On Foveation of Deep Neural Networks

by

Sanjana Srivastava

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Author .....
Department of Electrical Engineering and Computer Science
May 24, 2019

Certified by .....
Tomaso Poggio
Professor
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

On Foveation of Deep Neural Networks

by

Sanjana Srivastava

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2019, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The human ability to recognize objects is impaired when the object is not shown in full. "Minimal images" are the smallest regions of an image that remain recognizable for humans. [26] show that a slight modification of the location and size of the visible region of the minimal image produces a sharp drop in human recognition accuracy. In this paper, we demonstrate that such drops in accuracy due to changes of the visible region are a common phenomenon between humans and existing state-of-the-art convolutional neural networks (CNNs), and are much more prominent in CNNs. We found many cases where CNNs classified one region correctly and the other incorrectly, though they only differed by one row or column of pixels, and were often bigger than the average human minimal image size. We show that this phenomenon is independent from previous works that have reported lack of invariance to minor modifications in object location in CNNs. Our results thus reveal a new failure mode of CNNs that also affects humans to a lesser degree. They expose how fragile CNN recognition ability is for natural images even without synthetic adversarial patterns being introduced. This opens potential for CNN robustness in natural images to be brought to the human level by taking inspiration from human robustness methods. One of these is eccentricity dependence, a model of human focus in which attention to the visual input degrades proportional to distance from the focal point [7]. We demonstrate that applying the "inverted pyramid" eccentricity method, a multi-scale input transformation, makes CNNs more robust to useless background features than a standard raw-image input. Our results also find that using the inverted pyramid method generally reduces useless background pixels, therefore reducing required training data.

Thesis Supervisor: Tomaso Poggio
Title: Professor

Acknowledgments

I would like to acknowledge Dr. Xavier Boix and Dr. Guy Ben-Yosef for their immense support and guidance in my thesis project. I am also grateful to Prof. Tomaso Poggio and Prof. Shimon Ullman for helpful feedback and discussions. Finally, I would like to acknowledge Ryan Prinster for choosing the desk next to mine and entertaining me all year. This work is supported by the National Science Foundation Science and Technology Center Award CCF-123121, the MIT-IBM Brain-Inspired Multimedia Comprehension project, the MIT-Sensetime Alliance on Artificial Intelligence, and the Semiconductor Research Corporation Joint University Microelectronic Program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is partially a summary of a paper titled "Minimal Images in Deep Neural Networks: Fragile Object Recognition in Natural Images", published in the proceedings of the International Conference on Learning Representations 2019 [21]. Part of the text and figures are common with [21]; these have been noted throughout this thesis.

As stated in [21], convolutional neural networks (CNNs) have reached tremendous success in recognizing and localizing objects in images. The fundamental approach that led to CNNs consists of building artificial systems based on the brain and human vision. Yet in many important aspects, the capabilities of CNNs are inferior to those of human vision. A promising strand of research is to investigate the similarities and differences, and by bridging the gaps, further improve CNNs [12].

1.1 Investigating similarities and differences

This section appears in [21].

Studying cognitive biases and optical illusions is particularly revealing of the function of a visual system, whether natural or artificial. [26] present such a striking phenomenon of human vision, called "minimal images" [26, 4, 5]. Minimal images are small regions of an image (*e.g.* 10x10 pixels) in which only a part of an object is observed, and a slight adjustment of the visible area produces a sharp drop in human

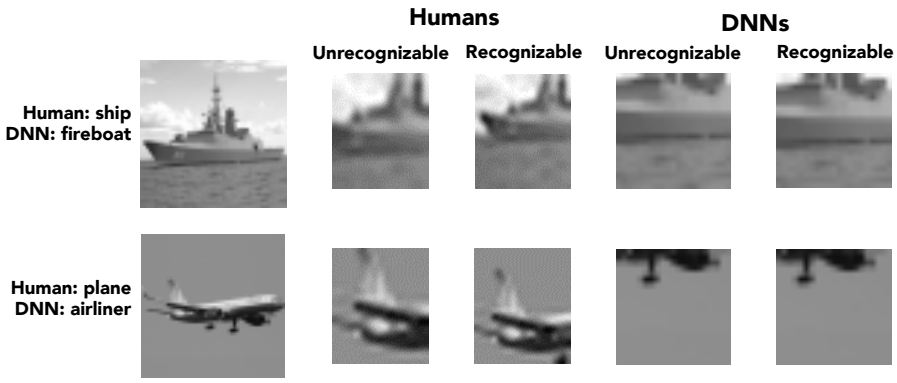


Figure 1-1: Qualitative examples of human and CNN minimal images.

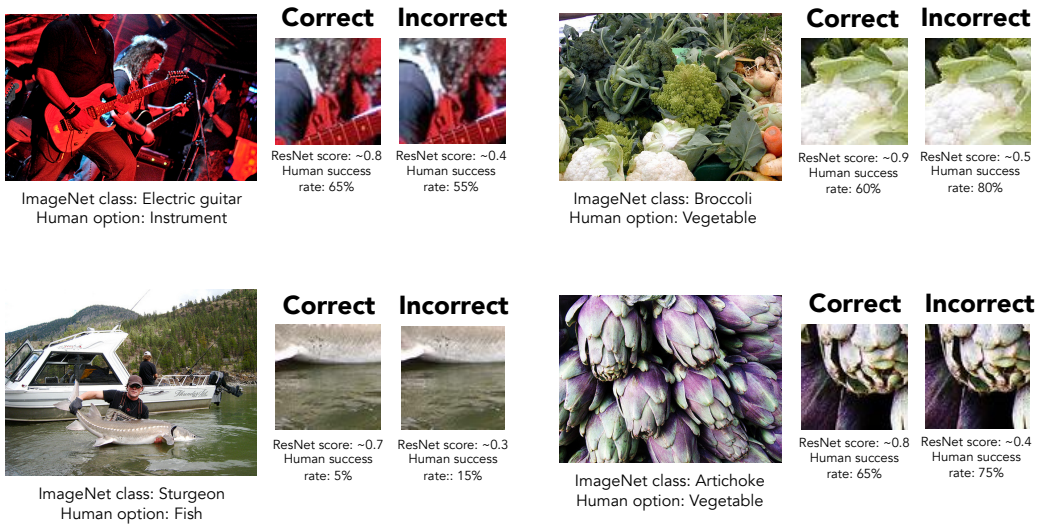


Figure 1-2: Qualitative examples of fragile recognition images (FRIs) (figure source: [21]).

recognition accuracy. Figure 1-1 provides examples of human minimal images.

[26] show that CNNs are unable to recognize human minimal images, and the CNN drop in accuracy for these minimal images is gradual rather than sharp. This begs the question of whether the sharp drop in accuracy for minimal images is a phenomenon exclusive to human vision, or there exist distinct but analogous images that produce a sharp drop in CNN accuracy.

In this paper, we provide evidence for the latter hypothesis by showing that there is a large set of analogs to minimal images that affect CNNs. These are different from the minimal images for humans in several aspects, namely region size and location, and frequency and sharpness of the drop in accuracy. We find that a slight adjustment of a one-pixel shift or two-pixel shrink of the visible image region produces a drop in

CNN recognition accuracy in many image regions. Figure 1-2 shows several examples of minimal image analogs of state-of-the-art CNNs. The examples are for ResNet [13] analogs to human minimal images (more examples of various kinds can be found in section A). The incorrectly classified image region is slightly (two pixels) smaller than the correctly classified crop. Even when CNNs show a significant change in confidence between regions, humans recognize them at similar rates.

The adjustments of the visible area that affect CNNs are almost indistinguishable to humans and can occur in larger regions than the "human minimal" region. To describe this phenomenon we introduce *fragile recognition images (FRIs)* for CNNs, which are more general than minimal images:

Fragile Recognition Image (FRI): *A fragile recognition image is a region of an image for which a slight change of the region's size or location in the image produces a large change in CNN recognition output.*

This definition is more general than the definition of minimal images for humans, as the latter is included in the definition for CNNs. Minimal images are the case of fragile recognition in which the slight change is a reduction in the size of the region, and the minimal image is one of the smallest possible FRIs. In human vision, the more general definition of fragile recognition that we are introducing here is not useful because human minimal images appear only when the visible area of the object is small.

Since FRIs are hardly distinguishable to humans but cause CNNs to fail, there is a connection with so-called *adversarial examples* presented by [23]. Adversarial examples are images with small synthetic perturbations that are imperceptible to humans, but produce a sharp drop in CNN recognition accuracy. There are several types, e.g. [11, 16], and the strategies to alleviate them are not able to fully protect CNNs [25, 17]. Furthermore, humans may suffer from adversarial attacks; human recognition ability is shown to be impaired by perturbations under rapid image presentation [9]. Unlike these adversarial examples, fragile recognition arise in natural images without introducing synthetic perturbation. This causes new concerns for use of CNNs in computer vision applications.

We evaluate FRIs in ImageNet [8] for state-of-the-art CNNs, specifically VGG-16 [20], Inception [22], and ResNet [13]. Results show that FRIs are abundant and can occur for any region size. Furthermore, we investigate whether fragile recognition is related to the lack of invariance to small changes in the object location, which has been recently reported in the literature to affect CNNs [10, 3]. Our results demonstrate that fragile recognition is independent from this phenomenon: bigger pooling regions reduce most of the lack of invariance to changes in object location, while pooling only marginally reduces fragile recognition. We also show that known strategies to increase network generalization, adding regularization and data augmentation, reduce the number of FRIs but still leave far more than humans have. These results highlight how much more fragile current CNN recognition ability is than human vision.

1.2 Bridging gaps to improve CNNs

An aspect of human vision that may improve robustness to boundary aberrations and resulting problems like minimal images is foveation [24]. Human vision weighs different parts of the field of view differently, giving more attention to certain fixation points and degrading focus radially outward from them. There are several models of human foveation that can be implemented for CNNs; the most naive of these is to simply crop a certain portion of a scene and give it as input to a CNN. This is equivalent to taking an image of a scene, which necessarily will not capture the entire scene, and giving it as input to a CNN. Therefore, CNNs inherently use a simple model of foveation.

In this project, we aim to analyze the effects of foveation in CNN testing and training. We consider eccentricity-dependent networks, which cause the attention to each region of the input to be dependent on its eccentricity, or distance from the center of the image. Because humans also view inputs at multiple scales, we use the inverted pyramid architecture, which takes multiple, differently-sized concentric crops of an image and inputs all of them together to a network [7]. The inverted pyramid architecture has been found to improve CNN accuracy and eccentricity dependence

in general has been found to improve computational complexity [7, 1]. We find that the inverted pyramid architecture maintains accuracy even with a small amount of training data, and is robust to various amounts of useless background features. We thus find that the inverted pyramid architecture reduces overall training data complexity and improves robustness to background features that can confound standard CNN architectures.

The contributions of this paper are:

- Grid-search method for detecting FRIs in CNNs (section 2.1)
- Evaluation and analysis of FRIs in CNNs as analog to human minimal images (sections 3.1, 3.2)
- Establishment of FRIs as a natural adversarial example that is distinct from translational adversarial examples (section 3.3)
- Evaluation of inverted pyramid's efficacy in reducing training data complexity (section 4.1)
- Evaluation of inverted pyramid's efficacy in CNN robustness to useless background features (section 4.2)

Chapter 2 will discuss the methods used for FRI detection/analysis and evaluation of the inverted pyramid in terms of reducing training data complexity and improving robustness to useless background features, even with small amounts of training data. Chapter 3 will present the results of the FRI detection method and analysis of their properties. Chapter 4 will present results and analysis of experiments on the inverted pyramid architecture. Chapter 5 will discuss FRIs and human minimal images as analogous failure modes caused by lack of robustness to background features and aberrations, and the potential of the inverted pyramid to improve robustness to varied backgrounds in CNNs.

Chapter 2

Methods

2.1 Extracting fragile recognition images

The following section appears in [21].

In this section we introduce the method for extracting FRIs for CNNs. The method for extracting human minimal images employs a tree search strategy [26]. The full object is presented to human subjects and they are asked to recognize it. If at least 50% of subjects recognize it correctly, smaller crops of the object, called descendants, are tested. If any of the smaller region is recognizable, the crop size is further reduced and tested. Once an image crop is found such that it is recognizable and all of its descendants are not recognizable, it is considered a human minimal image.

Our FRI extraction method for CNNs relies on an exhaustive grid search. This

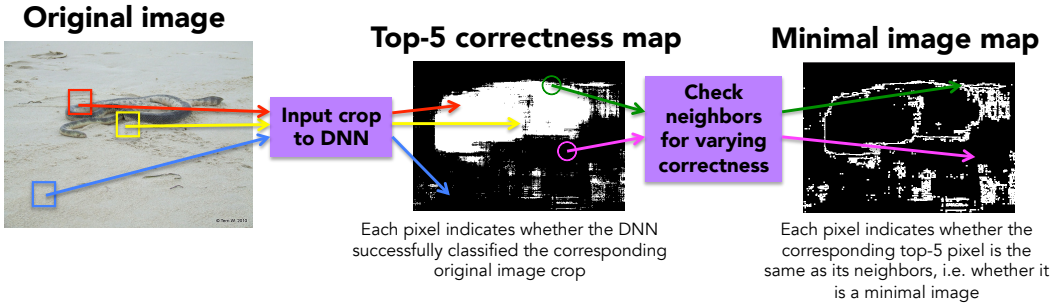


Figure 2-1: The process of generating a fragile recognition image (FRI) map from a full image (figure source: [21]).

is possible in CNNs due to parallelization across multiple GPUs, and would be prohibitively time-consuming to replicate with humans. The grid search consists of a two-step process: first, every possible square region is classified by the CNN and the correctness is annotated in the *correctness map*. From the correctness map, each region's correctness is compared with the region's slightly changed location or size in order to determine if there has been a change of the correctness. FRIs are regions that are classified correctly and a small change causes failure, as well as regions that are classified incorrectly and a small change causes success.

The two step process to detect FRIs is summarized in Figure 2-1, in which the FRI map shows FRIs for which a one-pixel shift in any direction of the visible region produces an incorrect classification; white pixels indicate FRIs and black pixels indicate non-FRIs. We now detail the two steps:

1. Correctness map generation. An exhaustive grid search is performed to see if each possible square image region of a fixed size is classified correctly by a given CNN. After extracting the region from the image, the region is resized to be of the size required by the network. The region size is parametrized by P, which is defined as the proportion of the image occupied by the region, i.e. $P = S/\min(h, w)$ where h and w are the height and width of the input image, and S is the region's side length. The results are arranged in a map such that a given map pixel contains the binary correctness for the square region centered at the corresponding pixel in the original image. The resulting map is of dimension $(h - S) \times (w - S)$ due to padding loss. The first two panels of Figure 2-1 show a visualization of the correctness map generation process.

2. Fragile recognition image (FRI) extraction from correctness maps. We define different variations of FRIs, depending if they are based on changes on the location or size of the image region, and on how strict we are when evaluating the changes in the correctness of CNN:

–*Shift or Shrink*: we define two types of "small changes" of the region that affect CNN correctness. "Shift" is a one-pixel translation of the region location; "shrink" is a two-pixel reduction of the region side length within the region's original boundaries,

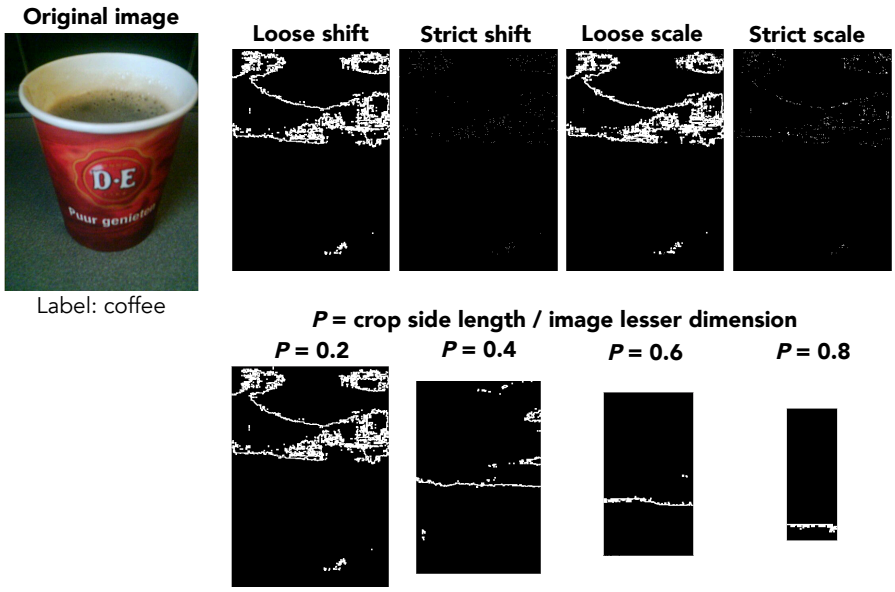


Figure 2-2: Fragile recognition image (FRI) maps for Inception [22] (figure source: [21]).

in a fixed full image size.

–*Strict or Loose*: the visible region can be shifted in various directions, or shrunk in different ways while remaining within its original boundaries. "Loose" FRIs are regions such that there exists a small change that flip network correctness. "Strict" FRIs are regions such that network correctness is flipped for all small changes.

These definitions yield four fragile recognition types: loose shift, loose shrink, strict shift, and strict shrink. Note that strict shrink is the most analogous to human minimal images. The correctness maps are used to detect fragile recognition due to shifts by comparing neighbouring pixels in a correctness map, and due to shrinks by comparing correctness maps at two slightly different region sizes. We use *fragile recognition image (FRI) maps* to visually represent the result of the grid search that extracts FRIs. As shown in the FRI map of Figure 2-1, each pixel of the map indicates whether the corresponding window in the original image is an FRI. The second and third panel show a visualization of the FRI map generation process. Figure 2-2 shows an ImageNet image and each of its FRI maps.

2.2 Determining efficacy of eccentricity dependence

In this section we outline our investigation of the benefits of eccentricity dependence in terms of training data volume and robustness to useless background features. We consider two architectures:

–*Vanilla*: a CNN that takes a raw image resized to have height and width S, where S is a predetermined constant.

–*Inverted pyramid*: a CNN that resizes the raw image to $S \times S$, then takes n concentric crops of the resized raw image along the channel axis. For $i = 1...n$, crop i has side length $\frac{iS}{n}$. Finally, these n crops are all resized to have side length $\frac{S}{n}$ and concatenated along the channel (last) axis. n can be considered the "pyramid depth".

This definition of the inverted pyramid, presented in [7], enables a CNN to see multiple scales of the object and surrounding area, and regions further from the center are seen at lower resolution. As a result, the dimensionality of the background features is significantly reduced. If the background features are useless to begin with, then we hypothesize that reducing them will allow a network to achieve the same performance with fewer training examples that it would more training examples, but the original amount of useless background features. Mathematical justification for this hypothesis is as follows:

> Consider a fully-connected neural network layer with weight matrix w, given input vector $x_i \in X$, and generating output vector $y_i = w^T x_i$.
>
> Consider $\hat{w}$ to be the ideal w.
>
> Consider $x_i' = [x_i \; x_i^*]$, where x_i^* is a set of useless features that may have any value, i.e. may be nonzero.
>
> Given that $\dim x_i' > \dim x_i$, a layer that processes x_i' must have weight matrix $w' = [w \; w^*]$ that has $\dim x_i'$ columns, while w has $\dim x_i$ columns.
>
> Given that all features in x^* are useless for classification, the ideal w', $\hat{w}' = [\hat{w} \; 0]$. Crucially, to achieve $\hat{w}'$, $w^* = 0$.

By the representer theorem [19], each column of w is a linear combination of input vectors: $w_i = \Sigma_i \alpha_i x_i$ where each α_i is a constant.

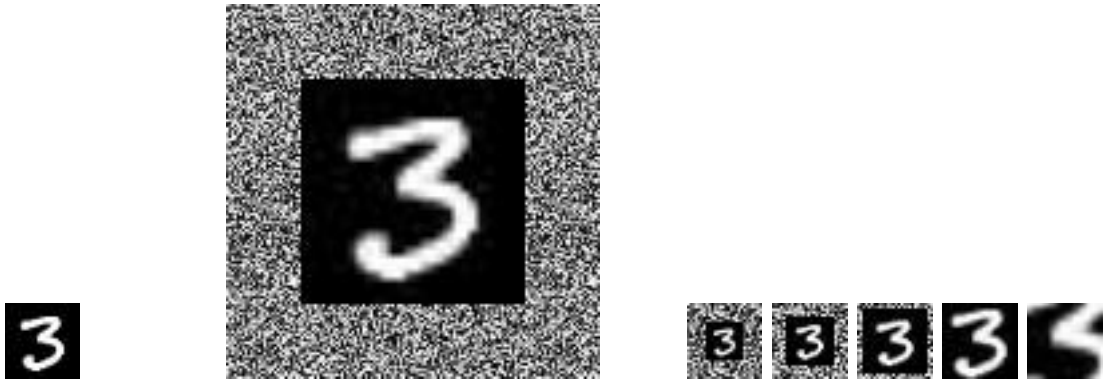
Given that x_i^* may be nonzero, and w^* is a linear combination of all x_i^*, there may be multiple linear combinations of x_i^* that form $[\hat{w}\ w^*]$, but $w^* = 0$ will not be true for all of them.

It will therefore generally take more training examples x_i' to achieve $\hat{w}'$ than it would x_i to achieve $\hat{w}$.

This result applies to a fully-connected layer with linearity, but convolutional layers in CNNs can also be written as matrix multiplications. Furthermore, the ReLU nonlinearities used in state-of-the-art CNNs are piecewise linear. Based on this reasoning, we aim to establish the ability of the inverted pyramid architecture to reduce required training data empirically.

To do so, we test the vanilla and inverted pyramid architectures on various transformations of the MNIST dataset [15]. MNIST images contain isolated grayscale "objects" (handwritten digits) on a black background, so they do not have any of their own background noise. The object in each image has very little space around it, i.e. its bounding box takes up the entire image and it is inherently centered. All MNIST images are 28×28 pixels, so they are square and all need to be resized the same amount for each architecture. The vanilla architecture takes inputs with height 140 pixels, width 140 pixels, and one channel. The inverted pyramid architecture has a pyramid depth of five, so it takes inputs with height 28 pixels, width 28 pixels, and five channels. These heights and weights were determined by maximizing the vanilla input size that our computational resources could handle, and qualitatively confirming that the network could still reach high accuracy. Both architectures have two convolutional layers followed by two fully-connected layers, which is standard for MNIST. The only differences are the input dimensions and the number of channels in the first convolutional layer's filter.

To establish the ability of inverted pyramid to reduce useless background effectively, we test each architecture on inputs with various amounts of added useless



(a) raw MNIST image (b) vanilla input (c) channels of inverted pyramid input

Figure 2-3: Sample inputs ($B = 28$).

background and various numbers of training examples. Our three primary experimental variables are therefore architecture, number of training examples shown to the CNN, and background size.

Given an image M with side length s (MNIST example in figure 2-3a), a vanilla input side length S, a pyramid depth n, and a background size B, the vanilla input is constructed via the following process. M is resized to $S - 2B$ using bilinear interpolation to approximate linear combination as closely as possible even in the object features. The resulting tensor is augmented with a border of width B on each edge, so that the result has side length S. The border consists of random values selected from a uniform distribution. This tensor is input to the vanilla architecture. Figure 2-3b shows an example for MNIST with background size 14. Next, the inverted pyramid transformation is applied to M'. n crops are taken at evenly decreasing side lengths, resized down to height/width $\frac{S}{n}$ with bilinear interpolation, then concatenated along the channel dimension to form a tensor of dimensions $(\frac{S}{n}, \frac{S}{n}, n)$. This tensor is input to the inverted pyramid architecture. Figure 2-3 shows a representation of the same example in MNIST with a pyramid depth of five.

We consider T to represent a number of training examples per class. For each architecture, the following experiments are performed for a range of T:

1. Training and testing on images with a predetermined B, such that all images get the same amount of background (experiments include various values of B)

2. Training with random B chosen from a uniform distribution over $[0, \frac{S-s}{2}]$, testing on fixed B

3. Training and testing on images with random B chosen as described in experiment 2

Experiment 1 most clearly demonstrates how more and more useless background features impact the performance of the two architectures when given various numbers of training examples. Experiment 2 analogizes best to natural images, as the CNN always sees images in which the size and variance of the background is unknown and not necessarily consistent with any other example. Experiment 3 is used to shed light on the inner workings of CNNs trained with random B, in which the filters cannot simply learn the exact location of the background region. During testing, when the location of the background region is known, we not only observe accuracy but also feature maps generated by the convolutional layers. This provides some insight into how these architectures learn to ignore useless background even when its magnitude is not consistent.

It is important to note that our inputs always contain centered objects, which is not something that can be assumed in nature. Expanding our study to address various locations along with various scales is a future step.

Chapter 3

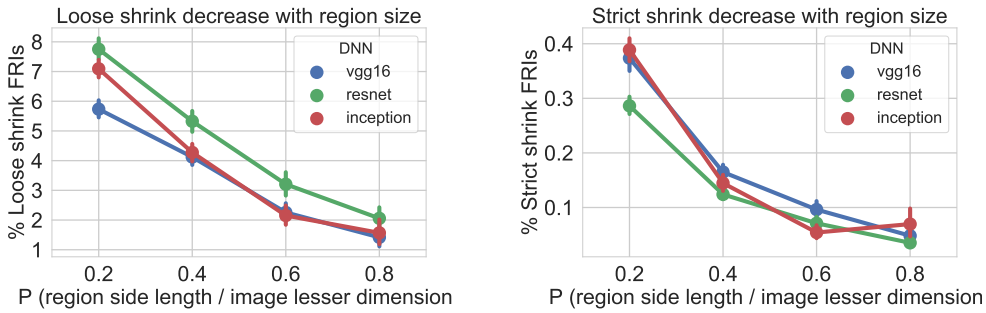
Fragile recognition images of state-of-the-art CNNs

The following chapter appears in [21].

In this section, we will first discuss occurrence of fragile recognition for state-of-the-art object recognition models through results based on ImageNet [8]. Then, we analyze if data augmentation and regularization help reduce fragile recognition through experiments in CIFAR-10 [14].

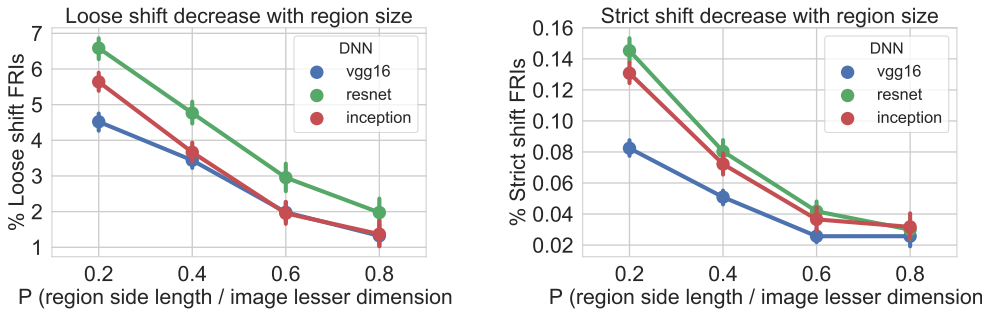
3.1 Fragile recognition images for state-of-the-art CNNs in ImageNet

The following experiments are performed on 500 images, randomly sampled from ImageNet's validation set, which consists of $50,000$ images and ground-truth object bound-in boxes [8]. The images are sampled from 10 supercategories (dog, snake, monkey, fish, vegetable, musical instrument, boat, land vehicle, drinks, furniture), each of which covers a set of ImageNet categories. Correctness is measured as top-5 accuracy, which is commonly used in ImageNet. FRIs are extracted for three architectures: VGG-16 [20], Inception [22], and ResNet [13]. Experiments are run in eight K80 NVIDIA GPUs. The exhaustive grid search takes about 5 minutes for one



(a) loose shrink FRIs (b) strict shrink FRIs

Figure 3-1: Shrink fragile recognition images (FRIs) in ImageNet (figure source: [21]).



(a) loose shift FRIs (b) strict shift FRIs

Figure 3-2: Shift fragile recognition images (FRIs) in ImageNet (figure source: [**?**]).

image using all eight GPUs. In the following paragraphs, we report the results of the experiments.

We evaluate how much a CNN is affected by fragile recognition by quantifying the proportion of possible image regions that are FRIs, i.e. we quantify the amount of regions affected by a shift or shrink. Recall that we consider a region to be an FRI when the classification changes from correct to incorrect, and also from incorrect to correct. As expected, we found that both of these cases are approximately equally probable under all tested conditions.

In Figure 3-1, (a) shows loose shrink FRIs, which indicate the general fragility of CNNs to a slight reduction in the visible region. (b) shows strict shrink FRIs, which are te equivalent of human minimal images. Shift FRIs also follow this pattern, as shown in Figure 3-2. Figure 3-1 shows the percentage of shrink FRIs in the image for a given region size, P; in Figure 3-2 we show the same for shift FRIs. All networks

are significantly affected by FRIs, with ResNet being the most: almost one out of 50 regions of a size that covers almost the entire image ($P = 80\%$) can affect ResNet. When the size of the image region is $P = 20\%$, FRIs are even more frequent (8 times more), i.e. almost two out of 25 regions in the image is a loose FRI for ResNet.

Comparing Figure 3-1a and Figure 3-1b, we see that the proportion of strict FRIs is less than that of loose FRIs because of the more stringent definition. The results show that there are many regions in an image for which the network is very sensitive to slight changes in the region, e.g. one out of 250 image regions of size $P = 20\%$ will be misclassified by ResNet when the region is slightly shrunk. Recall that strict shrink FRIs are the equivalent case of minimal images in humans. Thus, these results demonstrate the existence of many minimal images analogs in CNNs.

Note that smaller FRIs (lower P) are much more frequent than larger ones, as observed across the different network architectures and types of FRIs. This trend is expected: one-pixel shifts and two-pixel shrinks are proportionally larger changes for smaller regions, and larger regions generally allow for more high-level features to be included.

We verified that FRIs are not an artifact of the algorithm that resizes the region to the size required by the CNN (224×224 pixels for VGG-16). We took regions of side length 224 and removed any resizing before input, and we observed that this procedure produces the same results we reported.

Besides the qualitative examples in Figure 1-1 and Figure **??**, we display a more varied set of qualitative examples in Figure A-2. We observe that FRIs are usually located within object boundaries but can also be found in the background. This is because CNNs are able to recognize regions that only contain background [28], as they have been shown to exploit dataset biases.

In Figure 3-3 we display the CNN's output confidence in the true class for individual regions in map form. These maps are analogous to the top-5 correctness map seen in 2-1, except that each cell contains the CNN's confidence in the ground-truth class after the softmax layer rather than a binary correctness value. The figure shows several maps given a region size of $P = 0.2$ (the same conclusions are extracted for any

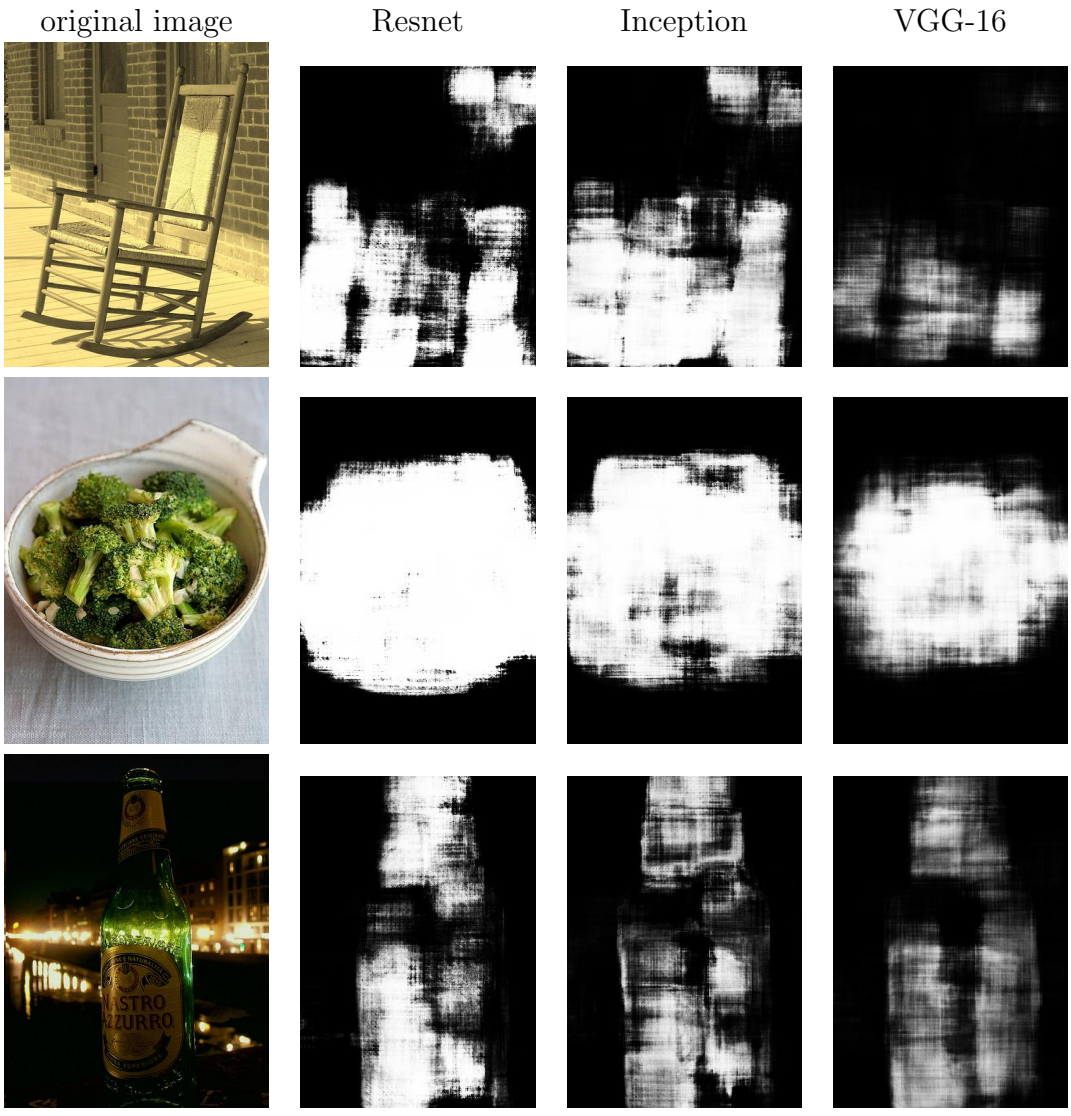


Figure 3-3: Confidence score of the DNNs for different regions (figure source: [21]).

of the other values of P we test in the paper). These maps show that the confidence can drop sharply within a small amount of change, but does not change much within informative regions, validating the fragility of CNNs on FRIs. Finally, in Figure A-3 we show qualitative examples of the activation maps at different layers of the CNN of the correctly classified crop and its shifted version. These examples show what we have observed in all cases: the activation maps are imperceptibly similar at the first layers but are clearly different at the last layers.

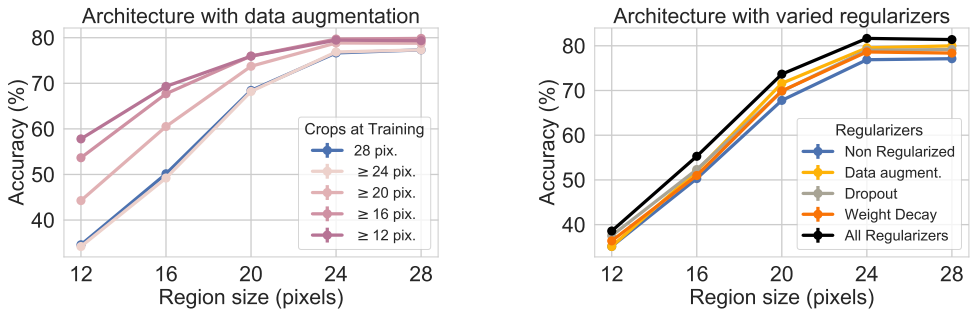
Finally, we show qualitative examples to illustrate that FRIs are a concern for computer vision systems based on similar CNNs architectures. We conducted a test on detection algorithms, to validate that FRIs dramatically affect both the location and the label of the detected objects. In Figure A-4 we show FRIs for the widely used object detector called "YOLO" [18].

3.2 Fragile recognition with data augmentation and regularization

We now investigate if data augmentation and regularization help alleviate fragile recognition. In this experiment we use the CIFAR-10 dataset [14], which contains 10 object categories, $50,000$ training images and $10,000$ testing images of size 32×32 pixels. The evaluation criteria is top-1 accuracy. We use CIFAR-10 for convenience and without loss of generality, since the data augmentation and regularization we test are used in the models tested in ImageNet.

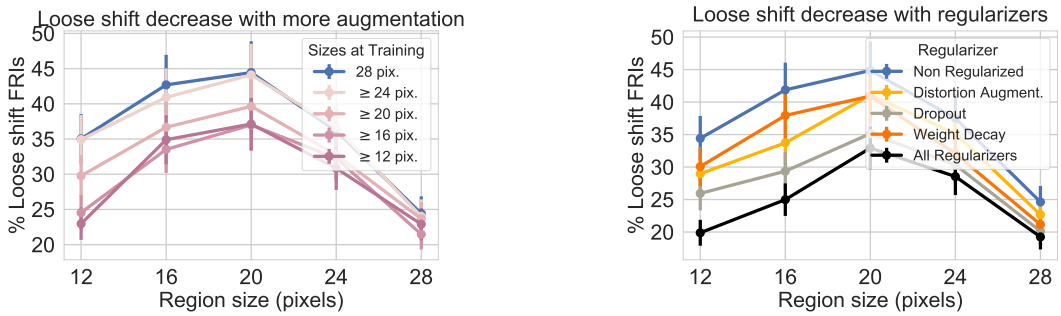
We reproduce the AlexNet version for CIFAR-10 introduced by [27], which consists of two convolutional-pooling-normalization layers followed by two fully connected layers. In the following, all regularizers and data augmentation are turned off unless stated otherwise. We will analyze the impact of each to fragile recognition.

For data augmentation, we augment the training dataset with FRI regions of at least a given size. For regularizers, we add weight decay, dropout, and distortions (e.g. reflections, altered brightness, altered contrast). Both the data augmentation and the



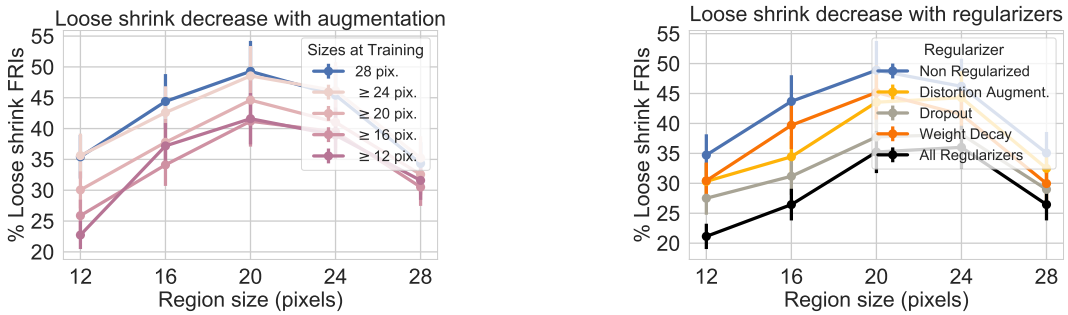
(a) accuracy data augmentation (b) accuracy regularizers

Figure 3-4: Accuracy for data augmentation and regularizers (CIFAR-10) (figure source: [21]).



(a) data augmentation by cropping image regions (b) CNN with regularization

Figure 3-5: Impact of data augmentation and regularization in fragile recognition (CIFAR-10) (figure source: [21]).



(a) loose shrink FRIs with data augmentation (b) loose shrink FRIs with regularization

Figure 3-6: Impact of the data augmentation and regularization in fragile recognition (CIFAR-10) (figure source: [21]).

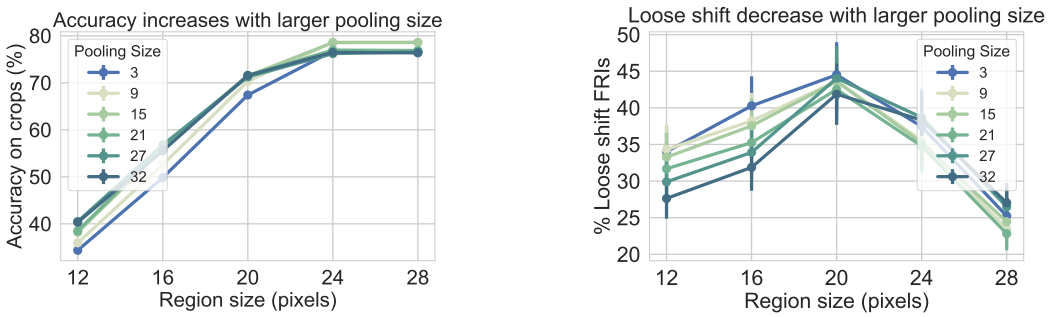
regularizers improve the accuracy of the CNN for the different crop sizes (Figure 3-4). Figure 3-5 shows results for loose shift FRIs and Figure 3-6 shows results for shrink FRIs). In each case, (a) shows that augmenting the training set with crops of FRI sizes (i.e. candidate FRIs) reduces overall FRI occurrence, but many remain; (b) shows that commonly used regularizers may also reduce the absolute percentage of FRIs, but will not eliminate the phenomenon. Both data augmentation and regularization have a clear impact on FRI occurrence in all cases. The largest general improvement comes for the 12-pixel region size, as data augmentation and regularization both lead to decrease of more than 12% of FRIs. For regularizers specifically, dropout provides the most individual improvement. However, all of these generalization efforts still allow a high failure rate. They are also unable to bring the FRI effect to the human level, as the CNN FRIs are indistinguishable for humans. This is further discussed in Chapter 5.

In ImageNet, smaller region sizes lead to higher FRI occurrence. In CIFAR-10, FRI occurrence in very small regions (smaller than 20 pixels) decreases with size. This is because 12- and 16-pixel regions are prohibitively small, so the number of correctly classified regions is severely reduced; consequently, there are fewer opportunities for FRIs to occur at all.

3.3 Fragile recognition is not lack of object location invariance

We now focus on the relationship of fragile recognition with recent works that show that CNNs are affected by small changes of the object location, scale and orientation [10, 3]. The phenomenon described in previous works is referred to as lack of invariance due to affine transformations of the object. Here, we focus only on location changes, as it will allow to distinguish FRIs and these previous works.

The procedure to evaluate location invariance is the following, quoting from [3]: "we embed the original image in a larger image and shift it in the image plane (while



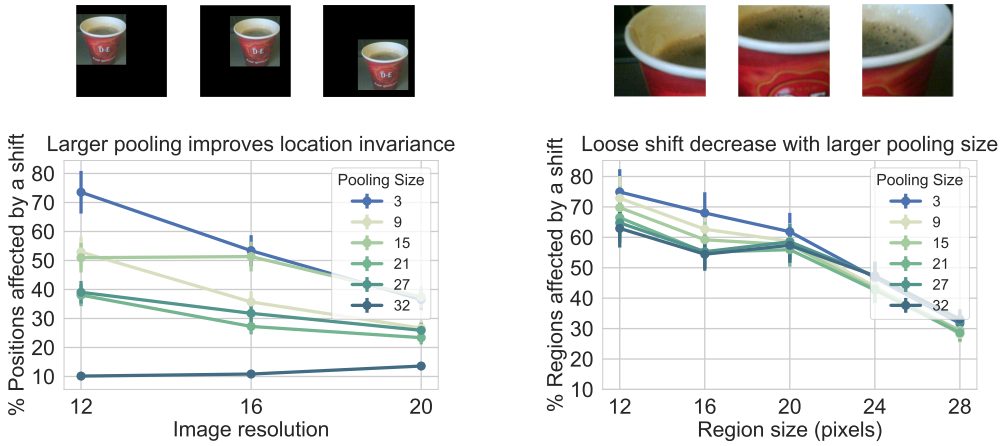
(a) accuracy of CNN on small crops (b) changes in correctness due to small shifts

Figure 3-7: FRIs of architectures with large pooling regions (figure source: [21]).

filling in the rest of the image with a simple inpainting procedure)". The embedding of the image can also be done with a black empty background as in [10], or using videos in which the background is static and only the object is moving [3]. In fragile recognition, both the object and the background, i.e. the entire image, change due to the shift or shrink of the image region. In previous works, only the object changes location. We introduce an experiment that reveals that this subtle difference makes fragile recognition an independent and more complex phenomenon than lack of location invariance.

Pooling induces location invariance. CNN architectures with large pooling regions are known to induce invariance to the object location, cf. [2]. A pooling layer operates independently for each feature by extracting the maximum activation across the spatial dimensions. For a pooling region of the same size as the full image, the network response is invariant to the object position within the image frame. This is under the assumption that there is enough spatial resolution in the responses in order to guarantee that the responses after the shift do not vary except for the shift. See [3] for a detailed explanation. Note that this assumption is not fulfilled in FRIs, as a shift of the image region introduces and removes patterns in the borders of the image, which can produce changes in the responses after pooling. We now show this experimentally.

Experimental results. We evaluate the network trained on CIFAR-10 and introduced in the previous section with different pooling region sizes for the second pooling



(a) output label changes due to shifts in object position

(b) output label changes due to shifts in visible region position

Figure 3-8: Comparison of location invariance and fragile recognition images (CIFAR-10) (figure source: [21]).

layer. These sizes range from three pixels to the entire image; theoretically, this should produce full location invariance. In order to guarantee sufficient spatial resolution to obtain location invariance, we use a stride of one pixel for the convolutional layers and add padding. In figure 3-7 we see that pooling has some mitigating effect on FRIs for smaller-sized crops, but does not significantly reduce them. (a) shows that a larger pooling size has little effect on CNN accuracy on crops, while (b) shows that a larger pooling size hardly reduces the occurrence of FRIs. Figure 3-7a demonstrates that adding larger pooling regions does not reduce network accuracy, but as we show next, it massively reduces the lack of location invariance.

We evaluate the CNN by embedding the object in a black background at all possible locations, and we quantify for how many locations a one-pixel shift of the object location produces a change in the output of the CNN (similar to the method in Chapter 2). In this experiment we evaluate changes to the output label and not correctness, in order to accurately match the definition of invariance.

In Figure 3-8, we see that FRIs are distinct from translation-based adversarial examples. (a) shows that pooling the entire input makes the CNN almost completely robust to translation by maximizing location invariance; the remaining 10% of translations that do cause variation are due to edge effects. This is seen in 3-9, which shows maps that are equivalent to FRI maps, except each pixel corresponds to a downscaled

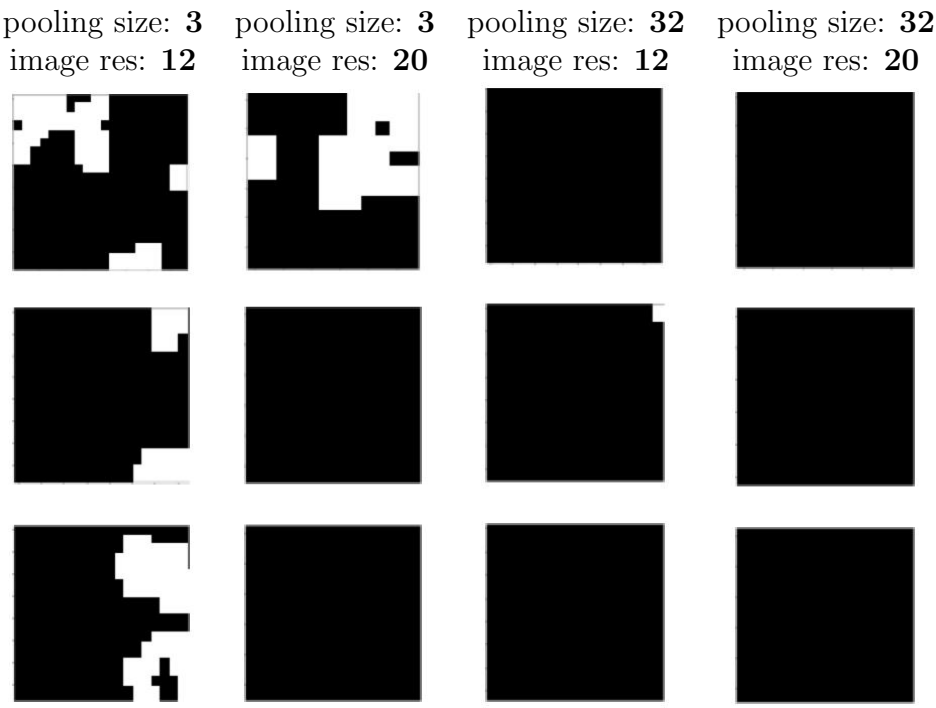
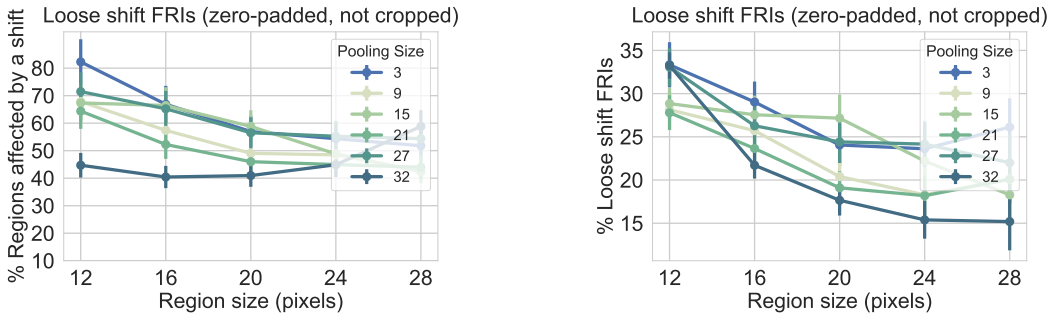


Figure 3-9: Effect of pooling to location invariance (figure source: [21]).

image with a given side length centered at that pixel and zero-padded, rather than a crop centered at that pixel. As seen in row 2 and column 3, most white pixels, i.e. translations for which shifts cause changes in classification, are at the edges.

Figure 3-8a shows that as expected, the lack of invariance is dramatically reduced when using large pooling regions. For embedded 12-pixel images, 75% of the one-pixel shifts affect the network with small pooling regions, while only 10% of the one-pixel shifts affect the output of the network. Pooling is not completely location invariant due to boundary effects near the perimeter of the image. We show qualitative examples of this reduction in Figure 3-9.

In Figure 3-8b, we see that increasing the pooling size only slightly decreases FRIs. The region size with a larger decrease is 12 pixels (note that the embedding size can not be directly compared with the region size of FRIs). For this region size, there are approximately 10% less FRIs, but a significant amount still remain (about 65% of the regions are FRIs). Since the pooling mechanisms that largely reduced the lack of invariance are not effective for fragile recognition, we can conclude that FRIs are a more complex phenomenon than lack of location invariance.



(a) output label changes due to shifts with **zero-padding**

(b) changes in correctness due to small shifts with **zero-padding**

Figure 3-10: FRIs with zero-padding for CNN with large pooling regions (figure source: [21]).

Note that for small region sizes the amount of FRIs increases, which is different from what we observe in the data augmentation and regularization experiment in the previous section. This is because in Figure 3-8b we report change in the output label rather than change in the correctness as in previous experiments. See Figure 3-7b for the effect of pooling on FRI occurrence, which is in accordance to previous results.

Finally, we control that the differences we observe between lack of invariance and FRIs are not caused by use of zero-padding in one case and not the other. In Figure 3-10, we evaluate FRIs with zero-padding instead of up-scaling the cropped region. Figure 3-10a shows that pooling has some mitigating effect on FRIs with zero-padding for lower crops, but does not significantly reduce them. Figure 3-10b shows that a larger pooling size provides little to no reduction of FRIs depending on crop size. The results support the same conclusions as for FRIs with up-scaling.

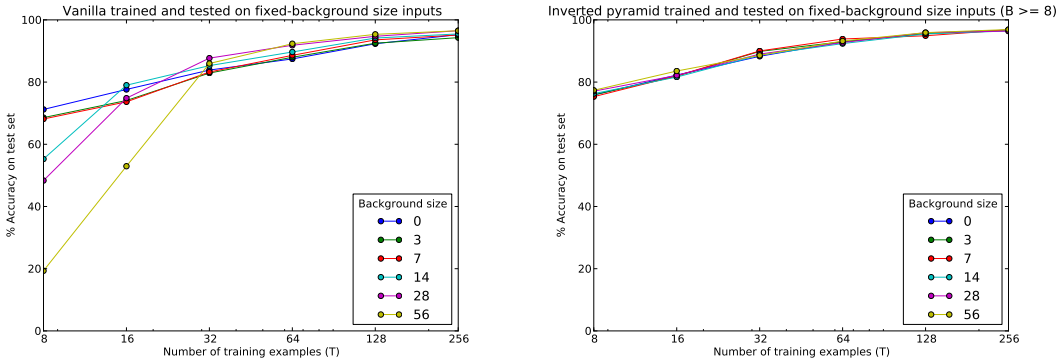
Chapter 4

Inverted pyramid as a solution to useless background features

In this section, we will first discuss the superiority of the inverted pyramid's architecture to the vanilla architecture in terms of accuracy when trained on small amounts of data, particularly when there are many useless background features. We will then perform a deeper analysis of how the inverted pyramid improves CNN robustness in variable conditions.

4.1 Inverted pyramid reduces amount of required training data

In Experiment 1 described in section 2.2, we train and test the vanilla and inverted pyramid architectures on images that all share the same predetermined background size B, i.e. all of them contain an object padded by random background pixels with a padding size of B in each direction. Let the original object side length be s; in MNIST, we consider this to be a raw image because MNIST images are all 28×28 pixels and contain a full-size, centered object. As a result, s in all experiments is 28. Note that the final input for the vanilla DNN contains a resized object based on B and the required input side length S. The networks are trained on various



(a) vanilla (b) inverted pyramid

Figure 4-1: Vanilla (a) and inverted pyramid (b) trained and tested on fixed B.

numbers of training examples T. They are optimized over learning rate (six values tested: $\{10^{-1}, 10^{-2}, ...10^{-6}\}$), and all results in this report come from the model with the optimal learning rate for given experimental parameters. Each MNIST model is trained for 30 epochs.

In this experiment, we consider $\{B|B = \lfloor 2^i s \rfloor; i \in [-3..1]\} \cup \{0\} = \{0, 3, 7, 14, 28, 56\}$. $B > 56$, i.e. $S > 140$, were not used because they were found to be excessive for the standard MNIST CNN architecture. For the vanilla architecture, we consider $\{T|T = 2^i; i \in [3..8]\}$; for the inverted pyramid architecture, we consider $\{T|T = 2^i; i \in [0..8]\}$. We do not use $T > 256$ because the accuracy gain with $T = 256$ over $T = 128$ is low, and both accuracies are high. The discrepancy between the lower boundary of T for vanilla vs. inverted pyramid is because of the comparative robustness of inverted pyramid, and will be detailed below.

When small amounts of training data are present, the inverted pyramid architecture is able to maintain robustness to any of the tested values of B. As seen in figure 4-1, the vanilla CNN performs consistently worse than the inverted pyramid CNN except at the highest T. While this may be a matter of a few percentage points for $T \geq 32$, at small amounts of data, there is a significant margin. This is true across different background sizes; no matter how much distracting background is added, the inverted pyramid architecture is able to maintain relatively high accuracy. Comparing the higher background size values, $B = 14, 28, 56$, we see that the vanilla architecture

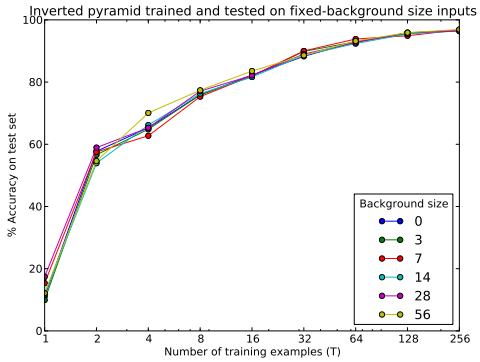


Figure 4-2: Inverted pyramid performance including very small T.

is quite unsuccessful without several training examples, but the inverted pyramid is both high and consistent.

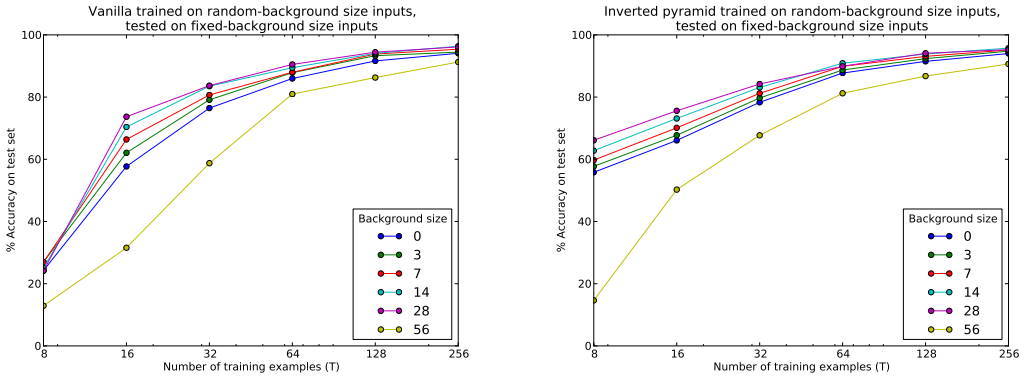
The above figures demonstrate that the inverted pyramid CNN converges and performs with high accuracy even when given small amounts of training examples. The reasonably high performance of inverted pyramid for all B with just eight training examples per class motivates us to record its performance at $T = 1, 2, 4$. In the case of the vanilla architecture, the (qualitative) direct relationship between B and the T at which accuracy spikes is already evident in 4-1a so testing smaller T seems superfluous. The results, shown in figure 4-2, validate the idea that the inverted pyramid architecture reduces training data complexity: even for $T = 2, 4$, accuracy is already at a comfortable point and increasing steadily. The sudden gain happens between $T = 1$ and $T = 2$. The inverted pyramid CNN is superior to the vanilla CNN, having higher accuracy at small T and never performing below the vanilla baseline. We see that this becomes increasingly true with larger B. As demonstrated in section 2.2, reduced useless background features imply reduced training data complexity. Here we show empirically that inverted pyramid architecture reduces training data complexity. Because inverted pyramid inputs are built from vanilla inputs and the experiments are identical in every way except this preprocessing, we investigate whether this reduced training data complexity is observed because the inverted pyramid architecture reduces useless background features.

4.2 Inverted pyramid vs. vanilla: learning to eliminate useless features

In Experiment 2 described in section 2.2, we train the network on images that have a random value for B, selected from a uniform distribution in $[0..\frac{S-s}{2}] = [0..56]$. As a result, the images all have side length 140 as required by the vanilla architecture. The smallest object a network may see is 28×28 pixels with $B = 56$, and the largest is 140×140 pixels with $B = 0$. We consider these random-background size objects to be an approximation of natural images because the location and content of the background is completely unknown. The networks are trained with $\{T|T = 2^i; i \in [0..8]\}$.

Training CNNs on inputs with random B demonstrates how the vanilla and inverted pyramid architectures deal with useless background features in general. When B was constant across all train and test images, there was potential for the networks to simply learn the exact location of the object vs. the background region. Though direct comparison demonstrated that inverted pyramid was inherently advantageous, the trained networks would generalize poorly to unpredictable (i.e. realistic) background. Training on random B forces the CNN to learn to ignore the useless background features themselves. We test these CNNs trained with random B on fixed-B inputs so that we know which parts of the inputs are supposed to be ignored. Along with evaluating their overall performance, we can visualize activations generated during test runs and evaluate the CNN's treatment of the background quantitatively, since we know the locations that should be downgraded.

Inverted pyramid is again shown to be more robust than vanilla. Figure 4-3 compares the performance of the vanilla random-B CNN (4-3a) with the inverted pyramid random-B CNN (4-3b). Inverted pyramid is generally more consistent across various background sizes for any given T, has higher accuracy than vanilla at lower T, and has comparable accuracy at higher T. For lower T, the vanilla CNN drops off sharply in accuracy while the inverted pyramid stays more robust, though it also experiences a significant drop in performance. This is true even for $B = 56$, which is



(a) vanilla (b) inverted pyramid

Figure 4-3: Vanilla (a) and inverted pyramid (b) trained on random B, tested on fixed B.

a clear outlier in terms of performance. The reasoning for this is discussed below.

For small T, the vanilla architecture is more consistent across B when trained with random B as compared to training and testing with a fixed B. When there is a small amount of training data, the random-B vanilla CNN is more robust to B than a vanilla CNN trained on images with a large fixed B. This suggests that the random-B CNNs are more general and therefore robust to varied amounts of useless background features, even if they do not achieve the same accuracy in basic conditions. Inverted pyramid is even more consistent; the inverted pyramid CNN trained on random B does not diverge as much when tested on various fixed B as the vanilla CNN does, for a given T. This supports the idea that the inverted pyramid is more robust to useless background features, because when it is trained on random B, it performs better and more consistently on test inputs with various B. Its significant advantage over vanilla for small T is the outcome we expect if we conclude that it is reducing useless background features.

It is important to note that the random-B inverted pyramid CNN is not as consistent when tested on various B as the fixed-B inverted pyramid CNN; the curves in 4-1b are closer together. However, in both the vanilla and inverted pyramid random-B cases, the order of the accuracy curves is the same for all but some cases at large T. As seen in 4-3, both random-B CNNs perform best on inputs with $B = 28$ and performance drops slightly as B decreases, except for a large drop when $B = 56$. The

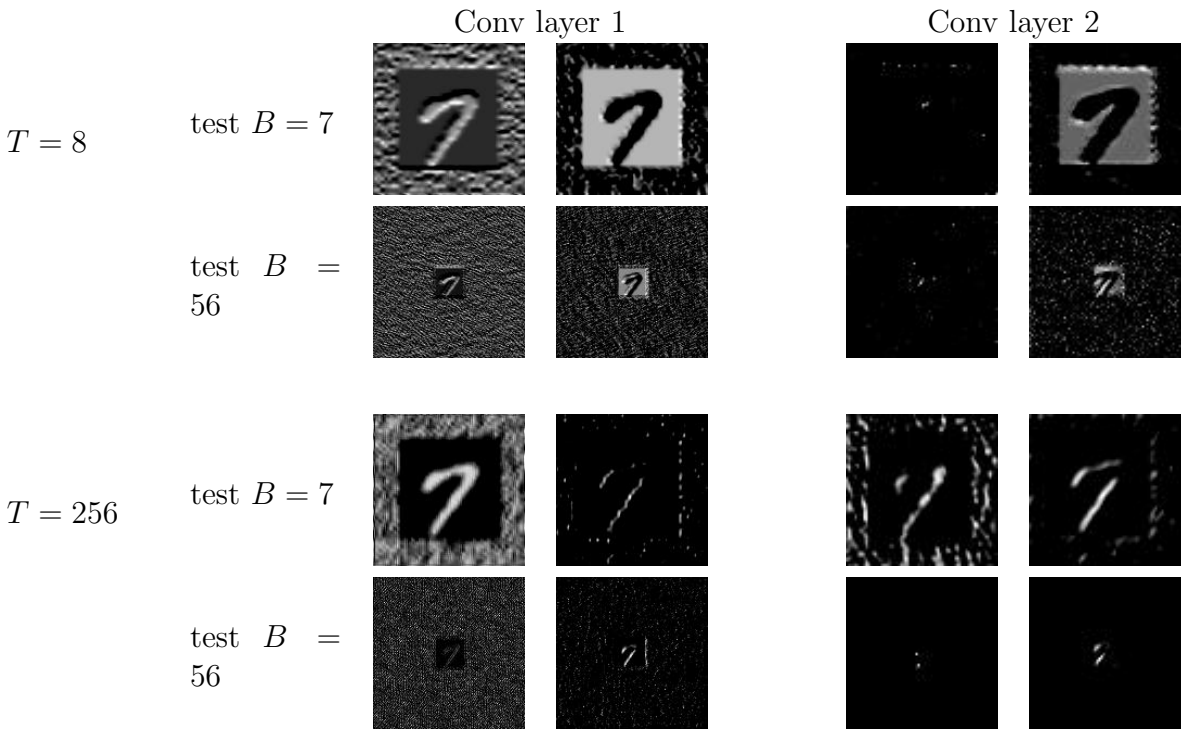


Table 4.1: Sample feature maps from vanilla CNN convolutional layers.

general direct relationship between B and performance accuracy makes sense: if the random-B CNN factors location into its approximation of background features at all, then input images with smaller objects are more likely to have the entire object (or at least, more of it) preserved through the CNN's convolutional layers during a test run, as the objects are centered. By observation, the obvious exception is $B = 56$. In this case, the original MNIST image is not resized at all because adding a 56-pixel-width frame to a 28×28-pixel image already brings the image to the input size of 140×140 pixels. As explained above, the $B = 56$ case contains the smallest object a random-B CNN could possibly see, and it may never see an image with such a small object. As a result, a random-B trained CNN in our experiments may not know how to isolate it. To test this analysis, further exploration will include allowing the random-B generation process to include $B > 56$.

By looking at activations from the CNNs' convolutional layers, we gain a qualitative understanding of how the CNNs handle useless background features. Table 4.1 shows selected outputs of the first and second convolutional layers of the vanilla

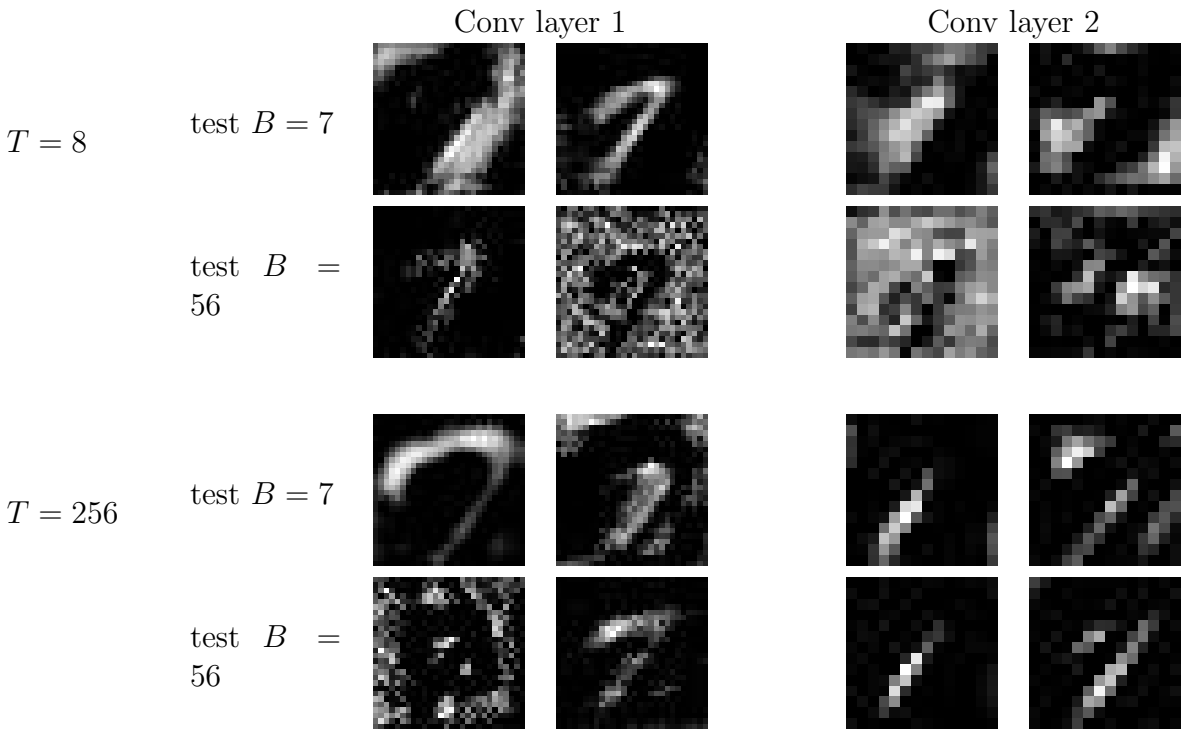


Table 4.2: Sample feature maps from inverted pyramid CNN convolutional layers.

architecture trained with random B and tested with fixed B, i.e. activations from Experiment 2. Table 4.2 shows the same for the inverted pyramid architecture. In each table, we use one CNN trained with $T = 8$ and one trained with $T = 256$; these CNNs are tested on inputs with $B = 7$ and inputs with $B = 56$. For each combination of architecture, T, testing B, and convolutional layer, two feature maps have been sampled. For each architecture, T, and convolutional layer, the testing $B = 7$ and testing $B = 56$ samples in a single column are the same feature map, i.e. the same convolutional filter has been applied to the two different B. Beyond this relationship, all feature maps are unrelated to each other both in this table and in the CNN itself. All feature maps have been pulled for the same original MNIST image, which was classified correctly in all test runs sampled from here.

In table 4.1, we see that the object size is maintained in nearly all feature maps; the only exceptions may be some feature maps from the second convolutional layer where so few pixels are activated that the object size is no longer clear. We observe that with large T and testing B, the random background pixels are smoother in the

first convolutional layer and minimized in the second convolutional layer. For small T and large testing B, the background region seems to have high variance in the first convolutional layer. This seems to continue into the second layer, particularly in the feature map that keeps some salience in the object region and does not simply downgrade the vast majority of the image. From the smoother background regions of the $T = 256$ feature maps, we see that more training data enables the CNN to learn that the background region, though highly variant, does not actually contain salient information. In the second convolutional layer particularly, the background region not only smoothens but darkens considerably. This implies weights for the background region that are closer to zero, which fits with the mathematical justification for this CNN's comparatively high performance.

In table 4.2 we see one similar trend: with higher testing B, the feature maps have more noise surrounding the image. However, the way this manifests is somewhat distinct: where the vanilla feature maps showed this noise to be in the expected background region that corresponded with the original image's background region, the feature maps from the inverted pyramid trained with $T = 256$ often show this unsmoothed, undarkened noise right around the handwritten digit itself. This cannot be directly from the input image, as digits in MNIST are written in a solid black background as seen in 2-3. This is due to the fact that even though the inverted pyramid takes an input with n channels (n being the pyramid depth as defined in section 2.2) and the first convolutional filter therefore has n channels, the filter's output is still one value, so each output channel from the first convolutional layer belongs to an independent feature map that is constructed from a combination of the n input channels. The inverted pyramid is therefore able to isolate the contours of the digit itself and treats the added useless background features much like the black background from raw MNIST, unlike what we see in many of the vanilla feature maps. These digit contours seem to be found in every case except small T/large testing B, which was shown in figure 4-3 to have particularly low performance for both architectures. An important future step is to visualize activations when testing these CNNs with inputs for which $28 \geq B < 56$. This is because testing on $B = 28$

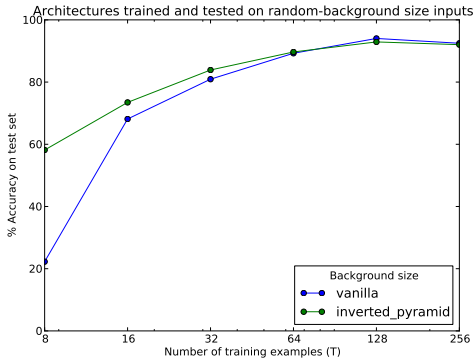


Figure 4-4: Comparison of vanilla and inverted pyramid when trained and tested on random B.

did not experience the massive performance hit and these inputs are more likely to be in range of B that the random-B CNNs have seen while still having small objects.

The key difference seen when comparing table 4.1 and table 4.2 is that the inverted pyramid seems to show more scale invariance. The integration of the multiple scales performed by the first convolutional filter seems to result in the full object being isolated and added background being largely eliminated. Some cases do not quite show this scale invariance, such as the left pair of feature maps for inverted pyramid at $T = 8$, sampled from the first convolutional layer: the $B = 7$ map seems to show a more zoomed-in view of the digit than the $B = 56$ feature map. Nonetheless, the other pairs show digits of similar sizes, and even that example does not reflect the magnitude difference that any one pyramid layer would have for $B = 7$ vs. $B = 56$. We thus see qualitative evidence that inverted pyramid allows the CNN to isolate the object more easily by raising the likelihood that it will see a view with few background pixels and a sizeable object. A key next step is to assess this effect quantitatively.

Finally, we test both random-B trained architectures on random-B test images, approximating natural images as closely as possible (experiment 3 as described in section 2.2. Figure 4-4 shows that particularly for small values of T, inverted pyramid performs better than vanilla and does not experience sharp drops in performance at the values of T tested. This supports the hypothesis that inverted pyramid has better general robustness to useless background features and can learn to downgrade them

with fewer training examples. There is a large margin for improvement in both cases and as a result, the vanilla CNN catches up to the inverted pyramid CNN at smaller T than in experiments 1 and 2.

Chapter 5

Discussion

5.1 Comparing fragile recognition in humans and CNNs

This section appears in [21].

In this section, we further compare the FRIs found for CNNs with the human minimal images found by [26]. Recall that minimal images are equivalent to strict shrink FRIs of small size. Both humans and CNNs are susceptible to small image changes, but these two sets of fragile images are not necessarily the same in humans and in CNNs. As shown by [26, 4], when CNNs are trained and tested on human minimal images, i.e. images that humans can still recognize, CNNs are unable to recognize the objects.

Here we show the converse, namely that when humans are tested on CNN fragile recognition images, they do not exhibit the same fragile response. This can be seen in our qualitative examples: CNN fragile recognition images and their incorrectly-classified counterparts are difficult to distinguish. To verify this, we randomly sample 40 CNN shrink FRIs generated from ImageNet. We present the correctly-classified FRI and the slightly changed image (both resized to 100×100 pixels) to separate groups of 20 subjects in Mechanical Turk [6]. The subjects annotate the images using one of our 10 supercategory labels. The results show a small gap in correct

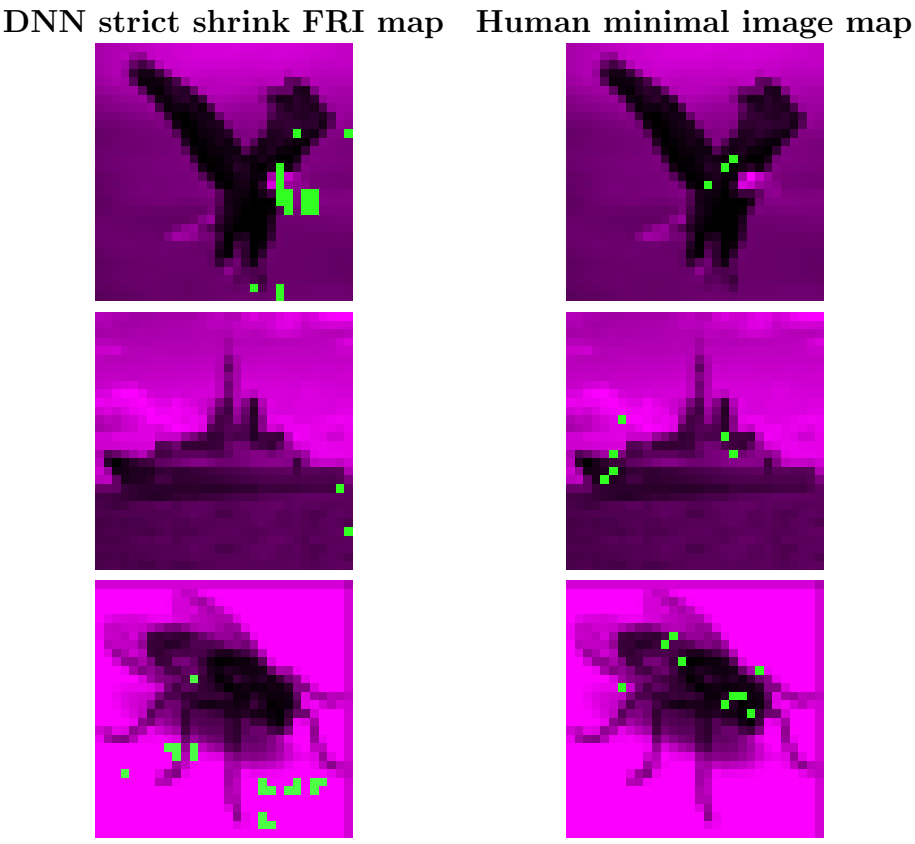


Figure 5-1: Comparison of DNN and human minimal image maps (figure source: [21]).

recognition: on average, success rates among humans for CNN fragile recognition images and incorrect counterparts differ by 14.5%. CNNs experience an average gap in confidence of 56% for the same pairs.

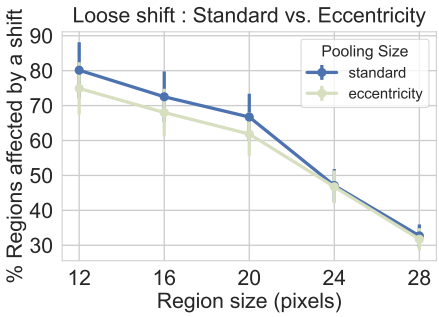
Furthermore, we directly compare human and CNN minimal images of the same image. We use six of the original images from [26] that were used to find human minimal images. We identify the CNN FRIs for these. Comparing CNN FRIs and human minimal images of the same region size, P, we find two additional differences: first, CNNs have more FRIs (5.3 minimal images per image for humans on average, 13.4 for CNNs). Second, CNN FRIs differ in location within the object region (Intersection Over Union between human minimal image maps and CNN FRI maps was small, $< 6\%$ for all tested images). We see examples in 5-1, which shows human minimal images and the equivalent for CNNs (strict shrink FRIs) for an image of an eagle. The left column shows CNN strict shrink FRI locations in green; the right column shows

human minimal image locations in green. Both show non-minimal image locations in pink. While human minimal images are centered in meaningful object parts (e.g. the eagle head and wing in row 1, column 1), CNN FRIs contain mostly a background set of pixels (row 1, column 2). Note that the human minimal images shown have $0.35 \leq P \leq 0.45$, and the DNN FRIs shown have $P = 0.4$ to maximize comparability. More minimal images/FRIs exist for both perceptual systems at other scales.

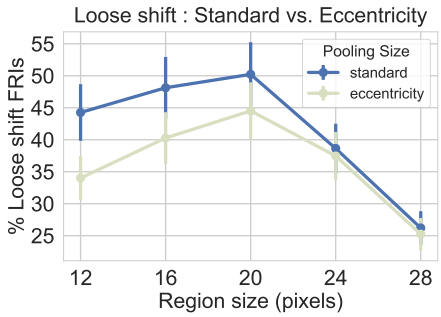
5.2 Eccentricity dependence as a background robustness method

In this section, we evaluate eccentricity dependence as a strategy to improve CNN robustness to background features that do not positively contribute to object recognition. Our approach involves the inverted pyramid architecture with a depth of five, and our results show that it provides an advantage over a vanilla CNN in two key ways: reducing required training data, and improving robustness through better scale invariance. We see from the results in 4.1 that when faced with the same amounts of useless background features, the inverted pyramid architecture can generally learn the various MNIST classes with far fewer training examples than the vanila architecture can learn with. Direct comparison of the two architectures' performance on various amounts of useless background pixels (B) and numbers of training examples (T) shows the significant advantage of inverted pyramid for all but the largest tested amounts of data. Furthermore, when the two architectures' accuracy completely converges, the accuracy values themselves are $\geq 90\%$ and there is limited margin for more concurrent growth; the inverted pyramid architecture holds some advantage for nearly all possible performance gain.

In section 2.2, we established that for a fully connected layer, reduced useless features in training data implied less training data needed to learn high-performing weights. Because convolutional layers can also be written as matrix multiplications, we hypothesized that an architecture that eliminated useless background features in



(a) output label changes due to shifts in visible region position for multi-scale eccentricity



(b) loose shrink FRIs with multi-scale eccentricity

Figure 5-2: Impact of multi-scale eccentricity dependent architecture (CIFAR-10) (figure source: [21]).

images would reduce training data needed for a CNN to perform well. The results describe above indicate that the inverted pyramid model of eccentricity dependence satisfies the consequent of this implication.

The robustness advantage of the inverted pyramid architecture is demonstrated in the generally superior performance of inverted pyramid architecture trained on images with random amounts of background, as presented in section 4.2. We see that the random-B trained inverted pyramid CNN is more accurate than the vanilla CNN for any given T, and shows more consistency when tested on inputs with various B. Furthermore, activations from the inverted pyramid CNN's convolutional layers seem to show more scale invariance than those from the vanilla CNN's convolutional layers. Qualitative analysis shows that the inverted pyramid preprocessing provides CNNs with more opportunity to get a useful view of the object and eliminate the useless background regions. We therefore conclude that the inverted pyramid model satisfies the antecedent of the aforementioned implication as well: inverted pyramid needs less training data because it effectively reduces useless background features, evidenced by the improved scale invariance.

We thus have reason to believe that the inverted pyramid can effectively reduce useless background features in images, improving robustness and reducing required training data complexity. However, this robustness to background features does not extend to minimal images, which can partially be considered a failure of imperfect

boundaries. Figure 5-2 shows occurrence of FRIs when using a two-layer inverted pyramid architecture. We use one layer that covers 20×20 pixels and one that covers the entire image at half resolution, and the two scales are combined by the first convolutional layer. The two-layer pyramid does not provide much of a solution to the minimal image problem, though the deeper one provided significant robustness to useless background features even with reduced training data. Just as humans are subject to minimal images even with their ability to focus and degrade attention to boundaries of an image, CNNs are subject to FRIs even when observing at multiple scales.

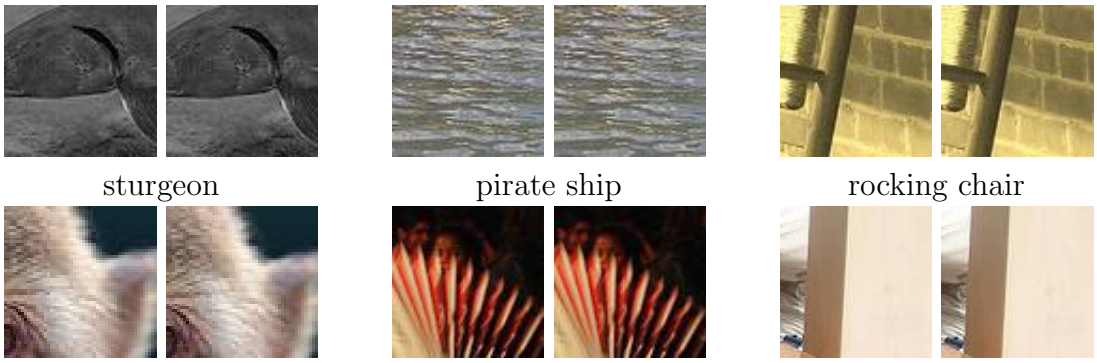
We thus see that humans and CNNs both suffer from minimal images even if they both employ eccentricity dependence. Future steps include using a deeper inverted pyramid, such as the five-layer one used for MNIST in this study, to try and bring CNN FRI occurrence to human level. Though eccentricity dependence cannot solve a simple lack of sufficient information, it does have potential to diminish the overall boundary fragility of CNNs that gives rise to FRIs.

Appendix A

Figures

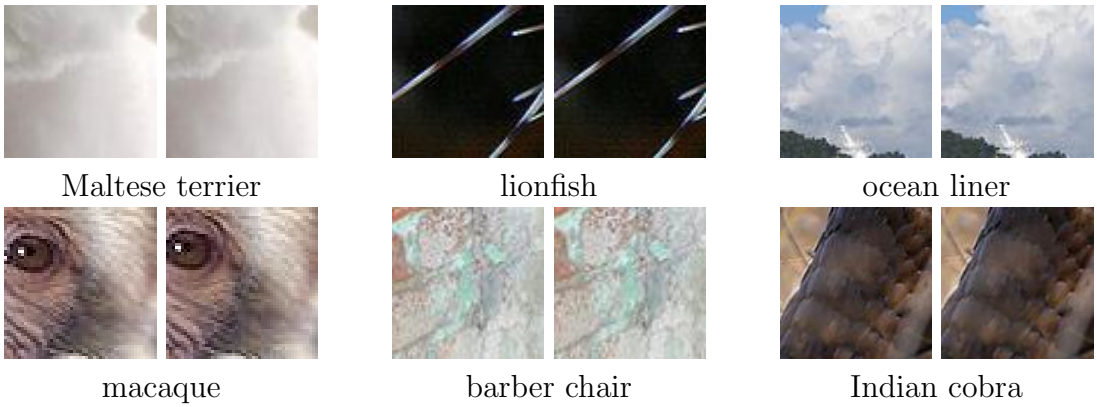
In all example FRI pairs here and below, the left image is classified correctly and the right image is classified incorrectly.

Loose shift FRIs; $P = 0.2$, ResNet



Loose shift FRIs; $P = 0.6$, ResNet

Loose shrink FRIs; $P = 0.2$, ResNet



Loose shrink FRIs; $P = 0.6$, Resnet

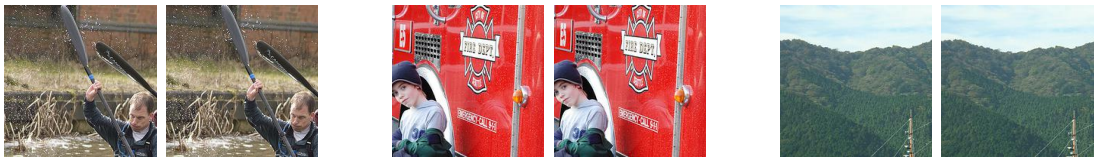


Figure A-1: FRI examples for ResNet (figure source: [21]).

Loose shrink FRIs; $P = 0.2$, VGG-16

thunder snake | titi monkey | saxophone

Loose shrink FRIs; $P = 0.2$, Inception

macaque | freight car | jersey

Figure A-2: FRIs with smaller P contain less of the object (figure source: [21]).

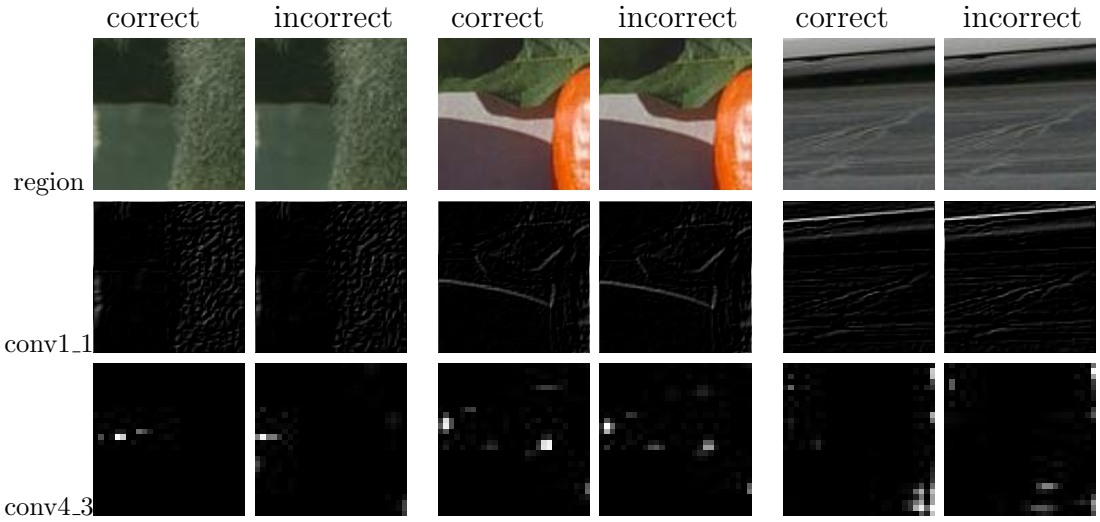


Figure A-3: Activations for loose shift FRIs and their incorrect counterparts (figure source: [21]).
The FRIs were generated using VGG-16 with $P = 0.2$. The first row shows the region themselves, the second row shows the activations from the first convolutional layer, and the third row shows the output from the tenth convolutional layer.

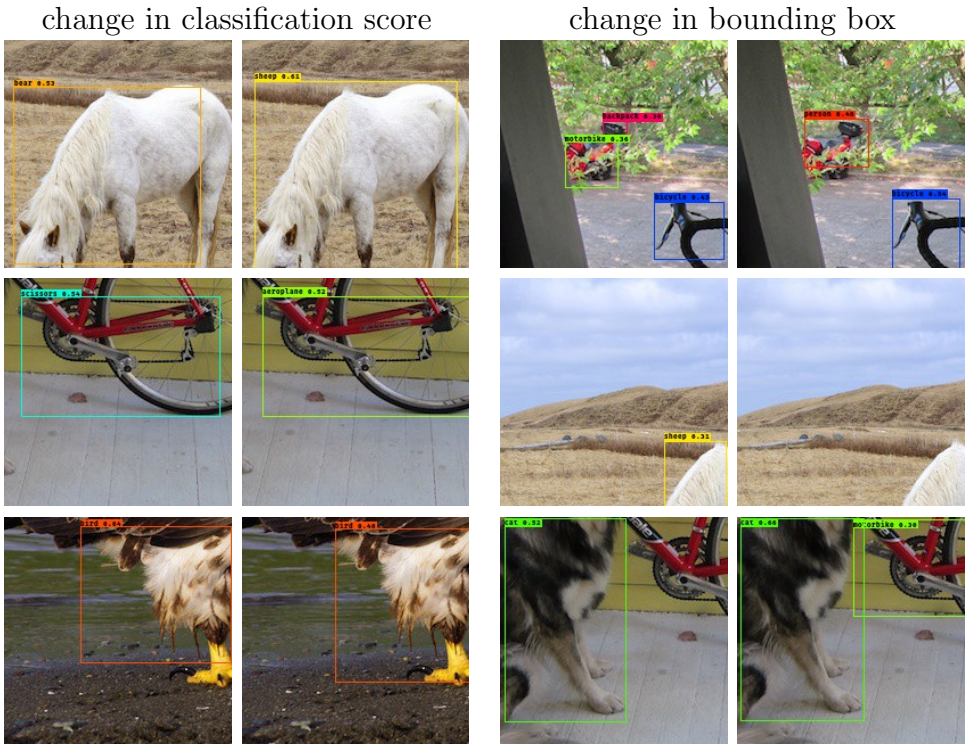


Figure A-4: FRIs for the CNN-based "YOLO" [18] object detection algorithm (figure source: [21]).
These examples were obtained by applying the YOLO algorithm on two adjacent windows of size 200^2 pixels created by 1 pixel shift in the rows dimension. The results demonstrate how detection algorithms are fragile too: the output bounding boxes and their corresponding label scores are dramatically different for these two cropped regions.

Bibliography

[1] Emre Akbas and Miguel P Eckstein. Object detection through search with a foveated visual system. *Public Library of Science Computational Biology*, 2017.

[2] Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. On invariance and selectivity in representation learning. *arXiv preprint arXiv:1503.05938v1*, 2015.

[3] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

[4] Guy Ben-Yosef, Liav Assif, and Shimon Ullman. Full interpretation of minimal images. *Cognition*, 171:65–84, 2018.

[5] Guy Ben-Yosef and Shimon Ullman. Image interpretation above and below the object level. *Journal of the Royal Society Interface Focus*, 8(4):20180020, 2018.

[6] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[7] Francis X Chen, Gemma Roig, Leyla Isik, Xavier Boix, and Tomaso Poggio. Eccentricity dependent deep neural networks: Modeling invariance in human vision. 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: a large-scale hierarchical image database. 2009.

[9] Gamaleldin F Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*, 2018.

[10] Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[12] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[15] Yann LeCun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[16] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[18] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[21] Sanjana Srivastava, Guy Ben-Yosef, and Xavier Boix. Minimal images in deep neural networks: fragile object recognition in natural images. In *International Conference on Learning Representations*, 2019.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842v1*, 2014.

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[24] Jim Mutch Tomaso Poggio and Leyla Isik. Computational role of eccentricity dependent cortical magnification. 2014.

[25] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[26] Shimon Ullman, Liav Assif, Ethan Fetaya, and Daniel Harari. Atoms of recognition in human and computer vision. *Proceedings of the National Academy of Sciences*, 113(10):2744–2749, 2016.

[27] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[28] Zhuotun Zhu, Lingxi Xie, and Alan L Yuille. Object recognition with and without objects. *arXiv preprint arXiv:1611.06596*, 2016.