The Effect of Image Resolution on Neural Network Performance

Katherine Wu
Stanford University
450 Serra Mall
kjwu00@stanford.edu

Daniel Kang *
Stanford University
450 Serra Mall
ddkang@stanford.edu

Abstract

Neural networks are becoming deeper and more costly to evaluate. One potential way to reduce the computational cost of neural networks is to use images with lower resolutions. The impact of using lower resolution training and testing images on the accuracy of image classification has not previously been studied. We train image classification CNN to accomplish standard fine tuning tasks, using images from several data sets (to represent tasks of various difficulties) at several resolutions, evaluating the CNN using their testing accuracies. We find that the more difficult a task is, the higher the resolution the images the model uses should be: ranging from 64x64 for easier tasks to 224x224 for difficult tasks.

1. Introduction

Many popular image serving websites, including Instagram and Flickr, store several resolutions of images, from thumbnails to full resolution images. However, all classification networks we are aware of use standard image sizes, such as 224x224 for the standard ResNet configuration.

State of the art neural networks are becoming deeper and more costly to evaluate, exemplified by the winning 2014 ImageNet competition neural network by Google, which had 22 layers, in comparison to the wining 2015 ImageNet competition neural network by Microsoft, which had 152 layers. Many are developing less computationally expensive methods which are able to achieve the same goals as the more costly methods. NoScope is a recently developed system which takes a video, object to detect, and reference neural network, and trains a specialized neural network which is specific to the video and object and is able to accomplish the task two to three times more quickly than the reference neural network [11]. Decoding smaller images can dramatically reduce the costs of processing both images and video. Chameleon is a recently developed controller which adjusts configurations, including the resolution of the video, to keep accuracy of the model above a certain threshold, while minimizing the resource consumption. They found that high video resolution allows for accurate object detection, although it requires many resources. The accuracy threshold needed depends on the task: traffic light changes, for example, require, less accuracy than amber alert detection [10].

Decreasing the resolution of training and testing photos would make image classification less computationally expensive, however it is not clear whether or not decreasing the image resolution would affect the accuracy of the classifiers. Several benchmark tasks, such as CIFAR10, show that full resolution images are not necessary for high accuracy on simple tasks. To the best of our knowledge, the effect of resolution on image classification accuracy for different tasks has not been rigorously studied. For example, while images with lower resolution may still capture overall object configurations, it is possible that specific details, such as logos, may become blurred and diminished. With high resolution images, it has been shown that the middle layers of CNNs are used to identify detailed features such as object parts, while higher layers identify the overall object.

In order to create models which are able to accomplish difficult image classification tasks on images with lower resolution, models could be first trained on high resolution photos (e.g. 227 by 227 pixels), and then fine-tuned on training images with artificially lowered resolutions which match the resolution of the testing images (e.g. 50 pixels by 50 pixels). It was found that these models are able to capture mid-level features, distinguishing between different types of birds and different make, model, and years of cars [15].

For this project, we study how resolution affects the classification accuracy for a variety of standard fine-tuning tasks. Many popular networks, including ResNet [6], ResNeXt [19], and DenseNet [8], use a fixed image size of 244x244. However, using images of smaller resolutions could make image classification less computationally expensive. We aim to determine the relationship between im-

*Daniel Kang is not enrolled in CS231N.

age resolution and the performance of CNNs on various types of image classification tasks and provide guidelines on the resolution necessary for different tasks. We find that the more difficult the task, the higher resolution the images the CNN should be trained and tested on: for easy tasks, images with resolutions 65x65 could be used, for more moderately difficult tasks, images with slightly higher resolutions such as 97x97 or 129x129 should be used, and for more complex tasks, standard-sized images (with resolution 224x224) should be used.

2. Related Work

Resolution is being used to scale up CNN models: CNNs are often developed at a lower fixed-cost then scaled up when there are more computational resources available. GPipe, for example, scales up a baseline model four times larger, achieving 84.3% ImageNet top-1 accuracy [9]. Despite demanding more computational resources, using a larger image resolution allows a model to learn more finely gained patterns, which are lost in the lower resolution images. Thus, many models are using images with larger resolutions. In particular, images of high resolutions, such as 600x600, are now used in object detection CNNs [5, 12]. However, it has been noted that as the resolution of the image increases, the increase in accuracy diminishes, as illustrated in Figure 1 [17]. Furthermore, the impact that resolution has on the accuracy of a model may differ for tasks of different levels of difficulty.

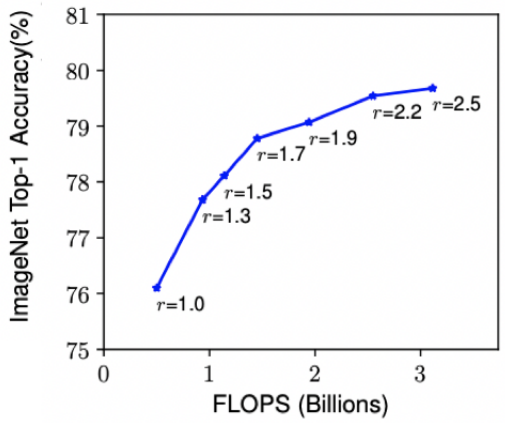

Figure 1. *Using images with higher resolutions increases accuracy, although as resolutions become larger, the accuracy gained becomes smaller. Source: Adapted from [17] Figure 3.*

Others are optimizing models which are tested on images or videos with low resolution. Chameleon adjusts the configurations of videos, including the resolution, to decrease resource consumption during video analysis. They find that a high resolution allows for more accurate object detection in video, although the higher resolution requires more resources, so they adjust the resolution depending on the task, increasing resolution for more difficult tasks [10]. This allows the computational cost for simpler tasks to be diminished, although the cost for difficult tasks remains unchanged. Furthermore, image classification models which are trained initially on images with high resolution, then fine-tuned on training data with artificially lowered resolution (to match the resolution of the photos in the testing data set) are able to relatively accurately complete difficult classification tasks, such as differentiating between different species of birds [15]. This reduces the computational costs, however, high resolution images are still needed to train the model, and the initial training of the model is still computationally expensive, as it is conducted using the higher resolution images.

Using images of a smaller resolution is not the only way in which we could attempt to reduce the computational cost of a model while maintaining the accuracy of the model: we are also able to use knowledge distillation. Knowledge distillation is a method which was designed to improve the performance of deep learning models on mobile devices. Large, complex models are trained to extract features from data. A smaller, compressed network is then trained to mimic the output of the larger network, compressing the knowledge contained in the larger model and achieving similar accuracy as the larger models. This smaller network would then have a reduced computational cost, while still containing knowledge of the larger network. One method to achieve this by using the larger model to label a large set of unlabelled set of data with the logits for each class (the values before the softmax activation), then using then using this synthetically labelled data to train the smaller model [3]. Another proposed method suggests that if the labels for the second unlabelled set of data are known, then we could label the set of data using a weighted average of two objective functions: the cross-entropy with the soft targets and the cross-entropy with the correct labels [7]. The knowledge distillation technique has now also been applied to more complex classification set-ups, such as multi-class object detection. Multi-class object detection presents challenges such as classes which are not equally important and tasks involve elements of classification and bounding box regression. A weighted cross-entropy loss for classification can be implemented to account for the difference in importance of classes [4]. For neural network video analysis, NoScope, system for querying videos, was developed to reduce the computational cost. NoScope takes a video, an object, and a reference neural network, then produces a specialized neural network, which is less computationally ex-

pensive than the reference neural network [11]. Knowledge distillilation and small specialized neural networks present other ways to decrease the computational costs of CNNs, which could potentially be compounded with techniques involving resolution.

3. Methods

We construct image classification CNNs using various resolutions, models, and data sets. The resolution 224x244 serves as the baseline method, as it is the standard ResNet image size. We construct ResNet18, ResNet34, ResNet50, and ResNet101 models, where ResNetX is a ResNet with X layers. We use images of different resolutions each data set we consider (Birds 200, Kaggle cat dog, and Bird bike from adversarial examples).

We choose to construct standard ResNets, as these are popular image classification models, and thus it would be useful to develop a guideline for the image resolution that should be used to train and test ResNets, based on the difficulty of the task, although the results should also apply more generally to other models. ResNets are deep models which utilize residual blocks, which use short cut connections which skip one ore more layers by performing identity mappings. These identity mappings carry information from previous layers into later layers without adding any parameter or computation complexity. If for a stack of a few layers we have a desired underlying mapping of $H(x)$, instead of having the few stacked layers attempt to fit $H(x)$, we would have them fit $F(x) = H(x) - x$, making the original function $F(x) + x$, as illustrated in Figure 2. This helps to solve the vanishing gradient issue which deep networks often face. The specific architectures for ResNets consists of a convolution layer, an adaptive average max pooling layer (which is an average pooling layer which produces an output of size 56x56 for any input size), residual blocks with interspersed, periodically doubling the number of filters and down-sampling spatially with stride two (a total of 4 times), a pooling layer, a fully-connected layer, and a softmax layer, as shown in Table 1. Deeper networks implement a bottleneck building block, diagramed in fig:bottleneck, which consists of 1x1, 3x3, then 1x1 convolutions, where the 1x1 layers first reduce then restore the dimensions, so the the 3x3 convolution has a reduced dimension, improving the efficiency of the model [6].

In addition to 224x224, we consider the resolutions 65x65, 97x97, and 129x129, so that we can observe the accuracy of CNNs trained and tested using images of various degrees of lower resolution. We choose to explore various resolutions to determine the effect of resolution on accuracy, as decreasing the resolution has a quadratic effect on reducing the computation, since the both the width and the height of the image would be reduced. Thus, decreasing the resolution of training and testing images would be an effective method to decrease the computational costs of a CNN, with costs that decrease more rapidly as the resolution is scaled down more.

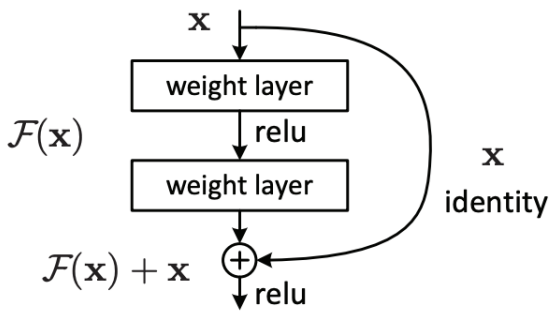

Figure 2. *This illustrates a residual block from a ResNet. Source: Adapted from [6] Figure 2.*

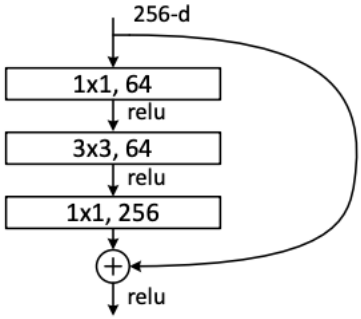

Figure 3. *This figure illustrates a bottleneck building block. Source: Adapted from [6] Figure 5.*

We construct and evaluate a CNN for every combination of resolution (65x65, 97x97, 129x129, 224x224), models (ResNet18, ResNet34, ResNet50, and ResNet101), and data set (Birds 200, Kaggle cat dog, and Bird bike from adversarial examples).

We used the torch [14] and numpy [13] libraries and used code from the PyTorch Fine-tuning Torchvision Models tutorial [1] to implement the ResNet 18. I then implemented the code specific for the ResNet 97, ResNet 129, and ResNet 224. I then created a structure which would train and test CNNs given a set of resolutions, models, and a data set (and its number of classes).

4. Dataset and Features

We use multiple data sets to reveal how task complexity affects the necessary resolution. The data sets which we will be using are Bird bike from adversarial examples (Bird Bike)[16], Animals-10 [2], and Birds 200 [18]. For each data set we down-sampled all of the images in the data set to

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				

Table 1. *This figure illustrates the architecture of ResNets. Source: Adapted from [6] Figure 5.*

obtain images with resolutions of 65x65, 97x97, 129x129, and 224x224. Furthermore, the images were normalized for each data set at each resolution by using standard ImageNet preprossessing, by changing the means of the RGB channels to be 0.485, 0.456, and 0.406, respectively, and the standard deviations of the RGB channels to be 0.229, 0.224, and 0.25, respectively. For data augmentation, we flipped the images horizontally. For each resolution and data set, we then constructed and tested a set of image classification CNN.

We use Bird Bike as a easy data set, as it contains photos of birds and bicycles (and thus also has two categories): it contains only two categories which look extremely different. For this data set, we have 26000 training examples, 1000 validation examples, and 1000 testing examples [16].

Animals-10 is a moderately difficult data set. It contains images of 10 categories of animals (dog, cat, horse, spider, butterfly, chicken, sheep, cow, squirrel, elephant). There are a moderate number of categories which differ in looks moderately. For this data set, we have 22608 training examples, 2629 validation examples, and 2825 testing examples [2].

Birds 200 is a difficult data set. It which contains photos of 200 different species of birds, and has large variation in the positioning of the bird and lighting of the photo, and several species of birds look similar to each other. For this data set, we have 3000 training examples, 1516 validation examples, and 1515 testing examples [18].

5. Results and Discussion

5.1. Experimental Setup

I used a SGD optimizer with a learning rate of 0.001 and momentum of 0.9, as when I initially tested with a small of epochs, these values produced the highest testing accuracy. I chose to train the models across 15 epochs with a batch size of 8, as with this number of epochs and batch size the training and validation loss and accuracy across epochs seemed to level out. Cross-validation was not used. To avoid over fitting, we used data augmentation, flipping the images horizontally.

For each data set, I observed images at the resolutions 65x65, 97x97, 129x129, and 224x224 to observe how the image changes with down sampling and to determine whether or not humans would still be able to use characteristics of the image to classify the image at the lower resolutions. For each CNN I trained, I observed the training and validation accuracy and loss across all epochs and determined the testing accuracy of the model, which is the primary metric.

5.2. Easy Task: Bird Bike data set

For models trained using the Bird Bike data set, an easy data set, we notice that all of the models performed nearly identically. In terms of loss and accuracy for both training and validation across all epochs, we see that models which used images of a higher resolution generally have a slightly lower loss and slightly higher accuracy at each epoch, however, the validation accuracies and losses were comparable, as illustrated for ResNet 18 in Figure 4 and Figure 6. In terms of testing accuracy we notice that all models achieved within 2% accuracy of each other, as demonstrated in Table 2. Thus, for easy tasks, images with lower resolution can be used without sacrificing much accuracy.

Looking at down-sampled images from the Bird Bike data set in Figure 5, we are able to see how birds and bikes have extremely different overall features, which re-

		Resolution			
Dataset	ResNet	65x65	97x97	129x129	224x224
bird-bike	18	98%	99%	99%	100%
	34	98%	99%	99%	100%
	50	98%	99%	99%	100%
	101	99%	99%	99%	100%
animals-10	18	81%	85%	89%	93%
	34	83%	87%	91%	94%
	50	83%	87%	92%	94%
	101	83%	90%	92%	95%
birds200	18	17%	28%	37%	54%
	34	20%	29%	39%	56%
	50	22%	31%	43%	59%
	101	19%	32%	41%	60%

Table 2. *This table contains the testing accuracy each CNN created using each data set, resolution, and model. We notice that increasing the resolution of an image generally increases the accuracy of the CNN, although the extent of the increase in accuracy differs greatly between data sets, with bird-bike having minimal increases and birds 200 having significant increases. The model used also has a slight impact, with .*

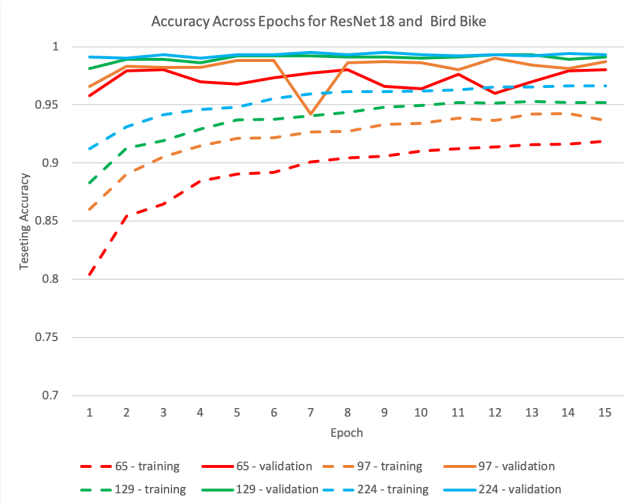

Figure 4. *This plot shows both the training and the validation accuracy of the ResNet 18 model trained using images from the Bird Bike data set at resolutions 65, 97, 129, and 224. The training accuracy was slightly higher for models trained and tested using images of higher resolutions. The validation accuracies were all comparable for all resolutions.*

main when an image is downsampled. Thus, we believe that using downsampled images did not affect the accuracy much because birds and bikes differ greatly in over all structure, which can observed when the image is downsampled.

We notice that for the Bird Bike models, we notice that the training accuracy is lower than the testing accuracy and validation accuracy, for all of the models, suggesting that the models are not over fitting the Bird Bike data.

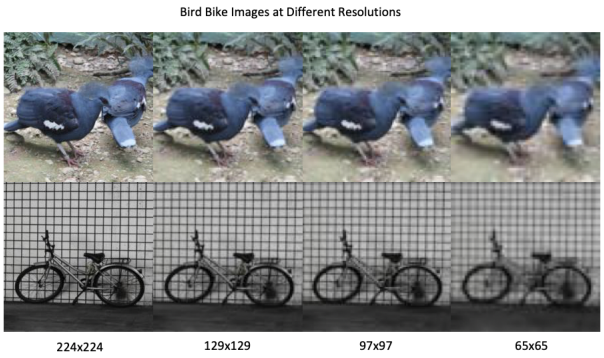

Figure 5. *We observe photos of a bird and a bike, the two categories in the Bird Bike data set at resolutions 65x65, 97x97, 129x129, and 224x224. We note that the two categories are still extremely distinguishable to the human eye in the lower resolution images.*

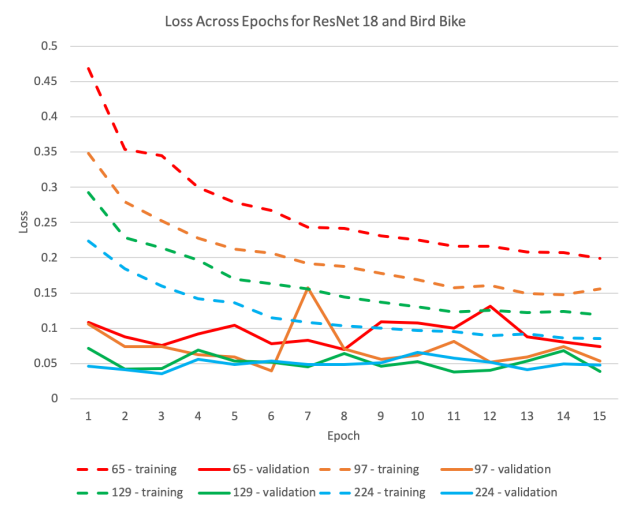

Figure 6. *This plot shows both the training the validation loss for the ResNet18 model trained using images from the Bird Bike data set at resolutions 65, 97, 129, and 224. The training loss was slightly lower for models trained and tested using images of higher resolutions. The validation losses were comparable for all resolutions.*

5.3. Moderately Difficult Task: Animals 10 data set

For models trained using the Animals 10 data set, a data set of moderate difficulty, we notice that models which used images with higher resolutions performed slightly better. In terms of loss and accuracy for both training and validation across all epochs, we see that models which used images of a higher resolution generally have a slightly lower loss and slightly higher accuracy at each epoch, as illustrated for ResNet 18 in Figure 7 and Figure 8. In terms of testing accuracy we notice that for ResNet 18, a shallower model,

there are uniform, moderate increases in accuracy as the resolution is increased, however for ResNet 101, a deeper model, the testing accuracy for models which were trained and tested on images with moderate resolutions (97x97 and 129x129) became slightly more comparable to the model trained and tested on images with a standard resolution (224x224)Table 2. Thus, for moderate tasks, images with moderate resolutions could be used without sacrificing significant amounts accuracy.

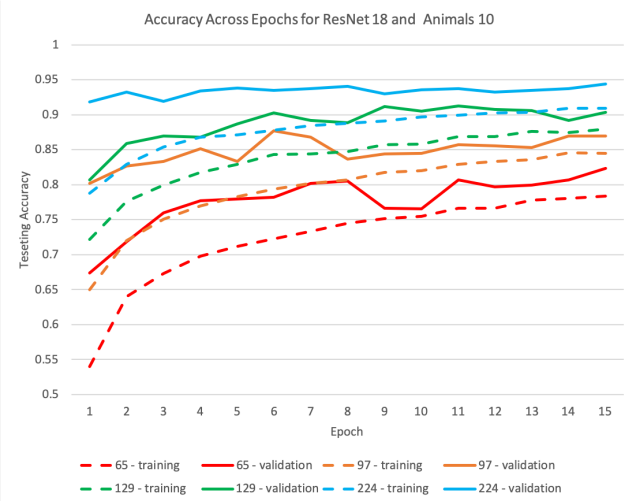

Figure 7. *This plot shows both the training and the validation accuracy of the ResNet 18 model trained using images from the Animals 10 data set at resolutions 65, 97, 129, and 224. The training and validation accuracy were slightly higher for models trained and tested using images of higher resolutions.*

Looking at down-sampled images from the Animals 10 data set in Figure 9, we are able to see how the features which distinguish different animals are slightly diminished. Thus, we believe that for Animals 10, using downsampled images slightly affects the performance of the CNN because because although the features of animals are generally more similar to each other than those between a bike and a bird, the animals all still have slightly different general structures and they differ greatly in structure, and these differences can still be observed in downsampled images.

We notice that for the Animals 10 models, the training accuracy is lower than the testing accuracy and validation accuracy, for all of the models, suggesting that the models are not over fitting the data.

5.4. Difficult Task: Birds 200 data set

For models trained using the Birds 200 data set, a difficult data set, models which were trained and tested on images with low resolution performed significantly worse than models which were trained and tested on images with high resolution. In terms of loss and accuracy for both training

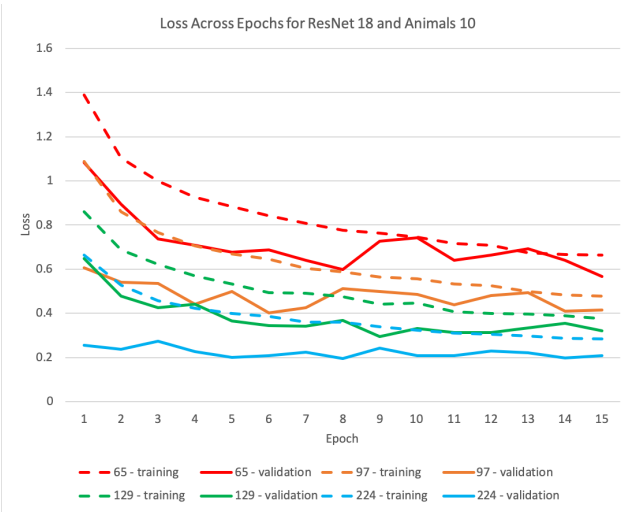

Figure 8. *This plot shows both the training the validation loss for the ResNet18 model trained using images from the Animals 10 data set at resolutions 65, 97, 129, and 224. The training and validation losses were slightly lower for models trained and tested using images of higher resolutions.*

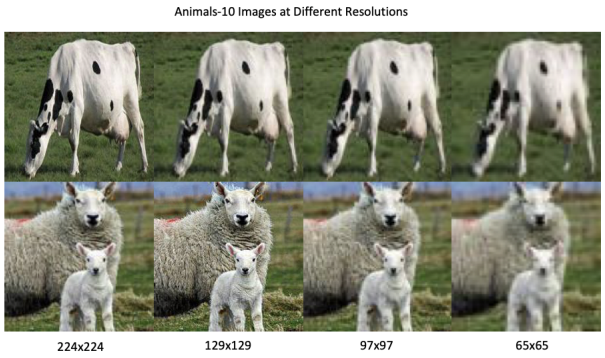

Figure 9. *We observe photos of a cow and a sheep, two of the ten categories in the Animals 10 data set at resolutions 65x65, 97x97, 129x129, and 224x224. We note that the two categories are still distinguishable to the human eye in the lower resolution images, although the difference is less clear.*

and validation across all epochs, we see that models which used images of a higher resolution have a lower loss and a higher accuracy at each epoch, as shown for in Figure 10 and Figure 11. In terms of testing accuracy, for the models trained using Birds 200, we notice that models which used images of higher resolutions achieve much greater testing accuracy Table 2. Thus, for difficult tasks, images of standard (224x224) or high resolution should be used.

Looking at down-sampled images from the Birds 200 data set in Figure 12, we are able to see how the small features which distinguish different species of birds are greatly diminished, which may explain why CNNs trained

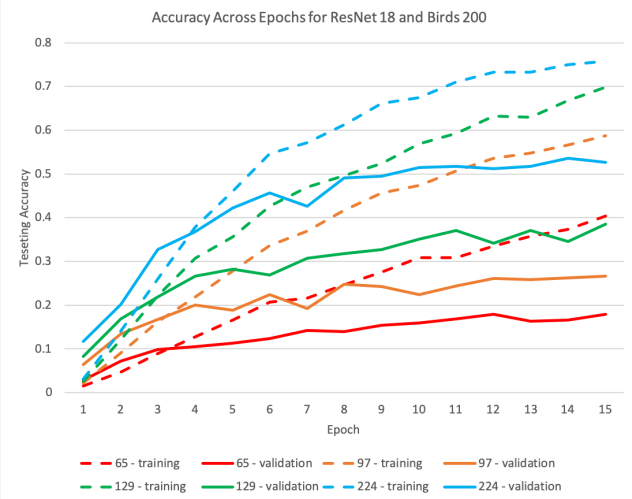

Figure 10. *This plot shows both the training and the validation accuracy of the ResNet 18 model trained using images from the Birds 200 data set at resolutions 65, 97, 129, and 224. The training and validation accuracies were higher for models trained and tested using images of higher resolutions.*

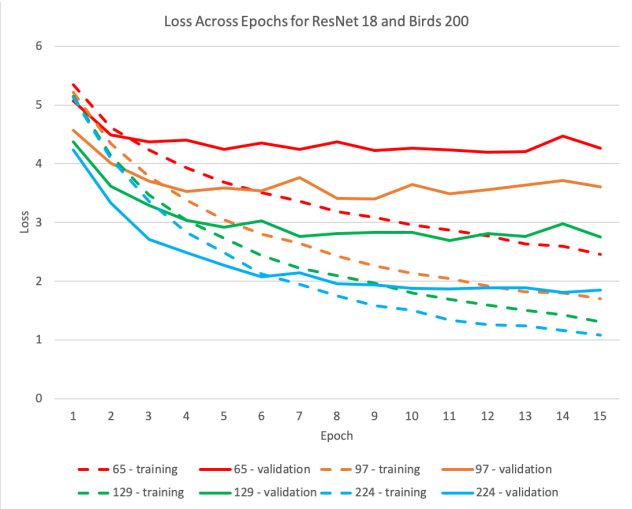

Figure 11. *This plot shows both the training the validation loss for the ResNet18 model trained using images from the Birds 200 data set at resolutions 65, 97, 129, and 224. The training and validation losses were lower for models trained and tested using images of higher resolutions.*

and tested using images with lower resolutions should perform worse than CNNs trained and tested using images with higher resolutions.

The Birds 200 data set may have been slightly over fit, as the training accuracy was higher than both the validation and and testing accuracy. However, we also notice that models which used images with higher resolutions were less over fit than models which used images with lower resolutions.

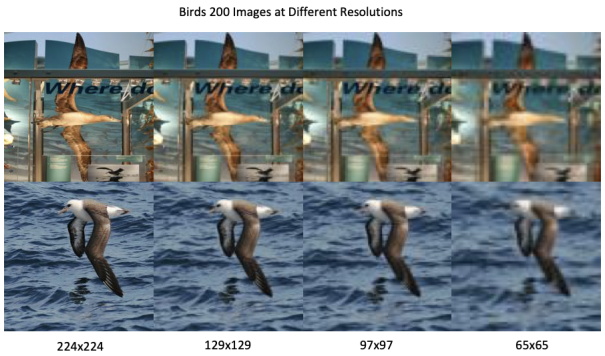

Figure 12. *We observe photos of a Layasan Albatross (top) and Black-footed Albatross (below), two of the 200 categories in the Birds 200 data set at resolutions 65x65, 97x97, 129x129, and 224x224. The two species look similar, even with the high resolution images, and the two categories look even more similar in the low resolution images.*

6. Conclusion and Future Work

Image classification CNNs have become deeper and more costly to evaluate. One way to reduce the computational cost of a CNN is to reduce the resolution of the training and test image set: scaling down images. Thus, we worked to establish a guideline for the resolution needed to complete various tasks by training and testing CNNs using images of various resolutions for tasks of various difficulties. We find that improving the resolution of the training and testing data set will only improve the accuracy of the CNN, however, the amount of accuracy gained by increasing the image resolution depends on the difficulty of the task. For simple tasks, such as distinguishing between birds and bikes, a model trained and tested with images with resolution 65x65 will perform nearly as well as a model trained and testing with images with resolution 224x224. However, as the difficulty of the tasks increases, the difference in testing accuracy for models which use different resolutions will increase. Thus, for easier tasks, we could use images of low resolution (64x64) to lower the computational cost without sacrificing accuracy, but we should avoid doing so for more difficult tasks. For moderate tasks, depending on the accuracy wanted, images with moderate resolutions (such as 97x97 or 129x129) could be used. However, for difficult tasks, images with standard resolution (224x224) should be used. This may have occurred because for easier classification problems, the there is a larger structural difference between the different classes, which are still present when the resolution is reduced, in comparison to more difficult classification problems, where there may only be some mid-level differences, which may be greatly reduced when the

resolution of the image is reduced. In the future, we hope to develop a method which would be able to determine the resolution for training and testing photos which is needed to complete different classification tasks without significantly reducing the accuracy of the classifier which does not involve training multiple models using images of different resolutions.

7. Contributions and Acknowledgements

D.K proposed the project idea, advised on the project, and provided GPUs. K.W. implemented CNNs and wrote the paper.

References

[1] Finetuning torchvision models.

[2] C. Alessio. Animals-10, Oct 2018.

[3] J. Ba and R. Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2654–2662. Curran Associates, Inc., 2014.

[4] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 742–751. Curran Associates, Inc., 2017.

[5] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, page arXiv:1503.02531, Mar 2015.

[8] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[9] Y. Huang, Y. Cheng, D. Chen, H. Lee, J. Ngiam, Q. V. Le, and Z. Chen. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. *arXiv e-prints*, page arXiv:1811.06965, Nov 2018.

[10] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica. Chameleon: scalable adaptation of video analytics. pages 253–266, 08 2018.

[11] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia. Optimizing deep cnn-based queries over video streams at scale. *CoRR*, abs/1703.02529, 2017.

[12] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[13] T. Oliphant. *Guide to NumPy*. 01 2006.

[14] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[15] X. Peng, J. Hoffman, S. X. Yu, and K. Saenko. Fine-to-coarse knowledge transfer for low-res image classification. *CoRR*, abs/1605.06695, 2016.

[16] Y. Song, R. Shu, N. Kushman, and S. Ermon. Generative adversarial examples. *CoRR*, abs/1805.07894, 2018.

[17] M. Tan and Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv e-prints*, page arXiv:1905.11946, May 2019.

[18] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[19] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016.