Fakultät für Mathematik und Informatik
Universität Heidelberg

**Ist künstliche Intelligenz gefährlich? (Seminar)**

# Applications of Neural Networks

*Part II[1] of the Topic "Neural Networks"*

Master Seminar Transcript
submitted by

**Nicolas Roth**
enrolled in Department of Physics and Astronomy

under the supervision of
**PD Dr. Ullrich Köthe**

[1] Part I on "Fundamentals of Neural Networks" by Nasim Rahaman

## Abstract

Almost all recent algorithms in the field of machine learning or artificial intelligence utilize neural networks. This work accompanies an introductory talk on their applications, predominatly in the field of computer vision. Some presented algorithms extract high-quality information from visual input, others are more playful and show behavior that could be interpreted as first steps towards creativity or imagination of computer systems.

## Contents



[2]Functional principles were covered by the talk of Nasim Rahaman. This section is a short compendium to provide context for later applications.

# 1 Introduction

*If we want to evaluate the potential risks of Artificial Intelligence (AI) on a scientific basis, the obvious starting point is to understand how it is achieved.*

## 1.1 Symbolic vs. "real" AI

In its early days, AI used to be logic driven and therefore basically a huge collection of rules. The Oxford Dictionary defines *Intelligence* as *"the ability to learn, understand and think in a logical way about things"*. Logic driven, symbolic AI improve by adding more rules to its decision process. Systems could react on inputs but never abstract from them, not to speak of actually making sense out of it. What we want is a system that is able to improve intrinsically by getting more and more input. The idea behind Machine Learning (ML) is learning a model from data - and the most powerful tool to do so turned out to be via Artificial Neural Networks (ANNs). They tackle problems by considering training examples and do not necessarily require task-specific programming. Depending weather the input data is labeled or not, we distinguish between supervised and unsupervised learning. Whereas the latter mostly aims to group (or "cluster") similar inputs together, supervised algorithms predict labels which are classes for classification tasks or continuous numbers for regression.

## 1.2 Fundamentals[3]

The building block of ANNs, a perceptron, is basically a computational graph: it multiplies every input with some number (weight) and adds all of them together with an additional number (bias). We call that a "neuron", which can be stacked together to layers . To be not restricted to linear functions, we need to apply a nonlinearity to the output of every neuron. The inputs of the network should be expressible in numbers (e.g. data points or gray values) and we need as much output neurons as classes (for classification tasks) or variables (for regression tasks). How good the resulting predictions are, is measured by the loss function. In supervised learning, it is a metric to determine how far away a prediction of the network is from the real value. To improve the results the network-parameters (weights and biases) are altered in such a way, that the loss decreases. Therefore, we want to follow the gradient of the loss in parameter space. To know, which parameter accounted for how much of the final prediction, we propagate backwards through the network by applying the chain rule.

The more layers such networks have, i.e. the deeper they become, the more flexibility and expressional power they get. The downside is that the number of parameters increases rapidly. Especially for large input data as images, it has proven to be way more efficient to restrict the input of a neuron to a smaller neighborhood of the previous layer. Also, weights are shared between perceptrons of one layer, meaning that the same function is learned for all neighborhoods. A layer of perceptrons can then be understood

[3]Functional principles were covered by the talk of Nasim Rahaman. This section is a short compendium to provide context for later applications.

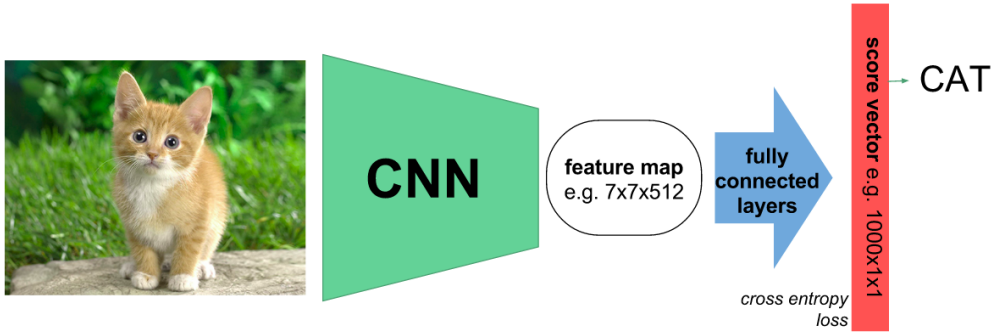


Figure 1: Schematic view of image classification using a CNN

as a filter that is convolved with the input which is why these types of ANNs are called Convolutional Neural Networks (CNNs).

# 2 Applications

*Since most of modern AI is based on ANNs, other talks within the scope of this seminar can mostly be considered to also fall under the title "Applications of Neural Networks". Due to future elaborations on speech recognition, applications of reinforcement learning etc., the focus of this work will lie on computer vision and related tasks.*

## 2.1 Image Classification

Classification of images is historically the task neural networks gained popularity for. They were successfully used already in the previous century for recognizing text and handwritten digits by [LeCun et al., 1998]. They almost sank into obscurity until [Krizhevsky et al., 2012] won the ILSVRC-2012 competition with a deep CNN (five convolutional layers and three fully-connected layers,). Due to a novel method called *dropout*, they prevented their model from overfitting, made it more robust and therefore the new state of the art. Since their publication, every winning team of ILSVRC (for details on the challenge see [Russakovsky et al., 2014]) used CNNs.

The functional principle is shown in Fig. 1, which is still the basis to recent approaches as [Huang et al., 2016]. Processing the image through a CNN allows for a efficient representation, a feature map. From there, scores for a given number of classes are calculated through a fully connected classifier. Experience showed that generally best results are achieved if the whole network is trained jointly.

## 2.2 Localization

If not only the dominant subject in a picture is asked for but also where it is located, we speak of localization. This combines classification with regression but the functionality

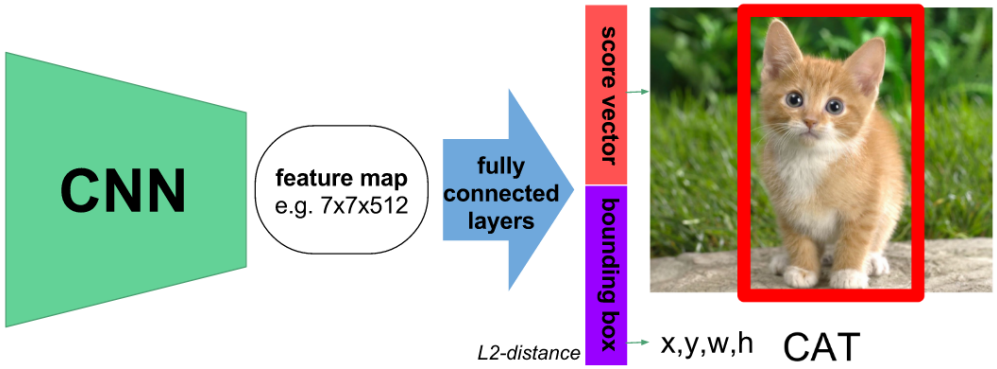


Figure 2: Schematic view of a localization task

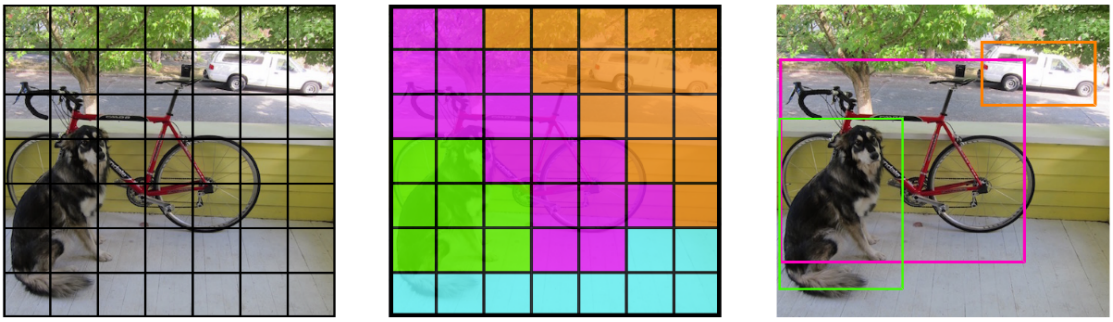

Figure 3: Detection with the YOLO system (colors correspond to classes: green → dog, pink → *bike*, orange → *car*, cyan → *dining table*)

stays unchanged (see Fig. 2). The network predicts now not just scores but four additional values that determine a bounding box. These can be learned by applying L2-loss (as long as proper training data is available). One way of computing bounding boxes more efficiently is presented by [Sermanet et al., 2013].

## 2.3 Detection

Locating objects gets a great deal harder if there is not always one target per image but a variable number. One of the state-of-the-art detectors by [Redmon and Farhadi, 2016], Fig. 3, utilizes again a CNN to get a feature map representation. The features are assigned to discrete locations in the image with one class and some number of bounding boxes. These boxes have a location, a class (given by the feature) and some confidence which also is learned. Finally, only bounding boxes with the highest confidence are displayed. These systems are fast enough to run in real time.

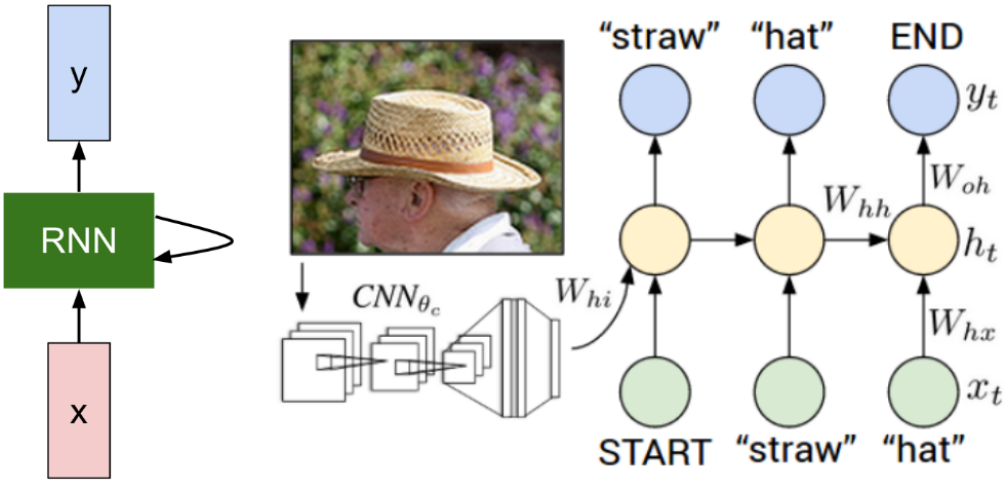


Figure 4: Utilizing RNNs to generate image captions

## 2.4 Generating Text

Detecting objects in pictures is only the first step to actually *understand* the context of a scene. There exists a variety of approaches to generate text to visual input, like [Karpathy and Fei-Fei, 2014], [Johnson et al., 2015] or [Lu et al., 2016]. Instead of just detect objects from a fixed number of classes, a sequence of words from a large vocabulary is generated. A meaningful and grammatically correct sentence can only be produced if every new word depends on the already generated ones. A suitable architecture for such tasks are Recurrent Neural Networks (RNNs). As shown in Fig. 4 they have loops and get their own output as input in the next time step. Unrolled in time this is basically a feedforward ANN again which can - in principal - be trained jointly. In reality, this looping is prone to cause exponential vanishing or exploding gradients of the loss function. A workaround to this problem was proposed already by [Hochreiter and Schmidhuber, 1997] which is with slight variations still state of the art. This idea generalizes to basically all natural language models. For example, translators or question answering problems like in [Kumar et al., 2016] have vectors of words as inputs instead of arrays of gray values. How they process it and generate outputs follows the same principles. Applications like these were covered in the talks of Enes Witwit and Maximilian Müller-Eberstein.

## 2.5 Game Engines

AI surpassing human performance in the board game Go and Atari 2600 games gained a lot of media attention. CNNs are used for game engines to derive efficient representations of the environment as shown by [Mnih et al., 2013]. From that, an agent can derive a policy through reinforcement learning. How this works in detail was elaborated in the talks held by Patrick Dammann and Florian Fallenbüchel.

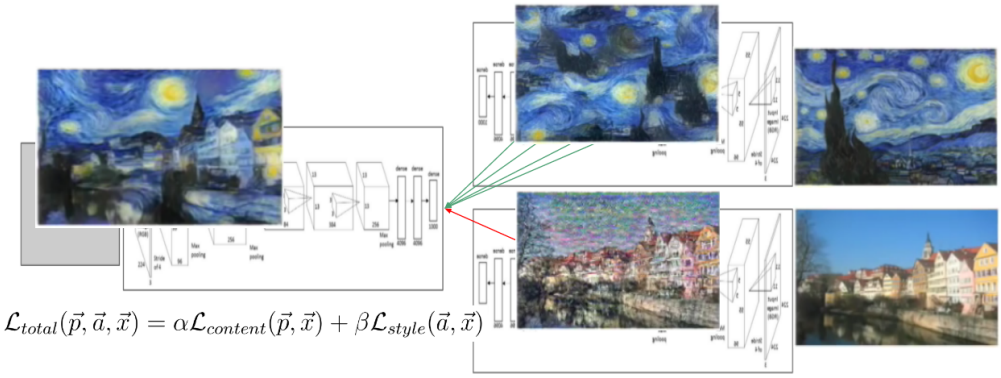


Figure 5: Functional principle of artistic neural style

## 2.6 Neural Style

That computers are better in calculating and evaluating date is a long known fact; creation of new things, like art was thought to be a human domain. Given the work of [Gatys et al., 2015] we maybe have to reconsider. The Tübingen working group used simple CNN architectures to turn a photograph into an image with unchanged content but the style of any painting of one's choosing. Photograph and painting are fed into CNNs, respectively. Then a new image is created from scratch where the activations of single neurons should be as closely to the original picture and the pairwise activations should match the ones of the painting. A measure how closely neurons are interconnected is given by the Gram matrix which is the inner product between vectorized feature maps in one layer. With that, the training loss can be written as a simple combination of L2-distances of activations (Fig. 5).

This can also be extended to combinations and mixtures of certain styles as seen in [Dumoulin et al., 2016].

## 2.7 Deep Dream

Another example of how algorithms show behavior that could be considered as somehow creative was demonstrated by a Google project called *Deep Dream*. They feed a picture into a CNN, amplify the activations of some neurons and backpropagate how an image with these new activations would have looked. After few iterations of this process results as in Fig. 6 are observed. Starting from a picture of clouds, the network "hallucinated" identifiable but previously unseen objects.

## 2.8 Super Resolution

Neural networks can also be used to reconstruct - or rather make up - images from little information. This has the potentially very serious application of identifying people from low-quality image data. [Dahl et al., 2017] deal with dramatically underspecified input data (cf. Fig 7) and recover plausible high resolution versions. To generate a

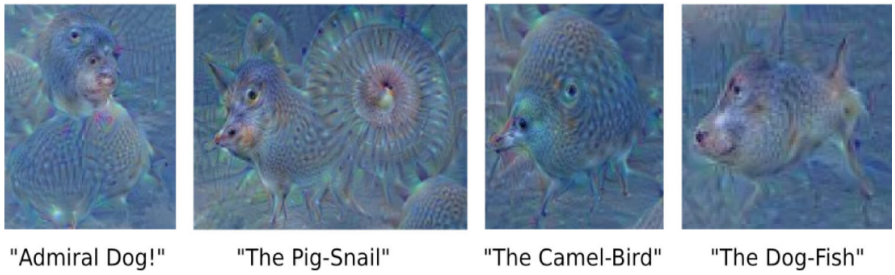


Figure 6: Results of Google Deep Dream applied on an image of clouds

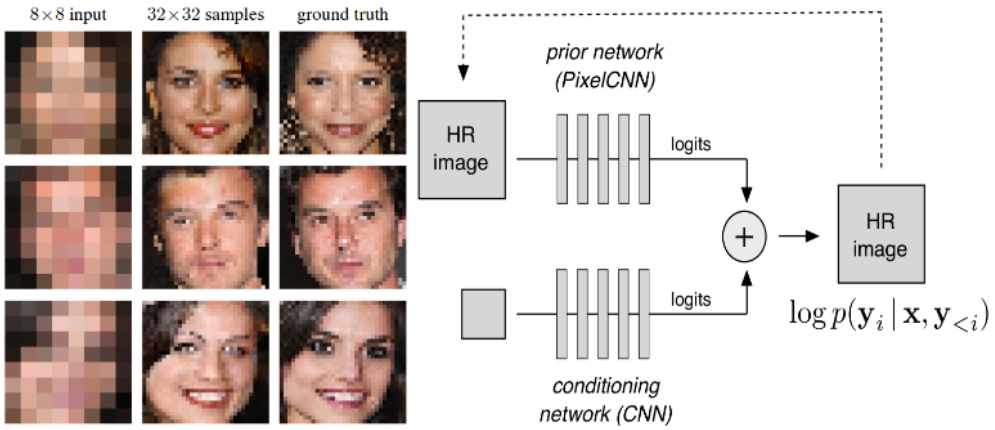


Figure 7: Results and schematic algorithm of pixel recursive super resolution

reasonable result, the prediction has to look human-like while still matching up with the input. Instead of presupposing strong priors, deep networks are trained end-to-end on a dataset of celebrity faces. A *conditioning* CNN outputs probabilities for high-res pixel values on basis of the input. Another network, a so called *PixelCNN*, iterates predictions for the overall result. Learning priors of faces and their typical variations gives impressive results. On the other hand training data is critical for the results which is why its use for official investigations or the like is potentially dangerous.

## 2.9 Generate Images

Generating images (or any other kind of suitable data) in an unsupervised fashion made a huge step by the introduction of Generative Adversarial Networks (GANs) by [Goodfellow et al., 2014]. They basically consist of a *Generator* network $G$ and a *Discriminator* network $D$ contesting with each other in a zero-sum game framework (Fig. 8). $G$ samples a vector $z$ from latent space and generates a fake image $x_{fake}$. $D$ receives either $x_{fake}$ or some actual picture $x_{real}$ sampled from a database and has to decide weather it is real or not. Given the feedback of $D$ being correct or fooled, both networks are trained until (in theory, actually training is eminently difficult) $G(z)$ generates

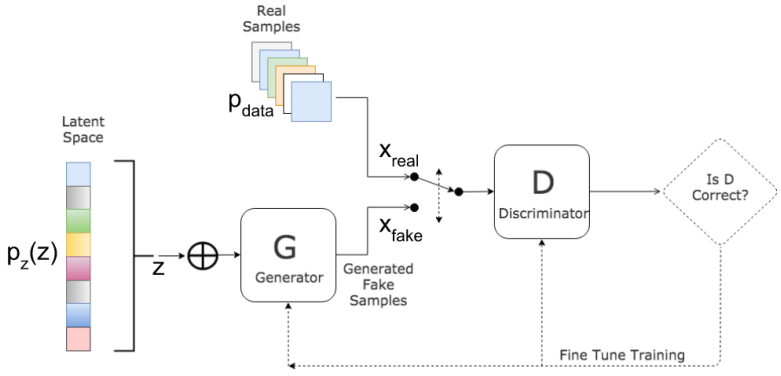


Figure 8: Generative Adversarial Networks, modified version of `kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html`

Figure 9: The leftmost and rightmost pictures were not part of the training data. Still, the network is able to find vectors $z$ that correspond most likely to the inputs By linear interpolation in latent space, $G(z')$ produces realistic images that morph from one input to the other in high resolution.

perfectly realistic images and $D$ with $D(x_{fake}) = \frac{1}{2} = D(x_{real})$ only left to guess.

How powerful this approach is for generating realistic images, interpolating between inputs or using $D$ and $G$ for other tasks, like feature extraction, was shown among others by [Radford et al., 2015] and [Berthelot et al., 2017] (one result in Fig9).

# 3 Outlook

*All the examples in Sec. 2 demonstrate that neural networks are key to recent progress in AI.*

The introduced concepts are by no means limited to the field of computer vision. Visual input can be generalized, so can acoustic signals be represented by waveforms and be processed in similar ways (overview on speech recognition by [Hinton et al., 2012]). There is progress in almost every field of science by applying ML-algorithms based on ANNs. Describing them would go beyond the scope of this work but considering progress in computer vision gives already a good impression of the impact of neural networks.

Tasks like classification (Sec. 2.1) can be further specialized for medical use like diagnosing cancer (e.g. [Esteva et al., 2017]). Detecting objects in real time (Sec. 2.3) is crucial for self-driving cars. Image captioning (Sec. 2.4) can be a new way for blind people to see the world. Developing strategies and anticipation of future events is not only helpful for games (2.5) but for every decision making process.

If we talk about AI, we are not satisfied by a computer system that solves one specific task above human performance. At least today, we expect more of it, like autonomous problem solving or creativity. Algorithms that are capable of creating art (Sec. 2.6) or previously unseen (2.7) or realistic (Sec. 2.9) objects are a first step in this direction.

# References


[Berthelot et al., 2017] Berthelot, D., Schumm, T., and Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks.

[Dahl et al., 2017] Dahl, R., Norouzi, M., and Shlens, J. (2017). Pixel recursive super resolution.

[Dumoulin et al., 2016] Dumoulin, V., Shlens, J., and Kudlur, M. (2016). A learned representation for artistic style.

[Esteva et al., 2017] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118.

[Gatys et al., 2015] Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neu-

ral networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Huang et al., 2016] Huang, G., Liu, Z., and Weinberger, K. Q. (2016). Densely connected convolutional networks.

[Johnson et al., 2015] Johnson, J., Karpathy, A., and Fei-Fei, L. (2015). Densecap: Fully convolutional localization networks for dense captioning.

[Karpathy and Fei-Fei, 2014] Karpathy, A. and Fei-Fei, L. (2014). Deep visual-semantic alignments for generating image descriptions.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. pages 1097–1105.

[Kumar et al., 2016] Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1378–1387, New York, New York, USA. PMLR.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lu et al., 2016] Lu, J., Xiong, C., Parikh, D., and Socher, R. (2016). Knowing when to look: Adaptive attention via a visual sentinel for image captioning.

[Mnih et al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning.

[Nguyen et al., 2014] Nguyen, A., Yosinski, J., and Clune, J. (2014). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.

[Radford et al., 2015] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

[Redmon and Farhadi, 2016] Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger.

[Russakovsky et al., 2014] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Li, F. (2014). Imagenet large scale visual recognition challenge.

[Sermanet et al., 2013] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks.

[Szegedy et al., 2013] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks.