Project Proposal

Yiming Bian

April 2022

Major Professor

Arun Somani(arun@iastate.edu

Committee Members

Henry Duwe (duwe@iastate.edu)
Aditya Ramamoorthy (adityar@iastate.edu)
Alexander Stoytchev (alexs@iastate.edu)
Cindy Yu (cindyyu@iastate.edu)

1 Introduction

Object detection is one of the most popular topics in machine learning area. It is a cross field of computer vision and image processing that detects instances of a certain class in digital images and videos. In 2009, a large database, which contains over 14 million hand-annotated images, designed for visual object recognition research called ImageNet was presented at the Conference on Computer Vision and Pattern Recognition(CVPR) in Florida by Fei-Fei Li et al. In 2010, the ImageNet project began an annual contest called the ImageNet Large Scale Visual Recognition Challenge(ILSVRC). This challenge uses a subset of ImageNet that has $1,000$ non-overlapping classes of objects [1]. In Fig. 1, it shows the top-5 error rate of the ILSVRC winner network each year from 2012 to 2016. As the average recognition ability of human has an error rate of 5%, the winner in 2015, ResNet, beat human level for the first time with an error rate of 3.6%.

ResNet is one of the most famous modern convolutional neural networks(CNN). There is a suite of ResNet networks such as ResNet-18, ResNet-34, ResNet-50 etc. The number stands for the number of layers, or depth, of the network. When passing an image to the network, it produces a result as shown in Fig. 2.

Object recognition has achieved significant breakthroughs over the past decade. Compared to good-quality images, however, there are more low-quality images(LQI) in the real world. Many reasons for the flaw are data erosion,

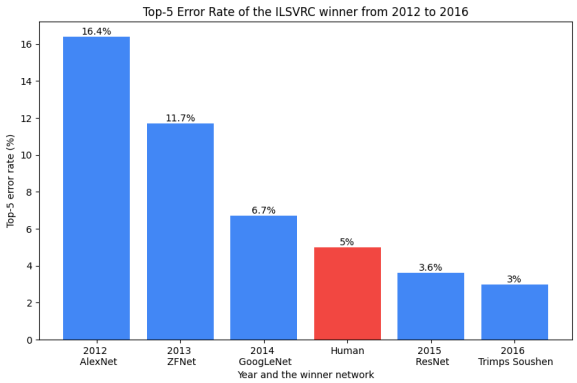

Figure 1: Top-5 Error Rate of the ILSVRC winner from 2012 to 2016: top-5 error refers to that the correct answer is not among the top five predictions by the network

Figure 2: Input image and output predictions: The network can be configured to output top n predictions, each with a label and a probability. In this example, it predicts the object in the image as a Samoyed with a probability of over 88%

storage hardware malfunction, physical damage to a photo and cheap cameras. To recognize such flawed images is a great challenge to existing models because they were all trained with fine quality data. Hence, the project I would like to do is to explore strategies that improve the prediction accuracy on different kinds of LQI.

The rest of the report is organized as follows. Section 2 lists some previous research and results. Section 3 explains two experiments and the preliminary results prove the poor performance with current models on LOI and the potential improvement of accuracy on LOI using transfer learning with a training set of LOI. Section 4 states the general plan for this project and the expectation before preliminary exam in three months.

2 Related Work

Searching for keywords as low quality image processing, recognition, classification, there is not any general researches. Then I narrows it to specific areas and got the following results.

In [2](2016), authors show the existing networks(VGG16, GoogleNet, VGG-CNN-S and Caffe Reference) are susceptible to quality distortions, particularly to blur and noise. In [3](2018), their results indicate that both hand-crafted and deep-learning based face detectors are not robust enough for low-quality images. In [4](2018), authors concluded several techniques to improve the performance of low-quality face recognition(LQFR) such as super-resolution processing, deblurring and learning a relationship between different resolution domains. In [5](2019), authors proposed a framework for recognizing objects in very low resolution images through the collaborative learning of two deep neural networks including image enhancement network and object recognition network.

All previous works either focus on a specific object recognition(face) or a specific quality deficiency(low-resolution). What distinguishes my project is that it is an all-around research work that focuses on general objects and many kinds of quality deficiencies.

3 Experiments and Preliminary Results

The CNN in the following two experiment is ResNet-18 and PyTorch has a built-in implementation with pre-trained weights.

In the first experiment, I used three sample images: Samoyed, cray fish and bike-for-two. Then I added salt-and-pepper noise to the original images at four different levels to generate low-quality images as test data. The test data were passed to the pre-trained ResNet-18 and the predictions were recorded.

In the second experiment, I randomly picked 5 objects from ImageNet: goldfinch(1300 samples), hamster(1300 samples), frying pan(1222 samples), pitcher(1300 samples) and upright(1300 samples). Then generate low-quality images with salt-and-pepper noise at four levels, thus, each object now has a fourfold sample

size. There are four models as well: pre-trained model(no change), re-trained model on a small original data set, re-trained model on a small noise data set and re-trained model on a small mix data set. The test accuracy of four models on different test sets were recorded.

3.1 Pretrained Model on Low Quality Images

In Fig. 3, it is an example of how the noise images look like and how ResNet-18 performs on each input image. In this case, the prediction for Samoyed drops from 88.64% to 46.41% while there is an probability increase with level 1 noise. Another good news is that Samoyed is the top-1 prediction in all test cases.

In Table 1, however, unstable performance is observed. In the case of crayfish, it is not the top-1 prediction in neither the original image nor the image with level 1 noise. In level 2 noise image, it is a top-5 error, thus, crayfish is not present among the top 5 predictions. When the noise level increases further, the prediction of crayfish has a probability around 8.9% to 9%.

In the case of bike-for-two, the increase of noise level does not affect much to the prediction. One explanation to all these strange behavior is that the test image has various difficulties to recognize. For example, Samoyed and Arctic fox are alike; crayfish, cockroach and scorpion look like each other; bike-for-two has a very unique structure and even compared to the second prediction, mountain bike($< 1\%$).

This simple experiment justify that the lack of robustness of existing CNN models when dealing with low-quality images.

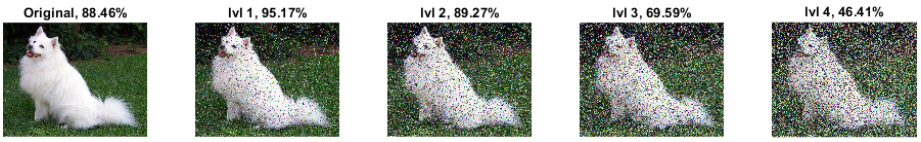

Figure 3: Correct prediction probabilities on original image and flawed images at 4 noise levels

Table 1: The probability of correct predictions of ResNet-18 on different input images: N stands for the correct prediction is not top-1. ER stands for the occurrence of top-5 error.

	Original	**lvl. 1**	**lvl. 2**	**lvl. 3**	**lvl. 4**
Samoyed	88.46%	95.17%	89.27%	69.59%	46.41%
Crayfish	10.98%(N)	2.95%(N)	ER	8.99%	8.91%(N)
Bike-for-two	99.70%	99.63%	98.91%	93.24%	89.97%

3.2 Models Comparisons

In this experiment, I construct nine data sets for multiple purposes: training, validation and testing. Data set I, II and III are three non-overlapping small subsets of IV and all contain only original images, thus, images without any noise. Data set IV is constructed using samples of five objects in ImageNet. The rest data sets from V to IX only contain noise data. Data sets V, VI and VII are non-overlapping small subsets of VIII. And data set VIII and IX are non-overlapping as well. The reason to have these two data sets rather than combining them together is that VIII is prepared for future training and IX is for future validating. As you may notice, $\#Samples(VIII + IX) = 4 \times \#Samples(IV)$. It is because IV is the whole set of original data and the combination of VIII and IX is the whole set of noise data. We generate 4 levels of noise based on an original image. As of now, all these data sets are used for testing purpose.

Data set I and II are used to train and validate the pre-trained ResNet-18 model into RtO model. Data set V and VI contribute to RtN model and the combination of I and V, II and VI generate the RtM model. In Table 3, it shows the testing accuracy of each model on every testing data set. As you may notice the sizes of these training sets are very small, I did this on purpose as I would like to prove that even with a little training, there is an accuracy improvement.

Without any re-training, the pre-trained model has a descent performance on original testing sets($> 77\%$) but a very poor performance on noise testing sets($< 42\%$). After re-training using original data, it has a even better accuracy on original testing sets($> 95\%$) and the accuracy on noise sets is well-improved to about 75%. However, a clear drop is observed. The accuracy improvement over pre-trained model on both the original and noise data compared is because of the last fully-connected layer difference. For pre-trained model, is has $1,000$ potential output choices while RtO model only has 5.

Re-training using only noise data, RtN model has a stable performance of $82.09\% - 88.86\%$. Re-trained on a mix of original and noise data, RtM model has the best performance on every testing sets. However, it is not solid to conclude RtM is the best among all models because its training sets is twice the size as that of RtO and RtN model. Nevertheless, clear improvement of recognizing image with noise can be observed after re-training on noise data.

When testing on noise data sets VII, VIII and IX, detailed error types are collected as well. In Fig. 4, 5 and 6, the error types and corresponding percentages of four models on each data set are shown. There are abnormal behaviors observed such as for RtM model, the error rate of level 4 noise images is lower than that of level 3 noise images in all testing sets. I cannot explain this behavior with 100% confidence for now. But maybe it is because of hyperparameters tuning issues since the size of training set and validation set is very small(1000 and 500), hyperparameters should be adjusted accordingly.

Table 2: Test sets configurations: Dis. stands for distribution. Take 24×5 under **Objects Dis.** as an example, it stands for 5 objects and each has 24 samples. In **Noise Dis.** column, the tuple with five elements indicates the percentage of original image and image with each noise level in the data set. For example, $(0, 25, 25, 25, 25)$ stands for the percentages of original image and images of each noise level from 1 to 4 are 0% and 25%.

Data set	**Purpose**	**# Samples**	**Objects Dis.**	**Noise Dis.**
I	train	500	100×5	$(100, 0, 0, 0, 0)$
II	validate	250	50×5	$(100, 0, 0, 0, 0)$
III	test	120	24×5	$(100, 0, 0, 0, 0)$
IV	test	6,422	$1300 \times 4 + 1222$	$(100, 0, 0, 0, 0)$
V	train	500	100×5	$(0, 25, 25, 25, 25)$
VI	validate	250	50×5	$(0, 25, 25, 25, 25)$
VII	test	298	$62 \times 4 + 50$	$(0, 24.16, 24.16, 25.84, 25.84)$
VIII	test	17,122	$3,466 \times 4 + 3,258$	$(0, 25.01, 25.01, 24.99, 24.99)$
IX	test	8,566	$1,734 \times 4 + 1,630$	$(0, 24.97, 24.97, 25.03, 25.03)$

Table 3: Test accuracy of four models: **Pre_acc** stands for the accuracy of **Pre**-trained ResNet-18 model. **RtO_acc** stands for the accuracy of the model **R**e-**t**rained on **O**riginal data set. **RtN_acc** stands for the accuracy of the model **R**e-**t**rained on **N**oise data set. **RtM_acc** stands for the accuracy of the model **R**e-**t**rained on a **M**ixed data set.

Data set	**Pre_acc**	**RtO_acc**	**RtN_acc**	**RtM_acc**
III	81.67%	97.50%	85.00%	97.50%
IV	77.86%	95.27%	82.09%	95.31%
VII	41.95%	74.83%	88.59%	90.60%
VIII	35.69%	76.19%	87.94%	90.43%
IX	36.00%	76.83%	88.86%	90.89%

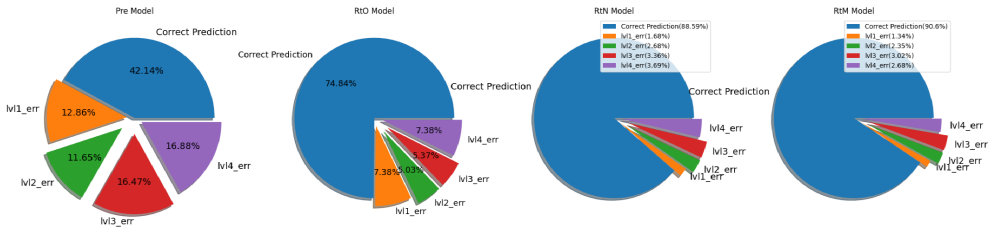

Figure 4: The distributions of all models' predictions on data set VII

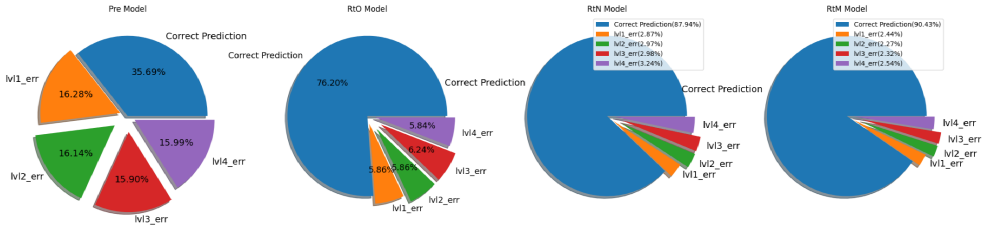

Figure 5: The distributions of all models' predictions on data set VIII

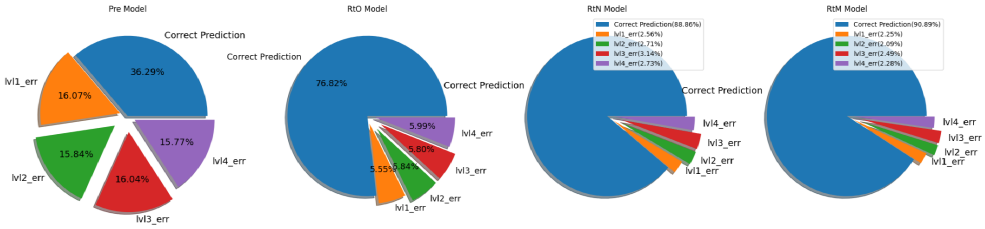

Figure 6: The distributions of all models' predictions on data set IX

4 Project Plans

The target of this project is to improve the prediction accuracy on different kinds of low quality images.

First, a comprehensive construction of LQI data set needs to be determined. It can either be based on existing data sets such as ImageNet and CIFAR-10, photos captured by me or a mixture. To generate flawed images, there are multiple ways such as adding Gaussian noise, salt-and-pepper noise etc. The flawed images should be very hard for current pre-trained models to recognize while maintaining only a moderate difficulty level of recognition for human being. I may design a image recognition test and collect results from some volunteers to compare with those from existing models.

Then, to deal with different kinds of LQI, specific strategies need to be explored. For example, when dealing with scattered noises, one could use convolution operation or other de-noise methods. When dealing with lacking evidence, one could use generative adversarial network(GAN) to generate more potential features in the given image. The aforementioned strategies need to be supported by experiments and mathematical deductions. They are just quick thoughts for the current stage.

There is a 3-month gap before the preliminary exam. In this time of period, I plan to write a progress report and share with you every two weeks,

thus, the next report will be delivered on April 22nd. All the files relating to this project are uploaded to a GitHub repository at https://github.com/YimingBian/Project-Model_on_LQI-Low-quality-image-.git. For everyone's convenience, I set this repository to public so that it can be accessed without requirement of a registration.

To improve the prediction accuracy on different kinds of LQI, here are several questions that need to be answered 1) What is the definition of low quality image? 2) What are the training and testing data sets? 3) What strategies can be applied to achieve accuracy improvement?

I will take the following steps to proceed this project. First, I will prepare several LOI data sets based on existing ImageNet, CIFAR-10 and other public data sets. Flawed images can be generated by adding random noise, doing random cropping etc. The goal of constructing the LOI data set is to generate images that can hardly be recognized by existing pre-trained networks while maintaining a moderate difficulty for an average human to recognize. Then I will train and test the data sets on different models such as ResNet, VGGNet etc using transfer learning. After that, I will compare the prediction accuracy of tested models to determine if transfer learning on LQI helps the improvement. Another potential method is to adding an additional layer of pre-processing the input flawed image to denoise and convert it to a higher quality one. There are specific techniques for each flaw and I need to investigate the previous works in different areas before proposing a novel one.

What to expect before the preliminary exam at least includes 1) One or several proper LOI data sets are formed and the proof is provided. 2) Whether transfer learning is an improvement method or not and the proof.

References

[1] I. L. S. V. R. Challenge, Olga russakovsky, jia deng, hao su, jonathan krause, sanjeev satheesh, sean ma, zhiheng huang, andrej karpathy, aditya khosla, michael bernstein, alexander c. berg, li fei-fei. 2014, Computing Research Repository, Vol. abs/1409.0575.

[2] S. Dodge, L. Karam, Understanding how image quality affects deep neural networks, in: 2016 eighth international conference on quality of multimedia experience (QoMEX), IEEE, 2016, pp. 1–6.

[3] Y. Zhou, D. Liu, T. Huang, Survey of face detection on low-quality images, in: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018), IEEE, 2018, pp. 769–773.

[4] P. Li, L. Prieto, D. Mery, P. Flynn, Face recognition in low quality images: a survey, arXiv preprint arXiv:1805.11519 (2018).

[5] J. Seo, H. Park, Object recognition in very low resolution images using deep collaborative learning, IEEE Access 7 (2019) 134071–134082.