ReWaRD: Retinal Waves for Pre-Training Artificial Neural Networks Mimicking Real Prenatal Development

Benjamin Cappell
benjamin.cappell@fau.de

Andreas Stoll
andi.stoll@fau.de

Williams Chukwudi Umah
chukwudi.umah@fau.de

Bernhard Egger
bernhard.egger@fau.de

Cognitive Computer Vision Lab, Chair of Visual Computing
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Editors: Marco Fumero, Emanuele Rodolà, Clementine Domine, Francesco Locatello, Gintare Karolina Dziugaite, Mathilde Caron

Abstract

Computational models trained on a large amount of natural images are the state-of-the-art to study human vision – usually adult vision. Computational models of infant vision and its further development are gaining more and more attention in the community. In this work we aim at the very beginning of our visual experience – pre- and post-natal retinal waves which suggest to be a pre-training mechanism for the primate visual system at a very early stage of development. We see this approach as an instance of biologically plausible data driven inductive bias through pre-training. We built a computational model that mimics this development mechanism by pre-training different artificial convolutional neural networks with simulated retinal wave images. The resulting features of this biologically plausible pre-training closely match the V1 features of the primate visual system. We show that the performance gain by pre-training with retinal waves is similar to a state-of-the art pre-training pipeline. Our framework contains the retinal wave generator, as well as a training strategy, which can be a first step in a curriculum learning based training diet for various models of development. We release code, data and trained networks to build the basis for future work on visual development and based on a curriculum learning approach including prenatal development to support studies of innate vs. learned properties of the primate visual system. An additional benefit of our pre-trained networks for neuroscience or computer vision applications is the absence of biases inherited from datasets like ImageNet.

Proceedings of the I edition of the Workshop on Unifying Representations in Neural Models (UniReps 2023).

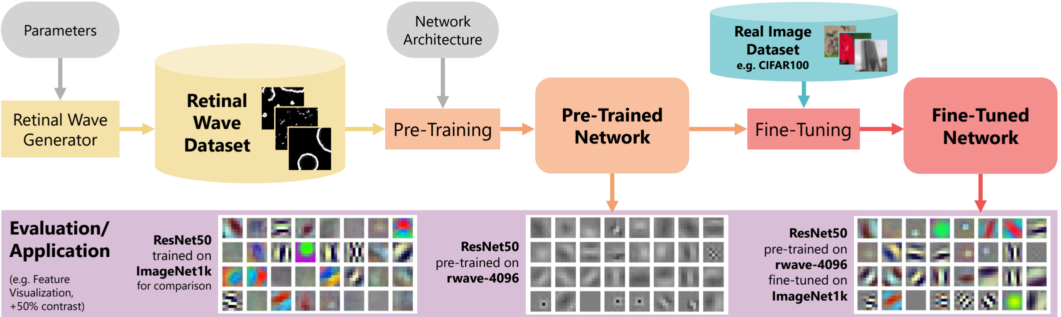

Figure 1: Overview of our framework including retinal wave generation, pre-training and fine-tuning. Feature visualizations show first convolutional layer.

1 Introduction

Retinal waves are a phenomenon of randomly initiated wave-like patterns, which travel over the developing retina, for mammals first reported by Meister et al. in 1991 (18). They are difficult to observe, e.g. using calcium imaging and multielectrode array recordings (7), as they occur prenatally before eye opening and cone/rod maturation. Retinal waves play a role in the development of neural pathways (in and between retina, LGN and V1). One example is the development of long range horizontal connections, affecting the orientation map in the visual system based on the directions of retinal waves (14).

At different stages of the development, retinal waves are initiated in various biological methods. We focus on earlier, cholinergic amacrine cell mediated waves (7). Reported retinal waves differ in shape, size and propagation speed. This variety is what could be a strong basis to interpret retinal waves as input for prenatal development of the visual system.

In pre-natal mice, directional retinal waves have been shown to simulate future optical flow, therefore priming visual motion detection even before the onset of vision. This suggests the development of higher-order visual processing regions prior to eye opening (8).

We assume especially the early layers of visual processing to profit from pre-training through retinal waves, as those are known to capture basic features similar to Gabor features. Later regions like V4 or IT might get an initial signal for wiring and feature learning, but those regions are more object-centric and therefore likely profit less from this first visual experience. At eye opening, the wiring of the visual system is initialized and can be refined by utilizing visual impressions of the real world.

Similar to a biological pre-training of the visual system with retinal waves, pre-training of Artificial Neural Networks (ANNs) (even with non-natural images) is used to generalize faster and to boost accuracy (12; 13).

In our framework, we perform training in a fully supervised setting. This is possible by assigning an individual class label to each wave. This comes with the heuristic, that a retinal wave is processed in a somewhat holistic way and therefore such a supervision signal (e.g. current stage of the development process) could be also available during training of a biological system. Whilst unsupervised training would be an option with our synthetic retinal wave data, we focus on supervised learning, as this has been shown to match brain activity better on various benchmarks.

Related work has shown that Slow Feature Analysis (SFA) units derived from simulated retinal waves share a number of properties with cortical complex-cells (6) and that pre-training a classifier ANN with real and simulated retinal waves using a Hebbian learning rule improves separability of NN-internal representations and classification accuracy (17).

This leads us to the following research questions:

- What features would arise in ANNs if we mimic development of the visual system through retinal waves by pre-training ANNs with simulated retinal wave images?
- How similar will these ANNs be to the human brain internally?
- How well will these networks perform when fine-tuned for image classification tasks?

In summary the core contributions of our work are as follows:

- We adapt an existing retinal wave simulator for the purpose of image generation for pre-training ANNs and create two different retinal wave datasets.
- We pre-train and fine-tune ANNs in various different settings and evaluate them qualitatively and quantitatively in terms of how well they match biological visual systems and how they compare to another pre-training approach with generated images of fractals (12) (13).
- We release all datasets, code and pre-trained networks to foster the development of variants of our experiments. Therefore, custom retinal wave based curriculum learning strategies may be built upon our pre-trained networks.

The paper is structured as follows: we first describe the retinal wave generator and the particular datasets we generated. Second, we provide details about our pre-training and fine-tuning scheme. Finally we present different qualitative and quantitative evaluations of our method by showing the learned features, measuring accuracy on an object recognition task and comparing how similar the activations in our models are to activations measured in biological vision systems.

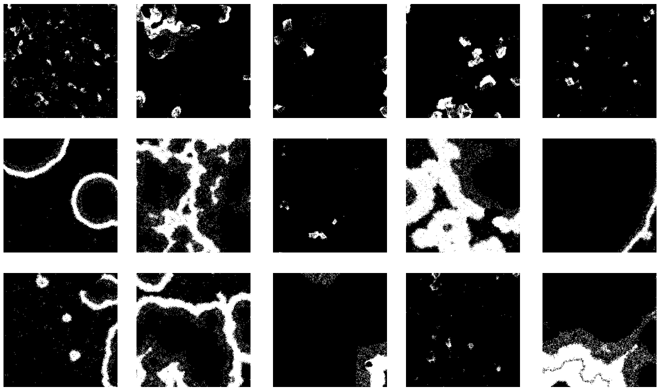

Figure 2: Simulated retinal wave images (random examples from rwave-4096)

2 Retinal Wave Datasets

To obtain a large amount of retinal wave data required to pre-train ANNs, we modified a biologically accurate retinal wave simulator (9). It simulates and renders retinal waves by outputting activations of dendritically interconnected amacrine cells. These cells are organized in a hexagonal grid and fill a circle with specifiable radius.

Sample parameter sets for the amacrine cells are available to closely reproduce observed retinal waves of different species during different stages of development (9). Every time frame of a retinal wave, whose propagation, shape and size is influenced by multiple adjustable parameters of the amacrine cells, can be simulated. The parameters we control affect the wave shape, speed, duration and size. For further details about these parameters we refer the interested reader to (9).

Our modifications to the retinal wave simulator include: Generating datasets by choosing base parameters of amacrine cells and specifying parameter spread, which leads to multiple parameter combinations; Retina-to-image projection of retinal waves; Modifying the simulator to support larger retinas; Storing generated retinal waves as .png image files. In Figure 2, exemplary images from generated waves are visualized.

Different parameter combinations for retinal waves act as different classes to train a network in a fully supervised setting. As the parameters are continuous, an unlimited number of classes can be created. A specific parameter constellation corresponds to one class for the supervised learning task. By having multiple frames of a retinal wave, each class contains multiple image frames from the same wave acquired at different time steps. Instances of three different example classes are shown in Figure 3.

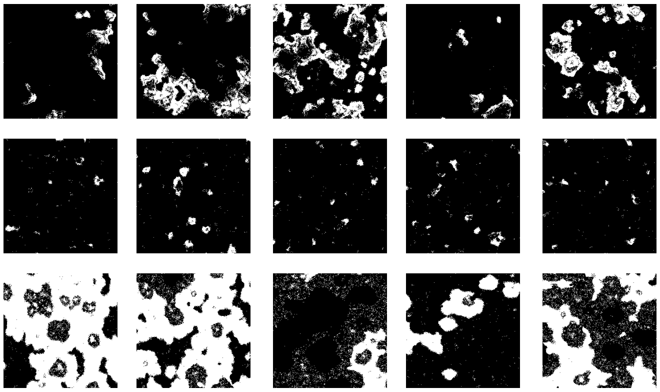

Figure 3: Simulated images: random examples of three classes (rows). We generate a set of retinal waves based on different parameters and each wave is given a separate class label. Different time frames of the same retinal wave are assigned the same class label. This enables us to train ANNs in a fully supervised fashion.

With this approach, an arbitrary number of different image instances can be generated for each class. This enables us to generate training data for a fully supervised learning task, comparable to the one proposed by FractalDB (12; 13) and comparable to the standard ImageNet based training which leads to high similarity to biological processing.

Not every temporal frame of a retinal wave is used for the image dataset: we skip retinal wave images during dataset creation if they are temporally too close to one another (spacing, only every n-th image of the wave is considered for a dataset) or if they don't contain a specific amount of pixels (threshold). This choice helps to only include images, where retinal waves are clearly visible, keep the size of the training dataset reasonable and have more variance in the dataset.

We used random mirroring and random continuous 360° rotation as data augmentation strategies. If for a particular class the specified amount of images could not be generated due to too high threshold or temporal spacing, threshold and spacing are dynamically adjusted for the affected class, until the desired amount of images is generated for the class.

The adapted retinal wave generator can output two types of image data. Firstly, cropped, square images. The corners of the cropped images line up with the retina circle outline such that all pixels of the image represent amacrine cells of the simulated retina. The image size is specifiable, independent of the amount of simulated amacrine cells (default: 256x256 pixels). The resulting images are binary .png files, 1 = cells active, 0 = cells inactive. An example image is shown in Figure 4 (right).

Secondly, images containing raw, unresampled retinal data can be outputted as 3x8 bit RGB .png images (red channel: simulated calcium imaging response of the retina (8 bit), green channel: raw amacrine cell activity (2 bit), blue channel: retina boundary (1 bit)). An example is shown in Figure 4 (left). These raw images could be a basis to adapt and reuse the retinal wave dataset for recurrent time series learning, as there are no temporal jumps (i.e. every generated frame is stored) and the data is not randomly augmented.

Additionally, for each class a textfile is generated, containing the exact parameter values used for generating retinal waves of that class. Thus, the dataset could also be used for regression tasks.

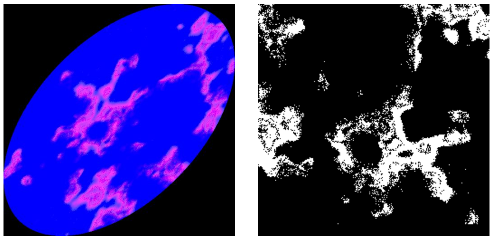

Figure 4: Raw retinal wave image (left); Cropped, shifted and augmented retinal wave image (right).

Table 1: Created retinal wave datasets. For each of the parameters (affecting wave shape, speed, duration, spacing and size), four different values were used. The class count results by generating classes for all of the possible parameter value combinations (e.g. $4^5 = 1024$).

dataset	classes	images/class	altered params	dimensions
rwave-1024	1024	1000	5	256x256 pixels
rwave-4096	4096	2000	6	256x256 pixels

We created two retinal wave datasets that are made publicly available (3; 4). Further details are provided in Table 1. Random images of the rwave-4096 dataset are shown in Figure 2.

The parameter ranges were set according to example parameter combinations provided by (9) for retinal waves occurring in ferrets, rabbits, mice, chickens and turtles.

Both datasets are split into training set (80%), validation set (10%) and test set (10%). The split is done balanced - for one set, all classes contain the same number of instances.

3 Pre-Training ANNs

By mimicking mammalian development of the visual cortex during ANN training, we would expect similar features to arise as used by the human visual system. As the first visual cues in this development are retinal waves, we try to mimic the mammalian visual development in ANNs by providing the first visual input to ANNs as retinal waves. We achieve this in the pre-training phase, by training a classifier network on one of the retinal wave datasets.

We base our pre-training on an existing approach of pre-training with synthetically generated fractals and thus, reuse their codebase (12; 13). This is possible, as their training approach is very similar to ours: firstly, train an image classifier neural network on a pre-training dataset (in the case of FractalDB: fractal images, in our case: retinal wave images), later: fine-tune the pre-trained network on the final task-specific image dataset.

In order to start the artificial development process, an ANN architecture is selected and used to create a new model, typically a convolutional neural network. We focused experiments on the ResNet50 Architecture (10), however, we also release pre-trained ResNet34 (10) and AlexNet (16) networks since these are heavily used in the neuroscience and computational cognitive science community. The training pipeline (reused from (12; 13)) additionally supports VGG16, VGG19 (23); ResNet18, ResNet101, ResNet152, ResNet200 (10); ResNeXt101 (25) and Densenet161 (11).

The training set of the selected retinal wave dataset is used to optimize the model's trainable parameters (task: classify one retinal wave image). The validation set can be used to monitor performance and to tune hyperparameters (e.g. learning rate). After 90 epochs, the pre-trained model is saved for fine-tuning.

4 Fine-Tuning of Pre-Trained Networks

As soon as one opens their eyes for the first time, different visual cues become apparent to the visual system: The real world! We mimic this by fine-tuning our pre-trained ANN. No weights are frozen/fixed – as in the real world, the visual system (resp. the ANN) heavily adapts after eye opening. From our perspective this is an instance of a data-driven inductive bias through pre-training. The last layer (classification) of the ANN is swapped out for a layer with the output size equal to that required for the fine-tuning dataset, to make classification possible. This pre-trained model is then used in an instance of transfer learning, as the model with all its weights is re-used for fine-tuning (apart from the last layer). Task-specific and labeled data is now used for object classification like ImageNet (20) or CIFAR (15), which are also the image datasets used to train current state-of-the-art models for visual perception. We again reuse the FractalDB codebase for this fine-tuning step. (12; 13)

5 Evaluation

We present different qualitative and quantitative evaluations of our method in the following. All evaluations were carried out using the trained ResNet50 networks if not mentioned otherwise.

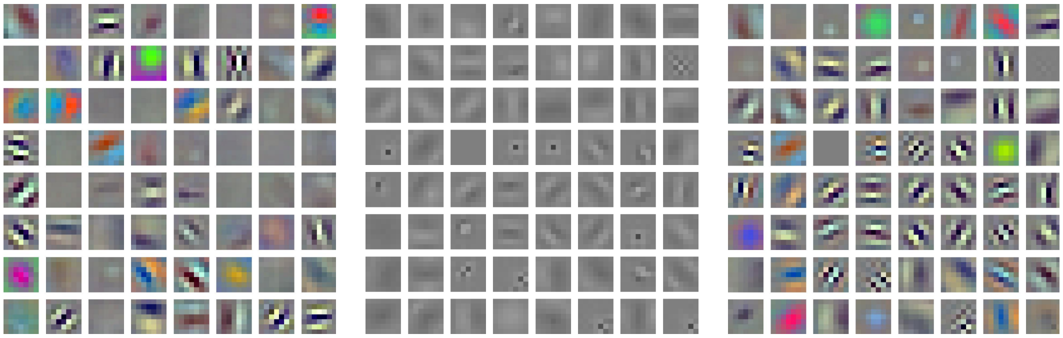

Figure 5: Learned first layer filters of ResNet50 trained from scratch ImageNet1k (left), pre-trained on rwave-4096 (middle) and pre-trained on rwave-4096 + fine-tuned on ImageNet1k (right). For training, the default configuration of the FractalDB pipeline was used.

Filters: The learned filters of early layers of pre-trained ResNet50 ANNs closely match gabor-filters, making them similar to the real V1 filters in the human visual system. Similar filters are observed on trained AlexNet networks. The filters are visualized by showing the weight values. An example is provided in Figure 5. All visualizations show filters of the first layer of a ResNet50 obtained by training with the default parameters of the FractalDB training pipeline.

The filters of both the pre-trained and fine-tuned networks are similar, as seen in Figure 5. Most of the gabor-like features are preserved, and increase in contrast or additionally implement color cues. In comparison to filters of a network only trained on ImageNet1k, there are more gabor-like features on the first layer of a network that was pre-trained with retinal waves and then fine-tuned on ImageNet1k.

Accuracy: For fine-tuning on CIFAR100, We observe higher accuracies after pre-training on retinal waves and consequently fine-tuning (see Table 2), compared with training from scratch. We observe accuracies similar to pre-training with FractalDB1k. Also, we observe that pre-training on ImageNet1k yields the highest accuracies (but at the trade-off of a bias being introduced, as classes and images in ImageNet are hand-picked) These observations closely matches the results of prior work which was pre-trained on FractalDB (12; 13), however, our training diet is biologically plausible.

Table 2: Accuracies and losses for tested ResNet50 training strategies. For ImageNet1k, the validation set has been used for testing. **Best** for the last dataset in the training strategy is marked in bold. For these training strategies, the default configuration of the FractalDB training pipeline was used (initial learning rate of 0.01).

training datasets	Test accuracy ↑	Test Loss ↓
CIFAR100	60.29%	1.727
rwave-1024 → CIFAR100	69.96%	1.419
rwave-4096 → CIFAR100	72.30%	1.330
FractalDB1k → CIFAR100	71.22%	1.286
ImageNet1k → CIFAR100	**79.46%**	**0.929**
ImageNet1k	68.23%	1.388
rwave-4096 → ImageNet1k	**70.97%**	**1.295**

For fine-tuning on ImageNet1k, we observe the same for the default configuration of the FractalDB training pipeline. Yet, with an initial learning rate of 0.1 during the final training step, training from scratch results in the highest accuracy (see Table 3. This observation matches the results of FractalDB (12; 13).

Table 3: Accuracies and losses for tested ResNet50 training strategies with an initial learning rate of 0.1 during fine-tuning. For testing, the validation dataset has been used. **Best** for the last dataset in the training strategy is marked in bold.

training datasets	Test accuracy ↑	Test Loss ↓
ImageNet1k	**72.42%**	1.218
rwave-4096 → ImageNet1k	71.77%	**1.201**
rwave-1024 → ImageNet1k	71.67%	1.247
FractalDB1k → ImageNet1k	71.68%	1.244

Generalization Time for CIFAR100: In comparison to training from scratch, we observe faster generalization when pre-training on retinal waves and fine-tuning on CIFAR100. The generalization process is very similar to FractalDB1k pre-training. ImageNet1k pre-training enables the fastest generalization. A comparison of training and validation accuracy for the first 45 epochs of training is shown in Figure 6. This again matches the previous observations reported for FractalDB and gives a strong argument why such a pre-training might be a sensible choice – using such pre-training later processing steps can be built more efficiently and the training data seen later does not have to first train the early processing steps. This could partially explain the high degree of data-efficiency we observe in the training of the human visual system.

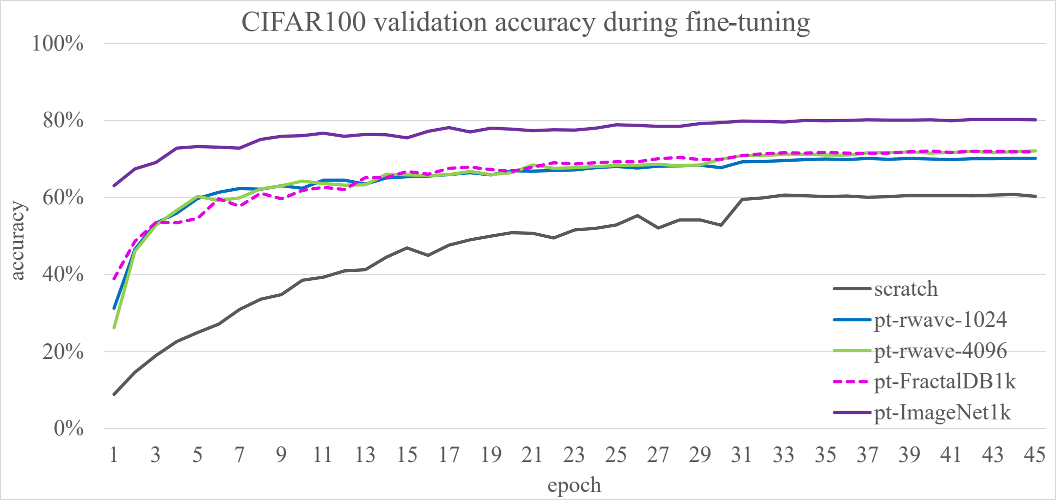

Figure 6: Comparison of ResNet50 training on CIFAR100. scratch: no pre-training was done. pt-: pre-trained on the mentioned dataset. Default parameters of the FractalDB pipeline were used.

Brain-Score combines many vision benchmarks for artificial neural networks and computes scores to measure how well the ANNs match physiological measurements of each region of the ventral stream of the primate brain (21; 22). To evaluate how well our models match regions in the ventral stream of the primate brain (V1, V2, V4 and IT), we compare the ceiled Brain-Score results of multiple models in Table 4. The Behavior score is obtained by evaluating the model on various behavioral benchmarks (21; 22).

Models pre-trained on retinal waves and fine-tuned on CIFAR100 show increased scores in all regions compared to training from scratch. This supports strongly that pre-training ANNs with retinal waves is beneficial and could lead to better models of the primate visual system. Especially the improved representation in early layers makes sense in our eyes, as retinal waves might enable to learn those features in a less noisy and cluttered way.

Table 4: Brain-Score results for tested ResNet50 training strategies. Scores range from 0 to 1, higher is better. Currently best performing model on BrainScore (19) achieves an average score of .465. **Highest score** in each category is marked in bold. (For these training strategies, the default configuration of the FractalDB training pipeline was used (initial learning rate of 0.01). Only for training strategies containing the ImageNet1k dataset, the initial learning rate was set to 0.1 during fine-tuning.)

training datasets	Avg↑	V1↑	V2↑	V4↑	IT↑	Behavior↑
rwave-1024	.230	.437	.217	.252	.185	.059
rwave-4096	.239	.434	**.226**	**.279**	**.190**	.065
FractalDB1k	**.267**	**.572**	.222	.250	**.190**	**.101**
CIFAR100	.216	.389	.154	.305	.225	.010
rwave-1024 → CIFAR100	**.290**	.455	**.267**	.392	**.276**	.060
rwave-4096 → CIFAR100	.282	.444	.236	**.399**	.266	.067
FractalDB1k → CIFAR100	.285	**.506**	.222	.372	.251	**.073**
ImageNet1k → CIFAR100	**.313**	.458	**.277**	**.437**	**.296**	**.097**
ImageNet1k	**.414**	**.538**	.315	**.497**	.378	.342
rwave-1024 → ImageNet1k	.399	.524	.314	.491	**.390**	.277
rwave-4096 → ImageNet1k	.411	.517	.308	.488	.383	**.361**
FractalDB1k → ImageNet1k	.407	.525	**.324**	.489	**.390**	.306

Interestingly, models fine-tuned on ImageNet1k do not show large differences in the BrainScores, which could be explained by the higher initial learning rate (0.1) which was used during fine-tuning.

The Brain-Scores have been obtained by utilizing the base model of Brain-Score; Layers of the CNN are matched automatically to regions of the primate ventral stream when using the base model. Between different models, the different layers were matched inconsistently, which could explain some score discrepancies.

6 Limitations

The parameters for the retinal wave simulator used for our experiment are based on animal experiments and primate or human retinal waves might behave slightly different. We however do not have as much observations for human retinal waves. Using observations of human retinal waves could yield even more insights in how the human brain behaves regarding vision.

We only investigated pre-training in a fully supervised fashion. One could argue, that unsupervised learning would be more biologically plausible. By releasing our data generator and training data we also enable other researchers to try different routes and training schemes.

As the visual cortex has not yet fully developed when retinal waves do occur, a fixed ANN architecture is a limiting factor. An interesting extension of our approach could go in the direction of neural architecture search.

7 Conclusion

We have trained ANNs based on retinal waves and observe not only the emergence of strong and biologically plausible features in early layers, but also a match to primate brain activity of our pre-trained networks and an induced performance gain similar to that of state-of-the-art pre-training dataset FractalDB (12; 13). ReWaRD enables future research to study human visual development with a curriculum learning approach. The approach could be extended, e.g. by expanding the training strategy with different pre-training steps for each stage of retinal waves or even more steps that mimic different stages of early visual experience, e.g. using the SAYCam dataset (24). An additional benefit we would like to emphasize is the absence of bias towards faces, animals or similar object categories in our datasets which could be very beneficial for applications in both neuroscience and computer vision, as it might reduce misleading results arising from a biased training diet and might reduce bias in downstream applications after fine-tuning.

Furthermore, the pre-trained networks (5) can be fine-tuned for many different applications, as they are made publicly available. Also, the generated retinal wave datasets (3; 4) are made publicly available.

Our code base for generating retinal waves, training the models and for analyzing (BrainScore) and visualizing trained networks along with further example feature visualizations is made publicly available as well. (1; 2)

ReWaRD Project Page: `https://github.com/bennyca/reward`

Retinal Wave Simulator Code (1): `https://zenodo.org/records/8150777`
ReWaRD Code (2): `https://zenodo.org/records/10148723`
rwave-1024 dataset (3): `https://zenodo.org/records/7811860`
rwave-4096 dataset (4): `https://zenodo.org/records/7779499`
ANNs pre-trained and fine-tuned (5): `https://zenodo.org/records/10148752`

Acknowledgments and Disclosure of Funding

This project was funded by the Federal Ministry of Education and Research of Germany (BMBF) project 01IS22082 (IRRW) and the FAU Emerging Talent Initiative (ETI). The authors gratefully acknowledge support by the Google Cloud Research Credits program award GCP19980904. The authors are responsible for the content of this publication.

References

[1] Benjamin Cappell. Bennyca/retinal-wave-simulator: Initial release, July 2023.

[2] Benjamin Cappell. Bennyca/reward: Updated brainscore results, Nov. 2023.

[3] Benjamin Cappell. rwave-1024 - retinal wave dataset, Apr. 2023.

[4] Benjamin Cappell. rwave-4096 - retinal wave dataset, Apr. 2023.

[5] Benjamin Cappell and Andreas Stoll. Anns pre-trained on retinal waves, Nov. 2023.

[6] Sven Dähne, Niko Wilbert, and Laurenz Wiskott. Slow feature analysis on retinal waves leads to v1 complex cells. *PLoS Computational Biology*, 10(5):e1003564, May 2014.

[7] Sally I. Firth, Chih-Tien Wang, and Marla B. Feller. Retinal waves: mechanisms and function in visual system development. *Cell Calcium*, 37(5):425–432, May 2005.

[8] Xinxin Ge, Kathy Zhang, Alexandra Gribizis, Ali S Hamodi, Aude Martinez Sabino, and Michael C Crair. Retinal waves prime visual motion detection by simulating future optic flow. *Science*, 373(6553):eabd0830, 2021.

[9] Keith B Godfrey and Nicholas V Swindale. Retinal wave behavior through activity-dependent refractory periods. *PLoS Computational Biology*, 3(11):e245, Nov. 2007.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[12] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. In *Asian Conference on Computer Vision (ACCV)*, 2020.

[13] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. In *International Journal on Computer Vision (IJCV)*, 2022.

[14] Jinwoo Kim, Min Song, Jaeson Jang, and Se-Bum Paik. Spontaneous retinal waves can generate long-range horizontal connectivity in visual cortex. *Journal of Neuroscience*, 40(34):6584–6599, 2020.

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[17] Andrew Ligeralde, Miah N Pitcher, Marla Feller, and SueYeon Chung. Geometry reveals an instructive role of retinal waves as biologically plausible pre-training signals. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022.

[18] Markus Meister, Rachel O. L. Wong, Denis A. Baylor, and Carla J. Shatz. Synchronous bursts of action potentials in ganglion cells of the developing mammalian retina. *Science*, 252(5008):939–943, May 1991.

[19] Alexander Riedel. Bag of tricks for training brain-like deep neural networks. In *Brain-Score Workshop*, 2022.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[21] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint*, 2018.

[22] Martin Schrimpf, Jonas Kubilius, Michael J Lee, N Apurva Ratan Murty, Robert Ajemian, and James J DiCarlo. Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*, 2020.

[23] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. pages 1–14. Computational and Biological Learning Society, 2015.

[24] Jessica Sullivan, Michelle Mei, Andrew Perfors, Erica Wojcik, and Michael C Frank. Saycam: A large, longitudinal audiovisual dataset recorded from the infant's perspective. *Open mind*, 5:20–29, 2021.

[25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.