An energy efficient model based on the feature pseudo-embedding

Lingfeng Chen[1][0009−0001−5319−0579] and Iker Pastor López[1][0000−0002−3068−6248]

University of Deusto, Deusto BI 48007, Spain[1]
lingfeng.chen@opendeusto.es
iker.pastor@deusto.es

Abstract. During the last decades, neural networks have proven to be an effective means of handling repetitive tasks that require human intelligence in industry, especially in the field of computer vision, albeit at a considerable computational cost and subsequent energy consumption. In this paper, we present a new approach for designing neural network architecture that reduces the number of model parameters, minimizes the computational cost, and improves energy efficiency without significantly impacting performance, and as an example, we present iViT (Information Visual Transformer), an architecture we designed for image classification under this methodology based on "pseudo-embeddings" and mixture of experts instead of traditional patch embedding.

1 Introduction

In recent years, the increasing size of neural networks has come with a high level of energy consumption that has caused considerable negative consequences from an environmental and an economic perspective [3]. In industry, CNNs have been proven to be a useful tool to detect anomalies, classify items, and predict failure. In medical scenarios, CNNs are also of significant importance in diagnostic tasks, making the optimization of NNs to speed them up without sacrificing performance a fundamental challenge. In this paper, we focus on improving energy efficiency [27]. in the classification tasks through architectural design.

The performance of the models has improved exponentially over the years, and so has the energy consumption. As a consequence, the improvement in energy consumption is essential in terms of the economy, environmental impact, and scalability of the models. [29]. The requirements for developing high-performance models have risen to a point that is no longer accessible for most of the companies and universities. This means that in order to develop high-performance models for specific-purpose tasks, normally heavy pre-trained models and fine-tuning are required, which usually leads to a poor efficiency/performance ratio. In order to overcome this dilemma, we present a new methodology to design the architecture of neural networks based on a structure of 3 separate blocks that can outperform large-scale models in terms of efficiency.

This paper is structured into five parts. In the State of the Art section, we briefly discuss techniques used to optimize neural networks and improve performance. We mention approaches considered for use in iViT, such as the Mixture

of Experts [16] and attention mechanisms [17]. In the Proposed Methodology section, we describe the phases of our methodology, the type of architecture we aim to achieve, the rationale behind the design, the explainability potential of the architecture, and the overall structure of iViT. We also present our experimental results, including a 95% confidence interval. We applied our model in EMNIST, FashionMNIST [5] and NEU dataset [9], and compared our model with famous models (ResNet [8], MobileNet [10], VGG [16], EfficientNet [26]) in terms of energy efficiency and performance. In the Discussion and Conclusion and Future Work sections, we address the limitations of our methodology, possible improvements, and considerations regarding the obtained results. Additional results, such as the evolution of the loss function, the performance of each tested model across various metrics, the algorithms used in the experiments, inference times, and measurement methods can be found in the GitHub repository.

2 State of the art

Currently, from a pure performance perspective, there are several approaches to improving the metrics of the models. The most popular one is increasing the size of the models when aiming for a general-purpose model or using pretrained general-purpose state-of-the-art models [33,15] and fine-tuning them when aiming for specific purposes. This approach, with a certain level of certainty, gives very good results, but it can lead to redundancy and inefficiency in the least complex tasks.

Mixture of experts is a “divide and conquer" approach that improves the efficiency of the models while enhancing performance. It consists of having several smaller models that are experts in a specific task, along with a router (or gating system) [34] with several adaptations. But this approach usually involves very difficult implementations. Our methodology solves this problem by training a population of small experts to encode the features of the input, a kind of pseudo embedding, and pruning the experts after the training process. These experts also allow us to interpret how the model is much better.

A significant improvement in neural networks was the introduction of transformers and the attention mechanism [28]. This concept is at the core of the current SOTA models [15]. Instead of the classic patch embeddings, we used pseudo embeddings of features instead of the traditional embeddings [1], since we want our model to generalize in different test distributions.

One of the most interesting approaches to achieving the SOTA in specific-purpose tasks is NAS (neural architecture search) algorithms, with genetic algorithms being the most interesting one [17]. This strategy usually comes with a much higher cost in the training process to achieve the final model, since it involves a search phase that also implies several training phases, which, alongside the complexity of the implementation, means that the hardware necessity is also higher, as well as the energy and time cost. Considering the number of parameters that an architecture may have, our idea proposes to maintain the smallest number of variables possible.

3 Proposed methodology

We proposed a new methodology based on 3 phases (**extraction of features, extraction of information, decision making** that an architecture must represent), and as an example, we present an architecture called iViT (Information Visual Transformer). The objective of our methodology is to create architectures where these 3 phases are separated clearly to improve the explainability of the model, minimize the number of computations, and maximize energy efficiency.

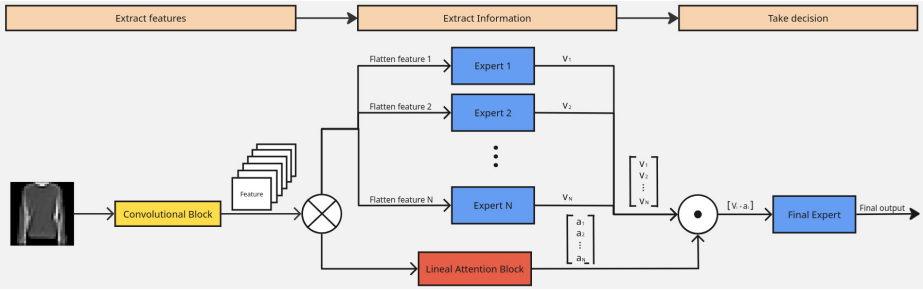

Fig. 1: The main architecture of the proposed methodology (the normalization layers are not shown here; for the explicit details, check the source code; batch norm and pool are not shown)

This methodology has mainly 5 phases:

1. We built a segment in charge of extracting features from the given input; for obvious reasons, we used convolutional neural networks.
2. For each feature, we generate a pseudo-embedding independently that contains the necessary information to accomplish the task. Features that contain no necessary results will be discarded by the attention mechanism.
3. We concatenate all the vectors into a single one and perform our task. This is our **decision taker**; in our case, we build an MLP (Multi-Layer Perceptron).
4. We train and evaluate the model by splitting our dataset into train, validation, and test, and we use train to train the model and valid and test to evaluate the model.
5. After training the model, we pruned the unnecessary experts by the following approaches:
   (a) 1. Use the valid and the test set to predict and record the output of the experts and the attention mechanism. If the product of some expert or attention mechanism is always a zero-tensor, we prune that expert.
   (b) 2. Use explainability techniques to evaluate the impact of each expert on the final decision by mapping the output of the expert to the final output of the model. If the expert has almost no impact, we can remove the expert and fine-tune the model.

The architecture of the model must have these 5 phases separated clearly, for these reasons:

1. It makes it much easier to interpret the model and detect potential undesired behavior after the training process. Which also allows us to find better ways to improve efficiency and evaluate potential issues or improvements that the model may have.
2. This allows us to discard useless experts permanently after the training process and filter the features of the input data that give no information. We can use the output of the experts for our task using other methods, such as decision trees, which also give us more flexibility. The idea is to minimize the necessary computation.
3. By recording the output of the experts and the output of the decision-maker, we can have a brief check of how the model is making decisions based on the given information using explainability techniques for deep learning models, which also provides us a traceable way back to the data. As an example, we perform a CART decision tree with the vectors of the experts as input and the final output of the model as the target value, and another tree with the ground truth as the target value; finally, we extract the tree plot. This process must be performed using the validation or test set (i.e., inputs the model has never seen before) to avoid potential overfitting that would yield unrealistic results. This is a very simple approach, but it can give us a brief check of how the model is working.
   By targeting the model output, the nodes that are just above the leaf nodes are the least important experts, and the top nodes are the most important experts. In the results, you can see an image of the whole tree plot. With this information, we can prune experts and fine-tune the model by removing the lowest experts in the tree. The interpretation is more precise when the predictions of the trees are more accurate.
   By targeting the real label, we see how much the vectors of the experts explain about the given data for the given task, and it also tells us how our final block that makes the decision (based on this information) is working. If the performance of the tree is higher than that of the final decision-taker (in this experiment, an MLP), it means that the vector of the experts explains the given information sufficiently well but exceeds the capacity of the final model to assimilate it; otherwise, it means the decision-taker is working fine. In some cases the decision tree may perform better than the original decision-taker. In this case it implies that the decision-maker has a lack of capacity to interpret the pseudo-embeddings.
4. Processing each feature using separate experts allows us a much easier and more efficient base to perform NAS.

iViT is based on the extraction of the features using convolutional layers and an MoE, where there is an initial population where each feature corresponds to an expert, giving the result a tensor per feature of numbers that codifies the information into a pseudo embedding. We used an attention mechanism (for this

experiment we used linear attention) that provides a single scalar value for each feature, resulting in a tensor of attention values, which are the weights of the experts. These experts are created automatically with the random initialization of weights and adjusted during the training process. After the training process, we prune some experts if they are redundant under the proposed criteria; this reduces the size of the models by 5-30% (it depends on the dataset and the pruning strategy) without impacting performance significantly. Then another block of the architecture called **Decision Taker** will use the result of the MoE process as a final result. Figure 1, you can check the main architecture.

The experts consist of dense layers followed by a batch normalization layer. The main idea of the experts is to map simple decisions of the small features into a tensor of k numbers. This forces the architecture to use these numbers to predict instead of classical convolutional layers followed by a couple of dense layers. The use of the attention mechanism helps to filter out irrelevant parts and gives more importance to certain features and increases the sparsity of the experts.

The number of digits required to encode the information, denoted by k, depends on the dataset. A simple Neural Architecture Search (NAS) can help find the optimal k; we performed a limited Breadth-First Search (BFS) to explore candidate values to study how the performance changes depending on the pseudo-embedding dimension (check figure 4).

After training, we form the pruning dataset as the union of the test and validation, which is then used to prune the experts. We apply the following pruning criterion: if for the valid and test the expert always returns a null vector, then the expert is considered redundant and can be pruned. This strategy has no impact on the model in terms of performance.

4 Experiment and results

We performed the methodology on different datasets: EMNIST-digits, EMNIST-balanced, FashionMNIST, and the NEU dataset. The experiment was conducted on an Intel i9-14900K and an Nvidia RTX 4090 running Ubuntu LTS 24.04 using PyTorch as the main framework [22]. We processed the images in grayscale and in original dimension (each dimension with a different variation of the architecture of iViT) and split the dataset into train and test for the MNIST datasets and train, valid, and test for the NEU dataset. We also replicated the experiments from other models to compare the results. Some models could not be replicated because no source code was provided, but the number of parameters can provide an approximation of the energy cost. Check Figure 6 and Table 4.

For this experiment, we aimed to evaluate how the proposed methodology works in terms of energy efficiency and performance while also comparing other models and observing how energy consumption evolves with the number of parameters. You can check the results in Figure 6. We trained the model with Adam as the optimizer and cross-entropy as the loss function. We used a learning rate of 0.00125 and 1000 epochs. We used Adam as the optimizer with a

learning rate of 0.000125 and 1000 epochs. The training time was 0.7 hour per model with 8 minutes of error. No learning rate schedule was applied, nor were transfer learning techniques.

As we explained before, the size of the vectors that are the output of the experts can directly impact the performance of the model, so in order to study how the dimension of the "pseudo-embeddings" impacts the results, we trained for each size (1-10) (an arbitrary search range) and recorded the metrics by using the test set. Check the results at Figure 4. And at the same time, we record the output of the experts, the attention values, the prediction, and the ground truth of our best model in order to prune the experts; we use these records to generate the decision trees and the attention-based GradCAM 5. We also calculate how many kWh the training process of each model costs 2.

In order to study the energy efficiency of the models, the CPU consumption of the machine has been monitored using "psutil." [24] In the literature, it is common to use tools such as "RAPL" [14] or "HWMonitor" [6]; however, these are no longer applicable to the latest Intel processors, so alternatives based on the firmware of physical devices have been sought. For the GPU, consumption has been monitored through NVIDIA's official driver tools (such as "nvidia-smi," [32] in the case of Linux) for the training process in parallel.

We also reproduced experiments from other architectures to compare results. Table 4 shows our best model's performance compared to these approaches, demonstrating that our method achieves competitive results on this task. Additionally, we applied our methodology to a more complex dataset: NEU dataset (see Table 1). The results across different datasets are presented in Tables for various iViT models, denoted iViT-d, where d is the number of features extracted (pseudo-embedding dimension).

Table 1: Performance of iViT in the NEU Dataset for Metal Defect Classification

Model	Accuracy	Parameters	Model Size
iViT-64/2	94.3%	0.53 M	2.3 MB
DECAF+MLR[7]	99.7%	-	244 MB
SDC-SN-baseline [4]	99.7 %	1.27 M	3.2 MB

In Figure 6, we show the training cost in terms of energy. Within the size range of 1–10, the output dimension of the experts does not significantly affect the training energy cost, even though the overall model size increases. This is due to the gating mechanism, which activates only the necessary experts. However, as shown in Figure 4, the dimension of the experts' output vectors does have a direct impact on performance. We observe a gradual variance in performance with a clear maximum and minimum that depend on the dataset. Figure 6 shows how each model architecture evolves in terms of energy consumption according

Table 2: Comparison of the models in performance and energy efficiency in Fashion-MNIST

Model	Accuracy	Parameters	J/img
iViT-64/1 (Proposed)	89.44%	0.4 M	0.013J
Resnet18 [21]	93.20%	11 M	0.828J
Mobilenet [21]	93.96%	3.5 M	0.088J
Efficientnet [21]	93.64%	5.3 M	0.098J
MCNN15 [21]	94.04%	2.6 M	-
HSViT-C4A8 [31]	95.92%	6.9 M	-
MixMobileNet [20]	95.37%	7.3 M	-
TSD-B [25]	96.56%	3.8 M	-

Table 3: Comparison of the models in performance and energy efficiency in EMNIST-Digits

Model	Accuracy	Parameters	J/img
iViT-64/2 (Proposed)	99.43%	0.36 M	0.012J
resnet18 [8]	99.58%	11 M	0.828J
Mobilenet [10]	99.39%	3.5 M	0.088J
Efficientnet [21]	99.53%	5.3 M	0.098J
VGG-5 [13]	99.75%	3.6 M	-
CNN (2 Conv, 2 Dense) [2]	99.46%	1.2M	-

to its number of parameters. The results indicate a generally linear increase in consumption.

5 Discussion

The iViT architecture has demonstrated remarkable performance despite having a notably low number of parameters. The use of pseudo-embeddings spares us from the cold-start problem that is often very prominent in traditional ViT models employing patch embeddings. Its energy efficiency and compact model size extend the reach of AI to devices with lower computational power, where highly specialized tasks become far more appealing.

In Figure [23] you can see the model's Grad-CAM. Some interesting patterns emerge, but without additional contextual data for the images, it's not possible to confirm any of these patterns as a demonstrable description of the pattern beyond empirical tests through the trees. Therefore, a strategy is needed to allow us to verify the patterns in a demonstrable way.

Despite the low cost of the proposal. And the automatization of the expert pruning and local search to find the best model is practical and is the easiest

Table 4: Comparison of the models in performance and energy efficiency in Balanced-MNIST

Model	Accuracy	Parameters	J/img
iViT (Proposed)	85.78%	0.34 M	0.017J
Resnet18 [8]	87.74%	11 M	0.855J
Mobilenet [10]	86.40%	3.5 M	0.088J
Efficientnet [26]	85.37%	5.3 M	0.097J
VGG-5 [13]	90.71%	3.6M	-
WaveMix-Lite-128/7 [12]	91.06%	2.4 M	-
EfficientNAS [20]	91.48%	2.25	-
CNN (2 Conv, 2 Dense) [2]	87.18%	1.2M	-

approach in implementation terms. But we think that it can be improved with genetic algorithms that mix experts of different sizes of the informational vector. The proposed model achieved great efficiency, but due to the lack of computational resources, we haven't tested it on a large-scale dataset like ImageNet or CIFAR-100, which does not let us know what the performance of the model is for general-purpose tasks.

6 Conclussion and future work

It is still necessary to develop more criteria and strategies to prune the experts in order to minimize the size of the decision-maker, since the proposed ones are very simple and greedy. Many classical model reduction techniques like weight pruning [18] or expert distillation [30] could have been applied. The proposed methodology presents a recipe to follow when designing a neural network, which can also be applied in other fields of deep learning than computer vision. The ability to focus on key features of the given inputs and discover relations between the variables, plus the interpretability of the structure, helps to discover information and knowledge that may lead to developing new explainability techniques.

In terms of the energy efficiency architecture of the proposed methodology, it shows a huge advantage over the biggest ones, which means that we must try to build a model to test if the model is able to generalize, especially with huge datasets from general-purpose models such as ImageNet or CIFAR, but since the model leads very easily to biases, we cannot discard the possibility of this leading to undesirable results.

The application of the knowledge distillation of the decision maker would also be interesting to reduce the size of the final part of the architecture, since it is the only part of iViT that we haven't developed any technique to reduce the size of, and there must be some computation that is not really necessary.

The potential implementation of iViT in devices such as FPGAs [19] may considerably improve the energy efficiency of the model. Since we are reduc-

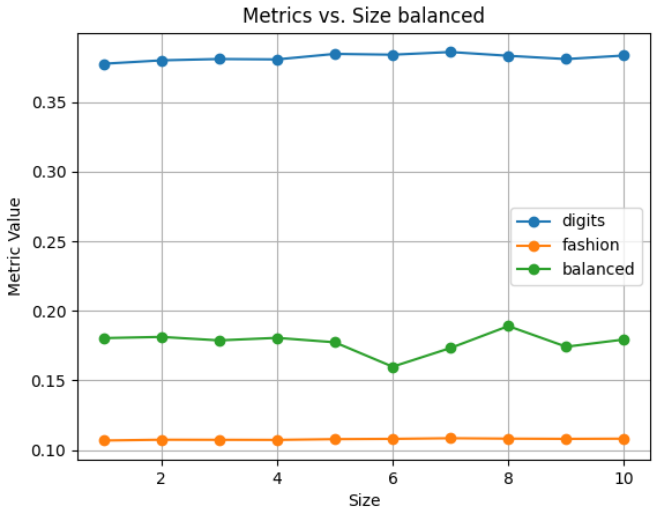

Fig. 2: The energy cost of the training process

ing the number of parameters by pruning the experts, the final model is much smaller, which may partially help with the memory problem of the FPGA. We also provide the quantized models in case the reader wants to implement them [11].

The figures we mentioned before can be observed in the last pages. The source code can be found at Github Repository. Specific instructions to replicate the code and result can be found in the README.md.

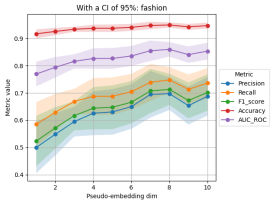

Fig. 3: Fashion-MNIST

Fig. 4: The evolution of the performance related to the pseudo-embedding size. The metrics are micro-averaging to enhance the difference between variants of iViT

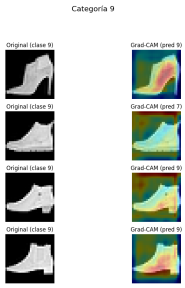

Fig. 5: Ankle boot

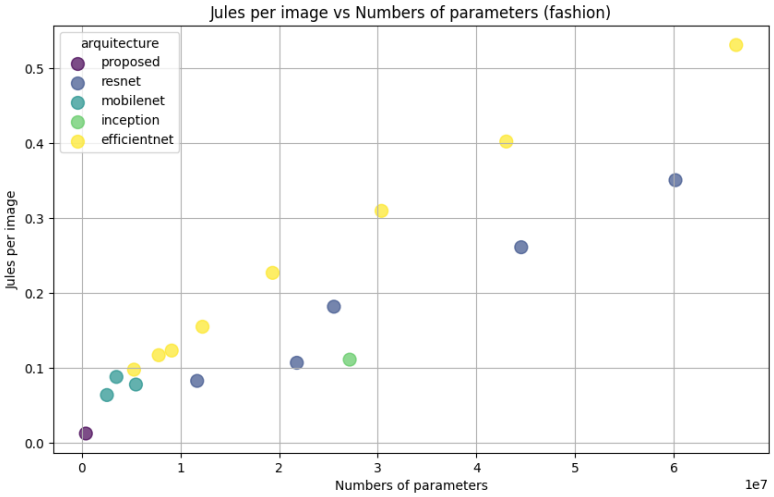

Fig. 6: The evolution of the energy cost of different architectures

References

1. Almeida, F., Xexéo, G.: Word embeddings: A survey. CoRR **abs/1901.09069** (2019), http://arxiv.org/abs/1901.09069
2. Cavalin, P., Soares de Oliveira, L.: Confusion Matrix-Based Building of Hierarchical Classification: 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings, pp. 271–278 (03 2019). https://doi.org/10.1007/978-3-030-13469-3_32
3. Chakraborty, S.: Towards a comprehensive assessment of ai's environmental impact (2024), https://arxiv.org/abs/2405.14004
4. Chazhoor, A.A.P., Ho, E.S.L., Gao, B., Woo, W.L.: A review and benchmark on state-of-the-art steel defects detection. SN Computer Science **5**(1), 114 (Dec 2023). https://doi.org/10.1007/s42979-023-02436-2, https://doi.org/10.1007/s42979-023-02436-2

5. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: an extension of MNIST to handwritten letters. CoRR **abs/1702.05373** (2017), http://arxiv.org/abs/1702.05373
6. Fieni, G., Acero, D.R., Rust, P., Rouvoy, R.: PowerAPI: A Python framework for building software-defined power meters. Journal of Open Source Software **9**(98), 6670 (Jun 2024). https://doi.org/10.21105/joss.06670, https://hal.science/hal-04601379
7. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics **SMC-3**(6), 610–621 (1973). https://doi.org/10.1109/TSMC.1973.4309314
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
9. He, Y., Song, K., Meng, Q., Yan, Y.: An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. IEEE transactions on instrumentation and measurement **69**(4), 1493–1504 (2019)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017), http://arxiv.org/abs/1704.04861
11. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. Journal of Machine Learning Research **18**(187), 1–30 (2018), http://jmlr.org/papers/v18/16-456.html
12. Jeevan, P., Viswanathan, K., S, A.A., Sethi, A.: Wavemix: A resource-efficient neural network for image analysis (2024), https://arxiv.org/abs/2205.14375
13. Kabir, H.M.D., Abdar, M., Jalali, S.M.J., Khosravi, A., Atiya, A.F., Nahavandi, S., Srinivasan, D.: Spinalnet: Deep neural network with gradual input. CoRR **abs/2007.03347** (2020), https://arxiv.org/abs/2007.03347
14. Khan, K.N., Hirki, M., Niemi, T., Nurminen, J.K., Ou, Z.: Rapl in action: Experiences in using rapl for power measurements. ACM Trans. Model. Perform. Eval. Comput. Syst. **3**(2) (Mar 2018). https://doi.org/10.1145/3177754, https://doi.org/10.1145/3177754
15. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM computing surveys (CSUR) **54**(10s), 1–41 (2022)
16. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). pp. 730–734 (2015). https://doi.org/10.1109/ACPR.2015.7486599
17. Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G.G., Tan, K.C.: A survey on evolutionary neural architecture search. IEEE transactions on neural networks and learning systems **34**(2), 550–570 (2021)
18. Liu, Z., Sun, M., Zhou, T., Huang, G., Darrell, T.: Rethinking the value of network pruning. arXiv preprint arXiv:1810.05270 (2018)
19. Ma, Y., Cao, Y., Vrudhula, S., Seo, J.s.: Optimizing the convolution operation to accelerate deep neural networks on fpga. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **26**(7), 1354–1367 (2018). https://doi.org/10.1109/TVLSI.2018.2815603
20. Meng, Y., Wu, P., Feng, J., Zhang, X.: Mixmobilenet: A mixed mobile network for edge vision applications. Electronics **13**(3) (2024). https://doi.org/10.3390/electronics13030519, https://www.mdpi.com/2079-9292/13/3/519

21. Nocentini, O., Kim, J., Bashir, M.Z., Cavallo, F.: Image classification using multiple convolutional neural networks on the fashion-mnist dataset. Sensors **22**(23) (2022). https://doi.org/10.3390/s22239544, https://www.mdpi.com/1424-8220/22/23/9544
22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
23. Raghavan, K., B, S., v, K.: Attention guided grad-cam : an improved explainable artificial intelligence model for infrared breast cancer detection. Multimedia Tools and Applications **83**(19), 57551–57578 (Jun 2024). https://doi.org/10.1007/s11042-023-17776-7, https://doi.org/10.1007/s11042-023-17776-7
24. Rodola, G.: psutil: process and system utilities (Version 7.0.0). https://pypi.org/project/psutil/ (Feb 2025), accessed: 10 May 2025
25. Shao, R., Bi, X.J.: Transformers meet small datasets. IEEE Access **10**, 118454–118464 (2022). https://doi.org/10.1109/ACCESS.2022.3221138
26. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. CoRR **abs/1905.11946** (2019), http://arxiv.org/abs/1905.11946
27. Tripp, C.E., Perr-Sauer, J., Gafur, J., Nag, A., Purkayastha, A., Zisman, S., Bensen, E.A.: Measuring the energy consumption and efficiency of deep neural networks: An empirical analysis and design recommendations (2024), https://arxiv.org/abs/2403.08151
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
29. van Wynsberghe, A.: Sustainable ai: Ai for sustainability and the sustainability of ai. AI and Ethics **1**(3), 213–218 (Aug 2021)
30. Xiang, L., Ding, G., Han, J.: Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification (2020)
31. Xu, C., Li, C.T., Lim, C.P., Creighton, D.: Hsvit: Horizontally scalable vision transformer (2024), https://arxiv.org/abs/2404.05196
32. Yang, Z., Adamek, K., Armour, W.: Accurate and convenient energy measurements for gpus: A detailed study of nvidia gpu's built-in power sensor. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis. SC '24, IEEE Press (2024). https://doi.org/10.1109/SC41406.2024.00028, https://doi.org/10.1109/SC41406.2024.00028
33. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: Contrastive captioners are image-text foundation models (2022), https://arxiv.org/abs/2205.01917
34. Yuksel, S.E., Wilson, J.N., Gader, P.D.: Twenty years of mixture of experts. IEEE transactions on neural networks and learning systems **23**(8), 1177–1193 (2012)