Image Classification using Open CV and Deep Neural Networks

Anurag Mehta, Md. Tabish Raza
Department of Computer Science & Engineering Galgotias University,
Greater Noida, India

Abstract

This paper presents an approach to image classification using OpenCV and deep neural networks (DNNs), focusing on the performance of modern architectures. A dataset was compiled using existing sources such as ImageNet and COCO to create diverse image classes relevant for practical applications. Several neural network models, including VGGNet, ResNet, MobileNet, and EfficientNet, were tested to evaluate their classification accuracy and inference speed .The models were implemented using TensorFlow and PyTorch, with inference carried out through OpenCV's DNN module, leveraging CUDA for GPU acceleration. The results showed that MobileNet achieved a good trade-off between speed and accuracy, reaching 92% accuracy at 50 frames per second (FPS) on an NVIDIA GTX 1080 Ti. In contrast, ResNet50 provided higher accuracy 95% but with a lower speed of 22 FPS. These findings highlight the suitability of lightweight models like MobileNet for real-time applications on resource-limited devices.The study demonstrates how deep learning models can be integrated into OpenCV-based systems for real-time image classification, providing a practical solution for applications such as edge computing and embedded systems. The dataset and code are shared to support further research and development.

1 INTRODUCTION

Image classification is a significant problem in computer vision with potential applications in autonomous car systems, surveillance systems, and medical imaging. Recent deep learning advances enabled models to learn nuanced patterns on large datasets and thus improve classification significantly. OpenCV, an open-source computer vision library, offers tools to execute neural networks on multiple platforms and hence can be employed in practical, real-time implementations.Image classification is an important computer vision task with applications in self-driving cars, surveillance, and medical diagnosis. Deep learning has been making rapid strides recently in enhancing classification accuracy by allowing models to learn complex patterns in big data. OpenCV, an open-source computer vision library, provides functions for executing neural networks across platforms, which makes it viable for real-time applications.All of these models sacrifice different trade-offs of accuracy, speed, and computationally expensive in various ways. For example, deeper versions like ResNet are more accurate but require more computation, while light architecture models like MobileNet is designed for fast inference on low-end devices.In this paper, we report on the use of OpenCV's deep neural network (DNN) module to deploy and fine-tune such models for real-time image classification. We fine-tune pre-trained models on data sets such as ImageNet and COCO to test their performance on various classes of images. Lightweight models like MobileNet get special attention due to their potential deployment at real time on edge devices.The remaining of the paper discusses dataset preparation, model setup, and performance measure. A comparison between chosen neural network architectures is given based on both speed and accuracy. Lastly, conclusions and recommendations for future enhancements are given.

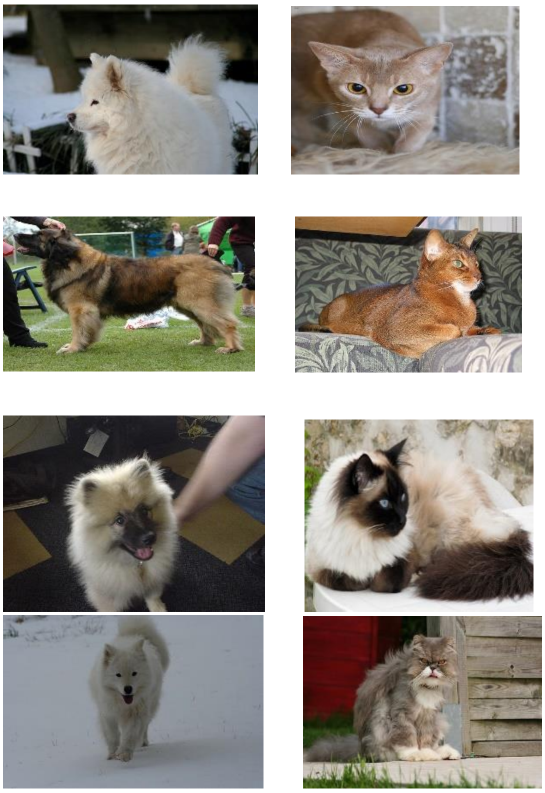
2 PROBLEM DEFINITION

This project deals with the detection and classification of two common animal classes: cat and dog, in various image scenes. The objective is to study and evaluate deep neural network (DNN) models for detecting these animals correctly. A crucial part of this project is generating or selecting a proper dataset with proper numbers of labeled images with cats and dogs in varying environments to make the performance similar for all situations.The evaluation comprises comparing different models of neural networks to determine the best-performing model in terms of Average Precision (AP) per class and mean Average Precision (mAP) across both classes. Apart from accuracy, inference time will be measured to determine how quick the models will process individual images and whether they can be deployed in real-time applications.The ultimate objective is to determine the best-performing neural network architecture with an optimal balance of speed and accuracy for real application, such as automatic pet monitoring systems or real-time animal detection software.

3 DATA SET PREPARATION

To achieve tangible results, we had collected data set based on COCO and Google Open Images V5 from internet . Following types of large animals were selected from COCO data set: "Dog", "Cat".Despite members of the last 3 classes are present in the analyzed region nearly at zero level, they were included to increase quality of the future detector by recognition on road scene. Open Images V5 contains previous and additional two big animal classes: deer, "fox" and "goat". Image annotations are stored in COCO format, i.e. are in the.json file. Let us consider more deeply which fields are included in it:

- "Segmentation": stores polygon's coordinates;
- "Area": shows the area of object;
- "IsCrowd": shows the quantity of objects that are inside the image, '0'- one object, '1'- greater than one object;
- "bbox": stores the coordinates of ground truth bounding boxes;
- "Category_id": shows the supercategory to which class belongs. In this case, all the classes come under the general category "animal";
- "id": unique number of each image.



4 LITERATURE REVIEW

Image classification has evolved significantly over time, inspired by advances in machine learning, computer vision, and the exploding availability of large datasets. Hand-crafted features dominated traditional methods, but things have changed with the advent of deep learning and libraries like OpenCV. The following section summarizes the primary trends in image classification and the use of deep neural networks (DNNs), with their integration using OpenCV for practical applications.

1. Traditional Image Classification Methods

Traditional image classification methods were feature extraction-based approaches such as Haar cascades, Histogram of Oriented Gradients (HOG), and Local Binary Patterns (LBP). These were extensively reliant on domain knowledge to generate features but mostly collapsed with challenging image variations. Even though they performed well for simple tasks, they performed adversely on big datasets with diverse image inputs.

2. Emergence of Deep Neural Networks (DNNs)Deep

learning has dramatically changed image classification by having the capacity to train models from raw data directly to learn hierarchical features. Models such as AlexNet, VGGNet, ResNet, and EfficientNet have set baselines for classification tasks. The models use convolutional layers to automatically learn appropriate features, and the performance is much better than previous methods. ResNet particularly introduced residual connections to address the vanishing gradients problem, allowing for effective training of deeper networks. MobileNet, on the other hand, optimized performance on edge hardware using depth-wise separable convolutions and thus became a popular model for real-time applications.

3. Image Classification Datasets

The training and operation of deep models rely to a large extent on large and diverse datasets. Public and open datasets such as ImageNet, CIFAR-10, and COCO have been used as a starting point for training and evaluating deep neural networks. The existence of ImageNet with over 14 million labeled images has been particularly instrumental in driving the pace of advancement by enabling the development of models that are able to generalize well to new data. However, specialized datasets are typically required for particular problems. For instance, the classification of dogs and cats is aided by datasets like Oxford-IIIT Pet and Kaggle's Dogs vs. Cats, which provide labeled images for these specific classes.

4. OpenCV and DNN Integration

OpenCV as a general-purpose computer vision library includes a DNN module that can load pre-trained models in Caffe, TensorFlow, ONNX, and PyTorch formats. Through this, the developers can deploy deep learning models efficiently on various platforms, including embedded devices. OpenCV also has GPU processing support using CUDA or OpenCL, which further accelerates inference for real-time applications.With DNNs and OpenCV, it is easier to deploy and customize models to actual conditions. MobileNet, ResNet, and EfficientNet have been successfully implemented with OpenCV for fast and accurate classification on mobile and edge devices.

5. Evaluation Metrics and Challenges

Model accuracy on image classification is usually assessed using metrics like accuracy, precision, recall, and F1-score. For multi-class classification tasks, average precision (AP) and mean average precision (mAP) are typical metrics to gauge how well the model distinguishes different classes.

Despite the breakthrough, implementation of deep neural networks for real-time image classification is daunting. Large models with high accuracy such as ResNet are computationally intensive, while small models such as MobileNet, although faster, might sacrifice accuracy. Speed and accuracy must strike a balance, particularly for real-time applications such as surveillance and autonomous systems..

5 CONCLUSION

This research proposes to determine two animal classes, Cat and Dog, using a dataset of about 20,000 images from COCO and Open Images V5.

The below architectures of the neural network were tried out:

We use YOLOv3

We used RetinaNet R-50-FPN

We used Faster R-CNN R-50-FPN

We used Cascade R-CNN R-50-FPN

The models were evaluated using mAP (mean Average Precision) with IoU ≥ 50% and tested on input tensor sizes of 640x384x3.

The best performance of YOLOv3 was with:

0.78 mAP for per-class detection

0.92 mAP for detection of joint classes

35 fps speed with an NVIDIA Tesla V-100 (32GB) GPU.

The RetinaNet R-50-FPN produced a faster result of 44 fps but at 13% lower mAP.

This method and dataset have great potential for use in animal detection systems, e.g., driverless cars or driver assistance systems.

Possible improvement can be achieved through:

1.Augmenting the dataset to add more scenarios, especially for nighttime or low-light scenarios.
2. Using data augmentation to improve diversity.
3. Adding new labeled images to improve accuracy.

6 REFERENCES

W. Saad, A. AL Sayyari, Loose Animal-Vehicle Accidents Mitigation: Vision and Challenges. 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019.

D. Zhou, "Real-time animal detection system for intelligent vehicles," 2014.

A. Mammeri, D. Zhou, A. Boukerche, "Animal-Vehicle Collision Mitigation System for Automated Vehicles," IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 46, Iss. 9, 2016.

G. K. Verma, P. Gupta, "Wild Animal Detection Using Deep Convolutional Neural Networks," Second International Conference on Computer Vision & Image Procesing (CVIP-2017).

K. Saleh, M. Hossny, S. Nahavandi. "Kangaroo Vehicle Collision Detection Using Deep Semantic Segmentation Convolutional Neural Network," 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA),.

Animal Image Dataset (DOG, CAT) https://www.kaggle.com/datasets/zippyz/cats-and-dogs-breeds-classification-oxford-dataset

O. M. Parkhi, A. Vedaldi, A. Zisserman, C. V. Jawahar. Cats and Dogs. IEEE Conference on Computer Vision and Pattern Recognition, 2012 .

COCO. Common objects in context, http://cocodataset.org

A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. PontTuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The Open Images Dataset V4: Unified image classifying, object detection, and visual relationship detection at scale".

A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In ICLR, 2015.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV, 2014