Exploring Diversity in Neural Architectures for Safety

Michał Filipiuk[1], Vasu Singh[1]

[1]NVIDIA, Einsteinstraße 172, Munich, Germany

Abstract

Apart from the predominant convolutional neural networks (CNNs), several new architectures like Vision Transformers (ViTs) and MLP-Mixers have recently been proposed. Research also shows that these architectures learn differently. Ensembles based on different state-of-the-art neural architectures thus provide diversity, an important characteristic in designing safety-critical systems. To quantify the benefit of ensembles, we investigate different metrics like error consistency and diversity metric that have been proposed in the literature. We observe that with comparable individual performance, an ensemble of diverse architectures performs not only more accurately than an ensemble of one architecture, but also more robustly to diverse input corruptions.

Keywords

diversity, ensemble, safety, deep learning, image classification, robustness, safety-critical systems

1. Introduction

The development of safety-critical systems relies on stringent safety methodologies, designs, and analyses to prevent hazards during operation. Automotive safety standards like ISO26262 [1] and ISO/PAS 21448 [2] mandate methodologies for system, hardware, and software development for automotive systems. Diversity is an important concept in safety-critical systems that prevents against common cause failures. For example, diversity in hardware is provided through lockstep execution across different HW engines. Diversity in software is guaranteed through diverse algorithmic implementations.

Deep neural networks [3] based on convolutional neural networks (CNN) are well-known for vision tasks using machine learning. These include safety-critical applications like autonomous driving and robotics, where CNN models are used for object detection and image segmentation as perception units to process sensor data. Over the last few years, new neural architectures have disrupted the dominance of CNNs in vision tasks: Vision Transformers (ViTs) [4], inspired by the transformer model [5] that was originally proposed for natural language processing (NLP) tasks, leverages self-attention layers instead of convolution layers to process the input split into set of non-overlaping patches. Similarly, MLP Mixers [6] have been proposed as a competitive but conceptually a simple alternative that - instead of convolutions or self-attention - are based entirely on multi-layer perceptrons (MLPs) that are repeatedly applied across either spatial locations or feature channels.

To improve the confidence in prediction, ensembles [7] of neural networks are commonly used. Multiple models are trained on the same data, then each of the trained models is used to make a prediction before combining the predictions in some way to create the final prediction. Ensembles have also shown to reduce the variance [8]. The inherent diversity in an ensemble has been shown to be a key factor for their superior performance. Different diversity metrics have been proposed in the machine learning literature. Error consistency [9], based on the Cohen's kappa metric, measures the similarity of classification normalized by chance of common prediction. Diversity [10] allows to define diversity metrics based on different loss functions.

The objective of our work is to quantify the diversity of ensembles created using different models, and evaluate their benefits. We choose two CNNs, two ViTs, and two MLP Mixers, and create 30 in total ensembles by averaging the models' outputs. Our results show that ensembles created using different architectures are more diverse than ensembles from the same architecture. We show that an ensemble of different architectures with similar accuracy further improves the performance. In our experiments, we observe the best ensemble results for a CNN and a ViT.

The paper is organized as follows. Section 2 describes the properties of CNNs, Vision Transformers, and MLP-Mixers, how they compare to each other including a summary of related work, and an overview of different diversity metrics. Section 3 provides our experimental results. Section 4 concludes the paper with a summary of our ongoing work and future directions.

2. Background

We describe the evolution of different neural architectures and their strengths and weaknesses.

The IJCAI-ECAI-22 Workshop on Artificial Intelligence Safety (AISafety 2022), July 24–25, 2022, Vienna, Austria

mfilipiuk@nvidia.com (M. Filipiuk); vasus@nvidia.com (V. Singh)
0000-0003-4926-8449 (M. Filipiuk)

2.1. Neural architectures

Convolutional Neural Networks. The convolution operation predates the first convolutional neural networks. With hand-engineered features, it was used in classical computer vision applications many years before it appeared in first neural networks in 1980s. However, the rise of CNNs started with AlexNet in 2012, which defeated by a large margin other, non-neural approaches in the ImageNet competition. Over the last 10 years, we have seen multiple improvements to this architecture, but they were more evolutionary than revolutionary.

The fact that convolutions managed to be in the spotlight for such a long time may seem quite surprising, however an analysis of their properties gives us the answer: Convolutions have two key inductive biases that allow them to excel at high-dimensional data with strong spatial correlation like images: the spatial inductive bias allows them to focus on local information in the input images. Applying the same kernel over the whole image results in the translation equivariance as input translations result only in the shifted output of convolutional layers. The convolution operation is also a very simple and compute-efficient operation. Its memory usage is not only small, but also constant with regard to the size of the image what combined with possibility to apply it in parallel, makes it feasible for every hardware.

Vision Transformer. The Transformer architecture [5] was initially introduced in 2017 for NLP tasks. In 2020, this architecture was applied to image classification problem and called the Vision Transformer (ViT) [4]. Here, an input image is split into a set of non-overlapping patches, which after being embedded are provided to the ViT encoder blocks. ViTs have much less image-specific inductive bias than CNNs. In CNNs, the locality and translation equivariance are inherent to convolutional layers throughout the whole model. In ViT, the self-attention layers are global, and only the MLP layers are performed locally and translationally equivariant on the patch level. The two-dimensional neighborhood is not present in the network architecture as transformers treat the input as an unordered set. This information needs to be input to the first layer in form of position embedding together with image patches.

Reducing the inductive biases has twofold consequences: Transformers have to learn properties that would otherwise be inherited from the convolution operation, that proved to be successful: to be invariant to the input shifts and balancing the local and global perception in encoding blocks. But at the same time, they can improve upon them, can leverage the global perception to their advantage and discover its own priors based on data, what results in performing the task distinctly and bringing diversity of solutions to the field.

MLP-Mixer. Presented in 2021, MLP-Mixers [6] provide an alternative to CNNs and ViTs that does not use convolutions or self-attention. Mixers use two types of MLP layers: channel-mixing and token-mixing MLPs. The channel-mixing MLPs are applied to every patch separately, exchanging the information between channels, while the token-mixing MLPs work on one channel, but across all patches, allowing the communication between the patches.

Matrix multiplications in MLPs are a simpler operation than a convolution, which require more specialized hardware or a costly conversion to a matrix multiplication operation.

As MLP-Mixers perform similarly to Vision Transformers on a level of encoder layers, they have similar properties: both architectures have global perception fields and they both suffer of no translation equivariance due to the use of image patches as input. Regarding the differences of these two architectures: MLP-Mixers do not need position encoding as MLP layers differentiate between different elements of its input, in contrast to the multi-head attention in ViTs.

2.2. Related Work

As the three architectures present different approaches to image classification — using convolutions, multi-head attention, or multilayer perceptrons to process the input — the comparison between them should not restrict just to experimental accuracy, e.g. on a single dataset like ImageNet, but should also include more experiments, analyzing in-detail the different aspects of image classification problem (e.g. robustness to input corruption or transformations like translations or rotations) and internal properties of each model. Bhojanapalli et al. [11] conduct multiple experiments, assessing the robustness of Vision Transformers to multiple corruptions with regard to model sizes and their pre-training datasets, in comparison to various ResNet models. They show that (1) adversarial attacks like Fast Gradient Sign Method and Projected Gradient Descent similarly influence both ViTs and CNNs, (2) corrupted images with an attack are not transferable, resulting in only a modest, few percentage points drop between the architectures, while they are transferable between the models of the same architecture. Regarding less artificial corruptions and distribution shifts, present in ImageNet-C, -R, and -A datasets: performance of different architectures seems to be similar. One important conclusion is how the accuracy changes with the size of the pretraining dataset – for ILSVRC-2012, ViTs perform worse than CNNs, however for ImageNet-21k and JFT-300M performance is comparable. Under a closer inspection of ImageNet-C dataset, ViTs and CNNs perform significantly different on various ImageNet-C corruptions: e.g. on glass blur Vision Transformers perform significantly better than CNNs, while they perform worse on contrast corruption, on the highest level of severity – this observation is crucial for our research presented in this paper. Naseer et al. [12] extends this comparison to e.g. input occlusions or input patches permutation, where ViTs perform much more robustly than CNNs. They investigates also the shape-texture bias of these architectures and show that transformers are less biased towards local textures than CNNs.

In [13], authors analyze the information that every layer processes, how the reception fields looks like for Transformers (which are not restricted by the convolution operation) and how different layers learn depending on the dataset size. Their research shows that CNNs and ViTs perform their computation significantly differently. It also briefly describes how MLP-Mixers behave closer to ViTs with regard to the intermediate features learned.

There have also been architectures that combine CNNs and ViTs. For example, *Cvt: Introducing convolutions to vision transformers* [14] apply convolutions over input image and intermediate feature token maps, which are next processed by a transformer block. While the Swin Transformer [15] doesn't feature convolution layers, it introduces a hierarchical approach of CNNs and the locality of convolutions to transformers: it applies MHA to small, local set of patches (windows), while the patches are being merged into bigger patches as we progress deeper into the model. To support the information propagation between patches, the model shifts the windows with every layer to overlap with previously used windows. These changes can also be introduced to a MLP-Mixers, resulting in the performance improvement.

The results of the aforementioned research inspire us to investigate how this variety of these three architectures, proved by multiple various experiments, can be leveraged for improving the diversity in safety-critical systems.

2.3. Diversity metrics

While the intuition behind the diversity may be straightforward, quantifying it is not. We present below three distinct metrics from the literature that try to capture models' diversity.

Ortega et al. [10] provide a metric of diversity for different loss functions like 0/1 loss, cross-entropy loss, and squared loss. As we are focused on the classification problem, we'll use 0/1 and cross-entropy losses, which formulas are presented below:

$$\begin{aligned}\mathbb{D}_{0/1}(\rho) &= \mathbb{E}_\nu\left[\mathbb{V}_\rho\left(1\left(h_W\left(\boldsymbol{x};\boldsymbol{\theta}\right)\neq y\right)\right)\right]\\ \mathbb{D}_{ce}(\rho) &= \mathbb{E}_\nu\left[\mathbb{V}_\rho\left(\frac{p(y\mid\boldsymbol{x},\boldsymbol{\theta})}{\sqrt{2}\max_{\boldsymbol{\theta}} p(y\mid\boldsymbol{x},\boldsymbol{\theta})}\right)\right]\end{aligned}$$

where $\mathbb{E}_\nu$ and $\mathbb{V}_\rho$ stand for an expected value over the whole data generating distribution ν (which is approximated using a dataset) and a variance of models' predictions that the ensemble consists of. The formulas are derived from a loss analysis of every classifier and their ensemble, where the diversity upper bounds a difference between an averaged loss of classifiers and the loss of their ensemble. In summary: these metrics measure how diverse the predictions of different models for a dataset are by calculating the variance of prediction, averaged over every data point. In case of CE diversity, the predictions are being additionally scaled to [0,1] range.

From our perspective, the CE loss diversity should be more interesting as we are going to ensemble models by averaging their prediction, but CE loss diversity is more complex than 0/1 diversity and eventually we evaluate models using accuracy, which binarizes their outputs to count them as correct and incorrect classification. At the same time, CE loss diversity is able to provide us with more information e.g. in a case when both models classify identically, but with different probabilities assigned.

Error consistency [9] is a metric measuring how much errors of two classifiers coincide. It calculates a number of items classified either correctly or incorrectly by both models and compares it to an expected rate of equal responses in case when both models were totally statistically independent. The exact formula is presented as follows:

$$\kappa = \frac{c_{obs} - c_{exp}}{1 - c_{exp}}$$

where c_{obs} stands for a fracture of equal classification (either correct or incorrect) and c_{exp} is an expected rate of equal responses, which is calculated using models' accuracies: $c_{exp} = acc_1 acc_2 + (1 - acc_1)(1 - acc_2)$. This metric can only compare two models in contrast to the diversity metrics which does not have such a restriction.

3. Experiments

Model selection. Setup. We have chosen the best performing models that were available to us at the time of conducting the research, pretrained on ImageNet-21k and fine-tuned to ImageNet-1k. We considered the arguments raised in the previous section to determine the size of the pretraining dataset. This has the best potential to perform robustly on ImageNet-C [16], which we'll use to compare the architectures. ImageNet-C is a dataset created by artificially applying various corruptions (blurs, noises, digital corruptions, and weather conditions), which feature different severity levels, to the ImageNet (ILSVRC2012) validation set. The models are as follows, and ensembles are created by averaging the returned softmax outputs of two models. We use only two at the time to observe how ensembles of different architectures perform compared to the single models that build them. Also using more models in the ensembles would prohibit us from using the error consistency metric. Ensembles are created by averaging the softmax outputs as it is the simplest way of building ensembles. While it has its disadvantages (e.g. models are calibrated differently and overconfident ones can dominate under-confident ones with their predictions), we choose it for its simplicity, leaving potential improvements to future work.

Vision Transformers:

- Vision Transformer B/8 (86M parameters)[1]
- Vision Transformer L/16 (307M parameters)[2]

Convolutional Neural Networks [17]:

- ConvNeXt-Base (89M parameters)[3]
- ConvNeXt-XLarge (350M parameters)[4]

MLP-Mixers:

- MLP-Mixer B/16 (59M parameters)[5]
- MLP-Mixer L/16 (207M parameters)[6]

Using six distinct models allows us to create 30 different ensembles that are used for the experiments. We do not create the ensemble of a model with itself.

To compare the models, apart from the diversity metrics, we use Top10 accuracy and the retention metric [18] (an accuracy on corrupted dataset divided by the accuracy on the original data). We picked Top10 accuracy to smoothen out the achieved scores as some images from ImageNet may contain multiple objects of different classes, which introduces variance to the accuracy prediction.

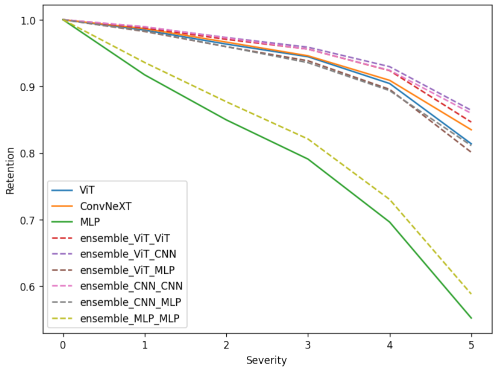
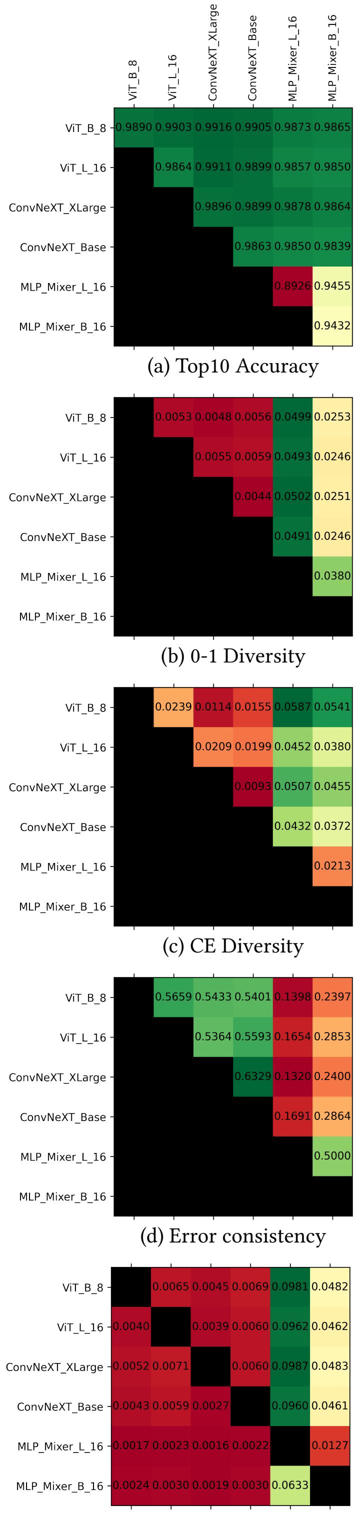


Figure 1: Retention curves with regard to severity, averaged over all ImageNet-C corruptions



(a) Top10 Accuracy

(b) 0-1 Diversity

(c) CE Diversity

(d) Error consistency

(e) 0-1 Div. Components

Figure 2: Metrics performance on original data

[1] available here: https://storage.googleapis.com/vit_models/augreg/B_8-i21k-300ep-lr_0.001-aug_medium2-wd_0.1-do_0.0-sd_0.0--imagenet2012-steps_20k-lr_0.01-res_224.npz

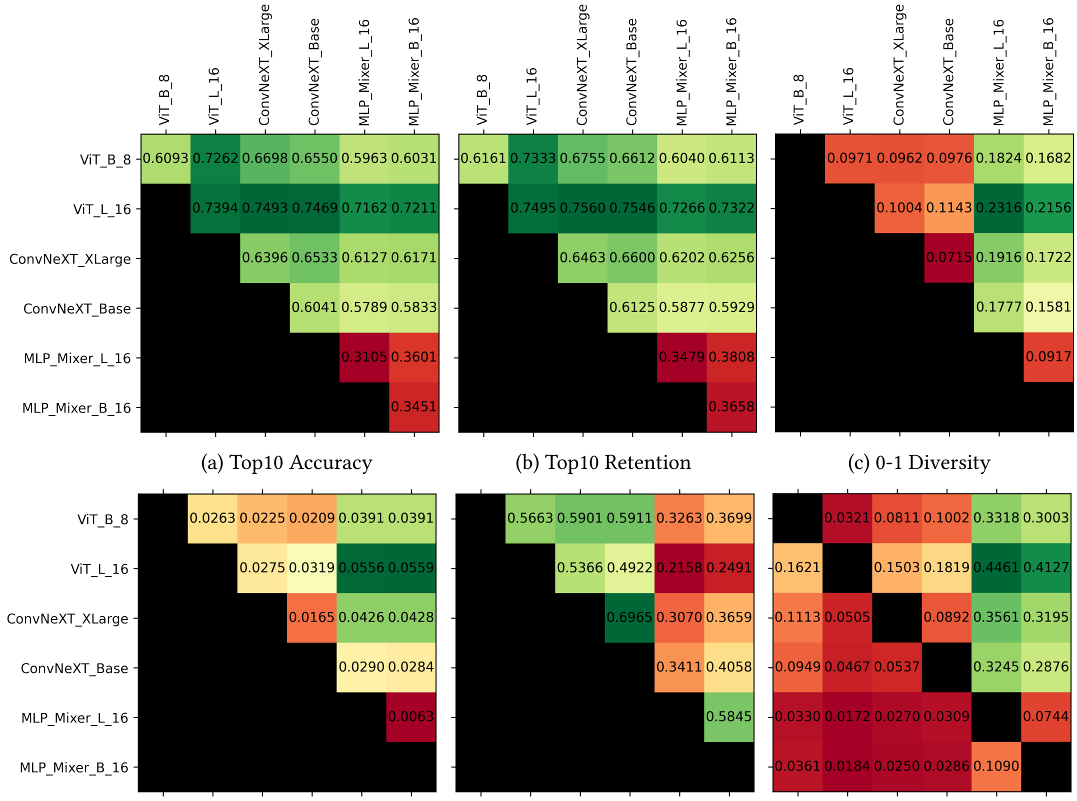
[2] https://storage.googleapis.com/vit_models/augreg/L_16-i21k-300ep-lr_0.001-aug_medium2-wd_0.03-do_0.1-sd_0.1--imagenet2012-steps_20k-lr_0.01-res_224

[3] https://tfhub.dev/sayakpaul/convnext_base_21k_1k_224/1

[4] https://tfhub.dev/sayakpaul/convnext_xlarge_21k_1k_224/1

[5] https://tfhub.dev/sayakpaul/mixer_b16_i21k_classification/1

[6] https://tfhub.dev/sayakpaul/mixer_l16_i21k_classification/1



(a) Top10 Accuracy

(b) Top10 Retention

(c) 0-1 Diversity

(d) CE Diversity

(e) Error consistency

(f) 0-1 Div. Components

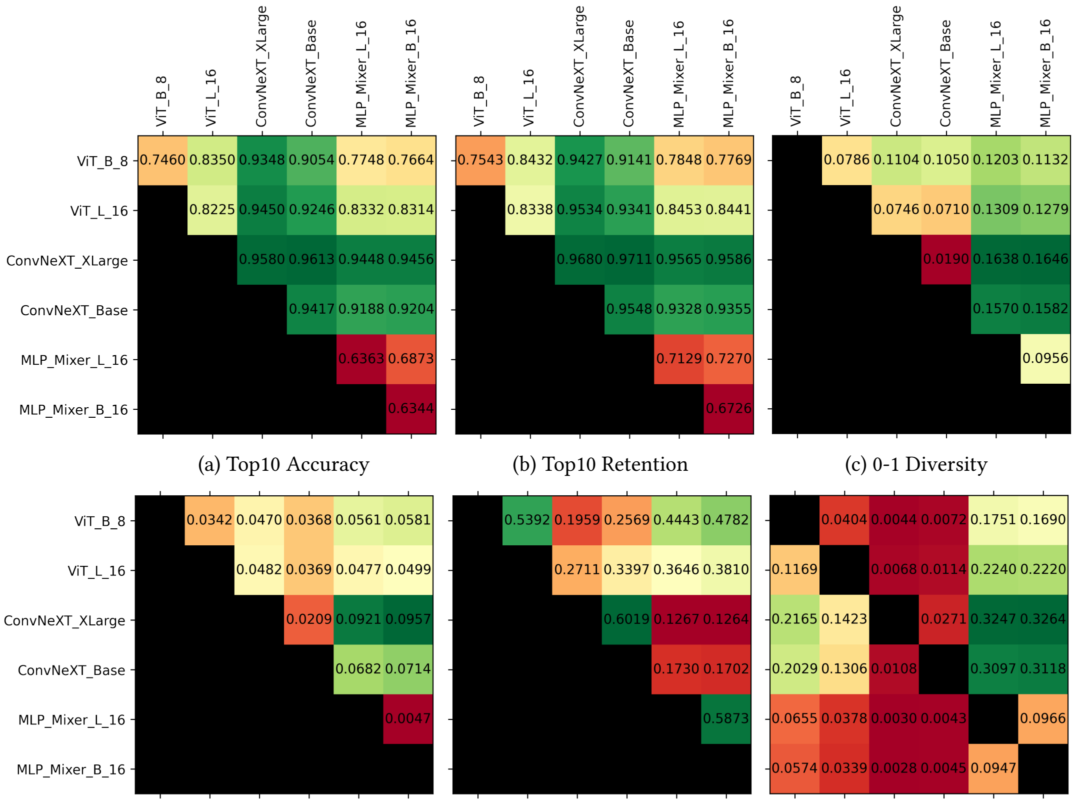
Figure 3: Metrics performance on Gaussian Blur 5 corruption

Results. While values averaged over different corruptions do not play a key role in our comparison, they allow us to comprehend a broader picture of the research subject. In Figure 1, solid lines represent a retention of specific architectures (a mean of two models using this architecture), while dashed ones shows a retention of different ensembles (also averaged over all ensembles of each kind). We clearly see that MLP Mixers perform significantly worse than ViTs and CNNs. However, when MLP Mixers are combined with ViTs or CNNs, the ensembles (brown and grey dashed lines) performance only slightly worse than single ViTs or CNNs models respectively. When we take a look at the top performing ensembles, ViT+CNN ensembles are followed by pure CNN and ViT ensembles. This suggests that mixing different architectures is beneficial for their robustness. The next experiments will support these two hypotheses with more concrete examples and results.

Figure 2 presents accuracy, diversity metrics, and error consistency calculated on original ImageNet data. Each cell represents a metric value scored by an ensemble created by models from corresponding columns and rows. At the diagonal, we have the scores of single models. The last, non-triangular one called *0-1 Diversity components* (0-1 Diversity is calculated by averaging the two values from this plot, located symmetrically to the diagonal) presents a fraction of images that are classified correctly by one model (the one in the row) and incorrectly by the second one (the column model).

Starting with the accuracy plot, we see that the best performing model is ConvNeXt-XLarge, followed by ViT Base, ViT Large, MLP-Mixer Base, and MLP-Mixer Large. In cases of ViTs and MLP-Mixers, smaller models perform better than their bigger counterparts - this might be an artifact of insufficient training. Regarding their ensembles, it is not surprising that the best accuracy is presented by the ensemble of the best performing models (ViT-B and ConvNeXt-XL). We also observe that ensemble performance deteriorates only slightly when one of its components performs significantly (e.g. MLP-Mixer Large) worse than the other.

When we analyze all diversity metrics, we see that MLP-Mixers stand out from other models, especially the Large one. That is caused by much lower accuracy than



(a) Top10 Accuracy

(b) Top10 Retention

(c) 0-1 Diversity

(d) CE Diversity

(e) Error consistency

(f) 0-1 Div. Components

Figure 4: Metrics performance on Contrast 4 corruption

others – when we take a look at 0-1 Diversity Components plot, it shows that Mixers misclassify a significant fraction of images. We also see that the diversity is higher for CNN+ViT ensembles than for intra-architecture ensembles. This allows CNN+ViT ensembles to perform better, e.g. ConvNeXt-B + ViT-B performs better than ConvNeXt-B + ConvNeXt-XL, although ViT-B has lower accuracy.

Another interesting insight: MLP-Mixer L ensembles perform slightly better than all MLP-Mixer B ensembles, while MLP-Mixer L has lower accuracy than MLP-Mixer B by 5 p.p.. One of the possible explanations is that the MLP-Mixer L and B are not that different although they have significantly different accuracy (what results is different 0-1 diversity and Error Consistency values with regard to all other models) – CE diversity between Mixers is as low as between ViTs (which classify very similarly, without 5 p.p. gap). Another evidence that these model behaves similarly is that they have similar CE diversity values with all other models.

To keep the paper concise, we investigate in detail two specific selected corruptions from ImageNet-C: Gaussian Blur at severity 5 and Contrast at severity 4. We have chosen them as they exemplify how different architectures perform on various corruptions. The results for these corruptions are present at figures 3 and 4. Next to the accuracy plots, we also present the retention values.

The Gaussian blur corruption is favored by the Vision Transformer as ViTs perform better than their CNN and MLP-Mixer counterparts. However this time, the best performing model is the ViT-Large instead of Base, what suggests that while its learning process was not sufficient to perform better than the smaller model, but it was sufficient to learn it to perform robustly (ViT-Large is thrice as big as ViT-B).

When we take a look at metrics, the highest (or lowest in case of error consistency) values belong to MLP-Mixers, which perform poorly in comparison to ViTs and CNNs, so we may expect that this diversity comes mostly from their misclassfication. We see it in the 0-1 diversity components, which state that Mixers classify around 30-40% of images incorrectly in contrast to other models. Regarding ViTs and CNNs ensembles, pure CNNs ensembles are less diverse than ensembles of ViTs and CNNs or pure ViT ensembles. If we focus on ConvNeXt-B+XL ensemble and compare it to ConvNeXt-B+ViT-B, we see that it performs slightly better, while ViT-B is less accurate than ConvNeXt-XL. While it's not the most diverse pair between CNNs and ViTs, it's according to all metrics more diverse than the pure CNN ensemble. Other interesting comparison is ViT-L+B vs. ViT-L+ConvNeXt-B: We substitute a Base ViT with a worse performing CNN, what creates a better performing ensemble and more diverse.

Regarding the contrast corruption in figure 4, CNNs dominate performance with only a modest drop in accuracy, while other models perform much worse, especially Mixers. The highest diversity values are related to the worst performing MLP-Mixers. But at the same time, Mixers ensembled with CNNs perform similar to ViT+CNN: worst performing MLP-Mixer Base, which is almost 20 p.p. worse than ViT-L, performs marginally better when ensembled with ConvNeXt-XL - which we find intriguing.

4. Conclusions

While our approach to combine the inherent diversity across models by an ensemble is simple, it manages to capture a synergy that arises from the use of different architectures. The ViT+CNN ensemble has proven to perform not only on average better than other combinations but also regardless of the corruption type, it succeeds to perform satisfactorily.

The diversity metrics and error consistency provide valuable quantitative tools to compare models and quantify the differences in classifications. However, they only allow us to understand the relationships between the models when they are inferred on a specific input. Unfortunately, these metrics may be deceiving in case of two models, where one performs significantly worse than the other. High diversity does not translate to an improved performance of their ensemble which might seem counter-intuitive. The metrics capture how diversely models classify, not the potential of the ensemble of the two models. These two objectives coincide when models perform similarly on the accuracy metric, while a discrepancy in accuracies causes them to misalign. This behavior requires a careful analysis of the metric on every corruption separately.

We list several possible extensions to our work. The first one is an improvement on diversity metrics to metrics assessing the ensemble potential. Secondly, our research was limited to three different architectures. While the results look promising, to fully evaluate and quantify how ensemble aggregates robustness of various models, more experiments should be run, involving more models of different architectures, pretrained on different datasets, and of different sizes. Another direction is to improve the ensemble technique. The potential improvement spans from a weighted ensemble that would average the models e.g. based on their individual performance to a mixture of experts that could predict which model will perform better at some input, and thus precisely leverage the advantages of each particular model to tackle particular corruptions. Such a mixture of experts solution would also be viable in a resource-constrained environment, where running multiple models simultaneously may be unacceptable. The last one is to continue this research for more complex problems like object detection and image segmentation. We need to define diversity metrics for these problems and then investigate the quality of ensembles created using different neural architectures.

References

[1] International Standards Organization, ISO 26262: Road vehicles - functional safety, parts 1 to 11, in: Road Vehicles - Functional Safety, Second Edition, 2018-12.

[2] International Standards Organization, ISO/PAS 21448: Road vehicles - safety of the intended functionality, in: Road Vehicles - Safety of the intended functionality, 2019-01.

[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale (2020).

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

[6] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, A. Dosovitskiy, Mlp-mixer: An all-mlp architecture for vision, 2021. URL: https://proceedings.neurips.cc/paper/2021/file/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Paper.pdf.

[7] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 8 (2018). URL: https://doi.org/10.1002/widm.1249. doi:10.1002/widm.1249.

[8] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems

2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 6402–6413.

[9] R. Geirhos, K. Meding, F. A. Wichmann, Beyond accuracy: quantifying trial-by-trial behaviour of cnns and humans by measuring error consistency, 2020. URL: https://proceedings.neurips.cc/paper/2020/file/9f6992966d4c363ea0162a056cb45fe5-Paper.pdf.

[10] L. A. Ortega, R. Cabañas, A. R. Masegosa, Diversity and generalization in neural network ensembles, 2021. URL: https://arxiv.org/abs/2110.13786. doi:10.48550/ARXIV.2110.13786.

[11] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, A. Veit, Understanding robustness of transformers for image classification (2021). URL: https://arxiv.org/abs/2103.14586. arXiv:2103.14586.

[12] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, Intriguing properties of vision transformers, 2021. arXiv:2105.10497.

[13] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, A. Dosovitskiy, Do vision transformers see like convolutional neural networks?, 2021. arXiv:2108.08810.

[14] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing convolutions to vision transformers (2021). URL: https://arxiv.org/abs/2103.15808. arXiv:2103.15808.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows (2021).

[16] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=HJz6tiCqYm.

[17] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s (2022). URL: https://arxiv.org/abs/2201.03545. arXiv:2201.03545.

[18] D. Zhou, Z. Yu, E. Xie, C. Xiao, A. Anandkumar, J. Feng, J. M. Alvarez, Understanding the robustness in vision transformers, 2022. URL: https://arxiv.org/abs/2204.12451. doi:10.48550/ARXIV.2204.12451.