Yin Yang Convolutional Nets: Image Manifold Extraction by the Analysis of Opposites

Augusto Seben da Rosa*, Frederico Santos de Oliveira†, Anderson da Silva Soares‡ and Arnaldo Candido Junior§
*Federal University of Technology - Paraná; Medianeira, Brazil
†Federal University of Mato Grosso; Cuiabá, Brazil
‡Federal University of Goiás; Goiânia, Brazil
§São Paulo State University; São José do Rio Preto, Brazil
Email: nosaveddataoz1@gmail.com, fred.santos.oliveira@gmail.com, andersonsoares@ufg.br, arnaldo.candido@unesp.br

***Abstract*—Computer Vision in general presented several advances such as training optimizations, new architectures (pure attention, efficient block, vision language models, generative models, among others). This have improved performance in several tasks such as classification, and others. However, the majority of these models focus on modifications that are taking distance from realistic neuroscientific approaches related to the brain. In this work, we adopt a more bio-inspired approach and present the Yin Yang Convolutional Network, an architecture that extracts visual manifold, its blocks are intended to separate analysis of colors and forms at its initial layers, simulating occipital lobe's operations. Our results shows that our architecture provides State-of-the-Art efficiency among low parameter architectures in the CIFAR-10 dataset. Our first model reached 93.32% test accuracy, 0.8% more than the older SOTA in this category, while having 150k less parameters (726k in total). Our second model uses 52k parameters, losing only 3.86% test accuracy. We also performed an analysis on ImageNet, where we reached 66.49% validation accuracy with 1.6M parameters. We make the code publicly available at: https://github.com/NoSavedDATA/YinYang_CNN.**

***Keywords*—Convolutional Neural Networks; Bio-inspired Neural Networks.**

I. Introduction

The field of Neural Computer Vision has presented a great advancement, for instance, Convolutional Neural Networks (CNN) were proposed as a way to reduce the computational complexity of images in relation to densely connected neural networks [1]. AlexNet [2] revolutionized Computer Vision and Artificial Neural Networks (ANN) with an efficient GPU implementation of convolution operations. Further, improved neural network architectures were proposed.

Some of these architectures include: mobile model families as MobileNet [3], EfficientNet V2 [4] and RegNet [5]; two-branch neural networks for semantic segmentation, as BiSeNet [6], Deep Dual-Resolution networks [7] and SeaFormer [8]; pure attention mechanisms applied into image classification, such as ViT [9] and MaxViT [10]; image generative models like Stable Diffusion [11] and DALL-E-2 [12]; and lastly, vision-language models, as CLIP [13]. The majority of these architectures focuses on increasing model efficiency by improving the micro-architecture – there is, by making adjustments relative to the inside of a network block, as in mobile model families. Some of these architecture also leverage the potential of CNNs and Transformers into high level computer vision tasks, as image generation or vision-language models creation.

Although few of the famous modern architectures of Computer Vision aim to reach a more realistic neuroscientific approach to the brain, the bio-inspired approach of Spiking Neural Networks implementation [14] demonstrated State-of-the-Art (SOTA) restuls at object detection [15].

In this regard, we base our research on two neuroscientific findings about function specialized occipital lobe areas, that is, edge detection that happens on V1 area [16] and color processing at V4 area [17]. Also, this form of specialization is also observed in the human eye, in which rod components are related to white and black processing [18], and cone components to color processing [19]. Besides, we suppose that color information may be irrelevant to the classification of different objects (such as plane and boat) and relevant to the classification of similar objects (such as cat and jaguar).

In this research, we present Yin Yang Convolutional Net (YYNet), a neural network model which makes adjustments relative to the global scale of the model. This is performed in the macro-architecture by aggregating blocks (or single purpose networks) that extracts visual manifolds by doing a separate analysis of colors and forms from its input. We find that this architecture provides State-of-the-Art (SOTA) efficiency among low parameter architectures applied to the low data and image resolution dataset, CIFAR-10, using less parameters and less training epochs than existing models.

II. Related Work

Vision neural network research has grown in quality at a very fast pace. In this section, we present related architectures and

their given tasks. Related work can be categorized into five approaches: mobile networks; two branch networks; transformers; vision-language models; generative models.

Some of this approaches are usually divided in three parts: stem, stage and head. The stem is usually single convolution with stride 2, but may present optional extra convolutions. The stage contains the main architecture of the model, that can be divided into blocks with layers, in which each stage block shares hyperparameters (number of channels and extra hyperparameters) across all its layers. The head may or not contain convolutions and then it is followed by average pooling, an optional linear layer and the final classification linear layer followed by a softmax. This is the case for Mobile Nets and MaxViT.

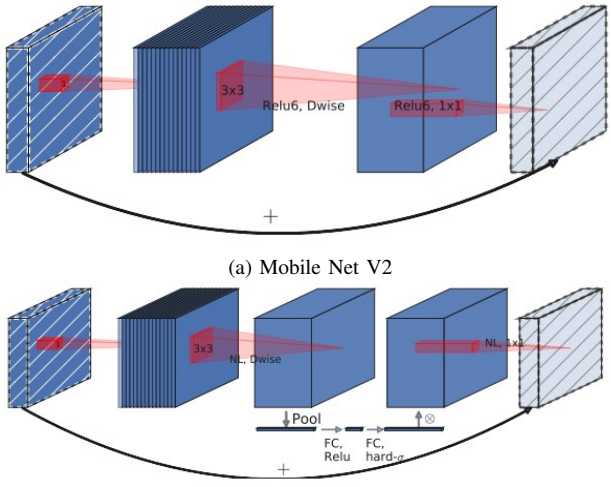
A. Convolutional Neural Networks

AlexNet [2] achieved a revolution at the fields of Computer Vision and ANN by implementing a GPU efficient convolutional operation, dropout [20] and data augmentation. Following this line, the ResNet [21] presented a residual connection to improve the gradient flow into earlier layers of deep neural networks. This is a simple design trick which is now almost ubiquitous at neural networks

More recently, mobile networks were proposed. This approach focus on building efficient models with fewer parameters. The authors of MobileNet V2 [22], shown in Figure 1, propose residual inverted bottleneck blocks, a micro-architecture that became more efficient than the reference model ResNet. The objective of this micro-architecture is to reduce data dimensionality in a way that the manifold spans the entire space of lower dimensional sub-spaces. They do so by inserting inverted bottlenecks at each block, instead of keeping the channels number constant at repeating blocks as in ResNet. That is, they first expand the number of channels for a given factor (e.g. 4), then they apply convolutions in this higher dimension and finally go back into the original dimension, similar to feed-forward networks in the transformer architecture [23]. On their architecture, they first increase the channels number with 1x1 kernels, then use 3x3 depth-wise kernels on the higher dimension and, lastly, they reduce the dimension back to what it was before with a 1x1 kernel. The reason to use 1x1 kernels followed by 3x3 is because the authors of Mobile Net V1 [24] found it more efficient than directly expanding using 3x3 kernels.

Further, on Mobile Net V3 [3], they improve Mobile Net V2 efficiency by applying a Squeeze and Excitation [25] mechanism after the 3x3 depth-wise convolution. We will refer to the block of Mobile Net V3 as the MBConv.

Besides, in EfficientNet V2 [4], the authors have changed the original architecture of MobileNet V3 into the Fused MBConv



(a) Mobile Net V2

(b) Mobile Net V3

Figure 1: Mobile Net Micro-Architectures [3]

and interpolated this new block with the original MBConv, finding the best parameter configuration given their applied model search. They have also proposed slight modifications to the compound scaling method of EfficientNet [26], which consists in adjusting model depth, channel number and image size to find the best scale inside a family of models.

We harness the parameter efficiency of Mobile Net V3 at the micro-architecture level of our proposed neural network.

B. Bio-inspired Networks

Recent advances of ANNs involving bio-inspiration are represented by the Spiking Neural Networks (SNN) [14]. They are discrete and sparse networks with a mechanism that decides wether a certain neuron should fire a spike at a given time step [27]. [15] leveraged these characteristics to adapt existing CNNs with an additional SNN module. With this, they achieved SOTA results on object detection.

C. Vision Transformers

Vision Transformers apply transformer encoder blocks to explore the attention paradigm into vision tasks. On ViT [9], as shown in Figure 2, the authors do this by dividing the image into multiple embedding patches of equal window size, in which each patch embedding is cross-attended with attention to all other patches. They also reserve a patch embedding for classification (usually referred as “cls” token in language models), which does not come from the image, and so aggregates global information about other patches. The advantage of using a pure transformers architecture is that the multiple heads mechanism turns it possible to apply tensor parallelism [28].

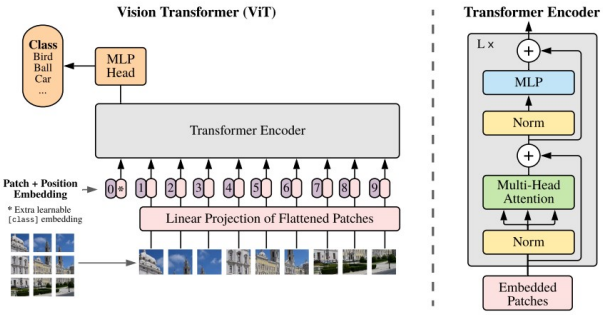


Figure 2: ViT [9].

Then, MaxViT [10] enhances ViT with more efficient attention mechanisms for the vision paradigm, the authors use Axial Attention for local details and Grid Attention for global interactions between pixels. They also append a MBConv at the start of each block.

Transformer networks harness hardware parallelism to build training-efficient networks, but they move away from bio-inspired approaches

D. Two-branch Neural Networks

Two-branch based models allow the signal to travel along two different paths, conventionally after a shared stem - except for BiSeNet. This is done to extract manifold efficiently. An example of different paths for signal propagation can be seen in BiSeNet [6] and Deep Dual-Resolution Networks [7], these architectures are built for the purpose of semantic segmentation, in which one of the branches is shallow and wide, extracting local details, and the other is deep and narrow, capturing high-level semantics. For simplicity, we show only the Deep Dual-Resolution Network at Figure 3.

Another example of two branch network is SeaFormer [8], that increases the efficiency by using mobile transformers, from which the embedding of the stem backbone is processed and fused at multiple steps with mobile transformer blocks embeddings, as can be seen at Figure 4.

This architecture achieved the best trade-off between accuracy and latency on ADE20K and Cityscapes semantic segmentation datasets.

We adopt the two-branch design at the macro-architecture level of our network, but we use no shared stem as we want to analyze shapes and colors fully separate.

III. Architecture

Yin-Yang Net uses a **micro-architcture** presented in Figure 5. In each repeating block, we start with a sub-block of ResNet

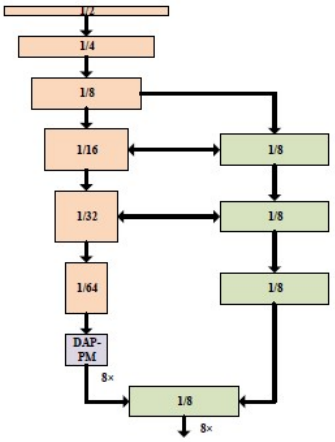


Figure 3: Deep Dual-Resolution Network [7].

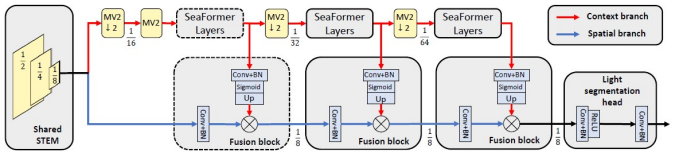
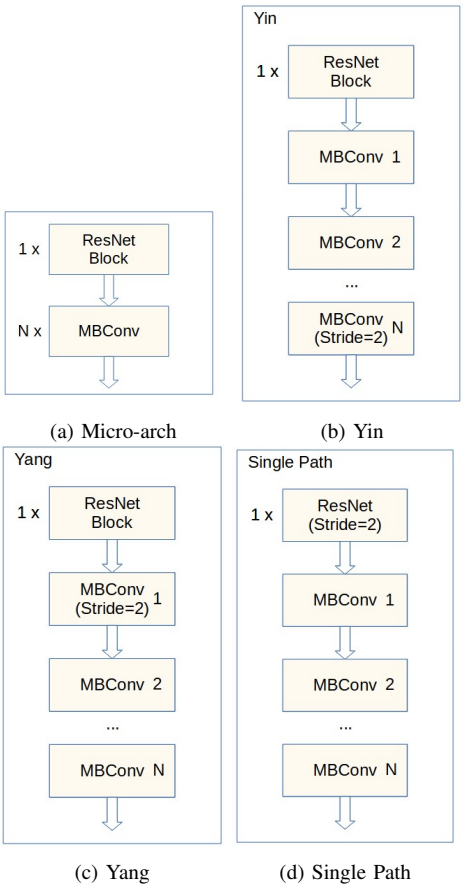


Figure 4: SeaFormer [8].

and then use n sub-blocks of MBConv from Mobile Net V3. We found this configuration has better accuracy when training solely with MBConv given our hyperparameters, and it is more efficient during training than using only ResNet blocks. At the two branch layers, stride 2 is applied on the last or the first sub-block according to the micro-architecture type, Yin or Yang, respectively. The single branch layers use stride 2 on the ResNet block.

Our work is inspired by two branch architectures. However, our approach differs from classical two branch networks in 2 aspects. First, instead of building a shallow and a deep branch for details and semantics extraction, YYNet uses the same number of layers and channels at blocks on the same level, but stride 2 at different parts of these layers. Second, in our work, there is no common stem backbone to the branches, the stem is the focus of this paper, where different manifolds are analyzed.

The Yin branch has the purpose of form analysis. Yin blocks can use the first channel of the input or the mean of all channels. We found that both configurations perform well. For simplicity, when the first channel approach is in use, the first block receives the red channel in the network input layer. This way, as there is no other color to extract, it is obliged to the task of extracting the form manifold. Also, in order to focus on local/higher scale



(a) Micro-arch (b) Yin

(c) Yang (d) Single Path

Figure 5: Micro-architecture

details, a strategy of later stride 2 is used, meaning that the last MBConv of each block applies striding.

On the other hand, the Yang branch analyzes colors. With that in mind, as colors in nearly pixel are generally the same, we use an early stride 2 on this branch to remove color redundancy and only care about different colors interactions. That is, the first MBConv on each block applies striding. This block resembles standard single branch architectures, as its input is the three RGB channels and early stride 2 is applied.

Then, at the **macro-architecture** level, as shown in Figure 6, we send the same input to both these micro-architectures. Their final embeddings have the same shape, as they have the same number of layers and channels, with the exception of the first sub-block of each micro-architecture. We then apply a Fusion Gate mechanism adapted for our embeddings.

We apply a Fusion Gate, similar to SeaFormer and Multi-modal Chain-of-Thought [29]. For this, we use an embedding fusion mechanism at outputs from Yin and Yang branches, each

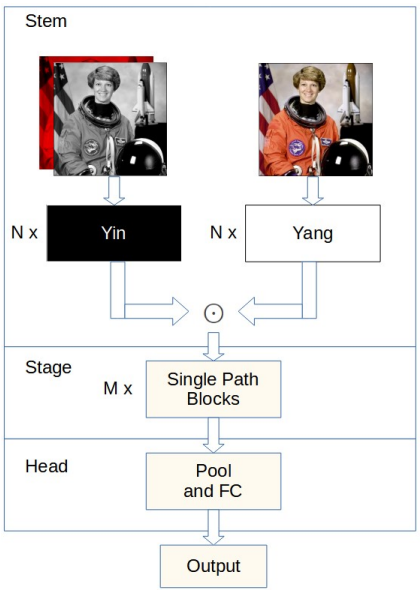


Figure 6: Macro-architecture.

branch having a different embedding meaning. This is done to unify both embeddings, or the manifold, as presented in (1).

$$SP_X = A_Y + I_Y \tag{1}$$

X represents the input of one network and Y represents an output, A represents the Yang blocks, I represents the Yin blocks, $\odot$ represents a Hadamard product (elementwise product). We have tested several combinations of A and I (as described in Section IV). The operations presented in (1) are those with best performance. On CIFAR-10, this gated fusion yielded a slightly better accuracy than the concatenation – about less than 0.5%, yet halving the channels number compared to concatenation.

After this, we send the embedding into Single Path blocks, that consists of a sub-block of ResNet with stride 2 on the first convolution and then n sub-blocks of MBConv with no stride 2 (except for CIFAR-10, as described in Section IV). We use GELU [30] as the activation function of any sub-block, since it was used by the MaxViT. As the head of our model, we use average pooling, flattening, a linear layer, GELU, dropout and the final classification linear layer followed by a softmax.

IV. Experiments and Results

A. Experiments

We tested multiple fusion approaches to combine Yin and Yang outputs. We selected the approach presented in Equation 1. However, Table I presents other approaches tested in this

work. We performed 3 runs with batch size 512 on CIFAR-10 for each approach and report the mean and standard deviation.

Table I: Fusion Approach

Formula	Mean	STD
A*(1-I)	87.61	0.09
A*I + A+I	87.63	0.08
A*(1-I) + A-I	87.81	0.19
A*I	87.87	0.23
A*(1-I) + A+I	87.98	0.22
A+I	88.21	0.46

We then reduced the batch size into 64 for the final model on CIFAR-10. We do this because batch normalization seems to work better in batch sizes on the range of 50 to 100 [1]. We use a smaller batch size in ImageNet due to computational constraints.

We used a traditional gradient clip of 1 and mixed precision for faster training/inference speed. We use the AdamW optimizer [31] since it was chosen at recent SOTA reinforcement learning models [32]. Also, one of the graphs at [31] shows that there are specific values of weight decay that works better with specific values of learning rate. We therefore use an adaptive value for the weight decay. At the end of each epoch, we set the weight decay to be equal to the learning rate*1.56 – a value which was on the optimal area demonstrated at the AdamW paper. The input resolution for CIFAR-10 is 32x32 and ImageNet is 224x224. Further hyperparameters and settings are provided in Table II. Specific Hyperparameters adjustments for CIFAR-10 and ImageNet are presented on Table III. We designed 3 models for CIFAR-10 and one for ImageNet.

Table II: General Hyperparameters and Settings

Hyperparameter	YYNet Small	YYNet
Optimizer	AdamW [31]	AdamW
LR Scheduler	One Cycle	One Cycle
Max LR	1e-2	18e-4
Epochs	40	300
Batch Size	64	32
GPU	RTX 2060	RTX 2080 TI
Dataset	CIFAR-10	ImageNet

We choose values of 1e-2 as the learning rate for the CIFAR-10 benchmark because the learning rate of [32] was a constant 1e-4, but the One Cycle schedule recommends using higher values. We reduce the learning rate at ImageNet due to the increased amount of steps.

Regarding CIFAR-10, we use a constant channel number over all the sub-blocks, exploring 3 models variants. They have 1 Yin Yang layer with 3 MBConvs and 1 single branch layer with 2 MBConvs. We apply an extra stride 2 at the first MBConv of the single branch layer on CIFAR-10 networks.

Table III: Dataset Specific Hyperparameters

Hyperparameter	CIFAR-10	ImageNet
YY Starting Channels	(16, 32, 64)	16
SP Starting Channels	(16, 32, 64)	64
Channels added per MBConv	0	2
Extra SP stride 2	Yes	No
YY Layers	1	1
SP Layers	1	4
YY MBConv per Layer	3	3
SP MBConv per Layer	2	2
Pre-Classification Linear Neurons	40	500

The linear layer before the classification layer on CIFAR-10 has 40 neurons.

Regarding ImagetNet, we start with 16 channels, then a constant number of 2 channels is added at each MBConv on the Yin and Yang branches. This is a simple heuristic adopted to lightly grow the network parameters. After that, a fixed number of channels 64 is provided for the first single branch sub-block. We then continue adding channels after each block. We also use a single layer and 3 MBConvs for the Yin Yang branches and 2 MBConvs for the single branch sub-blocks. We use 4 layers of single-branch and no extra stride 2 is applied in this dataset. The linear layer before the classification layer has 500 neurons.

B. Results

We conduct experiments with the small version of YYNet on CIFAR-10, in which we reach State-of-the-Art (SOTA) at model efficiency for models with few parameters. These results are provided in Table IV.

Table IV: CIFAR-10

Model	Test Accuracy	Parameters
ExquisiteNetV2 [33]	92.52	890,000
YYNet Small 64 channels (ours)	**93.32**	726,274
kMobileNet 16ch [34]	89.81	240,000
YYNet Small 32 channels	91.91	191,330
YYNet Small 16 channels	89.46	**52,882**

We also test a model version on the ImageNet validation dataset. We do not provide results on the test dataset, since ImageNet team only send results on the test set when the challenge is open. Comparison with similar size models are provided in Table V. Our model uses 6% less parameters than [8] reaching an validation accuracy only 1.2% smaller. Regarding with MobileNet V3 [3], the authors do not provide validation accuracy but we assume the validation accuracy is similar to test accuracy. In this case, our model is considerable smaller and presents similar efficiency.

Table V: ImageNet

Model	Validation Acc	Test Acc	Parameters
MobileNet V3 – Small [3]	-	67.4	2.5M
SeaFormer – Tiny [8]	67.7	67.9	1.7M
YYNet (ours)	66.49	-	1.6M

V. Conclusion and Discussion

In this work, we took inspiration in neuroscience research to model efficient neural networks. We developed a two branch stem for CNNs intended to analyze colors and shapes of images separately. Our model reached State-of-the-Art results on CIFAR-10 considering models with few parameters. We reached 93.32% test accuracy with 726k parameters, 0.8% more than the older SOTA in this category, while having close to 150k less parameters. Our model with 52k parameters, 17 times smaller than ExquisiteNet, lose only 3.86% test accuracy. We reached 66.49% validation accuracy on ImageNet with 1.6M parameters.

These results suggests that bio-inspiration – such as the separate processing of shapes and colors – helps on the design of efficient ANNs.

Future work includes parameter search for the Yin Yang network at the ImageNet dataset. Yin branch would also benefit from this parameter search, as it applies a stride 2 relatively late in this branch, increasing processing cost.

We also plan to investigate if our architecture is useful for other tasks beyond classification. One example is applying YYNets in generative AIs such as Stable Diffusion, by changing the latent space currently generated by U-Nets [11]. Another possible use is combining YYNets shape and color separation with architectures such as ViT, i.e, adding gray-scale input patches or queries.

References

[1] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," 2023.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[3] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[4] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[5] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollar, "Designing network design spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[6] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, vol. 129, pp. 3051–3068, 2021.

[7] H. Pan, Y. Hong, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[8] Q. Wan, Z. Huang, J. Lu, G. Yu, and L. Zhang, "Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation," 2023.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[10] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "Maxvit: Multi-axis vision transformer," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*. Springer, 2022, pp. 459–479.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[12] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," 2022.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[14] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[15] W. Li, J. Zhao, L. Su, N. Jiang, and Q. Hu, "Spiking neural networks for object detection based on integrating neuronal variants and self-attention mechanisms," *Applied Sciences*, vol. 14, no. 20, p. 9607, 2024.

[16] J. H. Elder and A. J. Sachs, "Psychophysical receptive fields of edge detection mechanisms," *Vision Research*, vol. 44, no. 8, pp. 795–813, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0042698903007533

[17] M. M. Bannert and A. Bartels, "Human v4 activity patterns predict behavioral performance in imagery of object color," *Journal of Neuroscience*, vol. 38, no. 15, pp. 3657–3668, 2018.

[18] D. Beatty, "Visual pigments and the labile scotopic visual system of fish," *Vision research*, vol. 24, no. 11, pp. 1563–1573, 1984.

[19] B. R. Conway, "Color vision, cones, and color-coding in the cortex," *The neuroscientist*, vol. 15, no. 3, pp. 274–290, 2009.

[20] G. Hinton, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[26] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[27] C. Zhou, H. Zhang, L. Yu, Y. Ye, Z. Zhou, L. Huang, Z. Ma, X. Fan, H. Zhou, and Y. Tian, "Direct training high-performance deep spiking neural networks: a review of theories and methods," *Frontiers in Neuroscience*, vol. 18, p. 1383844, 2024.

[28] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[29] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, "Multimodal chain-of-thought reasoning in language models," 2023.

[30] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[32] M. Schwarzer, J. S. O. Ceron, A. Courville, M. G. Bellemare, R. Agarwal, and P. S. Castro, "Bigger, better, faster: Human-level atari with human-level efficiency," in *International Conference on Machine Learning*. PMLR, 2023, pp. 30 365–30 380. [Online]. Available: https://proceedings.mlr.press/v202/schwarzer23a/schwarzer23a.pdf

[33] S.-Y. Zhou and C.-Y. Su, "A novel lightweight convolutional neural network, exquisitenetv2," 2022.

[34] J. P. Schwarz Schuler, S. Romaní, M. Abdel-nasser, H. Rashwan, and D. Puig, "Grouped pointwise convolutions reduce parameters in convolutional neural networks," *Mendel*, vol. 28, pp. 23–31, 06 2022.