Understanding How Images Affect The Classification Models

*Based on Image classification models

1st Zhe Xuanyuan
Data Science
United International College
Zhuhai, China
zhexuanyuan@uic.edu.hk

1st Rui Meng
Data Science
United International College
Zhuhai, China
ruimeng@uic.edu.hk

2nd Changyuan Liang
Data Science
United International College
Zhuhai, China
m730026058@mail.uic.edu.hk

***Abstract*—Image quality is an important practical challenge and is often overlooked in the design of machine vision systems. Generally, machine vision systems are trained and tested on high-quality image data sets, but in practical applications, it cannot be assumed that the input image is of high quality. The more computing power is required for high-quality images, which is often not available. In this article, we provide evaluations of various picture classification models, which are used to classify images of different quality images. In terms of picture clarity, we consider high quality and low quality. In the picture color channel, we consider RGB and GRAY.**

I. Introduction

Traditionally, the focus of visual quality assessment is on the perception of quality from the perspective of human subjects. However, with the increasing popularity of computer vision, it is also important to characterize the impact of image quality on computer vision systems. These two concepts of image quality may not be directly comparable, because computers may treat humans as the same image [3], or in some cases, computers can recognize images that are indistinguishable from human observers [4]. Therefore, it is important to consider how image quality alone affects computer vision applications.

With the improvement of computer computing power, it is no longer a problem to perform a large number of matrix operations, and it is no longer a problem to build a neural network based on CNN. In the process of studying how to classify pictures, I found two problems, one is the picture size and the picture color. This article will mainly discuss the impact of pictures on traditional machine learning models and neural network models.

II. Related Works

A. Motivation and Background

Part of my final year project is related to image classification, we need to find cars on different images, so I want to have a study first. On the other hand, now that the computing power of the computer is very powerful, it is feasible to train a small neural network. For the iamge size and image quality part, regardless of the published paper and the speech of other students, most of them did not elaborate on this issue. But they seem to be colluding well, choosing a certain size consistently, and giving no reason. Another big problem is that, most of them also converted RGB images into grayscale images, and did not explain the reason.

B. Related Research

In computer vision, recent techniques based on deep neural networks (DNN) have begun to achieve state-of-the-art results in many problem domains [4]. Of particular interest to DNN models is image classification performance on large scale datasets with millions of images and thousands of categories. These problem domains were previously thought to be extremely difficult, but DNNs have achieved very impressive results. For example, in the ILSVRC 2010 challenge, the AlexNet DNN [5] achieved the best result with nearly 9% better classification accuracy than the second-best result based on hand-crafted features.

For many applications in computer vision it is assumed that the input images are of relatively high quality. However, in certain application domains such as surveillance, image quality is an important consideration. Additionally, with the advent of many computer vision applications on cellular phones, the requirement of high-quality images may need to be relaxed. In surveillance applications, face recognition in low-quality images is an important capability. There are many works that attempt to recognize low-resolution faces [6]. Besides low-resolution, other image quality distortions may affect performance. Karam and Zhu [7] present a face recognition dataset that considers five different types of quality distortions. They however do not evaluate the performance of any models on this new dataset. Tao et al. [8] present an approach based on sparse representations that achieve good performance on this dataset. For hand-written digit recognition, Basu et al. [9] present the n-MNIST database, which is a modification of the benchmark MNIST dataset. n-MNIST adds Gaussian noise, motion blur, and reduced contrast to the original images. Additionally, the authors in [9] propose a modification of deep belief networks to achieve greater accuracy on this dataset.

III. Method

We will describe how we do the feature extraction in both traditional machine learning models and deep learning models. For machine learning model, we use Histograms of Oriented Gradients (HOG) [2] to do the feature extraction.

A. Histograms of Oriented Gradients (HOG)

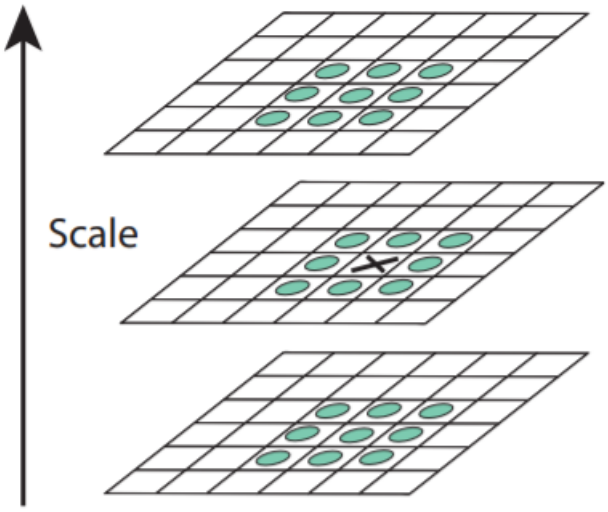

Fig. 1. Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

1) Local extrema detection: In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see Figure 1). It is selected only if it is larger than all of these neighbors or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few check.

Therefore, we must settle for a solution that trades off efficiency with completeness. In fact, as might be expected and is confirmed by our experiments, extrema that are close together are quite unstable to small perturbations of the image. We can determine the best choices experimentally by studying a range of sampling frequencies and using those that provide the most reliable results under a realistic simulation of the matching task.

2) Accurate keypoint localization: Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

The initial implementation of this approach (Lowe, 1999) simply located keypoints at the location and scale of the central sample point. However, recently Brown has developed a method (Brown and Lowe, 2002) for fitting a 3D quadratic function to the local sample points to determine the interpolated location of the maximum, and his experiments showed that this provides a substantial improvement to matching and stability. His approach uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x \quad (1)$$

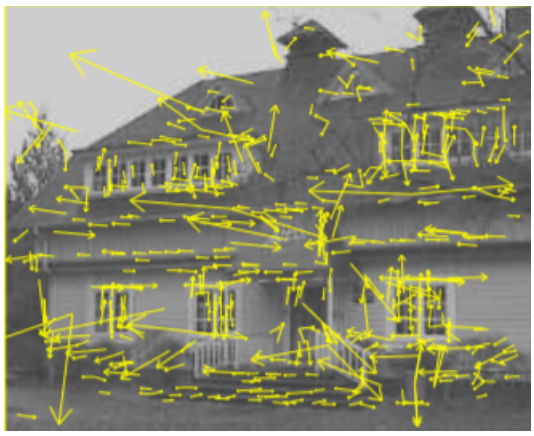

Fig. 2. This figure shows the keypoint selection. The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location.

Figure 2 shows the effects of keypoint selection on a natural image. In order to avoid too much clutter, a low-resolution 233 by 189 pixel image is used and keypoints are shown as vectors giving the location, scale, and orientation of each keypoint.

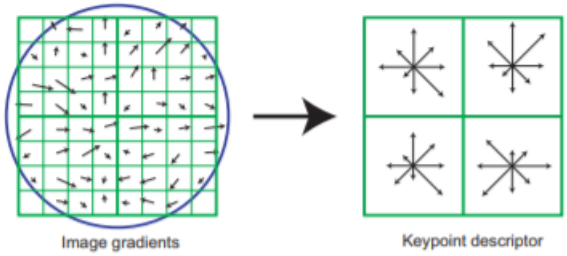

Fig. 3. A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

3) The local image descriptor: Figure 3 illustrates the computation of the keypoint descriptor. First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation

invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. For efficiency, the gradients are precomputed for all levels of the pyramid. These are illustrated with small arrows at each sample location on the left side of Figure 3.

B. Convolutional Neural Networks(CNN)

For Convolutional Neural Networks model, we will not elaborate too much here. We will implement two models:

- Self constructed model.
- VGG16 with fully connected layer designed by ourself.

IV. Experimental

To understand the effects of different distortions on classification task, we select some state-of-the-art CNN architectures that achieved impressive results in ImageNet challenge. To evaluate the models, we have developed two datasets. One of them is a synthetic digits dataset and another is a natural images dataset. After training a model with a particular dataset, we apply different image degradations, e.g., motion blur, Gaussian blur, additive noise, salt and pepper noise etc., on each image individually and measure the accuracy of individual models. Details of these datasets are as follows.

A. Dataset

1) Natural Images Dataset: State-of-the-art CNN models trained on the ImageNet dataset can correctly classify 1000 classes even in very complex environmental context. On the other hand, the behavior of capsule network is not wellexplained for natural images like ImageNet or COCO dataset. As it is difficult to train CapsuleNet with large number of classes, we compiled a dataset containing natural images to evaluate the performance of different CNNs including CapsuleNet for complex input images under various image degradations. The dataset contains 8 different classes- airplane, car, cat, dog, flower, fruit, motorbike and person having 727, 968, 885, 702, 843, 1000, 788 and 986 image samples respectively from the 8 classes. Call this dataset as 'ISINI' dataset. [10]

B. Implementation Details

1) Data processing: For the data, I generate four kinds of data. RGB64, RGB32, GRAY64 and GRAY32.

2) Model building: For the machine learning part, I deploy Linear Discriminant Analysis Logistic Regression Random Forest Decision Tree Gaussian NB 5NN, but mainly focused on the SVM model. All models are trained under 10-fold cross-validation. We also do the grid search on the SVM model and the best parameters is $C = 3$, $\gamma = 0.1$ and use rbf kernel. For the CNN models, we deploy self constructed model and VGG16 with self constructed fully connected layers.

V. Result Analysis

A. SVM Model

- RGB64 0.892
- GRAY64 0.887
- RGB32 0.830
- GRAY32 0.823

According to the results obtained, the grayscale image has basically no effect on the model, but the quality of the image has a greater impact. Why? Because features that yield shape information are based on image brightness gradients, color information is not used.in the area of object detection, color information is not useful on the HOG algorithm [11].

B. Convolutional Neural Networks Models

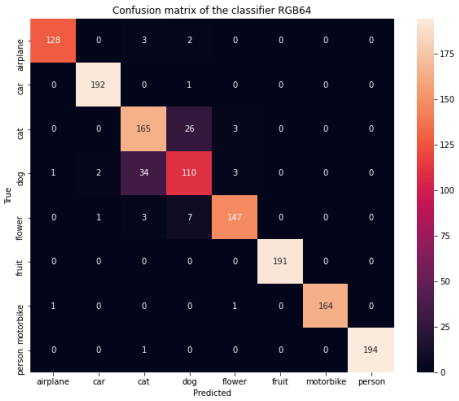

Fig. 4. Color space: RGB, Image size: 64*64

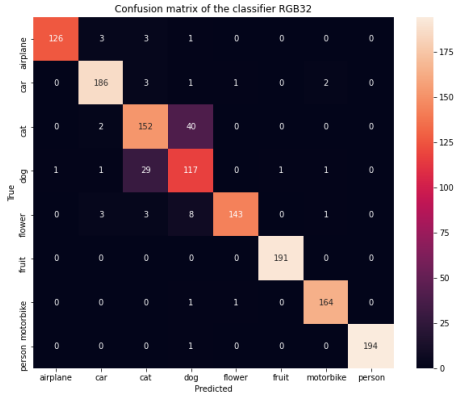

Fig. 5. Color space: RGB, Image size: 32*32

Figure 4 is base on RGB64. You can see that most error-prone classifications are dog and cat. Why? That's their much more similar, you need more information to distinguish them. Figure 5 is base on RGB32. Obviously, due to the reduction in image size, the error is magnified. Figure 6, due to the color loss, the error continues to be magnified, also infected the flower. Figure 7 is based on GRAY32, things are like the original idea, the accuracy getting worse. We also do a "cross-validation" on this, the result of VGG16 is almost the same.

VI. Conclusion

For color-insensitive Tasks, such as car plate recognition, color does not help expressing the image, just use grayscale

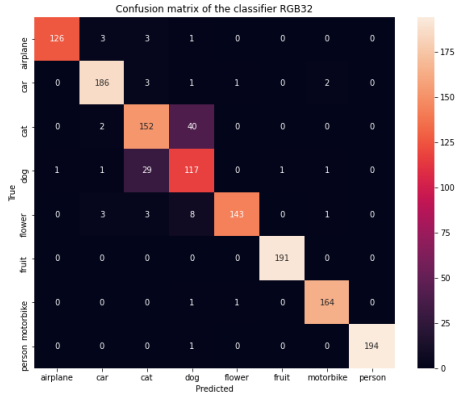

Fig. 6. Color space: GRAY, Image size: 64*64

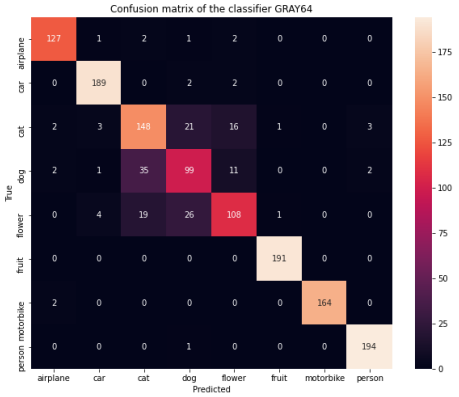

Fig. 7. Color space: GRAY, Image size: 32*32

images. For color-sensitive Tasks, Such as dog,cat or multi-object classification tasks, use RGB images. When the computing power allows, try to improve the picture quality to achieve higher accuracy. You need to balance batch size, time consumption, picture quality. It's hard to balance them.

VII. Future Works

For the future works, find the balance point between batch size, time consumption, picture quality and train a better model base on what I analyze today.

References

[1] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005 San Diego, CA, USA.

[2] Lowe, D.G., Distinctive image features from scale-invatiant keypoints, International Journal of Computer Vision (2004) 60: 91.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," International Conference on Learning Representations (ICLR), 2015.

[4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via adaptive representation," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 2, pp. 210–227, 2009.

[5] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," CVPR Workshop on Deep learning in Computer Vision, pp. 806–813, 2014.

[6] C.-X. Ren, D.-Q. Dai, and H. Yan, "Coupled kernel embedding for low-resolution face image recognition," Image Processing, IEEE Transactions on, vol. 21, no. 8, pp. 3770–3783, 2012.

[7] L. J. Karam and T. Zhu, "Quality labeled faces in the wild (qlfw): a database for studying face recognition in real-world environments," in IS and T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2015, pp. 93 940B1–93 940B10.

[8] J. Tao, W. Hu, and S. Wen, "Multi-source adaptation joint kernel sparse representation for visual classification," Neural Networks, vol. 76, pp. 135–151, 2016.

[9] S. Basu, M. Karki, S. Ganguly, R. DiBiano, S. Mukhopadhyay, and R. Nemani, "Learning sparse feature representations using probabilistic quadtrees and deep belief nets," European Symposium on Artificial Neural Networks, ESANN, pp. 367–375, 2015.

[10] P. Roy, S. Ghosh, S. "Bhattacharya, U. Pal, Effects of Degradations on Deep Neural Network Architectures" arxiv.org

[11] G. Yuhi, Y. Yuji, F. Hironobu, "CS-HOG: Color Similarity-based HOG", The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision