Revisiting Spatial Invariance with Low-Rank Local Connectivity

Gamaleldin F. Elsayed [1] **Prajit Ramachandran** [1] **Jonathon Shlens** [1] **Simon Kornblith** [1]

Abstract

Convolutional neural networks are among the most successful architectures in deep learning with this success at least partially attributable to the efficacy of spatial invariance as an inductive bias. Locally connected layers, which differ from convolutional layers only in their lack of spatial invariance, usually perform poorly in practice. However, these observations still leave open the possibility that some degree of relaxation of spatial invariance may yield a better inductive bias than either convolution or local connectivity. To test this hypothesis, we design a method to relax the spatial invariance of a network layer in a controlled manner; we create a *low-rank* locally connected layer, where the filter bank applied at each position is constructed as a linear combination of basis set of filter banks with spatially varying combining weights. By varying the number of basis filter banks, we can control the degree of relaxation of spatial invariance. In experiments with small convolutional networks, we find that relaxing spatial invariance improves classification accuracy over both convolution and locally connected layers across MNIST, CIFAR-10, and CelebA datasets, thus suggesting that spatial invariance may be an overly restrictive prior.

1. Introduction

Convolutional neural networks (CNNs) are now the dominant approach across many computer vision tasks. Convolution layers possess two main properties that are believed to be key to their success: local receptive fields and spatially invariant filters. In this work, we seek to revisit the latter. Previous work comparing convolutional layers, which share filters across all spatial locations, with locally connected layers, which have no weight sharing, has found that convolution is advantageous on common datasets (LeCun, 1989; Bartunov et al., 2018; Novak et al., 2018). However, this observation leaves open the possibility that some departure from spatial invariance could outperform both convolution and local connectivity (Figure 1).

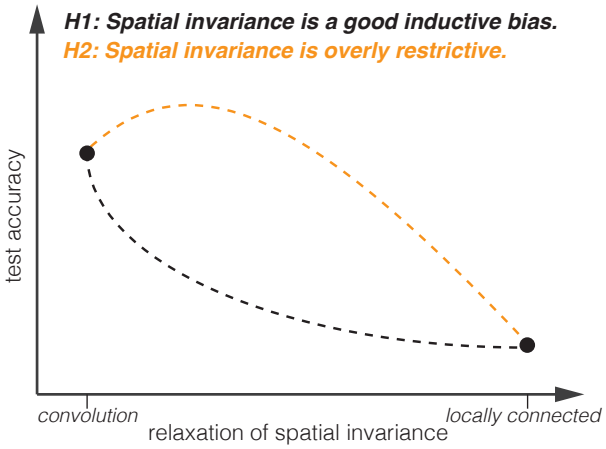

Figure 1: **Is spatial invariance a good inductive bias?** Convolutional architectures perform better than locally connected (or fully connected) architectures on computer vision problems. The primary distinction between convolutional and locally connected networks is requiring spatial invariance in the learned parameter set. Spatial invariance is imposed through weight sharing. One long-standing hypothesis (H1) is that this spatial invariance is a good inductive bias for images (Ruderman & Bialek, 1994; Simoncelli & Olshausen, 2001; Olshausen & Field, 1996). H1 posits that predictive performance would systematically degrade as spatial invariance is relaxed. An alternative hypothesis (H2) suggests that spatial invariance is overly restrictive and some degree of variability would aid predictive performance. The degree to which H1 or H2 is a good hypothesis is largely untested across natural and curated academic datasets and the subject of this work.

The structure of CNNs is often likened to the primate visual

Author contributions: G.F.E. proposed the project idea. G.F.E. designed the methods with contributions from S.K., P.R., and J.S. G.F.E. wrote the code, conducted experiments, and collected results. S.K. reviewed the code. P.R. conducted FLOP count analysis and designed Figure 2. G.F.E., S.K., P.R., and J.S. wrote the paper. [1]Google Research, Brain Team. Correspondence to: Gamaleldin F. Elsayed <gamaleldin@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

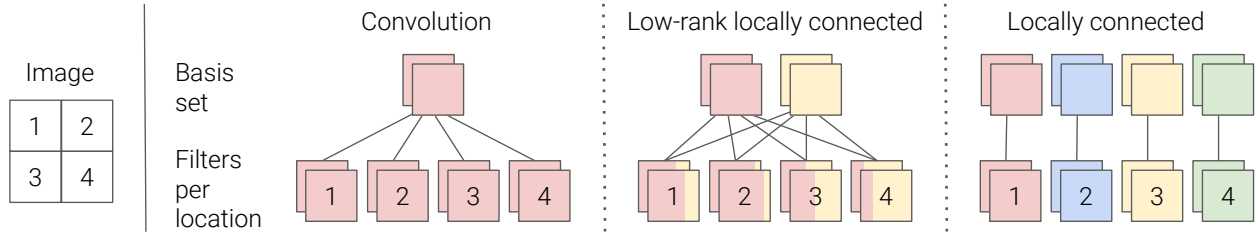

Figure 2: **Filters for each spatial location.** Convolutional layers use the same filter bank for each spatial location (left). Locally connected layers learn a separate filter bank for each spatial location (right). By contrast, low-rank locally connected (LRLC) layers use a filter bank for each spatial location generated from combining a shared basis set of filter banks (middle). Both the basis set and the combining weights are learned end-to-end through optimization. The number of filter banks in the basis set (i.e., the rank parameter) thus determines the degree of relaxation of spatial invariance of the LRLC layer.

system (LeCun et al., 2015). However, the visual system has no direct mechanism to share weights across space. Neurons comprising retinotopic maps have selectivity properties that vary with their position within the map, particularly in high-level visual areas (Hasson et al., 2002; Arcaro et al., 2009; Lafer-Sousa & Conway, 2013; Rajimehr et al., 2014; Srihasam et al., 2014; Saygin et al., 2016; Livingstone et al., 2017). Moreover, the retina contains several types of cells whose distribution and features are organized according to low-rank spatial gradients (Dacey & Petersen, 1992).

Motivated by the lack of synaptic weight sharing in the brain, we hypothesized that neural networks could achieve greater performance by relaxing spatial invariance (Figure 1). Particularly at higher layers of the neural network, where receptive fields cover most or all of the image, applying the same weights at all locations may be a less efficient use of computation than applying different weights at different locations. However, evidence suggests that typical datasets are too small to constrain the parameters of a locally connected layer; functions expressible by convolutional layers are a subset of those expressible by locally connected layers, yet convolution typically achieves higher performance (LeCun, 1989; Bartunov et al., 2018; Novak et al., 2018).

To get intuition for why some relaxation of spatial invariance could be useful, consider images of natural scenes with ground and sky regions. It may be a bad idea to apply different local filters to different parts of the sky with similar appearance. However, it may also be overly limiting to apply the same filter bank to both the sky and the ground regions. Some degree of relaxation of the spatial invariance, such as a different sky and ground filters, may better suit this hypothetical data.

To test the hypothesis that spatial invariance is an overly restrictive inductive bias, we create a new tool that allows us to relax spatial invariance. We develop a *low-rank* locally connected (LRLC) layer[1] that can parametrically adjust the degree of spatial invariance. This layer is one particular method to relax spatial invariance by reducing weight sharing. Rather than learning a single filter bank to apply at all positions, as in a convolutional layer, or different filter banks, as in a locally connected layer, the LRLC layer learns a set of K filter banks, which are linearly combined using K combining weights per spatial position (Figure 2).

In our experiments, we find that relaxing spatial invariance with the LRLC layer leads to better performance compared to both convolutional and locally connected layers across three datasets (MNIST, CIFAR-10, and CelebA). These results suggest that some level of relaxation of spatial invariance is a better inductive bias for image datasets compared to the spatial invariance enforced by convolution layers or lack of spatial invariance in locally connected layers.

2. Related Work

The idea of local connectivity in connectionist models predates the popularity of backpropagation and convolution. Inspired by the organization of visual cortex (Hubel & Wiesel, 1963; 1968), several early neural network models consisted of one or more two-dimensional feature maps where neurons preferentially receive input from other neurons at nearby locations (Von der Malsburg, 1973; Fukushima, 1975). Breaking with biology, the Neocognitron (Fukushima, 1980) shared weights across spatial locations, resulting in spatial invariance. However, the Neocognitron was trained using a competitive learning algorithm rather than gradient descent. LeCun (1989) combined weight sharing with backpropagation, demonstrating considerable gains over locally connected networks (LCNs) on a digit recognition task.

Although the last decade has seen revitalized interest in

[1]Code is available at github.com/google-research/google-research/tree/master/low_rank_local_connectivity.

CNNs for computer vision, local connectivity has fallen out of favor. When layer computation is distributed across multiple nodes, weight sharing introduces additional synchronization costs (Krizhevsky, 2014); thus, the first massively parallel deep neural networks employed exclusively locally connected layers (Raina et al., 2009; Uetz & Behnke, 2009; Dean et al., 2012; Le et al., 2012; Coates et al., 2013). Some of the first successful neural networks for computer vision tasks combined convolutional and locally connected layers (Hinton et al., 2012; Goodfellow et al., 2013; Gregor et al., 2014), as have networks for face recognition (Taigman et al., 2014; Sun et al., 2014; 2015; Yim et al., 2015). However, newer architectures, even those designed for face recognition (Schroff et al., 2015; Liu et al., 2017), generally use convolution exclusively.

Work comparing convolutional and locally connected networks for computer vision tasks has invariably found that CNNs yield better performance. Bartunov et al. (2018) compared the classification performance on multiple image datasets as part of a study on biologically plausible learning algorithms; convolution achieved higher accuracy across datasets. Novak et al. (2018) derived a kernel equivalent to an infinitely wide CNN at initialization and showed that, in this infinite-width limit, CNNs and LCNs are equivalent. They found that SGD-trained CNNs substantially outperform both SGD-trained LCNs and this kernel. However, d'Ascoli et al. (2019) found that initially training a convolution layer and then converting the convolutional layers to equivalent fully connected layers near the end of training led to a slight increase in performance.

Other work has attempted to combine the efficiency of convolution with some of the advantages of local connectivity. Nowlan & Hinton (1992) suggested a "soft weight-sharing" approach that penalizes the difference between the distribution of weights and a mixture of Gaussians. Other work has used periodic weight sharing, also known as tiling, where filters n pixels away share weights (Le et al., 2010; Gregor & LeCun, 2010), or subdivided feature maps into patches where weights are shared only within each patch (Zhao et al., 2016). CoordConv (Liu et al., 2018) concatenates feature maps containing the x and y coordinates of the pixels to the input of a CNN, permitting direct use of position information throughout the network.

Input-dependent low rank local connectivity, which we explore in Sections 3.2.2 and 4.2, is further related to previous work that applies input-dependent convolutional filters. Spatial soft attention mechanisms (Wang et al., 2017; Jetley et al., 2018; Woo et al., 2018; Linsley et al., 2019; Fukui et al., 2019) can be interpreted as a mechanism for applying different weights at different positions via per-position scaling of entire filters. Self-attention (Bahdanau et al., 2015; Vaswani et al., 2017), which has recently been applied to image models (Bello et al., 2019; Ramachandran et al., 2019; Hu et al., 2019), provides an alternative mechanism to integrate information over space with content-dependent mixing weights. Non-local methods (Wang et al., 2018; Zhang et al., 2019) and graph convolution approaches (Chen et al., 2019a) are additional ways to perform content-dependent spatial aggregation. Other approaches apply the same convolutional filters across space, but select filters or branches separately for each example (McGill & Perona, 2017; Fernando et al., 2017; Gross et al., 2017; Chen et al., 2019b; Yang et al., 2019). The dynamic local filtering layer of Jia et al. (2016) uses a neural network to predict a separate set of filters for each position. Our approach predicts only the combining weights for a fixed set of bases, which provides control over the degree of spatial invariance through the size of the layer kernel basis set. The CondConv layer of Yang et al. (2019) predicts combining weights per example that are shared across all spatial locations, whereas our approach learns weights per spatial location, optionally dependent on the example. Further, the computation of spatial filters in the input-dependent LRLC layer can be thought of as a form of dynamic routing, which relates to Capsule networks (Sabour et al., 2017). However, in Sabour et al. (2017), the first capsule layer (PrimaryCaps) is convolutional and fully connects to every DigitCaps capsule, which does not allow partial relaxation of spatial invariance as in the LRLC layer.

3. Methods

3.1. Preliminaries

Let $I \in \mathbb{R}^{H \times W \times C_{\text{in}}}$ be an input with C_{in} channels (H: input height, W: input width, and C_{in}: input channels). In convolution layers, the input I is convolved with a filter bank $F \in \mathbb{R}^{h \times w \times C_{\text{in}} \times C_{\text{out}}}$ to compute $O \in \mathbb{R}^{H \times W \times C_{\text{out}}}$ (h: filter height size, w: filter width size, and C_{out}: filter output channels). For clarity of presentation, we fix the layer output and input to have the same size, and the stride to be 1, though we relax these constraints in the experiments. More formally, the operation of F on the local input patch of size $h \times w \times C_{\text{in}}$ centered at location (i, j), $I_{i,j}$, is:

$$O_{i,j} = I_{i,j} \star F \stackrel{\text{def}}{=} \sum_{x=1}^{h} \sum_{y=1}^{w} \sum_{z=1}^{C_{\text{in}}} (I_{i,j} \odot F)_{x,y,z} \quad (1)$$

where $O_{i,j} \in \mathbb{R}^{C_{\text{out}}}$ is the output at location (i, j) $\forall i \in \{1, \ldots, H\}$ and $\forall j \in \{1, \ldots, W\}$ ($\odot$ is defined as the element-wise multiplication of the input and the filter along the first 3 axes). The spatial invariance of convolution refers to applying the same filter bank F to input patches at all locations (Figure 2 left).

Locally connected layers on the other hand do not share weights. Similar to convolution, they apply filters with local receptive fields. However, the filters are not shared

across space (Figure 2 right). Formally, each output $O_{i,j}$ is computed by applying a different filter bank $F^{(i,j)}$ to the corresponding input patch (i.e., $O_{i,j} = I_{i,j} \star F^{(i,j)}$).

Empirically, locally connected layers perform poorly compared to convolutional layers (Novak et al., 2018). Intuitively, local regions in images are not completely independent and we expect filters learned over one local region to be useful when applied to a nearby region. While locally connected layers are strictly more powerful than convolutional layers and could *in theory* converge to the convolution solution, in practice they don't and instead overfit the training data. However, the superior performance of convolution layers over locally connected layers (LeCun, 1989; Bartunov et al., 2018; Novak et al., 2018) does not imply that spatial invariance is strictly required.

Below, we develop methods that control the degree of spatial invariance a layer can have, which allows us to test the hypothesis that spatial invariance may be overly restrictive.

3.2. Low-rank locally connected layer

Here, we design a locally connected layer with a spatial rank parameter that controls the degree of spatial invariance. We adjust the degree of spatial invariance by using a set of K local filter banks (*basis set*) instead of 1 filter bank in a convolution layer or $H \times W$ filter banks in a classic locally connected layer (K is a hyperparameter that may be adjusted based on a validation subset; $1 \leq K \leq H \times W$). For each input patch $I_{i,j}$, we construct a filter bank to operate on that patch that is a linear combination of the members of the basis set. That is,

$$F^{(i,j)} = \sum_{k=1}^{K} w_{i,j}^{(k)} F^{(k)} \tag{2}$$

where $w_{i,j}^{(k)} \in \mathbb{R}$ are the weights that combine the filter banks in the basis set $\forall i \in \{1, \ldots, H\}$ and $\forall j \in \{1, \ldots, W\}$. This formulation is equivalent to a low-rank factorization with rank K of the layer locally connected kernel. Thus, we term this layer the "low-rank locally connected" (*LRLC*) layer (Figure 2 middle).

Note that, in this paper, we use basis set with filters of similar structure. However, this layer could also be used with a basis set containing filters with different structure (e.g., different filter sizes and/or dilation rates).

The filters in the basis set are linearly combined using weights that are specific to each spatial location. In particular, with input size $H \times W$ and K filter banks in the basis set, we need $H \times W \times K$ weights to combine these filter banks and formulate the filter bank at each spatial location. We propose two ways to learn these combining weights. One method learns weights that are shared across all examples while the second method predicts the weights per example based on a function of the input.

3.2.1. Fixed combining weights

The simplest method of learning combining weights is to learn K scalars per spatial position. This approach is well-suited to datasets with spatially inhomogeneous features, *e.g.* datasets of aligned faces. The number of combining weights scales linearly with the number of pixels in the image, which may be large. Thus to reduce parameters, we learn combining weights per-row and per-column of location (i, j) as follows:

$$\tilde{w}_{i,j}^{(k)} = \alpha_i^{(k)} + \beta_j^{(k)} \tag{3}$$

This formulation reduces the number of combining weights parameters to $(H+W) \times K$, which limits the expressivity of the layer (i.e., constrains the maximum degree of relaxation of spatial invariance). This formulation also performs better in practice (Figure Supp.2).

We further normalize the weights to limit the scale of the combined filters. Common choices for normalization are dividing by the weights norm or using the softmax function. In our early experimentation, we found that softmax normalization performs slightly better. Thus, the combining weights are computed as follows:

$$w_{i,j}^{(k)} = \frac{\exp\left(\tilde{w}_{i,j}^{(k)}\right)}{\sum_{l=1}^{K} \exp\left(\tilde{w}_{i,j}^{(l)}\right)} \tag{4}$$

The filter banks in the basis set and the combining weights can all be learned end-to-end. In practice, we implement this layer with convolution and point-wise multiplication operations, as in Algorithm 1, rather than forming the equivalent locally connected layer. This implementation choice is due to locally connected layers being slower in practice because current hardware is memory-bandwidth-limited, while convolution is highly optimized and fast. We initialize the combining weights to a constant, which is equivalent to a convolution layer with a random kernel, though our main findings remained the same with or without this initialization (Figure Supp.1).

At training time, the parameter count of the LRLC layer is approximately K times that of a corresponding convolutional layer, as is the computational cost of Algorithm 1. However, after the network is trained, the LRLC layer can be converted to a locally connected layer. When convolution is implemented as matrix multiplication, locally connected layers have the same FLOP count as convolution (Figure Supp.4), although the amount of memory needed to store the weights scales with the spatial size of the feature map.

Algorithm 1 Low Rank Locally Connected Layer

Input: $I \in \mathbb{R}^{H \times W \times C_{\text{in}}}$
Trainable Parameters:
$\{F^{(1)}, \ldots, F^{(K)}\} \in K\mathbb{R}^{h \times w \times C_{\text{in}} \times C_{\text{out}}}$ (basis set)
$\alpha^{(1)}, \ldots, \alpha^{(K)} \in \mathbb{R}^{H}$ (combining weights for rows)
$\beta^{(1)}, \ldots, \beta^{(K)} \in \mathbb{R}^{W}$ (combining weights for columns)
$b^{\text{row}} \in \mathbb{R}^{H}$ (biases for rows)
$b^{\text{column}} \in \mathbb{R}^{W}$ (biases for columns)
$b^{\text{channel}} \in \mathbb{R}^{C_{\text{out}}}$ (biases for channels)
Initialize $O = 0 \in \mathbb{R}^{H \times W \times C_{out}}$.
for $k = 1$ **to** K **do**
  $O^{(k)} = I \circledast F^{(k)}$ (convolution with filters in basis set)
  $W^{(k)} = \alpha^{(k)} 1_W^\top + 1_H \beta^{(k)\top}$ (combining weights)
  $O = O + W^{(k)} \odot O^{(k)}$
end for
$B_{i,j,c} = b_i^{\text{row}} + b_j^{\text{column}} + b_c^{\text{channel}}$ (biases)
$O = O + B$
Return: O

Spatially varying bias Typically, a learned bias per channel is added to the output of a convolution. Here, we allow the bias that is added to the LRLC output to also vary spatially. Similar to the combining weights, per-row and per-column biases are learned and are added to the standard channel bias. Formally, we define the layer biases (B) as:

$$B_{i,j,c} = b_i^{\text{row}} + b_j^{\text{column}} + b_c^{\text{channel}} \tag{5}$$

where $b^{\text{row}} \in \mathbb{R}^{H}$, $b^{\text{column}} \in \mathbb{R}^{W}$, and $b^{\text{channel}} \in \mathbb{R}^{C_{\text{out}}}$. The special case of the LRLC layer with $K = 1$ is equivalent to a convolution operation followed by adding the spatially varying bias. We use this case in our experiments as a simple baseline to test if relaxing spatial invariance in just the biases is enough to see improvements.

3.2.2. Input-Dependent Combining Weights

The fixed combining weights formulation intuitively will work best when all images are aligned with structure that appears consistently in the same spatial position. Many image datasets have some alignment by construction, and we expect this approach to be particularly successful for such datasets. However, this formulation may not be well-suited to datasets without image alignment. In this section, we describe an extension of the LRLC layer that conditions the combining weights on the input.

Formally, we modify the combining weights in equation 3 to make them a function of the input:

$$\tilde{w}_{i,j}^{(k)} = g_{i,j}^{(k)}(I) \tag{6}$$

where g is a lightweight neural network that predicts the combining weights for each position. More formally, g takes in the input $I \in \mathbb{R}^{H \times W \times C_{\text{in}}}$ and outputs weights $\tilde{w} \in \mathbb{R}^{H \times W \times K}$. The predicted weights are then similarly normalized as in equation 4 and are used as before to combine the filter banks in the basis set to form local filters for each spatial location. Similar to section 3.2.1, a spatially varying bias is also applied to the output of the layer. The architecture used for g has low computational cost, consisting of several dilated separable convolutions applied in parallel followed by a small series of cheap aggregation layers that output a $H \times W \times K$ tensor. The full architecture of g is detailed in the supplementary section B and shown in Figure Supp.5.

4. Experiments

We performed classification experiments on MNIST, CIFAR-10, and CelebA datasets. We trained our models without data augmentation or regularization to focus our investigation on the pure effects of the degree of spatial invariance on generalization. In our experiments, we used the Adam optimizer with a maximum learning rate of 0.01 and a minibatch size of 512. We trained our models for 150 epochs starting with a linear warmup period of 10 epochs and used a cosine decay schedule afterwards. We used Tensor Processing Unit (TPU) accelerators in all our training.

We conducted our study using a network of 3 layers with 64 channels at each layer and local filters of size 3×3. Each layer is followed by batch normalization and ReLU nonlinearity. The network is followed by a global average pooling operation then a linear fully connected layer to form predictions. Our network had sufficient capacity, and we trained for sufficiently large number of steps to achieve high training accuracy (Table Supp.2). For all our results, we show the mean accuracy $\pm$ standard error based on models trained from 10 different random initializations. Our division of training, validation and test subsets are shown in Table Supp.1.

4.1. Spatial invariance may be overly restrictive

In this section, we investigate whether relaxing the degree of spatial invariance of a layer is a better inductive bias for image classification. We replaced convolution layers at different depths of the network (first, second, third or at all layers) with the designed low-rank locally connected (LRLC) layer. We varied the spatial rank of the LRLC layer, which controls the deviation degree from spatially invariant convolution layers towards locally connected layers. If the rank is small the network is constrained to share filters more across space and the higher the rank the less sharing is imposed. We trained our models and quantified the generalization accuracy on test data at these different ranks.

When rank is 1, the LRLC layer is equivalent to a convolution layer with an additional spatial bias. Adding this spatial

Table 1: **Spatial invariance may be overly restrictive.** Top-1 accuracy of different models (mean ± SE). The optimal rank in LRLC is obtained by evaluating models on a separate validation subset.

LAYER	MNIST	CIFAR-10	CELEBA
CONVOLUTION	98.84 ± 0.01	78.80 ± 0.12	96.66 ± 0.04
CONVOLUTION + SPATIAL BIAS (1ST LAYER)	99.37 ± 0.01	80.94 ± 0.10	97.08 ± 0.02
CONVOLUTION + SPATIAL BIAS (2ND LAYER)	99.25 ± 0.02	80.60 ± 0.05	97.06 ± 0.03
CONVOLUTION + SPATIAL BIAS (3RD LAYER)	99.07 ± 0.02	79.94 ± 0.12	96.76 ± 0.02
CONVOLUTION + SPATIAL BIAS (ALL LAYERS)	99.36 ± 0.01	80.89 ± 0.08	97.12 ± 0.03
LRLC (1ST LAYER)	99.39 ± 0.02	80.94 ± 0.10	97.25 ± 0.02
LRLC (2ND LAYER)	99.42 ± 0.02	80.67 ± 0.09	97.53 ± 0.01
LRLC (3RD LAYER)	$\mathbf{99.51 \pm 0.01}$	$\mathbf{82.70 \pm 0.08}$	$\mathbf{97.73 \pm 0.02}$
LRLC (ALL LAYERS)	99.45 ± 0.01	81.03 ± 0.11	97.51 ± 0.03

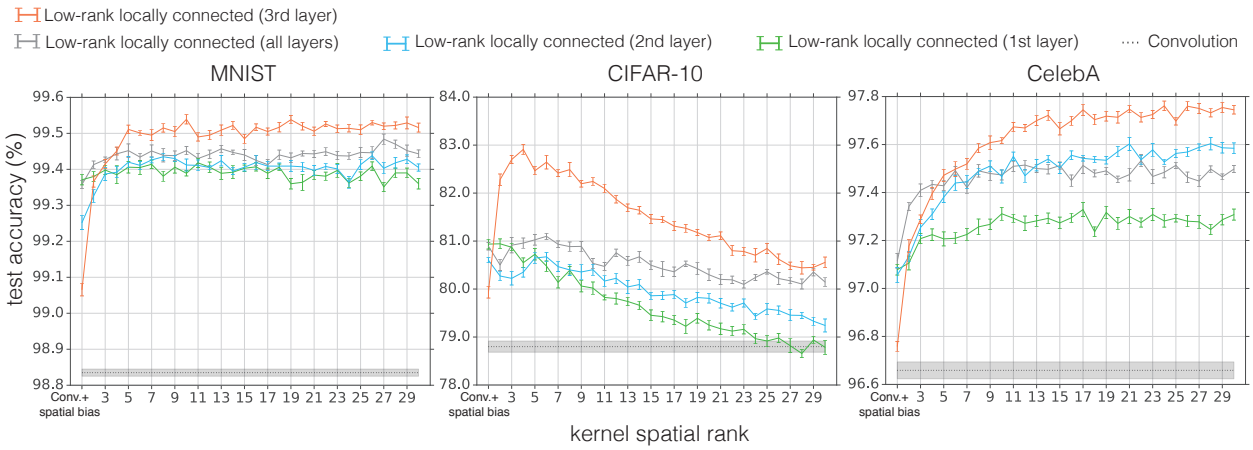

Figure 3: **Low-rank local connectivity is a good inductive bias for image datasets.** Vertical axis shows top-1 test accuracy on digit classification task on 28×28 images from MNIST dataset (left), object classification task on 32×32 images from CIFAR-10 dataset (middle), and gender classification task on 32×32 images from CelebA dataset (right). Horizontal axis shows the locally connected kernel spatial rank used for the low-rank locally connected (LRLC) layer placed at first, second, third, layer or all layers of the network. Note, we only includes low ranks (≤ 30) due to computational constraints. An LRLC layer would require rank $32^2 = 1024$ for a CIFAR-10 for example to match the effective rank of a locally connected layer. The accuracy of a regular convolutional network is shown as a dotted black line for reference. Error bars indicate ± standard errors computed from training models from 10 random initialization. The LRLC layer outperforms classic convolution, suggesting that convolution is overly restrictive and consistent with H2 in Figure 1.

bias to the convolution boosted the accuracy over normal convolution layers (Table 1). Increasing the spatial rank allows the layer to use different filters at different spatial locations, and deviate further from convolution networks. Our results show that doing so further increases accuracy (Figure 3). We find that accuracy of networks with LRLC layers placed at any depth, or with all layers replaced by LRLC layers, is higher than that of pure convolutional networks (Figure 3 and Table 1). These findings provide evidence for the hypothesis that spatial invariance may be overly restrictive. Our results further show that relaxing the spatial invariance late in the network (near the network output) is better than early (at the input). Relaxing the spatial invariance late in the network was also better than doing so at every layer (Table 1). The optimal spatial rank varied across different datasets; rank was the lowest for CIFAR-10 data and was the highest for CelebA.

The LRLC layer has the ability to encode position, which vanilla convolution layers lack. This additional position encoding may explain the increased accuracy. Previous work has attempted to give this capability to convolution networks by augmenting the input with coordinate channels, an approach known as CoordConv (Liu et al., 2018). To test whether the efficacy of the LRLC layer could be explained solely by its ability to encode position, we compared its performance to that of CoordConv. Our results show that

Table 2: **LRLC outperforms baselines.** Top-1 accuracy of different models (mean ± SE). The optimal rank for LRLC and the optimal width for wide convolution models are obtained by evaluating models on a separate validation subset.

LAYER	MNIST	CIFAR-10	CELEBA
LRLC (3RD LAYER)	$\mathbf{99.51 \pm 0.01}$	$\mathbf{82.70 \pm 0.08}$	$\mathbf{97.73 \pm 0.02}$
COORDCONV (LIU ET AL., 2018)	99.46 ± 0.01	81.29 ± 0.13	97.40 ± 0.02
LOCALLY CONNECTED (1ST LAYER)	98.74 ± 0.01	62.54 ± 0.14	96.32 ± 0.02
LOCALLY CONNECTED (2ND LAYER)	98.72 ± 0.02	69.29 ± 0.09	97.19 ± 0.02
LOCALLY CONNECTED (3RD LAYER)	99.10 ± 0.01	71.86 ± 0.10	97.31 ± 0.01
WIDE CONVOLUTION (3RD LAYER)	99.10 ± 0.02	82.40 ± 0.10	97.20 ± 0.02

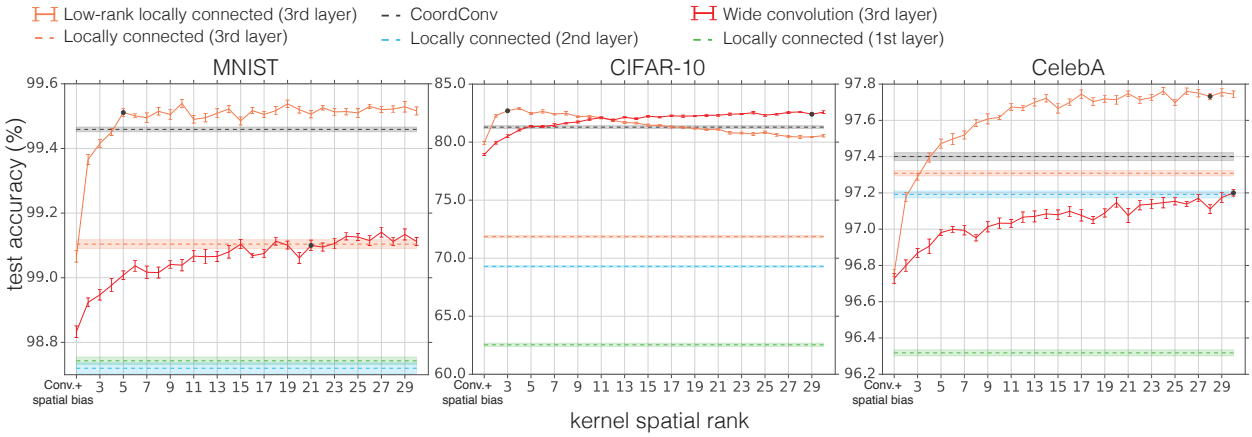

Figure 4: **LRLC outperforms baselines.** Similar to Figure 3, comparing the LRLC layer to different baselines. Baselines include standard locally connected layers, CoordConv (Liu et al., 2018), and convolution networks with wider channels than 64 with width adjusted to match the number of parameters in the LRLC layer. Black markers indicate the best model across spatial ranks for LRLC models and across different widths for wide convolution models. The best models are obtained by performing evaluation on a separate validation subset.

CoordConv outperforms vanilla convolution, but still lags behind the LRLC network (Table 2 and Figure 4), suggesting that the inductive bias of the LRLC layer is better-suited to the data. Unlike CoordConv, the LRLC layer allows controlling and adapting the degree of spatial invariance to different datasets by adjusting the spatial rank. However, with CoordConv, this adjustment is not possible. This gives an intuition of why the LRLC layer suits the data better than CoordConv.

Although locally connected layers have inference-time FLOP count similar to standard convolution layers, the relaxation of spatial invariance comes at the cost of an increase number of trainable parameters. In particular, the number of trainable parameters in the LRLC layer grows linearly with the spatial rank (ignoring the combining weights and spatial biases as they are relatively small). This increase in model parameters does not explain the superiority of the LRLC layer. Locally connected layers have more trainable parameters than LRLC layers, yet perform worse (Figure 4 and Table 2). Moreover, even after widening convolutional layers to match the trainable parameter count of the LRLC layer, networks with only convolutional layers still do not match the accuracy of networks with low-rank locally connected layers (Figures 4, Supp.3 and Table 2). Thus, in our experiments, LRLC layers appear to provide a better inductive bias independent of parameter count.

4.2. Input-dependent low-rank local connectivity is a better inductive bias for datasets with less alignment

In the previous section, our results show that the optimal spatial rank is dataset-dependent. The spatial rank with highest accuracy (the optimal rank) was different across datasets and was generally far from the full rank (i.e., the spatial size of the input), which gives an intuition why convolution layers work well on images in the context of convolution being closer to the optimal rank compared to the vanilla locally connected layers. The optimal rank seems to depend on alignment in the dataset. For example, the optimal rank was highest for CelebA dataset, which comprises approximately

Table 3: **Fixed vs input-dependent combining weights.** Top-1 accuracy of different models (mean ± SE). The optimal rank is obtained by evaluating models on a separate validation subset.

LAYER	MNIST	CIFAR-10	CELEBA
LRLC (3RD LAYER & INPUT DEPENDENT WEIGHTS)	**99.57 ± 0.01**	**84.91 ± 0.07**	97.49 ± 0.03
LRLC (3RD LAYER)	99.51 ± 0.01	82.70 ± 0.08	**97.73 ± 0.02**

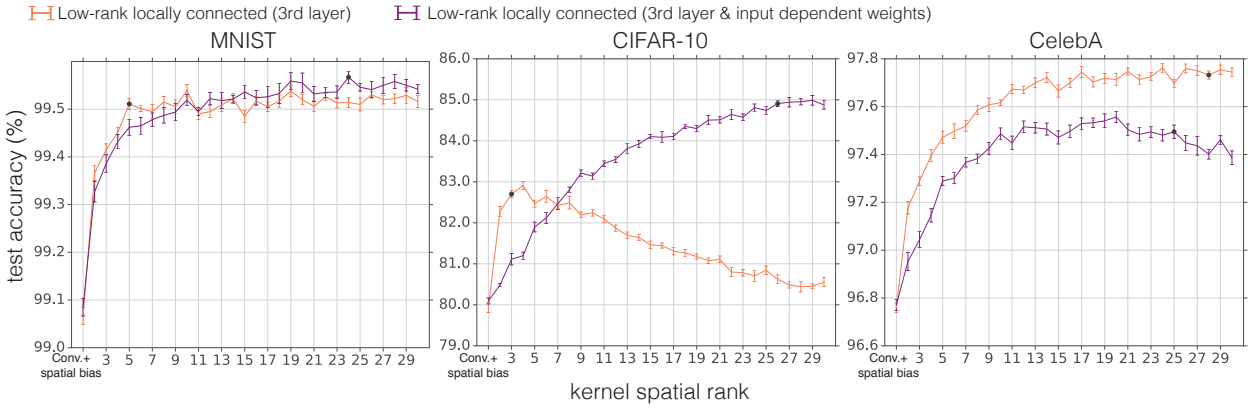

Figure 5: **Input-dependent combining weights.** The LRLC layer learns fixed weights to combine filter banks in the basis set and construct a filter bank to be applied to each input location. The input-dependent LRLC layer uses a simple network to adapt the combining weights to different inputs, making it more suitable for less aligned data such as CIFAR-10. The accuracy of the input-dependent LRLC layer substantially exceeds the accuracy of the fixed LRLC layer on CIFAR-10. However, for more spatially aligned datasets such as MNIST and CelebA, input-dependent LRLC yields modest or no improvement.

aligned face images. By contrast, on CIFAR-10, the optimal rank was low, which may reflect the absence of alignment in the dataset beyond a weak bias toward objects in the center of the images.

These findings raise the question whether one can achieve more gains if the allocation of local filters across space was not fixed across the whole dataset, but rather was conditioned on the input. To answer this question, we modified the LRLC layer to allow the layer to assign local filters based on the input (see Section 3.2.2). This approach has some resemblance to previous work on input-dependent filters (Yang et al., 2019; Jia et al., 2016). We tested whether using this input-dependent way of selecting local filters can give more gains in the less aligned CIFAR-10 dataset. Our results show that the input-dependent LRLC network indeed achieves higher accuracy on CIFAR-10 compared to the fixed LRLC layer, and yields a higher optimal spatial rank (Figure 5 and Table 3). We also experimented the input-dependent LRLC on MNIST and CelebA. We found that the input-dependent LRLC only helped a little on MNIST and hurt accuracy on CelebA compared to the LRLC with fixed weights (Figure 5 and Table 3). This finding suggests that the low-rank local connectivity is a better inductive bias for highly aligned data while the input-dependent low rank local connectivity is better suited to less aligned datasets (Figure 5).

To further investigate this finding, we destroyed the alignment in CelebA by placing the 32×32 face images uniformly within a 48×48 image with random uniform noise, thus randomly translating the CelebA faces. Our results show that the LRLC accuracy on 'Translated CelebA' dropped while input dependent LRLC accuracy remained largely invariant to the translation (Figure Supp.6). We further visualized the combining weights with models of rank 2 so that the results may be easily interpreted. Our results show that the combining weights for the LRLC layer uses one filter bank for central positions where translated faces overlap most, and the other for the periphery (Figure Supp.7 left). For the input dependent LRLC, the combining weights tracked the translated faces, which enables the layer to capture spatially varying information in less aligned datasets (Figure Supp.7 eight).

Table 4: **Application of LRLC layer to ImageNet** Top-1 accuracy of different Resnet-50 models (mean ± SE). The optimal rank for LRLC models is obtained by evaluating models on a separate validation subset.

LAYER	INSERT LAYER	REPLACE 3X3 CONVS
CONVOLUTION	77.22 ± 0.03	76.93 ± 0.07
COORDCONV(LIU ET AL., 2018)	77.23 ± 0.03	77.07 ± 0.08
LRLC	77.47 ± 0.03	77.08 ± 0.02
LRLC (INPUT DEPENDENT WEIGHTS)	77.45 ± 0.03	77.80 ± 0.02
WIDE CONVOLUTION	**77.48±0.05**	**78.54±0.04**

4.3. Feasibility of application of low-rank local connectivity to large scale problems

In this section, we demonstrate the feasibility of using the low-rank locally connected layers in large scale problems. Locally connected layers are not suitable to large scale problems as the number of trainable parameters scale with spatial dimension, which can be prohibitively large in dataset with high-resolution images. For example, a locally connected layer applied to $224x$ images from ImageNet would require 50176 local filter banks in a locally connected layer. In contrast, the number of filter banks in the low-rank locally connected layers only scales with the rank parameter, which in practice is much smaller than the spatial dimensionality.

To demonstrate the feasibility of using the LRLC layers in practice, we conducted two experiments with ResNet-50 on ImageNet (see Appendix C for training details). In the first experiment, we inserted one additional LRLC layer after the first convolution layer. In the second experiment, we replaced all 3×3 convolutions in the network blocks with the LRLC layer. Note that these experiments would have been prohibitively expensive if we used a vanilla locally connected layer instead. We explored spatial ranks 1, 4 and 7 and picked the best rank using a holdout dataset split. Similar to the previous results in MNIST, CIFAR-10, and CelebA, the LRLC models outperformed convolution, which suggests that ImageNet also benefits from relaxing spatial invariance (Table 4). However, on ImageNet a wider version of ResNet-50 which matches the number of parameters in LRLC either matches or outperforms LRLC (Table 4). The feasibility of running these large scale experiments opens the door for the utilization of the LRLC layer in many computer vision problems.

5. Conclusion

In this work, we tested whether spatial invariance, a fundamental property of convolutional layers, is an overly restrictive inductive bias. To address this question, we designed a new locally connected layer (LRLC) where the degree of spatial invariance can be controlled by modifying a spatial rank parameter. This parameter determines the size of the basis set of local filter banks that the layer can use to form local filters at different locations of the input. The LRLC layer has a similar limitation to locally connected layers that it has more trainable parameters than convolution layers. However, the LRLC parameters' count scale only with the spatial rank, which is much smaller scaling compared to the scale by spatial dimensionality in locally connected layers.

Our results show that relaxing spatial invariance using our LRLC layer enhances the accuracy of models over standard convolutional networks, indicating that spatial invariance may be overly restrictive. However, we also found that our proposed LRLC layer achieves higher accuracy than a vanilla locally connected layer, indicating that there are benefits to *partial* spatial invariance. We show that relaxing spatial invariance in later layers is better than relaxing spatial invariance in early layers. Further, we find that the input dependent LRLC layer, which adapts local filters to each input, appears to perform better when data are not well-aligned.

Locally connected layers have largely been ignored by the research community due to the perception that they perform poorly and the complexity of the number of their trainable parameter. However, our findings suggest that this pessimism should be reexamined, as locally connected layers with our low-rank parameterization achieve promising performance and solves the trainable parameters complexity problem. Further work is necessary to capture the advantages of relaxing spatial invariance on other computer vision problems and datasets. One interesting direction to achieve this goal could be to utilize our LRLC formulation and explore using basis set with mixed filter sizes and dilation rates to construct a variety of layers that could suit datasets from different applications.

6. Acknowledgements

We are grateful to Jiquan Ngiam, Pieter-Jan Kindermans, Jascha Sohl-Dickstein, Jaehoon Lee, Daniel Park, Sobhan Naderi, Max Vladymyrov, Hieu Pham, Michael Simbirsky, Roman Novak, Hanie Sedghi, Karthik Murthy, Michael Mozer, and Yani Ioannou for useful discussions and helpful feedback on the manuscript.

References

Arcaro, M. J., McMains, S. A., Singer, B. D., and Kastner, S. Retinotopic organization of human ventral visual cortex. *Journal of neuroscience*, 29(34):10638–10652, 2009.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

Bartunov, S., Santoro, A., Richards, B., Marris, L., Hinton, G. E., and Lillicrap, T. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems*, pp. 9368–9378, 2018.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3286–3295, 2019.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017b.

Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., and Kalantidis, Y. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 433–442, 2019a.

Chen, Z., Li, Y., Bengio, S., and Si, S. You look twice: Gaternet for dynamic filter selection in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9172–9180, 2019b.

Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., and Andrew, N. Deep learning with cots hpc systems. In *International conference on machine learning*, pp. 1337–1345, 2013.

Dacey, D. M. and Petersen, M. R. Dendritic field size and morphology of midget and parasol ganglion cells of the human retina. *Proceedings of the National Academy of sciences*, 89(20):9666–9670, 1992.

d'Ascoli, S., Sagun, L., Biroli, G., and Bruna, J. Finding the needle in the haystack with convolutions: on the benefits of architectural bias. In *Advances in Neural Information Processing Systems*, pp. 9330–9340, 2019.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pp. 1223–1231, 2012.

Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. PathNet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

Fukui, H., Hirakawa, T., Yamashita, T., and Fujiyoshi, H. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Fukushima, K. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20(3-4):121–136, 1975.

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

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

Gregor, K. and LeCun, Y. Emergence of complex-like cells in a temporal product network with local receptive fields. *arXiv preprint arXiv:1006.0448*, 2010.

Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. Deep autoregressive networks. In *International Conference on Machine Learning*, 2014.

Gross, S., Ranzato, M., and Szlam, A. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6865–6873, 2017.

Hasson, U., Levy, I., Behrmann, M., Hendler, T., and Malach, R. Eccentricity bias as an organizing principle for human high-order object areas. *Neuron*, 34(3): 479–490, 2002.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Hu, H., Zhang, Z., Xie, Z., and Lin, S. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3464–3473, 2019.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Hubel, D. H. and Wiesel, T. Shape and arrangement of columns in cat's striate cortex. *The Journal of physiology*, 165(3):559–568, 1963.

Hubel, D. H. and Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

Jetley, S., Lord, N. A., Lee, N., and Torr, P. Learn to pay attention. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyzbhfWRW.

Jia, X., De Brabandere, B., Tuytelaars, T., and Gool, L. V. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pp. 667–675, 2016.

Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

Lafer-Sousa, R. and Conway, B. R. Parallel, multi-stage processing of colors, faces and shapes in macaque inferior temporal cortex. *Nature neuroscience*, 16(12):1870, 2013.

Le, Q. V., Ngiam, J., Chen, Z., Chia, D., Koh, P. W., and Ng, A. Y. Tiled convolutional neural networks. In *Advances in neural information processing systems*, pp. 1279–1287, 2010.

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. Building high-level features using large scale unsupervised learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 507–514, 2012.

LeCun, Y. Generalization and network design strategies. In Pfeifer, R., Schreter, Z., Fogelman, F., and Steels, L. (eds.), *Connectionism in Perspective*, Zurich, Switzerland, 1989. Elsevier. an extended version was published as a technical report of the University of Toronto.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Linsley, D., Shiebler, D., Eberhardt, S., and Serre, T. Learning what and where to attend with humans in the loop. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJgLg3R9KQ.

Liu, R., Lehman, J., Molino, P., Such, F. P., Frank, E., Sergeev, A., and Yosinski, J. An intriguing failing of convolutional neural networks and the CoordConv solution. In *Advances in Neural Information Processing Systems*, pp. 9605–9616, 2018.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. SphereFace: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

Livingstone, M. S., Vincent, J. L., Arcaro, M. J., Srihasam, K., Schade, P. F., and Savage, T. Development of the macaque face-patch system. *Nature communications*, 8 (1):1–12, 2017.

McGill, M. and Perona, P. Deciding how to decide: Dynamic routing in artificial neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2363–2372. JMLR. org, 2017.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. 2018.

Nowlan, S. J. and Hinton, G. E. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992.

Olshausen, B. A. and Field, D. J. Natural image statistics and efficient coding. *Network: computation in neural systems*, 7(2):333–339, 1996.

Raina, R., Madhavan, A., and Ng, A. Y. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pp. 873–880. ACM, 2009.

Rajimehr, R., Bilenko, N. Y., Vanduffel, W., and Tootell, R. B. Retinotopy versus face selectivity in macaque visual cortex. *Journal of cognitive neuroscience*, 26(12): 2691–2700, 2014.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, 2019.

Ruderman, D. L. and Bialek, W. Statistics of natural images: Scaling in the woods. In *Advances in neural information processing systems*, pp. 551–558, 1994.

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

Saygin, Z. M., Osher, D. E., Norton, E. S., Youssoufian, D. A., Beach, S. D., Feather, J., Gaab, N., Gabrieli, J. D., and Kanwisher, N. Connectivity precedes function in the development of the visual word form area. *Nature neuroscience*, 19(9):1250–1255, 2016.

Schroff, F., Kalenichenko, D., and Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Simoncelli, E. P. and Olshausen, B. A. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

Srihasam, K., Vincent, J. L., and Livingstone, M. S. Novel domain formation reveals proto-architecture in inferotemporal cortex. *Nature neuroscience*, 17(12):1776, 2014.

Sun, Y., Wang, X., and Tang, X. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1891–1898, 2014.

Sun, Y., Liang, D., Wang, X., and Tang, X. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. DeepFace: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

Uetz, R. and Behnke, S. Large-scale object recognition with cuda-accelerated hierarchical neural networks. In *2009 IEEE international conference on intelligent computing and intelligent systems*, volume 1, pp. 536–541. IEEE, 2009.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Von der Malsburg, C. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14(2):85–100, 1973.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., and Tang, X. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2017.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

Woo, S., Park, J., Lee, J.-Y., and So Kweon, I. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Yang, B., Bender, G., Le, Q. V., and Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. In *Advances in Neural Information Processing Systems*, pp. 1305–1316, 2019.

Yim, J., Jung, H., Yoo, B., Choi, C., Park, D., and Kim, J. Rotating your face using multi-task deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 676–684, 2015.

Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Zhang, S., He, X., and Yan, S. Latentgnn: Learning efficient non-local relations for visual recognition. In *International Conference on Machine Learning*, pp. 7374–7383, 2019.

Zhao, K., Chu, W.-S., and Zhang, H. Deep region and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3391–3399, 2016.