# Wavelet-based Prototype Learning for Image Classification

Hanna-Georgina Lieb, Tamás Kaszta, Lehel Csató

Babeş-Bolyai University of Cluj-Napoca - Computer Science Department
1 Kogalniceanu street, RO-400084, jud. Cluj, Romania

**Abstract**.
The need for transparent decisions is becoming a burning question in safety-critical domains. We propose a novel interpretable architecture – named WaveProtoPNet – that is a prototype-based architecture, where feature extraction is done via wavelet decomposition. We show the interpretability and the reduced parameter size of the model, but also focus on its performance on the NCT-CRC-HE-100K data set on human tissue types. By using wavelets, the parameters are reduced substantially, leading to significant saving in both memory and computation. As with all prototype-based methods, interpretability is a consequence of prototype locations, providing an "explanatory" insight into the classification.

## 1 Introduction

The adoption of convolution-based intelligent systems is embraced in safety-critical domains, despite the deficiencies in understanding the "inner mechanisms" and their limits. Therefore, an urgent need is presented to get insight into the decision-making process to prevent dysfunctions that could lead to serious shortcomings.

To investigate the inner states of a model and to obtain possible "explanation" for its working, prototypes were introduced, similar to vector quantisation [5]. Prototypes are structural parts of a decision-making system that can be visualised in the input space. Various approaches have been discussed in PrototypeDL [6], where the authors built a model based on prototypes of input image sizes. The idea was further refined into ProtoPNet [1], which uses prototypes smaller than the input size. In such a way, important details about different classes may be distilled.

We introduce wavelets [7] into the prototype-based model family, to make the prototypes interpetable and to reduce computation. Wavelets are signal processing tools with excellent results in image processing [8], and are also used together with convolutional networks [2]. We will present a network specialised for image processing based on prototypes and wavelets, WaveProtoPNet (wavelet-based prototype network), see Figure 2.a. This system is a fusion of ideas from ProtoPNet and wavelets. Due to the simplicity of wavelets, it can address both the extremely large parameter size of CNNs and the requirement towards interpretability.

We structure the rest of the paper as follows: first we go through the fundamentals of wavelets (section 2), then our model, the WaveProtoPNet, is in-

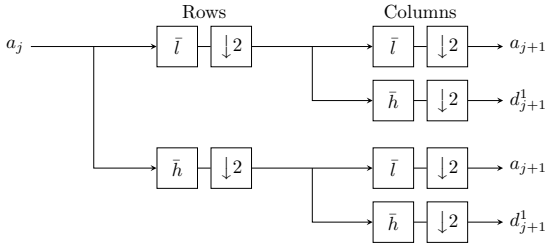



Fig. 1: Fast two-dimensional wavelet decomposition.

troduced (section 3), and after that the experiments are discussed (section 4), ending with the conclusions and future work (section 5).

## 2 Wavelets

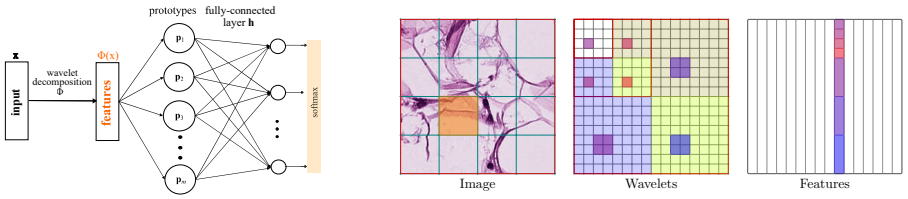
The wavelet transform is developed using the idea behind the Fourier transform, with the difference that they keep both the time and the frequency components of the functions [7]. Wavelets have unit norm and zero mean, and from a mother wavelet, by scaling and translating it, we can gain a wavelet family. When we work with wavelets as convolutions, translation is redundant. A wavelet can be built from the low pass filter $l$ and the high pass filter $h$ corresponding to it. The fast wavelet transform is implemented with the use of these filters.

Images are 2D signals; therefore, to analyse them, we use the two-dimensional discrete wavelet transform. The procedure is as follows (Figure 1): the filters are applied to each row of the image. After this comes a down-sampling. The results are convoluted column-wise, followed by another down-sampling, resulting in four quarter-sized images: average (application of the low-pass filter twice) and three detail components. This could be repeated until the desired decomposition level is achieved. The reconstruction of an image is done in the same fashion with execution of the above operations in reverse order: up-sampling, then convolution.

## 3 Wavelet-based Prototype Network - WaveProtoPNet

The first part of the network in Figure 2.a is a wavelet decomposition ($\phi : x \rightarrow \phi(x) \in \mathbb{R}^D$), which has the duty of extracting features from its input (Figure 2.b). After this comes the layer of prototypes ($g_p$). Finally, the data flow through a fully connected layer,$h$ ($w_h$ weights), is responsible for producing classification probabilities. A prototype learns information from a region of an image, called a patch. There are a set of prototypes for each class that capture the complexity of it in detail.

Given a feature map $\phi(x) = z$, the $j^{th}$ prototype computes the distance between each patch and the prototype itself; a global max-pooling is applied before passing the value onto the fully connected layer. For any $f(x) = z$ the




(a) WavePPNet-architecture (b) Patch-extraction mechanism

Fig. 2: (a) WaveProtoPNet structure. (b) *(left)* Original image with the observed patch. *(mid)* Highlighted: the information belonging to the observed patch on the decomposed image. *(right)* Information collected in a feature vector.

output of the $j^{th}$ prototype is

$$g_{p_j}(z) = \max_{\tilde{z} \in patches(z)} \log\left(\left(||\tilde{z} - p_j||_2^2 + 1\right) / \left(||\tilde{z} - p_j||_2^2 + \epsilon\right)\right).$$

The function above converts proximity into strength: the higher the value, the closer the patch is to the prototype.

WaveProtoPNet training has two steps: first, a prototype training is performed (with the fully-connected layer fixed at their initial values), after which comes the training of the whole model. For $D = \{(x_i, y_i)\} = [X, Y]$ a set of labelled images, a first set of loss function comprises the *cross-entropy loss* ($\mathcal{L}_{Clst}$), *cluster cost* ($\mathcal{L}_{Clst}$) - responsible for the closeness of the feature vectors to at least one prototype of its own class, and *separation cost* ($\mathcal{L}_{Sep}$) - separation of prototypes of different classes:

$$\mathcal{L}_{Clst} = \frac{1}{n}\sum_{i=1}^{n} \min_{k:p_j \in P_{y_i}} \min_{z \in patches(f(x_i))} ||z - p_k||_2^2,$$

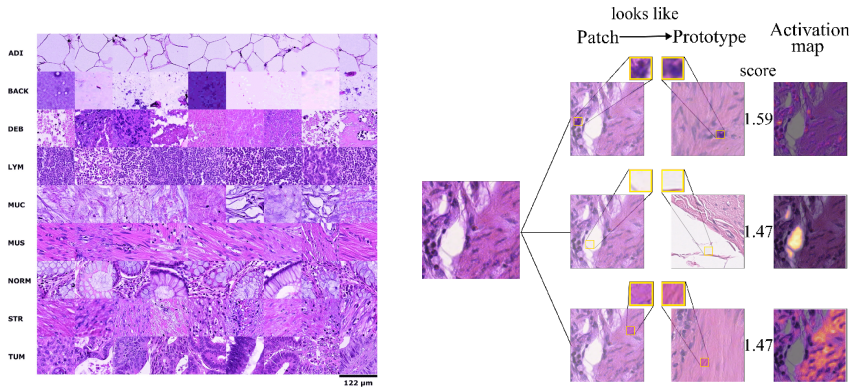
$$\mathcal{L}_{Sep} = -\frac{1}{n}\sum_{i=1}^{n} \min_{k:p_j \notin P_{y_i}} \min_{z \in patches(f(x_i))} ||z - p_k||_2^2,$$

$$\mathcal{L}_1 = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{Clst} + \lambda_2 \mathcal{L}_{Sep}.$$

The second loss focuses on avoiding negative reasoning, by keeping the fully-connected layers weights close to zero, at edges connecting protypes with a class their not belonging to:

$$\mathcal{L}_2 = \mathcal{L}_{CE} + \lambda \sum_{k=1}^{K} \sum_{j:p_j \notin P_k} |w_h^{(k,j)}|.$$

The discrete wavelet transform ($\phi$) can be thought of as a convolution, with the use of four filters, corresponding to the average (low-low) and to the three detail components (low-high, high-low and high-high). The gpu-compatible wavelet architecture has two main structural parts: filtering and rearrangement




(a) data samples (b) explanation

Fig. 3: (a) samples from the NCT-CRC-HE-100K dataset [4]; (b) activation maps of three prototypes projected onto a source image from class "smooth muscle" (MUS), with corresponding activation scores.

of the decomposed part to achieve a form of the result needed at the last level. We defined two possible ways of extracting patches from an image: one in which the most representative parts of the patches are independent, and another where the patches overlap with half of the neighbouring patch.

The level of decomposition and overlap are modifiable parameters of the model, which greatly influence the output of the system. The *decomposition level* is determining how many times the decomposition operation is repeated, and it directly influences the size of prototypes and the number of patches in which an image is split. Taking into account the *overlap* flag (binary: 1 if overlap is applied, 0 otherwise), the number of channels $C$ and the decomposition level $k$, the size of a patch is the following: $patchSize_k = (2^k)^2 * C * (4^{overlap})$.

The *receptive field* of a patch is the region of the original image from which the patch contains information. The receptive field, based on decomposition level $k$, with filter length $l$, and overlap flag *overlap*, has a shape of: $(2^k(l - \neg overlap) - l + 2) \times (2^k(l - \neg overlap) - l + 2)$.

## 4 Experiments and results

The language of implementation of the ML network was `Julia`. We investigated the performance of WaveProtoPNet in a medical database with various tissue samples. We cover the different parameter settings, prototypes, and wavelets used. The NCT-CRC-HE-100K data set [3] contains 100 000 images of human colorectal cancer (CRC) and normal tissue. RGB images of size $224 \times 224$ come from nine classes (Figure 3.a). In a study [4], an accuracy of 98.7% was achieved in the internal test set.

With the first type of loss, the system runs for 4 epochs, with a 0.01 learning rate, then for other 4 epochs, with a 0.005 learning rate. Then comes the thawing

Table 1: Classification accuaracy in case of different decomposition level and overlap, using 45 protype per class and Daubechies2 wavelet.

| ProtoSize | Level | Overlap | Train | Test |
|---|---|---|---|---|
| 12 | 1 | 0 | 89 | 88 |
| 48 | 1 | 1 | 93 | 92 |
| 48 | 2 | 0 | 91 | 89 |
| 192 | 2 | 1 | 94 | 92 |
| 192 | 3 | 0 | 91 | 89 |
| 768 | 3 | 1 | 81 | 80 |

of the fully connected layer, and the training of the entire model, with the second type of loss, first with a learning rate of 0.01 for 2, then of 0.005 for 3 epochs, and finally switches to a learning rate of 0.001 for the rest of 6 epochs. During the experiments, Adam optimizer and the following loss parameters were used: $\lambda_1 = 0.9$, $\lambda_2 = 0.4$, $\lambda = 0.001$.

By adopting wavelets, the trainable parameters were reduced considerably. In the paper we built upon (ProtoPNet), numerous types of backbones were used, of which the smallest has 8 million parameters. With wavelet backbone, there is no need for more parameters than 5000 - which is 1600 times less than what the smallest backbone requires - to achieve the above 90% accuracy.

We observed that as the number of prototypes increases, the accuracy of the model also increases to a certain point. With 45 prototypes per class, the precision was above 90%, with Daubechies2 wavelet, decomposition level of 2, and no overlap. From then on, the variation in precision was inconsiderable. With a larger prototype number, the accuracy did not decrease, so the model is not overfitting. For the rest of the experiments, we used 45 prototypes per class, as it is the best trade-off between less memory usage and achievable accuracy.

We tested the performance, as a function of the decomposition level and overlap, with the Daubechies2 wavelet and 45 prototypes per class (Table 1). On the first and second levels of decomposition, better results were achieved with overlap, as more patches from the image are converted to prototypes (+4%). In the case of the decomposition level 3, the overlap caused a decrease in the precision of 9%, due to the larger size of the prototype. The best results were achieved on the first and second levels of decomposition, in both cases with overlap applied, the precision being 92% on the test dataset.

We conducted our experiments mostly on the Daubechies wavelet, but also tested the Coiflet and Symlet wavelets. The best results were achieved using the Daubechies2 and Daubechies1 wavelet: 94% train and 92% test accuracy.

Suppose that we have an image that belongs to the mucus class of the tissue samples dataset. If this image is classified as mucus, an activation map is generated for the three closest prototypes of that class. These values are then projected back into the input space onto the original image in the form of a heatmap. There is also an activation value for each prototype calculated by multiplying the similarity score by the appropriate weight in the fully connected

layer (Figure 3.b). The similarity score is obtained as a result of the comparison between the prototype and the patch of the image. The similarity value with the activation map provides an explanation for the actual prediction.

## 5 Conclusions and future work

Wavelet-based feature extraction ideally substitutes for traditional convolution-based architecture for image classification. Memory consumption was significantly reduced compared to convolution-based backbones (min. 1600 times). Combining wavelets with prototypes created a powerful and robust model that produces highly accurate and interpretable predictions. Significant achievements were obtained with the explainable architecture of WaveProtoPNet: the whole system is scalable and runs on GPUs. The precision was on par with the cutting-edge technologies when applied to a medical data set. It also generated an explanation for each prediction that offers information on the decision-making process to assess possible shortcomings or eventual biases that otherwise would go unnoticed. The utility of the prototype set learnt by the network could be further investigated. The usefulness of individual prototypes remains unclear; therefore, a pruning mechanism could be implemented to reduce the number of prototypes while keeping the classification accuracy high. A second direction could be the implementation of segmentation to easily differentiate between background and information-rich areas of images.

## References


[1] C. Chen, O. Li, A. Barnett, J. Su, and C. Rudin. This looks like that: deep learning for interpretable image recognition. In *NIPS 33*. Curran Associates, 2018.

[2] T. Guo, T. Zhang, E. Lim, M. Lopez-Benitez, F. Ma, and L. Yu. A review of wavelet analysis and its applications: Challenges and opportunities. *IEEE Access*, 10:58869–58903, 2022.

[3] J. N. Kather, N. Halama, and A. Marx. 100,000 histological images of human colorectal cancer and healthy tissue, May 2018. URL `https://doi.org/10.5281/zenodo.1214456`.

[4] J. N. Kather, Krisam, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

[5] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, New York, NY, third edition, 1989.

[6] O. Li, Liu, et al. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proc. of AAAI Conf. on Artificial Intelligence*, volume 32, 2018.

[7] S. Mallat. *A Wavelet Tour of Signal Processing*. Elsevier, 2009.

[8] G. Othman and D. Q. Zeebaree. The applications of discrete wavelet transform in image processing: A review. *Journal of Soft Computing and Data Mining*, 1(2):31–43, 2020.