Object Detection in Pytorch Using Mask R-CNN

Tobi Makinde
Computer Science and Quantitative method
Austin Peay State University
Clarksville, Tennessee, United States of America

Abstract:- This research paper aims to investigate the idea of object detection in PyTorch employing the most widely known object detection and localization algorithm that employs image segmentation techniques and deep learning approach, which is Mask Region-based Convolutional Neural Network. Mask R-CNN is widely used in many fields, such as industrial and medical applications, due to its ability to accurately identify objects and generate segmentation masks for each instance. The Mask R-CNN algorithm combines the region proposal generation and object classification stages of Faster R-CNN with an additional branch for pixel-level segmentation.

Keywords:- *Convolutional Neural Network, Object Detection, Pre-trained Model, PyTorch, Object Detection, Image Preprocessing, Pandas, NumPy, Pretrained Model, Mask Region-Based Convolutional Neural Network.*

I. INTRODUCTION

The ability to identify and characterize objects in an image or video is one of the primary functions of computer vision. Many applications, including autonomous driving, robotics, image understanding, and surveillance systems, depend on accurate object detection. In the past couple of years, computer vision has advanced dramatically, particularly regarding object detection techniques. Object detection algorithms combine the tasks of object localization and image classification to identify and locate objects within an image or video. These algorithms achieve precise and effective object detection by leveraging deep learning techniques. The R-CNN family, which includes Mask R-CNN, Fast R-CNN, and Faster R-CNN, is one well-liked family of object detection algorithms [6].These algorithms have gained considerable attention and have been widely used in various fields due to their superior performance and versatility.

A. Mask R-CNN: An Extension of Faster R-CNN

Mask R-CNN is an extension of the Faster R-CNN algorithm, which has been a significant breakthrough in the field of object detection. The idea of region-based convolutional neural networks for object detection was first presented by Faster R-CNN [3]. The creation of region proposals and object classification are its two phases. Using a neural network approach to forecast potential object locations, regions of interest are put in place in the first stage of Faster R-CNN. The convolutional neural network created have the regions that are categorized into distinct object categories in the second stage. Faster R-CNN, however, does not offer pixel-level segmentation; instead, it only offers bounding box information for object localization. Due to this restriction, Mask R-CNN was created [7]; an addition to Faster Region-Based Convolutional Neural Network that creates a supplementary branch needed for object segmentation mask generation. The Mask R-CNN technique aims to address the insufficiency of the Faster Region-Based Convolutional Neural Network by integrating instance segmentation features. [1]. Implementing Mask Region-based Convolutional Neural Network therefore makes it possible to obtain pixel-wise segmentation masks for each object in an image [2]

B. Using PyTorch for Object Detection

A popular deep learning library called PyTorch offers an easy-to-use interface for developing training and object detection models. It supports usage of several previously trained models, and development of various machine learning algorithms including Mask Region-Based CNN. Using PyTorch, researchers can quickly install and configure the Mask R-CNN model for object detection. Additionally, the Mask R-CNN implementation in PyTorch enables model customization and fine-tuning using new datasets for particular instance segmentation tasks. Researchers can easily incorporate the Mask R-CNN algorithm into their object detection pipeline by utilizing PyTorch's capabilities. In order to apply Mask R-CNN in PyTorch, researchers must take the subsequent actions:

- Design the neural network architecture for Mask R-CNN by combining the networks for feature extraction, region proposal, instance detection, and segmentation. This can be achieved by leveraging the power of PyTorch's modular design, which allows researchers to easily define and customize the different components of the Mask Region-Based CNN architecture.
- Prepare the data by loading the dataset and transforming it into a format compatible with PyTorch's DataLoader.
- Implement the necessary data augmentation techniques to increase the diversity and robustness of the training dataset.
- This can include techniques such as random cropping, rotation, and flipping of images to introduce variations in object appearance and enhance the model's generalization.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1069

- Using the loaded dataset, adjust the model that is pre-trained. Researchers can use transfer learning to adjust the already trained Mask R-CNN so that it is more suited to their particular instance segmentation task.
- Examine the trained Mask R-CNN model's output on the validation set to gauge its precision and capacity for generalization.

II. LITERATURE REVIEW

The Mask R-CNN algorithm has become a major player in the object detection space because it can reliably localize objects and create pixel-wise masks [2].There are numerous applications for the Mask R-CNN algorithm, including those in the industrial and medical domains. - Mask R-CNN has been applied to medical tasks like tumor detection and segmentation, where precise abnormality localization is essential for diagnosis and treatment planning. Furthermore, the application of Mask R-CNN in industrial settings has demonstrated potential for jobs like quality assurance and defect identification. Our goal in this work is to investigate Mask R-CNN's object detection and localization. To fulfill our research objective, we employed the Mask R-CNN. The Mask R-CNN algorithm, an extension of Faster R-CNN, utilizes deep learning and image segmentation techniques to achieve pixel-level object detection and segmentation at a high level of accuracy. The first step of implementing the Mask R-CNN algorithm involves object detection, which is an essential characteristic of the task. To accomplish this, we utilized the proposed approach of using the Mask R-CNN neural network architecture. As a cutting-edge convolutional neural network, the Mask R-CNN neural network architecture is ideal for object detection and instance-based segmentation in image segmentation processing. Faster R-CNN, a region-based convolutional neural network with an object detection focus, is the foundation of this architecture [3]. One popular technique in the field of object detection and localization is the Mask R-CNN algorithm [1]. For object detection, pre-trained models such as DenseNet [4], Google Net, and Resnet [5] can be employed.

III. DATA

The dataset used was acquired from Kaggle, it has 8,434 images of different four different categories of car, which are are suv, sedan, Minivan and Convertible. All the images are of the size (160 × 240). The dataset is divided into three broad categories: training, validation and test.

Fig 1: Image of car (Sedan)

Fig 2: Image of Car (SUV)

Fig 3: Image of Car (Minivan)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1069

Fig 4: Image of Car (Convertible)

The dataset used for training consists of 4,957 images. The dataset for validation consists of 2,887. The dataset used for testin consists of 590 images. The data preprocessing steps applied to the dataset are:

- *Cropping*

The different parameters needed to crop the image were defined using the PyTorch transform sub module in order to facilitate the extraction of features.

- *Horizontal Flipping*

The image was flipped using an interval of [-90, 90].

- *Resizing*

An image size of 224 served as the basis for the resizing; this value was likewise specified in the PyTorch transform sub module.

- *Normalizing*

The picture was next converted into a tensor image after being normalized using a stipulated Pytorch Mean and Standard Deviation of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225], respectively.

IV. METHODS

There are multiple crucial steps in the suggested Mask R-CNN object detection method in PyTorch. Feature extraction, region-proposal, instance detection, and segmentation networks are combined to create the first step of the neural network architecture for Mask R-CNN. In order to achieve precise and effective object detection results, this architecture is essential. The next step is to prepare the data by loading it into a format that PyTorch's DataLoader can read by loading the dataset. To improve the training dataset's robustness and diversity, data augmentation techniques are also used. To improve the model's generalization, these techniques involve randomly cropping, rotating, and flipping images to introduce variations in object appearance. Following the preparation of the data, the pre-trained Mask R-CNN model is fine-tuned using the loaded dataset. By fine-tuning the pre-trained Mask R-CNN model, researchers can leverage transfer learning to adapt the model to their specific instance segmentation task. To evaluate the performance of the trained Mask R-CNN model, it is essential to conduct an assessment on a separate validation set. In this assessment, the accuracy and generalization capabilities of the implemented Mask R-CNN model are evaluated.

V. IMPLEMENTATIONS AND EXPERIMENTS

Initially, the car images were resized using the transform sub module from torchvison from 160 X 240 to 224 X 224. After that, it was turned into a tensor image by rotating it in a range of [-90, 90] and flipping it horizontally. After that, the pictures were normalized using a specified mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225].

After the dataset was loaded, a batch size of eight was used. The cross entropy loss was used as the surrogate soft max classifier in a newly added layer. The Adam optimizer was defined with a 0.002 learning rate.

The model weights are updated, the gradients are reset to zero, and the loss and gradients are computed for each batch of loaded training data during the thirty training epochs. The training loss for every epoch is also measured. Our model is evaluated using the validation dataset; to do this; we switch off auto grading and put the model in an evaluation mode. The number of accurate validation predictions is determined, along with the computation of the total loss.

VI. RESULTS

Parameters including accuracy, precision, recall, and F1 score were examined in the analysis of the obtained results. After training the model for thirty epochs, the average validation accuracy was 91.2%, while the average training accuracy was 87.6%.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1069

Table 1: Classification Report

	Precision	Recall	F1-Score	Accuracy
SUV	**0.814**	**0.781**	**0.890**	**0.825**
SEDAN	**0.750**	**0.717**	**0.672**	**0.876**
CONVERTIBLE	**0.961**	**0.527**	**0.897**	**0.954**
MINIVAN	**0.992**	**0.899**	**0.871**	**0.927**

VII. DISCUSSIONS AND CONCLUSIONS

This project describes an object detection model that uses Mask R CNN in PyTorch to detect different images of car. Preprocessing methods such as resizing, flipping the horizontal axis and normalization were used to optimize the model. The model is trained for 30 epochs with an Adam optimizer and a learning rate of 0.002, with a batch size of 8. The proposed model demonstrated effectiveness of 91.2% accuracy, 89.6% F1-Score, 90.4% precision, and 88.7% recall.

In conclusion, there are a number of benefits leveraging PyTorch's Mask Region-based CNN for object detection and instance segmentation, including its cutting-edge performance, adaptability, and accessibility to pre-trained models for transfer learning. The prospects for object detection using Mask R-CNN in PyTorch are bright given the ongoing developments in machine learning and computer vision. Researchers can investigate developments in neural network architectures, such as adding new backbone networks or attention mechanisms, to further improve the precision and effectiveness of object detection. Enhancements can also be achieved by fine-tuning the model on datasets specific to a given domain and optimizing the Mask R-CNN hyper-parameters. Future prospects of object detection using Mask R-CNN in PyTorch are also influenced by the availability of large-scale training datasets for the network. Scholars may investigate the utilization of publicly accessible datasets, like COCO or Pascal VOC, that offer annotated examples for training and assessment. Additionally, by adding segmentation masks and bounding box annotations to photos, researchers can also create their own datasets. This allows them to customize the training data to meet their unique needs and enhance the model's performance on their target objects or scenarios.

REFERENCES

[1]. Widiyanto, S., Nugroho, D. P., Daryanto, A., Yunus, M., & Wardani, D. T.. (2021, January 1). Monitoring the Growth of Tomatoes in Real Time with Deep Learning-based Image Segmentation. https://scite.ai/reports/10.14569/ijacsa.2021.0121247.

[2]. Kim, J., Kwon, S., Fu, J., & Park, J. (2022, October 14). Hair Follicle Classification and Hair Loss Severity Estimation Using Mask R-CNN. https://scite.ai/reports/10.3390/jimaging8100283.

[3]. Islam, M. N., & Paul, M.. (2021, October 15). Video Rain-Streaks Removal by Combining Data-Driven and Feature-Based Models. https://scite.ai/reports/10.3390/s21206856

[4]. G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, ‘‘Densely connected convolutional networks,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261–2269.

[5]. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1-9). June 2015.

[6]. Thomas, E. A., Gerster, S., Jean, H., & Oates, T.. (2020, October 26). Computer vision supported pedestrian tracking: A demonstration on trail bridges in rural Rwanda. https://scite.ai/reports/10.1371/journal.pone.0241379

[7]. Su, Peifeng, J. (2022, January 25). New particle formation event detection with Mask R-CNN. https://scite.ai/reports/10.5194/acp-22-1293-2022