MSc. Thesis - Deep Restoration
Paper Summary

Frederik Harder

August 13, 2017

1 Introduction & Motivation

Hierarchical feature learning has been a driving force behind recent advances in machine learning. Since the success of the model by Krizhevsky et al. [6], now commonly referred to as AlexNet, at the 2012 ImageNet visual recognition challenge, deep neural networks have begun dominating computer vision and several other areas in machine learning, such as reinforcement learning and speech recognition, with more applications in active exploration. Along with this growing popularity of deep learning methods, there has been an increasing interest in visualizing these learned representations, and elucidating, what information these deep models base their decisions on.

Especially in the field of computer vision, where learned visual representations can be easy to interpret, a number of different approaches have been published on this matter and will be discussed in some detail below. A common theme is that, in order to visualize representations at a certain layer in the network, (sometimes a subset of) the network's activations in that layer are, in some way or another, related back to the image space at the network input. This is not a trivial task, as the processing steps that transform the input up to that layer are highly non-linear and generally non-invertible. Going through a deep neural network, information from the input is compressed and details that are deemed irrelevant can be omitted. As a result, the task of reconstructing an input from the activation of a single layer is usually under-constrained and there is a class of inputs, which cause the same or nearly identical activations in the layer.

This project builds on work by Mahendran & Vedaldi [7, 8], who constrain the reconstruction problem with image priors, and aims to improve upon their results by employing more expressive priors both in image space and over the feature map activations of the network in question.

In the remainder of this section, the challenges and limitations of two major approaches to network inversion are discussed, in order to give an intuition of the task. Following that, section 2 provides an overview of the related work in this field and introduces some of the references that this project draws from. In section 3, the models introduces the used models in detail, followed by an

overview of the experimental results up to this point in section 4. Section 5 specifies the future goals of the project and outlines the steps that still need to be completed, as well as pointing out some expected challenges and unresolved questions.

1.1 Theoretical Considerations

1.1.1 Limitations of inverse networks

Claim: Convolutions are not sufficient for inverting convolutions.

Argument sketch: Given a feature map as f_{j+1}, where $\boldsymbol{W_j}$ is toeplitz and f_j is a variable with Gaussian log prior given as $\frac{1}{2} f_j^T \boldsymbol{\Sigma_{f_j}^{-1}} f_j + const$, the optimization over f_j is as follows:

$$\min_{f_j} \frac{1}{2}\lambda ||f_{j+1} - \boldsymbol{W_j} f_j||^2 + \frac{1}{2} f_j^T \boldsymbol{\Sigma_{f_j}^{-1}} f_j + const \tag{1}$$

which resolves to

$$\hat{f_j} = \left(\frac{1}{\lambda} \boldsymbol{\Sigma_{f_j}^{-1}} + \boldsymbol{W_j^T W_j} \right)^{-1} \boldsymbol{W_j^T} f_{j+1} \tag{2}$$

It can be shown, that $\left(\frac{1}{\lambda} \boldsymbol{\Sigma_{f_j}^{-1}} + \boldsymbol{W_j^T W_j}\right)^{-1} \boldsymbol{W_j^T}$ is not toeplitz (under what conditions?, why?), and therefore the optimal inverse operation is not a convolution.

Assuming no prior and going for the maximum likelihood estimate can instead be approximated with the Moore-Penrose pseudoinverse $\hat{f_j} = \boldsymbol{W_j^+} f_{j+1}$. $\boldsymbol{W_j^+}$ is also not toeplitz (under what conditions?). Therefore using convolutions to approximate $\boldsymbol{W_j^+}$ limits reconstruction performance by design.

1.1.2 Limitations of optimization based approaches

- non-convex optimization
- not clear what to optimize (MSE is not necessarily the best measure)
- underconstrained. priors are necessary. what priors to use is an open problem

2 Related Work

- in detail:

  Dosovitskiy & Brox: Inverting networks [2, 1]

  Mahendran & Vedaldi: [7, 8]

  Yosinsky et al.: using classifiers for generative modeling [9]

- Gatis et al.: Style transfer using different feature map measures [3]

- Zeiler & Fergus: generally de-convolutional approaches should maybe be mentioned. What to they visualize, what does this approach? [13]

  then also Kindermans et al. (patternnet, patternrlrp) [14]

- also others (list in progress)

3 Methods

3.1 Image classifiers

Basic section on deep image classifiers coming up!

3.1.1 AlexNet

Overview of the AlexNet architecture coming up!

3.1.2 Vgg16

Overview of the Vgg16 architecture coming up!

3.2 Modular inverting networks

Following the work by Dosovitskiy & Brox [2], a first baseline has been set using inverting networks. Here, a modular approach is used, where each module is trained to in invert a small part of the network, i.e. either a convolution operation followed by a nonlinearity, a pooling operation, or, in case of AlexNet, a local response normalization operation. These modules can then be combined and trained synchronously, where each module output computes a reconstruction loss and is fed as input to the next module, as shown in figure 1. Different modules have been tested and a transpose convolution, followed by a ReLU and then a convolution has performed best. Future results will follow Odena et al. [10] and use upscaling and convolution instead of a regular transpose convolution in order to prevent artifacts.

3.3 Optimization based methods

The central approach to network visualization is based on work by Mahendran & Vedaldi [7, 8]. Their method, illustrated in figure 2, optimizes a *pre-image* variable to match one of its feature map representations with that of an image, while constraining the *pre-image* additionally by a hand-crafted natural image prior. These results have been reproduced, with the goal of improving upon them by using learned priors instead of handcrafted ones. In addition to a natural image prior, priors trained on the feature maps will be used to constrain the problem further and hopefully lead to better reconstructions as a result. This setup is shown in figure 3.

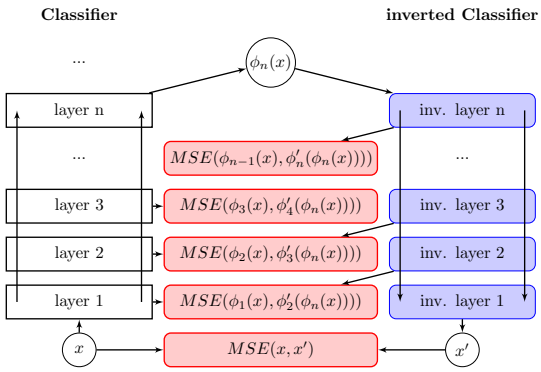


Figure 1: Modular inverting network architecture

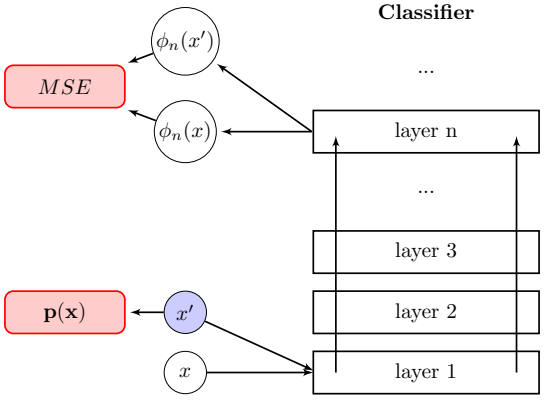


Figure 2: Model by Mahendran & Vedaldi [7, 8]

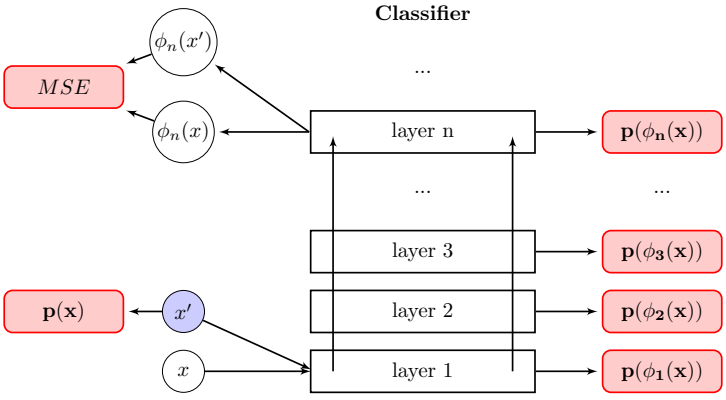


Figure 3: Extension of Mahendran & Vedaldi with feature map priors

3.4 Priors

3.4.1 Mahendran & Vedaldi's work and handcrafted image priors

In [7], the authors employ an optimization method, using a normalized mean squared error loss on the feature maps and two image priors.

The normalized mean squared error takes the form

$$||\Phi(\sigma\mathbf{x}) - \Phi_0||_2^2 / ||\Phi_0||_2^2 \tag{3}$$

where Φ_0 is the feature map representation of the target image and $\Phi(\sigma\mathbf{x})$ is the representation of the re-scaled *pre-image* $\mathbf{x}$. The scaling factor σ = 2.7098e+4 is chosen as the average Euclidean norm of natural images in the training set to the effect that $\mathbf{x}$ has roughly unitary Euclidean norm. Dividing the mean squared error by the squared Euclidean norm of the target image confines errors near an optimum to the interval $[0, 1]$. This predictability can be convenient when considering the scaling of the priors and as a result, the same normed mean squared error loss has been adopted for the experiments using learned priors, discussed below.

The authors constrain the optimization of the feature level error with two hand-designed priors. The first one is an α-norm prior, defined as

$$\mathcal{R}_\alpha(\mathbf{x}) = ||\mathbf{x} - \text{mean}(\mathbf{x})||_\alpha^\alpha \tag{4}$$

It is meant to limit the overall range of pixel values. Results are reported with $\alpha = 6$.

In addition, a total variation prior is used, which penalizes the difference between neighboring pixels and thus encourages smoothness in the pre-image. It takes the following form, with $\beta = 2$ being chosen in most experiments:

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{\beta}{2}} \tag{5}$$

In their follow-up publication [8], the authors maintain the MSE loss and total variation prior, but refine the range prior to

$$N_\alpha(\mathbf{x}) = \sum_{i,j} \left(\sum_k \mathbf{x}_{ijk}^2 \right)^{\frac{\alpha}{2}} \tag{6}$$

where i, j are pixel positions and k denotes the channel. The authors argue that this version resolves a problem of color bias introduced by the alpha norm prior, because unlike the previous one, it is isotropic in RGB space. They couple this soft constraint with a hard one, which enforces $\sqrt{\sum_k \mathbf{x}_{ijk}^2} \leq B_+$ for each image position (i, j). They also develop the weighting of the different parameters further and introduce an additional regularization method called jittering, which makes small shifts to the pre-image on each iteration and produces crisper images overall. Jittering may be considered in combination with learned priors at a future point and a detailed treatment of the method will be provided, but is omitted for now.

3.4.2 ICA based sparse coding priors

In his paper on the method of score matching [4], the author introduces a model for image patches $\mathbf{x}_{(k)}$ based on over-complete independent component analysis (ICA). The log probability of is defined, using the function $G(s) = -\log\cosh(s)$ and an unspecified normalizing constant Z.

$$p(\mathbf{x}_{(k)}) = \exp(Z(\mathbf{w_1}, ..., \mathbf{w_N}, \alpha_1, ..., \alpha_N)) \prod_{i=1}^{N} \exp\left(G(\mathbf{w}_i^T \mathbf{x}_{(k)})\right)^{\alpha_i} \tag{7}$$

Following the derivations of [4], this prior for image patches can be trained via score matching, using mini-batch gradient descent, by minimizing the sampled loss function $\tilde{J}$ over a training set of image patches. For this purpose the first two derivatives of G are computed as $g(s) = -\tanh(s)$ and $g'(s) = -\cosh(s)^{-2}$.

$$\tilde{J} = \sum_{k=1}^{m} \alpha_k \frac{1}{T} \sum_{t=1}^{T} g'(\mathbf{w}_k^T \mathbf{x}_{(t)}) + \frac{1}{2} \sum_{j,k=1}^{m} \alpha_j \alpha_k \mathbf{w}_j^T \mathbf{w}_k \frac{1}{T} \sum_{t=1}^{T} g(\mathbf{w}_k^T \mathbf{x}_{(t)}) g(\mathbf{w}_j^T \mathbf{x}_{(t)}) \tag{8}$$

The trained patch prior can then be turned into an image prior, by the same consideration, that is used in [11]. Viewing the image as a Markov random field with a node for each pixel position and a patch centered at each position, nodes are connected, iff they share a patch. As the patch prior in equation (7) is strictly positive and the patches constitute the maximal cliques of the Markov random field, the patch prior can serve as a clique potential and the probability of the field is proportional to the product of its clique potentials (source).

$$\begin{aligned} p(\mathbf{x}) &\propto \prod_{k=1}^{K} \exp(Z(\mathbf{w_1}, ..., \mathbf{w_N}, \alpha_1, ..., \alpha_N)) \prod_{i=1}^{N} \exp\left(G(\mathbf{w}_i^T \mathbf{x}_{(k)})\right)^{\alpha_i} \\ &\propto \prod_{k=1}^{K} \prod_{i=1}^{N} \exp\left(G(\mathbf{w}_i^T \mathbf{x}_{(k)})\right)^{\alpha_i} \\ \log p(\mathbf{x}) &\propto \sum_{k=1}^{K} \sum_{i=1}^{N} \alpha_i G(\mathbf{w}_i^T \mathbf{x}_{(k)}) \end{aligned} \tag{9}$$

3.4.3 Fields of experts priors

The field of experts prior developed by Roth and Black [11] is a patch based prior of the form

$$p(\mathbf{x}) = \frac{1}{Z(\mathbf{\Theta})} \prod_{k}^{K} \prod_{i=1}^{N} \phi_i(\mathbf{w}_i^T \mathbf{x}_{(k)}; \alpha_i) \tag{10}$$

for a set of K equally sized patches $\mathbf{x}_{(k)}$, where ϕ_i are Student's t distributions $\phi_i(\mathbf{w}_i^T\mathbf{x}_{(k)};\alpha_i) = (1+\frac{1}{2}(\mathbf{w}_i^T\mathbf{x}_{(k)})^2)^{-\alpha_i}$, parameterized by $\mathbf{w}$ and α and $Z(\mathbf{\Theta})$ is a normalizing factor[1].

Looking at the prior for an individual patch $\mathbf{x}_{(k)}$, the log likelihood can be derived up to an additive constant. This makes it possible to define the prior identically to the over-complete independent component analysis discussed above, excepting the different definition for function G.

$$\begin{aligned}
p(\mathbf{x}_{(k)}) =& \frac{1}{Z(\mathbf{\Theta})}\prod_{i=1}^{N}\phi_i(\mathbf{w}_i^T\mathbf{x}_{(k)};\alpha_i) \\
=& \frac{1}{Z(\mathbf{\Theta})}\prod_{i=1}^{N}\left(1+\frac{1}{2}(\mathbf{w}_i^T\mathbf{x}_{(k)})^2\right)^{-\alpha_i} \\
\log p(\mathbf{x}_{(k)}) =& \sum_{i=1}^{N} -\alpha_i \log\left(1+\frac{1}{2}(\mathbf{w}_i^T\mathbf{x}_{(k)})^2\right) - \log Z(\mathbf{\Theta}) \\
\propto& \sum_{i=1}^{N} -\alpha_i \log\left(1+\frac{1}{2}(\mathbf{w}_i^T\mathbf{x}_{(k)})^2\right) \\
\propto& \sum_{i=1}^{N} \alpha_i G(\mathbf{w}_i^T\mathbf{x}_{(k)})
\end{aligned} \tag{11}$$

Computing the derivatives of G then allows for the training of parameters $\mathbf{w}$ and α via score matching.

$$\begin{aligned}
G(s) =& -\log(1+\frac{1}{2}s^2) \\
g(s) =& -\frac{2s}{s^2+2} \\
g'(s) =& \frac{2(s^2-2)}{(s^2+2)^2}
\end{aligned} \tag{12}$$

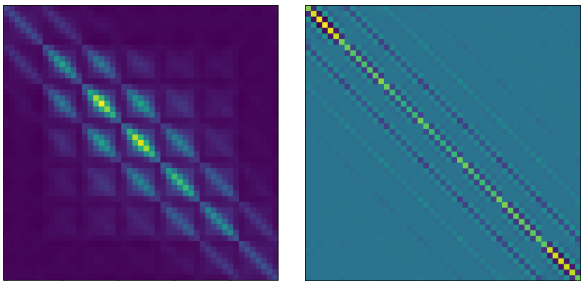
Directly following Hyvarinen 2003, equations (11) and (12), this results in the same sampled loss function $\tilde{J}$, shown in equation (8), which finds the optimal $\mathbf{w}$ and α for a set of natural image patches, when minimized.

4 Results (clear split from methods & discussion in progress)

4.1 Feature map statistics

In an initial investigation, three feature map characteristics were looked at. The average covariance and its inverse, shown in figure 4, substantiate an assumption

[1] In [11], $\mathbf{J}$ is used in place of $\mathbf{w}$, but the notation is changed here, to match [4]



(a) Average covariance (b) Inverse average covariance

Figure 4: both figures are based on flattened feature maps in the fifth 7 by 7 pooling layer of VGG16 over 200 images

Figure 5: on the left: original image, then: inverse networks reconstructions based on VGG16 feature maps from pooling layers 1, 2, and 3 (will need more detail & deeper layers)

of spatial stationarity (ground in math) in the feature map activation. Border effects are visible, especially in the inverse matrix, but leaving these aside, the inverse is close to toeplitz, which is expected to become more accurate for larger sets of images. (and this matters because math). This finding serves to motivate convolutional and patch-based priors. Gram matrices and sparsity statistics of the feature maps were also evaluated, and may be used at a later stage in evaluating the trained feature map priors.

4.2 Modular inverting networks

The inverting networks were trained on 48000 images of the 2012 ImageNet validation set, with the remaining 2000 set aside for testing. All training runs used the Adam optimizer [5] with a learning rate of 3e−4. Models were trained both on the AlexNet and the VGG16 classifier. Figure 5 shows reconstructions

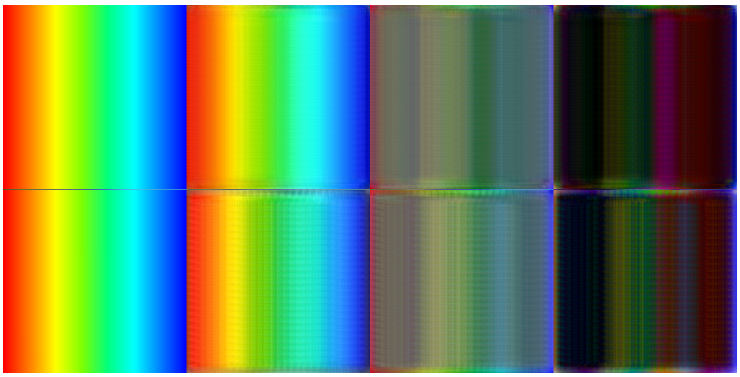


Figure 6: top row: modules trained in parallel, bottom row: stacked modules. left to right: original image, reconstruction from AlexNet, two pixel error measures for visualization

from the first three pooling layers of VGG16, illustrating, how image details fail to be recovered from deeper layers.

Besides setting a baseline in terms of reconstruction quality, this approach is also intended to explore two questions. The first minor insight concerns the performance of stacked modules that were trained individually based on feature map pairs as compared to models, which are trained together. As figure 6 shows, simultaneous training yields better results.

The second and major incentive is to motivate the use of constraints in the intermediate feature map layers. For this purpose, the modules have been trained end to end, leaving out all intermediate losses. A comparison with stacked modules is shown in figure 7. This result shows, that constraining the intermediate representations in the model to match the feature maps of the classifier to be inverted can lead to overall better reconstructions. It is therefore reasonable to assume that the same will hold in the following approach using optimization based methods.

Because differences between image and reconstruction can be quite small, two difference measures have been added. The first is a color/channel accurate measure, showing the difference between image and reconstruction, renormalized to the interval [0,1]. So if, for example, the reconstructed image has less blue in it, the measure will be blue. Areas without error (or equal errors across channels) are gray. The second difference measure is the renormalized absolute difference, which highlights areas with errors more clearly. Areas without error are black. Given renormalization, pixel intensity only informs about relative error within the image, not absolute error compared to other images. Without normalization, the errors were hardly visible. The figures are ordered left to right as: image, reconstruction, color measure, absolute measure.

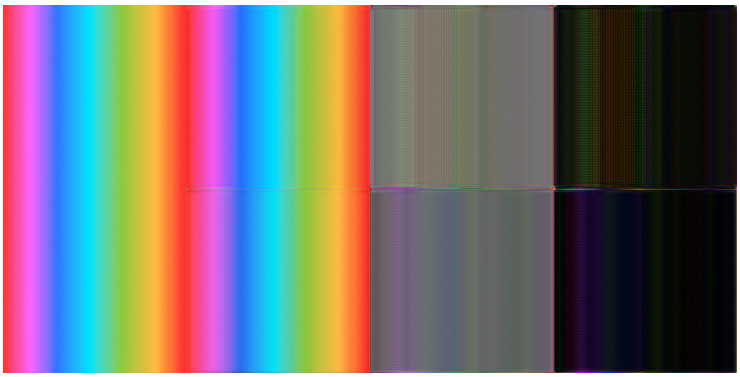


Figure 7: Top row: end to end trained model, bottom row: stacked modules. left to right: original image, reconstruction from the first VGG pooling layer, two pixel error measures for visualization

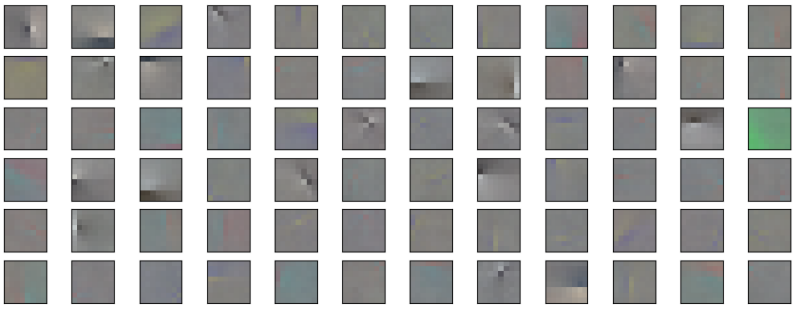
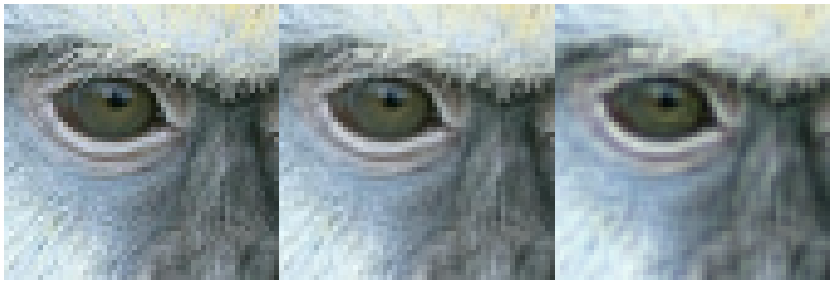


Figure 8: Subset of the 512 filters from an over-complete ICA prior

4.3 Optimization based reconstruction

An ICA-based image prior with 512 filters, shown in figure 8, has been used for performing initial reconstructions from the first and fourth layer of AlexNet. The results are displayed in figures 9 and 10. Both are trained using the Adam optimizer. The first layer reconstruction used a stable learning rate of 0.004 over 5000 iterations, while for the fourth layer, a decreasing learning rate was used over 10000 iterations (0.01 at the beginning, 4e−3 at iteration 1000, 1e−3 at 1200, 4e−4 at 3000, 1e−4 at 6000 and 4e−5 at 9000). The same normed mean squared error loss as in [7] is used and relative to it, the ICA prior is weighted at 1e−4 in the first layer and at 1e−3 in the fourth. A comparison to the results of [8], as well as an exhaustive treatment of all network layers, is in progress.

(a) 224x224 pixel image



(b) 56x56 pixel detailed view

Figure 9: Comparison of reconstructions from the first layer (Convolution + ReLU) of AlexNet. Left to right: original image, ICA image prior, reproduction of [7]

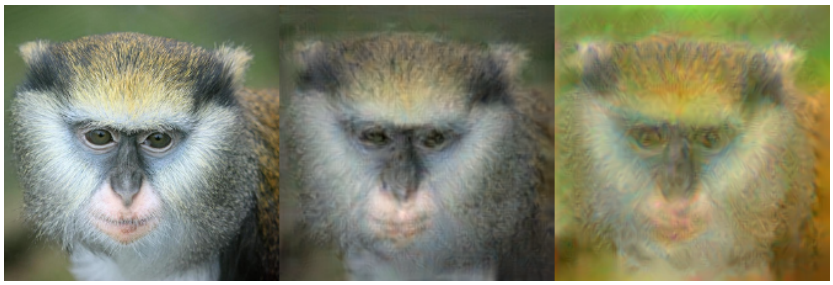


Figure 10: Comparison of reconstructions from the fourth convolutional layer of AlexNet in detail. Left to right: original image, ICA image prior, reproduction of [7]

5 Next steps and research goals

5.1 Immediate next steps:

The training of image and feature map priors based on over-complete ICA is set up and will be completed shortly. There is one potentially unresolved problem with regards to the patch sizes on feature maps. Given that the number of channels grows quickly, the patches should be chosen as small as possible. The receptive field size in the previous layer seems like a good first guess. Should the number of patch features grow too large regardless, separate priors for subsets of channels may have to be considered. This, however, would make independence assumptions that are almost certainly wrong and can hopefully be avoided.

When the priors are trained, a prime concern will be the adequate relative weighting in optimization based reconstruction, as each one of them represents an un-normalized negative log probability and as a result can vary greatly in scale. If a principled approach is not found, parameter exploration schemes will have to be designed.

Experiments will then investigate the quality of reconstructions following [8] with learned priors perhaps augmenting, but most likely replacing the hand-crafted ones. If these efforts prove fruitful, related applications like activation maximization and caricaturization [8] can be explored in a quick follow-up. If, instead, the results point to limitations of the prior model, alternative models will need to be explored immediately. Here I propose two options: The field of experts model is closely related to the model using ICA based filters, but uses Student-t experts instead of a logistically distributed components and is well established as a model. The second option comes from [9], which uses denoising auto-encoders to approximate the gradient of natural image priors. This should also work on feature maps and is distinct, in that it is not explicitly a patch-based approach. On the other hand, training times may become more of a concern in that case.

5.2 Goals:

The minimal empirical project goal at this point is the reproduction of the experiments conducted by [8], with the use of learned image and feature map priors.

A variety of new experiments could be enabled with a good sampling technique. As far as I am aware, unfortunately, neither the patch-based methods, nor the DAE prior lend themselves very well to this task. The highest goal, of sampling images jointly from weighted priors at different feature maps poses an additional problem, which I can currently offer no good solution to. Of course the modes of these distributions can be explored through optimizing pre-images. Generative models, for instance Variational Auto-Encoders, can be trained on individual feature maps and serve as targets for optimization based approaches or as inputs to inverting networks. This could show, how the distribution of feature map activations corresponds to natural images.

The fact that the related literature up to this point has mostly been concerned with classical convolutional networks such as AlexNet or Vgg16, while the state of the art has moved on. It may therefore be fruitful to apply the methods developed here to inception-based or ResNet models for comparison.

A wider review of related work may also provide some additional setups that can be improved with priors. (A day or two of reading may be necessary though)

5.3 Some thoughts on evaluating priors

- measures:

  reconstruction MSE in image space compared with M&V is a bit of a cheap shot, because that's not really something they set out to improve (they do report something which they call reconstruction error, but which, from the numbers and everything has to be the representation MSE). also, these scores will be abysmal regarding deeper layers, no matter which approach is taken.

  for Dosovitskiy, MSE is fair game, but might be harder, because the explicitly optimize for it. So qualitatively nicer reconstructions might still lose that contest.

  Structural similarity index might be an option, because it's at least translation invariant (right?). It is also perception focused, which might not be all bad, but is not necessarily all that interesting. As it is an accumulated measure, maybe some of its components are useful. (details follow)

- expectations:

  the results by M&V on lower layer reconstructions look really good. not sure if learned priors can actually beat that. maybe in conjunction with some of the M&V parts.

  on lower layers, feature map priors could become increasingly important and these should likely show the first improvements

- very rough test setups:

  test each prior on natural image feature maps to get an idea of the spread of values

  all priors can be tested individually, e.g. with the M&V image priors in place. maybe some layers have more of an impact than others? (deep vs. shallow)

  To get to the point where all priors are used in the model, maybe it's a good idea to add them iteratively. The question then is, do you start deep or shallow? maybe just start with whatever worked best previously.

References

[1] Dosovitskiy, A., & Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In Advances in Neural Information Processing Systems (pp. 658-666).

[2] Dosovitskiy, A., & Brox, T. (2016). Inverting visual representations with convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4829-4837).

[3] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[4] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research, 6(Apr), 695-709.

[5] Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[7] Mahendran, A., & Vedaldi, A. (2015). Understanding deep image representations by inverting them. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5188-5196).

[8] Mahendran, A., & Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. International Journal of Computer Vision, 120(3), 233-255.

[9] Nguyen, A., Yosinski, J., Bengio, Y., Dosovitskiy, A., & Clune, J. (2016). Plug & play generative networks: Conditional iterative generation of images in latent space. arXiv preprint arXiv:1612.00005.

[10] Odena, A. Dumoulin, V. & Olah, C. (2016) Deconvolution and Checkerboard Artifacts. Distill. http://doi.org/10.23915/distil

[11] Roth, S., & Black, M. J. (2005, June). Fields of experts: A framework for learning image priors. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 860-867). IEEE.

[12] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[13] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

[14] Kindermans, P. J., Schütt, K. T., Alber, M., Müller, K. R., & Dähne, S. (2017). PatternNet and PatternLRP–Improving the interpretability of neural networks. arXiv preprint arXiv:1705.05598.