# It's All About MLPs 🧠

**Gregor Bachmann**

gregor.bachmann@inf.ethz.ch

## Abstract

We showed in a recent work (Bachmann et al., 2023) that even simple multi-layer perceptrons (MLPs) can make very decent image classifiers when subjected to enough compute. This is interesting/-surprising for a few reasons: (1) MLPs are mathematically super simple, they literally consist of just a bunch of matrix multiplications and non-linearities. Yet they can be quite powerful! (2) On the other hand, they have a terrible inductive bias (they are not made for vision), so how to they "see" 👀?! (3) Due to essentially being a sequence of matmuls, MLPs are very efficient at inference time.

There is still a lot to be done on top of our findings; how to MLPs process images compared to CNNs? Can models be improved? How much is the gain in efficiency? Can we restore some of the inductive bias for better performance? Check-out the project ideas below! 😊

## 1 What Is This About?

**First off:** I think the simplest way for you to understand whether you find this project suggestion interesting, is to read the relevant work (Bachmann et al., 2023) first. If you find those results and the raised questions interesting, you should definitely continue reading! If you ask yourself *"Why the hell would you train MLPs for vision 🤨??"* then you're most likely not going to be interested in the rest of this document. No offense taken if that's the case.

**Brief Summary.** Very cool if you made it this far 😊. Let me give a quick summary. We showed in a recent work (Bachmann et al., 2023) that very simple multi-layer perceptrons (MLPs) can reach very strong performance on various image-based tasks if the model and the pre-training dataset are *verrry large*. Why is this interesting?

(1) MLPs are really not built for vision at all!! They are not spatially aware, don't respect shift invariance and share no parameters. It's actually very surprising that they can work at all. This is further evidence that *inductive bias* might become harmful at large scale, as suggested by the Vision Transformer crowd as well (Dosovitskiy et al., 2021; Zhai et al., 2022).

(2) MLPs are super simple, they're literally a bunch of dense matrix multiplications and a few non-linearities. All theoretical works are built upon this model. Can we gain more insights now with these well-performing models?

(3) MLPs are extremely efficient at processing images. While we cannot compete with the SOTA models on ImageNet, we are somewhate competitive on CIFAR10/CIFAR100, especially if inference time is taken into consideration.

## 2 What Can You Do?

### 2.1 Inner Workings of MLPs (Mechanistic Interpret. / Texture Bias)

One of the most fascinating aspects in my opinion is the fact that MLPs are really not built for vision. It is completely unclear to me right now how it makes a decision on a given image. Can we unravel the decision process of an MLP? What features are important for classification? Can we understand what the units at layer 7 are doing? Are MLPs more adversarially robust (Goodfellow et al., 2015)? How do all these things differ from CNNs/ViTs? Check-out the works and blog posts of Anthropic (basically the new *OpenAI*) and of Neel Nanda to get an understanding of how to "reverse-engineer" neural networks (i.e. mechanistic interpretability). While these works all targeted Transformers, I think MLPs are actually even more interesting since they offer a way simpler mathematical structure. Another line of work studies texture versus shape bias in classifiers (Geirhos et al., 2019).

We publish lots of pre-trained networks on our GitHub, so you don't need to train anything from scratch but can just basically play with the models.

### 2.2 Can You Improve the Models?

While pretty good on CIFAR10/CIFAR100 and STL10, we still have quite the gap on ImageNet... 😕 We however observed that simple tricks can make a massive difference; training with strong data augmentation during pre-training really helps performance a lot, adding label smoothing also boosts performance, incorporating test-time augmentation also pushes the accuracy etc. Here you can be quite creative and try out a lot of stuff. Are there smart augmentations tailored towards MLPs that could help? Is there a good regulariser that encourages MLPs to become more local? Can we prune neurons in the MLP after training to enforce locality? This road is of course a bit more high risk, it is very much possible that there is no more good tricks to increase performance. But if you have a good idea, give it a try!

Again the pre-trained models are published, so ideally one would look for tricks during fine-tuning or even test-time. For smaller MLPs you might even be able to pre-train them on ImageNet21, but I wouldn't bet on that (or you at least cannot iterate too much with your resources 💵).

### 2.3 Inference Time vs Performance

Something that we did not really crystallise out in our work is how the inference time of MLPs compares to other models (whether pre-trained or not). One of the critisism of our work is that MLPs are all nice and cool, but simply training a CNN on the task (without any pre-training) can often be on-par or even beat our heavily pre-trained networks (check this 🐦 thread, you might need to login to see it).

**However:** How does inference time compare against these models? Ideally, having a plot of inference time versus performance for lots of networks would be amazing. Again you could largely rely on pre-trained networks for that (e.g. check-out the timm library). For now the study would need to be limited to CIFAR10/CIFAR100 since we're only really competitive there. Especially without pre-training, I'm pretty certain that MLPs should dominate a lot of models, especially if the time to transfer the data is removed.

### 2.4 "Softer" MLPs

Another question is how much can performance be improved again by bringing back some of the things an MLP lacks? One idea that is very popular these days is the notion of "patchifying" an image (Dosovitskiy et al., 2021), as done in the Vision Transformer architecture. This brings back locality as well as parameter-sharing, at least for the first layer. There's a ton of work on such MLP modifications, e.g. Tolstikhin et al. (2021); Liu et al. (2021) to name but a few, so you would need to do a proper literature review first.

## 3 What Will I do?

Since I'm very interested in the topic, I would be open to meet roughly every two/three weeks to discuss ideas and progress/questions. If the outcome of a project is interesting, I would definitely be available to try and push for a publication at a conference.

## References


Bachmann, G., Anagnostidis, S., and Hofmann, T. (2023). Scaling mlps: A tale of inductive bias.

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

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019). Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples.

Liu, H., Dai, Z., So, D. R., and Le, Q. V. (2021). Pay attention to mlps.

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., and Dosovitskiy, A. (2021). Mlp-mixer: An all-mlp architecture for vision.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). Scaling vision transformers.