Exploration Between Image resolutions and Classification

Abstract

- Studies shown the importances of image resolution in convolutional operation [1]. It further implies the need for higher resolutions images. This paper would explore the need of super resolution with Deep Learning methods which would further improve classification performance.

Project Overview

- The reason we chose MSRN is that it has several advantages over traditional approaches. First of all, it is easy to reproduce the experimental results. Unlike most SR models which are sensitive to the subtle network architectural changes and highly rely on the network configuration, MSRN blabla. Secondly, it avoids inadequate of features utilization. It enhances the performance not by blindly increasing the depth of the network. Instead. Last but not lease, it has good scalability. Therefore, this method is easy for us to observe the result and save time of computation.
- Overall, there are 3 training. In the first training, the original images are feed into denseNet directly. And the result will be as the baseline. In the second training, the image is down-scaled and then up-scaled with naive algorithm which results low resolution image. Then feed it into denseNet to do classification. In the third training, we do naïve down-scale on the image too. we feed it into MSRN. And after MSRN, we feed it into denseNet. By comparing second and third training group, we can know how much resolution improved by MSRN. Also, we compared the classification result of first and third training group. So we can compared the classification result to see the differences

Datasets

- This paper uses a annoated image dataset known as the CIFAR-10 dataset [2]. It contain ten classes of images. As **figure 0.1** as shown, the dataset has classes of airplane, automobile , bird , cat , deer , dog , frog , horse, ship and truck. The CIFAR-10 contains total of 50,000 training images and 10,000 test images. Furthermore, the trainingsets contains exactly 5000 images for each class[2].
- – ***figure 0.1 cifar 10 datasets*** *it smaple of the cifar 10 dataset*
- The motiviation for choosing this dataset is that each object is clearly distinguishable by other class. It evades the concern of illumination , deformation and occlusion of the image. For example, the important feature points that identifies bird is clearly distinguishable then the key feature points of a truck. The learning curve for training such classifier is less computationally expensive than other datasets.
- Therefore, this experiment conducts on training a identical densenet with two different of inputs, a naive resized images and output images of MSRN. The evaluation metris of the classifiers is the test error and training loss variances. Training loss variance identifies how fast the classifier learn from the inputs relative to labels. Statistically, loss variances implies how distinguishable are the features between classes. In this context, feeding naive and super resolutions images can help this paper to conclude the impact of resolution on classification in terms of test accuracy and training cost.

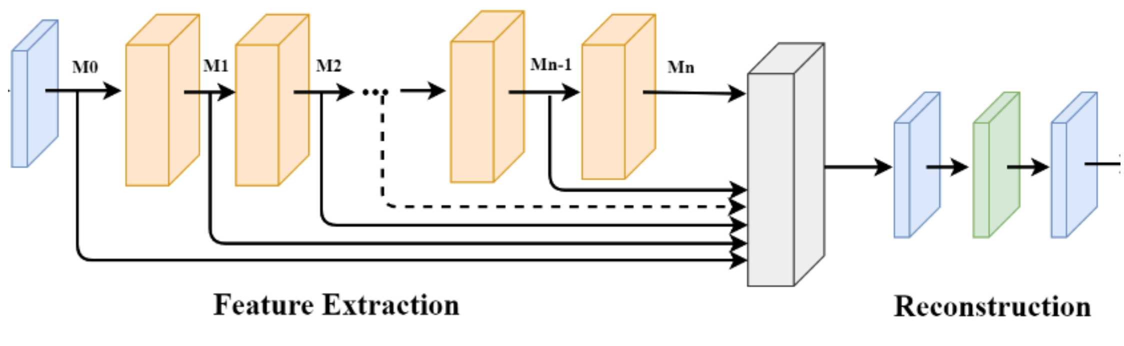
Approache Analysis

- All of the implmentation and running instruction are in this repository

1. MSRN Resized Image preprocessing

- The implementation of this approach can be lookedup here

- MSRN stands for **Multi-Scale Residual Network**, it is a supervisied learning model that learns to upscale the image from low resolution to a higher resolution in arbitary scale. This network serves a purpose of scaling an image to a given ratio by perserves the key features such as line and shape from a distorted image.
- MSRN is bult by **Multi-Scale-Residual Block** (msrb) which consists of multiple residual network blocks to perform lower resolution feature extraction on **y** channel of the image which spanned in color space **yCbCr**[3]. Then concatenated all of the filters and feeded to a **Sub-Pixel Convolutional layer** to reconstruct a higher resolution image. Sub-pixeling Convolutional Neural Network is a network structure learn to upscale the lower resolution image to a higher resolution output by estimating ratio resolution arangement[4]. This enable MSRN to generate super resolution image based on ground truth.



–
    * ***figure 1.1**, this show the architecture of MSRN it consists of n block of residual layer and reconstructed by a sub-pixel convolutional layer in reconstruction layer*

- The **blue block** in figure1.1 is a 64 channels of Convolutional Layers and **orange block** is MSRB block. All of $M_1..M_n$ will be concatenated at gray block to linearized by feeding into another **one kernel Convolutional** layer. This allows the network to learn the distinguishing pixel region from all of the previous channels. Finally, the output would feed into the **Sub-Pixel Convolutional Layer** to recreate the y image format.
- – ***figures 1.2**, more classes of image super resolution result presenting*
- The following function L is the cost function of training an MSRN. I_i^{LR} is y channel of **naive downscaled** images in $yCrCb$ colorspace. I_i^{HR} is y channel of the **origin image** in $yCrCb$ and F_θ is the forward pass of the MSRN. This allow the network to evaluate how far are the features between pixle spaces which allow back propagation to derivate gradient to optimizes the network.
    - $L(F_\theta(I^{LR}, I^{HR})) = \sum_{i=0}^{n} ||F_\theta(I_i^{LR} - I_i^{HR})||_1$[3]
- With the back propagation, the network is able to adjust the weights to generate an image output that is closer to I^{HR}. Super resolution enable application of resizes and scaling of an image while perserve the key details of an object. This allow the exploration of relationship between resolution and classification.

2. Naive Resized Image preprocessing

- The implementation of this approach is here

- Naive often refer brute forces to solve the given problem. However, in this given approaches refer as `cv2.resize` API in openCV. The goal is applying different geometric transformation to images like scaling by reprojecting image points on a different plane and perserve its key features like ratios of distances between points.
- The algorithm started with input image X has shape of (h_x, w_x) and output y image is (h_y, w_y). First compute the scaling factor by computing $(\frac{h_y}{h_x}, \frac{w_y}{w_x})$ and mulitplied the identiy matrix I_n obtain the scaling matrix $M = \begin{bmatrix} \frac{h_y}{h_x} & 0 \\ 0 & \frac{w_y}{w_x} \end{bmatrix}$ then computed $f : X \to \ (Mx)I_{|X|} \ \to y$ to obtain image $Y_{downscale}$ [5].
- In this approach, training and testing images will processed by this algorithm twice. First downscaling the image by half and upscale once to origin sizes which perserves the same input size during classification. This allow this project to explorate the result of classification with different scaling preprocessing.

-

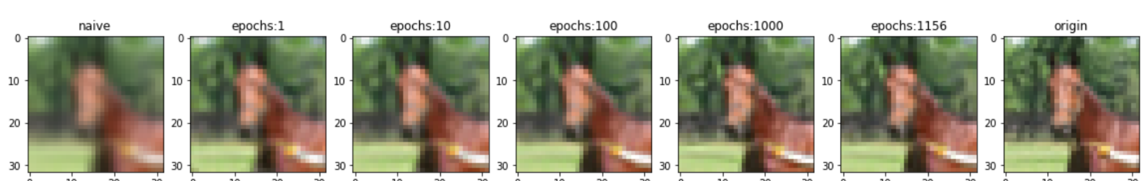
  figures 2.1: *present the resize prprocessing of this approach*

3. Result & Analysis

- The demo implementation is in here

3.1 MSRN

- In this experiment, MSRN is trained with 49,000 training images with 1,000 of test images. With `L1loss` (2) function, MSRN is able generated feature perserving image which further enhanced classification performances.



-
  - ***figures 3.1***, *this show the learning steapness of **MSRN**. It show a horse image that generated by msrn in different epoches.*
- As the figure 3.2 as shown, the resolution of the output is slowly increases as the epochs numbers increases. The image become more smoother as the network is learning. It startes improve the faces region of the horse froma he input. This shows the performance metric comparison learning of `l1 loss`.

This shown the out performance of template matching of vision in deep learning, it only require one epochs to generate an image that can out perform traditional approch. However, as network is trained with more epochs, it becomes difficult for network to learn.

- As **figure 3.2** has shown, `l1 loss` become so small that it became challenging for the network to learn. Another challenge is the network can't perform preprocessing to train. For example, perform back propagation on value between -1 to 1 is faster than 0 to 255. It takes 1000 epochs to deduces the loss to close to 5. Furthermore, **figure 3.1** shows the output at epoch 100 and 1000 are not visually different.
- – ***figures 3.2****, this show the learning curve of* ***MSRN****. it plot against the epochs vs l1loss.*

3.2 Resolution Comparison

- **Figure 3.3** shows the preprocessing of three test image in terms of traditional algorithm approach, super resolution and origin image. In column `Naive`, the algorithm causes the input image to be loss some important resolutions, including the legs and head of the deers. In comparison, the super resolution image appears more detailed and smoother toward the origin image.
- – ***figures 3.3****, comparison of resolutions between traditional approaches and super resolutions*

3.3 Classification

-

References

1. Etten Adam Van. Quantifying the Effects of Resolution on Image Classification Accuracy. Medium.com. retrieved from https://goo.gl/v2xa2T
2. Alex krizhesky, Vinod N, GEoffrey H. The CIFAR-10 dataset, Univeristy of Toronto. retreived from https://www.cs.toronto.edu/~kriz/cifar.html
3. MSRN
4. pixle shuffling
5. Geometric Operations: Affine Transform, R. Fisher, S. Perkins, A. Walker and E. Wolfart.