Deep Residual Learning for Enhanced Image Classification Across Diverse Datasets

Possakorn Kittipipatthanapong
The University of Adelaide, Assignment 2, CNNs for image classification
a1873765@adelaide.edu.au

November 1, 2023

Abstract

This study explores image classification, leveraging the capabilities of deep convolutional neural networks (CNNs) to tackle the challenges posed by varying image complexities. We conducted comprehensive evaluations on three well-known datasets: Fashion MNIST, CIFAR10, and CIFAR100, employing a range of network architectures such as AlexNet, VGG19, and various configurations of ResNet. Our work underscores the efficacy of employing simplified residual architectures in deep networks to facilitate seamless information flow across layers, effectively mitigating the degradation problem associated with increased network depth. Among the diverse range of network architectures evaluated, ResNet18 emerged as the standout performer, demonstrating exceptional performance across all tested datasets. It showcased robustness and computational efficiency, achieving accuracy rates of 93.7%, 89.6%, and 65.2% on Fashion MNIST, CIFAR10 and CIFAR100, respectively. Our findings underscore the critical role of deep residual learning in enhancing the performance of image classification tasks, particularly when dealing with complex image data.

1 Introduction

In today's digital era, where trillions of images are captured annually, the integration of digital tools and computer vision technologies offers numerous solutions to real-world challenges [8][10]. Image classification is one of the potential solutions, aiming to interpret and define the contents of an image, and it holds the potential to empower object detection systems in identifying items within images [15].

Deep learning, particularly convolutional neural networks (CNNs) [11], has brought about groundbreaking advancements in image labeling, object identification, and scene categorization across the world. These developments led to create strategies to address issues related to object detection and scene categorization [21]. When compared to conventional methods of feature extraction and algorithms such as linear classification [18], support vector machines (SVM) [2], or decision trees [13], CNNs offer a superior understanding of the complexities of images. This leads to the discovery of numerous underlying patterns, resulting in enhanced image recognition, segmentation, and detection [16].

Within deep Convolutional Neural Networks, various renowned networks exist, including VGGNet [17], AlexNet [8], and ResNets [5]. These networks demonstrate exceptional ability in accurately classifying images across diverse categories and objects. In this particular study, we have chosen to concentrate on deep residual learning techniques or ResNet, highlighting the benefits of shortcut connections. These connections allow the network to bypass one or more layers, demonstrating that residual networks are more straightforward to optimize and can achieve higher accuracy with significantly increased depth [5].

Some leading competitor studies are dedicated to refining deep neural network performance, specifically through the enhancement of ResNet v2-20 using Mish activation. This approach has yielded encouraging outcomes, achieving an impressive accuracy rate of around 92.02% on CIFAR-10 [14]. Another competitive study have also utilized adversarial model perturbation (AMP), opting to minimize an alternative "AMP loss" through stochastic gradient descent (SGD) on the ResNet model, rather than directly reducing the empirical risk, resulting in an accuracy of 96.03% [20].

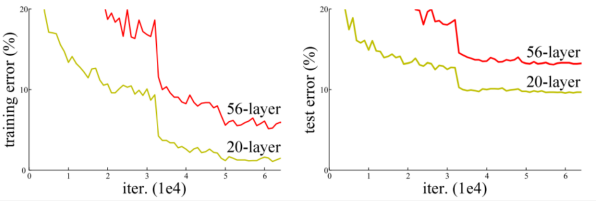

Figure 1: demonstrate the Phenomena of degradation problem with Plain network architectures. Left: training error. Right: test error on CIFAR-10 [5].

2 Method Description

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have established themselves as a powerful tool in the realm of image classification, primarily due to their ability to learn hierarchical features directly from image data [9]. Originating from the groundbreaking work, CNNs utilize convolutional layers to apply filters across input images, capturing local dependencies and reducing the dimensionality of the problem [11]. The architecture of a typical CNN consists of alternating convolutional and pooling layers, followed by fully connected layers for classification [8]. One of the paramount architectures, VGGNet, introduced in the previous sections, employs deep layers with small convolutional filters, demonstrating the benefits of depth in CNNs for image recognition tasks [17]. Residual Networks (ResNets), another milestone in CNN development, address the vanishing gradient problem in deep networks, enabling the training of networks that are substantially deeper than previous architectures [5].

2.2 Residual Learning

The development of deep learning has brought significant growth in various domains, with deeper architectures generally providing superior performance [3]. However, this trend encounters a bottleneck when the network becomes very deep, leading to the degradation problem [1] as illustrated in Figure 1. Contrary to expectations, increasing the network's depth beyond a certain point does not result in continual improvements in performance [4][3]. Instead, it leads to a higher training error as the network struggles to converge, ultimately causing a degradation in performance verified on thoroughly verified experiments [5]. To prevent this situation, we employ the residual learning framework, an innovative approach facilitating the training of networks that are significantly deeper than previously possible [5].
Residual learning introduces a shift from the traditional learning approach of directly learning the desired underlying mapping. Instead, it focuses on learning the residual, or the difference, between the input layer and the desired output. This approach contrasts sharply with conventional methods that attempt to learn the target functions directly without referencing the inputs.

2.3 Residual Block Formulation

To identify the complexity of residual learning, the specific building block, or shortcut connections as shown in Figure 2, was constructed to execute an identity mapping function, and directly add their results to the outcomes of the consecutive layers [5]. Following the underlying mapping in the Residual Learning model as H(x), where x denotes the input to the network, the framework aims to learn the residual function: F(x) = H(x) - x rather than directly learning H(x). Therefore, the original function transforms

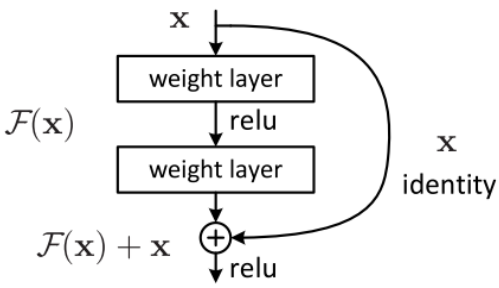

Figure 2: Residual learning: block represent the skip connection techniques which directly fit a desired underlying mapping instead of relying on the performance of several sequentially arranged layers [5].

into F(x) + x as defined as:

$$y = F(x, \{W_i\}) + x$$

where x is the input vector, y is the output vector, and W_i can refer to the weights of the convolutional layers within the residual block.

This formulation intuitively introduces an identity shortcut connection that skips one or more layers, aiding in the training of deep networks. The incorporation of the identity shortcut ensures that the introduction of additional layers does not hinder performance, as these layers can easily approximate the identity function, ensuring that the deeper model performs at least as well as its shallower counterpart.

Moreover, one of the most groundbreaking studies from 2018, Visualizing the Loss Landscape of Neural Nets, by Hao Li and colleagues [12], reveals the significant impact of skip connections in neural networks. These connections contribute to smoothing a smoother gradient flow which is crucial as it helps the model avoid getting trapped in local minima or extremely sharp regions of the loss landscape as illustrated in Figure 3, facilitating easier optimization.

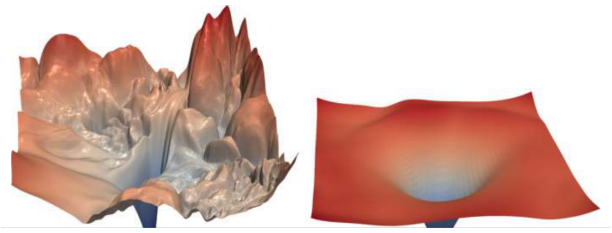

Figure 3: Loss Landscape on Neural Nets with/without skip connection. Left: Loss surfaces without skip connection. Right: Loss surfaces with skip connection [12].

2.4 Network Architectures

The residual network architecture, commonly referred to as ResNet, innovatively integrates shortcut connections that bypass one or more layers, forming a unique structure known as a residual block. Unlike conventional neural architectures, these shortcut connections are not parameterized, ensuring that they do not contribute additional computational complexity to the network [5].

In its design, ResNet employs batch normalization and ReLU activation functions placed strategically before the weight layers, as depicted in Figure 4. This setup is crucial for maintaining healthy gradients throughout the network, facilitating a more rapid convergence during training [6]. By stacking these residual blocks, we form the backbone of our deep networks, allowing for the training of architectures with significantly greater depth than was previously possible. The incorporation of skip connections is vital, as it prevents the vanishing gradient problem, ensuring that gradients can flow through these connections and bypass layers that may struggle with gradient propagation [5].

Looking at other architectures like VGG19 and AlexNet provides context for understanding the evolution of deep learning. VGG19, a deep convolutional network with 19 layers, follows a more traditional approach, emphasizing depth but lacking the residual connections found in ResNet [17]. It consists of

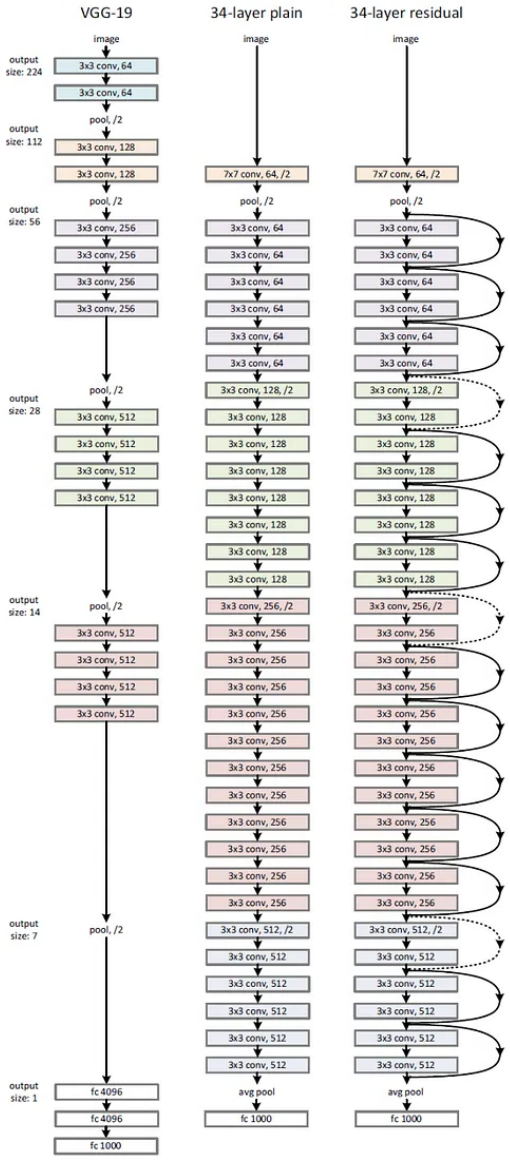

Figure 4: Network architectures on sampling with ImagesNet. Left: VGG-19 model. Middle: Plain network with 34 layers. Right: ResNet or residual network

a series of convolutional and pooling layers, topped off with fully connected layers. AlexNet, a somewhat shallower network, was instrumental in bringing deep learning to the forefront of image recognition, featuring five convolutional layers and three fully connected layers [8].

While VGG19 and AlexNet represented significant progress in deep learning, they also highlighted the challenges associated with training very deep networks, particularly the vanishing gradient problem. ResNet addresses these challenges directly, introducing residual learning and shortcut connections, laying the groundwork for the development of deeper and more efficient deep learning models [1]

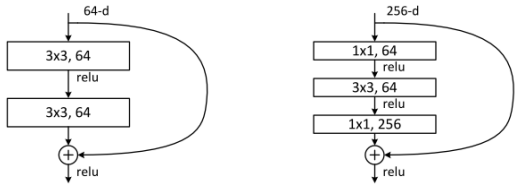

Figure 5: A deeper residual function or block formulation for ResNet-50/101/152. Left: Basic Block,Right:Bottleneck building Block [5].

Deeper Bottleneck Architectures: To further enhance the performance of our deep networks, especially when venturing into architectures with hundreds of layers, we adopt a bottleneck design for our residual blocks. As present in Figure 5, each bottleneck block comprises three layers: the first and third layers perform dimension reduction and restoration respectively, while the second layer performs convolution. This design significantly reduces the computational complexity of the network, making it feasible to train very deep networks. The residual blocks with bottleneck design are pivotal in achieving superior performance on a variety of visual recognition tasks, as they allow for increased depth without a proportional increase in computational complexity. The skip connections play an integral role in these deep bottleneck architectures, ensuring that the introduction of additional layers results in performance gains, or in the worst case, no degradation in performance[5].

3 Method Implementation

In this investigation, we utilize a diverse array of established neural network architectures, ranging from a AlexNet, to the more complex VGG19, and various versions of ResNet, following the methodologies outlined in [5]. The primary focus of our implementation was to assess the performance across different datasets, namely Fashion MNIST [19], CIFAR10 [7], and CIFAR100 [7], which vary in terms of image complexity and both the training and testing datasets were directly acquired from sources.

3.1 Data Pre-processing

To ensure a consistent and fair comparison of outcomes, all images were resized to 32x32 pixels. From these, a 32x32 crop with padding 4 is randomly selected, either from the original image or its horizontally flipped counterpart. Subsequently, the pixel values of the tensors are normalized, bringing them into a range that is more standardized and easier for the model to interpret.

In the next step, we divided the training images into training and validation sets, adhering to a ratio of 0.8:0.2. This division was carried out using a random split function, with a specific seed for reproducibility. Furthermore, a DataLoader was employed to manage the datasets, with a batch size set at 16.

3.2 Network Initialization

The network architectures used in this study include a AlexNet, celebrated for its straightforward design, operational efficiency, and popularity in the domain, ensuring its replicability in various settings following the practice in [8]. We also utilized VGG19 as shown in Figure 4, a deep convolutional network known for its depth and robustness. Additionally, we incorporated various versions of ResNet, renowned for its residual learning capabilities and variable depths, ensuring efficient use of computational resources implemented following [5].

For smaller ResNet layers, such as 18 and 34, the networks leverage basic blocks to facilitate skip connections without the need for additional parameters. On the other hand, larger layers, like ResNet 50, utilize bottleneck blocks within their architecture.

3.3 Model Training

In this phase, we initialized the weights and proceeded to train all the comparing and residual networks from scratch without any pre-trained, following the previously mentioned methodologies. The initial learning rate was set at 0.001, and we utilized the CosineAnnealingLR scheduling technique, which begins with a significantly high learning rate before aggressively reducing it towards a value near zero. The models underwent training for up to 20 epochs.

For the optimization process, Stochastic Gradient Descent (SGD) was employed as the loss function, applying a momentum of 0.9 and a L2 regularization term with a weight decay set at 0.001. It is noteworthy to mention that dropout techniques were not utilized in this particular implementation. In the testing phase, the downloaded data was subjected to normalization, and the trained model was used for evaluation. Finally, the best-performing model was tested across various datasets to observe its accuracy and understand the relationship between the model's performance and the complexity of the input images.

3.4 Implemented Code

The analysis was conducted using **Python**, leveraging the **PyTorch** framework, and **the results with specific detail** have been made publicly available through **GitHub** at the following URL: https://github.com/possakorn/UoA_DL_2023_3_PK (Assignment02_v4.0.0_dataset.ipynb which develop on different data sets)

4 Experimental Analysis

4.1 Image Classifications

Within the confines of this research, we carried out an in-depth evaluation applying our methodology to three distinct datasets. Fashion MNIST comprises 70,000 grayscale images (28×28 pixels each) of fashion articles, distributed across 10 different categories.

This dataset mirrors the training and testing distribution of the original MNIST dataset, with 60,000 images for training and 10,000 images for testing. CIFAR10, a subset of Tiny Images, includes 60,000 color images (32x32 pixels each) categorized into 10 unique classes, with each class containing 6,000 images. The division for training and testing is 5,000 and 1,000 images per class, respectively. CIFAR100, also derived from Tiny Images, consists of 60,000 color images (32x32 pixels each) classified into 100 unique classes, with a distribution of 500 training images and 100 testing images per class.

Following the completion of the training and validation phases, which were based on an initial split designed to optimize the loss functions, we proceeded to evaluate the results with a focus on accuracy.

The primary focus of our experimental setup was to investigate the behavior of profoundly deep networks. In doing so, we made a conscious decision to employ relatively simpler residual architectures. This approach was taken to ensure a smoother transition of information through the deeper layers of the network, providing a strategic solution to the challenges of optimization and the degradation problem commonly associated with such complex structures.

4.2 Analyzing Model Performance Across Differing Image Complexities

The results from various datasets have been meticulously compiled in Table 1, and a comprehensive analysis is provided based on the derived data.

Fashion MNIST Dataset: This particular dataset, comprising 28x28 grayscale images, manifested almost identical performance levels across all the evaluated models. However, a slight edge was noted in ResNet18, which achieved an impressive accuracy rate of 93.6%, while both AlexNet and VGG19 lagged slightly behind, each securing an accuracy rate of 89.4% and 89.6%, respectively. Given their relatively simpler architectural designs, AlexNet and VGG19 managed to perform comparably to one another when tasked with analyzing a straightforward dataset such as Fashion MNIST. In this scenario, the deeper layers and residual connections of ResNet18 were not fully exploited, leading to only a marginal increase in performance.

CIFAR-10 and CIFAR-100 Dataset: Moving on to the more complex CIFAR-10 and CIFAR-100 datasets, the ResNet variants (18, 34, 50) demonstrated consistent and competitive performance in CIFAR-10, achieving accuracies around 90%. This performance totally contrasted with that of AlexNet and VGG19, which experienced a substantial drop in accuracy, falling to 77.6% and 74.7% respectively. Upon shifting to CIFAR-100, a dataset characterized by 100 classes and a more formidable classification task, resulted in a decrease in accuracy for all models. This was especially pronounced for AlexNet and VGG19, with their accuracies tumbling down to 40.0% and 47.9% respectively. Conversely, the ResNet models were able to maintain a relatively higher accuracy of around 66%. The CIFAR-100 dataset, with its high-resolution color images and extensive range of classes, served to highlight the advantages of ResNet's deep architecture and residual connections. These features were instrumental in capturing the complex patterns present in the data, preventing the model from becoming trapped in local minima, and ensuring a superior level of performance, particularly when dealing with the extensive class diversity present in CIFAR-100.

The experimental results clearly illustrate the significant impact that image complexity has on model performance. Simpler models, such as AlexNet and VGG19, are able to compete effectively on datasets with lower levels of complexity. However, their performance tends to deteriorate markedly as the complexity of the task at hand increases. On the other hand, ResNet, with its intricate architecture and innovative implementation of residual connections, consistently demonstrates robust performance across a variety of datasets, irrespective of their complexity. This highlights its suitability for tasks that necessitate the identification of complex patterns and nuanced features within the data. The ability of ResNet to maintain performance levels in the face of increasing complexity underscores the critical importance

Table 1: Accuracy (%) on Fashion MNIST, CIFAR10, and CIFAR100 following the CNN, AlexNet, and ResNet-18/34/50

Evaluate the model performance					
Sources	Alex	Vgg19	Res18	Res34	Res50
# layer	9	19	18	34	50
# params	56M	20M	11M	21M	23M
Model	Model Test Accuracy				
Fashion MNIST	89.4	89.6	93.4	93.7	93.3
CIFAR10	77.6	74.7	89.3	89.6	89.2
CIFAR100	40.0	47.9	65.7	65.2	65.6

of considering both the nature of the dataset and the challenges it presents when choosing a neural network architecture for a specific task.

5 Reflection on project

The experimental investigations conducted in this study provide a clear illustration of the transformative impact of deep residual learning on image classification tasks. By focusing on a range of datasets with varying degrees of complexity, we were able to elucidate the strengths and limitations of different CNN architectures. Simpler models like AlexNet and VGG19 demonstrated competitive performance on less complex datasets but struggled to maintain accuracy as the image complexity increased. In contrast, the ResNet variants consistently delivered superior performance across all datasets, showcasing their robustness and adaptability to complex image patterns.

Our studies highlight the importance of considering the nature of the dataset and the inherent challenges it presents when selecting a neural network architecture for image classification tasks. The consistent performance of ResNet across diverse datasets underscores its utility in applications that require the discernment of underlying patterns and nuanced features within image data. The success of ResNet in this context is attributed to its innovative residual learning framework, which facilitates the training of deeper networks without struggling with the degradation problem.

However, it is crucial to note that this research is not without its limitations. The full potential of the deep learning models could not be completely explored due to computational constraints, preventing us from achieving the highest optimal performance. Additionally, the non-implementation of pre-trained models may have limited our insights and the overall outcomes.

In light of these limitations, future work should focus on extending computational resources to allow for more extensive training iterations, integrating pre-trained models to enhance performance, and conducting a comparative analysis to better understand the impact of these strategies in diverse image classification scenarios. These steps are imperative to further unravel the capabilities of deep residual learning and solidify its application in extracting nuanced features from image data, ultimately contributing to the progression of image classification and deep learning as a whole.

References

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. 2

[2] Mayank Chandra and Sarabjeet Bedi. Survey on svm and their application in image classification. *International Journal of Information Technology*, 13, 01 2018. 1

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington,

editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 2

[4] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. *CoRR*, abs/1412.1710, 2014. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 2, 3, 4, 5

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015. 3

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012. 5

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 1, 2, 4, 5

[9] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. 2

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 1

[11] Yann Lecun, Leon Bottou, Y. Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998. 1, 2

[12] Hao Li, Zheng Xu, Gavin Taylor, and Tom Goldstein. Visualizing the loss landscape of neural nets, 2018. 3

[13] Kun-Che Lu and Don-Lin Yang. Image processing and image mining using decision trees. *J. Inf. Sci. Eng.*, 25:989–1003, 07 2009. 1

[14] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *CoRR*, abs/1908.08681, 2019. 2

[15] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 12 2013. 1

[16] Neha Sharma, Vibhor Jain, and Anju Mishra. An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, 132:377–384, 01 2018. 1

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 2, 3

[18] Michel Vidal-Naquet and Shimon Ullman. Object recognition with informative features and linear classification. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, page 281, USA, 2003. IEEE Computer Society. 1

[19] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. 5

[20] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation, 2021. 2

[21] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 1, 05 2015. 1