

An optimization method of machine learning model for image recognition by transferring images in grayscale

Bowen Xiao*

Pittsburgh Institute, Sichuan University, Chengdu, 610041, China

* Corresponding Author Email: 2022141520238@stu.scu.edu.cn

Abstract. When applying image recognition based on machine learning, TensorFlow framework based on convolutional neural network is often taken in consideration. Generally speaking, current image recognition program designed on TensorFlow are comprehensive enough with high accuracy in identifying images. However, when analyzing blurry images with less targets to recognize, existing models may not work as accurately as expected. This study aims at providing ideas for optimizing machine learning model for image recognition by transferring images in grayscale. In this article, animal data set was applied as test data to analyze since animal images are usually simple and concise which exactly meet our requirements. The study compared the prediction outcomes by the scale of accuracy. As the test showed, the proposed model archived better results using gray scaled image for classifying blurred image with the accuracy of the prediction raised from 33.03% to 94.01% in one test. The result suggested that applying grayscale images to improve the accuracy of blurry image recognition based on machine learning is a feasible and efficient proposal.

Keywords: Machine learning, Image recognition, Convolutional neural network, Grayscale.

1. Introduction

With the development of the times and technology, machine learning is constantly evolving and improving, and its applications, including language proceeding, auto-driving, data analysis etc., now play very important roles in our daily lives [1]. Among them, applying machine learning based on neural network for image recognition is one of the most commonly used functions. To be more specific, image recognition can automatically recognize objects, scenes, or features in images, and is widely used in fields such as facial recognition, vehicle recognition, medical image analysis, etc. However, although existing programs show good performance recognizing both complex images and simple ones, shortages still exist in some aspects, for instance, current image recognition models lack accuracy while analyzing fuzzy and indistinct images. Hence, this article will discuss an alternative way to improve accuracy in recognizing fuzzy and indistinct images, taking animals images set for example.

This article first introduces Convolutional neural network, TensorFlow and their shortages in the field of image recognition. Then, this article introduces the alternative research method and lists the research methods and processes. Afterwards, this article discusses the analysis of the research results and draws conclusions. In the end, the article concludes by outlining the challenges of this approach and its future prospects.

2. Basic Theory Analysis

2.1. TensorFlow

TensorFlow is an open-source machine learning framework developed by Google, aimed at implementing large-scale machine learning and deep learning applications. It has a flexible architecture that can run efficiently on various hardware devices, including CPUs and GPUs. All operations are executed by using tensor objects as the foundation. Complex neural network algorithms fundamentally comprise elementary operations including tensor multiplication and addition [2]. During execution, eager mode and graph mode are offered by TensorFlow [3]. The former mode is executed immediately while the latter mode first creates a dependency graph and

executes nodes in that graph only where needed. TensorFlow is widely used in various fields, such as natural language processing, computer vision, speech recognition, and recommendation systems, and has been widely applied and recognized in both academia and industry. Image recognition is also a main application of the TensorFlow framework. The main steps of the recognition are as follows: Firstly, it is necessary to prepare a data set containing a large number of image samples, each with corresponding labels or categories. These datasets may include millions of images, covering various categories such as animals in this essay. Next, use TensorFlow's API and libraries to build deep learning models, commonly known as Convolutional Neural Networks (CNNs). The construction of the model includes defining the structure, hierarchy, and parameter settings of the network. During the construction process, pre-trained models such as ResNet and Inception can be chosen to accelerate development speed and improve analysis accuracy. Through backpropagation algorithms and optimizers, the model gradually learns the relationship between image features and categories after the prepared data set is input into the model for training. During the training process, the model will continuously interact with the data, adjust parameters to minimize the loss function, until reaching a certain accuracy or meeting the specified convergence conditions. After the training is completed, the model needs to be evaluated. Typically, validation or testing sets are used to evaluate the performance of a model, including metrics such as accuracy, precision, and recall. By evaluating the results, we can understand the generalization ability and accuracy of the model on unprecedented data. Finally, when the model reaches a satisfactory level of performance, it can be deployed to practical applications. The deployment can be on a local device or on a cloud server, depending on the requirements. The deployed model can be used to identify new image data and solve various image classification, detection, and recognition tasks. Overall, TensorFlow achieves the task of machine learning image recognition by constructing and training deep learning models, utilizing a large amount of image data for training, and is suitable for dealing with image recognition [4]. The TensorFlow works is shown in figure 1 [5].

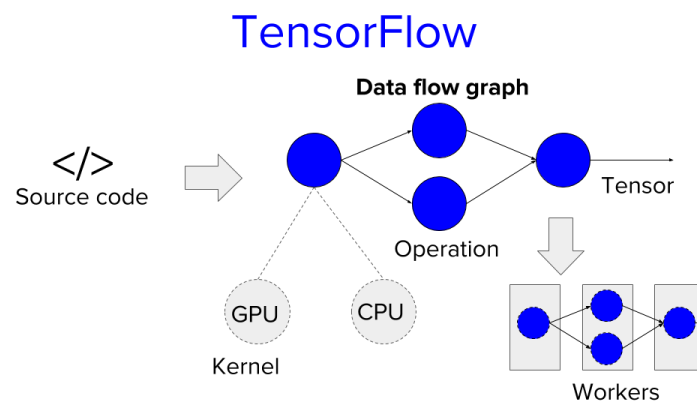


Fig. 1 TensorFlow works [5]

2.2. Convolutional Neural network

The convolutional neural network stands out as one of the most effective algorithms in extracting sophisticated semantic features. Through sequential convolution of images across layers, it retrieves increasingly abstract features, enhancing the representation of semantic information within images and consequently improving image recognition performance [6]. The basic construction of convolutional neural networks includes convolutional layers, pooling layers, and fully connected layers. In the convolutional layer, the convolutional operation slides on the input image through a filter (convolution kernel) to extract features at different positions. The pooling layer reduces the dimensionality of the feature map by down sampling, retaining important information while reducing computational complexity. The fully connected layer maps the features output by the pooling layer to the final output category. TensorFlow provides rich convolutional neural network models, such as LeNet, AlexNet, VGG, ResNet, etc., and also supports user-defined models. Users can choose

appropriate models for training and use based on application scenarios and requirements [7]. Using TensorFlow to make image recognition easier and more efficient using convolutional neural networks.

2.3. Shortage of TensorFlow

Although TensorFlow has made significant progress in image recognition, there are still some shortages. One of them is the challenge facing small sample data set, where models often struggle to achieve sufficient generalization ability in limited data, resulting in poor performance on unseen data. Another challenge is data bias and imbalance, where the number of samples for certain categories in the training data far exceeds that of others, leading to a decrease in the accuracy of the model in identifying minority categories. This may lead the model to lean towards predicting categories with higher frequency of occurrence, while ignoring categories with lower frequency of occurrence. Except from these two issues, error caused by blurred boundaries while processing images is also a main challenge. In the real world, the boundaries of images may not be clear, which may be caused by limitations in image acquisition equipment, low image resolution, or poor image quality [8, 9]. Due to the fact that convolutional neural network usually come into contact with clear images during the training process, there may be misjudgments for images with blurred boundaries, leading to recognition errors. The image recognize program fail to classify the given elephant image, only 33% in accuracy. Besides, when processing images, the model may produce incorrect classification results due to slight rotation, displacement, or noise. This is because pictures may sometimes lack sufficient complexity, making it difficult for models to accurately capture these small changes. In addition, images may be affected by factors such as lighting conditions or shooting angles, which further increasing the difficulty of image recognition. Generally speaking, to settle these challenges, deeper networks or more convolutional layers can be used can be added to enhance the model's representation ability, in order to improve its ability to capture subtle features and background noise. Additionally, integrating the prediction results of multiple models, such as voting or averaging, to reduce the bias of individual models and improve overall recognition accuracy is also widely applied. However, these solutions are somehow complex or expensive to apply meaning that these are not the best option to choose. Hence, finding an alternative optimization method is essential.

3. Research

3.1. Optimization method

In order to achieve higher accuracy, an optimization method is proposed. The main idea of this optimization method is to expand the main features by transferring the color in the image to grayscale. Additionally, to make further researches, a new control group is set. In this group, after the image is grayscale, the program pixelates the image to see if the accuracy rises. To be more specific, before executing image recognition programmed, the color of the images is turned gray and pixelated to extract their rough contours, which is the main idea of improving the accuracy of image recognition.

3.2. Procedure

The research method is quite simple. We first set up a pre-trained model with various types of animal's images by training the model with the animal images data set we provide [10]. Then we insert a new image with animal inside. Before we execute the classification, we first turn the image to grayscale by the code: `"plt.imshow (img, cmap='gray')"`. This line of code utilizes the `imshow` function from the Matplotlib library to display an image. Here, `img` represents the image data to be displayed, and the parameter `cmap='gray'` specifies that the image should be shown using a grayscale color map. This means the image will be displayed in grayscale rather than in color. After generating the grayscaled image, do one more step further by code `"img_small = (resize (img, resolving power, preserve_range=True) / 2) + 120"` and generate a new pixelated image. The flow chart showing the procedure is shown in figure 2.

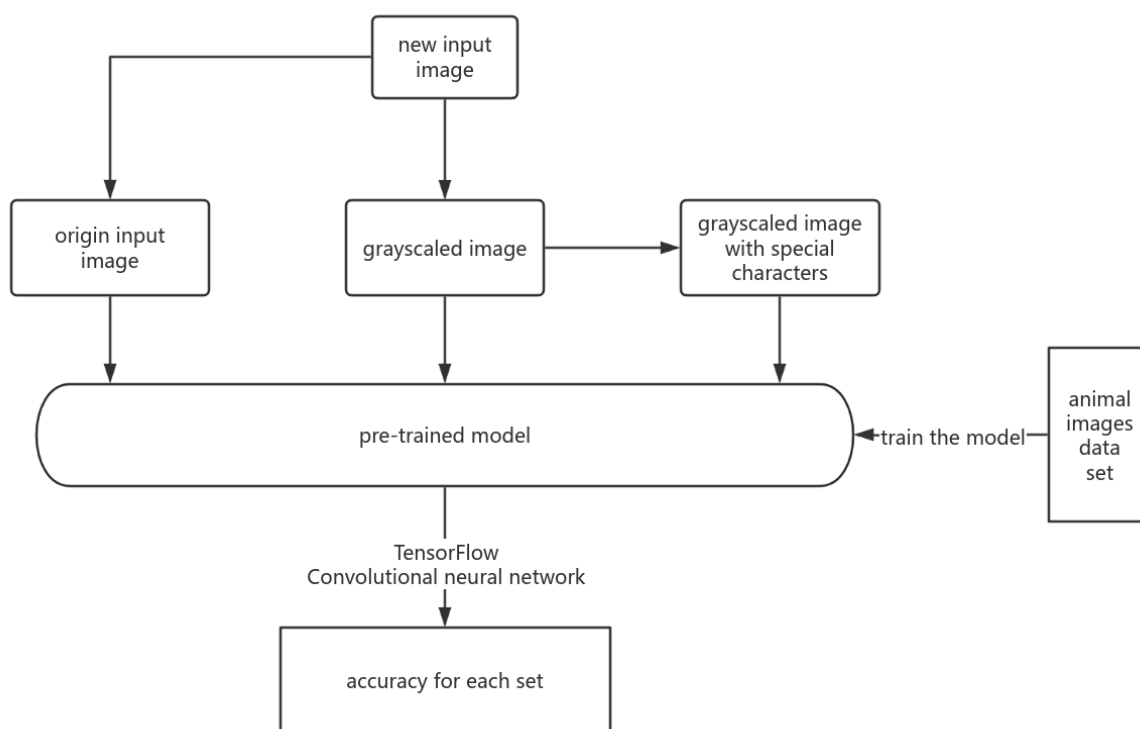


Fig. 2 Flow chart showing the procedure (Photo/Picture credit: Original)

By now, we have three different images with us: the origin input image, the grayscale image, and the pixelated grayscale image. Leave the programmed run and analyze to make the classification, the accuracy rates for each set should then be presented. The three to-be-tested images is shown in figure 3.



Fig. 3 The three to-be-tested images (Photo/Picture credit: Original)

3.3. Analysis

Due to the noise in the image, in the first test, the accuracy rate in the first test is only 33.03%, whereas the rate for the second test rises to 94.01%. This proves that grayscale images have advantages in recognizing and classifying such blurred images. Yet, in the third test, the accuracy rate falls to 26.02%. This is probably because the contour of the target is less clear. This suggests that pixelated images' performance is not as good as normal images' in recognizing and classifying images.

4. Conclusion

In conclusion, the method described offers a unique approach to image preprocessing and analysis by using pixelated grayscale images. The results suggest that while grayscale images can enhance the clarity of image contours and improve classification accuracy for simpler images, their performance diminishes when faced with more complex scenes. This indicates the importance of considering the complexity of the images and the information required for accurate classification when choosing preprocessing methods. Further research could explore alternative preprocessing techniques or model architectures to improve the performance of image analysis systems on complex datasets.

Though the research shows exciting results, there are still challenges and opportunities for future exploration in the realm of image recognition based on machine learning. One notable challenge lies in the limitations of grayscale images, particularly when faced with more complex scenes or intricate patterns. While grayscale images may enhance the clarity of image contours and improve classification accuracy for simpler images, their performance diminishes when dealing with intricate details or varied textures. Moreover, when analyzing clearer images in high quality, using this method does not meet our expectations perfectly, on the contrary, it actually reduces the accuracy, which makes this processing method limited while applying.

Looking ahead, the future prospects for image recognition based on machine learning remain promising. Continued advancements in machine learning algorithms and computational capabilities offer opportunities to develop more sophisticated models that can accurately classify a wide range of images, from simple to complex. Additionally, exploring interdisciplinary collaborations and incorporating insights from fields such as computer vision and cognitive science could lead to innovative approaches for improving image recognition systems.

References

- [1] Cui Tengheng, Liu Jiping, Luo An. An intelligent recognition method for map images based on Convolutional neural networks. *Science of Surveying and Mapping*, 2019, 44(1): 118-123.
- [2] Long L, Zeng X. *Beginning deep learning with TensorFlow: work with Keras, MNIST data sets, and advanced neural networks*. 2022.
- [3] Jiang Z, Xu D, Liang J. A deep reinforcement learning framework for the financial portfolio management problem. *arxiv preprint arxiv:1706.10059*, 2017.
- [4] Zhang Minmin, Xu Heping, Wang Xiaojie, et al. Google TensorFlow machine learning framework and Application. *Microcomputers and Applications*, 2017, 36(10):3.
- [5] Abadi M. TensorFlow: learning functions at scale. *Proceedings of the 21st ACM SIGPLAN international conference on functional programming*. 2016: 1-1.
- [6] Yan He, Wang Peng, Dong Yingyan, et al. Improved Convolutional neural network image classification and recognition method. *Journal of Computer Applications and Software*, 2018, 35(12): 193-198.
- [7] Yu Q, Shi C. An image classification approach for painting using improved convolutional neural algorithm. *Soft Computing*, 2024, 28(1): 847-873.
- [8] Zhang Shun, Gong Yihong, Wang Jinjun. Development of deep convolutional neural networks and their applications in computer vision. *Journal of Computers*, 2019, 42(3): 453-482. (in Chinese)
- [9] Uchida S. Image processing and recognition for biological images. *Development, growth & differentiation*, 2013, 55(4): 523-549.
- [10] Sanghvi K, Aralkar A, Sanghvi S, et al. Fauna image classification using convolutional neural network. *International journal of future generation communication and networking*, 2020, 13(1s): 08-16.