Supervised and Self-Supervised Learning in Deep Convolutional Neural Networks as Computational Models for Object Recognition

Philip Oosterholt

Brain and Cognitive Sciences, University of Amsterdam

Date:	09/10/2020
Student number:	10192263
Supervisor:	Dr. H.S. Scholte
Examiner:	Dr. H.S. Scholte
Assessor:	Dr. Y. Pinto

Abstract. Creating artificial vision has been a long-held goal for artificial intelligence. The introduction of Deep Convolutional Neural Networks (DCNNs) was a giant step in that direction. DCNNs are especially well-suited for object recognition tasks. While not designed as models for the brain, neuroscientists discovered that DCNNs predict neural data to an unprecedented degree. This resulted in a new wave of research—while at the same time drawing heavy criticism. Sceptics characterized DCNNs as black-boxes and argued that such biologically implausible models could not provide satisfactory explanations for object recognition. In this review, I discuss the scientific value of DCNNs as models for object recognition. I argue that by building computational models that are functionally similar to humans we create a new framework to test and explore theories and hypotheses. In this light, I evaluate supervised and self-supervised learning in DCNNs as models for object recognition. The simple supervised cost function allows the model to learn a rich inner world of features with a striking resemblance to the visual cortex. However, there are clear limitations to supervised learning. The performance is far less robust than human's performance while the input data and strategies are different. Most importantly, humans learn in a self-supervised and task agnostic manner. Recent breakthroughs in self-supervised learning now provide a more biologically plausible way of training DCNNs. These self-supervised models show similar predictive power in object recognition tasks despite being (pre)trained in a task agnostic manner. Features are likely applicable to many downstream tasks and can help us to go beyond modelling object recognition. I argue that future self-supervised models are well suited to research computational mechanisms underpinning perception.

Content

1. Introduction

1.1. Convolutional neural networks: a synergy between artificial intelligence and neuroscience

For a long time, human-level performance on visual tasks seemed far beyond the grasp of artificial intelligence. This abruptly changed when Krizhevsky, Sutskever and Hinton(2012) won the 2012 ImageNet competition for object classification by an overwhelming margin. Their model, a convolutional neural network (CNN), was able to predict neural data to an unprecedented degree. The origin of CNNs can be traced back to the work of neuroscientists Hubel and Wiesel. Over 60 years ago, Hubel and Wiesel (1959, 1962) described the response properties of neurons in the visual cortex and classified the neurons as *simple* or *complex cells*. Simple cells respond primarily to oriented edges and gratings and have small receptive fields. While complex cells respond to the same type of stimuli, they display a larger degree of invariance and the receptive fields are twice the size (Serre, 2014). Inspired by this work, computer scientist Fukushima built a multi-layered, self-organizing artificial network (Fukushima, 1980, 2007). The neurons in the network were modelled after simple and complex cells. Akin to the visual cortex, local features, such as lines in particular orientations, are extracted in the early layers. More global features are subsequently extracted in later layers. In 1998, LeCun, Bengio and Hinton released LeNet, a 7-level CNN. LeNet was trained through backpropagation, which computes the gradients with respect to the weights of the network (LeCun et al., 2015). The next breakthrough was not related to the architecture, but rather the computational power needed for training the networks. With the help of graphics processing units, researchers were able to train CNNs increasingly faster, paving the road for more complex networks. Building on this foundation, Krizhevsky et al. (2012) introduced AlexNet, a 60-million parameter CNN, containing a total of five convolutional layers and three fully-connected layers. In the following years the artificial intelligence field exploded, research groups around the world started to build deeper and more complex CNN's, each new addition improving upon previous architectures. We have reached the point where some consider CNN's to be superhuman in terms of performance on object recognition tasks (He, Zhang, Ren & Sun, 2015)

These breakthroughs in artificial intelligence provided fertile ground for interdisciplinary collaboration. For the last several decades, high-level vision research is framed in terms of object recognition (Cox, 2014). Despite the fact that vision is much broader than identifying to which category an object belongs, the approach helps us to understand the basic properties of high-level visual processing. Shortly after the introduction of AlexNet, neuroscientists discovered that deep CNNs (DCNNs) could predict neural data. Despite the successes, there is considerable debate about the status of DCNNs as scientific models (see for example Kay, 2018)**.** Here, I discuss DCNNs as models for object recognition with a particular focus on learning. This review starts with a discussion on how DCNNs can

be used as computational models. Then, the diversity in learning methods is reviewed followed by an evaluation of how supervised models learn features and strategies in comparison to humans. To reconcile the limitations of supervised learning as models for object recognition, I evaluate the present and future potential of self-supervised models. Finally, I discuss the current limitations and open questions while providing suggestions on how to move forward and increase our understanding of object recognition and perception as a whole.

Box 1. Convolutional neural networks. CNNs are a specific class of neural networks and are commonly used for analyzing images. The general architecture of CNNs consists of an input layer (usually an RGB image) followed by multiple convolutional layers that extract features and one or more fully connected layers to classify the image. Neurons in the convolutional layer are organized in feature maps, each neuron is connected to the feature maps of the previous layer through a set of weights called a filter (LeCun et al., 2015). Filters can be seen as feature detectors; they look at a small part of the input image (corresponding to their receptive field) to see if those specific features are present. Mathematically, this is done by a convolutional operation between the input image and the filter. This operation is applied across the whole input. Convolutional operations can be followed by a pooling operation. Pooling operations reduce the size of the feature maps to speed up the computations. An example of a pooling operation is max-pooling, where only the maximum activation value of (usually) non-overlapping subregions of the feature maps are extracted. After each convolutional layer, a non-linearity is applied to the feature maps, for example, ReLU sets all the negative input values to zero while maintaining positive values. The convolutional part of the network is followed by a set of fully-connected layers that use the extracted features to classify the image. Finally, the network normalizes the output to a probability distribution of the output classes.

Difference between CNNs and fully connected networks. Theoretically, a fully-connected network could learn features, however, without the convolutional and pooling operations, the network would need to contain an infeasible number of neurons. In a fully connected network, neurons do not have a receptive field, rather, each pixel is treated as a relevant variable. On the other hand, CNNs use filters to exploit the repeating structure of the world. Once learned, these filters can be applied across the whole image. This approach reduces the required number of parameters dramatically and makes the task computationally feasible.

Similarities and differences between the architecture of the brain and CNNs. CNNs are inspired by biological visual systems, many elements are thus biologically plausible. Convolutional operations followed by pooling are based on the classic notions of simple cells and complex cells (LeCun et al., 2015). Neurons in both systems have receptive fields and both increase in size along the hierarchy of the system. Moreover, akin to CNNs, the visual cortex is thought to have a series of non-linear operations (Wielaard et al., 2001). Despite using the brain as a source of inspiration, the machine learning field is not constrained by biology and their approach is pragmatic. This has resulted in implementations that (arguably) cannot be implemented in biological networks, such as the neuron's access to non-local information (see *3.2.3. biological plausibility of gradient descent*). Moreover, CNNs are a simplification of biological neural networks. For example, unlike the visual cortex, DCNNs generally do not contain lateral and feedback connections (Lamme, Super & Spekreijse, 1998). Moreover, the artificial neurons are highly abstracted and lack most of the dynamics of their biological counterparts (Cichy & Kaiser, 2019).

2. Deep Convolutional Neural Networks as Computational Models in Neuroscience

2.1. What are computational models?

In the broadest sense, computational models are mathematical descriptions of a system and/or the behaviour of a system. In neuroscience, computational models aim to capture complex adaptive behaviour and the underlying neural information processing. Building a perfect all-encompassing model of a cognitive phenomenon is still a far cry from reality. Instead, the model maker should make choices between desirable properties of theoretical and non-theoretical nature (Cichy & Kaiser, 2019). Theoretical desirable properties are realism (how close the model is to the phenomenon), precision (how precise the model's predictions are) and generalizability (how well the model generalizes to different instances). When building a model, the level of detail should be taken into consideration. For example, when modelling neuronal microcircuits, one has to make a decision what internal mechanisms of a neuron should be described. Other desirable properties are of non-theoretical nature, such as speed and the efficiency of computation, ease of manipulation and ethical considerations. Since no neuroscientific computational model can have all these desirable properties, scientists came up with a large number of widely different computational models. Even considering the wide variety of computational models in neuroscience, DCNNs are the odd one out. DCNNs were never intended to be a computational model per se, instead, they are designed to perform similar tasks as humans. In practice, this meant that biological realism was often disregarded in favour of precision. Neuroscientists have to decide ad hoc how DCNNs should be employed as computational models. Cichy and Kaiser (2019) suggest deep neural networks have two main goals: prediction and explanation. Beyond these two main goals, deep neural networks could/can be explored in an unprecedented way, which can help us generate novel hypotheses. In the next section, I discuss DCNNs in relation to these goals.

2.2. Prediction

DCNNs are unquestionably successful in terms of predictive power (see Box 1 for the techniques and 4.1. for predictive studies). However, when it comes to scientific models, explanation is often preferred over prediction (see for example Kay, 2018). This view overlooks the fact that prediction and explanation are mutually dependent, a model explaining a system without the ability to predict the system is of little scientific value. Moreover, a model with perfect predictions does not necessarily translate into powerful explanations since it could result in the exchange of one impenetrable black box for another. Practical machine learning models are opportunistic; the models use any type of data which uniquely explains variance to the outcome. There is no assurance that these models capture the true

interaction between variables—and even if the models did, the explanation would be abstract and high-level. DCNNs have much more value than such machine learning models. DCNNs are not designed to predict certain outcomes, but rather perform a certain task. Nevertheless, the internal state of DCNNs has been shown to be predictive of independent data such as neural data and human behaviour patterns (see 4.1.). Even though this behaviour could come about via different mechanisms, the fact that DCNNs are able to predict different types of data and produce behaviour at the same times is of much bigger scientific value than either capability by themselves. Prediction should serve as a validation of the models, the outcome can subsequently guide the successful development of better models (see 2.5.).

Box 1. Methods: How to evaluate the predictive power of DCNNs

Prediction of neural data. To predict neural data a linear read-out layer is added on top of one of the DCNN layers and subsequently trained and tested on a hold-out set. Neural data can be from various imaging techniques: e.g. fMRI, EEG, MEG, invasive electrophysiology (single or multi-unit recordings). **Example:** Yamins et al. (2014) recorded responses to images of neurons in the Inferior Temporal cortex (IT) of rhesus macaques. A linear read-out layer was trained on top of the DCNN output layer. For each IT neuron, the unit with the highest predictive power for the IT neurons response was selected and subsequently tested on a new set of images. Yamins et al. found that the DCNN read-out layer was highly predictive of neural responses in the IT, predicting 48.5% of the variance.

Prediction of internal representations. Another way of probing the brain is testing if DCNNs representations can predict the internal representations of the brain. An example of such a method is Representational similarity analysis (RSA). The method uses Representational dissimilarity matrices (RDM), which store the dissimilarities of a system's response (neural or model) to all pairs of experimental conditions (Kietzmann, McClure & Kriegeskorte, 2019). RDMs characterize the information carried by a given representation in the system (Kriegeskorte, Mur & Bandettini, 2008)). The advantage of RSA is that responses measured by different imaging modalities and computational models can be directly compared with each other. **Example:** Khaligh-Razavi and Kriegeskorte (2014) used RSA to compare different models on their ability to account for IT representational geometry by comparing RDMs of DCNNs, human (fMRI) and monkey IT cortex (single units) for the same stimulus set. Of all the tested computational models, DCNNs were the most similar to IT in that DCNNs showed greater clustering of representational patterns by category. The authors suggest that features derived from supervised learning might be needed to create a behavioural relevant division of categories in IT.

Prediction of behaviour. Since DCNNs are designed to perform tasks, the behaviour can be compared to humans. It is crucial that the training set contains the necessary information needed for the task, as DCNNs first have to learn the task. The most straightforward measure is the overall accuracy. For this to work, the test-set should force differences in performance. For example, image perturbations such as noise can be applied to see how this influences the accuracy. Another method is to compare the errors between humans and models, this tells us things about the strategies and features of both systems. This can be done on object-level or image-level. Even though the accuracy can be similar on the object level, the type of errors can widely diverge. The previously discussed method RSA can be used as well since RSA can be used across different types of modalities (Kriegeskorte et al., 2008).

2.3 Explanation

Models should lead to an accurate understanding of the modelled system. Traditional computational models in neuroscience contain a limited number of relevant variables and interactions between those variables. The variables and interactions are then modelled mathematically (Cichy & Kaiser, 2019). Since the variables and their interactions are determined a priori the resulting changes in the variables are directly interpretable. This approach is bound to fail since it requires knowledge of the solution itself. It is unlikely that we can infer the solutions by reason or even by gathering neural data. On the other hand, DCNNs learn the solutions by experience. The consequence of this approach is that the solutions are encoded in millions of parameters in DCNNs. It is therefore challenging to find the direct mapping between the parameters and a part of the modelled system. We thus run the risk of exchanging one black box for another. Many disagree with the statement and argue that DCNNs can help us understand the brain (see e.g. Scholte, 2018a; Serre, 2019; Yamins & DiCarlo, 2016; Kriegeskorte & Douglas, 2019).

Cichy & Kaiser argue it is deceiving to use DCNNs in the same way as traditional mathematical–theoretical models. Rather, we should limit ourselves to the variables that give rise to the solutions, such as the architecture, cost functions and training data. DCNNs thus choose to highlight a different aspect of the system and provide explanations of the same quality as the traditional models. Moreover, DCNNs offer a different kind of explanation. We can look at neurons and circuits of neurons as carrying out certain functions within the overall objective of the system. Finally, DCNNs provide a diverse and ever-expanding set of toolboxes to explore their inner world (see Box 2). In the next paragraph, we discuss the added benefit to DCNNs unprecedented degree of access.

2.4 Exploration

DCNNs lend themselves for exploration and provide fertile ground for the creation of new hypotheses. Whereas we have a wide toolset to explore the behaviour and cognitive abilities of humans, we are limited in how we can probe the underlying neural dynamics, both for ethical and technological reasons. As for DCNNs, we have complete access to every single neuron and its weights. We can change the architecture and training regime with a few lines of code, all without any welfare-concerns for humans or other animals. By exploration and experimentation, we can make much stronger inferences about what type of training and computational mechanisms explain behaviour and patterns of activity in neural data (Scholte, 2018a).

Exploration of the DCNNs under a wide variety of circumstances might reveal behaviour that we might not necessarily expect. Some of these findings might result in the reevaluation of neuroscientific theories. For example, classic vision models of segmentation presume an explicit process where an object is segregated from the background. DCNNs do not have such an explicit step built-in. Seijdel, Tsakmakidis, De Haan, Bohte, & Scholte (2020) found that an increase in network depth allows for a

better selection of the features that belong to the target object. The authors suggest that recurrent computations might be one of the possible ways in which scene segmentation is performed in the brain. Studies such as this provide a *learnability argument* for certain behaviour. This does not necessarily imply that the brain does the same thing, but the exploration of DCNNs does provide inspiration for new theories which can consequently be tested.

Box 2. Methods: How to explore the inner world of DCNNs?

Feature visualization techniques use the mathematical properties of neural networks to show what input makes specific parts of a network fire (Olah & Schubert, 2017). Neural networks are differentiable to their input, therefore we can iteratively tweak the input towards whatever input maximizes its response. We can do this for neurons, channels, layers, class logits and class probabilities. We can also search for which images cause neurons to fire maximally, however, this can be deceiving at times. For an example of feature visualization, see Fig. 1-7. Feature visualization is a new and active research area. There is still no consensus about what the correct optimization techniques are and at which level these optimization techniques should be applied, nor do we know how to exactly interpret the visualizations.

Attribution techniques show how networks (and specifically its parts) arrive at a decision (Olah et al. 2018). Just like with features, attributions can be visualized. The most common attribution technique is a saliency map, which shows which pixels of the input image contributed the most to the final decision. This approach has considerable flaws (Olah et al. 2018). First of all, salience maps show one single class at the same time. Second of all, pixels are most likely not the most interesting units (pixels are devoid of high-level constructs; they are not independent of their neighbours). To arrive at more insightful conclusions it is wise to combine both attribution and feature visualization techniques (see Fig. 8). Instead of asking which pixels contributed to the classification (for example a labrador retriever), Olah et al. (2018, 2020a) asked whether a high-level concept, such as a floppy ear, was important during the classification process. Combining feature visualization with attribution might be one of the most powerful ways to explore DCNNs and gain intuitions about the inner workings of our own visual cortex.

2.5 *Functional approach to modelling*

Even though the way DCNNs are implemented is unlikely to arise in biological systems (see next chapter for a discussion), it is important to note that this is not an invalidation of DCNNs as computational models. To draw an analogy, when building a bridge with a certain set of capabilities, the design will be dependent on the constraints (e.g. time, money, tools and the availability of materials). Different constraints will give different implementations of the bridge—importantly, functionally the bridge will remain the same. In line with this reasoning, a part of the brain might functionally be the same as a computational model while the details of the implementation are different. An important question to ask is if two systems behave in an identical manner under various circumstances, does the actual implementation of the systems matter? In this review, I argue that implementations do not matter as long as the behaviour is the same and we should foremost be interested in modelling behaviour instead of giving a mechanistic account. Attempting to provide a mechanistic account with a model that cannot perform the behaviour itself is nonsensical since one would be clearly missing some elements in

the model. The functional approach does not exclude us from taking note of the underlying structure of the biological system. In fact, it generally is a good idea to look at nature and attempt to copy its solutions. Nevertheless, we do not have to constrain ourselves to how we implement the solution *in silicio*. After constructing a functionally similar computational model, researchers can reconstruct the model step by step in a biologically plausible manner. If this is possible, we can say that we have a high-level understanding of the phenomenon. Sceptics might argue that we have exchanged one black box for another—however, later on, we see that this critique is unfounded. The functional approach implies that we should start out with a focus on building a DCNN that can perform the behaviour we want to study. Importantly, predictions are derived from the behaviour and the internal state of DCNNs and not merely the prediction of future variables in the modelled phenomenon. Later on, we see that this approach can result in explanations on a lower-level. In the next chapter, I review the different learning paradigms in DCNN.

3. Learning in Deep Convolutional Neural Networks

In this review, we evaluate DCNNs as computational models for object recognition in light of the functional approach to modelling. Regarding DCNNs there are broadly speaking three important components when it comes to their capabilities, namely the *architecture*, the *learning paradigm* and the *training data*. Learning can be divided into two subcomponents, namely *cost functions* and *learning rules* (Richards et al., 2019). In this review, I mainly focus on the pivotal role of the learning paradigm for acquiring its abilities while briefly touching on the role of architecture.

3.1. Learning paradigms

In machine learning, there are three dominant learning paradigms, supervised, self-supervised and reinforcement learning (LeCun et al., 2015). Learning paradigms differ from each other in how and what they learn from the data. Supervised learning methods learn a function that maps the input to the output data by using labelled data. Self-supervised learning does not require labels, rather, the model learns the structure of the dataset. Finally, in reinforcement learning, every decision the model makes is tied to a reward, and the model changes its strategy to maximize the reward.

Every model has a cost function (also known as a *loss function*) which maps the values of variables onto a real number to represent the cost associated with the decision. The model attempts to optimize the cost function by minimizing the loss. Every learning paradigm has its own set of cost functions and which can vary between specific instances of models. Simple cost functions can lead to models with rich features and capabilities. It is probable that the brain uses and optimizes cost functions in a similar manner (Marblestone, Wayne & Kording, 2016). These cost functions are diverse and differ across brain locations and development stage. The cost functions can be used to study the brain itself by

showing that a specific cost function can create behaviour and/or functional organization (for example see Scholte, Losch, Ramakrishnan, de Haan & Bohte).

3.2. Supervised learning

The most popular learning paradigm for endeavours such as object recognition is supervised learning. For supervised learning, we need both the input (e.g. an image of a dog) and the label. The model predicts the label and subsequently compares the prediction to the actual label. The cost function (typically cross-entropy loss) then provides the error. Subsequently, the model updates its parameters to reduce the loss. This is generally done through gradient descent (LeCun et al., 2015). Gradients give us an idea of how the loss function would increase or decrease when we increase the value of the parameter. Backpropagation, which is a practical application of the chain rule of derivatives, enables us to calculate the gradients backwards from the top layer to the input layer. Finally, we take a step in the direction of the gradients that reduces the loss. This process repeats for all the images in the training set for multiple epochs. After each epoch, we check how the network performs on previously unseen images. Ideally, the network has learned a set of features that are generalizable to new examples. When training networks, a balance has to be struck between under and overfitting. Overfitting is a phenomenon that occurs when the network learns features that correspond too closely to the idiosyncratic characteristics of the training dataset. In this case, the learned features are not generalizable to examples that are not part of the dataset. By using specific regularization and training techniques we attempt to train up until the point that the network has learned robust, invariant features that generalize to new data. The most important training technique is augmenting the training set by adding copies of slightly changed images of the original training data. For example, randomly rotating the image by a given number of degrees from 0 to 360. These augmentations improve the ability of DCNNs to detect features regardless of the variations of appearances. In practice this only works for invariance for features that are similar to the ones seen in the training set (e.g., if a specific scale or rotation of a feature is too different the model does not detect this feature).

When training DCNNs in a supervised manner, we force DCNNs to identify task-relevant features. There are non-trivial consequences to this training protocol. First of all, the network is highly dependent on the input, meaning the network will learn any predictive statistical regularities there are present in the data. This could result in learned features that are not intrinsic to the predicted class, but rather a recurring bias in the training set. The second consequence is that the network will only learn features that are relevant to the specific task. This can potentially lead to a very narrow set of features.

3.2.1. Psychological plausibility of supervised labels

Supervised learning requires millions of labels to reach human-level performance while humans learn to discriminate between objects based on only a handful of examples. On the other hand, humans have access to a constant stream of unlabelled visual input. Supervised models cannot learn from unlabeled data and are thus inherently limited to explain learning itself (although the learned solution might still be comparable to our brain's solution).

3.2.2. Lack of generalizability of supervised learning

supervised learning allows DCNNs to learn how to perform a certain task. While DCNNs show outstanding object recognition performance[1], the knowledge is not easily transferred across domains. In contrast, humans can perform a wide variety of tasks effortlessly and it takes little effort to learn a new type of task. The knowledge of these supervised DCNNs is thus confined to a very specific domain, while humans are flexible and can apply knowledge effortlessly across domains.

3.2.3. Biological plausibility of gradient descent

In DCNNs gradient descent is generally implemented with the backpropagation algorithm. For backpropagation to work, each neuron requires access to non-local information (i.e. all the weights of all the downstream neurons). This implementation is considered to be biologically implausible since real neurons do not have backward connections and subsequent access to the non-local information is thus impossible (Pozzi, Bohté & Roelfsema, 2018; Millidge, 2020). Backpropagation is used because it is currently the fastest way of training neural networks. The implausibility of a learning rule does not invalidate DCNNs as scientific models since optimizing cost functions can be done with many different types of learning rules.

3.3. Reinforcement learning as an alternative learning rule

Our brain cannot compute the loss and update all its synapses based on one cost function (Pozzi et al., 2018). The question then is how the brain implements learning with only locally available information and the one-way direction of information flow?[2] Reinforcement learning is a prime candidate for a biologically plausible learning rule as it is thought to be used throughout the brain (Marblestone et al., 2016). In the field of artificial intelligence, reinforcement learning is used to train an agent to learn to perform a certain task without the help of external labels. In a typical reinforcement learning model, the agent actively explores the environment while making decisions. Each decision has direct rewards coupled to it and the agent's job is to maximize its rewards (Mosavi, Ghamisi, Faghan & Duan, 2020). Reinforcement learning plays a prominent role in tasks such as autonomous driving. In object recognition, reinforcement learning can play a role with and without labels. With external labels, it

[1] Later on we see that this is not always the case.
[2]Information flow in recurrent networks is still only one way, a single neuron can not transfer information through its dendrites.

would learn in a similar fashion as supervised learning. In this case, it can be seen as a learning rule. Moreover, reinforcement learning can use internally derived rewards as a learning signal, for example reducing uncertainty might function as a reward. Importantly, reinforcement learning allows for local learning rules and thus cost functions to be optimized locally.

3.4. Learning without guidance: Self-supervised learning

Both supervised and (current) reinforcement learning methods require training signals designed by humans (Graves & Clancy, 2019). On the other hand, self-supervised learning methods do not require pre-existing human-derived labels to learn the structure of the world. Self-supervised learning is also known as unsupervised learning, however, there is a push in the artificial intelligence community to rename the learning method as the term is "*loaded and confusing*" (LeCun et al., 2020). While traditional unsupervised learning methods such as autoencoders had no form of external or self-derived labels, new techniques provide the supervisory signals themselves. This signal is much stronger than the signal of traditional supervised learning. Popular new learning methods are contrastive and adversarial learning. Contrastive learning uses the input to maximize agreement between similar images and minimize agreement between different images through a prediction task (Chen, Kornblith, Norouzi & Hinton., 2020a). For the prediction task, augmentations are used to derive labels. Adversarial learning is part of the generative model family and learns the structure of the world by attempting to recreate it (Goodfellow et al., 2014). In adversarial learning, a part of the model, the discriminator, is trained with supervision. The generative part creates its own instances of data and can thus be considered as self-supervised learning. For a more detailed description of generative models and adversarial learning see Box 5 & 6.

Finally, learning by predicting the future is currently underdeveloped in the field of computer vision. Such models are generative in the sense that the models generate predictions about the future state of the world. The self-supervisory signal comes from the actual input it receives at a later time point. Based on this information the model revises both the prediction of the current state of the world and its internal model of the world. This approach is closely related to the *predictive coding framework* in neuroscience. The framework states that the brain makes active predictions about incoming sensory signals based on the past and its internal model (Rao & Ballard, 1999). When the prediction and the incoming sensory information differ from each other, a prediction error is sent back. Only the part that is not predicted is passed through for processing. In the meantime, the internal model responsible for the prediction is updated to account for the discrepancies. Predictive coding thereby learns the structure of the world while making efficient use of its resources. To avoid confusion, I explicitly mention the word predictive when talking about these learning methods. In the next paragraph, I discuss contrastive learning, the most prominent self-supervised learning technique in computer vision.

Box 5. Learning by creating: Generative models

DCNNs can be trained with different goals in mind. This is (theoretically) independent from the chosen learning method. Discriminative models learn to discriminate between different categories based on learned features. It thus learns the conditional probability of the target Y, given an observation x (Mitchell, 2015). Discriminative models only need to learn features relevant for categorization. Generative models learn how to generate images themselves by computing the conditional probability of the observable X, given a target y. Even though generative models can be trained with various methods, most use self-supervised learning. Examples of generative models are variational autoencoders (VAE) and generative adversarial networks (GANs).

Box 6. Self-supervised adversarial learning with Generative Adversarial Networks

The most popular generative model, Generative Adversarial Networks (GANs) was developed by Goodfellow et al. in 2014. GANs consist of two separate neural networks, the generator and the discriminator. The generator is a CNN with reversed convolutional layers so that it can create images based upon randomized input. The generator's job is to fool the discriminator by generating realistic images. The discriminator, a traditional CNN, then classifies the image as either real (thus from the training distribution) or as synthetic. Just like in supervised learning, we can backpropagate through the networks to find how to change the generator's parameters to make its images more confusing for the discriminator. Eventually, the generator can mimic the real data distribution and the discriminator is unable to detect the difference. The beauty of this idea is that unlike supervised learning, generative models not only learn the features that are relevant for categorization, but also the features that are necessary to reconstruct the objects. The best performing generative models have a top-1 score of 72% (Chen et al., 2020c), despite the fact that object recognition is not the goal of generative models. Current generative models learn features that are generalizable across a wide range of visual tasks (Xu, Shen, Zhu, Yang, & Zhou 2020), and in the long run, generative models are thought to be able to automatically learn all the natural features of a dataset (Karpathy et al., 2016).

3.4. Self-supervised contrastive learning

In the last two years, contrastive learning has yielded impressive results (see Box 7). Contrastive learning methods use large amounts of unlabeled images (Chadhary, 2020). The idea behind contrastive learning (first introduced by Oord, Li & Vinyals, 2018) is simple and elegant, the DCNN is pre-trained by learning to predict which images are similar and dissimilar and as a result, the model learns the underlying structure of the visual world. In Box 8 the SOTA contrastive learning method SimCLR is described. Self-supervised contrastive learning builds stronger, invariant features by creating different "*views*" (through a set of augmentations) and subsequently contrasting what features are similar and dissimilar to each other. By doing so, the model learns features that support reliable and generalizable distinctions (Zhuang et al., 2020). After the self-supervised learning stage, the encoders can be optimized for specific tasks. However, the features are useful for a wide variety of downstream tasks, instead of just object recognition. Contrastive learning does not need an unrealistic amount of labels to perform well, for example, the latest version of SimCLR (Chen, Kornblith, Swersky, Norouzi & Hinton,

2020b) achieves an ImageNet top-1 of 74.5% and 77.5% with respectively 1% and 10% of the labels (see Table 1 for a comparison with supervised learning).

Box 7. State-of-the-art self-supervised learning

Up until 2018, self-supervised learning in computer vision was miles behind supervised learning. However, since then, self-supervised learning methods are quickly catching up in terms of performance. When used in the context of object recognition, a linear classifier is added on top of the pretrained network and subsequently trained in a supervised manner (Chen et al., 2020a). All other layers are frozen, the features are thus learned in a self-supervised manner. According to the ImageNet benchmark of Papers With Code the top-1 score for self-supervised learning, methods went from 35% in 2017, 54% in 2018, 70% in 2019 and now 80% in 2020, which is on par with the performance of a supervised ResNet-200[3]. The most successful self-supervised learning method for object recognition is contrastive learning.

Box 8. Self-supervised contrastive learning with SimCLR

SimCLR (Chen et al., 2020b) is currently the best performing contrastive learning method. SimCLR takes an image and then augments it with random transformations (e.g. crop or Gaussian noise) and then passes it through an encoder, which is a normal CNN, to get the image representations. The output is then passed through a projection head to apply non-linear transformation and project it into another representation. The pairwise cosine similarity between each augmented image is then calculated. Next, a softmax function obtains the probability that two images are similar, followed by a calculation of the contrastive loss. Based on the loss the encoder and projection head are subsequently optimized.

Model	Learning method	1% labels	10% labels	100% labels
ResNet-200	Supervised learning	23.1%	62.5%	80.2%
ResNet-152x3	Contrastive learning (SimCLR)	74.5%	77.5%	79.8%

Table 1. Top-1 ImageNet scores of DCNNs trained with supervised and contrastive learning methods on a limited number of labels. Supervised learning accuracy scores are reported in Hénaff et al. (2019) and contrastive learning accuracy scores in Chen et al. (2020b).

3.4.1. Biological plausibility of self-supervised learning

The scarcity of object labels encountered during learning in real-life implies that learning in biological systems is largely self-supervised. The lack of external labels is actually a good thing since the actual input to our visual system is much richer than any external label can provide. In addition, we can generate our own labels based by exposing the relations between the different parts of the data

[3] Self-Supervised Learning (2020, September 2). Retrieved from https://paperswithcode.com/task/self-supervised-learning.

(LeCun & Bengio, 2020). Therefore, it makes sense that we create models that exploit this richness. The learned features can be broadly applied as the features are not specifically designed for a certain task. Self-supervised models are thus more biologically plausible than their supervised counterparts.

Both contrastive and adversarial learning highlight some properties of the visual cortex that supervised learning is missing. Just like humans can visualize images, generative adversarial learning models can create synthetic images. Generative adversarial models thus might be able to serve as a computational model for certain tasks that the brain executes. Contrastive learning leverages the power of different views to learn features. On the other hand, we receive a continuous stream of visual information. Since we have two eyes and can change our gaze, turn our head and move our body, we have a continuous stream of different viewpoints. Arguably, the way contrastive learning is generally implemented is divergent from how we create different views as views are created to augmentations such as image crops.

In the next chapter, I discuss how supervised learning can be used as a model for object recognition.

4. Supervised Deep Convolutional Neural Networks as Models for Object Recognition

As discussed before, we evaluate DCNNs as scientific models for object vision on three different aspects, namely the capability to predict, explain and explore. In this chapter, I mainly focus on showing that at least to large degree supervised DCNNs learn similar features as humans (and other primates) do and that this accompanied with the capability to predict neural data. The chapter thus uses a combination of prediction and exploration.

4.1. The predictive power of supervised deep convolutional neural networks

DCNNs yield impressive results in terms of their predictive power for neural data (see Box 1 for a summary of the techniques). DCNNs predict neural responses in IT to a high degree, both for single-unit recordings in monkeys (Yamins et al. 2014) and fMRI recordings in humans (Storrs, Kietzmann, Walther, Mehrer & Kriegeskorte 2020). Furthermore, DCNNs can predict responses to early visual areas to a greater degree than previous models in both humans and other primates (V1, single-unit: Cadena et al. 2019; Kindel, Christensen & Zylberberg, 2019; V1, fMRI: Zeman, Ritchie, Bracci & de Beeck, 2020; V2: Laskar, Giraldo & Schwartz, 2020, V4, single-unit: Yamins et al. 2014). The hierarchy of DCNNs roughly corresponds to the hierarchy of the ventral stream, meaning downstream areas code for increasingly complex stimulus features that belong to increasingly deep layers in DCNNs.

This is seen both in space (fMRI: Güçlü & Van Gerven, 2015; Cichy, Khosla, Pantazis, Torralba & Oliva, 2016; Eickenberg, Gramfort, Varoquaux & Thirion 2017) and time (MEG: Cichy, Khosla, Pantazis & Oliva, 2017; Seeliger et al., 2018; EEG: Greene & Hansen, 2018). Finally, DCNNs replicate the representational structure of IT (Khaligh-Razavi & Kriegeskorte, 2014; Cadieu et al. 2014). While the predictive power of DCNNs is impressive, we still do not know how the correlation between the visual cortex and DCNNs comes about. Do both systems extract the same features (and in the same way) or can something else account for the correlation? Establishing a link between the two systems in terms of behaviour sets the stage for stronger inferences about what type of architecture, learning and computational mechanisms can explain behaviour and neural data (Scholte, 2018a).

4.2. Comparing features between DCNNs and humans

Features can be seen as the link between the system and its behaviour. Features refer to a set of properties of the visual input and are the building blocks of object recognition. An example of a low-level feature is a horizontal line. When we say that a system uses a certain feature, we mean that the system can extract this feature from the input. DCNNs (and arguably the visual cortex) extracts these features through a set of convolutional operations (LeCun et al., 2015). Throughout the hierarchy of both systems increasingly complex features are extracted (Serre, Oliva & Poggio, 2007). If we hypothesize that DCNNs perform object recognition in a similar fashion as the brain, we should encounter similar features. In this way, DCNNs serve as proof that certain features can be learned given a certain architecture and learning method. Moreover, by exploring features in DCNNs we might encounter unexpected findings. If these features cannot be found through subsequent imaging studies, this could point us to fundamental differences between both systems. On the other hand, if certain neurons or populations of neurons do turn out to be tuned for that feature, we have used exploration as a technique to drive and successfully test new hypotheses.

In the following section, I draw parallels between DCNNs and the brain. Importantly, this does not directly prove that the DCNNs and the brain do the exact same thing. Many of the upcoming findings are not yet compared to the brain. Before starting the comparison, I discuss the important caveats of the feature-based approach.

4.2.1. Caveats

In neuroscience, features are generally studied indirectly by finding the response properties of individual neurons or brain areas. The stimuli are usually artificial, simple and designed in advance. The disadvantage of this approach is that the set of stimuli tested is limited. We cannot exclude that there is another type of stimuli that results in stronger responses. On the other hand, feature visualization shows what type of stimuli maximizes the response of artificial neurons. Neurons are connected to hundreds of

other neurons and it is likely that each neuron plays multiple roles. Moreover, functionality that arises in large populations of artificial neurons will inadvertently be missed. Finally, it should be noted that studies on the response properties of single neurons and the population of neurons are generally done in animals. If not stated otherwise, all following neuroscientific studies on features are done in non-human primates.

4.3. Learning low-level features in supervised deep convolutional neural networks

Olah et al. (2020b) were the first to conduct an exhaustive search for low-level features in the first layers of a DCNN. Low-level features are the most straightforward features to research as all DCNNs seem to contain the same ones. Moreover, the features are relatively simple and the number of neurons is small. This allowed Olah et al. to track what neurons in the previous layer excite or inhibit a single neuron to build features. The authors optimized each neuron to maximize its response and then divided the neurons into ad-hoc determined categories based on their visualizations. This method helps us to think about the roles different neurons can possibly play. In order to compare this implementation with the implementation of the brain, I follow the hierarchy of the DCNN used by Olah et al. (InceptionV1, Szegedy et al., 2015). The following part is by no means an exhaustive comparison between low-level features in DCNNs and the brain, rather, I attempt to show that both systems contain many similar low-level features.

In the first layer, Olah et al. (see Fig. 1, 2020b) found a class of neurons that were sensitive to the specific orientation of edges. The authors labelled these neurons as *Gabor filters*, similar to a type of simple cells found in the V1 (Daugman, 1985). Besides the Gabor filters, *colour-contrast* neurons are present in the first layer, which detect one colour on one side of the receptive field and another colour on the other side. The colour-contrast neurons can also be found in V1 (*double opponent cells*, Shapley & Hawken, 2011). Rafegas & Vanrell (2018) noted that both the V1 and the first convolutional layer display a clear distinction between colour and non-colour neurons. Moreover, the colour neurons display low spatial selectivity while the non-colour neurons have high spatial frequency selectivity.

Figure 1. Neurons in the first layer of InceptionV1. The first three images are Gabor filter neurons (44% of all the neurons). The next three examples are colour-contrast neurons (42%). Finally, the role of the last neuron (15%) is unclear. The images are smoothed for visualization purposes. Images are adapted from Olah et al. (2020b) under Creative Commons Attribution CC-BY 4.0.

In the second layer we again find Gabor and colour-contrast neurons, albeit more complex and invariant (Fig. 2, Olah et al., 2020b). The complex Gabor neurons are made out of Gabor filters in the previous layer. These complex Gabor neurons are relatively invariant to the position of the edge (i.e. the neurons do not display *phase selectivity*) and the colour composition of the input. These neurons are behaviourally similar to *complex cells* present in the primary visual cortex. In both the DCNNs and the brain they non-linearly combine the input of previous (simple) neurons. The response profile of complex cells can be interpreted as the *magnitude* of the *Gabor components* extracted by simple cells (Shams & von der Malsburg, 2002). The Gabor magnitudes are tuned to a specific orientation and frequency but lack spatial phase selectivity. The DCNN complex Gabor neurons are formed by putting together multiple layer 1 Gabor filters with the same orientation but different phases. As a result, these neurons lose their spatial selectivity. The same is observed in complex cells present in V1 (Victor & Purpura, 1998).

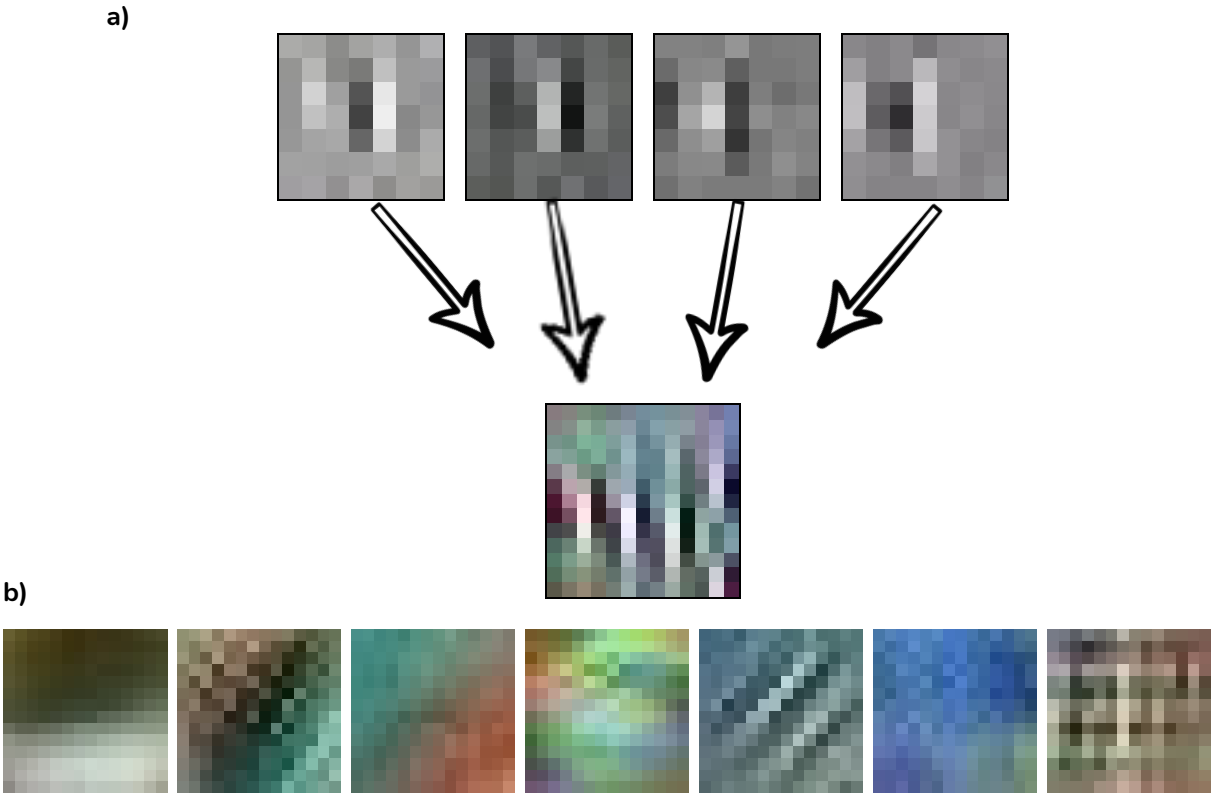


Figure 2. Neurons in the second layer of InceptionV1. (*a*) Simple Gabor filters in Layer 1 (the four at the top) are the building blocks of complex Gabor neurons (below). (*b*) Layer 2 shows a greater variety of neurons. From left to right, low-frequency edge pattern neuron (27% of all the neurons), Gabor-like neuron (broad category, 17%), colour-contrast neuron (16%), multi-colour pattern neurons (14%), complex Gabor neuron (14%), simple colour neuron (6%) and a hatch-like pattern neuron (2%). Images are adapted from Olah et al. (2020b) under Creative Commons Attribution CC-BY 4.0.

Moreover, Olah et al. found neurons that respond to multi-colour patterns, colours (specified for brightness or hue) and low-frequency edge patterns. In V1, so-called *single-opponent cells* respond to large areas of colour (Shapley & Hawken, 2010). Just like the DCNN single-colour neurons, these V1 single-opponent cells can be selective for hue (Xiao, Casti, Xiao & Kaplan, 2007) and brightness (Kinoshita & Komatsu, 2001).

In the third layer neurons responding to shape arise (Fig. 3, Olah et al., 2020b). Around 25% of the neurons respond to single lines, sometimes with different colours on each side or with small perpendicular lines to the main one (this peculiar feature can also be found in the visual cortex, see Tang et al. 2018). Besides straight lines, we see the start of curve, corner and divergence detectors. The origins of these neurons can be traced back to previous layers. The third layer is a 3x3 convolution, where we can, for example, see that a centred vertical line detector consists of three vertically orientated Gabor segments at the middle of the receptive field. From the response profile of a single complex cell in the human brain, it is impossible to determine if a stimulus is a line or an edge (Shams & von der Malsburg, 2002). Moreover, computational modelling has shown that biological neurons that respond to single lines or bars are preceded by simple and complex cells (Petkov & Kruizinga, 1997). In a similar vein, we only see line neurons after the appearance of complex Gabor neurons.

Figure 3. Neurons in the third layer of InceptionV1. From left to right, a colour-contrast neuron (21% of all the neurons), line neuron (17%), shited line neuron (8%), texture neuron (8%), colour centre-surround neuron (7%), tiny curves neuron (6%) and a texture contrast neuron (4%). Images are adapted from Olah et al. (2020b) under Creative Commons Attribution CC-BY 4.0.

In addition to colour-contrast detectors, we find colour centre-surround neurons in the third layer. These neurons are sensitive to one colour in the middle of the receptive field and another on the boundary. Similar centre-surround mechanisms can be found in the brain (Sceniak, Hawken & Shapley, 2002). Finally, the layer includes neurons that respond to textures. Textures are repeating structures and can be summarized by a set of statistics (Portilla & Simoncelli, 2000). V2 neurons, but not V1 neurons, are responsive to this statistical information of textures (Ziemba, Freeman, Movshon & Simoncelli, 2016). Moreover, Okazawa, Tajima and Komatsu (2015) showed that V4 neurons respond best to particular textures derived from sparse combinations of known higher-order image statistics. Likewise, DCNNs can reconstruct textures based on extracted statistics from earlier layers (Gatys, Ecker & Bethge 2015).

The fourth layer contains even more complex and diverse neurons (Fig. 4, Olah et al. 2020b). Apart from normal line detectors, there are line-ending, curve, angle (forming triangles and squares),

diverging-line and circle detectors. Based on only a handful of curve neurons, Cammarata et al. (2020) conducted a detailed study on how these curve detectors arise in DCNNs. They found that the curve detectors have sparse activations, responding only to 10% of the curves while the curves span all orientations. Cammarata et al. found that by creating numerous tuning curves based on a wide variety of stimuli, curve detectors respond to a wider range of orientations in curves with higher curvature since curves with more curvature contain more orientations. A perfect curve activation is up to 24 standard deviations higher than the average of the dataset. Moreover, the curve detectors are generally invariant to other features (e.g. colour) and fire moderately when an angle aligns with the tangent of the curve. By making use of the properties of these neurons, Cammarata et al. were able to create sophisticated curve tracing algorithms. These sets of experiments provide strong evidence that curve neurons genuinely detect a specific curve feature. Jiang, Li and Tang (2019) found V4 neurons that respond to curves and corners in both natural and synthetic stimuli. Using clustering techniques the authors found dimensions that represent straight lines, curves and corners separately. Moreover, the preferred natural stimuli of those clusters all contained the features these dimensions putatively encode. Similar to the artificial curve detectors, the tuning curves of biological neurons were sparse and clearly preferred specific curvature orientations, while they weakly responded to slight variations in orientation and curvature.

Besides these shape features, the fourth layer again contains colour-centre surround units, albeit the neurons are more complex, e.g. some detect textures in the middle and colours at the boundaries (Olah et al. 2020b). One-fourth of the neurons are texture neurons that look for simple repeating local structure over a wide receptive field[4]. Many of those neurons come from a maxpool followed by a 1x1 convolution layer. Neurons in this branch have by definition a large receptive field but are unable to control where in their receptive field each feature they detect is, nor the relative position of these features. This property makes the neurons ideally suited for detecting textures and repeating patterns.

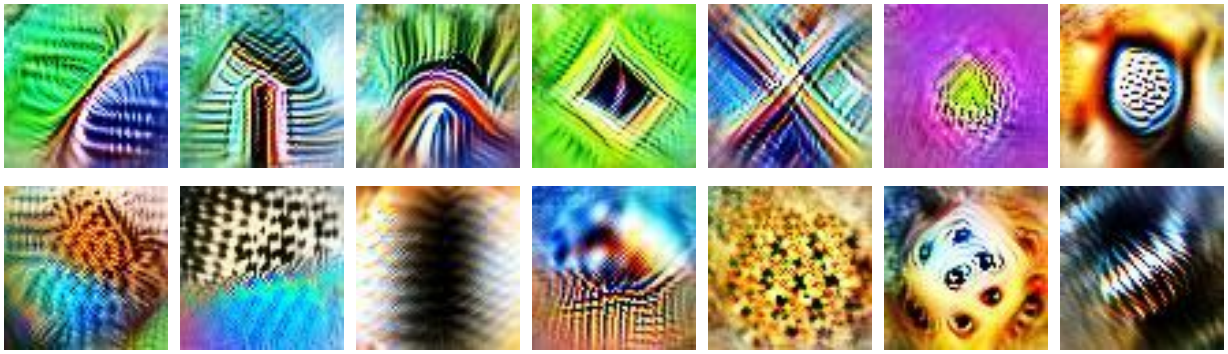


Figure 4. Neurons in the fourth layer of InceptionV1. From left to right, a line neuron (10% of the neurons), line ending neuron (1%), curve neuron (4%), angles neuron (3%), diverging line neuron (%), colour centre-surround neuron (12%), complex centre-surround neuron (5%), colour contrast neuron (5%), black and white vs colour neuron (4%), brightness gradient neuron

[4] There are multiple convolutional operations in this layer, so the receptive fields diverge between different neurons

(6%), high-low frequency neuron (6%), texture neuron (25%), repeating patterns neuron (5%) and an early fur neuron (3%). Images are adapted from Olah et al. (2020b) under Creative Commons Attribution CC-BY 4.0.

The fifth layer contains neurons that cannot be characterized as low-level features anymore (Fig. 5, Olah et al., 2020b). For example, neurons that detect boundaries of all sorts and are constructed from multiple types of neurons from the layer before can be found. The neurons are invariant to the features that change at the boundary. Moreover, this layer has increasingly complex curve detectors, including shapes such as spirals, divots and evolutes (curves facing away from the middle). Finally, we see neurons that can be characterized as (specifically orientated) fur detectors and neurons that seem to respond to detect head-like shapes or more specifically eyes. Due to the increasingly complex appearance of the features, there is little literature if and to what degree visual cortex neurons are tuned for these features. However, we can still draw parallels on a higher level. First of all, both DCNNs and the visual cortex have a clear distinction between shape and texture (Cant, Arnott & Goodale, 2009). Moreover, we observed that colour features in later layers are intertwined with other types of features such as shape. This intermingling of colour and shape is also observed in areas such as V4 posterior IT (Conway et al., 2010).

Figure 5. Neurons in the fifth layer of InceptionV1. From left to right (new types are included first), a boundary detecting neuron (8% of the neurons), proto-head detecting neuron (3%), generic-orientated fur neuron (2%), curve neuron (2%), divot neuron (2%), grid neuron (2%), eye neuron (1%), color center-surround neuron (16%), complex center-surround neuron (15%), texture neuron (3%), colour contrast/gradient neuron (5%), cross/corner divergence neuron (2%), pattern neuron (2%) and a curve-like shape neuron (2%). Images are adapted from Olah et al. (2020b) under Creative Commons Attribution CC-BY 4.0.

Most neurons classified by Olah et al. (2020b) show similar behaviour as neurons in the visual cortex. However, later layers also contain neurons with unexpected behaviour. These neurons apply a familiar structure in a new way. Examples are neurons that look for a colour/non-colour contrast, or centre-surround neurons that look for specific textures at the centre of their receptive field. Finally, there are multiple iterations of neurons responding to high-low frequency patterns on the opposing side of their receptive field. Later iterations use these patterns for the detection of boundaries. The behaviour of these neurons is less perplexing than at first glance. By looking at specific dataset examples we can gain intuitions about the functionality. For example, the behaviour of high/low-frequency neurons might

be related to the fact that the (ImageNet) objects are in focus while the background is out of focus. This property leads to an abrupt change in the frequency of the patterns and DCNNs use this property as a boundary detection mechanism. The discovery of certain response properties and mechanisms might inspire studies that explore if the same can be found in humans. If this is the case, then we might use DCNNs as a method to make inferences about the brain.

4.4. Learning high-level features in supervised deep convolutional neural networks

Intermediate and high-level features in both DCNNs and the visual cortex are less straightforward to study. In the case of the visual cortex, it is hard to find the exact response properties of neurons in high-order areas since the high-level features are constructed of lower-level features. Naturally, there are far more possible stimuli in the environment to perceive than we can test experimentally.

In later layers of DCNNs, we can find many neurons that are seemingly encoding for meaningful (i.e. features that correspond to a property of the input that is east to define), such as the parts that make up dogs, cars, faces, as well as their corresponding parts (Olah et al., 2020a). The variety of high-level features is inherently limited to the training dataset. There are for example many dog-related feature units since ImageNet contains 120 dog breeds. Even though a large part of the features are recognizable, these features are still noisy and imperfect. High-level features in DCNN are seemingly invariant to both position and orientation (Olah et al., 2020a). Invariance to orientation in DCNNs is likely the result of

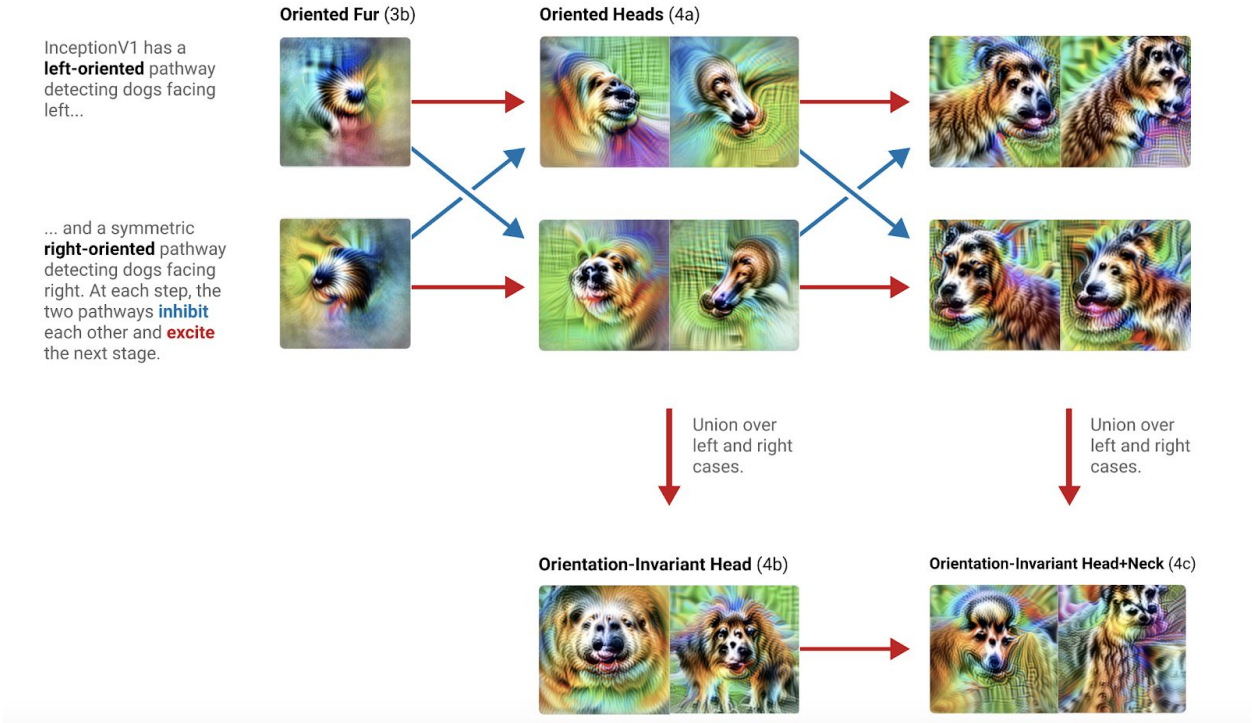


Figure 6. Dog head detecting circuit spanning over four layers. Through a series of steps, the DCNN learns to detect the head of the dog regardless of the orientation of the head and neck. The model separately detects two cases (left and right) and then merges them together to create invariance. Note that the model uses both excitation and inhibition to achieve this goal. Image adapted from Olah et al. (2020a) under Creative Commons Attribution CC-BY 4.0.

specific circuits spanning over multiple layers. Take for example the dog head detecting circuit spanning over four layers (see Fig. 6). Neurons looking for fur in a specific orientation are connected to neurons

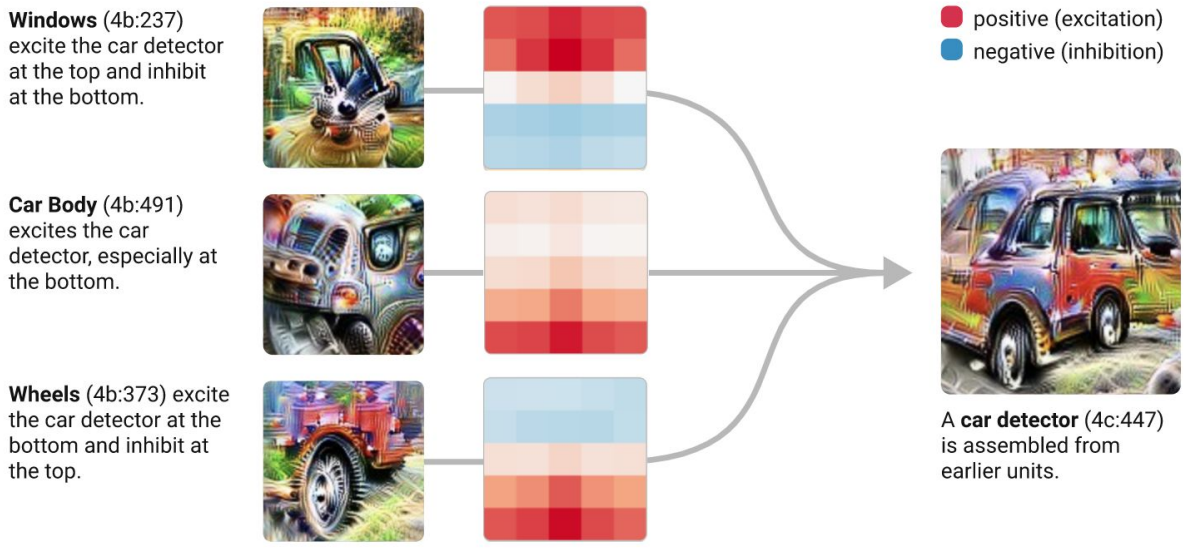


Figure 7. Car detecting circuit. This circuit assembles a car detector from individual parts in a specific spatial configuration. Image adapted from Olah et al. (2020a) under Creative Commons Attribution CC-BY 4.0.

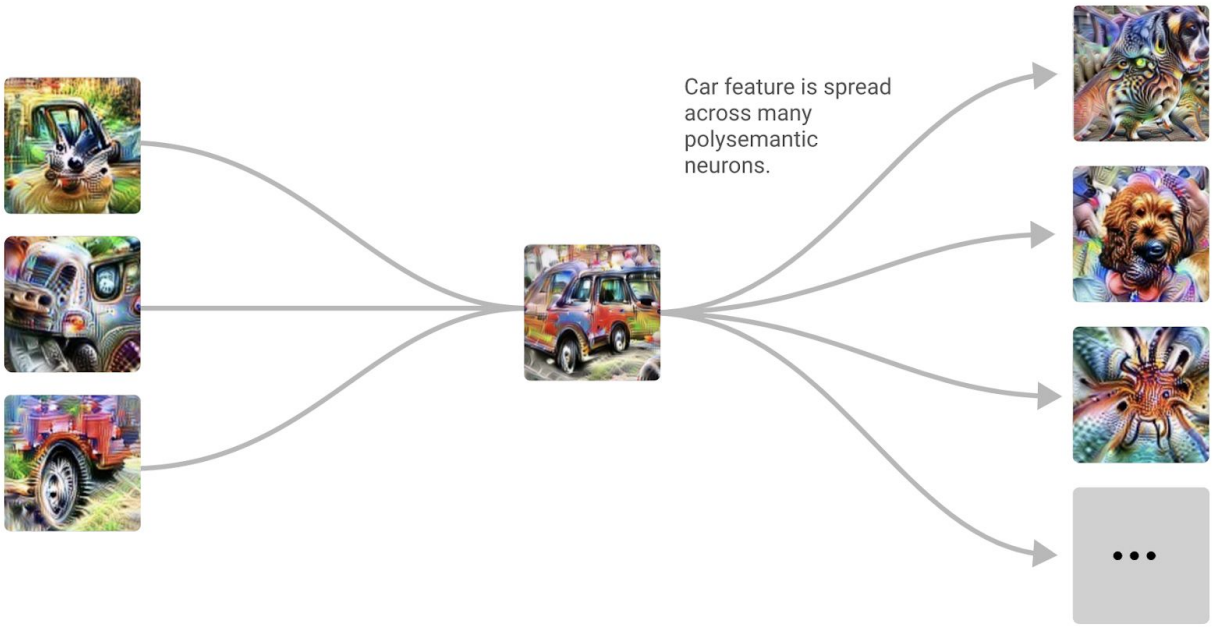
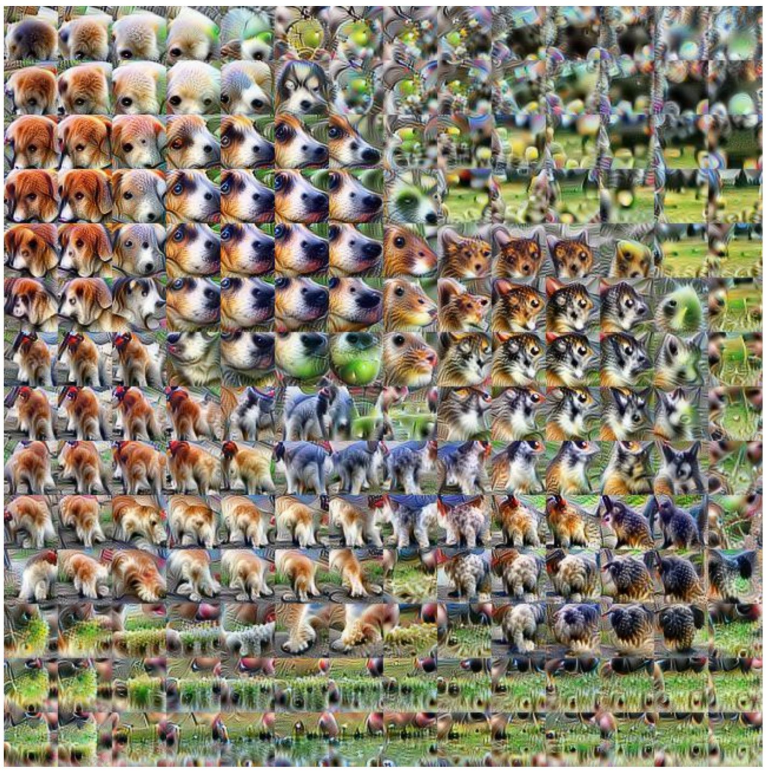


Figure 8. Polysemantic neurons. Neurons detecting seemingly meaningful features can influence many neurons that encode multiple features at the same time. Image adapted from Olah et al. (2020a) under Creative Commons Attribution CC-BY 4.0.

that look for dog heads orientated in a specific way. These oriented neurons are subsequently combined in the next layer to construct the dog head detecting neuron that is invariant for orientation. Importantly, the network could have chosen a different approach, for example, to just detect a mix of parts irrespective of their position. Fig. 7 shows another example of a feature with specific spatial relations.

a)

b)



c)

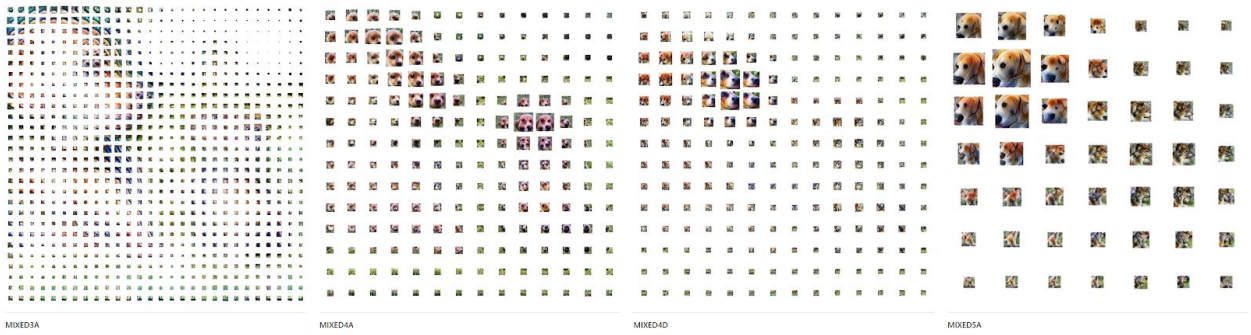


Figure 9. Combination of visualization and attribution techniques applied to an image. (*a*) The input image of a dog and a cat. (*b)* A grid containing the optimized image of a set of neurons that fire at that given spatial location. Each grid can be thought of as a visualization of what the model sees when looking at that area of the image. (*c*) The same technique as used in b applied over four layers but now the size of the grid is scaled in relation to the magnitude of the activations. The technique thus shows the importance of each part of the image. Images adapted from Olah et al. (2018) under Creative Commons Attribution CC-BY 4.0.

Here the car detecting neuron looks specifically for a window at the top of its receptive field, the car body in the middle and a wheel at the bottom. The deeper in the model, the harder it becomes to understand what a single neuron encodes for. The feature visualizations become increasingly complex, it is harder to specifically couple them to parts of objects or even objects at all. Many neurons are polysemantic, meaning they seemingly encode for a wide variety of features, often without any shared characteristics (see Fig. 8, Olah et al., 2020a). At this point, attribution becomes an important tool. With a combination of feature visualization and attribution, we can see what the network makes of a certain image (Olah et al., 2018). Instead of optimizing one specific neuron, we can optimize neurons that fire at a specific location in an image. By doing so, we more or less visualize what the network makes of the image. In Fig. 9 we see an image of a dog and a cat, you can see that at the position of ears, paws and the snout of a dog, the network sees those specific parts. This implies that the features are distributed over the network instead of a handful of neurons. DCNNs, both in terms of abilities and strategies. In the next chapter, the limitations will be discussed extensively.

a) **b)**

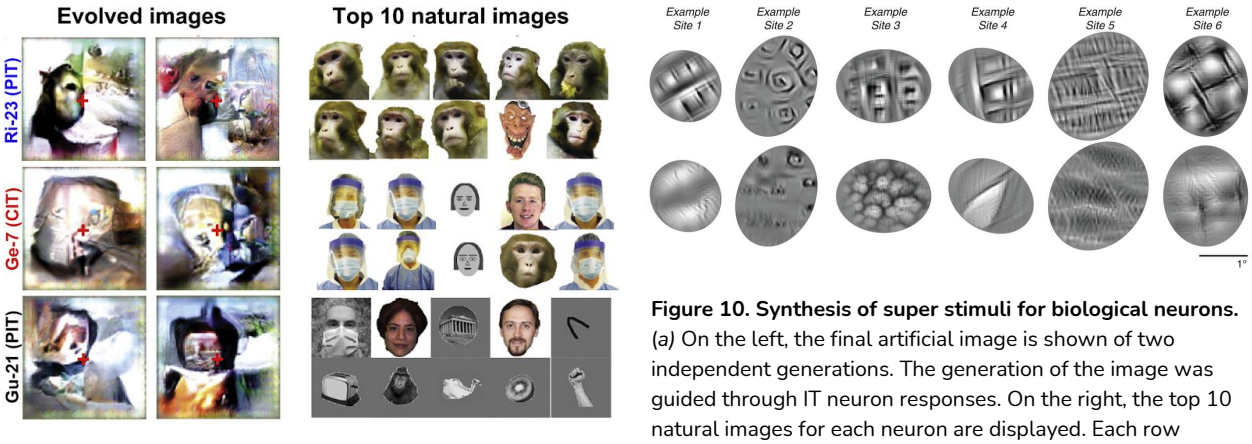


Figure 10. Synthesis of super stimuli for biological neurons. *(a)* On the left, the final artificial image is shown of two independent generations. The generation of the image was guided through IT neuron responses. On the right, the top 10 natural images for each neuron are displayed. Each row represents one IT Neuron. Adapted from Ponce et al. (2019) with permission. *(b)* Images generated for 6 V4 neural sites with two different optimization techniques. Image adapted from Bashivan et al. (2019) with permission.

Up until this point we have only discussed insofar DCNNs and the visual cortex seem to encode similar features. We have seen that DCNNs respond to meaningful features, however, many of the high-level features have an abstract (and often messy) appearance and most of the time seem to encode a wide variety of features. One question one might ask is if the response properties of individual neurons show similar behaviour. Inspired by artificial intelligence, neuroscientists now started to use adapted versions of the feature visualization techniques. These techniques use DCNNs to generate *super stimuli* for biological neurons. Ponce et al. (2019) used the responses of single IT neurons to generate artificial images from scratch. Most of the time, the IT neurons responded stronger to these artificial images than any of the natural images. The IT neurons evolved complex images containing many different features (see Fig. 10a). At times, the evolved images were hard to recognize. The resulting images are highly reminiscent of feature visualization techniques. The images are, just like in DCCNs, not easy to interpret, yet it maximizes the response in the IT. We thus seem to have polysemantic neurons in the IT as well. Moreover, it is plausible that objects are represented in a highly distributed manner (but see Higgins et al. 2020, later in this review). In a similar vein, Bashivan, Kar and DiCarlo (2019, see Fig. 10b) showed that V4 neurons and population of neurons can be activated beyond its naturally occurring maximum activation through the generation of synthetic images with a DCNN. These studies show that the response properties of biological neurons are closer to artificial neurons than previously thought which might indicate that the latter is a good model for the visual cortex.

5.5. Conclusion

In this chapter, we have seen that DCNNs show similar features as the visual cortex. While this is not direct evidence of similar computational mechanisms, it does show that DCNNs can be used as

tools to explore theories and hypotheses about the brain without the usual constraints. This chapter is thus above all a testament that the predictive power of DCNNs is not an accidental property. In the future, DCNNs could be used as a way of researching how certain architectural and learning constraints can give rise to behaviour (and eventually even provide an account for the computational mechanisms). That being said, the painted picture might give an overly rosy view on the subject. In the next chapter, the inherent limitations of supervised learning are discussed.

5. The Limitations of Supervised Deep Convolutional Neural Networks as Models for Object Recognition

5.1. Supervised deep convolutional neural networks performance is overestimated

DCNN performance in object recognition tasks is often described as equal or better than human performance (see for example He, Zhang, Ren & Sun, 2015). The claim is based upon the comparison of the performance of DCNNs and humans on the ImageNet database by Russakovsky et al. (2015). Here, the authors asked humans to classify an image by giving five labels out of 1000 total labels. The dataset consists of 120 breeds of dogs and many other sub-species of animals. It is not hard to imagine that such a task is quite difficult for humans. On the other hand, DCNNs are specifically trained on these 1000 categories with over a million images. The test set itself is derived from the same distribution as the training set (see the next paragraph why this is problematic). The comparison is thus highly biased towards DCNNs. When humans are trained on 40000 images (by annotating the images), all outperform state-of-the-art (SOTA) DCNNs on ImageNetV2 (Shankar et al., 2020). This is even the case with DCNNs are trained on an additional 1 billion (unlabeled) images with weak supervision. Moreover, there is an overestimation of how well DCNNs generalize to slightly more difficult images with similar distributions as the original training/test set (Recht, Roelofs, Schmidt & Shankar, 2019). These results indicate a stronger reliance on idiosyncratic features in the training set.

5.2. Adversarial features in supervised deep convolutional neural networks

In the previous chapter, we saw clear evidence for the similarity of the features in DCNNs and the visual cortex. Visualization techniques, however, miss features that are distributed over many neurons yet still heavily influence the classification process. *Adversarial examples* reveal features that are present in the training set but not in the real world. Adversarial examples are formed through gradient-based methods and apply small but intentionally worst-case perturbations to an image such that the model gives an incorrect answer with high confidence (Goodfellow et al., 2014). Most of the

time these perturbations are invisible for humans and in none of the cases humans would be fooled by the adversarial examples. Even one-pixel attacks can change the classification outcome of DCNNs (Su, Vargas & Sakurai, 2019). Adversarial examples are not bugs as they are present in all DCNNs. Rather, adversarial examples should be seen as non-robust features. These features are derived from patterns in the data distribution that are highly predictive, yet brittle or non-existing in real life and thus incomprehensible to humans (Ilyas et al., 2017). When trained on the same dataset, an adversarial example specifically designed for one DCNN can often fool other DCNNs. The non-robust features can be attributed to surface statistics in the training dataset (Jo & Bengio, 2017). Adversarial features are related to overfitting. Even though one aims to stop training before overtraining occurs, there is always some overfitting, DCNNs inherently perform rote memorization on the training set. DCNNs can even fit all the training data when the labels are shuffled randomly (Zhang et al., 2016). Even though we would not be able to see these traces with the help of feature visualization and attribution, they are definitely there. To date, there is no solution to bypass the adversarial problem

These findings point to a fundamental difference between the input data of DCNNs and the brain. If biological vision could reliably exploit these surface statistics it would have evolved to do so. Of course, in real life, these surface statistics are not present. It is still unclear if these problems are inherent to the training set or that a different architecture and learning method will avoid adversarial features despite their presence in the training data.

5.3. Supervised deep convolutional neural networks lack robustness

Even if DCNNs use similar features as humans, this does not imply that the features are as robust. The training and test images of DCNNs are usually clearly visible. If the object is occluded, performance sharply decreases for DCNNs, while humans are only barely affected (Zhu, Tang, Park, Park & Yuille 2019). Likewise, DCNNs were not robust to images with poor visibility, while humans effortlessly recognize images, even when the image visibility is reduced to under 15%. Moreover, DCNNs are not robust against noise and other image perturbations (Geirhos et al., 2018a; Geirhos et al., 2018b). DCNNs can be trained to deal with one type of noise, however, this will not generalize to other types of noise. In fact, training on one noise type often lowers performance on other noise types. Given that there are limitless real-life distortions, it is not feasible to train DCNNs on all types. Moreover, DCNNs make high confidence predictions of unrecognizable images and a slight change in position of the object can abruptly change detection of the object and other objects (Nguyen, Yosinski & Clune 2015; Rosenfeld, Zemel & Tsotsos, 2018).

The use of convolutional architecture is motivated by the desire to make networks invariant to irrelevant cues such as image translations, scalings, and other small deformations. In practice, this appears not to be the case. Even a shift of one pixel can have dramatic effects on the outcome (Azulay &

Weiss, 2019). Furthermore, the use of data augmentation does not solve this problem. The root of the problem might lie in the convolutional architecture itself. If there is no subsampling, each translation in the input image results in a translation of the features. The network is thus *invariant*. However, with downsampling methods, such as max-pooling, strided-convolution, and average pooling, the subsampling factor of the network becomes large (Zhang, 2019). A small translation in the image might now be missed by a feature detector. For example, subsampling by a factor of two causes the feature detector to only fire when the feature is centred on an even pixel in the receptive field but not on an odd pixel (Azulay & Weiss, 2019). Data augmentation helps to a degree, but only for images that are highly similar to the training dataset. Unfortunately, the training data is usually highly biased and the network might not be robust when tested on different data. In the next paragraph, I discuss the lack of robustness in relation to learning.

5.4. Supervised deep convolutional neural networks use shortcut learning

The capability of two systems to recognize objects as accurately and encode similar features does not imply that the strategies are similar, e.g. one system might put more emphasis on low-level features despite the capability to encode other types of features (e.g. surface statistics). The strategies DCNNs learn and employ might be fundamentally different from humans. We can explore strategies by looking at classification errors. Geirhos et al. (2020) looked at the trial-to-trial consistency of errors. Two systems should systematically make the same errors on the same input if the decision strategies are similar. The authors found that irrespective of architecture, DCNNs are highly consistent with one another. However, the consistency of errors between DCNNs and humans is only slightly better than what could be expected from chance alone. Another piece of evidence for diverging strategies is that simply using the DCNN low-level features in a new model can already result in high accuracies. In comparison to DCNNs, the model consisting of low-level features behaves similarly in terms of feature sensitivity, error distribution and interactions between image parts (Brendel & Bethge, 2019).

DCNN might make use of shortcut learning (Geirhos et al., 2020). Shortcuts are decision rules that work well on the training data but fail to transfer to other datasets with different distributions. Humans do not (always) have similar shortcuts, since the input data is fundamentally different. This is a side effect of gradient-based learning; the model will simply go in the direction that minimizes the error. Moreover, we cannot expect that DCNNs will consistently learn certain features. When two features both predict a label, models preferentially learn only one of the features. Easy features might suppress harder features, even if the latter one is more predictive (Hermann & Lampinen, 2020). In some situations, we do see similar shortcuts between DCNNs and humans, such as the use of background information to predict object class. Both DCNNs and humans exploit this relationship between object and background (Seijdel et al., 2020a).

5.5. *Looking on the bright side: Understanding through predictions*

Using the predictive framework to evaluate DCNNs as computational models for the brain can yield understanding at the same time. While searching for what features drive DCNNs (and humans) we can use the predictions to form explanations on a lower-level of what drives these differences. An example of how this can work is the debate on shape perceptions in DCNNs. Humans can effortlessly use shape for object recognition, for example, an object is easily recognized by its silhouette despite the absence of other features. Initially, it was thought that DCNNs used shape in a similar manner, for example, Kublius et al. (2016) showed that shape representations in DCNNs are similar to human shape representations. However, the study did not disentangle local and global shape. Later studies showed that DCNNs also have the tendency to use texture, rather than shape, for object classification (Geirhos et al., 2018a). When using training data that enforces the use of shape, DCNNs prioritized the use of local shape. More recent studies experimentally orthogonalized local and global shape and convincingly showed that DCNNs use local but not global shape (Baker, Lu, Erlikhman & Kellman, 2018; Baker, Lu, Erlikhman & Kellman, 2020). At the same time, results from a study conducted by my colleagues and I showed that depth in DCNNs enables models to selectively focus on features belonging to the object while ignoring features in the background (Seijdel, Oosterholt & Scholte, currently in preparation). These results suggest that DCNNs have some implicit mechanisms that segment the object from the background. DCNNs are thus not simply a bag-of-local features since many of these features are also present in the rest of the scene. Arguably, the behaviour of DCNNs shows signs of grouping a set of features in close proximity together despite the fact that they most likely do not use global shape to group features together. These studies show that through manipulation of training data and architecture (e.g. model depth) a predictive framework can yield a better understanding (albeit on a lower-level) of both DCNNs and humans.

5.6. *Architectural constraints*

As we saw in the previous paragraph, intuitions gathered from exploration and predictions might point to architectural constraints. Hinton, the godfather of deep learning, argues that DCNNs are not capable of learning global shape and spatial relations due to the pooling operations. Hinton hypothesizes that the brain has something unique, namely capsules, which encode things like position, size and orientation. Importantly, these capsules not only contain what type of features it detects but also its pose. Based on this hypothesis, Hinton and colleagues (Sabour, Frosst, N., & Hinton, 2017) proposed Capsule Neural Networks (CapsNet), the network contains capsules with groups of neurons representing features and uses an iterative routing-by-agreement mechanism. Low-level capsules predict the activity of high-level capsules via recurrent connections. When the predictions agree, the

corresponding high-level capsule is activated (Hinton, Sabour & Frosst, 2018). Recent studies have shown that CapsNets are capable of global shape perception, and even suffer from crowding (Doerig, Schmittwilken, Manassi & Herzog, 2019). This is strong evidence that architecture plays an important role in exploiting global shape.

Searching for which learning method provides the best computational model is futile if learning is hindered by architectural constraints. We thus should ask ourselves if there is good reason to think that architecture is the most important component for creating a model of object recognition. Many neuroscientists (see for example Kubilius et al., 2018) have argued that recurrent connections are necessary to improve the match between DCNNs and the visual cortex. Recurrent feedback might help create more robust features as in humans recurrent feedback facilitates object recognition under challenging conditions (Wyatte, Jilk & O'Reilly, 2014). Recurrent feedback only occurs in the later stages of visual processing (Lamme & Roelfsma, 2000). We can indeed see that the predictive power of DCNNs decreases over time. For example, DCNNs ability to predict response patterns in IT worsens over time (Kar, Kubilius, Schmidt, Issa & DiCarlo., 2019). Moreover, when the higher-order ventral prefrontal cortex is silenced the predictive power of DCNNs increases (Kar & DiCarlo, 2020). Recurrent DCNNs might thus provide a better model of the neural temporal dynamics of object recognition. Unfortunately, the search for such architecture remains challenging. Current implementations (e.g. CORnet by Kubilius et al., 2018) do not provide a better fit to neural data and the gap in strategies between recurrent DCNNs and humans remains unaltered (Geirhos et al., 2020). Even though these implementations contain recurrent connections it has been argued that these models are effectively feedforward when unrolled along time (van Bergen & Kriegeskorte, 2020). Both types can do the same computations, however, recurrent connections might do things more efficiently and effectively. It, therefore, seems a must to research recurrent computation in DCNNs if we want to model the brain with all its constraints. However, if we were predominantly interested in modelling the behaviour itself recurrent connections would only be necessary if they can do something feedforward connections cannot do.

Conversely, neuroscientists are working on improving the architecture of feedforward DCNNs to bring behaviour closer to humans. For example, Dapello et al. (2020) added a module, based on classical neuroscientific models of V1, to preprocess the input for DCNNs to create robust features. The addition made the DCNN more robust, outperforming the base encoders (18%) and SOTA methods (3%) on both images with common adversarial based perturbations. Moreover, Reddy, Banburski, Pant and Poggio (2020) improved the robustness of neural networks to small adversarial perturbations by adding two biological inspired retina based mechanisms to improve translation invariance in DCNNs. While these results are promising the gains are still insufficient to bridge the gap to humans.

5.7 Conclusion

Supervised DCNNs perform worse than often is reported. Beside useful features, supervised models learn features that are only present in the training data and hurt performance when tested on another distribution. In addition, the performance is far less robust than with humans, especially under challenging conditions. While changes to the architecture might provide us with small gains, this does not address the inherent limitations to supervised learning. Removing all these shortcuts is unfeasible and after all, we want the model to learn under noisy circumstances as well. All these problems are not solved by SOTA supervised models that use hundreds of millions of images in combination with weak supervision. In the previous chapter, we saw that DCNNs can learn meaningful features. We thus know that DCNNs have the inherent capabilities to learn features, but supervised learning results in unwanted byproducts. This, in combination with the fact that humans perform object recognition with a limited number of labels, argues for a different approach. A much better approach would be to find a learning method that learns the inherent structure of the world. In the next chapter, we explore these learning methods as a computational model for object recognition.

6. Self-Supervised and Reinforcement Learning in Deep Convolutional Neural Networks

Humans do not receive millions of labels for the objects they perceive during their lifetimes. Learning to make sense of the world is largely done without external supervision. In the field of neuroscience, the use of DCNNs is mostly limited to object recognition, yet vision is much more than recognizing objects. Features in the brain are likely task-agnostic and applicable to a wide variety of downstream tasks. The input we receive is much richer in structure than the labels themselves. Not surprisingly, learning without external supervision has been touted as the next step in deep learning (LeCun et al., 2015).

6.1. Contrastive learning

Contrastive learning methods have only just recently been used as computational models for object recognition, hence, there are only a few studies done on this subject. Nonetheless, the self-supervised learning methods already reach levels of the predictive power of their supervised counterparts. Zhuang et al. (2020) compared various contrastive learning methods with supervised methods and found that the self-supervised models were either on par or even better at predicting neural data of V1, V4 and IT. Moreover, the mapping of the layers is consistent with the hierarchy of the visual cortex.

Konkle & Alvarez (2020) created their own contrastive learning method, instance-prototype contrastive learning (IPCL) combined with an AlexNet base encoder and compared it to neural data. IPCL is based on Wusnet (Wu, Xiong, Yu & Lin 2018), with a few biologically-inspired modifications. Even though the top-1 ImageNet score of IPCL is significantly lower than the SOTA, IPCL and Wusnet achieved the highest general correspondence with the visual system hierarchy of all tested supervised and self-supervised models. The authors tested AlexNet with both group and batch norm layers and found that group layers resulted in the highest correlations. These results are somewhat surprising since AlexNet is not among the best performing DCNNs in terms of Brain-Score (Schrimpf et al., 2018). Interestingly, some (but not all) supervised models showed a higher correlation with higher-order visual areas in comparison to their self-supervised counterparts[5], indicating that there is a more category-like shift in the representational structure of this region. A possible explanation of these observations is that the cost function of those areas is different and more akin to supervised learning (Marblestone et al., 2016).

Contrastive learning can take advantage of noisy data arising from real (developmental) datastreams to learn strong features. Zhuang et al. (2020) trained a contrastive learning algorithm, Video Instance Embedding (VIE), on SAYcam (Sullivan, Mei, Perfors, Wojcik & Frank, 2020), which contains head-mounted video camera data from children spanning ages 6-32 months. VIE learns feature embeddings that identify and group similar features of videos together while pushing different features apart. While the amount of data in SAYcam is considerably less, the temporally aware VIE approached the predictivity of those trained on ImageNet.

Since self-supervised learning methods can be used with any type of DCNN encoder, we can tweak the DCNN architecture to make it more biologically realistic. Parthasarathy & Simoncelli (2020) created a model with oriented linear filters (corresponding to V1) and convolutional filters (corresponding to V2). Each stage had two types of nonlinearities, corresponding to simple and complex cells. The authors used a contrastive learning objective to maximize the distance between the distribution of V2 responses to individual images and the distribution of responses across all images. When evaluated on texture classification, the model was more data-efficient than a variety of supervised DCNNs while exhibiting stronger representational similarity to texture responses of populations recorded in V2.

SOTA contrastive learning training methods show less behavioural consistency on object level with humans in comparison to supervised learning methods (Zhuang et al., 2020). However, accuracy cannot distinguish between two strategies, since similar accuracy can be achieved by diverging strategies (Geirhos et al., 2020). In order to draw conclusions, the trial-to-trial consistency of errors must be compared, which has not yet been done.

[5]Self-supervised AlexNet was still the best model. ResNet-18 and CORnet-Z showed a higher correlation when trained in a supervised fashion.

In previous chapters we've seen that object recognition in DCNNs is not as robust as in humans, simple image translations and distortions can lead to abrupt changes in behaviour. Contrastive learning methods, on the other hand, are more robust against adversarial attacks, common noise and other image perturbations (Kim, Tack & Hwang, 2020; Hendrycks, Mazeika, Kadavath & Song, 2019). Adding a self-supervised objective term to supervised learning methods can also increase robustness. Augmentation is used for both supervised and self-supervised learning, however, in the case of supervised learning the model has no information about the identity of the augmentations. Hernández-García, König and Kietzmann (2019) included this information in the model and added a self-supervised term to the supervised cost function. Additionally, they modified the loss function to maximize the similarity between the activations of the augmented images. The resulting model showed increased invariance while retaining performance.

6.2. Adversarial learning

Object vision is more than just categorizing objects. Supervised models might miss important, non-categorical information, such as object positions and sizes. This information is encoded in high-order visual areas such as V4 and IT (Hong, Yamins, Majaj & DiCarlo, 2016). Generative models have to encode non-categorical information in order to be able to reconstruct images. Christensen & Zylberberg (2020) found that generative models trained with a classify-and-reconstruct objective indeed provide a better match to the representation of non-categorical information in V4 and IT neurons. Moreover, the generative model was more robust to noise than its supervised counterpart. Generative models might also be a better model for the feedback processes in the ventral stream. Al-Tahan & Mohsenzadeh (2020) showed that the representational dynamics of the initial feedforward sweep is best modelled by the discriminator of a generative model. Likewise, the representational dynamics of feedback activity is best modelled by the generator. Both studies show the potential of generative models to model a part of object vision which is missed by supervised (discriminative) models.

Another demonstration of the usefulness of generative models as computational models is directly investigating how features are encoded in ensembles and single neurons. Based on the recent successes of supervised DCNNs, scientists have suggested that high-level features are encoded in a distributed manner in high-order areas such as IT (see for example Eichenbaum, 2018). Recent work by Higgins et al. (2020) suggests that this is not necessarily the case as the authors used an unsupervised generative model to disentangle neural data into interpretable latent factors, such as gender or hair length. The discovered factors showed remarkable correspondence with single IT neurons. Moreover, face images could be reconstructed using the signals from just a handful of neurons.

As discussed before, models emphasize certain aspects of the world, while they hide others (Cichy & Kaiser, 2019). In particular, the recent generative models show how beneficial it can be to study

a phenomenon with a different breed of models. The generative models are capable of addressing specific aspects of the phenomenon which was not possible with supervised models. If we only used one type of models we would not see these aspects and subsequently, it would not be incorporated in our theories and hypotheses.

6.3. Biological learning rules

Even though self-supervised learning is more biologically plausible than supervised learning, the implementation of the learning rule is still implausible. In both cases, backpropagation is used in an end-to-end fashion, training losses are computed at the top layer and weight updates are computed on gradients flowing from top to bottom. Recently Xiong, Ren & Urtasun (2020) demonstrated that local learning can match end-to-end contrastive learning methods for the first time. Likewise, Kunin et al. (2020) created non-local learning rules through biologically-plausible weight estimation and showed that these learning rules matched backpropagation. Moreover, the neural predictivity of these learning rules was equal to backpropagation, indicating that the learning rule is independent of the features. Finally, backpropagation can be approximated within a model based upon the predictive coding framework while requiring only simple local Hebbian plasticity (Whittington & Bogacz, 2017). This is a convenient outcome for neuroscience. When building both supervised and self-supervised models, neuroscientists can use whichever training rule is the most efficient. Of course, when a promising computational model is identified, it is good practice to show that the model performs equally well with biologically plausible learning rules.

6.4. Reinforcement learning

Reinforcement learning arguably provides an alternative for supervised learning. While we humans receive a limited number of labels during our lifetime, for a large part of our evolutionary trajectory we did not possess any capabilities to communicate about the identity of objects. Moreover, even if we receive such labels, the updates have to take place locally. Reinforcement learning provides a framework for learning without external labels. Instead, rewards are used to adapt to the environment. For example, when encountering an animal of an unknown species, interactions with the animals provide feedback in form of rewards and punishments. Through the feedback, we learn how to identify and behave towards new instances of the species.

Cost functions that optimize for rewards (and avoidance of penalties) are ubiquitous in the brain (Wang et al., 2018). In the deep learning framework, reinforcement learning is used to learn agents which actions it should take to maximize its cumulative reward. With the help of reinforcement learning, researchers are capable of creating agents with superhuman performance level on a predefined scope (e.g. Google's *AlphaZero* (Silver et al., 2018) won all its matches against top-ranking players in the

game of *Go*). These developments show the power of learning rules based on reinforcement and show that it can create flexible and intelligent behaviour. Recent studies show that backpropagation can be implemented through reinforcement learning. For example, neural networks with biologically plausible implementations of dendrites that can use error-driven synaptic plasticity to approximate backpropagation (Sacramento, Costa Bengio & Senn, 2018). Moreover, CNNs can be trained to recognize objects with reinforcement learning (Pozzi, Bohté & Roelfsema., 2018).

6.5. Predictive self-supervised learning

While contrastive learning methods pre-train the models through a prediction task this approach is not how the brain implements self-supervised learning. For humans, different views can be seen as variations in the continuous stream of visual information. Importantly, the input is sequential. Each new view is highly correlated with the previous one. For example, moving our head sideways results in a new view and thus different appearance of objects. The brain is hypothesized to use predictive coding to predict changes in the sensory input (Rao & Ballard, 1999). Predictions are derived through an internal model of the world and previous events (e.g. when seeing a train passing by we can expect that the train continually moves in the same direction at the same speed). Future predictive self-supervised models could use a similar approach in which previous frames and the following changes in the visual input (e.g. through movement) are used to predict the next frame. Currently, there already are models that implement predictive coding with a more biologically plausible architecture. The best performing model is PredNet (Lotter, Kreiman & Cox, 2016). PredNet is trained to predict future video frames. Just like DCNNs, PredNet has a hierarchical (recurrent) architecture with convolutional layers. Each layer makes local predictions of its input. The prediction error (the difference between actual input and the prediction) is passed to the next layer. While PredNet is capable of predicting neural data (Lotter, Kreiman & Cox, 2020), it lacks behind supervised and self-supervised DCNNs in both performance and neural predictive power (Zhuang et al., 2020). Moreover, PredNet has problems scaling up to real-world data (Rane, Szügyi, Saxena, Ofner & Stober, 2020). A plausible reason why PredNet is trailing behind DCNNs is that the model only contains four layers. Zhuang et al. attempted to add additional layers but subsequently failed to properly train the model.

6.6. Conclusion

Self-supervised learning addresses the limitations of supervised learning insofar the limitations are inherent to learning and not architecture. Current techniques show similar predictive power for neural data on object recognition tasks despite the fact that the models were not initially trained for the task. Self-supervised learning arguably learns features are more robust and importantly applicable to many downstream tasks. Self-supervised is clearly the more biological plausible form of learning due to

the way it learns from the visual input. While conceptually the learning method is most likely similar to the brain, the current cost functions and specific operationalizations (e.g. of views/augmentations) are likely implemented in a different manner. There is still considerable work to be done to bring the two closer. That being said, the diversity in cost functions of self-supervised learning allows neuroscientists to research how certain objectives and learning rules can bring about behaviour.

7. Discussion

7.1 Why supervised learning is successful far from special

Despite some of the biologically implausible ways of learning, supervised models still learn meaningful features and can predict neural data better than any non-convolutional computational model. On the other hand, we see that self-supervised models can achieve the same results. This begs the question, how special supervised learning is in the first place? I think the answer lies in a couple of simple observations. In chapter 4 we saw the striking resemblance of low-level features in DCNNs and humans. Moreover, we know that the same low-level features can be found in every other DCNN variation regardless of the training set. At the risk of sounding redundant, all high-level features are made from a much smaller set of low-level features. Since categories share all these low-level features, these low-level features are not discriminative by themselves. Finally, the study of Higgins et al (2020) might provide a clue as the authors showed that IT encodes features in a low-dimensional fashion. This implies that even if supervised models predict IT activity, the contribution of discriminative high-level features might be limited to the creation of idiosyncratic feature combinations. These discriminative features are likely restricted to the final convolutional layers. The bulk of the features will be learned to support this trick and this is of course done with gradient descent and backpropagation. In fact, the significant most breakthroughs in computer vision have little to do with supervised learning. Arguably, the low-level features that are related to supervised learning are precisely the features we do not want (e.g. surface statistics). After AlexNet, DCNNs mainly improved because researchers found ways to bypass the vanishing gradient problem (e.g. ResNet, He et al., 2016). In sum, the convolutional architecture, gradient descent and backpropagation are the key components to DCNNs success. Supervised learning might have been hired in the past—but that does not mean its expendable.

7.2. Which learning method we should use for modelling object recognition?

For the sake of the argument, let's say that supervised and self-supervised DCNNs have equal predictive power as models for object recognition. Should we favour one above the other?[6] In my view,

[6]

self-supervised learning should be favoured, even in the context of object recognition. Of course, once tested on object recognition we have to use some form of supervision, but self-supervised models have the benefit of being trained beforehand. As a result, self-supervised models perform well on just a limited number of labels which is more similar to how humans learn (see Table 1 in Chapter 3). Therefore, self-supervised models are better models for object recognition learning. Moreover, self-supervised models are less vulnerable to surface statistics in comparison to supervised models since the cost functions are designed to learn the structure of the world and not whichever aspect of an image can help with the discrimination of categories. We indeed see that self-supervised models are more robust to common image perturbations and adversarial attacks. Moreover, the performance of self-supervised is still rapidly increasing, whereas the performance of supervised models plateaued.[7] Self-supervised models also benefit more from additional training. Although it should be noted that improved performance is no guarantee for improved predictive power, supervised model's improvements saturate after reaching a certain performance level (Schrimpf et al., 2018).[8] Moreover, the self-supervised model with the highest score on the ImageNet benchmark is far from the most predictive computational model in the study of Konkle & Alvarez (2020). In terms of behaviour, we cannot make any firm conclusions since the trial-to-trial consistency of errors has not yet been evaluated. Moreover, behavioural consistency has not been researched in challenging conditions (e.g. noise, occlusion).

In sum, models that are pre-trained with self-supervised learning methods and subsequently fine-tuned with a limited number, are better computational models. These models have a higher degree of realism, especially when it comes to how the brain learns object recognition in the first place. In the next chapter, we discuss an arguably even stronger argument when it comes to the question of which learning method to use.

7.3. Going beyond object recognition

While supervised learning will only learn features necessary for the task, the features can be used for a wide variety of tasks due to its task agnostic learning method. Self-supervised learning can thus enable us to research other aspects of visual perception (however, there are some inherent limitations to DCNNs, see 6.6. on architecture). Humans can for example effortlessly answer questions such as *What size is the object? Where in the scene is the object located in relation to me? What is the relation between the object and the scene? What is the object made of?* Besides perceiving properties of objects, humans can perceive spatial relationships between objects and their motion. Moreover, humans have not only the ability to perceive but also to visualize images. In this review, we already saw that

[7] There is little improvement for models that are trained with only ImageNet data, SOTA increased from top-1 accuracy of 84.4 to 85.8 in the last two years. The bulk of the gains are related to inclusion of weak-supervision and hundreds of millions extra images.

[8] The ceiling appears to be around 70% top-1 accuracy.

adversarial training provided a better fit for non-categorical information. Also, previously discussed studies have shown that adversarial training provides a better fit for neural data acquired during visualization. In the future, I expect that more studies validate the predictive power of self-supervised models in a wide range of such downstream tasks. Going into the direction of self-supervised learning will automatically yield us computational models that can go beyond object recognition.

7.4. Why is self-supervised learning not (yet) more predictive than supervised learning?

Self-supervised learning is currently on par with supervised learning in terms of predictive power in object recognition tasks[9]. It should be noted that the benchmark is specifically designed to test supervised models. Moreover, in the discussed studies the test set was from the same distribution as the training dataset for supervised models while the self-supervised models were initially trained on a different dataset. Additionally, supervised models are limited to categorization tasks, while self-supervised models are not. The comparison is thus lopsided at the beginning.

Taking that into account, there are a few additional explanations why supervised and self-supervised models are on par with each other. First of all, self-supervised learning methods, especially contrastive learning, are still rapidly improving. Next, it is possible that self-supervised learning might perform better on different types of input data. Currently, all models are training on data that is fundamentally different from the input humans receive. Perhaps the use of input data that approximate the data during developmental trajectories will improve the predictive power of the self-supervised models. Finally, there is the possibility that object recognition is the driving force behind the visual system (Zhuang et al., 2020) Tasks constrained to categorization will automatically approximate the ventral stream.

The last explanation in my view is the unlikeliest, even if categorization is the most important task, from an energy-conserving perspective the features must be useful for other tasks. Also, categorization can occur without supervision; clustering a group of features together could automatically give rise to categories. Moreover, we do not need verbal labels to be able to adapt our behaviour to maximize the object's use. Arguably, the first and second explanations are much likelier. The field of contrastive learning is brand new, there is ample room for improvement. Moreover, there is more variability in cost functions. It is very likely that we have not found the best performing (biologically plausible) cost function. Besides cost functions, there is still an ongoing search for which data

[9] Current evidence shows that self-supervised models might already be better. Zhuang et al. (2020) found supervised and contrastive learning models to be equal in predictive power, while the contrastive learning method by Konkle & Alvarez (2020) outperformed both supervised and the self-supervised models that were tested in Zhuang and colleagues. Further studies are needed to draw firm conclusions and especially how and why certain contrastive cost functions perform better.

augmentation methods are the most effective in creating invariant features. Data augmentation takes a much more prominent role in self-supervised learning since it is a form of automatically creating labels. Neuroscientists do not necessarily have to search for augmentations that maximize performance, but rather maximize predictive power or biological realism. Lastly, neuroscientists are now able to use similar training input as humans receive. In sum, through careful selection of training data, cost functions and augmentations, it is likely that we will soon find better performing self-supervised models on object recognition tasks and beyond.

7.5. Encoder architecture

In this review, we have mainly focused on the differences between self-supervised and supervised learning. However, the underlying (encoder) architecture is identical. This gives ample opportunity to compare the two. For example, with the help of feature visualization, we can see if the learned features are similar. Nonetheless, there is still a lot of work to be done in terms of finding the optimal architecture for computational modelling using DCNNs. Arguably, the architecture of neural networks in the brain is far more complex than current DCNNs. Moreover, current DCNNs do not model the temporal dynamics of the brain. Other considerations are related to the lack of robustness of the features or even the inability to encode these features. Some of these problems might arise directly from the encoder architecture and not the training method per se. For example, DCNNs are incapable of encoding global shape. This discrepancy to humans highlights the shortcomings of the current encoder architecture. As discussed before, other architectures might be able to model different properties of object vision, for example, CapsNets can encode global shape (Doerig et al., 2019). However, the overall performance of these alternatives is still trailing behind DCNNs. Of course, in the brain, there are vast differences in how individual networks are wired, the architecture of individual units, and what type of computations the units performed. Therefore it is not unlikely that we need to move on to hybrid models with various sub-architectures and cost functions.

We have seen that adding recurrent connections to DCNNs has yet to yield more predictive power. This is not to say that this will not happen in the future. However, the implementation of these recurrent DCNNs is still limited. For example, CORnet only uses recurrent connections within an area and not between areas. A reasonable question is if recurrent connections are per se necessary for DCNNs. For example, it is possible that the depth of CNNs enables similar solutions as recurrent connections (Seijdel et al., 2020a). On the other hand, DCNNs still trail behind in performance in situations where the brain employs recurrent connections. Moreover, current implementations of DCNNs are by design poorly suited for modelling the temporal dynamics of the visual cortex, so from a neuroscientific perspective, it makes sense to search for ways to successfully implement recurrency in DCNNs.

7.6. Cost functions

It has been suggested that the brain makes use of a diverse set of cost functions across brain areas and development (Marblestone et al., 2016). These cost functions and the accommodating architectures are thought to deterministically build circuits and behaviours with limited input. Whereas supervised models have simple cost functions (usually cross-entropy loss), the cost functions of self-supervised learning are more diverse and richer. As discussed before, the goal of self-supervised learning is to reduce the dimensionality of the world while still preserving as much information as possible. This is for example done by contrasting dissimilar features and grouping similar. The mapping in the latent space should preserve the relationship between different features. Compare this to the objective of supervised learning, namely to discriminate between a set of categories. Even though this objective enables the models to learn a wide variety of features, the cost function does not explicitly tell the model to learn all mutual relations between features. Instead, it only requires what is important for the task. At the beginning of this chapter, I discussed how this could give rise to such a diverse set of meaningful features. At the same time, it is not hard to see how this cost function lacks when applied to other tasks or even other datasets. This is not to say that current self-supervised cost functions are without their own problems (discussed in 8.3.). Rather, I want to draw attention to the fact that self-supervised learning is potentially much richer since it aims to capture the structure of the world upon which we can then learn to perform tasks, either through supervised or reinforcement learning. Of course, it is still an open question how these cost functions will exactly look like. For neuroscience, the question is then what cost functions the brain uses, how the cost functions vary across brain layers and areas and how the brain optimizes these cost functions.

7.7. Synergy between neuroscience and artificial intelligence

The field of artificial intelligence is mainly focussed on one metric: performance. Whether or not the resulting solution is biologically plausible is irrelevant. Due to this approach, many neuroscientists are sceptical or even contemptuous towards DCNNs. This position would be valid if the resulting models were far away from biology. However, in reality, the models are far closer to biological models than one would have thought. This is no coincidence, first of all, the field of artificial intelligence likes to draw heavily upon biology, which should be no surprise since it is the most effective and efficient solution around. Nonetheless, it is not a question of copying whatever we think we know about the brain and then hand engineering (the features of) our own model. This approach was dominant in the ’90s and it turned out to be a dead end. These setbacks resulted in the realization that even the features themselves should be learned, a simple cost function proved to be much more powerful than whatever we could conjure up by hand. As extensively discussed here, this change led to features and behaviour

with remarkable similarity to our own brains. The focus on performance thus has a welcome side effect, the chance that the model will find a solution that is similar to the solution nature found is quite high. A reasonable approach to model the brain would be to first strive to implement a model, inspired but not completely constrained by biology, that mimics the performance of our brain. While the solution might be complex, the way this solution is learned might actually be quite simple[10]. If the learned solution then shows a great similarity between the brain, the next step is to take the model apart and try to make a certain part more biologically realistic while making sure the learned solution stays the same (or even better, improves). In sum, do not try to find the solution yourself, build a model that will do the work for you.

8. Open Questions and Future directions

While the hype of deep learning and its use in neuroscience is certainly not misplaced, there are still many questions unanswered and significant hurdles to overcome before we have computational models that explain object recognition and perception as a whole. In the next chapter, I shortly discuss the open questions in relation to the discussed literature.

8.1. Features in the brain and DCNNs

In this review, we have seen that DCNNs make use of seemingly meaningful features. The next logical step is to study these features and underlying computational mechanisms in greater detail to form specific theories and hypotheses on how they are implemented in the brain. For example, do neurons in the brain create similar circuits as neurons in DCNNs in order to extract features? Can we derive algorithmic implementations for these circuits and consequently test it on neural data? Are the features the driving force behind the predictive power? Are high-level features distributed over many neurons or do they represent features by themselves and/or in small groups? Can we remove low-level surface features from the model to improve the neural data fit?

On the other hand, we have to ask critical questions about the techniques developed to explore DCNNs, especially if we want to use those techniques to understand how features are extracted in the brain. For example, the optimization's objectives are quite simple, yet the resulting visualizations are complex. How do we properly interpret the result? What effect does adding different terms to the objective have—give the better-looking visualizations per definition a more clear (meaningful) answer? Are the visualizations truly meaningful to begin with—or are we looking at abstract visualization detached from its true function? How do we know which technique we should use in the first place? How do different techniques relate to each other? Since we mostly optimize for the optimal stimulus, are

[10] As an analogy, natural selection is a simple algorithm, yet it can yield extraordinary solutions.

we not missing parts of the neuron's behaviour (e.g. the use of low-order statistics)? There are no clear answers to these questions yet, but we can make an educated guess. It is likely that these techniques all cover some part of the story, but each technique on its own is insufficient to paint the full picture. To tell the whole story, a large palette of techniques should be used, each corroborating the theory a little.

8.2. Supervised and self-supervised learning

Most discussed self-supervised learning studies are only a few months old. Naturally, there are many open questions remaining. We still do not know how the self-supervised models relate to supervised models in terms of their predictive power of neural data. Self-supervised models should learn to encode non-categorical information since the models are not (pre)trained to categorize objects. Granted self-supervised models indeed process such information then we would expect that the models explain additional variance. Moreover, we should gather evidence that self-supervised learning does indeed provide features that can be used beyond object recognition tasks and verify that this improves the fit to neural data. Additionally, there is still much work to do in relation to how cost functions relate to the predictive power of self-supervised models and the functional organization of the visual cortex. For example, in the study of Konkle & Alvarez (2020), the most predictive model was significantly worse than the current SOTA—why is this the case? Moreover, when going beyond object recognition, which cost functions should be used on top of the self-supervised cost function? Or do we need multiple self-supervised cost functions?

8.3. Contrastive and adversarial learning

Both contrastive and adversarial learning learn the structure of the world through self-supervised learning, yet the cost functions and learning rules are not the same. The first question is then naturally, does the brain use only one of these cost functions, if any—or are the cost functions complementary and does the brain use them both for different purposes? Both cost functions might contribute to unique solutions, for example, the cost functions of generative models might be used for visualization and reconstruction of images. Or better, can we integrate both cost functions into one model and does this help to predict and explain behaviour to a further degree?

With respect to contrastive learning, we have to think about what type of “views” work best for training and or computational modelling. As for adversarial learning, we have to ask if, why and how it matters what type of data the models learn to generate. Apart from the learning method, we have to think about what data and architecture are optimal for learning.

In the next paragraph, I will discuss how we can build predictive self-supervised models.

8.4. Building predictive self-supervised models

Learning through prediction is potentially the way our brain makes sense of the world, we still do not know how this is done exactly. While predictive coding is a promising overarching framework, there is much uncertainty and disagreement about the details (e.g. Kwisthout & van Rooij, 2019). One of the most important challenges is to define what exactly is the prediction error, how it is used and computed. Moreover, the framework to be a fundamental organization principle of the brain (Bastos et al., 2012). There is still considerable doubt if current versions of the predictive coding framework are even computationally tractable, especially in relation to higher-order cognition (Kwisthout & van Rooij, 2019).

In light of the functional approach, I suggest that we first attempt to prove that models can indeed learn through prediction before adding all types of constraints such as the ones raised above. It would be beneficial to not be overly concerned with the mechanistic account predictive coding attempts to provide. In practice, it means that we should not worry what a prediction and prediction error precisely entails, after all, we are only interested in building a model that is functionally equivalent to the observed behaviour. We could train such a model in a 3D computer-generated world. The model could create different views by moving through the environment. We could implement such a model with both contrastive and adversarial learning. The contrastive learning variant could use a contrastive cost function by maximizing the agreement views as is done in Chen et al. (2020a). In addition, we could add an agent on top of the model that decides which views it chooses. This agent could use reinforcement learning to decide which views it should select (rewards are based on the contrastive cost function). A different variant could use adversarial learning where the generator generates the next frame and a discriminator tries to tell the actual and predicted frame apart.

To bring these self-supervised models even closer to us, we should search for methods of learning that are task agnostic and applicable over many tasks and modalities. Current self-supervised models are pre-trained on a task whereas babies and other animals simply explore the world before learning a task (LeCun, 2020). In addition, we want these models to learn models of the world by simply observing. For example, learning a model of gravity allows us to predict the behaviour of objects even when seeing the object for the first time. Moreover, we still have to understand how to learn to represent probability distributions and uncertainty for visual data. If we, for example, look at predicting future video frames, current models do not know how to deal with the countless possible future states. In practice, this means that the models merge these possibilities together resulting in blurry predictions. Lastly, there still is a considerable amount of work to be done regarding finding the right architecture. It is highly plausible that the brain is wired in such a way that learning can be done with little input data. To be clear, these pragmatic self-supervised models are not direct evidence for the predictive coding framework but only show the overall plausibility of the concept. Nevertheless, it would be likely that the solution found by these models would be close to nature.

9. Conclusion

DCNNs are simple, elegant—yet parameter rich—models for object recognition. Despite the simplicity, the supervised cost function leads to a rich inner world of seemingly meaningful features. Supervised models predict brain activity to an unprecedented detail and the learned features show a striking resemblance between the response properties of neurons in the visual cortex. However, there are clear limitations to supervised learning, the performance is far less robust in comparison to humans and the models use different input data and strategies. In addition, humans do most of their learning without supervision. Rather, the labels are self-derived through prediction and feedback is provided by the sensory signal itself. Recent breakthroughs in self-supervised learning now enable DCNNs to learn in a similar fashion. These self-supervised models show similar predictive power in object recognition tasks despite being trained in a task agnostic manner. For this reason, features are likely applicable to many downstream tasks and can help us to go beyond modelling object recognition. Given the richness of the self-supervised cost function, the models are poised to beat its supervised counterpart, both in overall performance and its ability to predict neural data. To quote Yann LeCun (2020), "*If intelligence is a cake, the bulk of the cake is self-supervised learning*[11]*, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning*". Even though there is plenty of reason for optimism, there are still many hurdles to overcome before we will have self-supervised models that learn and behave like us. Since creating and understanding the cognitive abilities of the mind go hand in hand, it is in the best interest of both the neuroscience and artificial intelligence community to closely collaborate to help each other reach the shared goal.

References

LeCun, Y. (2020, February 10). *AAAI 2020 Keynote by Yann LeCun: Self-supervised learning* [Video file] Retrieved from https://www.youtube.com/watch?v=UX8OubxsY8w&ab_channel=ICMLIJCAIECAI2018ConferenceVideos

Al-Tahan, H., & Mohsenzadeh, Y. (2020). Reconstructing feedback representations in ventral visual pathway with a generative adversarial autoencoder. *bioRxiv*.

Azulay, A., & Weiss, Y. (2018). Why do deep convolutional networks generalize so poorly to small image transformations?. *arXiv preprint arXiv:1805.12177*.

Baker, N., Lu, H., Erlikhman, G., & Kellman, P. J. (2018). Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, *14*(12), e1006613.

Baker, N., Lu, H., Erlikhman, G., & Kellman, P. J. (2020). Local features and global shape information in object classification by deep convolutional neural networks. *Vision Research*, *172*, 46-61.

Bashivan, P., Kar, K., & DiCarlo, J. J. (2019). Neural population control via deep image synthesis. *Science*, *364*(6439).

[11] LeCun (2020) updated version of the quote replaces unsupervised with self-supervised.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., & Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, *76*(4), 695-711.

Breedlove, J. L., St-Yves, G., Olman, C. A., & Naselaris, T. (2020). Generative Feedback Explains Distinct Brain Activity Codes for Seen and Mental Images. *Current Biology*.

Brendel, W., & Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*.

Cadena, S. A., Denfield, G. H., Walker, E. Y., Gatys, L. A., Tolias, A. S., Bethge, M., & Ecker, A. S. (2019). Deep convolutional models improve predictions of macaque V1 responses to natural images. *PLoS computational biology*, *15*(4), e1006897.

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., ... & DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput Biol*, *10*(12), e1003963.

Cant, J. S., Arnott, S. R., & Goodale, M. A. (2009). fMR-adaptation reveals separate processing regions for the perception of form and texture in the human ventral stream. *Experimental Brain Research*, *192*(3), 391-405.

Cammarata, N., Goh, G., Carter, S., Schubert, L., Petrov, M., & Olah, C. (2020c). Curve Detectors. *Distill*, *5*(6), e00024-003.

Chaudhary, A. (2020). The Illustrated SimCLR Framework [Web page]. Retrieved from https://amitness.com/2020/03/illustrated-simclr/

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. (2020b). Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Dhariwal, P., ... & Sutskever, I. (2020c). Generative Pretraining from Pixels. In *Proceedings of the 37th International Conference on Machine Learning*.

Christensen, E., & Zylberberg, J. (2020). Models of the ventral stream that categorize and visualize images. *BioRxiv*.

Cichy, R. M., & Kaiser, D. (2019). Deep neural networks as scientific models. Trends in cognitive sciences, 23(4), 305-317.

Cichy, R. M., Khosla, A., Pantazis, D., & Oliva, A. (2017). Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. *NeuroImage*, *153*, 346-358.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., & Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, *6*, 27755.

Connor, C. E., Brincat, S. L., & Pasupathy, A. (2007). Transformation of shape information in the ventral pathway. *Current opinion in neurobiology*, *17*(2), 140-147.

Conway, B. R., Chatterjee, S., Field, G. D., Horwitz, G. D., Johnson, E. N., Koida, K., & Mancuso, K. (2010). Advances in color science: from retina to behavior. *Journal of Neuroscience*, *30*(45), 14955-14963.

Cox, D. D. (2014). Do we understand high-level vision?. *Current opinion in neurobiology*, *25*, 187-193.

Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D. D., & DiCarlo, J. J. (2020). Simulating a primary visual cortex at the front of CNNs improves robustness to image perturbations. *BioRxiv*.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, *2*(7), 1160-1169.

Doerig, A., Schmittwilken, L., Manassi, M., & Herzog, M. H. (2019). Towards Global Recurrent Models of Visual Processing: Capsule Networks. In *Conference on Cognitive Computational Neuroscience, Submission ID* (Vol. 1066).

Eichenbaum, H. (2018). Barlow versus Hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition?. *Neuroscience letters*, *680*, 88-93.

Eickenberg, M., Gramfort, A., Varoquaux, G., & Thirion, B. (2017). Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, *152*, 184-194.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), pp. 193-202

Fukushima, K. (2007). Neocognitron. *Scholarpedia*, *2*(1), 1717.

Gatys, L., Ecker, A. S., & Bethge, M. (2015). Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems* (pp. 262-270).

Geirhos, R., Meding, K., & Wichmann, F. A. (2020). Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. *arXiv preprint arXiv:2006.16736*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2018a). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., & Wichmann, F. A. (2018b). Generalisation in humans and deep neural networks. In *Advances in neural information processing systems* (pp. 7538-7550).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

Graves, A., & Clancy, K. (2019). Unsupervised Learning: The Curious Pupil. DeepMind blog, 25.

Greene, M. R., & Hansen, B. C. (2018). Shared spatiotemporal category representations in biological and artificial deep neural networks. *PLoS computational biology*, *14*(7), e1006327.

Güçlü, U., & van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, *35*(27), 10005-10014.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

Hénaff, O. J., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S. M., & Oord, A. V. D. (2019). Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*.

Hendrycks, D., Mazeika, M., Kadavath, S., & Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems* (pp. 15663-15674).

Hermann, K. L., & Lampinen, A. K. (2020). What shapes feature representations? Exploring datasets, architectures, and training. *arXiv preprint arXiv:2006.12433*.

Hernández-García, A., König, P., & Kietzmann, T. C. (2019). Learning robust visual representations using data augmentation invariance. *arXiv preprint arXiv:1906.04547*.

Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., & Botvinick, M. (2020). Unsupervised deep learning identifies semantic disentanglement in single inferotemporal neurons. *arXiv preprint arXiv:2006.14304*.

Hinton, G. E., Sabour, S., & Frosst, N. (2018, February). Matrix capsules with EM routing. In *International conference on learning representations*.

Hong, H., Yamins, D. L., Majaj, N. J., & DiCarlo, J. J. (2016). Explicit information for category-orthogonal object properties increases along the ventral stream. *Nature neuroscience*, *19*(4), 613.

Horikawa, T., & Kamitani, Y. (2017). Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, *8*(1), 1-15.

Huang, Y., & Rao, R. P. (2011). Predictive coding. Wiley Interdisciplinary Reviews: Cognitive Science, 2(5), 580-593.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, *160*(1), 106.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems* (pp. 125-136).

Jiang, R., Li, M., & Tang, S. (2019). Neural clusters encoding curvature and corners emerged in macaque V4. *bioRxiv*, 808907.

Jo, J., & Bengio, Y. (2017). Measuring the tendency of CNNs to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*.

Kar, K., & DiCarlo, J. J. (2020). Fast recurrent processing via ventral prefrontal cortex is needed by the primate ventral stream for robust core visual object recognition. *NEURON-D-20-00886*.

Kar, K., Kubilius, J., Schmidt, K., Issa, E. B., & DiCarlo, J. J. (2019). Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature neuroscience*, *22*(6), 974-983.

Karpathy, A., Abbeel, P., Brockman, G., Chen, P., Cheung, V., Duan, R., ... & Salimans, T. (2016). Generative models. *by openAI. June*.

Kay, K. N. (2018). Principles for models of neural information processing. Neuroimage, 180, 101-109.

Khaligh-Razavi, S. M., & Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS computational biology, 10(11), e1003915.

Kietzmann, T. C., McClure, P., & Kriegeskorte, N. (2019). Deep neural networks in computational neuroscience. In Oxford research encyclopedia of neuroscience.

Kim, M., Tack, J., & Hwang, S. J. (2020). Adversarial Self-Supervised Contrastive Learning. *arXiv preprint arXiv:2006.07589*.

Kindel, W. F., Christensen, E. D., & Zylberberg, J. (2019). Using deep learning to probe the neural code for images in primary visual cortex. *Journal of vision*, *19*(4), 29-29.

Kinoshita, M., & Komatsu, H. (2001). Neural representation of the luminance and brightness of a uniform surface in the macaque primary visual cortex. *Journal of neurophysiology*, *86*(5), 2559-2570.

Kriegeskorte, N., & Douglas, P. K. (2019). *Interpreting encoding and decoding models. Current opinion in neurobiology*, *55*, 167-179.

Konkle, T., & Alvarez, G. A. (2020). Instance-level contrastive learning yields human brain-like representation without category-supervision. *bioRxiv*.

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. Frontiers in systems neuroscience, 2, 4.

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).

Kubilius, J., Bracci, S., & Op de Beeck, H. P. (2016). Deep neural networks as a computational model for human shape sensitivity. *PLoS computational biology, 12(4)*, e1004896.

Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D. L., & DiCarlo, J. J. (2018). Cornet: Modeling the neural mechanisms of core object recognition. *BioRxiv*, 408385.

Kunin, D., Nayebi, A., Sagastuy-Brena, J., Ganguli, S., Bloom, J., & Yamins, D. L. (2020). Two Routes to Scalable Credit Assignment without Weight Symmetry. *arXiv preprint arXiv:2003.01513*.

Lamme, V. A., Super, H., & Spekreijse, H. (1998). *Feedforward, horizontal, and feedback processing in the visual cortex. Current opinion in neurobiology, 8(4)*, 529-535.

Laskar, M. N. U., Giraldo, L. G. S., & Schwartz, O. (2020). Deep neural networks capture texture sensitivity in V2. *Journal of vision*, *20*(7), 21-1.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Lotter, W., Kreiman, G., & Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*.

Lotter, W., Kreiman, G., & Cox, D. (2020). A neural network trained for prediction mimics diverse features of biological neurons and perception. *Nature Machine Intelligence*, *2*(4), 210-219.

Marblestone, A. H., Wayne, G., & Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. Frontiers in computational neuroscience, 10, 94.

Millidge, B. (2020). Deep active inference as variational policy gradients. Journal of Mathematical Psychology, 96, 102348.

Mitchell, T. M. (2015). "3. Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression" (PDF). Machine Learning.

Mosavi, A., Ghamisi, P., Faghan, Y., & Duan, P. (2020). *Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics. arXiv preprint arXiv:2004.01509*.

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 427-436).

Okazawa, G., Tajima, S., & Komatsu, H. (2015). Image statistics underlying natural texture selectivity of neurons in macaque V4. *Proceedings of the National Academy of Sciences*, *112*(4), E351-E360.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020a). Zoom In: An Introduction to Circuits. Distill, 5(3), e00024-001.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020b). An overview of early vision in inceptionv1. Distill, 5(4), e00024-002.

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*, *2*(11), e7.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., & Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*, *3*(3), e10.

Oord, A. V. D., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Parthasarathy, N., & Simoncelli, E. P. (2020). Self-Supervised Learning of a Biologically-Inspired Visual Texture Model. *arXiv preprint arXiv:2006.16976*.

Petkov, N., & Kruizinga, P. (1997). Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. *Biological cybernetics*, *76*(2), 83-96.

Ponce, C. R., Xiao, W., Schade, P. F., Hartmann, T. S., Kreiman, G., & Livingstone, M. S. (2019). Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *Cell*, *177*(4), 999-1009.

Portilla, J., & Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, *40*(1), 49-70.

Pozzi, I., Bohté, S., & Roelfsema, P. (2018). *A biologically plausible learning rule for deep learning in the brain. arXiv preprint arXiv:1811.01768.*

Rafegas, I., & Vanrell, M. (2018). Color encoding in biologically-inspired convolutional neural networks. *Vision research*, *151*, 7-17.

Rane, R. P., Szügyi, E., Saxena, V., Ofner, A., & Stober, S. (2020, June). PredNet and Predictive Coding: A Critical Review. In *Proceedings of the 2020 International Conference on Multimedia Retrieval* (pp. 233-241).

Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019). Do imagenet classifiers generalize to imagenet?. *arXiv preprint arXiv:1902.10811*.

Reddy, M. V., Banburski, A., Pant, N., & Poggio, T. (2020). Biologically Inspired Mechanisms for Adversarial Robustness. *arXiv preprint arXiv:2006.16427*.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., ... & Gillon, C. J. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, *22*(11), 1761-1770.

Rosenfeld, A., Zemel, R., & Tsotsos, J. K. (2018). The elephant in the room. *arXiv preprint arXiv:1808.03305*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211-252.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).

Sacramento, J., Costa, R. P., Bengio, Y., & Senn, W. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in neural information processing systems* (pp. 8721-8732).

Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., & Schmidt, L. (2020). Evaluating Machine Accuracy on ImageNet. In *International Conference on Machine Learning (ICML)*.

Shapley, R., & Hawken, M. J. (2011). Color in the cortex: single-and double-opponent cells. *Vision research*, *51*(7), 701-717.

Sceniak, M. P., Hawken, M. J., & Shapley, R. (2002). Contrast-dependent changes in spatial frequency tuning of macaque V1 neurons: effects of a changing receptive field size. *Journal of Neurophysiology*, *88*(3), 1363-1373.

Scholte, S. (2018). Fantastic DNimals and where to find them. *Neuroimage, 180*, 112-113.

Scholte, H. S., Losch, M. M., Ramakrishnan, K., de Haan, E. H., & Bohte, S. M. (2018). Visual pathways from the perspective of cost functions and multi-task deep neural networks. *Cortex*, *98*, 249-261.

Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., ... & Yamins, D. L. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like?. *BioRxiv*, 407007.

Seijdel, N., Tsakmakidis, N., De Haan, E. H., Bohte, S. M., & Scholte, H. S. (2020). Depth in convolutional neural networks solves scene segmentation. PLoS computational biology, 16(7), e1008022.

Seeliger, K., Fritsche, M., Güçlü, U., Schoenmakers, S., Schoffelen, J. M., Bosch, S. E., & Van Gerven, M. A. J. (2018). Convolutional neural network-based encoding and decoding of visual object recognition in space and time. *NeuroImage*, *180*, 253-266.

Serre, T. (2014). Hierarchical Models of the Visual System.

Serre, T. (2019). Deep learning: the good, the bad, and the ugly. *Annual Review of Vision Science*, *5*, 399-426.

Serre, T., Oliva, A., & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proceedings of the national academy of sciences*, *104*(15), 6424-6429.

Shams, L., & Von Der Malsburg, C. (2002). The role of complex cells in object recognition. *Vision Research*, *42*(22), 2547-2554.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Lillicrap, T. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *362*(6419), 1140-1144.

Storrs, K. R., Kietzmann, T. C., Walther, A., Mehrer, J., & Kriegeskorte, N. (2020). Diverse deep neural networks all predict human IT well, after training and fitting. *bioRxiv*.

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, *23*(5), 828-841.

Sullivan, J., Mei, M., Perfors, A., Wojcik, E. H., & Frank, M. C. (2020). SAYCam: A large, longitudinal audiovisual dataset recorded from the infant's perspective.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Tang, S., Lee, T. S., Li, M., Zhang, Y., Xu, Y., Liu, F., ... & Jiang, H. (2018). Complex pattern selectivity in macaque primary visual cortex revealed by large-scale two-photon imaging. *Current Biology*, *28*(1), 38-48.

van Bergen, R. S., & Kriegeskorte, N. (2020). Going in circles is the way forward: the role of recurrence in visual inference. *arXiv preprint arXiv:2003.12128*.

Victor, J. D., & Purpura, K. P. (1998). Spatial phase and the temporal structure of the response to gratings in V1. *Journal of neurophysiology*, *80*(2), 554-571.

Von Der Heydt, R., Zhou, H., & Friedman, H. S. (2000). Representation of stereoscopic edges in monkey visual cortex. *Vision research*, *40*(15), 1955-1967.

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., ... & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, *21*(6), 860-868.

Whittington, J. C., & Bogacz, R. (2017). An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, *29*(5), 1229-1262.

Wielaard, D. J., Shelley, M., McLaughlin, D., & Shapley, R. (2001). How simple cells are made in a nonlinear network model of the visual cortex. Journal of Neuroscience, 21(14), 5203-5211.

Wiggers, K. (2020, May 2). Yann LeCun and Yoshua Bengio: Self-supervised learning is the key to human-level intelligence [Web page]. Retrieved from https://venturebeat.com/2020/05/02/yann-lecun-and-yoshua-bengio-self-supervised-learning-is-the-key-to-human-level-intelligence/

Wu, Z., Xiong, Y., Yu, S., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*.

Wyatte, D., Jilk, D. J., & O'Reilly, R. C. (2014). Early recurrent feedback facilitates visual object recognition under challenging conditions. *Frontiers in psychology*, *5*, 674.

Xiao, Y., Casti, A., Xiao, J., & Kaplan, E. (2007). Hue maps in primate striate cortex. *Neuroimage*, *35*(2), 771-786.

Xiong, Y., Ren, M., & Urtasun, R. (2020). LoCo: Local Contrastive Representation Learning. *arXiv preprint arXiv:2008.01342*.

Xu, Y., Shen, Y., Zhu, J., Yang, C., & Zhou, B. (2020). Generative Hierarchical Features from Synthesizing Images. arXiv e-prints, arXiv-2007.

Yamins, D. L., & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, *19*(3), 356-365.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences, 111(23), 8619-8624.

Zeman, A. A., Ritchie, J. B., Bracci, S., & de Beeck, H. O. (2020). orthogonal Representations of object Shape and category in Deep convolutional neural networks and Human Visual cortex. *Scientific Reports*, *10*(1), 1-12.

Zhang, R. (2019). Making convolutional networks shift-invariant again. *arXiv preprint arXiv:1904.11486*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhuang, C., Yan, S., Nayebi, A., Schrimpf, M., Frank, M., DiCarlo, J., & Yamins, D. (2020). Unsupervised Neural Network Models of the Ventral Visual Stream. *bioRxiv*.

Zhu, H., Tang, P., Park, J., Park, S., & Yuille, A. (2019). Robustness of object recognition under extreme occlusion in humans and computational models. *arXiv preprint arXiv:1905.04598*.

Ziemba, C. M., Freeman, J., Movshon, J. A., & Simoncelli, E. P. (2016). Selectivity and tolerance for visual texture in macaque V2. *Proceedings of the National Academy of Sciences*, *113*(22), E3140-E3149.