LBD+89] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[OBZ17] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. *CoRR*, abs/1703.08961, 2017.

[OM14] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. *CoRR*, abs/1412.8659, 2014.

[SM13] Laurent Sifre and Stephane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 1233–1240, Washington, DC, USA, 2013. IEEE Computer Society.